

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN



ĐỀ CƯƠNG BÀI GIẢNG

ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

TRÌNH ĐỘ ĐÀO TẠO: **ĐẠI HỌC CHÍNH QUY**
NGÀNH ĐÀO TẠO: **CÔNG NGHỆ THÔNG TIN**
(INFORMATION TECHNOLOGY)

Hưng Yên, năm 2020

MỤC LỤC

MỤC LỤC	1
BÀI 1. KHÁI NIỆM VỀ CHẤT LƯỢNG PHẦN MỀM.....	5
1.1.Đặc điểm của phần mềm và môi trường phát triển phần mềm.....	5
1.2.Khái niệm phần mềm.....	8
1.3.Lỗi phần mềm và phân loại nguyên nhân gây ra lỗi phần mềm.....	9
1.3.2. Nguyên nhân gây ra lỗi phần mềm.....	10
BÀI 2. CÁC YẾU TỐ CHẤT LƯỢNG PHẦN MỀM.....	13
2.1.Định nghĩa chất lượng phần mềm và đảm bảo chất lượng phần mềm.....	13
2.2.Những mục tiêu đảm bảo chất lượng phần mềm.....	13
2.3.Phân loại yêu cầu phần mềm ứng với các yếu tố chất lượng phần mềm.....	14
BÀI 3. THẢO LUẬN VỀ CHẤT LƯỢNG PHẦN MỀM.....	19
BÀI 4. RÀ SOÁT HỢP ĐỒNG	22
4.1.Rà soát hợp đồng.....	22
4.1.1.Tiến trình rà soát hợp đồng và các bước thực hiện	22
4.1.2.Các mục tiêu rà soát hợp đồng	23
4.1.3. Thực thi rà soát hợp đồng.....	26
4.1.4. Những khó khăn của thực hiện xem lại hợp đồng cho các đề xuất chính..	28
4.1.5.Khuyến cáo cho việc thực hiện duyệt lại những hợp đồng chính	28
4.1.6.Các đối tượng rà soát hợp đồng.....	29
4.1.7.Rà soát hợp đồng cho các dự án nội bộ.....	29
BÀI 5. CÁC KẾ HOẠCH PHÁT TRIỂN VÀ KẾ HOẠCH CHẤT LƯỢNG.....	32
5.1.Các kế hoạch phát triển và kế hoạch chất lượng	32
5.1.1.Những mục tiêu của kế hoạch phát triển và kế hoạch chất lượng.....	32
5.1.2.Các thành phần của kế hoạch phát triển.....	32
5.1.3.Các thành phần của kế hoạch chất lượng.....	36
BÀI 6. THẢO LUẬN VỀ KẾ HOẠCH CHẤT LƯỢNG – SQA plan.....	43
BÀI 7. CÁC KỸ THUẬT RÀ SOÁT	43
7.1.Tích hợp các hoạt động chất lượng trong vòng đời dự án	43
7.1.1.Phương pháp phát triển phần mềm truyền thống và các phương pháp khác	43
7.1.2.Các yếu tố ảnh hưởng hoạt động đảm bảo chất lượng phần mềm.....	52
7.1.3.Xác minh, thẩm định và đánh giá chất lượng.....	53
7.2.Rà soát.....	54
7.2.1.Mục tiêu rà soát	54
7.2.2.Những rà soát thiết kế hình thức	54
7.2.4.Các ý kiến của chuyên gia.....	59
BÀI 8. CÁC YẾU TỐ KHÁC TRONG ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM	62
8.1. Đảm bảo chất lượng của các thành phần bảo trì phần mềm	62
8.1.1. Giới thiệu.....	62
8.1.2. Cơ sở cho chất lượng bảo trì cao.....	64
8.1.2.1. Cơ sở 1: Chất lượng gói phần mềm.....	64
8.2.1.2. Cơ sở 2: Chính sách bảo trì	66

8.1.3. Các thành phần chất lượng phần mềm tiền bảo trì	68
8.1.3.1. Xem xét lại hợp đồng bảo trì	68
8.1.3.2. Lập kế hoạch bảo trì	71
8.1.4. Các công cụ đảm bảo chất lượng bảo trì phần mềm	73
8.1.4.1. Công cụ SQA cho bảo trì sửa lỗi.....	73
8.1.4.2. Các công cụ SQA cho bảo trì cải thiện chức năng.....	75
8.1.4.3. Các thành phần cơ sở hạ tầng SQA cho bảo trì phần mềm.....	76
8.2. Các CASE tool và ảnh hưởng của nó lên chất lượng phần mềm.....	84
8.2.1. Khái niệm CASE tool.....	84
8.2.2. Đóng góp của CASE tool cho chất lượng sản phẩm phần mềm	87
8.2.3. Đóng góp của CASE tool cho chất lượng bảo trì phần mềm	91
8.2.4. Đóng góp của CASE tool cho quản lý dự án.....	92
8.3. Đảm bảo chất lượng phần mềm của các yếu tố bên ngoài cùng tham gia	93
8.3.1. Những thành phần bên ngoài đóng góp vào dự án phần mềm.....	93
8.3.2. Rủi ro và lợi ích của giới thiệu người tham dự ngoài.....	94
8.3.3. Những mục tiêu đảm bảo chất lượng về sự đóng góp người tham gia bên ngoài	96
8.3.4. Các công cụ đảm bảo chất lượng những đóng góp của các thành viên đóng góp bên ngoài	96
BÀI 9. CÁC THÀNH PHẦN CƠ BẢN CỦA CHẤT LƯỢNG PHẦN MỀM.....	98
9.1. Thủ tục, chỉ dẫn và các thiết bị hỗ trợ chất lượng	98
9.1.1. Các thủ tục và chỉ dẫn	98
9.1.1.1. Khái niệm thủ tục (procedure) và chỉ dẫn (instruction)	98
9.1.1.2. Sự cần thiết của các thủ tục và chỉ thị công việc.....	99
9.1.1.3. Các thủ tục và sổ tay thủ tục.....	100
9.1.1.4. Chỉ thị công việc và sổ tay chỉ thị công việc	101
9.1.2. Chuẩn bị, thực thi và cập nhật các thủ tục và chỉ dẫn	101
9.1.2.1. Quá trình chuẩn bị cho các thủ tục mới.....	101
9.1.2.2. Thực thi các thủ tục mới hoặc các thủ tục được sửa đổi	102
9.1.2.3. Cập nhật thủ tục.....	103
9.1.3. Khuôn mẫu (templates)	103
9.1.3.1. Đóng góp của các khuôn hình đối với chất lượng phần mềm.....	104
9.1.3.3. Sự chuẩn bị cho những khuôn mẫu mới.....	105
9.1.3.4. Áp dụng các khuôn mẫu	106
9.1.3.5. Cập nhật các khuôn mẫu	107
9.1.4. Danh mục kiểm tra (Checklists).....	108
9.1.4.1. Những đóng góp của checklists để phần mềm có chất lượng	108
9.1.4.2. Cấu trúc framework cho việc chuẩn bị, thực thi và cập nhật các checklist	109
9.1.4.3. Sự chuẩn bị cho những checklist mới	109
9.1.4.4. Xúc tiến sử dụng checklist	110
9.1.4.5. Cập nhật các checklist	110
9.2. Đào tạo đội ngũ và cấp chứng chỉ.....	111
9.2.1. Mục tiêu của đào tạo và cấp chứng chỉ	111
9.2.2. Tiến trình đào tạo và cấp chứng chỉ	112

9.2.3. Xác định yêu cầu kiến thức chuyên môn và sự cần thiết của đào tạo và cập nhật	113
9.2.4. Xác định những nhu cầu đào tạo và cập nhật (updating)	113
9.2.5. Lên kế hoạch đào tạo và chương trình cập nhật	114
9.2.6. Định nghĩa các vị trí yêu cầu cấp chứng chỉ	115
9.2.7. Lên kế hoạch các tiến trình cấp chứng chỉ	115
9.2.8. Phân phối các chương trình đào tạo và cấp chứng chỉ	116
9.2.9. Những công việc tiếp theo của việc đào tạo và cấp chứng chỉ	117
9.3. Các hành động sửa lỗi và phòng ngừa	117
9.3.1. Định nghĩa hoạt động sửa lỗi và phòng ngừa	118
9.3.2. Tiến trình hành động sửa lỗi và phòng ngừa	118
9.3.3. Thu thập, phân tích thông tin	118
9.3.4. Phát triển các giải pháp và thực thi	119
9.3.5. Tổ chức các hành động phòng ngừa và sửa lỗi	120
BÀI 10. BÀI TẬP VỀ CHECKLIST	122
BÀI 11. QUẢN LÝ CẤU HÌNH	125
11.1. Quản lý cấu hình	125
11.1.1. Các thành phần cấu hình phần mềm	125
11.1.2. Quản lý cấu hình phần mềm	125
11.1.3. Kiểm soát sự thay đổi phần mềm	126
11.2. Kiểm soát tài liệu	129
11.2.1. Các tài liệu kiểm soát và bản ghi chất lượng	129
11.2.2. Danh sách các tài liệu được kiểm soát	133
11.2.3. Chuẩn bị, phê chuẩn, lưu trữ và thu hồi tài liệu kiểm soát	133
BÀI 12. THẢO LUẬN VỀ QUẢN LÝ CẤU HÌNH PHẦN MỀM	135
Bài 13. CÁC THÀNH PHẦN QUẢN LÝ CHẤT LƯỢNG PHẦN MỀM	137
13.1. Điều khiển tiến độ dự án	137
13.1.1 Các thành phần điều khiển tiến độ dự án	137
13.1.2 Điều khiển tiến độ của các dự án nội bộ và các thành phần bên ngoài	137
13.1.3. Thực thi kiểm soát tiến độ dự án	137
13.1.4. Các công cụ kiểm soát tiến độ phần mềm	138
13.2. Độ đo chất lượng phần mềm	139
13.2.1. Các mục tiêu đo lường phần mềm và phân loại các độ đo	139
13.2.2. Các độ đo tiến trình	141
13.2.3. Các độ đo sản phẩm	144
13.2.4. Thực hiện đo chất lượng phần mềm	146
13.2.5. Những giới hạn của các độ đo phần mềm	147
13.3. Giá thành của chất lượng phần mềm	148
13.3.1. Các mục tiêu tính giá thành các độ đo chất lượng phần mềm	148
13.3.2. Mô hình truyền thông tính giá chất lượng phần mềm	148
BÀI 14. BÀI TẬP VỀ ĐỘ ĐO CHẤT LƯỢNG PHẦN MỀM	152
Bài 15. CHUẨN QUẢN LÝ CHẤT LƯỢNG ISO, CMM và CMMI	154
15.1. Chuẩn quản lý chất lượng	154
15.1.1. Phạm vi của các chuẩn quản lý chất lượng	154
15.2. ISO 9001 và ISO 9000-3	155

15.3. Các mô hình tăng trưởng khả năng – phương pháp đánh giá CMM và CMMI	157
BÀI 16. CÁC CHUẨN TIẾN TRÌNH DỰ ÁN SQA	159
16.1. IEEE/EIA Std 12207- các tiến trình vòng đời phần mềm	159
16.2. IEEE Std 1012 – xác minh và thẩm định.....	160
16.3. IEEE Std 1028 – rà soát.....	161
BÀI 17. TỔ CHỨC ĐỂ ĐẢM BẢO CHẤT LƯỢNG	164
17.1. Giới thiệu	164
17.1.1. Cơ cấu tổ chức phát triển phần mềm.....	164
17.1.2. Khung tổ chức phát triển phần mềm	164
17.2. Quản lý và vai trò của quản lý trong đảm bảo chất lượng phần mềm.....	165
17.2.1. Các hoạt động đảm bảo chất lượng của quản lý mức cao nhất.....	165
17.2.2. Những trách nhiệm quản lý phòng ban	168
17.2.3. Những trách nhiệm quản lý dự án	169
17.3. Đơn vị SQA và các tác nhân khác trong hệ thống SQA.....	170
17.3.1. Đơn vị SQA.....	170
17.3.2. Những ủy viên SQA và nhiệm vụ	171
17.3.3. Hội đồng SQA và nhiệm vụ	172
17.3.4. Nhiệm vụ và phương thức hoạt động của diễn đàn SQA	172
BÀI 18. THỰC HÀNH KHẢO SÁT SQA PLAN.....	174
BÀI 19. THỰC HÀNH QUẢN LÝ CẤU HÌNH PHẦN MỀM (1).....	178
BÀI 20. THỰC HÀNH QUẢN LÝ CẤU HÌNH PHẦN MỀM (2).....	187
BÀI 21. THỰC HÀNH QUẢN LÝ LỖI (1)	193
21.1. Mục tiêu.....	193
21.2. Nội dung	193
21.4 Chuẩn đầu ra	193
BÀI 22. THỰC HÀNH QUẢN LÝ LỖI (2)	200
BÀI 23. THỰC HÀNH ÁP DỤNG CHUẨN	200
BÀI 24 THỰC HÀNH ÁP DỤNG ĐỘ ĐO CHẤT LƯỢNG	209
BÀI 25. KIỂM TRA THỰC HÀNH	209

BÀI 1. KHÁI NIỆM VỀ CHẤT LƯỢNG PHẦN MỀM

1.1.Đặc điểm của phần mềm và môi trường phát triển phần mềm

Có thể nói phần mềm là một sản phẩm đặc biệt, nó không giống như các sản phẩm công nghiệp khác nên người ta thường gọi là phát triển phần mềm. Để phân biệt sự khác nhau giữa sản phẩm phần mềm với các sản phẩm khác ta sẽ xem xét ba đặc điểm sau :

(1) *Độ phức tạp của sản phẩm*: Độ phức tạp của sản phẩm có thể được đo bằng số lượng phương thức vận hành của sản phẩm. Một sản phẩm công nghiệp thậm chí là một máy tiên tiến cũng không cho phép nhiều hơn vài trăm phương thức vận hành. Trong khi đó, một gói phần mềm có thể có tới hàng triệu khả năng vận hành. Do đó, vấn đề đảm bảo vô số khả năng vận hành được xác định và phát triển đúng là một thách thức chính của công nghiệp phần mềm.

(2) *Tính trực quan của sản phẩm* : Trong khi các sản phẩm công nghiệp có thể nhìn thấy được, thì các sản phẩm phần mềm đều vô hình. Hầu hết các nhược điểm của một sản phẩm công nghiệp đều có thể phát hiện trong tiến trình sản xuất. Hơn nữa, rất dễ dàng nhận thấy được sự khuyết thiếu một phần nào đó trong một sản phẩm công nghiệp (ví dụ : một cái ô tô không có cửa sổ). Trái lại, các nhược điểm trong các sản phẩm phần mềm (được lưu trữ trong các đĩa mềm hay CD) đều không nhìn thấy được, vì vậy, thực tế là các phần của một gói phần mềm có thể thiếu ngay từ đầu.

(3) *Tiến trình sản xuất và phát triển phần mềm* : Các pha trong tiến trình sản xuất một sản phẩm

- Phát triển sản phẩm : trong sản xuất công nghiệp, người thiết kế và các nhân viên đảm bảo chất lượng kiểm tra nguyên mẫu để phát hiện các khuyết điểm của chúng. Trong sản xuất phần mềm, các chuyên gia đảm bảo chất lượng và đội phát triển có xu hướng tìm ra các lỗi sản

phẩm vốn có. Kết quả cuối cùng của pha này là một nguyên mẫu đã được phê chuẩn, sẵn sàng để sản xuất.

- Lập kế hoạch sản xuất sản phẩm : tại pha này, trong các ngành công nghiệp, tiến trình sản xuất và các công cụ được thiết kế và chuẩn bị. Một số dòng sản phẩm đặc

biệt cần phải được thiết kế và xây dựng. Do đó, pha này đã tạo thêm cơ hội xem xét sản phẩm, và có thể phát hiện

ra các khuyết điểm đã bị người rà soát và kiểm thử bỏ qua trong pha phát triển. Ngược lại, đây là pha không yêu cầu trong tiến trình sản xuất phần mềm, bởi việc sản xuất các bản copy phần mềm và in các sách hướng dẫn phần mềm được thực hiện tự động. Điều này được áp dụng cho bất kỳ sản phẩm phần mềm nào, từ nhỏ tới lớn.

- Sản xuất : Trong pha này, các thủ tục đảm bảo chất lượng trong sản xuất công nghiệp được áp dụng để phát hiện lỗi sản xuất. Các khuyết điểm trong sản phẩm được phát hiện ra ở giai đoạn đầu tiên của quá trình sản xuất có thể được hiệu chỉnh bằng một thay đổi trong thiết kế sản phẩm hoặc nguyên liệu, hay trong các công cụ sản xuất...Nhờ đó có thể tránh được các khuyết điểm này trong các sản phẩm được sản xuất trong tương lai. Ngược lại, như đã nói ở phần trước, việc sản xuất phần mềm đơn giản chỉ là sao chép các sản phẩm và in các sách hướng dẫn, do đó việc phát hiện các khuyết điểm của sản phẩm rất khó khăn.

Kỹ nghệ phần mềm đã có những bước phát triển đáng kể và vượt qua nhiều giai đoạn khủng hoảng. Những kết quả nghiên cứu về kỹ nghệ phần mềm đã giúp các tổ chức phát triển phần mềm một cách chuyên nghiệp hơn. Môi trường phát triển phần mềm cũng mang những nét đặc trưng riêng. Với bảy đặc trưng sau ta có thể hiểu rõ hơn về môi trường phát triển cũng như môi trường bảo trì phần mềm chuyên nghiệp:

(1) Các điều kiện hợp đồng : Là kết quả của các cam kết và điều kiện trong bản hợp đồng giữa nhà phát triển phần mềm và khách hàng, các hoạt động bảo trì và phát triển phần mềm cần đương đầu với các vấn đề :

- Một danh sách các yêu cầu chức năng được xác định mà phần mềm được phát triển và công việc bảo trì nó phải thực hiện.
- Ngân sách dự án.
- Thời gian biểu dự án.

Nhà quản lý việc phát triển phần mềm và bảo trì dự án cần nỗ lực lớn trong việc giám sát các hoạt động để đạt được các yêu cầu của hợp đồng.

(2) Mỗi quan hệ khách hàng – nhà cung cấp : Trong suốt quá trình phát triển và bảo trì phần mềm, các hoạt động đều nằm dưới sự giám sát của khách hàng. Đội dự án phải hợp tác liên tục với khách hàng : để xem xét các yêu cầu thay đổi, để thảo luận những gì khách hàng không bằng lòng về các khía cạnh khác nhau của dự án, và để đạt được sự chấp thuận cho các thay đổi theo sáng kiến của đội phát triển.

(3) Yêu cầu làm việc theo nhóm : 3 nhân tố thường thúc đẩy việc thành lập một đội dự án thay vì giao dự án cho một chuyên gia :

- Các yêu cầu về thời gian biểu. Nói cách khác, khối lượng công việc được thực hiện trong suốt thời kỳ dự án đòi hỏi sự tham gia của nhiều người nếu muốn dự án hoàn thành đúng thời hạn.
- Để thực hiện được dự án cần có nhiều chuyên ngành khác nhau.
- Sự rà soát lại và hỗ trợ lẫn nhau của các chuyên gia sẽ làm tăng chất lượng dự án.

(4) Hợp tác và phối hợp với các đội phần mềm khác : Để thực hiện được các dự án, đặc biệt là các dự án có quy mô lớn, cần nhiều hơn một đội dự án. Đây là điều rất phổ biến trong công nghiệp phần mềm. Trong các trường hợp như thế, có thể đòi hỏi phải hợp tác với :

- Các đội phát triển phần mềm khác trong cùng một tổ chức.
- Các đội phát triển phần cứng trong cùng một tổ chức.
- Các đội phát triển phần cứng và phần mềm của các nhà cung cấp khác.
- Các đội phát triển phần cứng và phần mềm của khách hàng – những người tham gia một phần vào sự phát triển dự án.

(5) Các giao diện với các hệ thống phần mềm khác : Ngày nay, hầu hết hệ thống phần mềm đều có các giao diện với các gói phần mềm khác nhau. Các giao diện này cho phép các dữ liệu dưới dạng điện tử được “chảy” giữa các hệ thống phần mềm. Có thể định nghĩa các loại giao diện chính sau đây :

- Các giao diện đầu vào – nơi các hệ thống phần mềm khác truyền dữ liệu tới hệ thống phần mềm của bạn.
- Các giao diện đầu ra – nơi hệ thống phần mềm của bạn truyền dữ liệu đã được xử lý tới các hệ thống phần mềm khác.

- Các giao diện đầu vào và đầu ra tới các bảng điều khiển của máy, như trong các hệ thống kiểm soát thí nghiệm và các hệ thống y tế, thiết bị chế biến kim loại...

(6) Sự cần thiết phải tiếp tục thực hiện một dự án mặc dù thành viên đội có sự thay đổi : Việc các thành viên trong đội rời khỏi đội trong thời gian phát triển dự án là khá phổ biến, do việc thăng chức với các công việc cấp cao hơn, chuyển sang một thành phố khác...Người lãnh đạo đội phải thay thế các thành viên trong đội bởi các nhân viên khác hoặc bởi một nhân viên mới được tuyển dụng. Không kể đến bao nhiêu nỗ lực cần đầu tư vào việc đào tạo một thành viên mới, việc thay đổi thành viên sẽ kéo theo thời gian thực hiện dự án sẽ thay đổi.

(7) Sự cần thiết phải tiếp tục thực hiện việc bảo trì phần mềm trong một thời gian dài: Các khách hàng mua hoặc phát triển một hệ thống phần mềm mong đợi sẽ tiếp tục sử dụng nó trong một thời gian dài, thường là từ 5-10 năm. Trong suốt thời kỳ dịch vụ, cuối cùng cũng cần tới sự bảo trì. Trong hầu hết trường hợp, dịch vụ bảo trì cần được cung cấp trực tiếp bởi nhà phát triển. Trong trường hợp các phần mềm được phát triển “trong nhà”, các khách hàng “nội bộ” sẽ cùng chia sẻ vấn đề bảo trì phần mềm trong suốt thời kỳ dịch vụ của hệ thống phần mềm.

1.2.Khái niệm phần mềm

Phần mềm bao gồm những thành phần sau đây:

- Chương trình máy tính
- Các thủ tục
- Tài liệu liên quan
- Dữ liệu cần thiết cho sự vận hành của hệ thống

Mỗi thành phần phần mềm đều có chức năng riêng và chất lượng của chúng đóng góp vào chất lượng chung của phần mềm và bảo trì phần mềm như sau:

1. Chương trình máy tính được cần thiết là hiển nhiên vì chúng giúp máy tính vận hành thực thi các yêu cầu ứng dụng.
2. Những thủ tục được yêu cầu để định nghĩa theo một thứ tự và lịch biểu của một chương trình khi thực thi, phương thức được triển khai và người chịu trách nhiệm cho thực thi các hoạt động cần thiết cho việc tác động vào phần mềm

3. Nhiều kiểu tài liệu là cần thiết cho người phát triển, người sử dụng và người có nhiệm vụ duy trì. Tài liệu phát triển (báo cáo yêu cầu, báo cáo thiết kế, mô tả chương trình, v.v) cho phép sự phối hợp và cộng tác hiệu quả giữa các thành viên trong đội ngũ phát triển và hiệu quả trong việc xem lại và rà soát cả sản phẩm lập trình và thiết kế. Tài liệu sử dụng(thường là hướng dẫn sử dụng) cung cấp một sự miêu tả cho ứng dụng sẵn sàng và những phương pháp thích hợp cho họ sử dụng. Tài liệu bảo trì (tài liệu cho người phát triển) cung cấp cho đội bảo trì tất cả những thông tin yêu cầu về mã nguồn và công việc và cấu trúc cho từng module. Thông tin này được sử dụng để tìm nguyên nhân lỗi (bugs) hoặc thay đổi hoặc bổ sung thêm vào phần mềm có sẵn.

4. Dữ liệu bao gồm các tham số đầu vào, mã nguồn và danh sách tên thích hợp với phần mềm để đặc tả những cái cần thiết cho người sử dụng thao tác với hệ thống. Một kiểu khác của dữ liệu cần thiết là chuẩn dữ liệu test, sử dụng để sách định rõ những thứ thay đổi không mong muốn trong mã nguồn hoặc dữ liệu phần mềm đã từng xảy ra và những loại sự cố phần mềm nào có thể được lường trước.

1.3.Lỗi phần mềm và phân loại nguyên nhân gây ra lỗi phần mềm

Có nhiều nguyên nhân gây ra lỗi phần mềm, biểu hiện của các lỗi cũng khác nhau ở mỗi giai đoạn phát triển phần mềm. Có ba loại lỗi phần mềm chính :

- Error: Là các phần của code mà không đúng một phần hoặc toàn bộ như là kết quả của lỗi ngữ pháp, logic hoặc lỗi khác được sinh ra bởi các nhà phân tích hệ thống, một lập trình viên hoặc các thành viên khác của đội phát triển phần mềm.
- Fault: Là các errors mà nó gây ra hoạt động không chính xác của phần mềm trong một ứng dụng cụ thể.
- Failures: Các faults trở thành failures chỉ khi chúng được “activated” đó là khi người dùng cố gắng áp dụng các phần mềm cụ thể đó bị faulty. Do đó, nguồn gốc của bất kì failure nào là một errors.

1.3.2. Nguyên nhân gây ra lỗi phần mềm

Việc phát hiện ra lỗi là cần thiết, nhưng tìm ra nguyên nhân gây lỗi để tránh lỗi trong tương lai mới thực sự quan trọng. Chín nguyên nhân gây ra lỗi phần mềm thống kê sau đây đã được tổng kết sau nhiều năm nghiên cứu :

1. Định nghĩa yêu cầu lỗi
2. Lỗi giao tiếp giữa khách hàng và người phát triển
3. Sự thiếu rõ ràng của các yêu cầu phần mềm
4. Lỗi thiết kế logic
5. Lỗi coding
6. Không phù hợp với tài liệu và chỉ thị coding
7. Thiếu sót trong quá trình kiểm thử
8. Lỗi thủ tục

Nội dung cụ thể mỗi nguyên nhân được xác định như sau:

- Định nghĩa các yêu cầu bị lỗi

Việc xác định các lỗi yêu cầu, thường do khách hàng, là một trong những nguyên nhân chính của các lỗi phần mềm. Các lỗi phổ biến nhất loại này là:

- Sai sót trong định nghĩa các yêu cầu.
- Không có các yêu cầu quan trọng.
- Không hoàn chỉnh định nghĩa các yêu cầu.
- Bao gồm các yêu cầu không cần thiết, các chức năng mà không thực sự cần thiết trong tương lai gần.

- Các lỗi trong giao tiếp giữa khách hàng và nhà phát triển

Hiểu lầm trong giao tiếp giữa khách hàng và nhà phát triển là nguyên nhân bổ sung cho các lỗi ưu tiên áp dụng trong giai đoạn đầu của quá trình phát triển:

- Hiểu sai các chỉ dẫn của khách hàng như đã nêu trong các tài liệu yêu cầu.
- Hiểu sai các yêu cầu thay đổi của khách hàng được trình bày với nhà phát triển bằng văn bản trong giai đoạn phát triển.
- Hiểu sai của các yêu cầu thay đổi của khách hàng được trình bày bằng lời nói với nhà phát triển trong giai đoạn phát triển.

■ Hiểu sai về phản ứng của khách hàng đối với các vấn đề thiết kế trình bày của nhà phát triển.

Thiếu quan tâm đến các đề nghị của khách hàng đề cập đến yêu cầu thay đổi và khách hàng trả lời cho các câu hỏi nêu ra bởi nhà phát triển trên một phần của nhà phát triển.

- Sai lệch có chủ ý từ các yêu cầu phần mềm

Trong một số trường hợp, các nhà phát triển có thể cố tình đi chệch khỏi các yêu cầu rong tài liệu, hành động thường gây ra lỗi phần mềm. Các lỗi trong những trường hợp này là sản phẩm phụ của các thay đổi. Các tình huống thường gặp nhất là:

■ Phát triển các module phần mềm Các thành phần sử dụng lại lấy từ một dự án trước đó mà không cần phân tích đầy đủ về những thay đổi và thích nghi cần thiết để thực hiện một cách chính xác tất cả các yêu cầu mới.

■ Do thời gian hay áp lực ngân sách, nhà phát triển quyết định bỏ qua một phần của các yêu cầu các chức năng trong một nỗ lực để đối phó với những áp lực này.

■ Nhà phát triển-khởi xướng, không được chấp thuận các cải tiến cho phần mềm, mà không có sự chấp thuận của khách hàng, thường xuyên bỏ qua các yêu cầu có vẻ nhỏ đối với nhà phát triển. Như vậy những thay đổi "nhỏ" có thể, cuối cùng, gây ra lỗi phần mềm.

- Các lỗi coding

Một loạt các lý do các lập trình viên có thể gây ra các lỗi code. Những lý do này bao gồm sự hiểu lầm các tài liệu thiết kế, ngôn ngữ sai sót trong ngôn ngữ lập trình, sai sót trong việc áp dụng các CASE và các công cụ phát triển khác, sai sót trong lựa chọn dữ liệu...

- Không tuân thủ theo các tài liệu hướng dẫn và mã hóa

Hầu hết các đơn vị phát triển có tài liệu hướng dẫn và tiêu chuẩn mã hóa riêng của mình để xác định nội dung, trình tự và định dạng của văn bản, và code tạo ra bởi các thành viên. Để hỗ trợ yêu cầu này, đơn vị phát triển và công khai các mẫu và hướng dẫn mã hóa. Các thành viên của nhóm phát triển, đơn vị được yêu cầu phải thực hiện theo các yêu cầu này.

- Thiếu sót trong quá trình thử nghiệm

Thiếu sót trong quá trình thử nghiệm ảnh hưởng đến tỷ lệ lỗi bằng cách để lại một số lỗi lớn hơn không bị phát hiện hoặc không phát hiện đúng. Những kết quả yếu kém từ các nguyên nhân sau đây:

- Kế hoạch thử nghiệm chưa hoàn chỉnh để lại phần không được điều chỉnh của phần mềm hoặc các chức năng ứng dụng và các trạng thái của hệ thống. Failures trong tài liệu và báo cáo phát hiện sai sót và lỗi lầm.
- Nếu không kịp thời phát hiện và sửa chữa lỗi phần mềm theo như của chỉ dẫn không phù hợp trong những lý do cho lỗi này.
- Không hoàn chỉnh sửa chữa các lỗi được phát hiện do sơ suất hay thời gian áp lực.

- Các lỗi thủ tục

Các thủ tục trực tiếp cho người sử dụng đối với các hoạt động là cần thiết ở mỗi bước của quá trình. Chúng có tầm quan trọng đặc biệt trong các hệ thống phần mềm phức tạp, nơi các tiến trình được tiến hành một vài bước, mỗi bước trong số đó có thể có nhiều kiểu dữ liệu và cho phép kiểm tra các kết quả trung gian.

- Các lỗi về tài liệu

Các lỗi về tài liệu là vấn đề của các đội phát triển và bảo trì đều có sai sót trong tài liệu thiết kế và trong tài liệu hướng dẫn tích hợp trong thân của phần mềm. Những lỗi này có thể là nguyên nhân gây ra lỗi bổ sung trong giai đoạn phát triển tiếp và trong thời gian bảo trì.

Cần nhấn mạnh rằng tất cả các nguyên nhân gây ra lỗi đều là con người, công việc của các nhà phân tích hệ thống, lập trình, kiểm thử phần mềm, các chuyên gia tài liệu, và thậm chí cả các khách hàng và đại diện của họ.

BÀI 2. CÁC YẾU TỐ CHẤT LƯỢNG PHẦN MỀM

2.1. Định nghĩa chất lượng phần mềm và đảm bảo chất lượng phần mềm

Theo IEEE, chất lượng phần mềm được định nghĩa như sau : Chất lượng phần mềm là:

- Mức độ mà một hệ thống, thành phần hoặc một tiến trình đạt được yêu cầu đã đặc tả
- Mức độ mà một hệ thống, thành phần hoặc một tiến trình đạt được những nhu cầu hay mong đợi của khách hàng hoặc người sử dụng.

Ban đầu đảm bảo chất lượng phần mềm có mục tiêu đạt được các yêu cầu đề ra, tuy nhiên thực tế phát triển phần mềm tồn tại rất nhiều ràng buộc đòi hỏi người phát triển cần tối ưu hóa công tác quản lý.

Theo Daniel Galin, khái niệm đảm bảo chất lượng phần mềm được xác định như sau :

Đảm bảo chất lượng phần mềm là một tập các hoạt động đã được lập kế hoạch và có hệ thống, cần thiết để cung cấp đầy đủ sự tin cậy vào quy trình phát triển phần mềm hay quy trình bảo trì phần mềm của sản phẩm hệ thống phần mềm phù hợp với các yêu cầu chức năng kỹ thuật cũng như với các yêu cầu quản lý mà giữ cho lịch biểu và hoạt động trong phạm vi ngân sách.

2.2. Những mục tiêu đảm bảo chất lượng phần mềm

Phát triển phần mềm luôn đi đôi với bảo trì, vì vậy các hoạt động bảo đảm chất lượng phần mềm đều có mối liên quan chặt chẽ đến bảo trì. Những mục tiêu đảm bảo chất lượng phần mềm tương ứng với giai đoạn phát triển và bảo trì được xác định cụ thể như sau :

- Phát triển phần mềm (hướng tiến trình)

1. Đảm bảo một mức độ chấp nhận được rằng phần mềm sẽ thực hiện được các yêu cầu chức năng.
2. Đảm bảo một mức độ chấp nhận được rằng phần mềm sẽ đáp ứng được các yêu cầu về lịch biểu và ngân sách
3. Thiết lập và quản lý các hoạt động để cải thiện và nâng cao hiệu quả của phát triển phần mềm và các hoạt động SQA.

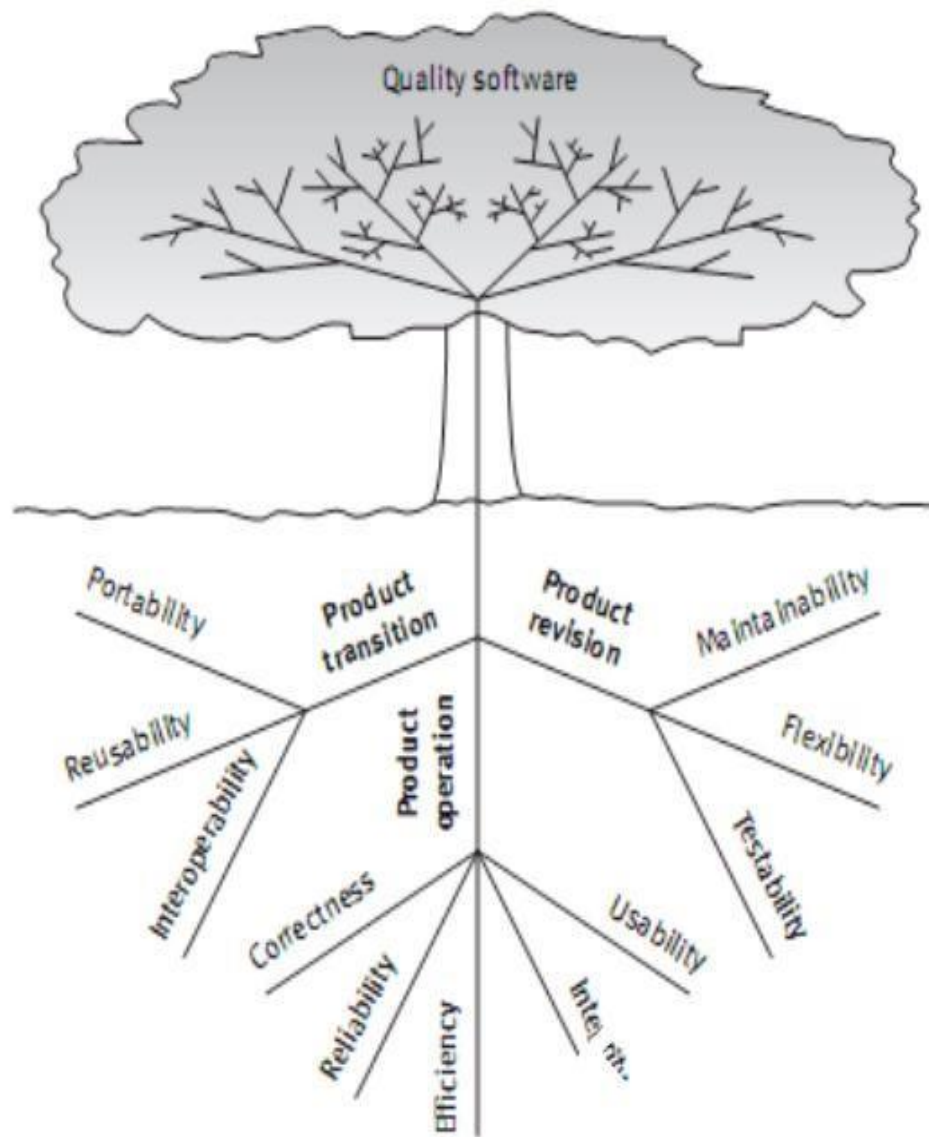
- Bảo trì phần mềm (hướng sản phẩm)

1. Đảm bảo một mức độ chấp nhận được rằng các hoạt động bảo trì phần mềm sẽ đáp ứng được các yêu cầu chức năng.
2. Đảm bảo một mức độ chấp nhận được rằng các hoạt động bảo trì phần mềm sẽ đáp ứng được các yêu cầu về lịch biểu và ngân sách
3. Thiết lập và quản lý các hoạt động để cải thiện và nâng cao hiệu quả của bảo trì phần mềm.

2.3. Phân loại yêu cầu phần mềm ứng với các yếu tố chất lượng phần mềm

Đã có nhiều tác giả nghiên cứu về các yếu tố chất lượng phần mềm từ các yêu cầu của nó. Theo thời gian có thể quan niệm về việc đảm bảo chất lượng phần mềm có phần thay đổi, tuy nhiên mô hình các yếu tố đảm bảo chất lượng phần mềm của McCall ra đời vào những năm 70 của thế kỷ trước vẫn còn được nhiều người nhắc đến như là cơ sở tham chiếu các yêu cầu phần mềm. Sau McCall cũng có một số mô hình được quan tâm như mô hình do Evans, Marciniak hay mô hình của Deutsch và Willis, tuy nhiên những mô hình này chỉ bổ sung hay sửa đổi một vài yếu tố chất lượng. Theo McCall, các yếu tố chất lượng phần mềm được chia làm ba loại :

- Các yếu tố hoạt động của sản phẩm bao gồm tính chính xác, tin cậy, hiệu quả, tính toàn vẹn, sử dụng được
- Các yếu tố rà soát bao gồm tính bảo trì, linh hoạt, có thể test được
- Các yếu tố chuyển giao bao gồm tính khả chuyển, có khả năng sử dụng lại, có khả năng giao tác.



Hình 1.1 Cây mô hình yếu tố chất lượng phần mềm theo McCall

Chi tiết các thuộc tính được phân tích như sau :

(1) Các yếu tố vận hành sản phẩm : Sự chính xác, độ tin cậy, tính hiệu quả, tính toàn vẹn và khả năng sử dụng được :

- Sự chính xác : Các yêu cầu về độ chính xác được xác định trong một danh sách các đầu ra cần thiết của hệ thống phần mềm, như màn hình hiển thị truy vấn số dư của khách hàng trong một hệ thống thông tin kế toán bán hàng... Các đặc tả đầu ra thường là đa chiều, một số chiều thông dụng là :

- o Nhiệm vụ đầu ra (ví dụ : bản in hóa đơn bán hàng, hay đèn báo động đỏ khi nhiệt độ tăng lên trên 250 độ F)

- o Độ chính xác yêu cầu của các đầu ra này; chúng có thể bị ảnh hưởng bất lợi bởi các tính toán không chính xác hay các dữ liệu không chính xác.

- o Tính đầy đủ của thông tin đầu ra; chúng có thể bị ảnh hưởng bất lợi bởi dữ liệu không đầy đủ.

- o Up-to-dateness của thông tin (xác định bằng thời gian giữa sự kiện và việc xem xét hệ thống phần mềm.

- o Độ sẵn sàng của thông tin (thời gian đáp ứng : được định nghĩa là thời gian cần thiết để có được các thông tin yêu cầu)

- o Các chuẩn cho việc code và viết tài liệu cho hệ thống phần mềm.

- Độ tin cậy : Các yêu cầu về độ tin cậy giải quyết các lỗi để cung cấp dịch vụ. Chúng xác định tỷ lệ lỗi hệ thống phần mềm tối đa cho phép, các lỗi này có thể là lỗi toàn bộ hệ thống hoặc một hay nhiều chức năng riêng biệt của nó.

- Tính hiệu quả : Các yêu cầu về tính hiệu quả giải quyết vấn đề về các tài nguyên phần cứng cần thiết để thực hiện tất cả các chức năng của hệ thống phần mềm với sự phù hợp của tất cả các yêu cầu khác. Các tài nguyên phần cứng chính được xem xét ở đây là khả năng xử lý của máy tính (được đo bằng MIPS – triệu lệnh/giây; MHz – triệu chu kỳ/giây...); khả năng lưu trữ dữ liệu (dung lượng bộ nhớ, dung lượng đĩa – được đo bằng MBs, GBs, TBs...) và khả năng truyền dữ liệu (thường được đo bằng MBPS, GBPS). Các yêu cầu này có thể bao gồm cả

các giá trị tối đa tài nguyên phần cứng được sử dụng trong hệ thống phần mềm.

Một yêu cầu khác về tính hiệu quả đó là thời gian giữa các lần phải sạc điện đối với các hệ thống nằm trên các máy tính xách tay hay các thiết bị di động.

- Các yêu cầu về tính toàn vẹn giải quyết các vấn đề về bảo mật hệ thống phần mềm, các yêu cầu này để ngăn chặn sự truy cập trái phép, để phân biệt giữa phần lớn nhân viên chỉ được phép xem thông tin với một nhóm hạn chế những người được phép thêm và thay đổi dữ liệu...

- Các yêu cầu về khả năng sử dụng được sẽ đưa ra phạm vi của tài nguyên nhân lực cần thiết để đào tạo một nhân viên mới và để vận hành hệ thống phần mềm.

(2) Các yếu tố về rà soát sản phẩm : bảo trì được, linh động và kiểm tra được :

- Khả năng bảo trì được : Các yêu cầu về khả năng bảo trì được sẽ xác định người dùng và nhân viên bảo trì phải nỗ lực thế nào để xác định được nguyên nhân của các lỗi phần mềm, để sửa lỗi và để xác nhận việc sửa lỗi thành công. Các yêu cầu của yếu

tổ này nói tới cấu trúc modul của phần mềm, tài liệu chương trình nội bộ và hướng dẫn sử dụng của lập trình viên...

- Tính linh động : Các yêu cầu về tính linh động cũng bao gồm cả các khả năng và nỗ lực cần thiết để hỗ trợ các hoạt động bảo trì. Chúng gồm các nguồn lực (man-day) cần thiết để thích nghi với một gói phần mềm, với các khách hàng trong cùng nghề, với các mức độ hoạt động khác nhau, với các loại sản phẩm khác nhau... Các yêu cầu về yếu tố này cũng hỗ trợ các hoạt động bảo trì trở nên hoàn hảo, như thay đổi và bổ sung vào phần mềm để tăng dịch vụ của nó và để thích nghi với các thay đổi trong môi trường thương mại và kỹ thuật của công ty.

- Khả năng test được : Các yêu cầu về khả năng kiểm tra được nói tới việc kiểm tra sự vận hành có tốt hay không của các hệ thống thông tin. Các yêu cầu về khả năng kiểm tra được liên quan tới các tính năng đặc biệt trong chương trình giúp người tester dễ dàng thực hiện công việc của mình hơn, ví dụ như đưa ra các kết quả trung gian. Các yêu cầu về khả năng kiểm tra được liên quan tới vận hành phần mềm bao gồm các chuẩn đoán tự động được thực hiện bởi hệ thống phần mềm trước khi bắt đầu hệ thống, để tìm hiểu xem có phải tất cả các thành phần của hệ thống phần mềm đều làm việc tốt hay không, và để có một bản báo cáo về các lỗi đã được phát hiện. Một loại khác của yêu cầu này là việc check các dự đoán tự động, được các kỹ thuật viên bảo trì sử dụng để phát hiện nguyên nhân gây lỗi phần mềm.

(3) Các yếu tố về chuyên giao sản phẩm : tính lưu động (khả năng thích nghi với môi trường), khả năng tái sử dụng và khả năng cộng tác được :

- Tính lưu động : các yêu cầu về tính lưu động nói tới khả năng thích nghi của hệ thống phần mềm với các môi trường khác, bao gồm phần cứng khác, các hệ điều hành khác... Các yêu cầu này đòi hỏi các phần mềm cơ bản có thể tiếp tục sử dụng độc lập hoặc đồng thời trong các trường hợp đa dạng.

- Khả năng tái sử dụng : Các yêu cầu về khả năng tái sử dụng nói tới việc sử dụng các modul phần mềm trong một dự án mới đang được phát triển mà các modul này ban đầu được thiết kế cho một dự án khác. Các yêu cầu này cũng cho phép các dự án tương lai có thể sử dụng một modul đã có hoặc một nhóm các modul hiện đang

được phát triển. Tái sử dụng phần mềm sẽ tiết kiệm tài nguyên phát triển, rút ngắn thời gian phát triển và tạo ra các moduls chất lượng cao hơn. Chất lượng modul cao hơn là dựa trên giả định rằng hầu hết các lỗi phần mềm đều được phát hiện bởi các hoạt động đảm bảo chất lượng phần mềm thực hiện trên phần mềm ban đầu, bởi những người sử dụng phần mềm ban đầu và trong suốt những lần tái sử dụng trước của nó. Các vấn đề về tái sử dụng phần mềm đã trở thành một phần trong chuẩn công nghiệp phần mềm (IEEE,1999).

- Khả năng cộng tác : Các yêu cầu về khả năng cộng tác tập trung vào việc tạo ra các giao diện với các hệ thống phần mềm khác. Các yêu cầu về khả năng cộng tác có thể xác định tên của phần mềm với giao diện bắt buộc. Chúng cũng có thể xác định cấu trúc đầu ra được chấp nhận như một tiêu chuẩn trong một ngành công nghiệp cụ thể hoặc một lĩnh vực ứng dụng.

BÀI 3. THẢO LUẬN VỀ CHẤT LƯỢNG PHẦN MỀM

1.1. Có 3 điểm khác nhau giữa sản phẩm phần mềm và các sản phẩm công nghiệp khác

- a) Hãy chỉ ra và mô tả sự khác biệt đó
- b) Những khác biệt này ảnh hưởng đến Đảm bảo chất lượng phần mềm (SQA) như thế nào?

Gợi ý:

a) 3 điểm khác biệt là:

- (1) Độ phức tạp của sản phẩm
- (2) Tính trực quan của sản phẩm
- (3) Tiến trình sản xuất và phát triển phần mềm

b) Ảnh hưởng đến SQA:

- (1) Độ phức tạp của sản phẩm làm cho công tác SQA gặp nhiều khó khăn, như các sản phẩm phải hoạt động chính xác cho tất cả các tùy chọn xác định
- (2) Tính trực quan của sản phẩm: Do sản phẩm phần mềm không nhìn thấy trực quan như các sản phẩm khác sẽ làm cho khó phát hiện lỗi.
- (3) Tiến trình sản xuất và phát triển phần mềm: qua các pha phát triển nên các nhà phát triển phần mềm cũng không thể chắc rằng sản phẩm của họ có khiếm khuyết hay không

1.2. Bảy đặc điểm cho sự phát triển phần mềm chuyên nghiệp và môi trường bảo trì

a) Hãy chỉ ra và mô tả 7 đặc điểm đó

b) Mỗi đặc điểm môi trường đều ảnh hưởng đến nỗ lực các yêu cầu để thực hiện các dự án phát triển và bảo trì phần mềm. Liệt kê và giải thích tại sao lại các nỗ lực yêu cầu lại cần thiết?

c) Mỗi đặc điểm môi trường đều ảnh hưởng đến nỗ lực quản lý các yêu cầu để thực hiện các dự án phát triển và bảo trì phần mềm. Liệt kê và giải thích tại sao lại các nỗ lực như vậy lại cần thiết?

Gợi ý:

a) Bảy đặc điểm cho sự phát triển phần mềm chuyên nghiệp và môi trường bảo trì là:

- Điều kiện hợp đồng và cam kết xác định nội dung và thời gian biểu
- Điều kiện của các mối quan hệ khách hàng-nhà cung cấp, như minh họa bởi nhu cầu tham khảo ý kiến khách hàng và bảo đảm thực hiện chính của họ.
- Yêu cầu làm việc nhóm

- Cần hợp tác và phối hợp với các nhóm phát triển phần mềm và phần cứng khác cả trong lẫn ngoài.
- Cần cho giao tiếp với các hệ thống phần mềm khác
- Cần cho tính liên tục trong việc thực hiện một dự án khi thành viên trong nhóm thay đổi.
- Cần cho bảo trì liên tục của hệ thống phần mềm trong nhiều năm

b) Các ảnh hưởng:

- Xác định các điều kiện của hợp đồng: sự cần thiết phải chuẩn bị một danh sách tài liệu chức năng và các yêu cầu khác của dự án
- Điều kiện của các mối quan hệ khách hàng-nhà cung cấp: sự cần thiết phải duy trì địa chỉ liên lạc liên tục với các chuyên gia của khách hàng cho các bài thuyết trình của phát triển sản phẩm, tham vấn với các khách hàng, và khách hàng đảm bảo chính của các sản phẩm phát triển.
- Yêu cầu làm việc nhóm: sự cần thiết cho các *leader* phải chịu trách nhiệm để được hướng dẫn chuyên nghiệp của các thành viên trong nhóm và kiểm tra sản phẩm của mình.
- Cần hợp tác và phối hợp với các nhóm phát triển phần mềm và phần cứng khác cả trong lẫn ngoài: sự cần thiết phải hiểu các nhiệm vụ được thực hiện bởi các đội khác ở mức độ cho phép chuyên môn phù hợp giao tiếp
- Cần cho giao tiếp với các hệ thống phần mềm khác: sự cần thiết để có được sự quen thuộc với các interfacing chuẩn hoặc interfacing thiết kế của đơn vị thiết bị và / hoặc gói phần mềm.

• Bài tập tình huống:

Bài 1: Một hệ thống giáo dục được đưa ra để chuẩn bị cho sinh viên đối phó với thực tế cuộc sống. Hãy kiểm tra các yêu cầu chức năng của dự án phát triển phần mềm hoặc dự án phần mềm cuối và xác định các yêu cầu để chuẩn bị cho cuộc sống thực tế và đưa ra ở trên

Gợi ý:

Chỉ có một phần của môi trường phát triển phần mềm có thể được thực hành trong khuôn khổ của hệ thống giáo dục

Chúng ta hãy xem xét vấn đề này theo 7 đặc điểm ở câu hỏi trên:

- Điều kiện hợp đồng và cam kết xác định nội dung và thời gian biểu: dự án sinh viên mô phỏng các điều kiện hợp đồng đến một mức độ nào. Một dự án sinh viên điển hình bao gồm các định nghĩa của các chức năng cần thiết cũng như tiến độ thời gian cho hoàn thành. Cam kết ngân sách là một cách tự nhiên không áp dụng

- Điều kiện của các mối quan hệ khách hàng-nhà cung cấp: Các giảng viên-học sinh mỗi quan hệ mô phỏng chừng mực nào đó mối quan hệ giữa khách hàng và nhà cung cấp.
- Yêu cầu làm việc nhóm: Dự án được thực hiện bởi các đội sinh viên kết hợp một số khía cạnh của làm việc theo nhóm, nhưng thường được thực hiện mà không cần một trưởng nhóm.
- Cần hợp tác và phối hợp với các nhóm phát triển phần mềm và phần cứng khác cả trong lẫn ngoài: không cần áp dụng

Bài 2: Giao diện hệ thống xử lý tiền lương như sau:

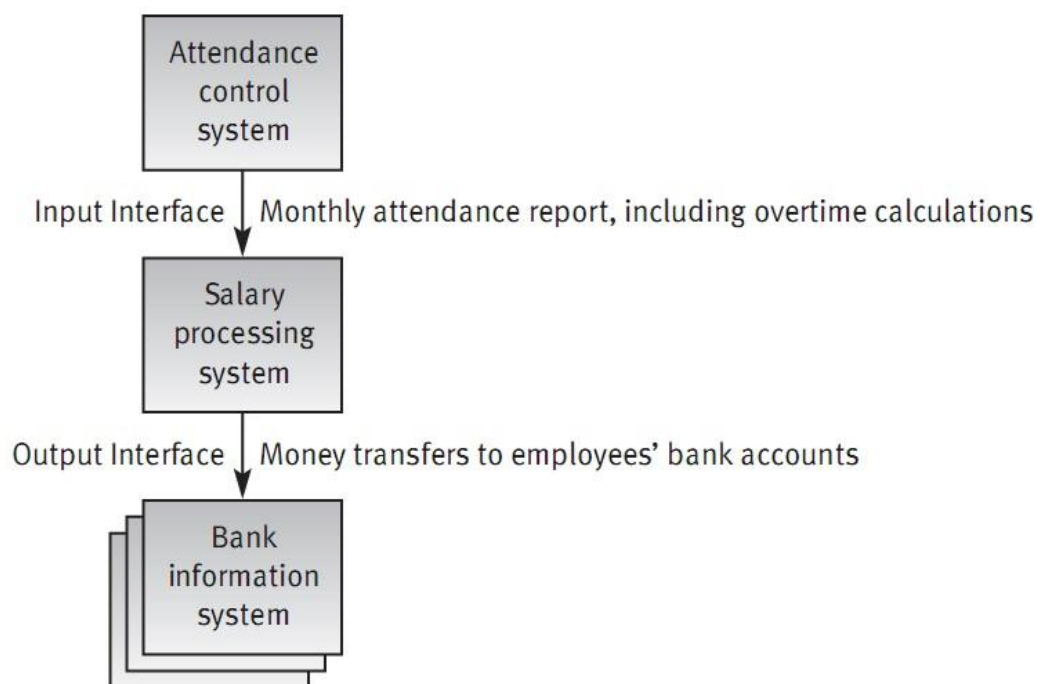


Figure 1.2: The salary software system – an example of software interfaces

- Đưa ra lợi ích chính của việc sử dụng giao diện máy tính thay thế cho việc chuyển các bản in
- Đưa ra 2 ví dụ áp dụng giao diện đầu vào
- Đưa ra 2 ví dụ áp dụng giao diện đầu ra

BÀI 4. RÀ SOÁT HỢP ĐỒNG

4.1. Rà soát hợp đồng

Một hợp đồng tồi chắc chắn là khó có thể chấp nhận được. Từ quan điểm của SQA, một hợp đồng tồi – thường mô tả các yêu cầu không chặt chẽ và đưa ra kế hoạch cũng như ngân sách phi thực tế - thì sẽ dẫn đến một phần mềm có chất lượng tồi. Vì thế, một chương trình SQA cần được thực hiện để đảm bảo chất lượng phần mềm bằng cách rà soát lại những đề xuất ban đầu và sau đó là bản dự thảo hợp đồng (hoạt động “rà soát hợp đồng” bao gồm cả 2 hoạt động trên). Cả hai hoạt động rà soát trên là nhằm mục đích cải thiện ngân sách và thời gian biểu, là những cơ sở cho những đề nghị và hợp đồng sau này, đồng thời có thể biết được những rủi ro tiềm năng sớm (trong mục tiêu ban đầu và trong bản dự thảo hợp đồng).

4.1.1. Tiến trình rà soát hợp đồng và các bước thực hiện

Có khá nhiều tình huống có thể giúp một công ty phần mềm (“nhà cung cấp”) ký hợp đồng với một khách hàng. Phổ biến nhất là:

- Tham gia trong một cuộc đấu thầu

- Đưa ra bản phác thảo dựa trên yêu cầu đề xuất (RFP-Request For Proposal) của khách hàng

- Nhận một đặt hàng từ một khách hàng của công ty

- Nhận một yêu cầu từ bên trong hoặc từ phòng ban khác trong một tổ chức

Rà soát hợp đồng là một thành phần của SQA được nghĩ ra để hướng dẫn xem xét lại những bản dự thảo của những tài liệu đề xuất và hợp đồng. Nếu có thể, rà soát lại hợp đồng còn cung cấp sự giám sát những hợp đồng được thực hiện với những đối tác dự án tiềm năng và các nhà thầu phụ. Tiến trình rà soát có thể được chia thành hai giai đoạn:

- Giai đoạn 1: rà soát lại bản dự thảo đề xuất trước khi giao cho khách hàng tiềm năng (“rà soát bản dự thảo đề xuất”). Giai đoạn này rà soát lại bản dự thảo cuối cùng và những cơ sở đề xuất: những tài liệu yêu cầu của khách hàng, chi tiết yêu cầu thêm của khách hàng và dự diễn giải các yêu cầu, các ước lượng chi phí và tài nguyên, những hợp đồng hiện tại hoặc là những bản dự thảo hợp đồng của nhà cung cấp với các đối tác và nhà thầu phụ.

- Giai đoạn 2: rà soát lại bản dự thảo hợp đồng trước khi ký (“rà soát bản dự thảo hợp đồng”). Giai đoạn này rà soát lại bản dự thảo hợp đồng dựa trên đề xuất và sự hiểu biết (bao gồm cả những thay đổi) đã đạt được trong quá trình thương thảo hợp đồng.

Quá trình rà soát có thể bắt đầu khi những tài liệu dự thảo liên quan đã hoàn thành. Những cá nhân thực hiện rà soát phải xem xét kỹ lưỡng bản dự thảo trong khi đề cập đến một phạm vi toàn diện các đối tượng đang rà soát. Một danh sách kiểm tra là rất hữu ích cho việc đảm bảo xem xét hết các vấn đề liên quan.

Sau khi hoàn thành giai đoạn rà soát, việc cần thiết những sự thay đổi, cái thêm vào và sự hiệu chỉnh phải được thông báo bởi đội đề xuất (sau khi rà soát bản dự thảo đề xuất) và bởi ban phụ trách về luật pháp (sau khi rà soát lại bản dự thảo hợp đồng)

4.1.2. Các mục tiêu rà soát hợp đồng

Những mục đích của công việc rà soát bản dự thảo đề xuất

Mục đích của việc rà soát bản dự thảo đề xuất là để đảm bảo rằng những hoạt động sau được thực hiện một cách thỏa đáng.

1) Những yêu cầu của khách hàng đã được giải thích chi tiết và có chú giải

Những tài liệu yêu cầu đề xuất (RFP) và những tài liệu công nghệ tương tự có thể quá chung chung và mơ hồ cho những mục tiêu của dự án. Kết quả là có nhiều chi tiết cần được thêm vào từ khách hàng. Việc giải thích chi tiết những yêu cầu mập mờ và những cập nhật của chúng nên được ghi lại trong một tài liệu riêng biệt đã được sự chấp nhận của cả khách hàng và công ti phần mềm.

2) Lựa chọn những phương pháp thực hiện dự án đã được kiểm tra.

Thông thường, những lựa chọn có triển vọng và phù hợp mà trên đó thể hiện một đề xuất thì đã được xem xét đầy đủ (nếu tất cả) bởi đội đề xuất. Điều kiện này đặc biệt muốn đề cập đến việc hoàn thành thay thế bao gồm tái sử dụng phần mềm, và những quan hệ đối tác hoặc là thầu lại với những công ti mà có hiểu biết chuyên môn hoặc nhân viên có chuyên môn có thể đảm bảo những điều khoản của đề xuất những khía cạnh hình thức của mối quan hệ giữa khách hàng và công ti phần mềm phải được ghi rõ.

3) Đề xuất nên định nghĩa những thủ tục bao gồm:

Sự giao tiếp với khách hàng và những kênh giao diện

Việc chuyển giao dự án và tiêu chuẩn được chấp nhận

Tiến trình phê chuẩn pha hình thức

Phương thức tiếp theo khách hàng thiết kế và kiểm tra

Thủ tục khách hàng thay đổi yêu cầu

4) Xác định những rủi ro khi phát triển

Những rủi ro khi phát triển, như không đủ kiến thức chuyên môn liên quan đến lĩnh vực nghiệp vụ của dự án hoặc cách sử dụng những công cụ phát triển yêu cầu, cần được xác định và giải quyết.

5) Ước lượng đầy đủ những tài nguyên và thời gian biểu của dự án

Lưu ý

Trong một vài tình huống, một nhà cung cấp cố ý đề nghị cung cấp một chi phí thấp, xem xét các yếu tố như tiềm năng bán hàng. Trong trường hợp này, khi mà đề xuất dựa trên ước lượng thực tế của thời gian, ngân sách và khả năng chuyên môn, những tổn thất phát sinh được coi là một mất mát có thể tính được, không phải là một hợp đồng thất bại

Việc ước lượng tài nguyên đề cập đến đội ngũ nhân viên chuyên nghiệp và ngân sách của dự án, bao gồm cả chi phí cho các nhà thầu con. Việc ước lượng thời gian nên đưa vào những yêu cầu về thời gian của tất cả những bên tham gia vào dự án.

6) Kiểm tra năng lực của công ti đối với dự án.

Việc kiểm tra này nên xem xét đến năng lực chuyên môn cũng như là khả năng sẵn sàng của những thành viên trong đội được yêu cầu và những khả năng phát triển trong thời gian đã được lập lịch.

7) Kiểm tra năng lực của khách hàng để đáp ứng những yêu cầu của mình

Việc kiểm tra này đề cập đến khả năng tài chính và tổ chức của khách hàng, như tuyển dụng và đào tạo nhân sự, cài đặt phần cứng yêu cầu và nâng cấp các thiết bị liên lạc.

8) Định nghĩa đối tác và nhà thầu phụ tham gia

Điều này bao gồm các vấn đề bảo đảm chất lượng, lịch trình thanh toán, phân phối thu nhập, lợi nhuận của dự án, và hợp tác giữa quản lý dự án và các đội

9) Định nghĩa và bảo vệ quyền sở hữu.

Những mục tiêu của việc rà soát bản dự thảo đề xuất

Những mục tiêu của rà soát lại bản dự thảo đề xuất được tổng kết trong bảng sau:

9 mục tiêu của việc rà soát bản dự thảo đề xuất đảm bảo rằng những hành động sau đây được thực hiện một cách thỏa đáng:

1. Những yêu cầu của khách hàng đã được giải thích chi tiết và có chú giải

2 Lựa chọn những phương pháp thực hiện dự án đã được kiểm tra

3. Những khía cạnh hình thức của mối quan hệ giữa khách hàng và công ty phần mềm phải được ghi rõ

4. Xác định những rủi ro khi phát triển

5. Ước lượng đầy đủ những tài nguyên và thời gian biểu của dự án

6. Kiểm tra năng lực của công ty đối với dự án

7. Kiểm tra năng lực của khách hàng để đáp ứng những yêu cầu của mình

8. Định nghĩa đối tác và nhà thầu phụ tham gia

9. Định nghĩa và bảo vệ quyền sở hữu

Yếu tố này có tầm quan trọng trong trường hợp tái sử dụng phần mềm, khi việc có thêm một gói mới vào hoặc có tái sử dụng phần mềm hiện nay trong tương lai hay không cần phải được quyết định. Nó cũng đề cập đến việc sử dụng các file độc quyền của các dữ liệu quan trọng cho hoạt động của hệ thống và các biện pháp an ninh.

Những mục tiêu của rà soát dự thảo hợp đồng.

Những mục tiêu của việc rà soát bản dự thảo hợp đồng để đảm bảo rằng những hoạt động sau đây được thực hiện một cách thỏa đáng:

Không có vấn đề chưa rõ ràng nào vẫn còn lại trong dự thảo hợp đồng

Tất cả những thỏa thuận đạt được giữa các khách hàng và công ty phải được giải thích đầy đủ và chính xác trong hợp đồng và phụ lục của nó. Những hiểu biết này được dùng để giải quyết tất cả các vấn đề chưa rõ ràng và khác biệt giữa khách hàng và công ty mà đã được đưa ra cho đến nay

Không có sự thay đổi, bổ sung, hoặc thiếu sót nào không được thảo luận và sự thỏa thuận nên được đưa vào dự thảo hợp đồng. Việc thay đổi, dù

có ý hay không, có thể dẫn đến sự bỏ sung đáng kể và những nhiệm vụ bất ngờ trong một bộ phận của nhà cung cấp.

Những mục tiêu của việc rà soát lại dự thảo hợp đồng có thể được tổng kết trong bảng

Ba mục tiêu của việc rà soát dự thảo hợp đồng nhằm đảm bảo những hoạt động sau đây được thực hiện một cách thỏa đáng:

Không có vấn đề chưa rõ ràng nào vẫn còn lại trong dự thảo hợp đồng

Mọi thỏa thuận đã đạt được sau khi xem xét những đề xuất phải được chú giải một cách chính xác

Không có sự thay đổi, bổ sung, hoặc thiếu sót đưa vào bản dự thảo hợp đồng

Những mục tiêu của việc rà soát dự thảo hợp đồng

4.1.3. Thực thi rà soát hợp đồng

Duyệt hợp đồng khác nhau về độ lớn của chúng, tùy thuộc vào các đặc tính của dự án đề xuất. Phức tạp này có thể là kỹ thuật hoặc tổ chức. Theo đó, mức độ khác nhau của nguồn lực chuyên môn được điều chỉnh phù hợp cho những duyệt hợp đồng khác nhau. Những nguồn lực chuyên môn đặc biệt cần thiết cho những đề xuất chính.

- Những yếu tố ảnh hưởng tới phạm vi của một bản duyệt hợp đồng

Các yếu tố quan trọng nhất của dự án xác định mức độ của hợp đồng nỗ lực xem xét lại yêu cầu là:

Độ lớn của dự án, thường được đo bằng các nguồn lực man-month.

Kỹ thuật phức tạp của dự án.

Trình độ và sự hiểu biết của nhân viên có kinh nghiệm trong lĩnh vực của dự án. Sự hiểu biết với các lĩnh vực dự án là thường xuyên được liên kết với khả năng tái sử dụng phần mềm; trong trường hợp có thể tái sử dụng phần mềm là cao, mức độ duyệt được giảm xuống

■ Tổ chức dự án phức tạp. Càng với số lượng lớn của các tổ chức (nghĩa là, các đối tác, nhà thầu phụ, và khách hàng) tham gia các dự án, thì càng yêu cầu công sức rà soát hợp đồng lớn hơn.

Do đó chúng tôi có thể cho rằng "đơn giản" việc duyệt hợp đồng sẽ được thực hiện bởi một người xem, họ sẽ tập trung vào một vài chủ đề và đầu tư ít thời gian để xem lại. Tuy nhiên, hợp đồng quy mô lớn có thể yêu cầu sự tham gia của một đội để nghiên cứu một loạt các vấn đề, một quá trình đòi hỏi sự đầu tư của nhiều giờ làm việc.

- Tác nhân thực thi duyệt hợp đồng

Nhiệm vụ duyệt lại hợp đồng có thể được hoàn thành bởi các cá nhân khác nhau, được liệt kê ở đây theo thứ tự tăng dần thông qua sự phức tạp của dự án:

Các nhà lãnh đạo hoặc thành viên khác của nhóm đề xuất.

Các thành viên của đội đề xuất.

Một đội ngũ nhân viên chuyên gia bên ngoài hoặc nhân viên của công ty những người không phải là một thành viên của nhóm đề xuất.

Một nhóm các chuyên gia bên ngoài. Thông thường, một nhóm rà soát lại hợp đồng bao gồm các chuyên gia bên ngoài được mời đến để đưa ra các đề xuất chuyên môn đặc biệt. Các chuyên gia có thể được mời đến để rà soát lại hợp đồng trong các tổ chức phát triển phần mềm nhỏ do không thể tìm thấy đủ các nhân viên thích hợp trong đội ngũ nhân viên của họ.

- Thực hiện rà soát lại hợp đồng cho đề xuất chính

Đề xuất chính được đề nghị cho các dự án đặc trưng bởi ít nhất một số các yếu tố sau: quy mô dự án rất lớn, kỹ thuật rất cao và phức tạp, lĩnh vực chuyên môn mới, tổ chức phức tạp cao (nhận ra bởi một số lượng lớn các tổ chức, nghĩa là, đối tác, nhà thầu phụ, và khách hàng, tham gia một phần trong dự án này). Thực hiện quá trình rà soát lại hợp đồng cho một dự án lớn thường liên quan đến đáng kể đến khó khăn của một tổ chức. Một số con đường để vượt qua những khó khăn được đề nghị ở đây, sau sự xem lại của các yếu tố giới thiệu nhiều khó khăn cho một hoành thành mìn của nhiệm vụ.

4.1.4. Những khó khăn của thực hiện xem lại hợp đồng cho các đề xuất chính

Hầu hết mọi người đều đồng ý rằng xem xét lại hợp đồng là một thủ tục chính cho giảm nguy cơ thất bại của dự án lớn. Về căn bản, việc rà soát hợp đồng là khó khăn, đặc biệt là đối với những tình huống rà soát các yêu cầu đề xuất chính.

■ **Áp lực thời gian.** Cả hai giai đoạn của việc xem xét hợp đồng, đề nghị xem xét lại dự thảo và xem xét dự thảo hợp đồng thường được thực hiện khi nhóm đề xuất là dưới áp lực đáng kể về thời gian. Kết quả là, mỗi giai đoạn của việc xem lại hợp đồng phải được hoàn tất trong vòng một vài ngày để giúp cho người tiếp theo điều chỉnh các văn bản được diễn ra.

Quy tắc duyệt lại hợp đồng yêu cầu phải làm việc chuyên nghiệp. Chuyên nghiệp hiệu suất của mỗi giai đoạn của việc xem lại hợp đồng yêu cầu sự đầu tư chuyên nghiệp đáng kể của các chuyên gia (số lượng thời gian yêu cầu khác nhau, tất nhiên, tùy theo bản chất của dự án).

Các thành viên tiềm năng trong nhóm rà soát Hợp đồng đều rất bận rộn. Những thành viên tiềm năng của đội duyệt hợp đồng thường là nhân viên cấp cao và các chuyên gia và họ thường cam kết thực hiện thường xuyên nhiệm vụ của họ tại tất cả thời gian được xem là cần thiết. Các chuyên gia nhàn rỗi có thể do đó là một vấn đề hậu cần quan trọng.

4.1.5. Khuyến cáo cho việc thực hiện duyệt lại những hợp đồng chính

Một kế hoạch xem lại hợp đồng một cách cẩn thận quyết định thành công của nhóm rà soát hợp đồng. Các bước sau đây được thực hiện để tạo điều kiện cho quy trình rà soát.

■ Các hợp đồng nên được rà soát theo lịch trình. Xem xét lại các hoạt động rà soát hợp đồng nên được đưa vào lịch trình chuẩn bị đề xuất, để lại đầy đủ thời gian cho việc xem lại và chỉnh tiếp theo sẽ được thực hiện.

■ Một nhóm thực hiện rà soát lại các hợp đồng. Làm cho nó có thể làm việc theo nhóm để phân phối các khối lượng công việc giữa các thành viên trong nhóm sao cho mỗi thành viên của đội xem lại hợp đồng có thể có đủ thời gian để làm (có thể bao gồm việc chuẩn bị một bản báo cáo bằng văn bản mà tóm tắt của mình những phát hiện và kiến nghị).

■ Lãnh đạo của đội rà soát hợp đồng nên được bổ nhiệm. Điều quan trọng là trách nhiệm cho các tổ chức, quản lý và kiểm soát các các hoạt động rà soát hợp đồng được

xác định, thích hợp hơn bằng cách bổ nhiệm một lãnh đạo của đội . Cái hoạt động của các lãnh đạo đội bao gồm:

Tuyển dụng của các thành viên trong đội

Phân phối các nhiệm vụ và soát giữa các thành viên của nhóm nghiên cứu

Phối hợp giữa các thành viên của đội duyệt

Phối hợp giữa các đội tuyển xem xét và đề xuất

Theo dõi các hoạt động, đặc biệt là việc tuân thủ theo lịch biểu

Tổng kết những phát hiện và phân phối chúng tới nhóm đề nghị

Lưu ý:

Theo xem lại hợp đồng có thể áp đặt một khối lượng công việc đáng kể và bổ sung áp lực vào các nhóm đề xuất, nghĩ nên được trao cho khi nó có thể được thích hợp để tránh không tiến hành rà soát hợp đồng. Tình huống có thể xảy ra với các dự án quy mô nhỏ, hoặc dự án quy mô chi phí vừa và nhỏ. Các thủ tục rà soát hợp đồng nên được xác định thành những loại dự án mà rà soát lại hợp đồng là không bắt buộc. Đối với các loại khác được định nghĩa dự án "đơn giản" , nó khuyến cáo rằng quyền được trao cho một

4.1.6. Các đối tượng rà soát hợp đồng

Duyệt hợp đồng được thực hiện bởi nhiều đối tượng, dựa trên việc đánh giá mục tiêu của hợp đồng. Bảng kiểm mục (checklist) là công cụ hữu ích cho việc giúp đội xem xét để tổ chức công việc của mình và đạt được nhiều thông tin của những vấn đề liên quan. Rõ ràng là rất nhiều vấn đề trên các danh sách này là không thích hợp đối với bất kỳ dự án cụ thể. Tại cùng một thời điểm, ngay cả một danh sách kiểm tra toàn diện có thể loại trừ một số vấn đề quan trọng có liên quan đến một đề xuất dự án đã định. Đây là nhiệm vụ của đội rà soát hợp đồng, nhưng đặc biệt là của các nhà lãnh đạo, để xác định danh sách các vấn đề thích hợp cho các đề xuất dự án cụ thể.

4.1.7. Rà soát hợp đồng cho các dự án nội bộ

Một số lượng đáng kể, nếu không phải là đa số, các dự án phần mềm là dự án các dự án nội bộ - dự án "trong nhà" - được thực hiện bởi một bộ phận của một tổ chức cho một bộ phận của tổ chức đó. Trong trường hợp như vậy, việc đơn vị phát triển phần mềm là nhà cung cấp, trong khi các đơn vị khác có thể được coi là khách hàng. Thường xuyên, dự án phát triển phần mềm nội bộ không dựa trên những gì sẽ được xem là một mối quan hệ đầy đủ khách hàng -nhà cung cấp . Trong nhiều trường hợp, các dự án này được dựa trên các thỏa thuận chung, với thiện chí đóng vai trò quan trọng trong mối quan hệ giữa hai đơn vị. Nó sau rằng

các đơn vị đang phát triển sẽ thực hiện chỉ một đoạn ngắn và "nhẹ" duyệt hợp , hoặc không ai cả.

Bảng: Chuẩn dự án nội bộ và khách hàng trong cùng tổ chức

Kiểu của dự án	Khách hàng	Ví dụ
Quản trị hoặc các phần mềm thao tác dùng trong nội bộ	Quản trị hoặc thao tác	Hệ thống bán hàng và hàng tồn kho Hệ thống quản lý tài chính
Những gói phần mềm được dự định bán theo kiểu “phần mềm đóng gói”	Thị trường phần mềm	Trò chơi máy tính Phần mềm giáo dục Xử lý ngôn ngữ
Những vi chương trình được nhúng trong sản phẩm của công ty	Bộ phận phát triển điện tử và cơ khí	Công cụ điện tử và các phần mềm điều khiển

Thật không may, mỗi quan hệ lỏng lẻo thường được đặc trưng bởi không đủ thí nghiệm các yêu cầu của dự án, lịch biểu, tài nguyên và phát triển các rủi ro. Kết quả là, những vấn đề sau đây có khả năng xảy ra:

Quá trình định nghĩa không tương xứng với yêu cầu dự án.

Nghèo nàn trong việc đánh giá yêu cầu.

Thời khóa biểu / lập lịch trình nghèo nàn.

Không tương xứng nhận thức về những nguy cơ về rủi ro.

Theo danh sách này cho thấy, chúng ta có thể dễ dàng kết luận rằng “in house”, thực hiện dự án cho các khách hàng nội bộ được nhiều dễ bị thất bại hơn là những hợp đồng dự án bên ngoài .

Có thể kết luận rằng mỗi quan hệ khách hàng-nhà cung cấp và việc xem lại hợp đồng đó được chứng minh là có hiệu quả cho các dự án bên ngoài nên được áp dụng cho các dự án nội bộ. Các cơ hội tránh nêu trên những vấn đề tiềm năng có thể được cải thiện đáng kể bằng cách thực hiện thủ tục đó sẽ xác định:

Một đề xuất phù hợp cho dự án nội bộ

Áp dụng một tiến trình xem lại hợp đồng đúng cách cho các dự án nội bộ

Thỏa thuận thích đáng giữa các khách hàng nội bộ và các nội bộ nhà cung cấp.

Nhược điểm của "các mối quan hệ lỏng" nội bộ dự án

Vấn đề	Nhược điểm của khách hàng	Nhược điểm của nhà phát triển nội bộ
Quá trình định nghĩa tương xứng với dự án	Cài đặt lệch hướng so với yêu cầu của chương trình Hải lòng thấp	Tốn nhân lực để đưa ra những thay đổi có thể tránh
Quá trình định nghĩa không tương xứng với yêu cầu dự án	Mong đợi phi hiện thực một hệ thống mềm dẻo	Độ lệch lớn từ ngân sách phát triển Xung đột được gây ra bởi các yêu cầu cho việc thêm ngân sách
Thời khóa biểu/ lập lịch trình nghèo nàn	Thiếu lịch trình ngay phân phối sản phẩm mới	Hoạt động phát triển dưới thời gian áp lực và khó chịu từ sản phẩm kém chất lượng
Không tương xứng nhận thức về những nguy cơ về rủi ro	Chưa chuẩn bị cho những rủi ro của dự án và những hậu quả của chúng	Sự khởi đầu chậm chạp của nhân lực để khắc phục khó khăn

BÀI 5. CÁC KẾ HOẠCH PHÁT TRIỂN VÀ KẾ HOẠCH CHẤT LƯỢNG

5.1. Các kế hoạch phát triển và kế hoạch chất lượng

5.1.1. Những mục tiêu của kế hoạch phát triển và kế hoạch chất lượng

Kế hoạch, như là một quá trình, có nhiều mục tiêu, mỗi mục tiêu trong số đó có nghĩa là để chuẩn bị đầy đủ trên cơ sở sau đây:

- 1) Các hoạt động lập thời gian biểu phát triển sẽ dẫn đến thành công và kịp thời gian hoàn thành dự án, và ước lượng được nguồn nhân lực, yêu cầu và ngân sách.
- 2) Các thành viên đội tuyển dụng và việc phân bổ nguồn lực phát triển (theo lịch trình hoạt động và yêu cầu ước lượng nguồn nhân lực).
- 3) Giải quyết các rủi ro phát triển.
- 4) Cài đặt các hoạt động được yêu cầu của SQA.
- 5) Cung cấp việc quản lý với dữ liệu cần thiết để kiểm soát dự án.

5.1.2. Các thành phần của kế hoạch phát triển

Căn cứ vào tài liệu đề nghị, kế hoạch phát triển của dự án là chuẩn bị sẵn sàng để hoàn thành các mục tiêu trên. Các yếu tố sau đây áp dụng cho các thành phần của dự án khác nhau bao gồm kế hoạch phát triển dự án.

(1) Sản phẩm dự án:

Kế hoạch phát triển bao gồm các sản phẩm sau:

- Tài liệu thiết kế chi tiết xác định ngày hoàn thành, chỉ ra những gì được chuyển giao cho khách hàng ("phân phối").
- Sản phẩm phần mềm (xác định ngày hoàn thành và cách cài đặt).
- Đào tạo nghiệp vụ (chỉ rõ ngày tháng, người tham gia và các site).

(2) Giao diện dự án:

Dự án bao gồm các giao diện:

- Giao tiếp với các gói phần mềm hiện có (giao diện phần mềm).
- Giao tiếp với các phần mềm khác và / hoặc nhóm phát triển phần cứng đang làm việc trên cùng một hệ thống, dự án (tức là, hợp tác và phối hợp liên kết).
- Giao tiếp với phần cứng hiện tại (giao diện phần cứng).

(3) Phương pháp luận và công cụ phát triển dự án được áp dụng ở mỗi giai đoạn của dự án.

Chỉ dẫn thực hiện

Khi đánh giá sự phù hợp của dự án với các phương pháp và công cụ phát triển được đề xuất, chúng ta cũng nên đưa vào kinh nghiệm chuyên môn của đội ngũ nhân viên, bao gồm cả nhân viên của nhà thầu phụ, thậm chí nếu tạm thời.

(4) Các thủ tục và các chuẩn phát triển phần mềm:

Nên có một danh sách các chuẩn và các thủ tục được áp dụng trong dự án.

(5) Sự ánh xạ các quá trình phát triển:

Việc ánh xạ các quá trình phát triển bao gồm việc cung cấp các định nghĩa chi tiết cho từng giai đoạn của dự án. Những mô tả bao gồm các định nghĩa các yếu tố đầu vào và đầu ra, và các hoạt động cụ thể được lên kế hoạch.

Hoạt động mô tả bao gồm:

- (a) Ước tính về thời gian của hoạt động. Những ước tính phụ thuộc nhiều vào kinh nghiệm thu được trong các dự án trước đó.
- (b) Chuỗi các hoạt động hợp lý, trong đó mỗi hoạt động được thực hiện, bao gồm mô tả sự phụ thuộc của hoạt động vào các hoạt động trước đó đã hoàn thành.
- (c) Các kiểu nguồn lực chuyên môn cần thiết và ước lượng cần bao nhiêu nguồn lực cần thiết cho từng hoạt động.

Chỉ dẫn thực hiện:

Các hoạt động SQA, như xem xét lại thiết kế và test phần mềm, nên được bao gồm trong các hoạt động dự án theo lịch trình. Việc này cũng được áp dụng cho các hoạt động thiết kế và chỉnh sửa mã. Không có kế hoạch các hoạt động này có thể gây ra sự chậm trễ unanticipated trong việc khởi xướng các hoạt động tiếp theo.

Một số phương pháp có sẵn cho việc lập lịch biểu và trình bày bằng hình ảnh (đồ họa) quá trình phát triển. Một trong những phương pháp phổ biến nhất là sử dụng biểu đồ Gantt, hiển thị các hoạt động khác nhau bởi các thanh ngang có độ dài tỉ lệ thuận với thời gian của hoạt động. Các thanh đại diện cho các hoạt động của mình, và được đặt theo chiều dọc, theo kế hoạch và bắt đầu kết luận của chúng. Một vài công

cụ trên máy vi tính có thể chuẩn bị các biểu đồ Gantt, thêm vào để cung cấp một danh sách các hoạt động bởi thời gian cần thiết để bắt đầu và kết thúc của chúng, v.v.

Thêm một phương pháp lập kế hoạch nâng cao, như CPM và Pert, cả hai đều thuộc loại phân tích con đường quan trọng, có trình tự phụ thuộc vào thời gian hoạt động thêm vào. Chúng cho phép tính toán thời gian sớm nhất và muộn nhất được chấp nhận cho mỗi hoạt động. Sự khác nhau giữa các thời gian bắt đầu phụ thuộc vào tính linh hoạt của các hoạt động được lập lịch. Đặc biệt chú ý đến các hoạt động thiếu thời gian biểu linh hoạt (được gọi là các hoạt động “critical path”), và hoàn thành vội vã có thể gây chậm trễ trong việc ký kết của toàn bộ dự án

Một số gói phần mềm, sử dụng kết hợp các phương pháp, hỗ trợ việc lập kế hoạch, báo cáo và theo dõi thời gian biểu của dự án. Một ví dụ về một gói phần mềm của loại hình này là dự Microsoft Project TM. Để thảo luận chi tiết hơn về việc lập lịch, hãy tham khảo các sách viết về quản lý dự án.

(6) Các mốc dự án:

Đối với mỗi mốc quan trọng, thời gian hoàn thành và các sản phẩm của dự án (các tài liệu và mã) được xác định.

(7) Tổ chức nhân viên trong dự án:

Kế hoạch tổ chức bao gồm:

- Cơ cấu tổ chức: định nghĩa các đội dự án và nhiệm vụ của họ, bao gồm các đội được bao gồm cả các công nhân của nhà thầu phụ.
- Các yêu cầu chuyên môn: chứng chỉ chuyên môn, kinh nghiệm trong một ngôn ngữ lập trình cụ thể hoặc công cụ phát triển, kinh nghiệm với một sản phẩm phần mềm cụ thể và các loại,.. v.v.
- Số thành viên của nhóm cần thiết cho từng thời kỳ, theo các hoạt động đã được lập lịch Dự kiến các đội sẽ bắt đầu hoạt động của họ vào các thời điểm khác nhau, và kích thước đội của họ có thể khác nhau tại mỗi khoảng thời gian, tùy thuộc vào các hoạt động đã được lập kế hoạch.
- Tên của các nhà lãnh đạo đội và các thành viên nhóm. Những khó khăn được dự kiến sẽ phát sinh đối với việc giao lâu dài của nhân viên để các đội vì sự thay đổi trong tập unanticipated hiện tại của họ.

Vì vậy, tên của các nhân viên được yêu cầu giúp theo dõi sự tham gia của họ như thành viên của nhóm.

Chỉ dẫn thực hiện:

Tính sẵn có lâu dài của nhân viên dự án nên được kiểm tra cẩn thận. Chậm lại trong hoàn thành nhiệm vụ trước đây có thể dẫn đến sự chậm trễ trong việc gia nhập nhóm dự án, trong đó tăng rủi ro không hợp lại được tại các mốc quan trọng của dự án. Ngoài ra, nhân viên bốc hơi "" gây ra bởi chức và / hoặc các chương trình khuyến mại, hiện tượng đó là đặc biệt thường xuyên trong ngành công nghiệp phần mềm, có thể gây ra tình trạng thiếu nhân viên. Vì vậy, ước lượng khả năng sẵn có của nhân viên nên được kiểm tra định kỳ để tránh những "ngạc nhiên". Cảnh báo sớm về tình trạng thiếu nhân viên bất khả kháng, làm cho nó dễ dàng để giải quyết vấn đề hơn.

(8) Các nhân tố phát triển:

Các nhân tố phát triển bắt buộc bao gồm phần cứng, phần mềm và công cụ phát triển phần cứng, không gian văn phòng, và các mặt hàng khác. Đối với từng nhân tố, khoảng thời gian cần thiết cho việc sử dụng nó nên được ghi trên thời gian biểu.

(9) Các rủi ro phát triển:

Rủi ro phát triển vốn có trong bất kỳ dự án. Để hiểu pervasiveness của chúng, và làm thế nào có thể kiểm soát được chúng, đầu tiên chúng ta phải xác định các khái niệm. Một rủi ro phát triển là "một tiểu bang hoặc tài sản của một công việc hoặc môi trường phát triển, trong đó, nếu bỏ qua, sẽ tăng khả năng thất bại của dự án" (Ropponen và Lyytinen, 2000). Các rủi ro phát triển tiêu biểu là:

- Các khoảng trống công nghệ - Thiếu kiến thức chuyên môn phù hợp và đầy đủ và kinh nghiệm để thực hiện các nhu cầu của các hợp đồng phát triển.
- Thiếu nhân viên - hụt Unanticipated của đội ngũ nhân viên chuyên nghiệp.
- Sự phụ thuộc lẫn nhau của các yếu tố tổ chức - Các khả năng của các nhà cung cấp phần cứng hoặc phần mềm chuyên dụng của nhà thầu phụ, ví dụ, họ sẽ không thực hiện nghĩa vụ của họ như trên thời gian biểu.

Hệ thống hoạt động quản lý rủi ro nên được khởi xướng để đối phó với chúng. Quy trình quản lý rủi ro bao gồm các hoạt động sau đây: nhận biết rủi ro, đánh giá rủi

ro, lập kế hoạch hành động quản lý rủi ro (RMAs), thực hiện RMAs, và giám sát các RMAs.

Phần mềm RMAs được đưa vào kế hoạch phát triển. Tầm quan trọng ngày càng tăng của phần mềm quản lý rủi ro được thể hiện trong mô hình xoắn ốc của các vòng đời phần mềm. Để đối phó với các loại rủi ro, một giai đoạn đặc biệt dành riêng để đánh giá rủi ro phần mềm được chỉ định cho mỗi chu kỳ xoắn ốc(Boehm, 1988, 1998).

(10) Các phương pháp kiểm soát:

Để kiểm soát việc thực hiện dự án, quản lý dự án và quản lý phòng áp dụng một loạt các giám sát thực tiễn khi chuẩn bị báo cáo tiến độ và phối hợp với các cuộc họp.

(11) Ước lượng chi phí dự án:

Ước lượng chi phí dự án là dựa trên đề xuất dự toán chi phí, sau đó là xem xét kỹ lưỡng về sự liên quan tiếp tục của chúng dựa trên các số ước lượng tài nguyên cập nhật con người, thương lượng hợp đồng với nhà thầu phụ và nhà cung cấp, và ..v..v. Ví dụ, một phần của dự án, kế hoạch sẽ được thực hiện bởi một nhóm phát triển nội bộ, cần phải được thực hiện bởi một nhà thầu phụ, do độ chưa sẵn sàng của đội. Một sự thay đổi của bản chất này thường được tham gia vào một ngân sách bổ sung đáng kể..

5.1.3. Các thành phần của kế hoạch chất lượng

Tất cả hay một số mục sau đây, tùy thuộc vào các dự án, bao gồm các thành phần của một kế hoạch quản lý chất lượng dự án :

1. Những mục tiêu của quản lý chất lượng

Thuật ngữ “quality goals” dùng để chỉ các yêu cầu chất lượng thực chất của việc phát triển hệ thống phần mềm . Đơn vị định lượng được ưa thích hơn so với các đơn vị định tính khi lựa chọn mục tiêu chất lượng, bởi chúng cung cấp cho nhà phát triển các đánh giá khách quan hơn về hiệu suất phần mềm trong suốt quá trình phát triển tiến trình và kiểm thử hệ thống. Tuy nhiên, không chỉ có một loại mục tiêu thích ứng với tất cả. Các thể thay thế chất lượng với đơn vị định lượng được minh họa trong ví dụ sau đây

Ví dụ

Một hệ thống phần mềm phục vụ các công việc văn phòng của một nhà máy sản xuất điện được phát triển. Hệ thống trợ giúp công việc văn phòng (HDS – Help desk system) được dự định hoạt động 100 giờ mỗi tuần. Đội quản lý chất lượng phần mềm được yêu cầu chuẩn bị trước 1 danh sách các mục tiêu chất lượng định lượng phù hợp với những yêu cầu chất lượng nhất định, như thể hiện trong bảng sau

HDS qualitative requirements (yêu cầu chất lượng)	Các mục tiêu định lượng chất lượng liên quan
HDS: thân thiện với người sử dụng	Một thao tác mới trợ giúp công việc văn phòng có thể học chi tiết theo một khóa học kéo dài ít hơn 8
HDS: rất tin cậy	HDS luôn sẵn sàng trên 99,5% (thời gian chết của HDS không
HDS: hoạt động liên tục	Thời gian phục hồi hệ thống không được quá 10 phút trong
HDS: có hiệu quả cao	Một thao tác HDS phải có khả năng xử lý ít nhất 100 yêu cầu của
HDS: cung cấp dịch vụ chất lượng cao cho khách hàng	Thời gian chờ đợi cho một thao tác phản hồi không vượt quá 30 giây trong 99% lời yêu cầu. Những thành công của mục tiêu này phụ

Những mục tiêu chất lượng phải phản ánh những tiêu chí chính được chấp nhận và được chỉ ra trong tài liệu yêu cầu của khách hàng(ví dụ các tài liệu RFP). Như vậy, những mục tiêu chất lượng như là đơn vị các thành công của các yêu cầu chất lượng của khách hàng

2. Kế hoạch đánh giá hoạt động

Kế hoạch quản lý chất lượng phải cung cấp một danh sách đầy đủ tất cả các kế hoạch đánh giá hoạt động: đánh giá thiết kế (Design Reviews –DRs), kiểm tra thiết kế, kiểm tra mã (code)…, với những xác định sau cho từng hoạt động :

- Phạm vi đánh giá hoạt động
- Các loại hình đánh giá hoạt động
- Lập lịch đánh giá hoạt động (được định nghĩa bởi độ ưu tiên và các hoạt động thành công của tiến trình dự án)
- Các thủ tục cụ thể được áp dụng
- Ai chịu trách nhiệm thực hiện các hoạt động đánh giá lại ?

3. Kế hoạch kiểm thử phần mềm

Kế hoạch quản lý chất lượng phải cung cấp một danh sách đầy đủ kế hoạch kiểm thử phần mềm, với những thiết kế sau cho mỗi lần kiểm tra:

- Đơn vị, hệ thống tích hợp hay hoàn chỉnh để kiểm tra
- Các loại hình của hoạt động kiểm thử sẽ được thực hiện, bao gồm các đặc điểm kỹ thuật của các lần kiểm thử phần mềm trên máy tính được áp dụng
- Lịch lịch cho kế hoạch kiểm thử (được định nghĩa bởi thứ tự ưu tiên của nó và những hoạt động thành công của tiến trình dự án)
- Các thủ tục cụ thể được áp dụng
- Ai chịu trách nhiệm thực hiện kiểm tra

4. Kế hoạch kiểm thử sự chấp nhận cho phần mềm phát triển bên ngoài

Một danh sách đầy đủ của kế hoạch kiểm thử sự chấp nhận cho phần mềm phát triển bên ngoài phải được cung cấp trong kế hoạch quản lý chất lượng bao gồm các mục :

- (a) Phần mềm được mua
- (b) Phần mềm được phát triển bởi các nhà thầu phụ
- (c) Phần mềm được khách hàng cung cấp

Kiểm thử sự chấp nhận cho phần mềm phát triển bên ngoài nên song song với việc kiểm thử phần mềm phát triển bên trong.

Quản lý cấu hình

Kế hoạch quản lý chất lượng phải xác định các công cụ và thủ tục quản lý cấu hình, bao gồm cả các thủ tục kiểm soát thay đổi và phải áp dụng trong suốt dự án.

Các thành phần quản lý chất lượng phần mềm được yêu cầu được liệt kê trong bảng sau:

Các thành phần của quản lý chất lượng phần mềm
<ol style="list-style-type: none">1. Danh sách các mục tiêu chất lượng2. Đánh giá các hoạt động3. Kiểm thử phần mềm4. Kiểm thử sự chấp nhận phần mềm phát triển bên ngoài5. Các công cụ và thủ tục quản lý cấu hình

** Tài liệu quản lý chất lượng, sự phê duyệt và định dạng của nó*

Quản lý chất lượng có thể được chuẩn bị như một phần của kế hoạch phát triển hay như một tài liệu độc lập. Trong một số trường hợp, kế hoạch được chia thành một số tài liệu bởi hạng mục, như kế hoạch DR, kế hoạch kiểm thử, và kế hoạch kiểm tra sự chấp nhận phần mềm phát triển bên ngoài. Đánh giá và chấp thuận của kế hoạch quản lý chất lượng nên được tiến hành theo thủ tục chuẩn của tổ chức cho các kế hoạch như vậy.

5.1.4. Các kế hoạch phát triển và kế hoạch chất lượng cho các dự án nhỏ và các dự án nội bộ

Việc các nhà lãnh đạo dự án cố gắng tránh khỏi những “rắc rối” trong quá trình chuẩn bị kế hoạch phát triển và quản lý chất lượng là điều tự nhiên. Điều này phản ánh xu hướng tránh việc “làm việc quan liêu” và kiểm soát chung chung mà khách hàng có thể dự tính thực hiện. Xu hướng này đặc biệt thường thấy trong 2 trường hợp: các dự

án nhỏ và các dự án nội bộ. Sự chuẩn bị kế hoạch cho các dự án như vậy sẽ được thảo luận trong 2 phần sau.

- Kế hoạch phát triển và quản lý chất lượng cho các dự án nhỏ

" một dự án chỉ có thời hạn 40 ngày, có thể được thực hiện bởi một chuyên gia và hoàn thành trong 12 tuần, với 1 man-days có phải chuẩn bị lập kế hoạch quản lý chất lượng và phát triển toàn thể ?

" một dự án được thực hiện bởi 3 chuyên gia với tổng vốn là 30 man-days và hoàn thành trong 5 tuần, đòi hỏi phải lập kế hoạch toàn thể?

Cần làm rõ rằng các thủ tục lập kế hoạch quản lý chất lượng và phát triển áp dụng cho các dự án lớn có thể sẽ không áp dụng được cho các dự án nhỏ. Các thủ tục đặc biệt là cần thiết. Các thủ tục này xác định cách giải quyết các dự án như trong các câu hỏi trên với việc chú trọng tới các kế hoạch :

(1) trường hợp không cần lập kế hoạch quản lý chất lượng và phát triển. Ví dụ các dự án đòi hỏi khoảng 15 man-days

(2) trường hợp việc chuẩn bị các kế hoạch phụ thuộc vào quyết định của lãnh đạo dự án. Ví dụ : một dự án đòi hỏi dưới 50 man-days mà không có rủi ro quan trọng đã được xác định. Trong trường hợp này có thể quyết định rằng việc lập kế hoạch là không cần thiết. Một ví dụ khác : một dự án nhỏ nhưng phức tạp cần được hoàn thành trong 30 ngày, trong đó sẽ có một hình phạt nghiêm trọng khi không hoàn thành đúng thời gian. Trong trường hợp này, lập kế hoạch là cần thiết để đối phó với những khó khăn của dự án.

(3) Trường hợp việc lập kế hoạch phát triển và quản lý chất lượng là bắt buộc.

• *Kế hoạch phát triển :*

" Các sản phẩm dự án, chỉ ra “sự phân phối”

" Các điểm chuẩn của dự án

" Các rủi ro trong việc phát triển

" Ước lượng giá thành dự án

• *Kế hoạch quản lý chất lượng :*

" Mục tiêu chất lượng

Một số lợi ích của các dự án nhỏ được lập kế hoạch so với các dự án không được lập kế hoạch có thể được xác định, thậm chí cho các kế hoạch “suy giảm”:

- (1) Sự hiểu biết đầy đủ và triệt để hơn các nhiệm vụ để có thể hoàn thành chúng
 - (2) Trách nhiệm lớn hơn có thể được phân công
 - (3) Dễ dàng hơn cho người quản lý và khách hàng trong việc chia sẻ quyền kiểm soát dự án và sớm phát hiện ra các sự chậm trễ không mong muốn
 - (4) Hiểu biết tốt hơn về các yêu cầu và thời gian biểu được đặt ra giữa người phát triển và khách hàng
- Kế hoạch phát triển và quản lý chất lượng đối với các dự án nội bộ

Các dự án nội bộ là các dự án dự định dành cho các bộ phận khác trong tổ chức hoặc cho toàn bộ tổ chức, hoặc dùng các dự án này trong việc phát triển các gói phần mềm cho thị trường phần mềm. Nhìn chung, tất cả các dự án loại này sẽ không có sự tham gia của các khách hàng bên ngoài trong việc phát triển dự án. Các dự án nội bộ có quy mô vừa hoặc lớn. Tuy nhiên, ngay cả trong trường hợp này cũng có xu hướng tránh việc chuẩn bị các kế hoạch phát triển và quản lý chất lượng một cách đầy đủ. Ví dụ sau minh họa hậu quả tiêu cực của một dự án nội bộ không được lập kế hoạch :

Bộ phận tiếp thị của công ty Toyware, một công ty sản xuất trò chơi vi tính mới, đã có kế hoạch tung ra thị trường “Super – Monster 2000” – trò chơi vi tính mới của công ty – trong mùa Giáng sinh sắp tới. Bộ phận phát triển phần mềm cho rằng việc xây dựng trò chơi nên được bắt đầu ngay lập tức để hoàn thành dự án đúng thời gian. Do đó, việc chuẩn bị một buổi thảo luận giữa bộ phận tiếp thị và bộ phận phát triển; và các kế hoạch phát triển và quản lý chất lượng không được xem là cần thiết. Bộ phận phát triển ước tính ngân sách khoảng 240 000\$, và đã được chuyển giao. Theo lịch tiếp thị, kiểm thử hệ thống phải được hoàn thành trước 1 tháng 10 để bộ phận tiếp thị thực hiện các yêu cầu xúc tiến và quảng cáo chiến dịch trong mùa bán hàng của dịp Giáng sinh. Khi dự án tiến triển, nó có thể có một sự chậm trễ. Vào cuối tháng 6, người ta nhận ra 3 tháng chậm trễ là không thể tránh khỏi. Các hoạt động quảng cáo diễn ra trước 30.6 trở thành vô nghĩa. Cuối cùng, dự án hoàn thành vào cuối tháng 2. Giá thành của dự án vượt đáng kể - thực tế chi phí vượt quá 385 000\$ -

nhưng thiệt hại lớn nhất là công ty mất cơ hội khai thác thị trường mùa Giáng sinh. Tuần trước, quản lý công ty quyết định sẽ tránh tất cả các dự án nội bộ phát triển trò chơi vì tính trong tương lai.

Ví dụ này cho thấy rõ ràng là việc chuẩn bị các kế hoạch phát triển và quản lý chất lượng toàn bộ cho các dự án nội bộ - kể cả việc giám sát thường xuyên – có thể rất có lợi cho việc thực hiện các dự án nội bộ.

Bộ phận phát triển phần mềm có thể đạt được những thuận lợi sau đây với việc chuẩn bị kế hoạch :

- (1) Tránh vượt quá ngân sách. Điều này có tầm quan trọng đặc biệt trong việc đảm bảo lợi nhuận.
- (2) Tránh thiệt hại cho các dự án khác do các chuyên gia tham gia vào dự án nội bộ bị chậm trễ nên chưa thể tham gia vào dự án khác.
- (3) Tránh trường hợp mất thị trường, đặc biệt liên quan đến danh tiếng của công ty, do các dự án nội bộ chậm trễ kéo theo các dự án bên ngoài phụ thuộc vào nó cũng bị trễ.

Khách hàng nội bộ có thể có những thuận lợi sau :

- (1) Ít trễ hơn và ngân sách ít bị vượt quá hơn.
- (2) Kiểm soát tốt hơn quá trình phát triển, trong đó có việc xác định sự chậm trễ sớm hơn, từ đó có thể tìm cách giải quyết nguyên nhân của chúng.
- (3) Ít thiệt hại chậm trễ

Nội bộ hơn Tổ chức có thể có những lợi ích sau :

- (1) Giảm rủi ro mất thị trường do sản phẩm bị trễ
- (2) Giảm rủi ro bị kiện do hoàn thành các sản phẩm bị muộn, vì thế, giảm thiệt hại vì vi phạm hợp đồng

Giảm nguy cơ danh tiếng của công ty bị ảnh hưởng Giảm rủi ro phải bổ sung ngân sách

BÀI 6. THẢO LUẬN VỀ KẾ HOẠCH CHẤT LƯỢNG – SQA plan

6.1. Mẫu kế hoạch chất lượng theo chuẩn IEEE

6.2. Cấu trúc các thành phần chính trong kế hoạch chất lượng

BÀI 7. CÁC KỸ THUẬT RÀ SOÁT

7.1. Tích hợp các hoạt động chất lượng trong vòng đời dự án

7.1.1. Phương pháp phát triển phần mềm truyền thống và các phương pháp khác

Bốn mô hình của quy trình phát triển được bàn luận trong phần này là:

- Mô hình vòng đời phát triển phần mềm (SDLC).
- Mô hình bản mẫu nhanh.
- Mô hình xoắn ốc.
- Mô hình hướng đối tượng.

Mô hình vòng đời phát triển phần mềm (SDLC) là một mô hình truyền thống (hiện nay vẫn có thể áp dụng được); nó cung cấp sự mô tả toàn diện nhất về quy trình phát triển phần mềm. Mô hình chỉ ra cần xây dựng những khối chính cho toàn bộ quá trình phát triển, được mô tả như là một chuỗi tuyến tính. Trong pha ban đầu của quy trình phát triển phần mềm, các tài liệu thiết kế sản phẩm được chuẩn bị, việc đánh giá phiên bản đầu tiên của chương trình máy tính chỉ diễn ra ở giai đoạn cuối của quy trình. Mô hình SDLC có thể đóng vai trò như một khung (framework) để biểu diễn các mô hình khác.

Mô hình bản mẫu dựa trên sự thay thế của một hoặc nhiều pha trong mô hình SDLC bằng một quy trình đánh giá, các bản mẫu phần mềm được sử dụng cho quá trình giao tiếp giữa người phát triển, người sử dụng cuối và khách hàng. Các bản mẫu được đưa ra để người sử dụng đánh giá. Sau đó, người phát triển tiếp tục phát triển một bản mẫu nâng cao, bản mẫu này cũng phải được đưa ra để đánh giá. Quá trình đánh giá này tiếp tục cho đến khi dự án phần mềm được hoàn thiện hoặc bản mẫu phần mềm đã đạt được những gì mà pha đó yêu cầu. Trong trường hợp này, phần còn lại của quy trình phát triển có thể được thực hiện theo một phương pháp luận khác, chẳng hạn như theo mô hình SDLC truyền thống.

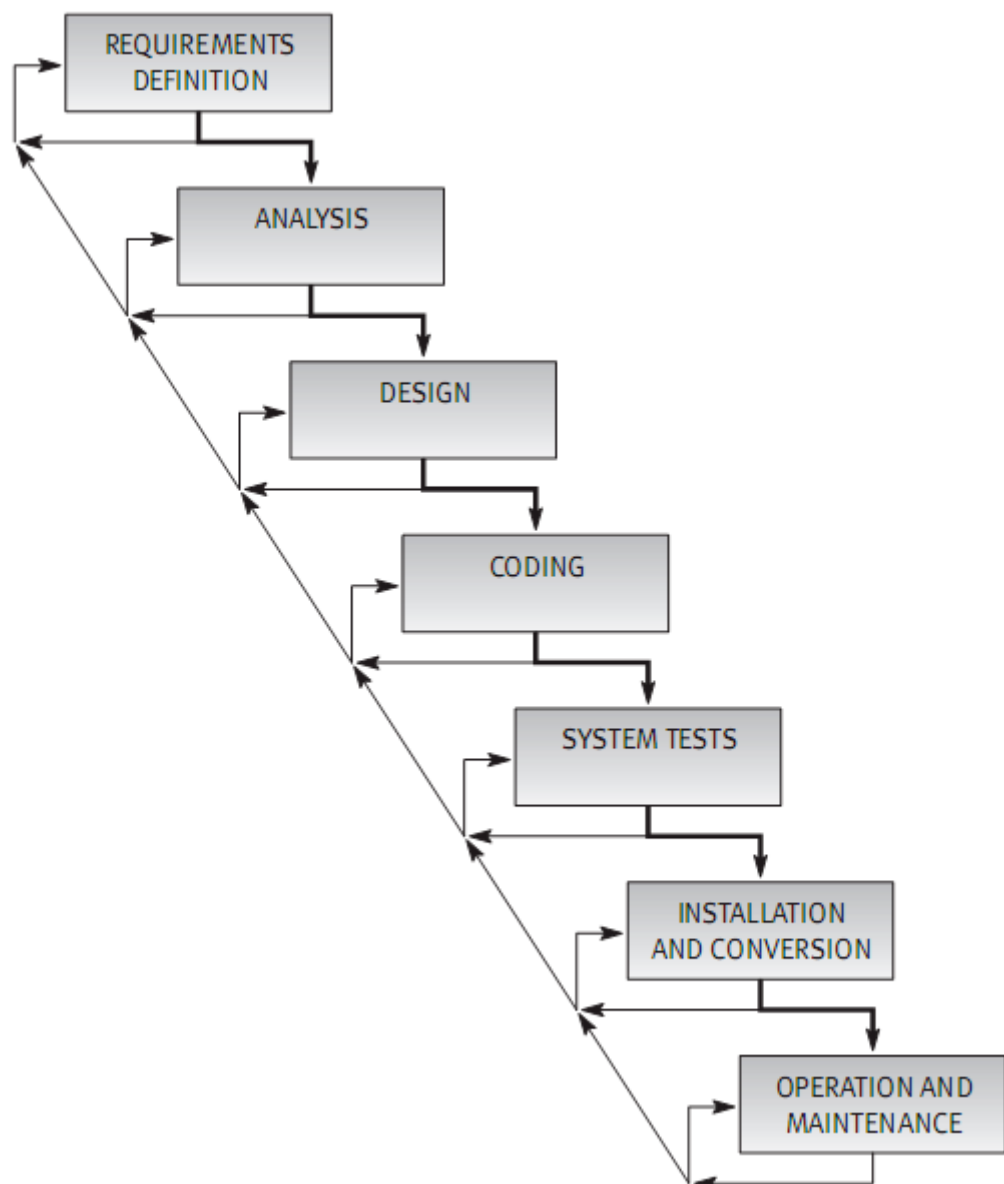
Mô hình xoắn ốc cung cấp một phương pháp luận nhằm đảm bảo hiệu quả về mặt hiệu năng của mỗi pha trong mô hình SDLC truyền thống. Mô hình này liên quan đến một quy trình lặp, tích hợp các yêu cầu thay đổi của khách hàng, phân tích và

giải quyết các rủi ro, các hoạt động về mặt kỹ nghệ và kế hoạch cho phát triển phần mềm. Một trong những tương tác của mô hình xoắn ốc có thể là yêu cầu hoàn thiện ở mỗi pha của dự án trong mô hình SDLC.

Mô hình hướng đối tượng kết hợp chặt chẽ việc sử dụng lại các phần mềm cỡ lớn bằng cách kết hợp các mô đun có thể sử dụng lại được thành các hệ thống phần mềm mới. Trong trường hợp không có mô đun phần mềm nào có thể sử dụng được (theo thuật ngữ đối tượng hoặc thành phần) sẵn có, người phát triển có thể thực hiện một quy trình bản mẫu hoặc quy trình SDLC để hoàn thành việc phát triển một hệ thống mới.

- Mô hình vòng đời phát triển phần mềm (SDLC)

SDLC là mô hình tuyến tính bắt đầu bằng việc định nghĩa các yêu cầu và kết thúc bằng sự vận hành của hệ thống và quá trình bảo trì. Thể hiện phổ biến nhất của SDLC là mô hình waterfall (thác nước). Mô hình này được minh họa trong hình dưới đây:



Hình Mô hình thác nước

Mô hình gồm có 7 pha: *xác định yêu cầu, phân tích, thiết kế, coding, kiểm thử hệ thống, cài đặt và chuyển giao, vận hành và bảo trì.*

Xác định yêu cầu: Mục đích của pha xác định yêu cầu là xác định các chức năng của hệ thống cần xây dựng, khách hàng phải đưa ra và xác định các quyết định của họ. Trong nhiều trường hợp hệ thống phần mềm là một phần của hệ thống lớn hơn. Thông tin về các phần khác của hệ thống được mở rộng sẽ giúp thiết lập sự cộng tác giữa đội dự án và các giao diện thành phần phát triển.

Phân tích: Nỗ lực chính ở đây là phân tích sự ảnh hưởng của những yêu cầu thành mô hình khởi tạo của hệ thống phần mềm.

Thiết kế: Giai đoạn này bao gồm việc định nghĩa chi tiết đầu vào, đầu ra và các thủ tục xử lý, bao gồm cấu trúc dữ liệu, cơ sở dữ liệu, cấu trúc phần mềm, ...

Coding: Trong pha này, toàn bộ thiết kế được chuyển thành code. Việc coding bao gồm những hoạt động đảm bảo chất lượng phần mềm như inspection, unit test, và test tích hợp.

Test hệ thống (System test): Test hệ thống được thực hiện sau khi pha coding hoàn thiện. Mục đích chính của việc kiểm thử là càng tìm ra nhiều lỗi phần mềm càng tốt để đạt được mức độ chấp nhận của chất lượng phần mềm. Kiểm thử hệ thống được thực hiện bởi những người phát triển trước khi bàn giao cho khách hàng. Trong nhiều trường hợp, khách hàng thực hiện kiểm thử phần mềm một cách độc lập (còn gọi là test chấp nhận sản phẩm) để đảm bảo rằng những người phát triển đã thỏa mãn tất cả những hứa hẹn và những phản ứng phần mềm bất ngờ hoặc lỗi có thể được thấy trước. Đây là điều khá phổ biến, khách hàng yêu cầu người phát triển để họ được tham gia vào trong quá trình kiểm thử hệ thống, một thủ tục đã tiết kiệm được về mặt thời gian và tài nguyên cho việc phân tách riêng biệt giữa test hệ thống và test chấp nhận.

Cài đặt và chuyển giao: Sau khi hệ thống phần mềm được phê chuẩn, hệ thống được cài đặt để phục vụ như là một phần sụn, có nghĩa là, nó như là một phần của hệ thống thông tin, biểu diễn một thành phần cơ bản của hệ thống được mở rộng. Nếu hệ thống thông tin mới được xây dựng để thay thế hệ thống đang tồn tại, một quy trình chuyển giao phải được khởi tạo để đảm

bảo rằng các hoạt động của tổ chức vẫn được tiếp tục mà không bị gián đoạn trong suốt pha chuyển giao.

Vận hành và bảo trì: Vận hành phần mềm bắt đầu sau khi pha cài đặt và chuyển giao đã được xong. Xuyên suốt thời kì vận hành, thường ít nhất là một vài năm hoặc cho đến khi xuất hiện phần mềm mới, việc bảo trì là cần thiết. Việc bảo trì kết hợp ba kiểu dịch vụ: thích ứng – sử dụng các đặc trưng của phần mềm đang tồn tại để thực hiện các yêu cầu mới; hoàn thiện – thêm một số đặc trưng để cải thiện hiệu năng của phần mềm; sửa lỗi – sửa các lỗi phần mềm được tìm thấy bởi người sử dụng trong quá trình vận hành.

Số lượng các pha có thể thay đổi tùy theo đặc trưng của từng dự án. Trong các dự án phức tạp, các mô hình phần mềm cỡ lớn, một số pha có thể bị chia ra, do đó, số lượng các pha có thể lên đến tám, chín hoặc thậm chí là nhiều pha hơn nữa. Trong các dự án nhỏ, một số pha lại có thể bị gộp lại, giảm số lượng pha của SDLC xuống còn sáu, năm, thậm chí là chỉ còn bốn pha.

Ở cuối mỗi pha, các đầu ra được xem xét và đánh giá bởi người phát triển, và trong một số trường hợp cũng có thể là khách hàng tham gia. Các kết quả có thể của quá trình xem xét lại bao gồm:

- Sự phê chuẩn của các đầu ra ở pha đó và quy trình của pha tiếp theo
- Các yêu cầu chỉnh sửa, làm lại hoặc thay đổi các phần của pha cuối cùng; trong trường hợp đảm bảo tính chắc chắn, có thể trả về các pha gần nhất theo yêu cầu.

Độ rộng của đường kết nối giữa các hình hộp chữ nhật trong hình minh họa phản ánh khả năng có những kết quả khác nhau. Do đó, hầu hết các quy trình được thực hiện như là một dòng tuyến tính. Tuy nhiên, cần chú ý rằng, mô hình nhấn mạnh các hoạt động phát triển trực tiếp và không chỉ ra sự tham gia của khách hàng trong quy trình phát triển.

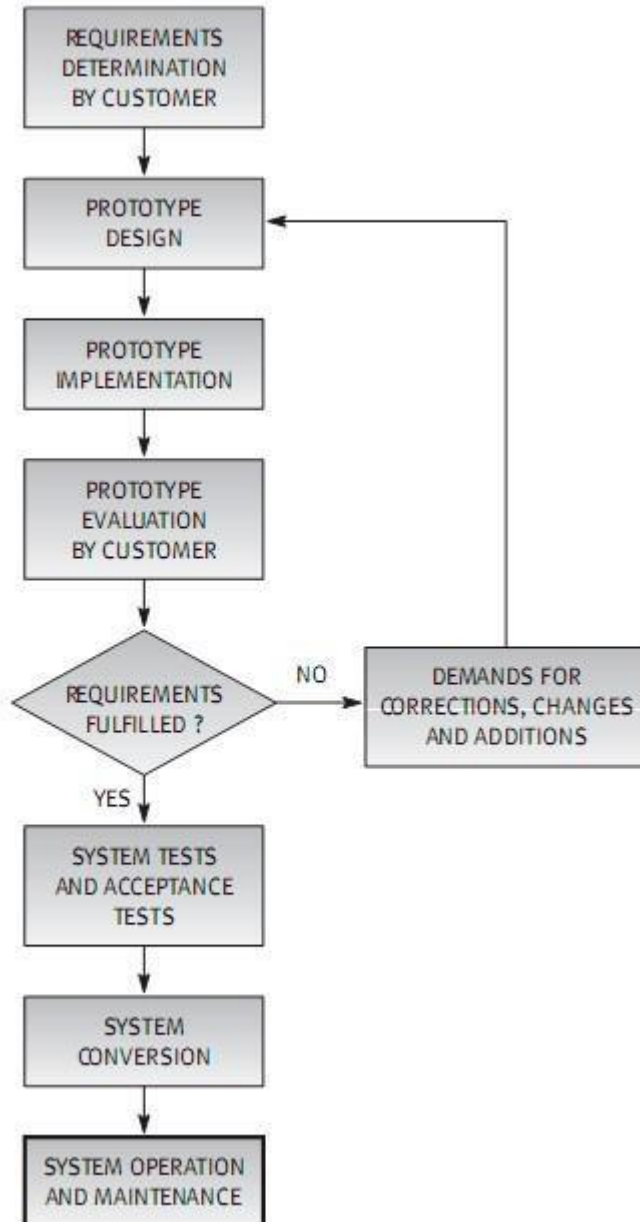
Mô hình Waterfall đã được đề xuất bởi Royce (1970) và sau đó đưa ra trong mô hình chung của nó được biết đến như là Boehm (1981). Mô hình này cung cấp các nền tảng cho phần lớn các chuẩn đảm bảo chất lượng phần mềm đã được triển khai, chẳng hạn như chuẩn IEEE 1012, IEEE 12207, ...

- Mô hình bản mẫu

Phương pháp bản mẫu:

- Cho phép phát triển nhanh và dễ dàng các bản mẫu phần mềm.
- Kết hợp với sự tham gia của khách hàng và người sử dụng trong quy trình phát triển để kiểm tra và đánh giá bản mẫu.

Khi áp dụng phương pháp bản mẫu, những người sử dụng tương lai của hệ thống được yêu cầu bình luận các phiên bản của bản mẫu phần mềm khác nhau đã được chuẩn bị bởi người phát triển. Từ sự phản hồi của khách hàng và người sử dụng, người phát triển sẽ sửa bản mẫu cho phù hợp, cho đúng và thêm các phần vào hệ thống theo cách biểu diễn như là một phiên bản phần mềm tiếp theo để người sử dụng đánh giá. Quy trình này được lặp lại cho đến khi đạt được mục đích của bản mẫu hoặc toàn bộ hệ thống phần mềm được hoàn thiện. Ứng dụng điển hình của phương pháp luận bản mẫu được minh họa trong hình dưới đây:



Mô hình bản mẫu

Phương pháp bản mẫu có thể được áp dụng trong sự kết hợp với các phương pháp luận khác hoặc có thể xem như là một phương pháp luận độc lập. Nói cách khác, sự mở rộng phương pháp bản mẫu có thể thay đổi, từ thay thế một pha SDLC (hoặc một

pha trong phương pháp luận khác) đến bản mẫu hoàn thiện của toàn bộ hệ thống phần mềm.

Phương pháp bản mẫu như là một phương pháp luận phát triển phần mềm đã được áp dụng có hiệu quả cho các loại dự án có kích thước từ nhỏ đến trung bình. Ưu điểm cũng như thiếu sót chính của phương pháp bản mẫu so với SDLC được tổng kết lại trong bảng dưới đây:

Ưu điểm:

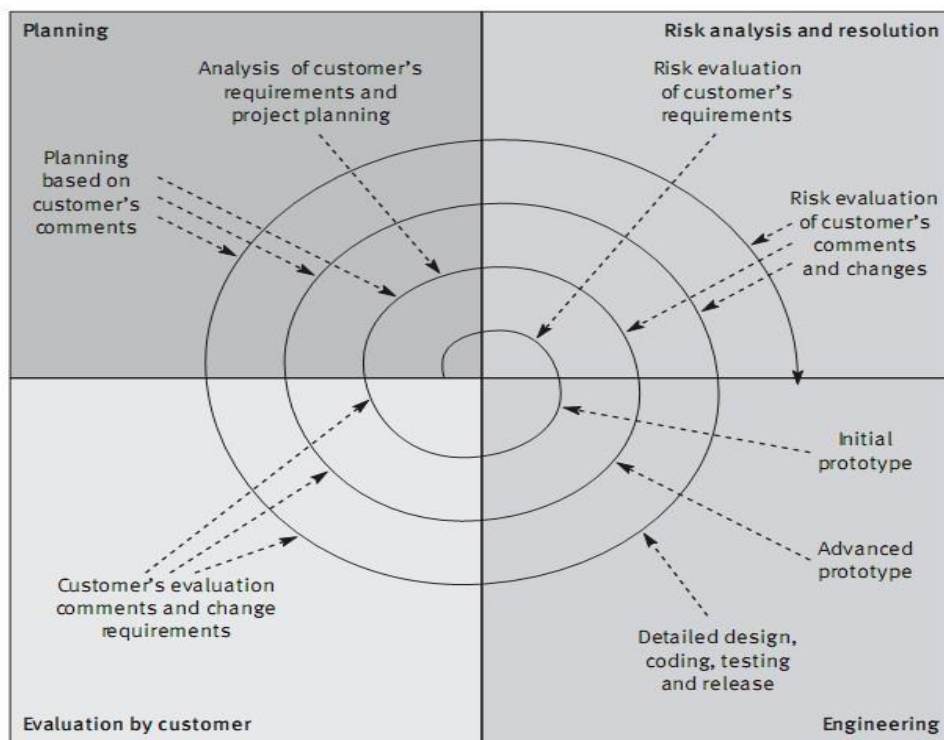
- Quy trình phát triển ngắn.
- Tiết kiệm tài nguyên (man-days).
- Phù hợp với các yêu cầu của khách hàng và giảm thiểu được rủi ro của dự án.
- Sự tiếp nhận (lĩnh hội) của người dùng về hệ thống mới nhanh hơn và dễ dàng hơn.

Nhược điểm:

- Giảm bớt khả năng mềm dẻo và thích nghi đối với những thay đổi và những bổ sung.
- Giảm sự chuẩn bị cho các trường hợp lỗi không mong đợi.
- Mô hình xoắn ốc

Mô hình xoắn ốc, được xem xét bởi Boehm (1988, 1998), đưa ra một phương pháp luận cải thiện nhằm phục vụ cho các dự án cỡ lớn và phức tạp.

Mô hình này được minh họa như hình dưới đây:

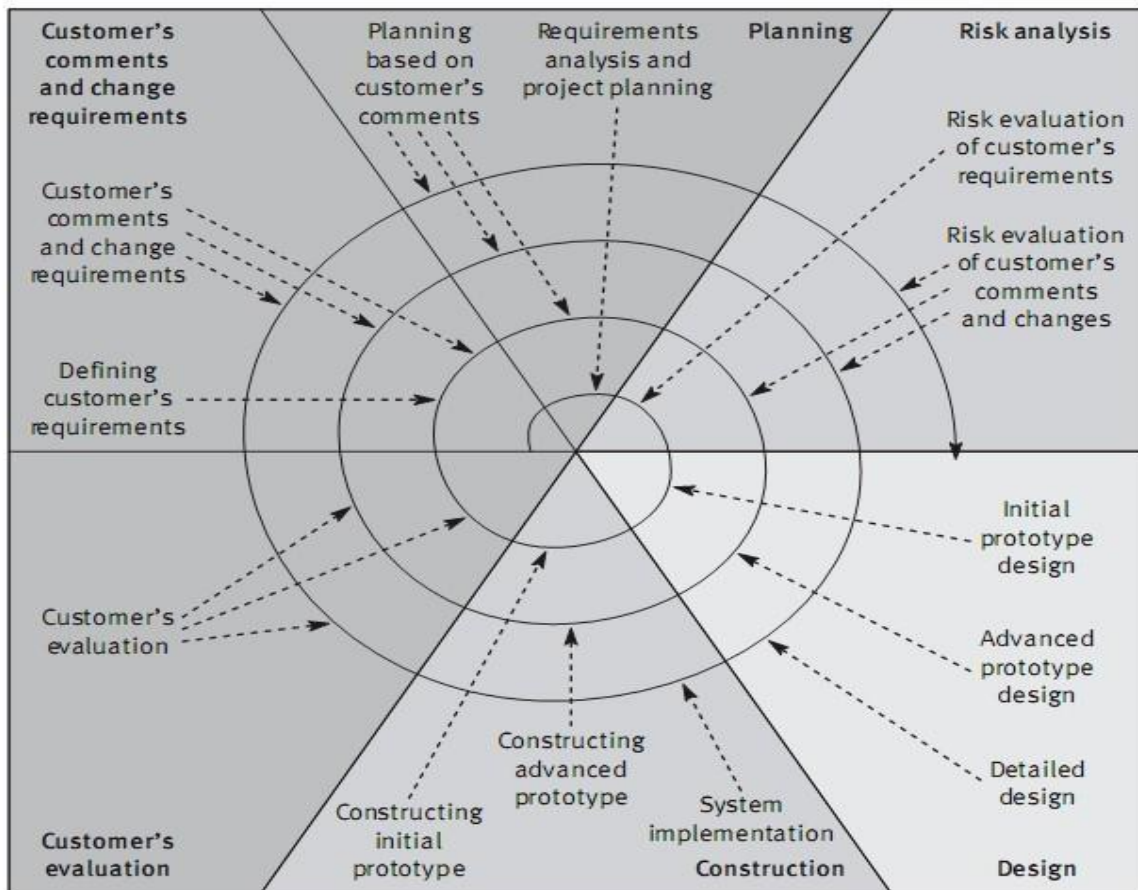


Mô hình xoắn ốc

Theo mô hình xoắn ốc, sự phát triển phần mềm được hiểu như làm một quy trình lặp; tại mỗi lần lặp, các hoạt động sau được thực hiện:

- Lập kế hoạch.
- Phân tích và giải quyết rủi ro.
- Các hoạt động kỹ nghệ theo giai đoạn của dự án: thiết kế, coding, test, cài đặt và chuyển giao.
- Sự đánh giá của khách hàng bao gồm các bình luận, các thay đổi và các yêu cầu bổ sung, ...

Một mô hình xoắn ốc tiên tiến, là mô hình xoắn ốc win - win. Mô hình xoắn ốc tiên tiến nhấn mạnh vào sự giao tiếp và thương lượng giữa khách hàng và người phát triển. Tên của mô hình ám chỉ thực tế rằng, bằng việc sử dụng mô hình này, khách hàng có thể nhận được hệ thống thỏa mãn nhất cái họ cần, và nhà phát triển có khả năng nhận được nhiều tiền và hoàn thành dự án đúng ngày. Điều này đạt được bằng việc nhấn mạnh vào sự tham gia của khách hàng và những hoạt động kỹ nghệ. Việc xem xét lại trong quá trình phát triển được diễn tả bằng hai đoạn đồ thị của xoắn ốc, với sự tham gia của khách hàng: đầu tiên là sự đánh giá của khách hàng, thứ hai là sự phản hồi và những yêu cầu thay đổi của khách hàng. Tương tự như vậy, những hoạt động kỹ nghệ được miêu tả bằng hai đoạn trong hình xoắn ốc: đầu tiên là thiết kế, và thứ hai là xây dựng. Bằng sự đánh giá tiến trình dự án ở cuối mỗi đoạn, nhà phát triển có thể điều chỉnh tốt hơn toàn bộ quá trình phát triển.



Mô hình xoắn ốc cải tiến

Vì vậy, trong mô hình xoắn ốc cải tiến, sáu hoạt động sau phải thực hiện trong mỗi lần lặp:

- Những đặc tả yêu cầu, những phản hồi và những đòi hỏi thay đổi của khách hàng.
- Những hoạt động lập kế hoạch của nhà phát triển.
- Phân tích và giải quyết những rủi ro của nhà phát triển.
- Những hoạt động thiết kế của nhà phát triển.
- Những hoạt động xây dựng của nhà phát triển, liên quan tới việc coding, test, cài đặt và release.
- Sự đánh giá của khách hàng.

- Mô hình hướng đối tượng

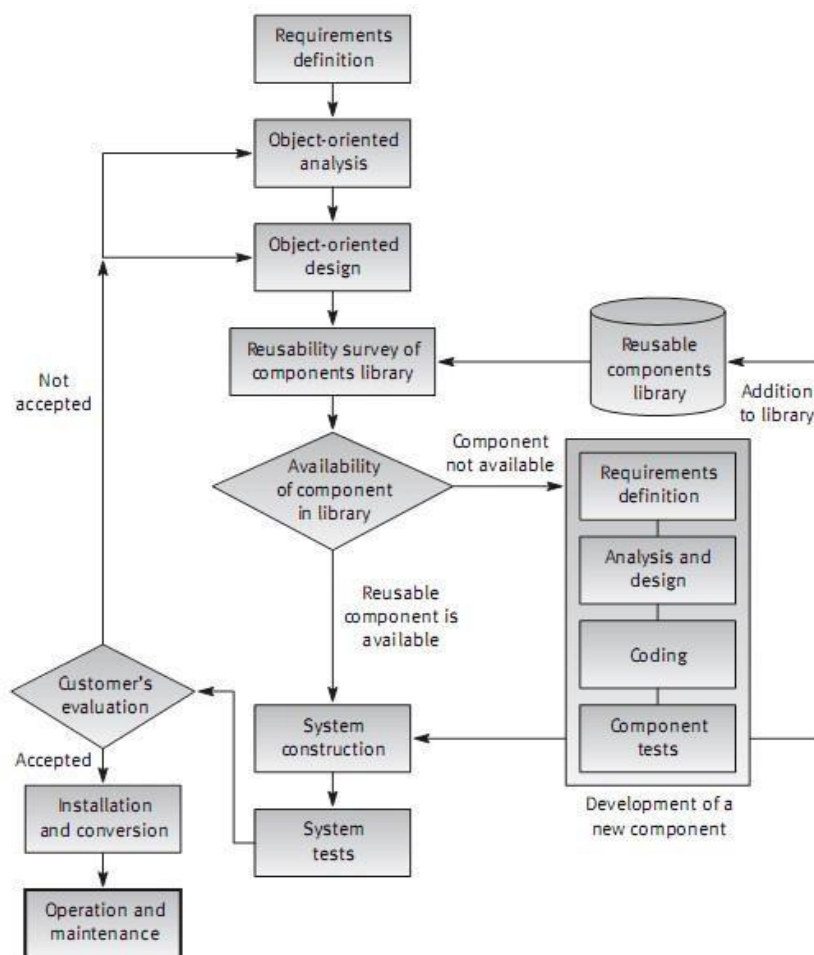
Mô hình hướng đối tượng khác với các mô hình khác bởi nó hướng tới việc sử dụng lại các thành phần phần mềm. Đặc trưng của phương pháp luận này là nó dễ dàng tích hợp các mô đun phần mềm đang tồn tại (được gọi là các đối tượng hoặc các thành phần) vào trong những hệ thống mới. Một thư viện thành phần phần mềm phục vụ

cho mục đích này bằng việc cung cấp những thành phần phần mềm cho việc sử dụng lại.

Vì thế, theo mô hình hướng đối tượng được chỉ ra trong hình 7.5, quy trình phát triển bắt đầu với chuỗi phân tích và thiết kế hướng đối tượng. Pha thiết kế được thực hiện bằng sự thu nhận các thành phần phù hợp từ thư viện phần mềm có thể sử dụng lại, khi các thành phần đó là ở trạng thái sẵn sàng trong thư viện. Sự phát triển “thông thường - regular” được thực hiện theo cách khác. Sao chép những thành phần phần mềm mới và sau đó đưa nó vào trong thư viện phần mềm cho mục đích sử dụng lại trong tương lai. Phương pháp này mong đợi sự lớn lên của kho thư viện thành phần phần mềm sẽ cho phép tăng việc sử dụng và chắc chắn sử dụng lại của phần mềm, một hướng sẽ cho phép mang lại ưu điểm lớn về mặt tài nguyên như sau:

- Kinh tế: Chi phí tích hợp thành phần phần mềm sử dụng lại ít hơn so với giá phát triển một phần mềm mới.
- Cải thiện chất lượng: Các thành phần phần mềm được sử dụng ít lỗi hơn là thành phần phần mềm mới xây dựng.
- Thời gian phát triển ngắn hơn: Việc tích hợp các thành phần phần mềm sử dụng lại làm giảm áp lực về mặt lịch biểu.

Như vậy, ưu điểm của phương pháp luận hướng đối tượng so với phương pháp luận khác là sẽ làm tăng sự phát triển của kho lưu trữ phần mềm có thể sử dụng lại được.



Mô hình hướng đối tượng

7.1.2. Các yếu tố ảnh hưởng hoạt động đảm bảo chất lượng phần mềm

Những hoạt động đảm bảo chất lượng là hướng quy trình, nói cách khác, có liên kết tới sự hoàn thành của một pha dự án, sự hoàn tất một mốc dự án, và nhiều hơn nữa. Những hoạt động đảm bảo chất lượng được tích hợp vào trong kế hoạch phát triển.

Những người lên kế hoạch đảm bảo chất lượng cho một dự án cần xác định:

- Danh sách những hoạt động đảm bảo chất lượng cần thiết cho dự án.
- Với mỗi hoạt động đảm bảo chất lượng:
 - " Thời gian.
 - " Loại hoạt động đảm bảo chất lượng áp dụng.
 - " Người thực hiện hoạt động và tài nguyên yêu cầu.
 - " Những tài nguyên yêu cầu cho việc khắc phục những nhược sai sót và những thay đổi.

Cường độ của những hoạt động đảm bảo chất lượng đã được lên kế hoạch được chỉ ra bởi số những hoạt động yêu cầu. Những yếu tố dự án và nhóm ảnh hưởng tới cường độ như sau:

Yếu tố dự án:

- Độ lớn của dự án.
- Sự phức tạp và khó của kỹ thuật.
- Phạm vi của những thành phần sử dụng lại.
- Hậu quả nếu dự án bị lỗi.

Yếu tố nhóm:

- Trình độ chuyên môn của những thành viên trong nhóm.
 - Sự quen thuộc của nhóm với dự án và Kinh nghiệm của nhóm trong lĩnh vực.
 - Tính sẵn sàng của những thành viên có thể trợ giúp nhóm.
- Sự hiểu biết giữa những thành viên trong nhóm, hay nói cách khác là số thành viên mới trong nhóm

7.1.3. Xác minh, thẩm định và đánh giá chất lượng

Ba khía cạnh đảm bảo chất lượng của sản phẩm phần mềm được sát hạch dưới những dạng Verification, Validation, và Qualification.

- ***Verification:*** quá trình đánh giá một hệ thống hay một thành phần để xác định xem những sản phẩm của một pha phát triển xác định có thỏa mãn những điều kiện được đặt gia khi bắt đầu pha đó hay không.
- ***Validation:*** quá trình đánh giá một hệ thống hay một thành phần trong suốt hoặc khi kết thúc quá trình phát triển để xác định xem nó có thỏa mãn những yêu cầu đã đặc tả hay không.
- ***Qualification:*** quá trình xác định một hệ thống hoặc một thành phần có phù hợp với việc sử dụng hay không.

Theo những định nghĩa của IEEE, verification kiểm tra tính nhất quán của sản phẩm đang phát triển với những sản phẩm đã được phát triển ở pha trước. Khi thực hiện, người thẩm tra đi sau quy trình phát triển và giả sử rằng tất cả những pha phát triển đang

trước đã được hoàn thành một cách chính xác hoặc là như kế hoặc gốc hoặc là sau khi đã sửa chữa những sai sót được phát hiện.

Validation miêu tả sự quan tâm của khách hàng, bằng cách thăm tra những yêu cầu gốc của họ.

Qualification tập trung vào những khía cạnh hoạt động, ở đó việc bảo trì là vấn đề chính. Một thành phần phần mềm đã được phát triển và tài liệu hóa theo những chuẩn, kiểu, và cấu trúc chuyên nghiệp sẽ dễ dàng để bảo trì hơn những thành phần có code lạ.

Người lên kế hoạch cần phải xác định xem những khía cạnh nào nên được sát hạch trong mỗi hoạt động đảm bảo chất lượng phần mềm.

7.2. Rà soát

7.2.1. Mục tiêu rà soát

Mục đích được chia làm 2 loại: mục đích trực tiếp và gián tiếp. Mục đích trực tiếp:

- Phát hiện lỗi phân tích và thiết kế.
- Xác định các rủi ro mới.
- Xác định sự sai lệch so với mẫu, các kiểu thủ tục và qui ước.
- Để phê chuẩn sản phẩm của phân tích hoặc thiết kế.

Mục đích gián tiếp:

- Nơi họp mặt không chính thức để trao đổi về những kiến thức chuyên môn.
- Ghi lại những lỗi phân tích và thiết kế sẽ hỗ trợ một cơ sở cho những hoạt động sửa chữa lỗi trong tương lai.

7.2.2. Những rà soát thiết kế hình thức

Rà soát thiết kế hình thức (DRs-formal Design Reviews) là rà soát duy nhất cần thiết cho việc phê duyệt sản phẩm thiết kế

Rà soát thiết kế hình thức có thể được thực hiện tại bất cứ mốc phát triển nào yêu cầu sự hoàn thiện của tài liệu phân tích hay thiết kế.

Một danh sách các review thiết kế chính thức :

- DPR – Development Plan Review : Review kế hoạch phát triển
- SRSR – Software Requirement Specification Review : Review đặc tả yêu cầu phần mềm
- PDR – Preliminary Design Review : Review thiết kế sơ bộ
- DBDR- Detailed Design Review : Review thiết kế chi tiết
- TPR – Test Plan Review : Review kế hoạch kiểm thử
- STPR – Software Test Procedure Review : Review thủ tục kiểm thử phần mềm
- VDR- Version Description Review : Review mô tả phiên bản
- OMR- Operator Manual Review : Review vận hành thủ công
- SMR- Support Manual Review :Review trợ giúp thủ công
- TRR- Test Readiness Review : Review sự sẵn sàng kiểm thử
- PRR- Product Release Review : Review bản phát hành sản phẩm
- IPR-Installation Plan Review : Review kế hoạch cài đặt

Các nhân tố ảnh hưởng tới DRs

- Những người tham gia
- Sự chuẩn bị trước
- Phiên DR
- Các hoạt động sau DR được đề xuất
- **Những người tham gia rà soát thiết kế**

Gồm có Review leader và review team.

Review Leader :

- Có kiến thức và kinh nghiệm trong việc phát triển kiểu dự án được review
- Có thâm niên ở mức độ bằng với hoặc cao hơn của project leader
- Có mối quan hệ tốt với project leader và đội dự án
- Có vị trí bên ngoài đội dự án

Review team

- Phần lớn không thuộc đội dự án
- Kích thước từ 3-5 người

– Đa dạng về kinh nghiệm và phương pháp

- Sự chuẩn bị cho một phiên bản DR

Được hoàn thành bởi 3 thành viên: review leader, review team và development team..

Chuẩn bị của Review leader

- Bổ nhiệm các thành viên nhóm
- Lập lịch các phiên review
- Phân chia tài liệu thiết kế cho các thành viên của nhóm

Chuẩn bị của Review team

- Xem lại tài liệu thiết kế
- Danh sách bình luận

Chuẩn bị của Development team

- Trình diễn ngắn tài liệu thiết kế
- Checklist các công việc review

- Phiên DR

Một cuộc họp phiên DR thông thường gồm có :

1. Trình diễn ngắn gọn về tài liệu thiết kế
2. Các bình luận của các thành viên review team
3. Kiểm tra và xác nhận thảo luận mỗi bình luận
4. Các quyết định về tài liệu thiết kế để xác định tiến trình dự án. Các quyết định có thể có 3 loại :

- Phê duyệt đầy đủ
- Phê duyệt từng phần
- Từ chối phê duyệt

- Các hoạt động hậu review

- Báo cáo Review
- Review leader thực hiện sau phiên review
- Bao gồm :

- Tổng kết các thảo luận review
- Quyết định về sự tiếp tục của dự án
- Danh sách các hoạt động cần thiết phải làm
- Tên thành viên chịu trách nhiệm theo sát việc hiệu chỉnh
- Tiến trình theo dõi
 - Review leader thực hiện
 - Chắc rằng việc hiệu chỉnh được thực hiện đúng đắn
 - Việc theo dõi cần được ghi lại

7.2.3. Các rà soát ngang hàng (peer review)

Mục đích chính của peer review là xác định lỗi và độ lệch dựa vào các chuẩn.

Có hai phương pháp peer reviews:

- xét duyệt (inspection)
- kiểm tra từng bước (walkthrough).

Walkthrough phát hiện sai sót và ghi chú lên tài liệu.

Inspection phát hiện sai sót và kết hợp với nỗ lực để cải tiến.

- Những người tham gia vào peer reviews

Một đội peer review tối ưu 3-5 người tham gia.

Tất cả những người tham gia nên là những người cùng địa vị của nhà thiết kế hệ thống phần mềm.

Một đội peer review đề cử bao gồm:

- Một leader review.
- Một người thực thi (author).
- Các chuyên gia đặc biệt (specialized professionals).

Phân công trách nhiệm trong đội (team assignments)

Hai trong số các thành viên sẽ là:

- một người dẫn chương trình
- một người viết tài liệu trong cuộc thảo luận.

- Chuẩn bị cho phiên peer review

- **Leader:**

- Xác định những đoạn trong tài liệu thiết kế sẽ được review
- Lựa chọn thành viên nhóm
- Lập lịch cho những phiên review
- Đưa tài liệu cho các thành viên trong đội trước phiên review

- **Đội peer review:** *yêu cầu của inspection khá tỉ mỉ, còn walkthrough chỉ yêu cầu đơn giản.*

- Inspection: Đọc & liệt kê chú thích của họ
- Walkth-rough : Đọc các đoạn sẽ được review

- **Phiên peer review**

- Inspection

- Presenter đọc một đoạn tài liệu và thêm vào nếu cần thiết.
- Những người liên quan hoặc đưa ra chú thích, hoặc phản ứng với những lời chú thích trong tài liệu.

- Walkthrough

- Bắt đầu bằng sự trình bày ngắn của author (thường ko phải là presenter) hoặc tổng quan về dự án và những đoạn thiết kế sẽ được review.

Scribe ghi lại vị trí, mô tả, kiểu, đặc điểm (sai sót, những phần thiếu, những phần thêm vào) của mỗi lỗi được chấp nhận.

Quy tắc thời gian: phiên không nên vượt quá 2 giờ, hoặc lập lịch không nhiều hơn 2 ngày.

Tài liệu sau mỗi phiên review:

- Báo cáo những phát hiện trong phiên inspection
- Báo cáo tóm tắt của phiên inspection

- **Các hoạt động sau peer review (post-peer review activities)**

Inspection:

- Nhắc nhở, sửa chữa hiệu quả, làm lại tất cả các lỗi
- Chuyển giao các bản báo cáo inspection tới CAB để phân tích.

- Hiệu quả của peer review (the efficiency of peer reviews)

Một vài độ đo phổ biến ước lượng hiệu suất của peer review:

- Số giờ trung bình trên một lỗi.
- Mật độ phát hiện thiếu sót (Số thiếu sót trung bình trên một trang tài liệu thiết kế).
- Hiệu năng peer review bên trong.

- Peer review coverage

Là tỉ lệ nhỏ của tài liệu và toàn bộ code đã từng trải qua peer review

7.2.4. Các ý kiến của chuyên gia

Ý kiến của chuyên gia rất hữu ích trong những trường hợp sau:

- Thiếu sự hiểu biết đầy đủ về lĩnh vực nào đó.
- Tạm thời thiếu những người chuyên nghiệp để tham gia vào đội xem xét lại

Các thành viên chuyên nghiệp cao cấp trong tổ chức không thống nhất được với nhau.

- Trong các tổ chức nhỏ số lượng ứng viên phù hợp cho đội xem xét lại là không đủ.

Để rõ hơn những đặc điểm của các phương pháp rà soát có thể tham khảo bảng so sánh giữa ba phương pháp rà soát sau đây:

Thuộc tính	Formal design reviews	Inspections	Walkthroughs
Mục đích chính trực tiếp	<ul style="list-style-type: none">- Phát hiện lỗi- Xác định rủi ro mới- Phê chuẩn tài liệu thiết kế	<ul style="list-style-type: none">- Phát hiện lỗi- Xác định sự sai lệch so với tiêu chuẩn	Phát hiện lỗi
Mục đích chính gián tiếp	Trao đổi kiến thức	<ul style="list-style-type: none">- Trao đổi kiến thức- Hỗ trợ hoạt động sửa chữa	Trao đổi kiến thức
Lãnh đạo việc xem xét lại	Người đứng đầu kỹ nghệ phần mềm hoặc là thành viên cao cấp	Người điều hành chỉ đạo (ngang hàng)	Người tổ chức

Người tham gia	Các thành viên cấp cao và đại diện khách hàng	Những người ngang hàng	Những người ngang hàng
Sự tham gia của trưởng dự án	Có	Có	Có, thông thường là khi bắt đầu xem xét lại
Thành viên chuyên gia trong đội	<ul style="list-style-type: none"> - Người thiết kế - Lập trình viên - Người kiểm thử 	<ul style="list-style-type: none"> - Người giám sát tiêu chuẩn - Chuyên gia bảo trì - Đại diện người dùng 	

Quá trình xem xét lại			
Họp tổng quan	Không	Có	Có
Sự chuẩn bị của những người tham gia	Có – chuẩn bị kỹ lưỡng	Có – chuẩn bị kỹ lưỡng	Có – chuẩn bị tóm lược
Phiên xem xét lại	Có	Có	Có
Bám sát việc sửa chữa	Có	Có	Có
Cơ sở hạ tầng			
Đào tạo chính thức cho người tham gia	Không	Có	Không
Sử dụng checklist	Không	Có	Không
Lỗi liên quan để thu thập dữ liệu	Không yêu cầu hình thức	Yêu cầu hình thức	Không yêu cầu hình thức
Tài liệu hóa việc xem xét lại	Báo cáo xem xét lại thiết kế hình thức	- Báo cáo đánh giá phiên thanh tra - Báo cáo tổng kết phiên thanh tra	Báo cáo đánh giá phiên walkthrough

BÀI 8. CÁC YẾU TỐ KHÁC TRONG ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

8.1. Đảm bảo chất lượng của các thành phần bảo trì phần mềm

8.1.1. Giới thiệu

Giai đoạn chính của vòng đời phần mềm là giai đoạn hoạt động, thường kéo dài từ 5 đến 10 năm, mặc dù cũng có những trường hợp hoạt động kéo dài đến 15 năm và thậm chí là hơn thế nữa cũng không phải là hiếm. Vậy điều gì đã khiến cho gói phần mềm có thể đạt được “tuổi thọ cao” với nhóm người dùng mà đã hài lòng nó, trong khi gói phần mềm khác cũng phục vụ đối tượng đó lại “bị hủy sớm”? Yếu tố chính chịu trách nhiệm cho dịch vụ dài hay thành công là bảo trì chất lượng. Việc bảo trì phần mềm quan trọng như thế nào có thể được giả định thông qua việc chú ý đến chủ đề được đưa ra trong chuẩn ISO 9000 – 3 (xem ISO (1977), phiên bản 4.19 và ISO/IEC (2001), phiên bản 7.5, IEEE (1998) và Oskarsson và Glass (1996).

Dưới đây là 3 thành phần của dịch vụ bảo trì và nó được xem như là bản chất của sự thành công:

- Bảo trì sửa lỗi: những dịch vụ hỗ trợ người dùng và sửa lỗi phần mềm.
- Bảo trì thích ứng: làm cho các gói phần mềm thích ứng với những yêu cầu mới của khách hàng và điều kiện môi trường thay đổi.
- Bảo trì cải thiện chất lượng: kết hợp (1) bảo trì hoàn thiện những chức năng mới được thêm vào phần mềm cũng như nâng cao hiệu suất, cùng với (2) bảo trì phòng ngừa – những hoạt động cải thiện độ tin cậy và cơ sở hạ tầng hệ thống cho dễ dàng làm cho việc bảo trì trong tương lai hiệu quả hơn.

Những dịch vụ hỗ trợ người dùng (“những trung tâm hỗ trợ người dùng”) trong bảo trì sửa lỗi có thể cần phải rõ ràng (clarification). Những dịch vụ hỗ trợ giải quyết tất cả những khó khăn xuất hiện khi sử dụng hệ thống phần mềm của người dùng; những dịch vụ sửa lỗi phần mềm thường được tích hợp trong dịch vụ này. Những khó khăn của người dùng có thể được gây ra bởi nguyên nhân sau:

- Lỗi code (thường được dùng với thuật ngữ “thất bại phần mềm”-software failure”).
- Lỗi viết tài liệu thủ công, giúp những màn ảnh hoặc các hình thức tài liệu hướng dẫn được chuẩn bị sẵn sàng cho người dùng. Trong trường hợp này, dịch vụ hỗ trợ có thể cung cấp cho người dùng với những chỉ dẫn đúng (mặc dù không có sự chính xác trong chính tài liệu phần mềm được thực hiện).
- Không đầy đủ, mơ hồ hoặc tài liệu không chính xác.
- Thiếu kiến thức về hệ thống phần mềm của người dùng hoặc thất bại của họ khi sử dụng tài liệu được cung cấp. Trong những trường hợp này không có xét đến thất bại hệ thống phần mềm. Ba nguyên nhân đầu tiên ở trên được xem như là những thất bại của hệ thống phần mềm. Thêm vào đó, sự tích hợp của những dịch vụ hỗ trợ người dùng và những dịch vụ hiệu chỉnh phần mềm nói chung là đã được hoàn thành trong sự kết hợp gần gũi, với nhiều thông tin chia sẻ. Những thành phần khác của dịch vụ bảo trì – cải thiện chất lượng và bảo trì thích ứng – khuynh hướng không được bắt đầu bởi những dịch vụ hỗ trợ người dùng. Trong hầu hết các trường hợp, những công việc cải thiện chức năng và khả năng thích ứng sẽ trình bày những đặc tính của một dự án nhỏ hoặc lớn, điều đó phụ thuộc vào những mong muốn của khách hàng. Trong trường hợp này, những công việc trên có thể được thực thi như là một phần của quá trình phát triển phần mềm. Khi xem xét ở trên, việc bao gồm những dịch vụ hỗ trợ người dùng trong những hoạt động bảo trì sửa đổi là hợp lý.

Nói chung, chúng ta có thể nói rằng việc bảo trì sửa lỗi bảo đảm những người dùng hiện thời có thể thao tác hệ thống như đã được vạch rõ, bảo trì thích ứng có thể mở rộng cho những đối tượng người dùng, và bảo trì cải thiện chất lượng đưa ra giai đoạn dịch vụ của gói.

Như đã đề cập trong mục trước, việc kết hợp 3 thành phần bảo trì phần mềm chiếm hơn 60% tổng số tài nguyên thiết kế và lập trình dành cho hệ thống phần mềm trong suốt vòng đời của nó (Pressman, 2000). Những ước lượng khác chia sẻ

những tài nguyên bảo trì kéo dài từ trên 50% (Lientz và Swanson, 1980) đến 65 – 75 % tổng số những tài nguyên phát triển dự án.

Mục tiêu hoạt động QA bảo trì phần mềm:

- (1) Đảm bảo, với mức độ tin cậy được chấp nhận, rằng những hoạt động bảo trì phù hợp với những yêu cầu kỹ thuật chức năng.
- (2) Đảm bảo, với mức độ tin cậy được chấp nhận, rằng những hoạt động bảo trì phù hợp với những yêu cầu quản lý lập lịch và ngân sách.
- (3) Những hoạt động khởi đầu và quản lý nhằm cải thiện và tăng hiệu quả cho bảo trì phần mềm và những hoạt động SQA. Điều này liên quan đến việc cải thiện cái nhìn toàn cảnh để đạt được những yêu cầu về chức năng và quản lý trong khi giá thành giảm.

8.1.2. Cơ sở cho chất lượng bảo trì cao

Nhiều người cho rằng chất lượng của gói phần mềm được bảo trì có lẽ là cơ sở quan trọng nhất nằm dưới chất lượng của những dịch vụ bảo trì. Nhưng cũng có những nhà phê bình khác cho rằng đó là chính sách bảo trì. Dưới đây là những thảo luận về chủ đề này.

8.1.2.1. Cơ sở 1: Chất lượng gói phần mềm

Chất lượng của gói phần mềm được bảo trì rõ ràng bắt nguồn từ sự thành thạo và những nỗ lực của nhóm phát triển cũng như những hoạt động SQA được thực hiện xuyên suốt quá trình phát triển. Nếu chất lượng của gói phần mềm là nghèo nàn thì cũng dẫn đến việc bảo trì nghèo nàn hoặc không có hiệu quả. Khi mà hiểu sâu sắc cơ sở này, chúng ta sẽ lựa chọn việc nhấn mạnh 7 trong 11 nhân tố đảm bảo chất lượng ban đầu mà có tác động trực tiếp trong bảo trì phần mềm. Đặc biệt là, chúng

ta sẽ thảo luận về 2 trong 5 nhân tố thao tác sản phẩm, 3 nhân tố xem xét lại sản phẩm và 2 trong 3 nhân tố chuyển giao sản phẩm.

Hai nhân tố thao tác sản phẩm:

- **Sự chính xác** (Correctness) – bao gồm:
 - Sự chính xác của đầu ra: Việc hoàn thành những đầu ra được chỉ rõ (nói cách khác là không có đầu ra mà đã được chỉ ra trước đó bị thiếu), sự đúng đắn của những đầu ra (những đầu ra của hệ thống được xử lý một cách chính xác), đầu ra hợp thời (thông tin được xử lý luôn được cập nhật như đã chỉ rõ) và đầu ra có tính sẵn sàng (Những lần tương tác không vượt những giá trị cực đại xác định, đặc biệt là những ứng dụng trực tuyến và thời gian thực).
 - Sự chính xác của tài liệu hướng dẫn: Chất lượng của tài liệu hướng dẫn: tính đầy đủ, sự đúng đắn, kiểu và cấu trúc tài liệu. Những hình thức tài liệu bao gồm bản sao trên giấy và những file máy tính – được in ấn thông thường cũng như những file “trợ giúp” điện tử - trong khi phạm vi của nó chứa đựng những tài liệu cài đặt, tài liệu hướng dẫn người dùng và tài liệu của người lập trình viên.
 - Viết mã đúng quy cách: phù hợp với những hướng dẫn mã, đặc biệt là những hạn chế và sự phức tạp trong mã biến đổi cũng như xác định kiểu viết mã chuẩn.
- **Độ tin cậy:** Tần số của những thất bại của hệ thống cũng như những lần phục hồi.

Ba nhân tố xem xét lại sản phẩm:

- **Sự bảo trì:** Những yêu cầu này được thực hiện đầu tiên và tốt nhất bằng cách làm theo cấu trúc phần mềm và những yêu cầu kiểu cũng như là theo cài đặt những yêu cầu trong tài liệu của lập trình viên.
- **Tính linh động:** Đạt được thông qua việc thiết kế, lập kế hoạch thích hợp và những đặc tính cung cấp không gian ứng dụng lớn hơn nhu cầu của người dùng hiện tại.

- **Có thể test được:** Có thể test bao gồm tính sẵn sàng của những chuẩn đoán hệ thống được đưa ra bởi người dùng cũng như những chuẩn đoán thất bại được áp dụng bởi trung tâm hỗ trợ hoặc những nhân viên bảo trì tại vị trí người dùng.

Cuối cùng là 2 nhân tố chuyển phát sản phẩm:

(1) ***Tính khả chuyển:*** Các ứng dụng tiềm tàng của phần mềm trong môi trường phần cứng và hệ điều hành khác nhau, nó gồm những hoạt động mà cho phép thực hiện những ứng dụng đó.

(2) ***Thao tác giữa các phần:*** Khả năng của các gói giao tiếp với các gói hoặc thiết bị tính toán khác. Bằng việc cung cấp khả năng đáp ứng những chuẩn về giao diện và áp dụng những giao diện đó một cách phù hợp kết hợp với những hướng dẫn của thiết bị sản xuất và phần mềm, thao tác giữa các phần có thể đạt được ở mức cao.

8.2.1.2. Cơ sở 2: Chính sách bảo trì

Những thành phần của chính sách bảo trì chính ảnh hưởng đến sự thành công của việc bảo trì phần mềm là sự phát triển các phiên bản và thay đổi chính sách để được áp dụng trong suốt vòng đời phần mềm.

Chính sách phát triển phần mềm

Chính sách này có quan hệ mật thiết với câu hỏi có bao nhiêu phiên bản phần mềm được vận hành cùng một lúc. Rõ ràng rằng, đây không phải là vấn đề dịch vụ của một tổ chức về phần mềm được đặt hàng mà vấn đề chính ở đây là số lượng các phiên bản hay chính là giá thành (COS s) của những gói phần mềm được lên kế hoạch để phục vụ cho nhiều khách hàng khác nhau. Chính sách phát triển phiên bản sau này có thể được thực hiện dưới dạng “tuần tự” hoặc “cây”. Khi áp dụng chính sách tuần tự, chỉ một phiên bản được tạo ra sẵn cho toàn bộ khách hàng. Phiên bản này gồm số lượng lớn những ứng dụng mà biểu lộ sự dư thừa cao, một thuộc tính mà cho phép phần mềm phục vụ tất cả những mong muốn của khách hàng. Phần mềm phải được xem lại một cách định kỳ nhưng một khi một phiên bản mới được

hoàn thành, nó sẽ thay thế phiên bản đang được sử dụng hiện thời bởi toàn bộ đối tượng người dùng.

Khi áp dụng chính sách phiên bản “cây”, nhóm bảo trì phần mềm hỗ trợ những nỗ lực marketing (quảng cáo/tiếp thị) bằng việc phát triển một phiên bản chuyên dụng, nhằm tới những nhóm khách hàng hay khách hàng chính một khi nó được yêu cầu. Một phiên bản mới được bắt đầu bằng việc thêm những ứng dụng đặc biệt hoặc bỏ qua những ứng dụng, phụ thuộc vào những gì liên quan đến nhu cầu của khách hàng. Những phiên bản này thay đổi theo sự phức tạp và mức của ứng dụng – những ứng dụng hướng công nghiệp được hướng tới, vv...Nếu chính sách này được chấp nhận, gói phần mềm có thể tiến triển thành một gói đa phiên bản sau vài năm của dịch vụ, có nghĩa rằng giống như một cái cây, với vài nhánh chính và nhiều nhánh phụ, từng phần nhánh đại diện cho một

phiên bản đã được duyệt lại một cách chuyên dụng. Trái với phiên bản phần mềm dạng tuần tự, việc bảo trì và quản lý của phiên bản phần mềm dạng cây phức tạp và tốn thời gian hơn nhiều. Xem xét sự thiếu sót này, những tổ chức phát triển phần mềm cố gắng áp dụng chính sách phiên bản dạng cây một cách giới hạn, chỉ cho phép một số lượng nhỏ những phiên bản phần mềm được phát triển.

Chính sách thay đổi

Chính sách thay đổi đề cập đến phương thức để kiểm tra mỗi yêu cầu thay đổi và tiêu chuẩn sử dụng cho những phương pháp đó. Một chính sách là rõ ràng nếu nó được thực thi bởi CCB (The Change Control Board _ Ban điều khiển thay đổi) hoặc những người được ủy quyền để phê duyệt những thay đổi đó. Chính sách cân bằng đòi hỏi có một cuộc kiểm tra tỉ mỉ những yêu cầu thay đổi, điều này là rất phù hợp vì như vậy nó cho phép nhân viên tập trung vào những thay đổi quan trọng và có ích nhờ thế chúng ta mới có thể thực thi công việc trong thời gian hợp lý và theo những chuẩn chất lượng mong muốn.

8.1.3. Các thành phần chất lượng phần mềm tiền bảo trì

Giống như những thành phần SQA tiền dự án, những hoạt động SQA trước bảo trì cần được hoàn thành để khởi tạo các dịch vụ bảo trì cần thiết cũng rất quan trọng. Nó bao gồm thứ tự các bước sau:

- Xem xét lại hợp đồng bảo trì
- Xây dựng kế hoạch bảo trì.

8.1.3.1. Xem xét lại hợp đồng bảo trì

Khi xem xét hợp đồng bảo trì, quan điểm mở rộng nên được khái quát. Đặc biệt, những quyết định được yêu cầu về các loại hình dịch vụ cần được ký kết trong hợp đồng. Những quyết định này phụ thuộc vào các loại hình dịch vụ khách hàng: dành cho khách hàng mà có gói custom-made được phát triển; dành cho khách hàng mà mua gói phần mềm COST và những khách hàng bên trong. Vì vậy, trước khi bắt đầu cung cấp những dịch vụ bảo trì phần mềm tới từng nhóm khách hàng trên thì hợp đồng bảo trì tương ứng phải được hoàn thành để nó đánh giá tổng số những trách nhiệm bảo trì theo những điều kiện liên quan.

Giả định sự thực thi

Những dịch vụ bảo trì tới khách hàng bên trong thường không có hợp đồng. Trong một số trường hợp cụ thể, một vài dịch vụ được cung cấp trong quá trình thực hiện mà không có sự xác định rõ ràng cho những dịch vụ đó. Trong những trường hợp như vậy thì sự không hài lòng dễ xảy ra trong cả 2 phía: những khách hàng bên trong cảm thấy rằng họ mong muốn được hỏi ý kiến một cách có thiện ý thay vì nhận những dịch vụ một cách đều đặn mà họ không mong chờ, bên cạnh đó thì nhóm phát triển lại luôn yêu cầu thực thi việc bảo trì như là một việc bắt buộc một khi họ đã làm việc trong một dự án khác.

Để tránh tình trạng căng thẳng, một “hợp đồng dịch vụ bên trong” nên được viết. Trong tài liệu này, những dịch vụ này được cung cấp bởi nhóm bảo trì bên trong tới những khách hàng bên trong được xác định một cách rõ ràng. Với việc bỏ qua hầu hết những hiểu sai liên quan đến những dịch vụ ảo, hợp đồng này có thể đáp ứng một cách cơ bản những bảo trì thỏa đáng tới những khách hàng bên trong.

Những hoạt động xem xét lại hợp đồng bảo trì bao gồm việc nhìn lại những bản nháp đề xuất cũng như là nhìn lại những bản hợp đồng nháp. Thực tế, việc nhìn lại mục tiêu và sự thực thi bản hợp đồng bảo trì là đi theo từng dòng trong việc xem xét lại bản hợp đồng tiền dự án.

Dưới đây là những mục tiêu chính trong việc xem xét lại hợp đồng bảo trì phần mềm:

- **Sự rõ ràng trong yêu cầu của khách hàng**

Những vấn đề sau đây đặc biệt được quan tâm:

- Loại dịch vụ bảo trì sửa đổi được yêu cầu: Danh sách những dịch vụ từ xa và những dịch vụ tại chỗ được cung cấp, giờ phục vụ, thời gian phản hồi...
- Số lượng người sử dụng và kiểu ứng dụng được dùng.

- Những người dùng địa phương, ở khoảng cách xa (hoặc qua đại dương) và các kiểu dịch vụ được cài đặt tại mỗi nơi đó.

- Cung cấp bảo trì cải thiện chức năng, khả năng thích ứng và thủ tục cho những yêu cầu của dịch vụ cũng như đề xuất và phê duyệt việc thực thi cho những dịch vụ này

- **Xem xét lại những phương pháp tiếp cận khác về các điều khoản trong văn bản bảo trì Xem xét những suy xét cụ thể dưới đây để đưa ra lựa chọn phù hợp:**

- Những hợp đồng con cho những site (địa điểm) hoặc kiểu dịch vụ

- Việc thực thi một vài dịch vụ thông qua khách hàng cùng với sự hỗ trợ từ nhóm bảo trì của nhà cung cấp.

- **Nhìn lại sự ước lượng về tài nguyên bảo trì được yêu cầu**

- Đầu tiên, những ước lượng nên được kiểm tra lại trên cơ sở của những dịch vụ được yêu cầu, sắp xếp theo đề xuất của nhóm. Sau đó, dựa vào năng lực của công ty để xem có đáp ứng được những khía cạnh chuyên môn cũng như là tính sẵn sàng của nhóm bảo trì đã phân tích hay không.

- **Xem xét lại những dịch vụ bảo trì được cung cấp bởi những hợp đồng con và/hoặc khách hàng.**

- Việc nhìn lại này đề cập tới việc định danh lại những dịch vụ được cung cấp bởi mỗi người tham gia, số tiền trả cho những hợp đồng con, đảm bảo chất lượng và những thủ tục tiếp theo được đáp ứng.

- **Xem xét lại những ước lượng về giá thành bảo trì**

- Những ước lượng này nên được nhìn lại dựa trên cơ sở của những tài nguyên được yêu cầu.

8.1.3.2. Lập kế hoạch bảo trì

Lập kế hoạch bảo trì phải được chuẩn bị cho tất cả các khách hàng trong và ngoài. Những kế hoạch này nên cung cấp framework trong việc tổ chức những điều khoản bảo trì.

Lập kế hoạch đó bao gồm những bước sau:

- **Danh sách những dịch vụ bảo trì được ký kết**
- Khách hàng bên trong và ngoài, số lượng người sử dụng, sự định vị vị trí cho mỗi site khách hàng.
- Đặc tính của những dịch vụ bảo trì sửa đổi (xa hoặc tại chỗ).
- Trách nhiệm của việc bảo trì cải thiện chức năng và khả năng thích ứng phục vụ điều khoản của mỗi khách hàng.

- **Mô tả tổ chức của nhóm bảo trì**

Kế hoạch tổ chức nhóm bảo trì tập trung vào những yêu cầu về nhân sự. Điều đó có thể được xem xét cẩn thận theo những tiêu chuẩn sau:

- Số lượng thành viên nhóm được yêu cầu.
- Chất lượng thành viên nhóm được yêu cầu theo công việc bảo trì, bao gồm những người quen với gói phần mềm cần được bảo trì.
- Cấu trúc tổ chức của những nhóm bảo trì, bao gồm tên của những người lãnh đạo nhóm.
- Định nghĩa các công việc (trách nhiệm của khách hàng, kiểu ứng dụng, ...) cho mỗi nhóm.
- Đào tạo khi cần thiết

- **Danh sách điều kiện thuận lợi cho bảo trì**

Những điều kiện thuận lợi cho bảo trì – cơ sở hạ tầng mà có thể cung cấp dịch vụ bao gồm:

- Trung tâm hỗ trợ bảo trì với thiết bị phần cứng và phần mềm đã được cài đặt để cung cấp những dịch vụ hỗ trợ cho người dùng và sửa đổi phần mềm.

- Tài liệu chính thức chứa tập các tài liệu đã hoàn thành:

(1) Tài liệu phần mềm, bao gồm tài liệu phát triển.

(2) Những hợp đồng dịch vụ

(3) Cấu hình phần mềm cho mỗi khách hàng và phiên bản của những gói phần mềm được cài đặt tại mỗi địa điểm và được cung cấp bởi nhà quản lý cấu hình.

(4) Những thông tin bảo trì lịch sử được ghi lại cho mỗi người dùng và khách hàng.

- Danh sách những rủi ro dịch vụ bảo trì được xác định

Rủi ro dịch vụ bảo trì liên quan đến hoàn cảnh mà thất bại xảy ra để cung cấp bảo trì tương ứng được dự đoán trước. Những rủi ro này bao gồm:

- Thiếu nhân viên trong suốt những dịch vụ bảo trì của tổ chức, trong trung tâm hỗ trợ bảo trì cụ thể hoặc trong những ứng dụng cụ thể.
- Sự hiểu biết và trình độ chuyên môn không tương xứng với từng phần của các gói phần mềm liên quan để thực thi những dịch vụ hỗ trợ người dùng và/hoặc những công việc bảo trì sửa đổi.
- Những thành viên nhóm không đủ khả năng để thực hiện những công việc cải thiện chức năng cũng như khả năng thích ứng.

- **Danh sách những thủ tục và điều khiển quản lý phần mềm được yêu cầu**
Hầu hết những thủ tục được yêu cầu để cập đến những tiến trình được thực thi bởi những nhóm bảo trì sửa đổi và trung tâm hỗ trợ người dùng. Những thủ tục này giải quyết:

- Vận hành những ứng dụng của khách hàng.
- Xử lý những bản ghi lỗi của phần mềm
- Ghi chép định kỳ và liên tục những dịch vụ hỗ trợ người dùng
- Ghi chép định kỳ và liên tục những dịch vụ bảo trì sửa đổi.
- Đào tạo và cấp giấy chứng thực cho những thành viên bảo trì.

- **Ngân sách bảo trì phần mềm**

Những ước lượng được sử dụng trong ngân sách bảo trì sửa đổi dựa trên kế hoạch tổ chức nhân sự, trình độ chuyên môn được yêu cầu và sự đầu tư vốn cần để tạo ra

những điều kiện, những nhu cầu của nhóm và tác vụ khác. Họ phải được chuẩn bị mỗi khi nhân sự, trình độ chuyên môn và các thủ tục được xác định. Những ước lượng cho việc bảo trì cải thiện chất lượng và khả năng thích ứng được chuẩn bị theo sự mong đợi, và theo sự thay đổi cũng như là cải thiện khi chúng được thực hiện.

8.1.4. Các công cụ đảm bảo chất lượng bảo trì phần mềm

Sự đa dạng của các công cụ đảm bảo chất lượng phần mềm được sử dụng trong các chu kỳ thực hiện của vòng đời phần mềm. Bản chất riêng biệt của mỗi thành phần bảo trì phần mềm: bảo trì sửa đổi, bảo trì thích ứng và bảo trì cải thiện chức năng, yêu cầu các tập công cụ SQA khác nhau cần phải sử dụng cho mỗi loại. Hơn nữa, chu kỳ thực hiện của phần mềm điển hình thì mở rộng việc sử dụng các công cụ SQA cơ sở và công cụ giám sát việc quản lý.

Ý tưởng về việc đưa SQA vào pha bảo trì được Perry đưa ra (1995). Trong 1 nghiên cứu ông thực hiện vào tháng 11 năm 1994, các bên tham gia được thông báo rằng: dựa trên kinh nghiệm của họ thì 31% kế hoạch bảo trì của họ đã được thực hiện để đảm bảo chất lượng.

Phân tiếp theo sẽ nói về các nội dung sau:

- 1) Công cụ SQA cho bảo trì sửa lỗi
- 2) Công cụ SQA cho bảo trì cải thiện chức năng
- 3) Công cụ SQA cơ sở cho bảo trì phần mềm
- 4) Công cụ SQA cho giám sát quản lý bảo trì phần mềm.

8.1.4.1. Công cụ SQA cho bảo trì sửa lỗi

Các hoạt động bảo trì sửa lỗi đưa ra đầu tiên: (a) các dịch vụ hỗ trợ người sử dụng và (b) sửa chữa phần mềm (sửa lỗi). Các dịch vụ hỗ trợ người sử dụng giải quyết các trường hợp lỗi về mã phần mềm và lỗi tài liệu, tài liệu không hoàn thành hoặc mập mờ. Chúng có thể chứa những câu lệnh mà người sử dụng không đủ kiến thức về phần mềm hoặc thất bại trong việc sử dụng tài liệu có sẵn. Các dịch vụ sửa

chữa phần mềm – gỡ lỗi và sửa tài liệu được sử dụng trong trường hợp lỗi phần mềm (failure), và được cung cấp trong suốt chu kỳ khởi tạo của việc thực hiện (mặc dù sự nỗ lực này được đầu tư vào việc testing) và tiếp tục được yêu cầu mặc dù không thường xuyên lắm. Dù hai kiểu dịch vụ khác nhau nhưng tập các công cụ đảm bảo chất lượng đều tập trung vào cùng một mục đích là đảm bảo chất lượng dịch vụ. Trong nhiều trường hợp cùng một đội có thể thực hiện cả hai kiểu bảo trì sửa lỗi.

Thêm vào các công cụ SQA giám sát việc quản lý và cơ sở hạ tầng (được thảo luận ở phần sau của chương) thì hầu hết các công việc sửa lỗi yêu cầu sử dụng các công cụ SQA cho vòng đời nhỏ, mainly mini-testing (test những phần nhỏ quan trọng). Thủ tục mini-testing được yêu cầu cho việc xử lý các tác vụ về lỗi đường dẫn (repair patch) (small - scale), nó được đặc trưng bởi số lượng nhỏ dòng lệnh thay đổi cùng nhau để hoàn thành việc sửa lỗi nhanh chóng. Những vấn đề liên quan đến sửa chữa trì hoãn giống như một abridged-mini-form của testing thường được sử dụng. Tuy nhiên, sử dụng những công cụ mini testing vẫn cần được thực hiện để tránh những trường hợp không thể test được.

Để đảm bảo chất lượng của mini testing, những chỉ dẫn sau nên được giữ vững:

- Testing được thực hiện bởi những tester chất lượng, không phải bởi người lập trình thực hiện sửa chữa
- Tài liệu thủ tục test (dài nhất là 2-3 trang) nên được chuẩn bị. Tài liệu bao gồm một bản mô tả về những ảnh hưởng biết trước được của việc sửa chữa, phạm vi sửa lỗi và một danh sách các test case được thực hiện. Tài liệu thủ tục trước test, tương tự như tài liệu thủ tục testing cũng nên được chuẩn bị để thực hiện test việc sửa lỗi được phát hiện trong test trước.
- Một bản ghi test đầy đủ viết về các lỗi phát hiện được dưới dạng tài liệu trong mỗi giai đoạn test và re-test nên được hoàn thành.

- Nhóm test chính xem lại tài liệu test về phạm vi sửa lỗi, tính đúng đắn của các test case và các kết quả test.
- Việc sửa chữa được cho là “simple và trivial”, đặc biệt đối với việc thực hiện tại site của khách hàng, mini-testing có thể được tránh đi.

Các dịch vụ bảo trì hợp đồng con, đặc biệt là các dịch vụ hỗ trợ người sử dụng đã trở nên khá phổ biến bất cứ khi nào có quá nhiều vấn đề hoặc không kinh tế cho nhà thầu bảo trì cung cấp các dịch vụ đó một cách trực tiếp. Công cụ chính để đảm bảo chất lượng của các dịch vụ bảo trì của nhà thầu phụ và mở đường cho các quan hệ suôn sẻ là hợp đồng contractor-subcontractor. Công cụ SQA được tích hợp vào hợp đồng nhằm mục đích:

- (i) Các thủ tục xử lý phân loại cụ thể các cuộc gọi bảo trì.
- (ii) Tài liệu đầy đủ về các thủ tục dịch vụ.
- (iii) Có sẵn các bản ghi được viết tài liệu kiểm thử chuyên nghiệp của thành viên đội bảo trì của nhà thầu phụ, cho nhà thầu xem xét lại.
- (iv) Cấp giấy phép cho nhà thầu thực hiện xem xét lại theo chu kỳ các dịch vụ bảo trì cũng như sự thỏa mãn của khách hàng.
- (v) Các điều kiện liên quan đến chất lượng thì yêu cầu chặt chẽ về hình phạt và sự kết thúc của ràng buộc hợp đồng con trong trường hợp cực đoan.

Một khi bảo trì sẵn sàng thực hiện, nhà thầu nên quản lý đều đặn việc xem lại dịch vụ bảo trì và sự thỏa mãn của khách hàng.

8.1.4.2. Các công cụ SQA cho bảo trì cải thiện chức năng

Do sự giống nhau của việc bảo trì cải thiện chức năng và dự án phát triển phần mềm, các công cụ vòng đời dự án được áp dụng cho bảo trì cải thiện chức năng. Những công cụ này cũng được thực thi cho bảo trì thích ứng phạm vi rộng (large scale adaptive maintenance). Các công cụ SQA mở rộng được thực thi cho bảo trì cải thiện chức năng là các công cụ điều khiển quản lý và cơ sở hạ tầng, được thảo luận thêm ở phần sau.

8.1.4.3. Các thành phần cơ sở hạ tầng SQA cho bảo trì phần mềm

Các công cụ cơ sở đảm bảo chất lượng phần mềm là các thành phần quan trọng của bảo trì phần mềm. Sự cân đối của mảng các công cụ SQA cơ sở là bản chất và được thực thi trong suốt vòng đời của hệ thống phần mềm. Hơn nữa, sự giống nhau của các tiến trình cải thiện chức năng phần mềm và phát triển phần mềm cho phép các tiến trình dùng chung các công cụ SQA cơ sở và chỉ có thay đổi nhỏ. Các công cụ cơ sở chuyên dụng được yêu cầu cho các hoạt động bảo trì sửa lỗi do các đặc trưng đặc biệt của các hoạt động này. Các hoạt động bảo trì thích ứng được phục vụ bởi các công cụ SQA cơ sở, theo các đặc trưng của nó. Các công cụ được áp dụng thường xuyên nhất là các công cụ SQA cải thiện chức năng, được theo bởi các công cụ SQA bảo trì sửa lỗi.

Thực tế, sự đóng góp của các công cụ SQA cơ sở cho việc bảo trì không bắt đầu tại thời điểm ban đầu của các tiến trình bảo trì. Rõ ràng là việc áp dụng thích hợp các công cụ SQA cơ sở bởi đội phát triển phần mềm đóng góp căn bản vào hiệu quả và hiệu năng các hoạt động của đội bảo trì. Nói cách khác, những công cụ này góp phần vào việc bảo trì đảm bảo chất lượng theo 2 cách: đầu tiên, bằng cách hỗ trợ đội phát triển phần mềm khi tạo ra phần mềm chất lượng cao, và thứ hai, bằng cách hỗ trợ đội bảo trì đáp ứng việc bảo trì của sản phẩm phần mềm giống nhau.

Các công cụ SQA cơ sở chuyên dụng được yêu cầu cho các tiến trình bảo trì phần mềm, đặc biệt là bảo trì sửa lỗi, thể hiện các đặc trưng đặc biệt. Ở đây chúng ta tập trung vào các công cụ SQA cơ sở chuyên dụng của các lớp sau:

- Các thủ tục bảo trì và các chỉ dẫn làm việc
- Hỗ trợ các thiết bị chất lượng
- Đào tạo và cấp chứng chỉ cho đội bảo trì
- Các hoạt động ngăn ngừa và sửa lỗi
- Quản lý cấu hình
- Viết tài liệu và điều khiển bản ghi chất lượng

Các thủ tục bảo trì và chỉ dẫn làm việc

Hầu hết các thủ tục bảo trì và các chỉ dẫn làm việc chuyên dụng được áp dụng cho các hoạt động bảo trì sửa lỗi và hỗ trợ người sử dụng, ví dụ:

- Xử lý từ xa các yêu cầu cho dịch vụ trong trường hợp lỗi phần mềm
- Xử lý tại chỗ các yêu cầu của khách hàng cho dịch vụ trong trường hợp phần mềm lỗi.
- Dịch vụ hỗ trợ người sử dụng
- Điều khiển đảm bảo chất lượng cho các hoạt động sửa lỗi phần mềm và hỗ trợ người dùng
- Điều tra sự thỏa mãn khách hàng.
- Cấp chứng chỉ cho các thành viên đội bảo trì sửa lỗi và hỗ trợ người sử

Cc thiết bị hỗ trợ chất lượng

Bảo trì được mong đợi để phát triển các thiết bị chuyên dụng hỗ trợ các hoạt động sửa lỗi phần mềm và hỗ trợ người dùng: templates (khuôn mẫu), checklists (danh sách kiểm tra) và những cái tương tự thế. Các thiết bị có thể bao gồm:

- Danh sách kiểm tra các phần có thể gây ra lỗi – được áp dụng bởi nhà chuyên môn về bảo trì
- Các khuôn mẫu báo cáo lỗi phần mềm được giải quyết như thế nào, bao gồm sự phát hiện của các tiến trình sửa lỗi
- Danh sách kiểm tra cho việc chuẩn bị tài liệu thủ tục testing nhỏ.

Đào tạo và cấp chứng chỉ cho đội bảo trì

Đào tạo đội bảo trì giải quyết các công việc cải thiện chức năng không khác mấy so với việc đào tạo đội phát triển phần mềm khác. Tu y nhiên, đào tạo đặc biệt và cấp chứng chỉ mang tính cốt yếu cho đội bảo trì sửa lỗi.

Việc đào tạo các chuyên gia bảo trì sửa lỗi được thúc đẩy bởi yêu cầu hỗ trợ các dịch vụ cụ thể trong hợp đồng bảo trì. Do vậy, kế hoạch đào tạo nên cung cấp các giải pháp cho các yêu cầu nhân sự trong suốt các chu kỳ trọng tải cao nhất và các yêu cầu của tổ chức để thay thế nhân sự bị sa thải. Trong nhiều trường hợp, việc đào

tạo các nhân viên bảo trì dự trữ là không đủ, phải đào tạo thêm hệ thống riêng biệt. Nói cách khác, chương trình đào tạo nghiêm ngặt được yêu cầu để cho phép tổ chức xác định phạm vi với mức độ ràng buộc của các dịch vụ riêng biệt cho các chu kỳ trọng tải cao và trong trường hợp thay đổi nhân viên bảo trì hoặc bất kỳ lý do nào khác.

Yêu cầu cấp chứng chỉ cho các nhân viên sửa lỗi phần mềm và hỗ trợ người dùng là gốc rễ trong các đặc trưng của các dịch vụ này. Sự quan tâm đặc biệt nên dành cho việc cấp chứng chỉ cho các nhân viên sửa lỗi phần mềm, người mà thường xuyên thực hiện công việc của họ dưới áp lực về thời gian lớn, làm việc 1 mình, và trong nhiều trường hợp làm việc tại site của khách hàng, nơi mà sự hỗ trợ chuyên gia từ team leader hoặc những người khác bị giới hạn.

Các hoạt động ngăn ngừa và sửa lỗi

Pha hoạt động của vòng đời phần mềm tạo ra thông tin có giá trị cao: các bản ghi lỗi phần mềm, sửa chữa các lỗi đó cũng như bản ghi các yêu cầu hỗ trợ khách hàng có thể dẫn tới các hoạt động ngăn ngừa và sửa lỗi do đó góp phần cải thiện hệ thống phần mềm đã có và phần mềm mới.

Để các tiến trình hoạt động hiệu quả, cần có các tiến trình thích hợp cho việc biểu diễn thông tin thu thập được, xem lại và phân tích những phát hiện, và đưa ra những gợi ý cho việc cải thiện các tiến trình bảo trì và phát triển liên quan. Những hoạt động SQA này được chỉ dẫn và điều khiển bởi ủy ban nội bộ - CAB (Corrective Action Board), được sáng lập trong tổ chức phát triển phần mềm chính.

Những vấn đề điển hình ban xúc tiến cần xem lại gồm:

- 1) Những thay đổi về nội dung và tính thường xuyên của khách hàng yêu cầu về các dịch vụ hỗ trợ người dùng.
- 2) Tăng thời gian trung bình được đầu tư để tuân theo yêu cầu hỗ trợ của khách hàng.
- 3) Tăng thời gian trung bình được đầu tư để sửa chữa các lỗi phần mềm của khách hàng.

4) Tăng phần trăm các lỗi hiệu chỉnh phần mềm.

Quản lý cấu hình

Nhóm bảo trì đưa ra hầu hết tập các phụ thuộc vào quản lý cấu hình. Sự phụ thuộc này là kết quả của các mối quan hệ dựa trên kinh nghiệm với các gói phần mềm dịch vụ qua nhiều năm, trong suốt các phiên bản mới được thêm vào, phiên bản cũ bị thay thế và nhiều sự cài đặt mới và các thay đổi phần mềm được thực hiện.

Hai ứng dụng chung dựa vào quản lý cấu hình là: (1) sửa lỗi và (2) sự thay thế nhóm (group replacement) của các phiên bản phần mềm đang được sử dụng bằng các phiên bản mới, được khởi tạo bởi tổ chức bảo trì.

1. Sửa lỗi (Failure repair): Trong vấn đề về sửa lỗi phần mềm, việc hỗ trợ cập nhật và độ tin cậy là cần thiết theo dạng của:

- Thông tin quan tâm tới phiên bản hệ thống phần mềm được cài đặt tại site của khách hàng.
- Bản copy code hiện thời và tài liệu của nó.

-> Sự đóng góp vào chất lượng phần mềm được thể hiện là lỗi ít hơn trong các thử nghiệm hiệu chỉnh lỗi và giảm bớt được tài nguyên đầu tư vào sửa lỗi.

2. Group replacement (Thay thế nhóm): Thuật ngữ group trong SQA đề cập đến tất cả khách hàng có cùng phiên bản phần mềm được cài đặt tại site của họ. Do vậy, group replacement chỉ rõ rằng tất cả khách hàng sử dụng phiên bản sẽ được nhân phiên bản mới được phát triển hoặc cập nhật tại cùng 1 thời gian. Quản lý cấu hình hỗ trợ cho vấn đề thay thế nhóm dựa trên thông tin về các thành viên của nhóm khách hàng

- Đưa ra quyết định về tính thích hợp của việc thực hiện thay thế nhóm dựa trên quy mô của việc thay thế và kiểu hợp đồng đã ký với khách hàng.
- Lên kế hoạch thay thế nhóm, phân phối tài nguyên và xác định thời gian.

-> Sự đóng góp vào chất lượng phần mềm thể hiện ở việc thay thế phiên bản phần mềm hiện thời bởi những phiên bản đã được cải thiện mà giảm lỗi phần mềm, yêu

cầu ít sự hỗ trợ. Cải thiện chất lượng cũng góp phần bảo trì phần mềm hiệu quả, yêu cầu ít tài nguyên cho bảo trì sửa lỗi.

Tài liệu bảo trì và bản ghi chất lượng

Những yêu cầu cụ thể cho tài liệu và bản ghi chất lượng có quan hệ chặt chẽ nhất với các hoạt động sửa lỗi phần mềm và hỗ trợ người sử dụng. Tài liệu và bản ghi chất lượng

được chuẩn bị để:

- Cung cấp dữ liệu cần thiết cho các hoạt động ngăn ngừa và sửa lỗi.
- Hỗ trợ việc xử lý các yêu cầu hỗ trợ người dùng và các báo cáo lỗi khách hàng trong tương lai.
- Cung cấp bằng chứng để đáp ứng những yêu sách từ phía khách hàng trong tương

Những yêu cầu tài liệu được liệt kê trong các thủ tục bảo trì khác nhau nên đáp ứng với tất cả những yêu cầu tài liệu trên

8.1.4.4. Những công cụ SQA giám sát quản lý cho bảo trì phần mềm

Trong khi các công cụ SQA giám sát quản lý cụ thể được yêu cầu cho các hoạt động bảo trì sửa lỗi, thì sự giống nhau của các tiến trình phần mềm mô tả việc cải thiện chức năng và bảo trì thích ứng cũng như việc phát triển phần mềm cho phép những tiến trình này được sử dụng cùng một công cụ quản lý. Đặc biệt, các thành phần SQA quản lý có ý nghĩa giúp cải thiện việc điều khiển bảo trì bằng cách tạo ra những báo động sớm tín hiệu làm giảm chất lượng của các dịch vụ và tăng tỷ số thất bại dịch vụ. Phần còn lại của mục này sẽ đưa ra những vấn đề điều khiển quản lý chất lượng, những phần chính này đạt tới dịch vụ sửa lỗi phần mềm và hỗ trợ người dùng ở trên:

- ❖ Điều khiển hiệu năng cho các dịch vụ bảo trì sửa lỗi.
- ❖ Đo chất lượng cho bảo trì sửa lỗi.
- ❖ Chi phí chất lượng bảo trì phần mềm.

Điều khiển hiệu năng cho các dịch vụ bảo trì sửa lỗi

Điều khiển hiệu năng quản lý cho các dịch vụ bảo trì sửa lỗi khác với khi áp dụng các dịch vụ sửa lỗi phần mềm và các dịch vụ hỗ trợ người dùng. Các công cụ điều khiển quản lý mang lại, bên cạnh thông tin hiệu năng theo chu kỳ, còn là *những báo động sự chú ý quản lý*, như sau:

- Sửa lỗi phần mềm
 - Tăng việc sử dụng tài nguyên
 - Giảm tỉ lệ sửa chữa lỗi từ xa so với sửa chữa on-site của khách hàng.
 - Tăng tỷ lệ sửa chữa on-site ở những vùng cục bộ ở xa so với các dịch vụ hải ngoại
 - Tăng phần trăm thất bại gặp phải khi sửa chữa yêu cầu lập lịch
 - Tăng tỉ lệ sửa lỗi, và liệt kê các trường hợp “model” cụ thể của các tình huống lỗi cục đoạn.
 - Giảm bớt sự thỏa mãn khách hàng dựa trên các điều tra sự thỏa mãn của khách hàng.
- Hỗ trợ người sử dụng
 - Tăng tỉ lệ các yêu cầu dịch vụ cho các hệ thống phần mềm cụ thể, cho các kiểu dịch vụ, ...
 - Tăng tài nguyên được sử dụng cho các dịch vụ hỗ trợ người dùng
 - Tăng tỉ lệ thất bại của việc cung cấp các dịch vụ được yêu cầu
 - Tăng tỉ lệ lỗi, và trường hợp cụ thể của các thất bại đáng chú ý
 - Thông tin thỏa mãn khách hàng dựa trên các nghiên cứu sự thỏa mãn của khách hàng

Những điều khiển sửa chữa lỗi quản lý này (những cái được mong đợi để đưa ra những cảnh báo) được thực hiện qua báo cáo theo chu kỳ, qua buổi họp nhân viên đều đặn, qua việc xem xét các trung tâm hỗ trợ bảo trì cung cấp dịch vụ, qua phân tích báo cáo xử lý độ đo bảo trì phần mềm và giá thành chất lượng bảo trì. Thông tin được tích lũy hỗ trợ các quyết định quản lý quan tâm tới kế hoạch và việc thực hiện bảo trì sửa lỗi.

Đo chất lượng bảo trì phần mềm

Đo chất lượng bảo trì phần mềm được sử dụng chủ yếu để nhận biết khuynh hướng trong hiệu quả bảo trì, hiệu năng và sự thỏa mãn của khách hàng. Đơn vị đảm bảo chất lượng phần mềm thường thực hiện các tiến trình đo chất lượng. Thay đổi xu hướng tích cực hay tiêu cực, cung cấp cơ sở định lượng cho các quyết định quản lý, quan tâm tới:

- Ước lượng các yêu cầu tài nguyên khi lên kế hoạch bảo trì sửa chữa cho chu kỳ tiếp theo
- So sánh các phương thức thực hiện
- Khởi tạo các hoạt động ngăn ngừa và sửa lỗi
- Ước lượng các yêu cầu tài nguyên như một cơ sở cho các đề xuất các dịch vụ bảo trì mới hoặc đã được điều chỉnh.

Chi phí chất lượng phần mềm

Như trong những phần trước, ở đây chúng ta chỉ quan tâm tới vấn đề bảo trì sửa lỗi. Giá thành chất lượng của bảo trì sửa lỗi được phân loại thành 6 lớp. Sau đây là định nghĩa cho mỗi lớp và ví dụ:

- Chi phí của việc ngăn ngừa: Chi phí ngăn ngừa lỗi, ví dụ như chi phí của việc xây dựng và đào tạo đội bảo trì, chi phí của các hoạt động ngăn ngừa và sửa lỗi.
- Chi phí của việc đánh giá chất lượng: chi phí của việc phát hiện lỗi, ví dụ như chi phí của việc xem xét lại các dịch vụ bảo trì được thực hiện bởi nhóm SQA, đội bên trong và các điều tra sự thỏa mãn của khách hàng.
- Chi phí của việc chuẩn bị và giám sát quản lý: Chi phí của các hoạt động quản lý được thực hiện để ngăn ngừa lỗi, ví dụ như chi phí của việc chuẩn bị kế hoạch bảo trì, việc tuyển đội bảo trì và bám sát hiệu năng bảo trì (follow-up of maintenance performance)
- Chi phí lỗi bên trong: Chi phí của lỗi phần mềm được đề xướng bởi đội bảo trì (ưu tiên cho các yêu sách nhận được từ phía khách hàng)

- Chi phí lỗi bên ngoài: Chi phí của các lỗi phần mềm được đề xướng bởi các yêu sách của khách hàng.
- Chi phí của lỗi quản lý: Chi phí lỗi phần mềm gây ra bởi các hoạt động quản lý hoặc tương tác, ví dụ như chi phí của các thiệt hại từ việc thiếu nhân viên bảo trì hoặc bảo trì không chính xác.

Sau khi xem lại các lớp định giá thành chất lượng phần mềm, được định nghĩa theo các lớp cũng như mô hình mở rộng, sẽ được thảo luận kỹ hơn ở chương 22.

Giá thành lỗi bên trong của các hoạt động bảo trì sửa lỗi phần mềm

Để định nghĩa giá thành lỗi bên trong, có hai chu kỳ bảo trì phải được quan tâm riêng biệt. Đó là: (a) chu kỳ đảm bảo (warranty) (thường từ 3 đến 12 tháng sau khi cài đặt phần mềm) và (b) chu kỳ các dịch vụ bảo trì được ký hợp đồng, bắt đầu ở thời điểm kết thúc chu kỳ đảm bảo. Vấn đề ở đây yêu cầu một quyết định trong trường hợp nào nên quan tâm đến lỗi bên ngoài, chỉ sau khi đưa ra quyết định này thì việc định giá chất lượng mới được xác định và ước lượng. Những định nghĩa được đề xuất về chi phí lỗi bên ngoài và các tham số của chúng cho các dịch vụ sửa lỗi phần mềm và hỗ trợ người dùng dưới đây:

- **Cho sửa lỗi phần mềm:**

- Tất cả chi phí của sửa lỗi phần mềm được đưa ra bởi người sử dụng trong suốt chu kỳ đảm bảo là chi phí chất lượng bên ngoài bởi vì chúng đề cập tới kết quả trực tiếp từ lỗi phát triển phần mềm, hơn nữa người phát triển có trách nhiệm cho việc sửa lỗi của họ trong suốt chu kỳ này.
- Sửa lỗi phần mềm được thực hiện trong suốt chu kỳ bảo trì được ký kết, nó được xem như là một phần chính của dịch vụ, khi trách nhiệm của người phát triển đối với việc sửa lỗi bị giới hạn ở chu kỳ đảm bảo. Chi phí của các dịch vụ này được xem là chi phí các dịch vụ chính thức chứ không phải là chi phí chất lượng...
- Trong suốt chu kỳ bảo trì kí kết, sau những nỗ lực của việc sửa lỗi ban đầu thì chỉ có chi phí cho việc sửa lỗi lại (re-correction) mới được xem là chi phí của lỗi bên ngoài.

- Cho các dịch vụ hỗ trợ người dùng:
 - Trong suốt chu kỳ đảm bảo, các dịch vụ hỗ trợ người dùng được xem như là một phần của nỗ lực làm theo chỉ dẫn (instruction effort), và do vậy không nên quan tâm tới chi phí lỗi bên ngoài.
 - Suốt chu kỳ bảo trì được kí kết, tất cả các kiểu dịch vụ hỗ trợ người dùng, dù là giải quyết một lỗi phần mềm được xác định hay bàn (consultation) về các tùy chọn ứng dụng, là tất cả các phần của dịch vụ chính thức, và chi phí của chúng không quan tâm tới chi phí lỗi bên ngoài.
 - Trong cả hai chu kỳ bảo trì, một lỗi bên ngoài được định nghĩa như một trường hợp nơi sự bàn bạc thứ hai (second consultation) được yêu cầu sau khi sự bàn bạc ban đầu chứng minh là chính xác. Chi phí cung cấp cho sự bàn bạc thứ hai và kỹ hơn cho cùng một trường hợp được xem như là chi phí lỗi bên ngoài.

Như trường hợp tổng quát, thông tin chi phí bảo trì chất lượng, cùng với thông tin giám sát quản lý khác được mong đợi nhằm giúp đỡ cho việc quản lý đưa ra quyết định, được quan tâm như sau:

- Chỉ dẫn cho việc đầu tư trong việc cải thiện các dịch vụ bảo trì bằng cách đưa ra điểm yếu của chi phí chất lượng cực cao và điểm mạnh của chi phí chất lượng cực thấp.
- Phát triển phiên bản được cải tiến của phần mềm (trong trường hợp phần mềm được làm cho khách hàng) hoặc việc thay thế gói phần mềm đã được mua.

8.2. Các CASE tool và ảnh hưởng của nó lên chất lượng phần mềm

8.2.1. Khái niệm CASE tool

Định nghĩa: Công cụ CASE là những công cụ máy tính để phát triển phần mềm mà chúng hỗ trợ người phát triển thực hiện một hoặc nhiều pha trong vòng đời của phần mềm và/hoặc hỗ trợ việc bảo trì phần mềm.

Theo định nghĩa này thì những bộ biên dịch, hệ thống gỡ lỗi tương tác, hệ thống quản lý cấu hình và hệ thống kiểm thử tự động cũng được coi là công cụ CASE. Nói

cách khác, những công cụ máy tính hỗ trợ việc phát triển phần mềm đã có từ lâu (ví dụ như những chương trình gỡ lỗi tương tác, bộ biên dịch hay hệ thống kiểm soát tiến trình dự án) có thể được xem là công cụ *CASE* cổ điển (*classic CASE tool*), trong khi đó những công cụ mới hỗ trợ người phát triển thực thi thành công các pha phát triển của dự án được xem là những công cụ *CASE* thực (*real CASE tool*). Khi nhắc đến những công cụ *CASE* thực ta cần phải phân biệt giữa công cụ *CASE* bên trên (*upper CASE tool*) hỗ trợ pha phân tích và thiết kế với công cụ *CASE* bên dưới (*lower CASE tool*) hỗ trợ cho pha *coding* (trong đó “bên trên” và “bên dưới” liên quan đến vị trí các pha trong mô hình thác nước), và công cụ *CASE* tích hợp hỗ trợ pha phân tích, thiết kế và *coding*.

Thành phần chính của công cụ *CASE* thực là một kho chứa (*repository*) chứa tất cả các thông tin liên quan đến dự án. Thông tin dự án được tích lũy trong kho chứa và được cập nhật mỗi khi có sự thay đổi trong suốt quá trình phát triển phần mềm và trong cả giai đoạn bảo trì. Kho chứa của pha phát triển phía trước sẽ là cơ sở cho pha tiếp theo. Thông tin phát triển tích lũy được chứa trong kho chứa cung cấp sự hỗ trợ cho giai đoạn bảo trì trong đó những công việc bảo trì sửa đổi, đáp ứng hay cải thiện chức năng được thực hiện. Hệ thống máy tính quản lý kho chứa đảm bảo thông tin dự án được duy trì và phù hợp với phương pháp luận phát triển mềm cũng như là đảm bảo các quy chuẩn dựa theo phong cách (*style*), các thủ tục xây dựng và chỉ dẫn công việc. Điều đó làm các công cụ *CASE* có khả năng đưa ra tài liệu dự án đầy đủ và cập nhật bất cứ lúc nào. Một vài công cụ *CASE* bên dưới và *CASE* tích hợp có thể tự động sinh ra mã chương trình dựa hoàn toàn trên thông tin thiết kế được lưu trong kho chứa. Những công cụ kỹ nghệ ngược (*Reverse engineering tool*) cũng được xem là các công cụ *CASE* thực. Dựa trên mã hệ thống, các công cụ này được áp dụng chính cho việc khôi phục và tái tạo tài liệu thiết kế cho hệ thống phần mềm đang được sử dụng (phần mềm “kế thừa”). Nói cách khác, những công cụ *CASE* kỹ nghệ ngược này hoạt động ngược với công cụ *CASE* thông thường: thay vì

tạo ra mã hệ thống dựa trên thông tin thiết kế, chúng tạo ra kho chứa đầy đủ, cập nhật và tài liệu thiết kế dựa trên mã hệ thống.

Sự hỗ trợ của các công cụ CASE cho người phát triển hệ thống được thể hiện trong bảng sau:

Kiểu công cụ CASE	Hỗ trợ
Sửa đổi và biểu đồ	Sửa đổi văn bản và biểu đồ, tạo biểu đồ thiết kế dựa trên các bản ghi trong kho chứa
Truy vấn kho chứa	Hiển thị các phần của văn bản thiết kế, biểu đồ... theo dõi yêu cầu
Viết tài liệu tự động	Tự động sinh ra các tài liệu được yêu cầu theo các bản ghi cập nhật trong kho chứa
Hỗ trợ thiết kế	Việc sửa đổi thiết kế được ghi lại bởi người phân tích hệ thống và quản lý từ điển dữ liệu
Sửa đổi mã	Biên dịch, thông dịch và gỡ lỗi tương tác mã cho ngôn ngữ lập trình cụ thể hoặc công cụ phát triển
Sinh mã	Chuyển từ các bản ghi thiết kế sang mẫu hoặc ứng dụng phần mềm thích hợp với ngôn ngữ lập trình (hoặc công cụ phát triển)
Quản lý cấu hình	Quản lý phiên bản của tài liệu thiết kế và mã phần mềm, kiểm soát sự thay đổi trong thiết kế và mã phần mềm.
Kiểm thử phần mềm	Kiểm thử tự động, kiểm thử tải và

	quản lý kiểm thử và sửa bản ghi
Kỹ nghệ ngược	Xây dựng kho chứa phần mềm và tài liệu thiết kế, dựa trên mã của hệ thống phần mềm thừa kế. Khi kho chứa của phần mềm đã có, nó có thể được cập nhật và sử dụng để tự động tạo ra phiên bản mới của hệ thống
Quản lý dự án và chỉ số phần mềm	Hỗ trợ việc quản lý dự án bằng cách theo dõi lịch và tính toán năng suất, chỉ số lỗi

8.2.2. Đóng góp của CASE tool cho chất lượng sản phẩm phần mềm

Lợi ích của các công cụ CASE đối với chất lượng sản phẩm phần mềm đó là làm giảm một số lượng lớn các lỗi trong các pha phát triển phần mềm. Để đánh giá được lợi ích này, chúng ta cùng xem xét việc cải tiến chất lượng mà các công cụ CASE thực hiện được trong 9 nguyên nhân gây ra lỗi (được liệt kê trong phần 2.3). Đánh giá của chúng ta sẽ bao gồm cả công cụ CASE cổ điển và thực.

Bảng sau sẽ liệt kê những đóng góp của công cụ CASE tới việc cải thiện chất lượng phần mềm:

Nguyên nhân gây ra lỗi	Công cụ CASE cổ điển	Công cụ CASE thực
1. Lỗi trong xác định yêu cầu		Hầu như không đóng góp Việc kiểm tra tính cố định của yêu cầu hay sự chính xác bằng máy tính gần như hiếm xảy ra

<p>2. Việc thất bại trong giao tiếp giữa khách hàng và người phát triển</p>		<p>Hầu như không đóng góp</p> <p>Trong hầu hết các trường hợp, việc xác định sự thất bại trong giao tiếp bằng máy tính là không thể. Những thất bại này chỉ có thể được xác định và ngăn chặn khi có thay đổi hoặc khi những thông tin khác được tìm thấy trở nên mâu thuẫn với những thông tin đã có.</p>
<p>3. Sự chênh lệch có chủ ý trong yêu cầu phần mềm</p>		<p>Đóng góp lớn</p> <p>Dựa trên những thông tin đã có, những chênh lệch từ yêu cầu được xác định như những mâu thuẫn và được liệt vào lỗi. Những chênh lệch này có thể được xác định bằng các công cụ theo dõi yêu cầu đã có và công cụ truy vấn (crossreferenced query tools)</p>

4. Những lỗi trong thiết kế theo logic		Đóng góp lớn <ul style="list-style-type: none"> • Việc thiết kế lại (reengineering) cho phép sản sinh tự động bản thiết kế cho những hệ thống kế thừa và những bản ghi của chúng. • Việc sử dụng nơi lưu trữ nhằm xác định những thiếu sót trong thiết kế, những thay đổi và những mâu thuẫn mới có với những bản ghi đã có (bản ghi đã được lưu trữ từ trước).
5. Lỗi trong coding	Đóng góp rất lớn Áp dụng những trình biên dịch (compiler, interpreters) và những trình gỡ lỗi tương tác (debugger) vào.	Đóng góp rất lớn Áp dụng các công cụ CASE bên dưới vào nhằm tạo ra một cách tự động những đoạn code tương thích hoàn toàn với thiết kế có sẵn. Thêm vào đó, vì code được tự động nên không có lỗi nào xảy ra.
6. Không làm đúng với	Đóng góp có giới hạn	Đóng góp rất lớn

những chỉ dẫn về code và viết tài liệu	Sử dụng những trình biên tập văn bản và kiểm định code để hỗ trợ chuẩn hóa cấu trúc, phong cách của văn bản, code và làm việc xác định những sai lệch không đúng để dàng hơn.	Áp dụng những công cụ CASE bên dưới vào nhằm tạo ra những đoạn code đảm bảo làm đúng với những chỉ dẫn về code và viết tài liệu
7. Những thiếu sót xảy ra trong quá trình test	Đóng góp lớn Những công cụ kiểm tra tự động thực hiện hồi quy và kiểm tra việc nạp một cách tự động. Việc quản lý test và sửa lỗi bằng máy tính làm giảm lỗi.	Đóng góp lớn Áp dụng những công cụ CASE bên dưới, đặc biệt là những công cụ CASE được tích hợp vào để ngăn chặn lỗi trong coding và giảm lỗi thiết kế. Ứng dụng của những công cụ lưu trữ tới việc sửa lỗi và thay đổi trong suốt quá trình phát triển nhằm ngăn ngừa hầu hết các lỗi của phần mềm
8. Lỗi thủ tục	Đóng góp lớn Việc điều khiển các phiên bản, xem xét lại và cài đặt phần mềm	Đóng góp có giới hạn Sử dụng những bản tài liệu đầy đủ và luôn cập nhật làm việc ngăn ngừa

	bằng các công cụ quản lý quản lý cấu hình	các lỗi bảo trì gây ra bởi những tài liệu không đầy đủ hoặc thiếu chính xác, đặc biệt nếu thiết kế đã được sửa lại nhiều lần.
9. Lỗi tài liệu	Đóng góp có giới hạn Áp dụng duy nhất trình biên tập văn bản vào	Đóng góp lớn Sử dụng nơi lưu trữ một cách tự động nhằm sinh ra những bản tài liệu đầy đủ và luôn cập nhật trước mỗi lần sửa lỗi và thay đổi.

8.2.3. Đóng góp của CASE tool cho chất lượng bảo trì phần mềm

Các công cụ CASE (đặt biệt là công cụ CASE thực) có mặt trong nhiều loại của chất lượng bảo trì phần mềm theo nhiều cách khác nhau.

Bảo trì sửa chữa (Corrective maintenance):

- Tài liệu của phần mềm được đã được cập nhật và CASE được đưa ra đầy đủ sẽ giúp tìm ra nguyên nhân gây lỗi (failure) của phần mềm một cách dễ dàng và chính xác hơn.
- Các câu truy vấn cross-referenced cho phép xác định trước kết quả của kế hoạch sửa chữa đang đề ra một cách tốt hơn.
- Sửa chữa bằng các công cụ CASE tích hợp hay bên dưới hỗ trợ coding tự động mà sẽ không có lỗi (error) lập trình nào cũng như tài liệu tự động của việc sửa chữa.

Bảo trì thích nghi (Adaptive maintenance):

- Tài liệu phần mềm đầy đủ và được cập nhật bởi các công cụ CASE cho phép xem xét kỹ lưỡng khả năng thích nghi của gói phần mềm đối với ứng dụng mới, người dùng mới.

Bảo trì cải thiện chức năng (Functional improvement maintenance):

- Việc sử dụng kho chứa cho phép những người thiết kế có thể đảm bảo tính nhất quán của các ứng dụng mới, các cải tiến mới với các hệ thống phần mềm vốn có.
- Các câu truy vấn kho chứa cross-referenced cho phép lên kế hoạch cho việc thay đổi, thêm chức năng một cách dễ dàng hơn
- Các thay đổi và việc thêm các chức năng thực hiện bằng các công cụ CASE tích hợp hay bên dưới cho phép coding tự động mà không có bất cứ lỗi (error) coding nào cũng như tài liệu tự động của các thay đổi và sửa chữa

8.2.4. Đóng góp của CASE tool cho quản lý dự án

(1) Phương pháp sử dụng CASE nâng cao mang lại tính kinh tế cao hơn phương pháp thông thường.

(2) Chất lượng của việc quản lý cả hai dự án là giống nhau với việc ước lượng lịch biểu và tài nguyên dưới mức yêu cầu.

Thông thường, việc áp dụng các công cụ CASE được mong đợi sẽ làm giảm giá thành và thời gian phát triển của dự án (*“shorter time to market”*). uy nhiên, vai trò của các công cụ CASE đối với các khía cạnh chất lượng của quản lý dự án (gồm: điều khiển chi phí và thời gian) là trọng điểm sự quan tâm của chúng ta. Hiện nay, đã có trường hợp sử dụng công cụ CASE hiện đại làm giảm độ lệch giữa kinh phí thực thi và lịch biểu theo kế hoạch, đặc biệt bởi vì chúng ngăn ngừa lượng lỗi (*error*) và cho phép sửa lỗi nhanh, dễ dàng hơn khi có yêu cầu. Để việc quản lý dự án được cải thiện tốt hơn, công cụ điều khiển dự án (ở đây được xem là loại của các

công cụ CASE cổ điển) và các phương pháp luận ước lượng thời gian, kinh phí được cải thiện phải được phát triển.

8.3. Đảm bảo chất lượng phần mềm của các yếu tố bên ngoài cùng tham gia

8.3.1. Những thành phần bên ngoài đóng góp vào dự án phần mềm

Những người tham gia một dự án phát triển phần mềm – tổ chức quan tâm đến hệ thống phần mềm (khách hàng) và tổ chức cam kết tiến hành phát triển (nhà thầu) – ngày nay thường không chỉ là những người tham dự trong dự án. Những người tham gia bên ngoài có liên quan đến dự án phát triển phần mềm đóng góp cho dự án nhưng không phải là nhà thầu, mà cũng không phải là những thành viên của nhà thầu. Sự đóng góp của họ cho dự án được cấu trúc thông qua những thỏa thuận với nhà thầu (nhà thầu phụ hay những người cung cấp của COTS software) hoặc thông qua những điều khoản của hợp đồng dự án, các phần của dự án sẽ được thực thi bởi chính khách hàng của họ. Dự án lớn hơn và phức tạp hơn, có ý nghĩa hơn có thể đúng mà những người tham gia bên ngoài được yêu cầu, phần lớn công việc được chuyển giao hoặc chia ra. Mục đích để chuyển hướng những người tham gia bên ngoài dựa vào một số nhân tố, xác định khoảng cách từ kinh tế đến kỹ thuật và đến những cá nhân liên quan (to personnel – related interests), và mang lại một sự gia tăng mối quan tâm trong sự phân phối của công việc liên quan trong việc hoàn thành những dự án phức tạp.

Những người tham gia bên ngoài có thể được phân chia thành 3 nhóm chính:

- Subcontractors (những nhà thầu phụ, hiện nay được gọi là những tổ chức “outsourcing”) họ cam kết thực hiện các thành phần của 1 dự án, lớn hay nhỏ, tùy theo từng trường hợp. Những nhà thầu phụ thường đưa ra hợp đồng ở mức tối thiểu một trong những lợi ích: khả năng huy động nhân viên, ý kiến về mặt chuyên môn đặc biệt hoặc giá thấp.
- Những nhà cung cấp COTS software và những module phần mềm sử dụng lại. Lợi ích của sự tích hợp các thành phần đó là rất rõ ràng, sự xác định khoảng

cách từ kế hoạch làm việc và giảm bớt giá thành đến chất lượng. Một sự mong đợi mà sự tích hợp của những thành phần sẵn sàng để dùng sẽ được lưu trữ trong những mã nguồn phát triển, một kế hoạch làm việc ngắn hơn và phần mềm chất lượng cao hơn. Phần mềm chất lượng cao hơn thì được chờ đợi nhưng những thành phần đã được kiểm tra và được sửa chữa bởi những người phát triển, tốt như được sửa chữa theo những thiếu sót được xác định bởi khách hàng xem trước. Các tính chất của COTS software và các vấn đề chất lượng liên quan đến sử dụng chúng đã được nhận định bởi Basili và Boehm (2001).

- Khách hàng, bản thân họ như là người tham gia thực hiện dự án. Điều này khá chung để mỗi khách hàng thực hiện các phần của dự án: để áp dụng những chuyên môn đặc biệt của những khách hàng, đáp ứng cho giao dịch hoặc những yêu cầu bảo mật khác, giữ lại những nhân viên phát triển nội bộ đang sử dụng, ngăn ngừa những vấn đề bảo mật trong tương lai.v.v. Trường hợp này có những hạn chế trong những điều khoản của quan hệ khách hàng-người cung cấp cần thiết để thực hiện thành công một dự án. Vì thế, chắc chắn trường hợp này đã trở thành 1 thành phần chuẩn của nhiều dự án phát triển phần mềm và những quan hệ bằng hợp đồng.

8.3.2. Rủi ro và lợi ích của giới thiệu người tham dự ngoài.

Rủi ro chủ yếu tới chất lượng dự án liên quan với người giới thiệu tham gia từ bên ngoài trong cơ cấu của dự án là như sau:

- Sự trì hoãn hoàn thành của dự án.

Trong đó trường hợp người tham gia ở bên ngoài cung cấp chậm những thành phần cho hệ thống phần mềm, dự án nói chung sẽ bị hoãn lại. Sự chậm trễ này chủ yếu là do nhà thầu phụ và khách hàng gây ra chứ không phải do các nhà cung cấp phần mềm có sẵn (COTS).

Trong nhiều trường hợp, trách nhiệm kiểm soát sự phát triển phần mềm của nhà thầu phụ và khách hàng không cao, do đó gây ra tình trạng chậm chạp, trì hoãn và

không có thời gian cho sự thay đổi cũng như thời gian cần thiết để tổ chức lại gây ảnh hưởng tiêu cực với dự án.

- **Các phần dự án chất lượng thấp được cung cấp bởi các thành viên bên ngoài**

Các vấn đề về chất lượng có thể phân loại như

(a) **sai sót:** cao hơn so với số lượng sai sót mong đợi, thường cao hơn nhiều so với mong

đợi

(b) **Coding và tài liệu không chuẩn:** sự vi phạm của phong cách và cấu trúc trong việc xây dựng và các thủ tục (theo giả thuyết như đã ấn định trong bất cứ hợp đồng nào). Phần mềm chất lượng thấp và không chuẩn sẽ gây ra khó khăn trong giai đoạn kiểm thử và sau đó trong giai đoạn bảo trì. Việc yêu cầu thêm thời gian để kiểm tra và chỉnh sửa chất lượng phần mềm chất lượng thấp có thể gây ra sự chậm trễ trong dự án đặc biệt trong trường hợp khi các thành viên bên ngoài hoàn thành nhiệm vụ của họ đúng thời hạn.

- **Khó khăn về bảo trì trong tương lai.**

Thực tế một số tổ chức tham gia việc phát triển nhưng chỉ một trong số họ, nhà thầu, là người trực tiếp gây nên 2 khó khăn trong việc bảo trì:

a) Nhà thầu có thể đối mặt với việc các coding và tài liệu không hoàn thành và/hoặc không đúng tiêu chuẩn từ các thành viên bên ngoài, gây ra sự kém chất lượng trong dịch vụ bảo trì của nhóm bảo trì và nhà thầu sẽ tốn chi phí cao hơn.

b) Các dịch vụ bảo trì được cung cấp bởi nhiều hơn một tổ chức, có thể nhà thầu phụ, nhà cung cấp phần mềm có sẵn COTS và các bộ phận phát triển của khách hàng. Khi mỗi phần này bị hạn chế khả năng đáp ứng, khách hàng buộc phải tìm kiếm người chịu trách nhiệm cho các lỗi cụ thể của phần mềm mỗi khi các lỗi này được phát hiện.

- **Mất sự kiểm soát các bộ phận của dự án**

Dù cố ý hay không cố ý, sự kiểm soát việc phát triển phần mềm của thành viên bên ngoài có thể tạo ra một bức tranh lạc quan không thực tế về tình trạng của dự án. Sự trao đổi với nhóm các thành viên bên ngoài có thể làm gián đoạn tới một vài tuần, gây cản trở việc đánh giá tiến độ của dự án. Kết quả là, cảnh báo về khó khăn trong phát triển, thiếu đội ngũ nhân viên và nhiều vấn đề khác đến muộn với các nhà thầu.

Nhận thức được trước các khó khăn này, nhà thầu phải xem xét kết hợp lợi ích và rủi ro được đưa ra bởi các thành viên bên ngoài trong một dự án.

8.3.3. Những mục tiêu đảm bảo chất lượng về sự đóng góp người tham gia bên ngoài

Những mục tiêu nào thu được bằng việc áp dụng công cụ SQA được cung cấp bởi các thành viên bên ngoài? Những mục tiêu dưới đây có thể thu được trực tiếp từ việc liệt kê các rủi ro đã đề cập ở trên:

- Để tránh trì hoãn hoàn thành nhiệm vụ và để đảm bảo cảnh báo sớm để tính trước sự trì hoãn.
- Để đảm bảo mức độ chất lượng có thể chấp nhận được của bộ phận triển khai và đón nhận cảnh báo sớm của phạm vi chất lượng yêu cầu.
- Để đảm bảo đủ tài liệu phục vụ cho nhóm bảo trì.
- Để đảm bảo liên tục, toàn diện và đáng tin cậy kiểm soát việc thực hiện người tham gia bên ngoài.

8.3.4. Các công cụ đảm bảo chất lượng những đóng góp của các thành viên đóng góp bên ngoài.

Chúng ta mong muốn các thành viên đóng góp bên ngoài thực hiện các các phương thức SQA của chính họ, bao gồm các công cụ cần thiết để các sản phẩm phần mềm và các dịch vụ của họ đạt được tới mức chất lượng có thể chấp nhận được. Các công cụ được đề cập tới ở đây là những thứ mà các nhà thầu có thể áp dụng cho các thành viên đóng góp bên ngoài. Trong mục đích này, vấn đề chất lượng và thời gian là

quan trọng nhất được xác định. Các công cụ chính được áp dụng trước và trong suốt quá trình kết hợp các thành viên đóng góp bên ngoài trong một dự án phát triển phần mềm được liệt kê bên dưới.

- xem xét lại tài liệu yêu cầu.
- Đánh giá các tiêu chuẩn chọn lựa liên quan đến các thành viên đóng góp bên ngoài.
- Thành lập ủy ban điều khiển gia nhập và kết hợp của dự án.
- Sự đóng góp trong sự xem xét thiết kế.
- Sự đóng góp trong kiểm tra phần mềm.
- Cách trình bày các thủ tục đặc biệt
- Xác định các team leader của các nhà cung cấp và các thành viên.
- Chuẩn bị các báo cáo tiến trình phát triển của các hoạt động phát triển dự án.
- Xem xét lại các giao phẩm (các tài liệu) và acceptance tests

BÀI 9. CÁC THÀNH PHẦN CƠ BẢN CỦA CHẤT LƯỢNG PHẦN MỀM

9.1. Thủ tục, chỉ dẫn và các thiết bị hỗ trợ chất lượng

9.1.1. Các thủ tục và chỉ dẫn

9.1.1.1. Khái niệm thủ tục (procedure) và chỉ dẫn (instruction)

- Thủ tục là “một cách cụ thể để đạt được một việc nào đó”. Nói cách khác, thủ tục là những hoạt động và tiến trình chi tiết được thực hiện theo một phương thức đã đưa ra với mục đích hoàn thiện một công việc nào đó.

Những thủ tục được thực hiện trong một tổ chức được xem như liên kết các nhân viên trong tổ chức đó, nghĩa là mỗi nhân viên thực hiện tác vụ của họ theo các bước có trong tài liệu thủ tục liên quan, thường mang tên của tác vụ được chọn lựa. Những thủ tục cũng có xu hướng phổ biến trong mỗi tổ chức, nghĩa là chúng được cung cấp bất cứ khi nào tác vụ được thực thi, bất chấp cá nhân thực hiện tác vụ hay phạm vi tổ chức.

○ Chỉ thị công việc được sử dụng chủ yếu trong trường hợp một phương pháp thực hiện công việc trong suốt tổ chức hoặc không thể làm được hoặc không ai muốn làm. Kết quả là, chỉ thị công việc rất rõ ràng cho nhóm hoặc các phòng, ban; chúng bổ sung các thủ tục bằng cách cung cấp các chi tiết rõ ràng chỉ phù hợp với yêu cầu của 1 nhóm, một phòng hay một đơn vị.

Chúng ta đặc biệt quan tâm đến các thủ tục và chỉ thị công việc đảm bảo chất lượng phần mềm bởi chúng có ảnh hưởng đến chất lượng của sản phẩm phần mềm, bảo trì phần mềm vào quản lý dự án.

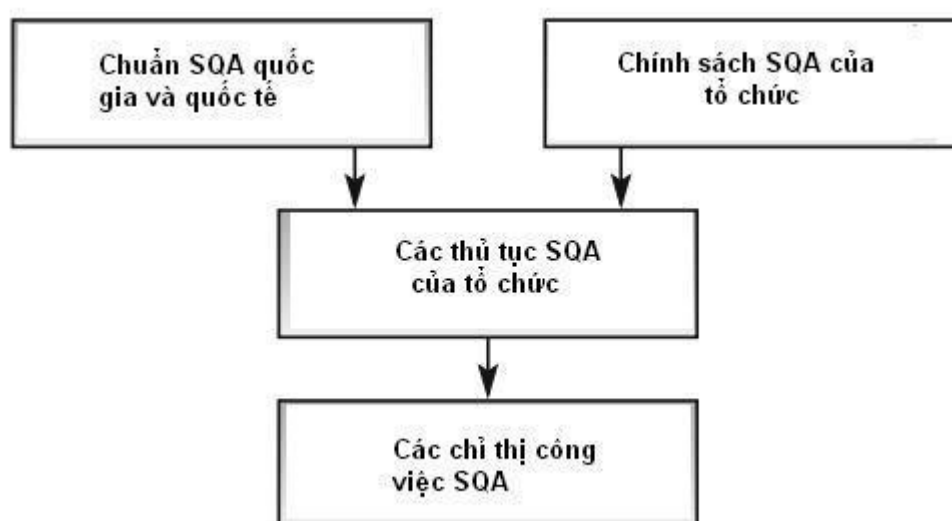
○ Các thủ tục SQA đã được bảo trì và phát triển một cách chuyên nghiệp đòi hỏi phải phù hợp với chính sách chất lượng của tổ chức nhưng cũng có xu hướng phù hợp tới các chuẩn SQA quốc tế và quốc gia. Một điểm quan trọng cần lưu ý khi chuẩn bị chúng là các thủ tục tuân theo chuẩn SQA hỗ trợ chứng thực của hệ SQA của tổ chức. Chuẩn ISO 9000-3 (ISO, 1997; ISO/IEC, 2001) là một trong những chuẩn chứng nhận chính để hướng dẫn chuẩn bị các thủ tục.

9.1.1.2. Sự cần thiết của các thủ tục và chỉ thị công việc

“Tại sao chúng ta nên sử dụng thủ tục và chỉ thị công việc SQA?”

“Nó sẽ tốt hơn không nếu mỗi chuyên gia tin vào kinh nghiệm của anh ta và thực hiện công việc một cách tốt nhất mà anh ta biết?”

“Sẽ có lợi ích nào cho tổ chức nếu buộc tôi chỉ thực hiện các công việc theo cách mà họ lựa chọn?”



Hệ phân cấp mức khái niệm cho sự phát triển các thủ tục và chỉ thị công việc

Những câu hỏi giống như vậy thường xuyên được than phiền bởi nhân viên trong các tổ chức. Các câu trả lời mở ra thách thức phải đáp ứng các thủ tục và chỉ thị công việc: ứng dụng của kiến thức, kinh nghiệm, chuyên môn đã được tích lũy trong tổ chức.

Các thủ tục và chỉ thị công việc SQA nhằm mục đích:

- Thực hiện công việc, tiến trình hoặc các hoạt động theo cách có hiệu quả nhất mà không sai lệch yêu cầu chất lượng.
- Sự giao tiếp có hiệu quả giữa các nhân viên liên quan đến sự phát triển và bảo trì hệ thống phần mềm. Thống nhất trong quá trình thực hiện, đạt được bởi tuân theo các thủ tục và chỉ thị công việc, sẽ giảm thiểu sự hiểu lầm dẫn đến lỗi phần mềm.
- Đơn giản hóa sự phối hợp giữa các công việc và hoạt động được thực hiện bởi các bộ phận của tổ chức. Phối hợp tốt nghĩa là ít lỗi hơn.

9.1.1.3. Các thủ tục và sổ tay thủ tục

Thủ tục cung cấp tất cả những chi tiết cần thiết để thực hiện các tác vụ theo các phương pháp bắt buộc để hoàn thành các chức năng của tác vụ đó. Những chi tiết này có thể được xem lại khi đáp ứng năm vấn đề, được biết đến như là Five W's:

- Những hoạt động nào phải được thực thi? (Which)
- Cách thức mà mỗi hoạt động được thực thi? (how)
- Khi nào thì hoạt động được thực thi? (When)
- Hoạt động được thực thi ở đâu? (Where)
- Ai thực thi hoạt động? (Who)

Sự chuẩn hóa - ứng dụng của cấu trúc và định dạng cố định – là nguyên lý cung cấp cho tất cả các thủ tục SQA.

Các dạng báo cáo và cung cấp tài liệu, nhưng đặc biệt là các danh sách các điều kiện xác định một chuỗi các hoạt động đan xen và bảng xác định giới hạn quyền lực (trách nhiệm), theo hướng thay đổi các đáp ứng để phát triển bên ngoài và thay đổi bên trong các sản phẩm và tác vụ. Hầu hết các thay đổi không ảnh hưởng đến bất cứ sự biến đổi cố hữu nào của thủ tục. Phụ lục cung cấp các chi tiết đơn giản cung cấp cách thuận tiện để đưa ra các thay đổi mà không gây phiền hà cho chính thủ tục.

Sổ tay thủ tục

Tập hợp các thủ tục SQA thường được xem như các sổ tay thủ tục SQA. Nội dung sổ tay thủ tục của bất kỳ tổ chức nào đều theo những cách khác nhau sau:

- Các kiểu hoạt động phát triển và bảo trì phần mềm thực hiện bởi tổ chức
- Phạm vi hoạt động thuộc về mỗi kiểu hoạt động
- Phạm vi khách hàng (ví dụ: khách hàng bên trong, khách hàng làm phần mềm...) và nhà cung cấp (ví dụ: tự phát triển, bảo trì, người thầu phụ, ...)
- Điều chỉnh lựa chọn phương thức được cung cấp bởi tổ chức để đạt được những mục tiêu SQA mong muốn.

Những sự khác nhau này được biểu diễn trong sổ tay thủ tục SQA như thế nào? Trong khi một tổ chức có thể yêu cầu một phạm vi rộng các thủ tục, tổ chức khác lại có thể thỏa mãn với phạm vi thủ tục giới hạn. Tuy nhiên, số lượng thủ tục và cấu trúc của nó phụ thuộc đáng kể vào các quyết định về kiểu (style) và editorial chứ không chỉ phụ thuộc vào kiểu(type) thủ tục.

Một cách tiếp cận hữu ích để xác định cấu trúc bảng nội dung của các sổ tay thủ tục SQA là sử dụng bảng nội dung của chuẩn SQA liên quan như một bộ khung.

9.1.1.4. Chỉ thị công việc và sổ tay chỉ thị công việc

Như đề cập ở trên, chỉ thị công việc xử lý ứng dụng của các thủ tục, phù hợp với yêu cầu của một nhóm dự án khách hàng riêng biệt hoặc các bên liên quan khác.

Trong khi phương pháp luận chung được định nghĩa trong một thủ tục, các chi tiết rõ ràng cho phép ứng dụng cho một dự án hoặc một đơn vị cụ thể thường được trình bày trong một chỉ thị công việc.

Không có trường hợp nào có thể làm chỉ thị công việc mâu thuẫn với thủ tục cha của nó, mặc dù một số chỉ thị có thể kết hợp với bất kỳ thủ tục đưa ra. Điều này có nghĩa là một ai đó có thể thêm, thay đổi hoặc huỷ bỏ chỉ thị công việc mà không thay đổi thủ tục tương ứng.

9.1.2. Chuẩn bị, thực thi và cập nhật các thủ tục và chỉ dẫn

Một sổ tay thủ tục SQA “chủ động” bao gồm nhiều các hoạt động liên tiếp nhằm đảm bảo khả năng tiếp tục áp dụng của thủ tục: ví dụ: quá trình chuẩn bị của các thủ tục, quá trình cài đặt và cập nhật thường lệ. Các hoạt động liên tục này được thực hiện bởi các thành viên trong đội SQA cùng với các thành viên của các đội hoặc đơn vị thuộc tổ chức liên quan, đảm bảo các thủ tục hoàn toàn thích nghi với những thay đổi về công nghệ hoặc khách hàng và đối thủ.

9.1.2.1. Quá trình chuẩn bị cho các thủ tục mới

Các bước ban đầu để phát triển một sổ tay thủ tục SQA mới nên thực hiện với các framework có tính khái niệm và tổ chức nhằm xác định danh sách các thủ tục được

đề xuất và những người sẽ chịu trách nhiệm đối với quá trình chuẩn bị, cập nhật và phê chuẩn. Framework này cũng thường được thể hiện rõ ràng như một thủ tục (thường được gọi là the procedure of procedures).

Bước tiếp theo thông thường sẽ làm với các thủ tục cụ thể. Một cách tiếp cận thông thường để chuẩn bị các thủ tục là triệu tập một ủy ban ad hoc của các chuyên gia làm việc trong các đơn vị liên quan, các thành viên đơn vị SQA và các chuyên gia giải quyết các chủ đề tương ứng. Ủy ban duyệt qua các phác thảo dự định cho tới khi đạt được một phiên bản thỏa mãn yêu cầu, và chỉ dừng lại sau khi thủ tục được phê chuẩn bởi những người được ủy quyền. Một cách tiếp cận khác đối với việc chuẩn bị sổ tay thủ tục là dựa vào sự tư vấn, ở đó một chuyên gia ở bên ngoài được giao trách nhiệm thực hiện việc chuẩn bị một thủ tục, vài thủ tục hoặc hoàn toàn thủ công. Ưu điểm chính của việc thuê người cố vấn là giá trị gia tăng trong kinh nghiệm và chuyên môn của họ trong các tổ chức khác, giảm bớt gánh nặng đối với các chuyên gia lâu năm trong tổ chức cũng như làm giảm thời gian thực hiện công việc. Nhược điểm chính của cách tiếp cận này là làm giảm tính ứng dụng bởi vì các đặc trưng duy nhất của tổ chức.

9.1.2.2. Thực thi các thủ tục mới hoặc các thủ tục được sửa đổi

Sự phê chuẩn một thủ tục mới hoặc được sửa đổi làm dễ dàng hơn đối với quá trình thực thi thủ tục đó, nó tách biệt và thông thường khó đưa ra. Trong nhiều trường hợp, sự phân tán các tài liệu dạng giấy in hoặc email và các chỉ dẫn của đội hoặc đơn vị là không đầy đủ để đảm bảo đủ hoặc gần đủ sự phù hợp. Thực tế là các thành viên của một đội hoặc một ban trong quá trình chuẩn bị thủ tục thuyết phục đồng nghiệp của họ tuân theo các yêu cầu mới nhưng thường không được thỏa đáng. Các chỉ thị tiếp theo và riêng lẻ của những người này thường thiếu hoặc xem nhẹ các thủ tục mới như sự ủy nhiệm cho việc tích hợp các thủ tục bên trong các công việc thường ngày.

9.1.2.3. Cập nhật thủ tục

Động lực cho việc cập nhật các thủ tục hiện hành dựa trên những điều sau:

- Sự thay đổi công nghệ của các công cụ phát triển, phân cứng, thiết bị truyền thông ...
- Thay đổi các lĩnh vực hoạt động của tổ chức
- Các yêu cầu của người dùng tăng dần
- Quá trình phân tích thành công cũng như thất bại
- Đề xuất cải tiến bắt đầu bởi các báo cáo kiểm tra nội bộ
- Sự học hỏi từ kinh nghiệm của các tổ chức khác
- Kinh nghiệm của đội SQA

Khi sự cần thiết của việc cập nhật được nhận ra, một kỹ thuật tương tự như khi áp dụng quá trình chuẩn bị các thủ tục mới có thể được thực hiện: một đội ad hoc được triệu tập để chuẩn bị phiên bản cập nhật, theo sau bởi những người được ủy quyền và các hoạt động thực thi. Điều này ngụ ý rằng việc cập nhật nên được xem xét như là một quá trình được áp dụng trong toàn bộ khâu kiểm tra chất lượng sản phẩm, quan trọng như là việc chuẩn bị các thủ tục mới.

9.1.3. Khuôn mẫu (templates)

Template là một mẫu hoặc một khuôn được sử dụng như một sự chỉ dẫn cho hình dạng của một thứ gì đó sẽ được tạo ra (Từ điển C ao đẳng Webster). Nhưng khi được áp dụng trong công nghệ phần mềm, thuật ngữ template ám chỉ một định dạng (đặc biệt là các bảng khái niệm) được tạo ra bởi các tổ chức, được áp dụng khi soạn thảo một bản báo cáo hoặc viết tài liệu. Việc áp dụng các khuôn hình có thể là bắt buộc đối với một vài tài liệu và không bắt buộc đối với các tài liệu khác; trong một vài trường hợp chỉ một phần của khuôn hình (ví dụ các chương cụ thể hoặc cấu trúc chung) được yêu cầu.

Ví dụ về mẫu:

- Kế hoạch kiểm tra phần mềm (STP)

- Mô tả kiểm tra phần mềm (STD)
- Báo cáo kiểm tra phần mềm (STR)
- Yêu cầu thay đổi phần mềm (SCR)
- Tài liệu về cấu hình phần mềm

9.1.3.1. Đóng góp của các khuôn hình đối với chất lượng phần mềm

Sử dụng khuôn hình tạo nhiều thuận lợi cho các đội phát triển và đội xem xét lại. Đối với đội phát triển, sử dụng khuôn hình có các ưu điểm:

Làm cho dễ dàng hơn trong việc chuẩn bị tài liệu bằng cách tiết kiệm thời gian và năng lượng được yêu cầu để dựng lên cấu trúc của bản báo cáo.

Đảm bảo các tài liệu được chuẩn bị bởi các nhà phát triển đầy đủ hơn bởi tất cả các chủ đề trong tài liệu đã được định nghĩa và được xem xét lại nhiều lần bởi các chuyên gia thông qua việc sử dụng khuôn hình. Các lỗi thông thường, như là bỏ qua một chủ đề thì khó xảy ra hơn.

Dễ dàng tích hợp các thành viên mới trong đội thông qua sự quen thuộc. Cấu trúc tiêu chuẩn của tài liệu, được chuẩn bị theo các khuôn hình mà có thể các thành viên mới đã biết từ trước khi làm việc trong các đội hoặc tổ chức khác, làm cho việc tìm kiếm thông tin dễ dàng hơn. Nó cũng giải quyết ổn thỏa việc chuẩn bị tài liệu tiếp theo khi mà các phần tài liệu đã được chuẩn bị bởi thành viên trong đội khác có thể rời đi hoặc không.

Dễ dàng xem xét lại tài liệu bằng cách loại bỏ sự cần thiết nghiên cứu cấu trúc của một tài liệu và xác nhận sự đầy đủ của nó, khi mà tài liệu được dựa trên khuôn hình thích hợp. Nó cũng làm đơn giản việc xem xét lại tài liệu đầy đủ vì cấu trúc của nó là chuẩn và những người xem xét lại đã quen với nội dung trong đó (các chương, đo ạn và phụ lục). Kết quả của tính nhất quán này là việc xem xét lại được cho là kỹ lưỡng và tốn ít thời gian hơn.

Đối với các đội bảo trì phần mềm, việc sử dụng khuôn hình có ưu điểm:

Làm cho dễ dàng xác định thông tin được yêu cầu cho các nhiệm vụ thực thi bảo trì.

Framework của tổ chức cho việc chuẩn bị, cài đặt và cập nhật các khuôn mẫu

Các tổ chức đều có khuynh hướng tiết kiệm các tài nguyên nội tại, điều đó có nghĩa là những báo cáo thành công trong công việc thì được chuẩn bị cho 1 bộ phận hoặc với mục đích sẽ để nó trở thành những khuôn mẫu cho toàn bộ tổ chức. Do đó, nếu những báo cáo của ông Brown hoặc ông Johnson đã có tiếng là toàn diện và có tính chuyên nghiệp cao thì các kiểu mẫu nội dung của những báo cáo đó có thể được sử dụng như là khuôn mẫu cho những đồng nghiệp của họ. 1 điều bất lợi của việc này đó là không phải ai được hưởng thuận lợi từ những khuôn mẫu này cũng có thể nhận ra được sự tồn tại của chúng. 1 điều bất lợi khác là sự cải tiến của những khuôn mẫu, đã được hoàn thiện qua sự kiểm duyệt(review) của những đội ngũ chuyên nghiệp, có thể bị cản trở.

Đơn vị SQA thường có trách nhiệm cho việc chuẩn bị những khuôn mẫu có tính chuyên nghiệp chọn từ những kiểu thông dụng hơn của các báo cáo và tài liệu đã được yêu cầu của các nhân viên trong tổ chức. Những sự bắt đầu không chính thức từ 1 lĩnh vực có thể khuyến khích đơn vị SQA hoạt động, nhưng việc phát triển cơ sở hạ tầng chung cho việc sử dụng các khuôn mẫu, chủ đề của phần này, thì thuộc về nhiệm vụ của đơn vị SQA.

9.1.3.3. Sự chuẩn bị cho những khuôn mẫu mới

Sự phát triển của 1 cơ sở hạ tầng cho khuôn mẫu hiển nhiên được đặt vào vị trí trung tâm trong công việc của nhóm chuyên nghiệp được dành riêng cho nhiệm vụ này. Nhóm này (có thể là 1 ủy ban) nên thêm vào những nhân viên lâu năm, những người miêu tả những đường lối phát triển phần mềm khác nhau, trưởng kỹ sư phần mềm và các thành viên đơn vị SQA. Những người phát triển không chính thức của “các dịch vụ khuôn mẫu” cũng nên được khuyến khích tham gia vào nhóm

Một trong những nhiệm vụ đầu tiên của nhóm là biên soạn 1 danh sách các khuôn mẫu mục tiêu để phát triển. Khi danh sách được chấp nhận, thứ tự mức ưu tiên sẽ

được thiết lập. Những cái có mức ưu tiên cao được cho vào loại khuôn mẫu của những tài liệu thông thường nhất đã được chuẩn bị, cũng có thể được đưa vào những khuôn mẫu không chính thức đã sẵn sàng sử dụng (nó được ước lượng khi chỉ cần công sức tối thiểu cho việc hoàn thiện và phân quyền). Các ủy ban con được giao nhiệm vụ chuẩn bị những bản phác thảo đầu tiên. 1 thành viên đơn vị S QA có thể được xem xét giao nhiệm vụ làm leader, nhưng ngoại lệ cơ hội cũng có thể dành cho 1 thành viên của ủy ban. Bất kể người trưởng nhóm là ai, anh ấy (hoặc chị ấy) phải nhận thấy được sự phân bổ các bản thảo khuôn mẫu trong số các thành viên, việc tổ chức buổi họp và các việc kế tiếp được tạo ra bởi các ủy ban con có nhiệm vụ chuẩn bị khuôn mẫu. Sự phân bổ của các bản thảo khuôn mẫu trong các team leader cho việc việc bình luận của họ có thể tạo ra những sự phát triển quan trọng và cùng 1 lúc đẩy mạnh sử dụng các khuôn mẫu trong tương lai.

Các nguồn thông tin thông thường nhất được sử dụng trong việc chuẩn bị khuôn mẫu được liệt kê như sau:

- Các khuôn mẫu không chính thức đã được sử dụng trong tổ chức
- Các ví dụ khuôn mẫu được tìm thấy trong sự công bố của các chuyên gia
- Các khuôn mẫu đã được sử dụng bởi những tổ chức giống nhau

9.1.3.4. Áp dụng các khuôn mẫu

Vài quyết định cơ bản được đưa vào sự cài đặt những khuôn mẫu mới hoặc những khuôn mẫu đã được cập nhật:

- Những kênh nào nên được sử dụng để quảng bá những khuôn mẫu này
- Bằng cách nào các khuôn mẫu được tạo có thể sẵn sàng cho những “khách hàng” nội tại của tổ chức

Những khuôn mẫu nào sẽ là bắt buộc và làm sao có thể áp dụng được chúng

Tất cả các phương pháp liên lạc chuyên nghiệp nội bộ có thể được sử dụng để quảng bá các khuôn mẫu bên trong tổ chức: truyền đơn, e-mail, mạng SQA nội bộ cũng như những bài trình bày ngắn trong các buổi họp

Một trong số các phương pháp có hiệu quả nhất của việc làm cho các khuôn mẫu luôn sẵn sàng cho tổ chức là mạng nội bộ, được dùng nhiều hơn là các kênh truyền trên giấy tờ. Sự phân bố qua mạng nội bộ đảm bảo cho người sử dụng lựa chọn những phiên bản được cập nhật của các khuôn mẫu cần thiết và cùng 1 lúc lưu lại những cái chính (dành cho các khuôn mẫu dựa trên giấy tờ) của các bảng biểu nội dung của tài liệu.

Những chỉ thị bắt buộc phải sử dụng trong các khuôn mẫu đặc biệt thường được tìm thấy trong thủ tục hoặc hướng dẫn công việc của tổ chức. Người đứng đầu đội kỹ sư phần mềm hoặc các nhân viên lâu năm thường được ủy quyền để xác định danh sách các khuôn mẫu có tính bắt buộc và phù hợp với những thủ tục đã được chọn, mặc dù chúng ta có thể mong đợi rằng khuôn mẫu được tập hợp lại để đưa ra danh sách được đề cử của *chính nó*.

9.1.3.5. Cập nhật các khuôn mẫu

Việc quyết định cập nhật 1 khuôn mẫu đã có có thể được xem như là 1 tiêu chuẩn đánh giá lại (reactive measure), nó phụ thuộc vào bất kì yếu tố sau:

Ý kiến và đề xuất của người dùng

Những sự thay đổi bên trong lĩnh vực hoạt động của tổ chức

Những đề xuất đã được khởi xướng bởi việc xem xét lại thiết kế (design review) và đội kiểm tra dựa trên sự xem xét lại các tài liệu đã chuẩn bị tùy theo từng khuôn mẫu

Sự phân tích thất bại cũng như thành công

Kinh nghiệm khác của tổ chức

Đề xuất của đội SQA

Các công việc tiếp theo của việc cập nhật khuôn mẫu khá giống với của việc chuẩn bị khuôn mẫu.

9.1.4. Danh mục kiểm tra (Checklists)

Checklists được sử dụng bởi các nhà phát triển phần mềm để cập đến danh sách khoản mục đặc biệt được xây dựng cho từng loại tài liệu, hay một menu chuẩn bị các việc cần hoàn thành trước khi thực hiện một hoạt động (ví dụ như, cài đặt một gói phần mềm tại trang web của khách hàng). Checklists được lên kế hoạch toàn diện nếu công việc chưa hoàn thành. Thông thường, Checklists có xu hướng được coi là một công cụ tùy chọn cơ sở hạ tầng, phụ thuộc chủ yếu vào danh sách của thuộc tính chuyên nghiệp, sự hiểu biết của người dùng với danh sách và tính khả dụng. Một số checklists có 2 mục đích: trong khi cung cấp một danh sách đầy đủ các khoản mục được xác nhận, nó cũng cung cấp không gian cho việc tìm kiếm tư liệu.

Tiếp theo, chúng ta cần thấy được sự đóng góp của checklists vào chất lượng phần mềm và những nỗ lực cần thiết để thành lập, duy trì và áp dụng các checklists đó.

9.1.4.1. Những đóng góp của checklists để phần mềm có chất lượng

Cũng giống như các mẫu, checklists cung cấp nhiều lợi ích để phát triển team, duy trì phần mềm của team và chất lượng của tài liệu.

Những lợi thế để phát triển các nhóm như sau:

- **Giúp các nhà phát triển thực hiện tự kiểm tra các tài liệu hoặc code** của phần mềm trước khi tài liệu hoặc code của phần mềm hoàn thành và đưa vào thiết kế chính thức. Checklists được dùng để giúp đỡ nhà phát triển khám phá không đầy đủ các phần, cũng như phát hiện các lỗi mà không nhận thấy được. Checklists cũng được dự kiến sẽ đóng góp vào chất lượng của các tài liệu hoặc code của phần mềm để xem xét các vấn đề như chất lượng sẽ được khảo sát bằng cách xem xét lại các vấn đề đã được liệt kê trong checklist
- **Giúp phát triển chuẩn bị cho công việc của họ như:** cài đặt phần mềm tại trang web của khách hàng, thực hiện kiểm tra chất lượng tại các trang web của người thầu phụ hoặc ký kết hợp đồng với nhà cung cấp các mô-đun có thể sử dụng

lại của phần mềm. Checklists dự kiến để giúp các nhà phát triển trang bị tốt hơn cho công việc hiệu quả.

Những lợi thế để rà soát nhóm là:

- **Đảm bảo đầy đủ các tài liệu để xem lại bằng cách xem xét thành viên trong nhóm** cũng như tất cả các khoản mục có liên quan xuất hiện trên danh sách.
- **Tạo điều kiện cải thiện hiệu quả của các phiên review**, xem như là một môn học và trật tự các cuộc thảo luận được định nghĩa và cũng được biết trước.

9.1.4.2. Cấu trúc framework cho việc chuẩn bị, thực thi và cập nhật các checklist

Mặc dù được đánh giá cao, nhưng việc sử dụng các checklist vẫn mang tính tùy ý. Sự chuẩn bị và cập nhật checklist, cũng như quá trình sử dụng vẫn thường được ấn định cho “SQA unit”. Một “checklist group”, đứng đầu bởi 1 thành viên của “SQL unit”, có thể đảm trách công việc liên tục thu thập những danh sách đã được cập nhật. Đội ngũ khác bao gồm những người quan tâm đến việc xúc tiến việc sử dụng các checklist tình nguyện tham gia vào trong nhóm.

Tuy nhiên trong 1 số trường hợp thì việc trợ giúp của 1 cố vấn SQL là cần thiết. Ở phần còn lại, chúng ta sẽ miêu tả những quy trình cần thiết để duy trì cơ sở hạ tầng của 1 checklist như là sự chuẩn bị cho các checklist mới, quá trình xúc tiến sử dụng và cập nhật các checklist đó.

9.1.4.3. Sự chuẩn bị cho những checklist mới:

Một trong những công việc đầu tiên chuẩn bị cho “checklist group” là việc biên dịch những danh sách những checklist được nghiên cứu và phát triển theo sự định nghĩa của tiêu chuẩn chung cho tất cả những checklist được nhóm đưa ra.

Những checklist được nhóm chọn thường là những checklist bình thường đã được sử dụng bởi một số thành viên của nhóm phát triển và những nhà phê bình (reviewer). Trong hầu hết các trường hợp, 1 chút thay đổi và sự chỉnh sửa của những checklist này là đủ để thỏa mãn các tiêu chuẩn về nội dung hay định dạng mà

nhóm đã đưa ra. Sự chuẩn bị cho những checklist mới cũng như sự cải tiến những checklist thông thường được hỗ trợ từ những nguồn thông tin sau:

Những checklist thông thường đã được sử dụng trong quá trình tổ chức.

Những checklist mẫu được tìm ra trong sách vở và những ấn phẩm chuyên ngành khác.

Các checklist được sử dụng bởi những tổ chức tương tự.

Quá trình chuẩn bị những checklist mới cũng tương tự như việc chuẩn bị cho các mẫu.

9.1.4.4. Xúc tiến sử dụng checklist

Do việc sử dụng checklist này hiếm khi là bắt buộc, cho nên việc xúc tiến sử dụng đều dựa trên lợi ích của việc quảng cáo và bảo hành. Những kênh thông tin nội địa có thể được sử dụng cho việc giới thiệu những checklist này đến người sử dụng như: tờ rơi, e-mail, mạng SQA cũng như những cuộc họp cấp cao. Tuy nhiên mạng lưới nội địa vẫn là phương thức thích hợp hơn và có hiệu quả cao nhất giúp cho việc làm cho các checklist này đến với những tổ chức khách hàng nội địa.

9.1.4.5. Cập nhật các checklist:

Giống như các mẫu và các thủ tục, việc chủ động cập nhật các mẫu checklist đã có nhìn chung theo các nguồn sau:

- Sự đề xuất và gợi ý của người sử dụng.
- sự thay đổi về công nghệ, lĩnh vực hoạt động và nhóm khách hàng.
- Những đề nghị truyền thụ với ý định tổng quan và những đội kiểm tra Phát ra từ những tổng quan tài liệu.
- Việc phân tích thất bại cũng như thành công.
- Những kinh nghiệm tổ chức khác
- Năng lực của đội ngũ SQA

Quy trình cập nhật các checklist khá giống với sự chuẩn bị chúng.

9.2. Đào tạo đội ngũ và cấp chứng chỉ

Người ta luôn nói rằng muốn nhân viên theo kịp với những kiến thức nghề nghiệp mới có gần đây nhất là chìa khóa để đạt chất lượng trong phát triển và bảo trì phần mềm. Tuy nhiên, có một điều thường được chấp nhận rằng đào tạo chuyên nghiệp, đào tạo lại hoặc cập nhật là bắt buộc (mandatory) để khe hở giữa kiến thức nghề nghiệp được yêu cầu và kiến thức hiện tại càng hẹp càng tốt. Chứng chỉ bên trong (internal certification) (dưới đây chỉ “chứng chỉ”) của nhân viên giữ vị trí chính trong phát triển và bảo trì phần mềm là loại khác – một công cụ để đảm bảo chất lượng nghề nghiệp.

Điều quan trọng của việc đào tạo nghề nghiệp như là một phần của bất kỳ một hệ thống đảm bảo chất lượng (SQA) nào, được nhấn mạnh trong chuẩn ISO 9000 -3 cũng như trong CMM Guidelines (có thể tìm thấy ISO, 1997; ISO/IEC, 2001; Paulk et al., 1995). Phát triển mô tả nghề nghiệp và chương trình đào tạo cho toàn bộ cán bộ công nhân viên SQA đã được Mendis (1999) thảo luận. Chương trình cấp giấy chứng nhận cho các kỹ sư chất lượng phần mềm cũng được **American Society for Quality (ASQ)** quan tâm, được mô tả bởi Hamilton (1999) và một cuốn sổ ASQ (ASQ, 1999).

9.2.1. Mục tiêu của đào tạo và cấp chứng chỉ

Mục tiêu của việc đào tạo và cấp chứng chỉ đó là:

- Để phát triển sự hiểu biết và kỹ năng cho nhân viên mới cần để thực hiện các công việc phát triển và bảo trì phần mềm ở mức độ hiệu quả và có hiệu lực. Sự huấn luyện như vậy tạo điều kiện thuận lợi cho việc các nhân viên mới hòa nhập với đội dự án.
- Để đảm bảo sự phù hợp với những tiêu chuẩn của tổ chức cho những sản phẩm phần mềm (tài liệu và code) bởi những kiểu truyền và cấu trúc các thủ tục với những chỉ dẫn của công việc.

- Để cập nhật sự hiểu biết và kỹ năng của những nhân viên từng trải trong việc đáp lại những phát triển của tổ chức, và đảm bảo hiệu quả và việc thực hiện tốt các tác vụ cũng như sự phù hợp với kiểu của tổ chức và cấu trúc thủ tục và các chỉ dẫn công việc.
- Để truyền những tri thức của những thủ tục đảm bảo chất lượng phần mềm. Để đảm bảo rằng những ứng cử viên cho những vị trí chính trong phát triển và bảo trì phần mềm được xác định đầy đủ.

9.2.2. Tiến trình đào tạo và cấp chứng chỉ

Hoạt động của một hệ thống đào tạo và cấp chứng chỉ thành công yêu cầu những hành động sau phải được thực hiện một cách đều đặn:

- Xác định yêu cầu về sự hiểu biết nghề nghiệp với mỗi vị trí.
- Xác định nhu cầu đào tạo và cập nhật kiến thức nghề nghiệp (professional)
- Lập kế hoạch cho chương trình đào tạo chuyên nghiệp
- Lập kế hoạch cho việc cập nhật kiến thức nghề nghiệp
- Xác định chứng chỉ yêu cầu của từng vị trí
- Lập kế hoạch cho quy trình cấp chứng chỉ
- Thực hiện những việc tiếp theo của những nhân viên đã được đào tạo và cấp chứng chỉ.

Tất cả những hoạt động này được hội tụ vào trong một quá trình tổng hợp trong đó những phản hồi từ những hoạt động trước và những thông tin về những sự phát triển nghề nghiệp kích thích một chu trình của việc đào tạo, cấp chứng chỉ và sự thích nghi với những thay đổi về yêu cầu chất lượng.

Các hoạt động đào tạo và cấp chứng chỉ có nghĩa là để lấp đầy những nhu cầu của những nhân viên từng trải và những nhân viên mới. Tiếp theo những kết quả của chương trình hiện thời cũng như là theo dõi những sự phát triển trong nghề nghiệp

được yêu cầu để đảm bảo rằng chương trình được cập nhật một cách đầy đủ. Thảo luận chi tiết của mỗi hoạt động này được thể hiện trong những đoạn tiếp theo.

9.2.3. Xác định yêu cầu kiến thức chuyên môn và sự cần thiết của đào tạo và cập nhật

Các vị trí thông thường nhất trong một tổ chức phát triển và bảo trì phần mềm là: những người phân tích hệ thống, lập trình viên, trưởng nhóm phát triển phần mềm, trưởng nhóm lập trình, kỹ thuật viên bảo trì phần mềm, nhân viên kiểm thử phần mềm, trưởng nhóm kiểm thử phần mềm. Hầu hết các tổ chức đều đặt ra các yêu cầu về giáo dục và kinh nghiệm cho mỗi vị trí trên. Các thành viên của phòng ban là những người đáp các yêu cầu đào tạo tại chỗ cần phải bổ sung những hiểu biết và kỹ năng về “local” hoặc “internal”, có liên quan đến việc phát triển rõ ràng và các thủ tục bảo trì. Những hiểu biết về chuyên môn có thể được nhóm lại vào 2 loại:

- Kiến thức và kỹ năng về chủ đề kỹ nghệ phần mềm, thí dụ như các công cụ phát triển phần mềm, các phiên bản ngôn ngữ lập trình, các phiên bản CASE tool ứng dụng bởi tổ chức riêng biệt hoặc đơn vị. Các thủ tục và các chỉ dẫn công việc thích hợp này được biên soạn để thực hiện chúng đó cũng là bổn phận của mục này.
- Hiểu biết về chủ đề SQA, những thủ tục gắn liền với rất nhiều hoạt động phát triển và bảo trì. Sự phân công được thực hiện bởi một người nắm giữ một chức vụ đặc biệt.

9.2.4. Xác định những nhu cầu đào tạo và cập nhật (updating)

Những nhu cầu đào tạo và cập nhật được xác định bằng việc so sánh kiến thức hiện tại của nhân viên với những yêu cầu về kiến thức được cập nhật. Kiểu đào tạo này được điều chỉnh để phù hợp với 3 nhóm nhân viên riêng biệt:

- Đào tạo: Cho những nhân viên mới, theo vị trí họ được phân công.
- Đào tạo lại: Cho những nhân viên được phân công đến vị trí mới hoặc tiếp nhận những nhiệm vụ mới.
- Cập nhật: Cho đội ngũ nhân viên như yêu cầu cầu bới vị trí của họ.

Nhu cầu cập nhật nhân viên được đánh giá thường xuyên để thuận tiện lập kế hoạch cho những chương trình yêu cầu.

Cuối cùng, việc tiếp theo là đánh giá nhân viên trong các phòng đào tạo và cập nhật cung cấp đầu vào chính để sử dụng cho việc định nghĩa lại nhu cầu đào tạo.

9.2.5. Lên kế hoạch đào tạo và chương trình cập nhật

Từ thực tế thấy rằng, 2 chương trình cơ bản phải được đề ra - 1 cho chủ đề kỹ nghệ phần mềm và 1 cho chủ đề SQA.

- *Lập kế hoạch chương trình đào tạo và cập nhật cho chủ đề kỹ nghệ phần mềm.*

Sự tính toán về thời gian của những hoạt động đào tạo và đào tạo lại không thể được xác định sớm bởi vì những cán bộ công nhân mới được tuyển mộ và và những nhân viên kỳ cựu được chuyển vị trí thường xuyên sau khi được thông báo trước tương đối ngắn. Mặc dù, các hoạt động cập nhật có thể có thể được sắp xếp trước dễ dàng, với nội dung hoàn tất đóng kín đến thời điểm thực hiện chúng. Bất kể dù những chương trình này được tiến hành trong tổ chức hay bởi một tổ chức outsourcing, nhân viên cấp cao, ví dụ như kỹ sư trưởng phần mềm, thường xuyên tham gia chuẩn bị chúng.

- *Lập kế hoạch những chương trình đào tạo và cập nhật cho chủ đề SQA*

Những chương trình đào tạo cho các chủ đề SQA bao gồm đào tạo cho nhân viên mới cũng như cập nhật cho những nhân viên kỳ cựu. Đặc điểm chung của chương trình đào tạo SQA cho phép chúng được tổ chức theo định kỳ, một hoặc 2 tháng một lần, và phân phối đến tất cả nhân viên mới tuyển mộ trong thời gian đó. Đặc thù của chương trình cập nhật SQA là được thực hiện một lần một năm, hoặc cứ sáu tháng một lần, tùy thuộc vào tốc độ của sự thay đổi. Đơn vị SQA hoặc những người khác chịu trách nhiệm về kết quả SQA trong tổ chức thường tổ chức những chương trình đào tạo và cập nhật này.

9.2.6. Định nghĩa các vị trí yêu cầu cấp chứng chỉ

Thông thường việc chỉ định nhân viên vào vị trí chủ chốt trong phát triển phần mềm và các tổ chức bảo trì đòi hỏi phải hết sức thận trọng. Một trong những thủ tục được sử dụng để đảm bảo sự phù hợp của các ứng cử viên là giấy chứng nhận.

9.2.7. Lên kế hoạch các tiến trình cấp chứng chỉ

Cấp giấy chứng nhận là nhằm cung cấp một framework cho kiểm tra toàn diện khả năng của một ứng cử viên và một biểu hiện của kiến thức chuyên môn và kỹ năng. Những chi tiết của quá trình cấp giấy chứng nhận duy nhất cho các tổ chức. Chúng phản ánh những tính năng đặc biệt, lĩnh vực chuyên môn, phát triển phần mềm và công cụ bảo trì, khách hàng... Bởi vì các quá trình này hướng về những điều cần thiết và quyết định của các tổ chức cụ thể, nội bộ cấp giấy chứng nhận không thể tự động thay thế bởi các giấy chứng nhận chung mà công nhận bởi những người chuyên nghiệp và nhà cung cấp hàng đầu của công cụ phát triển và phần mềm mạng lưới thông tin liên lạc hoặc tương tự.

Việc cấp giấy chứng nhận trong từng chi tiết và mỗi vị trí, đòi hỏi phải chấp nhận như được định nghĩa trong các thủ tục cấp giấy chứng nhận. Các thủ tục cấp giấy chứng nhận điển hình.

Đối với những người đã từng được cấp giấy chứng nhận, một quá trình cấp giấy chứng nhận tiêu biểu đòi hỏi phải đáp ứng một số hoặc là tất cả các yêu cầu sau đây:

- Giáo dục chuyên môn: học tập hay kỹ thuật cao và một số trường hợp cấp giấy chứng nhận bởi một tổ chức chuyên nghiệp hoặc bởi một người lãnh đạo sản xuất phần mềm thương mại.
- Các khoá đào tạo nội bộ
- Chuyên gia kinh nghiệm trong tổ chức (có thể là một phần hoặc hoàn toàn thay thế bằng kinh nghiệm trong tổ chức khác)
- Đánh giá về những thành tựu và khả năng như đã nêu trong định kỳ thực thi việc đánh giá

- Ước lượng bởi các cấp cao hơn của ứng cử viên (thường được hoàn thành bởi một câu hỏi đặc biệt)
- Biểu hiện của kiến thức và kỹ năng nghề là một cuộc thử nghiệm hay dự án.
- Sự giám sát của người nhiều kinh nghiệm cho một giai đoạn nhất định về thời gian.

Chức năng của uỷ ban cấp giấy chứng nhận

Tương tự mô hình đề nghị cho chương trình đào tạo và tái đào tạo, cá nhân hay là uỷ ban chịu trách nhiệm về cấp giấy chứng nhận thường có thâm niên trong phát triển phần mềm và duy trì đội ngũ. Trách nhiệm của những người cấp giấy chứng nhận bao gồm:

- Thực hiện quá trình cấp giấy chứng thực trên cơ sở thực hiện các yêu cầu cá nhân hay đơn vị và chấp nhận cấp giấy chứng nhận cho những người đủ điều kiện.
- Tiếp tục cấp giấy chứng nhận (như cố vấn) thực hiện bởi những người khác.
- Cập nhật các yêu cầu cấp giấy chứng nhận trong hướng ứng sự phát triển trong tổ chức tốt như các chuyên gia.

Thay đổi danh sách các vị trí yêu cầu cấp giấy chứng nhận.

9.2.8. Phân phối các chương trình đào tạo và cấp chứng chỉ

Đào tạo và cập nhật có thể bao gồm các chủ đề như kỹ nghệ phần mềm, bảo đảm chất lượng phần mềm và kỹ năng quản lý (trong khuôn khổ cấp giấy chứng nhận hoặc cho các thông tin chung), tất cả đều được phối hợp với nhu cầu của tổ chức hay doanh nghiệp. Làm thế nào để đào tạo và cập nhật mang lại sự thay đổi phù hợp. Các khoá học có thể truyền trong các định dạng mà bài giảng và thuyết minh ngắn, thường chỉ kéo dài nửa ngày, để tổ chức các khoá học dài một vài tuần hoặc tháng. Những khoá học này có thể thực hiện tại nhà bởi các đơn vị đào tạo của tổ chức, hoặc ra bên ngoài, bởi chương trình hướng nghiệp hoặc cơ sở giáo dục học tập đặt chương trình nền móng cho các yêu cầu của tổ chức.

Thông tin về tổ chức và cung cấp đào tạo và chương trình cấp giấy chứng nhận có thể tìm thấy trong quản lý nhân lực.

9.2.9. Những công việc tiếp theo của việc đào tạo và cấp chứng chỉ

Những người quản lý và chuyên gia phần mềm thường nghi ngờ về những tác động của đào tạo và chứng chỉ trong công tác đào tạo và cấp giấy chứng nhận nói chung hoặc của một trong những hoạt động liên quan. Họ luôn đòi hỏi cho dù những nguồn nhân lực và nỗ lực đầu tư trong đào tạo thực sự có giá trị. Để kiểm định những nghi ngờ, hệ thống theo dõi là thực sự cần thiết để cung cấp thông tin phản hồi tới những đơn vị chuyên nghiệp. Những thông tin phản hồi này cho biết những nỗ lực đào tạo được điều chỉnh tại cùng thời điểm để đảm bảo sự phát triển tiếp diễn của những hoạt động đào tạo và cấp chứng chỉ. Những thông tin cung cấp bởi hệ thống theo dõi liên hệ tới:

- Tất cả những hoạt động đào tạo và thủ tục cấp chứng chỉ tiến hành – những bản ghi về hiệu suất của những người tham gia chương trình.
- Thông tin về trường hợp đặc biệt của các hoạt động đào tạo là minh chứng được đánh giá cao thành công hay thất bại trong việc cải thiện hiệu suất của nhân viên.
- Thông tin về những trường hợp của những thất bại của nhân viên đạt chứng chỉ trong việc thực thi rõ ràng không đạt những yêu cầu chứng chỉ.

9.3. Các hành động sửa lỗi và phòng ngừa

Các hoạt động mang tính hệ thống mà thực thi việc nâng cao hiệu quả và hiệu suất hoạt động được gọi là các hoạt động ngăn chặn và sửa lỗi (CAPA). Có những hoạt động không phải nhằm sửa các lỗi được tìm thấy ngay lập tức mà để loại bỏ nguyên nhân của các lỗi này trong suốt quá trình phát triển phần mềm.

Bằng việc không ngừng đẩy mạnh cải tiến hiệu suất và hiệu quả, tiến trình CAPA đã trở thành một trong những công cụ chính được sử dụng để giành được mục đích

hướng thực thi của SQA: vừa thỏa mãn các yêu cầu chức năng và các quản lý vừa giảm chi phí phát triển phần mềm, bảo trì và các hoạt động đảm bảo chất lượng.

9.3.1. Định nghĩa hoạt động sửa lỗi và phòng ngừa

Các hành động sửa lỗi: một tiến trình phản hồi được áp dụng thường xuyên bao gồm việc tập hợp thông tin về các hoạt động không tuân theo chất lượng, xác định và phân tích các nguồn không tuân theo các quy tắc cũng như việc phát triển và đồng hóa các thủ tục cải tiến, cùng việc kiểm soát việc cài đặt và đánh giá các kết quả của họ.

Các hoạt động ngăn chặn: một tiến trình phản hồi được áp dụng thường xuyên bao gồm việc tập hợp thông tin về các vấn đề chất lượng có khả năng xảy ra, xác định và phân tích sự sai lệch về các tiêu chuẩn chất lượng, việc phát triển và đồng hóa các thủ tục cải tiến, cùng việc kiểm soát việc cài đặt và đánh giá các kết quả của họ.

9.3.2. Tiến trình hành động sửa lỗi và phòng ngừa

Hoạt động thành công của 1 tiến trình CAPA bao gồm những thao tác sau

- (1) Thu thập thông tin
- (2) Phân tích thông tin
- (3) Việc phát triển các giải pháp và các phương thức cải tiến
- (4) Việc cài đặt các phương thức cải tiến
- (5) Follow-up (công việc tiếp theo)

Tiến trình này thường được làm bởi luồng thông tin từ các tài nguyên khác nhau. Để ước lượng sự thành công của tiến trình này, 1 vòng lặp phản hồi đóng được áp dụng để điều khiển luồng thông tin, sự thực thi của việc tìm kết quả thay đổi trong các tình huống thực tế và các thủ tục cùng với kích thước của các kết quả.

9.3.3. Thu thập, phân tích thông tin

- Thu thập thông tin

Sự đa dạng của các tài nguyên thông tin, bên trong và bên ngoài, cung cấp cho tiến trình CAPA khá rõ rệt. Theo sự phân đôi bên trong/ bên ngoài, 4 tài nguyên thông

tin chính bên trong là (1) tiến trình phát triển phần mềm, (2) việc bảo trì phần mềm, (3) Cơ sở hạ tầng SQA và (4) các thủ tục quản lý chất lượng phần mềm. Tài nguyên thông tin bên ngoài chủ yếu là các thông kê những yêu cầu của khách hàng và những phần này của họ.

- ***Phân tích thông tin***

Thao tác thông thường của tiến trình CAPA có mục đích là tạo 1 luồng lớn các tài liệu liên quan đến 1 phạm vi rộng lớn của thông tin.

Phân tích bao gồm:

- Lọc thông tin và xác định những sự cải tiến tiềm tàng. Các tài liệu đã nhận từ nhiều nguồn thông tin khác nhau được xem xét lại bởi những người chuyên nghiệp để xác định các cơ hội tiềm tàng cho CAPA. Bước này bao gồm sự so sánh các tài liệu trong cùng 1 kiểu đã nhận được từ các đơn vị khác nhau cũng như là việc so sánh các tài liệu của các kiểu khác nhau liên quan tới cùng 1 trường hợp giống nhau.
- Phân tích những sự cải tiến tiềm tàng. Cần cố gắng xác định:
 - Những kiểu và những mức của thiệt hại từ các lỗi được định danh
 - Lí do gây lỗi. Các lí do tiêu biểu là những chỉ dẫn và thủ tục không tuân theo công việc, kiến thức chuyên môn không đủ, quá giờ và/hoặc ngân sách quá bấp bênh do ước lượng không thực tế, và việc thiếu kinh nghiệm với những công cụ phát triển mới.
 - Việc ước lượng quy mô của tổ chức-các lỗi tiềm tàng của từng kiểu. Thông tin này cần thiết để ước lượng tổng thiệt hại và xác định mức độ ưu tiên cho từng trường hợp lỗi.
- Tạo ra phản hồi trong nội dung và sự hợp thức hóa của thông tin nhận được từ những nguồn thông tin đã được chỉ rõ.

9.3.4. Phát triển các giải pháp và thực thi

Phát triển các giải pháp Giải pháp để xác định nguyên nhân gây ra lỗi hệ thống phần mềm yêu cầu:

- Loại bỏ tái phát của các loại lỗi được phát hiện
- Đóng góp vào cải thiện hiệu quả của sản xuất và tạo năng suất cao hơn lịch trình ngắn hơn. Một số hướng dẫn cho các giải pháp thường được chọn:
 - Cập nhật các thủ tục liên quan. Các thay đổi có thể tham khảo một số thủ tục khác, từ những người cụ thể liên quan đến giai đoạn phát triển phần mềm hoặc bảo trì (ví dụ như, thay đổi trong nhận xét phần mềm, các thay đổi thủ tục hợp đồng trong giao dịch với điều khoản đề xuất cho các dự án nhỏ) để các thủ tục chung (ví dụ như các thay đổi của nhân viên tuyển dụng, các thay đổi tối đa và tối thiểu số của người tham gia nhận xét một thiết kế chính thức).
 - Thay đổi trong thực hành, bao gồm cả công việc cập nhật các hướng dẫn có liên quan (nếu tồn tại).
 - Chuyển đến một công cụ phát triển hiệu quả hơn và ít phát hiện ra lỗi.
 - Cải tiến phương pháp báo cáo, bao gồm các thay đổi trong nội dung báo cáo, tần số của báo cáo và công việc báo cáo. Đây là hướng cải thiện triển vọng cho việc xác định các lỗi và phần mềm hệ thống sớm phát hiện ra lỗi để mang lại kết quả đáng kể trong việc giảm thiệt hại.
 - Các sáng kiến đào tạo, đào tạo lại hoặc cập nhật nhân viên. Đây là hướng thực hiện chỉ trong trường hợp khi cùng một thiếu sót đang đào tạo được tìm thấy trong một vài teams - Thực hiện một quá trình CAPA Việc triển khai thực hiện các giải pháp CAPA dựa vào đúng các hướng dẫn và thường xuyên đào tạo, nhưng hầu hết trên tất cả các hợp tác của các đơn vị có liên quan và riêng lẻ. Vì vậy, việc triển khai thực hiện thành công mục tiêu, đòi hỏi nhân viên, các thành viên được thuyết phục thích đáng đối với các giải pháp đề xuất. Nếu không có hợp tác, sự đóng góp của một CAPA có thể không được quyết định.

9.3.5. Tổ chức các hành động phòng ngừa và sửa lỗi

Sự thể hiện đúng đắn của những hoạt động CAPA này phụ thuộc vào sự tồn tại của các đơn vị tổ chức trung tâm lâu dài cũng như rất nhiều người tham gia trong đội ad

-hoc. Nhân tố này nhìn chung được biết đến là Corrective Action Board (CAB), mặc dù nó có thể có những tên gọi khác nhau trong những tổ chức khác nhau, nhưng nó đều đảm nhiệm các công việc sau đây:

- Thu thập bản ghi CAPA từ các nguồn khác nhau.
- Hiện thị các thông tin đã được thu thập.
- Chỉ định toàn bộ đội CAPA ad-hoc tham gia vào công việc đã được đưa ra hoặc dẫn dắt một số các team đó.
- Xúc tiến việc thực thi CAPA trong các đơn vị, dự án....
- Liên tục thu thập thông tin, phân tích dữ liệu, quá trình được thực hiện bởi các đội ad-hoc và sự thực thi cũng như kết quả của các phương thức capa đã được cải tiến.

BÀI 10. BÀI TẬP VỀ CHECKLIST

Bài 1: Chọn ứng dụng và áp dụng checklist để đánh giá về khía cạnh giao diện

GUI Review Checklist

Project Code

Version of the work product

Reviewer (s)

Review date

Work product' size

Effort spent on review (man-hour)

Question	Yes	No	N/A	Note	Priority	Severity
Is graphical design of GUI reviewed or accepted by customer? If not					Mandatory	1
Has the waiver been stated?					Mandatory	1
Is one button set to be default?						4
Are the controls ordered according to normal behavior (Microsoft Style)?					Optional	4
Are all labels aligned uniformly?					Optional	4
Are text areas aligned uniformly?					Optional	4
Are there any spelling mistakes?					Mandatory	4
Are all non-enterable text areas of the same color (gray)?					Optional	4
Is the tab sequence correct?					Mandatory	4
Is cursor becomes hourglass when doing background processing?					Optional	3
Does cursor return to pointer after the task?					Optional	3
Do all fields match data length?					Optional	4
Do enterable fields accept entry only for the given lengths of String e.g. char (30)?					Optional	4
Are abbreviations used? If yes					Optional	4
Are they understandable for end users?					Optional	4
Are they consistently used across all screens?					Optional	4
Are both mouse and key strokes handled?					Optional	3
Are the color combinations used on screens consistent?					Optional	4
Is the usage of the screens intuitive?					Optional	4
If hotkeys or accelerator keys used, is it consistent?					Optional	4
If yes to the above, has this been tested?					Mandatory	3
Are error messages displayed in non-technical terms and are understandable to the user?					Optional	3
Are the error messages informative and not blaming the user?					Mandatory	4
Before an action that affects the system (such as a delete) is performed, is the user asked for confirmation?					Mandatory	2

Are dates and time formats across the system consistent?					Optional	4
Are number formats consistent?					Optional	4
If Tooltips are used, are they helpful or meaningful?					Optional	4
Are screens and menu mapped correctly?					Mandatory	2

* **Comment**

* **Suggestion**

- [] - Pass
- [] - Review again
- [] - Other

Bài 2. Chọn Test Case và áp dụng checklist sau để đánh giá chất lượng:

Test Case Review Checklist

Project Code:

Version of the work product:

Reviewer(s):

Review date:<DD-MMM-DD>

Number of test cases:

Effort spent on review (person-hour):

Question	Yes	No	N/A	Note	Priority
DOCUMENT CONTROL					
Verify whether document control procedures have been followed by checking for the following:					
<i>Does the title page contain the document name, version number, release date and issued date?</i>					
<i>Does the header and footer correctly specify the name, and version of the document?</i>					
<i>Does the page – numbering scheme indicate the total number of pages in the document?</i>					
<i>Is the history traceable?</i>					Mandatory
<i>Does it include list of reference documents?</i>					
<i>Are all documents checked Spelling and Grammar in MS Word or similar tools?</i>					
TEST CASE REVIEW CHECKLIST					
Do test cases cover all requirements?					Mandatory
Do test cases cover all Testing types described in Test plan?					Mandatory
Has each test case been assigned a test case identifier?					Mandatory
Does each test case specify:					
<i>Actions</i>					Mandatory

<i>Test condition</i>					Mandatory
<i>Expected result</i>					Mandatory
Have the expected results been recorded in detail?					Mandatory
Do test cases for field validations, record validations and database updates include the following?					
<i>Valid conditions</i>					Mandatory
<i>Invalid conditions</i>					Mandatory
<i>Boundary or unusual conditions</i>					Mandatory
Is error message/message code specified in expected result for abnormal cases?					
Do the test cases for reports include the test data along with the expected output?					
Have the inter test case dependencies been described?					
Have all requested environments been specified ?					Mandatory
Are pre-conditions for the test specified?					
Are all metrics of test cases meet standards?					
<Add more rows if needed>					

*** Suggestion**

- [] - Pass
- [] - Review again
- [] - Other

BÀI 11. QUẢN LÝ CẤU HÌNH

11.1. Quản lý cấu hình

11.1.1. Các thành phần cấu hình phần mềm

Tài liệu thiết kế

- Kế hoạch phát triển phần mềm (SDP)
- Tài liệu yêu cầu hệ thống
- Tài liệu yêu cầu phần mềm (SRD)
- Thông số thiết kế giao diện
- Tài liệu thiết kế ban đầu (PDD)
- Tài liệu thiết kế phê bình (CDD)
- Mô tả cơ sở dữ liệu
- Kế hoạch kiểm thử phần mềm (STP)
- Quy trình kiểm thử phần mềm (STPR)
- Báo cáo kiểm thử phần mềm (STR)
- Hướng dẫn sử dụng phần mềm
- Hướng dẫn bảo trì phần mềm
- Kế hoạch cài đặt phần mềm (SIP)
- Các yêu cầu bảo trì phần mềm (bao gồm các báo cáo vấn đề)
- Các yêu cầu thay đổi phần mềm (SCRs) và các lệnh thay đổi phần mềm (SCOs)
- Tài liệu mô tả phiên bản (VDD)

Mã phần mềm

- Mã nguồn
- Mã đối tượng
- Phần mềm mẫu

Hồ sơ dữ liệu

- Các trường hợp thử nghiệm và kịch bản thử nghiệm
- Tham số, mã số, vv

Các công cụ phát triển phần mềm

(Các phiên bản được áp dụng trong giai đoạn phát triển và bảo trì)

- Trình biên dịch và gỡ rối
- Máy phát điện ứng dụng
- Các công cụ CASE

11.1.2. Quản lý cấu hình phần mềm

Các hoạt động phát triển phần mềm và bảo trì thường xuyên liên quan đến các hoạt động chuyên sâu để sửa đổi phần mềm để tạo các phiên bản và bản phát hành mới.

Các hoạt động này được thực hiện trong toàn bộ thời gian dịch vụ phần mềm (thường kéo dài vài năm) để đối phó với những chỉnh sửa cần thiết, thích ứng với các yêu cầu cụ thể của khách hàng, cải tiến ứng dụng, v.v ... Các thành viên khác trong nhóm thực hiện các hoạt động này đồng thời, mặc dù họ có thể thực hiện các hoạt động khác nhau. Do đó, những nguy hiểm nghiêm trọng phát sinh, cho dù xác định sai các phiên bản hoặc bản phát hành, mất hồ sơ mô tả các thay đổi được thực hiện, hoặc mất tài liệu. Do đó có thể gây ra lỗi.

Quản lý cấu hình xử lý các mối nguy này bằng cách giới thiệu các quy trình để kiểm soát quá trình thay đổi. Các thủ tục này liên quan đến việc chấp thuận các thay đổi, ghi lại các thay đổi đã thực hiện, phát hành các phiên bản và phiên bản phần mềm mới, ghi lại phiên bản và thông số phát hành của phần mềm được cài đặt trong mỗi trang và ngăn ngừa bất kỳ thay đổi nào. Các phiên bản và bản phát hành được phê duyệt sau khi được phát hành. Hầu hết các hệ thống quản lý cấu hình đều sử dụng các công cụ trên máy vi tính để hoàn thành nhiệm vụ của họ. Các hệ thống trên máy vi tính này cung cấp các phiên bản cập nhật và thích hợp của phần mềm được cài đặt cho mục đích phát triển hoặc chỉnh sửa thêm. Các thủ tục cấu hình phần mềm thường cho phép một quản trị viên hoặc một ủy ban quản lý cấu hình để quản lý tất cả các hoạt động quản lý cấu hình cần thiết

11.1.3. Kiểm soát sự thay đổi phần mềm

Quản lý thay đổi phần mềm điều khiển quá trình giới thiệu các thay đổi chủ yếu bằng cách thực hiện như sau:

- Xem xét yêu cầu thay đổi và phê duyệt việc thực hiện các yêu cầu thích hợp.
- Đảm bảo chất lượng của mỗi phiên bản phần mềm mới trước khi nó hoạt động.

11.1.3.1. Phê duyệt để thực hiện các thay đổi được đề xuất

Một khi hệ thống phần mềm bắt đầu hoạt động trở lại, vấn đề chỉ là thời gian trước khi các đề xuất thay đổi bắt đầu phát triển. Những sáng kiến này có thể liên quan đến một hoặc nhiều SCI. Để phối hợp những nỗ lực đầu tư và đảm bảo rằng những thay đổi theo dự án hoặc các ưu tiên của khách hàng, cơ quan có thẩm quyền phải phân tích các yêu cầu và đưa ra các quyết định cần thiết.

Các yếu tố ảnh hưởng đến quyết định có thực hiện thay đổi được đề xuất bao gồm:

- Đóng góp dự kiến của đề xuất thay đổi
- Khả năng của sự thay đổi
- Ảnh hưởng của sự thay đổi đề xuất về thời gian biểu của dự án, mức độ dịch vụ, v.v.
- Nỗ lực cần thiết để biến đổi hoạt động
- Các nỗ lực đảm bảo chất lượng phần mềm
- Ước tính các nguồn lực nghề nghiệp và chi phí thực hiện thay đổi.

11.1.3.2. Bảo đảm chất lượng thay đổi phần mềm

Trong khi những nỗ lực thay đổi được hướng tới một hoặc nhiều SCI, người dùng trải nghiệm những thay đổi gián tiếp thông qua việc áp dụng phiên bản sửa đổi của hệ thống phần mềm. Mục tiêu đảm bảo chất lượng phần mềm là đảm bảo rằng

Chất lượng của phiên bản hệ thống phần mềm mới không thấp hơn phiên bản trước đó.

Các nỗ lực đảm bảo chất lượng được yêu cầu ở hai cấp:

- Bảo đảm chất lượng của mỗi SCIs thay đổi
- Bảo đảm chất lượng của toàn bộ phiên bản hệ thống phần mềm mới (bao gồm thay đổi SCI).

Đảm bảo chất lượng của SCIs đã thay đổi

Điều này đòi hỏi phải chuẩn bị một bài đánh giá và kế hoạch thử nghiệm ở mức độ phù hợp với đặc điểm của sự thay đổi. Như đã đề cập trong các chương trước, điều quan trọng nhất là kiểm tra và thử nghiệm phải được tiến hành bởi những người kiểm tra chuyên nghiệp chứ không phải bởi nhà phát triển của SCI. Tiến trình rà soát và kiểm tra, sửa chữa và kiểm tra lại (kiểm tra hồi quy) các SCIs thay đổi dự kiến sẽ kết thúc với sự chấp thuận của họ.

Đảm bảo chất lượng của toàn bộ phiên bản hệ thống phần mềm mới

Một phiên bản mới của phần mềm được coi là đã được hoàn thành sau khi SCI thay đổi thay thế các SCI cũ. Mặc dù người ta có thể mong đợi phiên bản mới của hệ thống phần mềm hoạt động hoàn hảo và chắc chắn tốt hơn so với phiên bản cũ, nhiều phiên bản mới, đặc biệt là các hệ thống phần mềm phức tạp, thực sự thất bại. Những thất bại của hệ thống thường xảy ra như là kết quả của thiệt hại được thực hiện cho các giao diện giữa SCIs thay đổi và SCIs khác không thay đổi và không kiểm tra lại bởi vì họ không bị ảnh hưởng bởi những thay đổi được thực hiện. Trừ khi toàn bộ phiên bản mới, hoặc ít nhất là toàn bộ phần mềm có thể bị ảnh hưởng, được kiểm tra để xác định các khiếm khuyết giao diện không mong muốn, triển vọng là ít ỏi rằng hệ thống phần mềm sẽ hoạt động đúng.

11.1.4. Kiểm soát quản lý cấu hình phần mềm

SCM bao gồm việc thực hiện nhiều nhiệm vụ khác nhau do cơ quan SCM, CCB và nhiều cơ quan khác tham gia vào việc phát triển và bảo trì phần mềm. Tất cả các nhiệm vụ tương ứng được định nghĩa trong các thủ tục SCM. Kiểm toán SCM được thực hiện bởi cơ quan SCM và CCB để

Kiểm soát tuân thủ các quy trình của SCM. Kiểm toán SCM có thể được kết hợp với các vấn đề chất lượng nội bộ (xem chương 27), và dự kiến sẽ bắt đầu cập nhật và thay đổi các thủ tục và hướng dẫn của SCM. Do đó, kiểm toán SCM kiểm tra xem các nhiệm vụ này có được thực hiện như thế nào cho mẫu yêu cầu thay đổi, SCI và

các phiên bản cấu hình phần mềm. Kiểm toán SCM cũng có thể được thực hiện cho một mẫu các bản phát hành kế hoạch, như được chỉ định trong SCMP. Tuy nhiên, mặc dù chúng ta có thể mong đợi kiểm toán của SCM cung cấp thông tin liên quan đến mức độ tuân thủ các quy trình của SCM (bao gồm các lỗi điển hình của các thủ tục này) nhưng chúng không thể phục vụ như các công cụ thực thi tuân thủ. Sau đây là danh sách các thông tin kiểm soát điển hình của kiểm soát SCM nhằm mục đích phát hiện và truyền tải đến ban quản lý:

- Phần trăm thay đổi không được chấp thuận trong hệ thống trong quá trình phát triển hoặc vận hành.
- Phần trăm các SCO không được thực hiện theo hướng dẫn và không tuân thủ đầy đủ các thủ tục.
- Phần trăm đánh giá thiết kế và phần mềm kiểm tra các SCIs thay đổi mà không được thực hiện theo thủ tục có liên quan.
- Phần trăm các SCO đã hoàn thành đúng tiến độ.
- Tỷ lệ trường hợp SCI bị ảnh hưởng bởi những thay đổi chưa được kiểm tra, với một số thay đổi cần thiết không được thực hiện.
- Phần trăm các SCIs và các phiên bản cấu hình phần mềm mới.
- Phần trăm các bản cài đặt phần mềm đã được ghi lại hợp lệ.
- Tỷ lệ phần trăm trường hợp không truyền được tất cả thông tin liên quan đến phiên bản cho khách hàng.
- Số trường hợp được ghi chép hàng năm khi các cơ chế phối hợp làm việc của SCI thất bại (tức là, không ngăn các nhóm khác nhau đồng thời giới thiệu những thay đổi trong cùng một SCI).

11.1.5. Các công cụ máy tính quản lý cấu hình phần mềm

Các công cụ SCM đã được tung ra thị trường trong nhiều năm. Những công cụ trên máy vi tính này khác nhau về mức độ toàn diện, tính linh hoạt của ứng dụng và dễ sử dụng. Các công cụ toàn diện hơn có thể cung cấp hầu hết hoặc hầu hết các dịch vụ thông tin SCM được liệt kê trong Phần 18.5.

Người ta hy vọng rằng một công cụ trên máy vi tính sẽ có thể đáp ứng yêu cầu về độ chính xác và đầy đủ của thông tin, và với mức độ sẵn có yêu cầu (đo bằng thời gian phản hồi từ yêu cầu cung cấp thông tin).

Các công cụ SCM trên máy tính cũng vận hành các cơ chế điều phối công việc trên những thay đổi của SCI và ngăn các nhóm khác nhau đồng thời giới thiệu những thay đổi trong cùng một SCI.

Lợi ích bổ sung của việc sử dụng một hệ thống SCM trên máy vi tính là mức độ bảo mật cao mà nó có thể cung cấp:

- Nó bảo đảm các phiên bản mã và phiên bản tài liệu bằng cách bảo vệ chúng khỏi mọi thay đổi, xóa bỏ và các thiệt hại khác.

- Nó kích hoạt các thủ tục sao lưu cần thiết để lưu trữ tệp SCM an toàn.

Các công cụ nâng cao hiện nay được đặc trưng bởi khả năng đầu vào dễ dàng hơn, sự hợp tác của các nhóm hỗ trợ SCM hoạt động trong các môi trường phát triển khác nhau, bao gồm các nhóm phân phối theo địa lý và cung cấp nhiều lựa chọn báo cáo khác nhau.

11.2. Kiểm soát tài liệu

11.2.1. Các tài liệu kiểm soát và bản ghi chất lượng

11.2.1.1. Định nghĩa và mục tiêu

Trước tiên chúng ta hãy thảo luận về tài liệu kiểm soát và hồ sơ chất lượng, như được định nghĩa trong Khung 11.1

Khung 11.1 Tài liệu kiểm soát và bản ghi chất lượng - định nghĩa

Hồ sơ chất lượng

Một bản ghi chất lượng là một loại tài liệu kiểm soát đặc biệt. Đây là tài liệu hướng tới khách hàng có thể được yêu cầu chứng minh sự tuân thủ đầy đủ các yêu cầu của khách hàng và hoạt động hiệu quả của hệ thống đảm bảo chất lượng phần mềm trong suốt quá trình phát triển và bảo trì.

Tài liệu kiểm soát

Một tài liệu hiện tại đang sống còn hoặc có thể trở nên quan trọng cho việc phát triển và duy trì hệ thống phần mềm cũng như để quản lý các mối quan hệ hiện tại và tương lai với khách hàng. Do đó, việc chuẩn bị, lưu trữ, thu hồi và thải bỏ được kiểm soát bằng các thủ tục tài liệu. Mục tiêu chính để quản lý tài liệu kiểm soát là:

- Đảm bảo chất lượng của tài liệu.
- Đảm bảo hoàn thiện kỹ thuật và tuân thủ các quy trình và hướng dẫn cấu trúc tài liệu (sử dụng các mẫu, ký kết hợp lệ, vv).
- Để đảm bảo tính sẵn có trong tương lai của các tài liệu có thể được yêu cầu để bảo trì hệ thống phần mềm, phát triển tiếp theo, hoặc phản hồi các khiếu nại của khách hàng (dự kiến) trong tương lai.
- Hỗ trợ điều tra các nguyên nhân thất bại phần mềm và phân công trách nhiệm như là một phần của các hành động khắc phục và các hành động khác.

Khung 11.2 trình bày tổng quan các loại tài liệu có thể được phân loại là tài liệu có kiểm soát. Việc kiểm tra danh sách tài liệu cho thấy rằng một số lượng tốt các tài liệu được kiểm soát có thể được phân loại là hồ sơ chất lượng. Mức độ của danh

Hồ sơ dự án

- Báo cáo đánh giá hợp đồng
- Biên bản cuộc họp đàm phán hợp đồng
- Hợp đồng phát triển phần mềm
- Hợp đồng bảo trì phần mềm
- Hợp đồng thầu phụ phát triển phần mềm
- Kế hoạch phát triển phần mềm

Tài liệu cuộc sống của dự án

- Tài liệu yêu cầu hệ thống
- Tài liệu yêu cầu phần mềm
- Tài liệu thiết kế sơ bộ
- Tài liệu thiết kế phê bình
- Mô tả cơ sở dữ liệu
- Kế hoạch kiểm thử phần mềm
- Báo cáo đánh giá thiết kế
- Các bản ghi tiếp theo của các mục hành động đánh giá thiết kế
- Quy trình kiểm thử phần mềm
- Báo cáo thử nghiệm phần mềm
- Hướng dẫn sử dụng phần mềm
- Hướng dẫn bảo trì phần mềm
- Kế hoạch cài đặt phần mềm
- Tài liệu mô tả phiên bản
- Yêu cầu thay đổi phần mềm
- Đơn đặt hàng thay đổi phần mềm
- Yêu cầu bảo trì phần mềm
- Báo cáo dịch vụ bảo trì

sách và thành phần của nó khác nhau giữa các tổ chức và phụ thuộc vào đặc điểm của khách hàng ngoài các gói phần mềm. Các hợp đồng cho các dự án phần mềm "custom-made" quy mô lớn thường đòi hỏi phải có danh sách các tài liệu được kiểm soát khá khác so với các gói phần mềm COTS.

Khung 11.2 Các tài liệu được kiểm soát điển hình (bao gồm hồ sơ chất lượng)

Tài liệu cơ sở hạ tầng SQA

- Thủ tục SQA
- Thọ viện mẫu
- SQA tạo thọ viện
- Biên bản cuộc họp CAB

Tài liệu quản lý chất lượng phần mềm

- Các báo cáo tiến độ
- Báo cáo số liệu phần mềm

Tài liệu kiểm toán hệ thống SQA

- Báo cáo đánh giá quản lý
- Biên bản cuộc họp tổng kết quản lý
- Báo cáo kiểm toán chất lượng nội bộ
- Báo cáo kiểm toán chứng nhận bên ngoài SQA

Tài liệu khách hàng

- Hồ sơ dự thầu phần mềm dự án
- Yêu cầu thay đổi phần mềm của khách hàng
- Biên bản đánh giá của nhà thầu phụ

Các loại tài liệu được liệt kê trong Khung 11.2 được tạo ra trong quá trình thực hiện một loạt các quy trình SQA, nhưng chỉ một số ít:

- Quy trình hợp đồng và đàm phán
- Quá trình phát triển
- Quá trình thay đổi phần mềm
- Dịch vụ bảo trì

- Chỉ số chất lượng phần mềm
- Kiểm toán chất lượng nội bộ.

11.2.2. Danh sách các tài liệu được kiểm soát

Chìa khóa để quản lý các tài liệu được kiểm soát (bao gồm hồ sơ chất lượng) là danh sách các loại tài liệu được kiểm soát. Xây dựng hợp lý danh sách này dựa trên việc thành lập một thẩm quyền để thực hiện khái niệm, cho dù được thể hiện trong một người hay một ủy ban. Cụ thể, cơ quan này có trách nhiệm:

- Quyết định loại tài liệu nào sẽ được phân loại là tài liệu được kiểm soát và loại tài liệu kiểm soát nào được phân loại là hồ sơ chất lượng.
- Quyết định xem mức độ kiểm soát có phù hợp với từng loại tài liệu được phân loại như một tài liệu có kiểm soát.
- Sau khi tuân thủ danh mục các loại tài liệu kiểm soát. Chủ đề này có thể được đưa vào kế hoạch kiểm toán chất lượng nội bộ (xem Chương 26).
- Phân tích những phát hiện tiếp theo và bắt đầu cập nhật, thay đổi, di chuyển và bổ sung cần thiết vào danh sách các loại tài liệu được kiểm soát.

Hầu hết các loại tài liệu kiểm soát là tài liệu được tạo ra nội bộ bởi chính tổ chức. Tuy nhiên, một số lượng đáng kể các loại tài liệu bên ngoài, chẳng hạn như tài liệu hợp đồng và biên bản cuộc họp chung, cũng thuộc loại này.

11.2.3. Chuẩn bị, phê chuẩn, lưu trữ và thu hồi tài liệu kiểm soát

Các yêu cầu về tài liệu liên quan đến việc tạo ra một sách các tài liệu mới hoặc sửa đổi một tập trung tài liệu hiện có về sự hoàn chỉnh, cải thiện khả năng đọc và sẵn có. Các yêu cầu này được thực hiện trong các tài liệu:

- Cấu trúc
- Phương pháp nhận dạng
- Định hướng tiêu chuẩn và thông tin tham khảo.

Cấu trúc của tài liệu có thể được miễn phí hoặc được xác định bởi một khuôn mẫu. Mẫu và đóng góp của họ cho chất lượng phần mềm được thảo luận trong mục 15.1. Một phương pháp xác định được đưa ra để cung cấp cho mỗi tài liệu, phiên bản và sửa đổi với một bản sắc độc đáo. Phương pháp thường đòi hỏi ký hiệu của (A) các hệ thống phần mềm hoặc tên sản phẩm hoặc số, (b) các tài liệu (type) mã và (c) các phiên bản và số sửa đổi. Phương pháp này có thể thay đổi đối với các loại văn bản khác nhau.

Cũng có thể yêu cầu định hướng và thông tin tham khảo của tài liệu. Thông tin định hướng và thông tin tham khảo hỗ trợ tiếp cận các tài liệu yêu cầu trong tương lai

bằng cách cung cấp thông tin về nội dung của tài liệu và tính phù hợp với nhu cầu của người sử dụng trong tương lai. Tùy thuộc vào loại tài liệu, thường yêu cầu một tỷ lệ lớn hơn hoặc nhỏ hơn của các mục thông tin sau:

- Tác giả của tài liệu
- Ngày hoàn thành
- Những người đã phê duyệt tài liệu, bao gồm (các) vị trí được tổ chức
- Ngày phê duyệt
- Chữ ký của tác giả và (những) người đã chấp thuận nó
- Các mô tả về những thay đổi được giới thiệu trong bản phát hành mới
- Danh sách các phiên bản cũ và bản sửa đổi
- Danh sách lưu hành
- Hạn chế bảo mật.

Các thủ tục tài liệu có liên quan và hướng dẫn làm việc liên quan đến giấy tờ cũng như các tài liệu điện tử (ví dụ, các ứng dụng e-mail và intranet).

BÀI 12. THẢO LUẬN VỀ QUẢN LÝ CẤU HÌNH PHẦN MỀM

Hãy xem xét những tình huống này, được rút ra từ các dự án khác nhau.

- Khách hàng đã yêu cầu hai thay đổi khác nhau. Người quản lý dự án đã phân công Rao thực hiện một yêu cầu, và cái còn lại phân cho Meera. Cả hai đã phải sửa đổi mã cho mô-đun X. Khi Meera hoàn thành sửa đổi của mình, cô lưu trữ một tập tin cho X, vô tình ghi đè lên các thay đổi Rao đã thực hiện trước đó một ngày.

- Thứ Sáu đã là thời hạn (deadline) để giao một mô-đun mà nhóm gồm ba thành viên đang phát triển mã. Việc tích hợp mã (đã được kiểm thử đơn vị) đã được lên kế hoạch cho hai ngày cuối cùng. Vào đêm thứ Ba, Subbu - nhà phát triển của một số chức năng then chốt - rời thành phố để tham dự một công việc khẩn cấp. Ngày hôm sau, người lãnh đạo mô-đun và các thành viên của nhóm nghiên cứu đã dành nhiều giờ tìm kiếm các tập tin của Subbu. Chúng được quản lý để xác định một số tập tin đang chứa các chức năng mà Subbu đang phát triển, nhưng họ đã tìm thấy nhiều phiên bản của các tập tin này. Một trong những thành viên trong nhóm đã phải làm việc về vấn đề này suốt cuối tuần. Bắt đầu từ một số phiên bản của các chương trình của Subbu, anh ta đã phát triển và kiểm thử đơn vị, làm lại hầu hết các công việc mà Subbu gần như đã hoàn tất trước khi rời khỏi. Mô-đun cuối cùng đã bị giao trễ ba ngày.

- Nhóm của Srinath đã có tinh thần làm việc tốt. Họ đã hoàn thành việc phát triển đúng hạn, và kiểm thử cuối cùng đã cho thấy không có lỗi. Phần mềm này được giao cho khách hàng. Nhưng ngày hôm sau, Srinath nhận được e-mail tức giận từ những người sử dụng và khách hàng, báo về các vấn đề trong phần mềm. Sau nỗ lực điên cuồng của nhóm làm việc, nguyên nhân đã được tìm thấy: Phiên bản phát hành của phần mềm đã chứa một phiên bản cũ của một thành phần (component) then chốt.

- Những vấn đề rút ra:

Những câu chuyện như thế này có thể được tìm thấy trong hầu hết các công ty. Quản lý cấu hình (CM - Configuration management) - còn được gọi là quản lý cấu hình phần mềm (SCM - software configuration management) - là một khía cạnh của quản lý dự án để tập trung vào việc kiểm soát có hệ thống các thay đổi để mà những vấn đề như bên trên không xảy ra.

- + Áp dụng quản lý cấu hình trong một dự án
 - Quy ước để đặt tên và tổ chức các tập tin (conventions for naming and organization of files)
 - Kiểm soát phiên bản (version control)
 - Khả năng theo dõi dấu vết nguồn gốc yêu cầu thay đổi (change request traceability)
 - Kiểm soát truy cập (access control)
 - Các thủ tục hòa giải (reconciliation procedures)
 - Các chương trình đăng nhập thay đổi (modification login programs)

Bài 13. CÁC THÀNH PHẦN QUẢN LÝ CHẤT LƯỢNG PHẦN MỀM

13.1. Điều khiển tiến độ dự án

13.1.1 Các thành phần điều khiển tiến độ dự án

Quản lý tiến độ dự án có một mục tiêu tức thời : sớm phát hiện ra sự kiện bất bình thường. Việc phát hiện đó thúc đẩy thời gian bắt đầu của đáp ứng giải quyết vấn đề. Thông tin chính xác trong quản lý tiến độ cũng như những thành công và những thất bại cũng phục vụ cho những đối tượng lâu dài: Những hành động đúng đắn đầu tiên.

Thành phần chính của quản lý tiến độ dự án:

- Điều khiển những hoạt động quản lý rủi ro
- Quản lý lịch trình dự án
- Quản lý tài nguyên dự án
- Quản lý ngân sách dự án

13.1.2 Điều khiển tiến độ của các dự án nội bộ và các thành phần bên ngoài

Quản lý tiến trình được khởi tạo để cung cấp cho người quản lý cái nhìn toàn diện về tất cả hoạt động phát triển phần mềm được thực hiện trong tổ chức. Tuy nhiên, trong hầu hết các tổ chức, quản lý dự án quy định, với các lý do khác nhau, có sự giới hạn về tầm nhìn của phát triển phần mềm bên trong và thậm chí còn bị giới hạn nhiều hơn của tiến trình được tạo bởi những người tham gia bên ngoài.

13.1.3. Thực thi kiểm soát tiến độ dự án

Điều khiển tiến độ dự án thường dựa trên thủ tục được chỉ ra:

- Sự phân chia trách nhiệm cho sự thực thi của các công việc điều khiển tiến độ mà nó thích hợp với thành phần của dự án, bao gồm các chuẩn mực(size):
- Cá nhân hoặc đơn vị quản lý có trách nhiệm thực thi công việc điều khiển tiến độ.
- Tính thường xuyên của các bản báo cáo yêu cầu từ mỗi đơn vị dự án hoặc mức độ hành chính.
- Các tình huống yêu cầu những người chỉ đạo dự án báo cáo trực tiếp đến người quản lý.
- Các tình huống yêu cầu người quản lý thấp hơn báo cáo trực tiếp cho người quản lý cao hơn.

- Quản lý kiểm tra của tiến độ dự án, giải quyết những vấn đề chính: (1) Làm thế nào có những báo cáo tiến độ tốt được đưa ra bởi những người chỉ đạo dự án và từ người quản lý thấp hơn đến người quản lý cao hơn, và (2) điều khiển quản lý chi tiết là tích cực để bắt đầu.

Trong những tổ chức phát triển phần mềm lớn, điều khiển tiến độ dự án có thể quản lý được những trình độ của những người quản lý khác nhau, như là quản lý lĩnh vực phần mềm, quản lý sự phân chia phần mềm và quản lý cao nhất (top management). Mặc dù mỗi mức độ đòi hỏi định nghĩa chế độ điều khiển tiến độ dự án, cái mà phản ánh tham số đầy đủ để ước định tiến độ của dự án từ vị trí riêng biệt đó, phối hợp giữa các mức độ khác nhau là có tính bắt buộc để điều khiển tiến độ có hiệu quả.

Toàn bộ bản báo cáo sẽ truyền thông tin được lựa chọn từ mức thuộc ban quản lý thấp nhất - báo cáo tiến độ định kỳ của người chủ dự án - nó tổng kết lại trạng thái của những rủi ro dự án, lập lịch dự án và tính hữu ích của các tài nguyên, đây cũng là 3 thành phần đầu tiên của quá trình điều khiển tiến độ, Người chủ dự án dựa trên báo cáo tiến độ của mình được tập hợp từ người chủ của các nhóm.

13.1.4. Các công cụ kiểm soát tiến độ phần mềm

Tổ chức các công cụ cho việc điều khiển tiến độ dự án là một nhu cầu dễ hiểu cho việc kích thích ngày càng tăng và độ phức tạp của dự ánvà lợi ích mà chúng mang lại cho họ trên những mặt khác. Những công cụ quản lý dự án toàn diện đã có sẵn trên thị trường nhiều năm có thể phục vụ hầu hết các thành phần điều khiển của những dự án phần mềm khá có hiệu quả. Phần lớn các gói đa dụng áp dụng cho việc phân tích PERT/CPM vì vậy mà những bản ghi kết quả giữ sự tương tác giữa các hành động và việc phê bình của mỗi hành động bên trong một tài khoản. Đây là những gói thông thường thích ứng với những trường hợp đặc biệt vì sự đa dạng lớn của những tùy chọn mà họ đưa ra.

Ví dụ về những dịch vụ mà những công cụ (được máy tính hóa) có thể cung cấp:

- Điều khiển những hoạt động quản lý rủi ro:
 - Danh sách rủi ro phần mềm theo từng loại và những kỳ hạn giải pháp được lập kế hoạch của chúng.
 - Danh sách ngoại lệ của những rủi ro phần mềm – việc vượt quá kỳ hạn giải

- pháp có thể làm ảnh hưởng tới việc hoàn thành dự án đúng kỳ hạn.
- Điều khiển lịch biểu dự án:
 - Danh sách phân loại của các hành động trì hoãn
 - Danh sách phân loại của các hành động phê bình – những trì hoãn có thể, nếu không phải sửa chữa, ảnh hưởng tới việc hoàn thành dự án đúng kỳ hạn
 - Các hoạt động cập nhật lịch biểu sinh ra tùy thuộc vào các bản ghi tiến độ và sửa chữa độ đo ứng dụng cho các nhóm, các đơn vị phát triển, vv...
 - Danh sách phân loại các cột mốc trì hoãn
 - Các hoạt động cập nhật lịch biểu sinh ra tùy thuộc vào các bản ghi tiến độ và sửa chữa độ đo ứng dụng cho các nhóm, các đơn vị phát triển, vv...
 - Điều khiển tài nguyên dự án:
 - Kế hoạch cấp phát tài nguyên dự án cho các hoạt động và mô đun phần mềm, cho các nhóm, đơn vị phát triển, chỉ định những thời hạn, vv...
 - Sử dụng tài nguyên dự án bởi các thời kỳ hay được tích lũy như đã được chỉ rõ ở trên
 - Những ngoại lệ sử dụng tài nguyên dự án bởi các thời kỳ hoặc được tích lũy như đã chỉ rõ ở trên
 - Việc cập nhật những kế hoạch phân bổ lại nguồn tài nguyên dự án phát sinh tùy thuộc vào các bản ghi tiến độ và ứng dụng sửa chữa độ đo.
 - Điều khiển ngân sách dự án:
 1. Kế hoạch ngân sách dự án bởi các hoạt động và các mô đun phần mềm, cho các nhóm, các đơn vị phát triển, chỉ định những thời hạn, vv...
 2. Những báo cáo sử dụng ngân sách dự án bởi các thời kỳ hoặc được tích lũy như đã chỉ rõ ở trên
 3. Sự chênh lệch trong việc sử dụng ngân sách dự án bởi các thời kỳ hoặc được tích lũy như đã chỉ rõ ở trên.
 4. Việc cập nhật kế hoạch ngân sách dự án phát sinh tùy thuộc vào báo cáo tiến độ và sửa chữa độ đo ứng dụng, vv...

13.2. Độ đo chất lượng phần mềm

13.2.1. Các mục tiêu đo lường phần mềm và phân loại các độ đo

a. Mục tiêu đo lường phần mềm

Mục tiêu chính của đo lường chất lượng phần mềm

- Để thuận tiện cho việc điều khiển quản lý cũng như lập kế hoạch và thực thi của sự can thiệp quản lý thích hợp. Đạt được mục đích này dựa trên sự tính toán của độ đo đối với:
- Độ chênh lệch giữa sự thực thi chức năng (chất lượng) thực tế từ sự thực thi đã lập kế hoạch.
- Độ chênh lệch của thực hiện ngân sách và thời gian biểu thực tế từ sự thực thi đã lập kế hoạch.
- Để xác định trạng thái (situation) yêu cầu hoặc cho phép cải tiến quy trình phát triển hay bảo trì dưới dạng các hoạt động ngăn ngừa và sửa đổi được đưa ra trong suốt tổ chức. Đạt mục tiêu này dựa trên:

Tích lũy thông tin về độ đo đối với sự thực hiện của cả nhóm, đơn vị,...

Việc so sánh sẽ cung cấp thực tế cơ bản cho ứng dụng quản lý của độ đo và sự tiến bộ của SQA nói chung. Độ đo được sử dụng cho việc so sánh dữ liệu thực thi với chỉ dẫn (indicator), giá trị định lượng như sau:

- Xác định các chuẩn chất lượng phần mềm.
- Tập hợp mục tiêu chất lượng cho tổ chức và cá nhân.
- Thành tựu chất lượng trong năm trước.
- Thành tựu chất lượng của dự án trước.
- Mức độ chất lượng trung bình đạt được bởi các nhóm khác áp dụng cùng công cụ phát triển trong môi trường phát triển giống nhau.
- Thành tựu chất lượng trung bình của tổ chức.
- Thực tiễn công nghiệp cho sự thỏa mãn yêu cầu chất lượng.

b. Phân loại độ đo chất lượng phần mềm

Độ đo chất lượng phần mềm có thể chia thành một số nhóm. ở đây chúng tôi sử dụng một hệ thống 2 mức độ:

- Thứ nhất, phân loại danh mục phân biệt giữa vòng đời và các pha khác nhau của hệ thống phần mềm:
 - Độ đo quy trình, liên quan đến quá trình phát triển phần mềm
 - Độ đo Sản phẩm, liên quan đến bảo trì phần mềm.
- Thứ hai, phân loại danh mục đề cập đến các chủ đề về đo lường:
 - Chất lượng.
 - Thời gian biểu.
 - Hiệu quả (của lỗi và xóa bỏ các dịch vụ bảo trì).
 - Năng suất.

Các mục này được chia với với các nhóm riêng biệt.

Một số lượng lớn độ đo chất lượng phần mềm liên quan đến một trong hai cách đo lường cho kích thước hệ thống như sau:

- KLOC - thước đo cổ điển này đo kích cỡ của phần mềm bởi hàng ngàn dòng lệnh. Vì số lượng các dòng mã cần thiết cho lập trình một tác vụ đã quy định về cơ bản là khác nhau đáng kể với mỗi công cụ lập trình, phép đo này là cụ thể cho ngôn ngữ lập trình và công cụ phát triển được sử dụng. Việc áp dụng độ đo đó bao gồm KLOC được giới hạn trong các hệ thống phần mềm phát triển bằng cách sử dụng cùng một ngôn ngữ lập trình hay công cụ phát triển.
- Điểm chức năng – một phép đo các tài nguyên phát triển (nguồn lực con người) được yêu cầu để phát triển một chương trình, dựa trên chức năng đã xác định cho hệ thống phần mềm.

Các độ đo thỏa mãn khách hàng sẽ không được trình bày ở đây, các độc giả có thể tìm thấy phạm vi mở rộng của chủ đề này trong bài giảng marketing.

13.2.2. Các độ đo tiến trình

Các độ đo quy trình bao gồm:

- Các độ đo chất lượng quy trình phần mềm
- Các độ đo thời gian biểu của quy trình phần mềm
- Các độ đo tính hiệu quả loại bỏ lỗi

- Các độ đo năng suất tiến trình phần mềm
- Độ đo chất lượng quy trình phần mềm bao gồm:

❖ Độ đo mật độ lỗi.

Code	Tên	Công thức tính
CED	Mật độ lỗi code	$CED = \frac{NCE}{KLOC}$
DED	Mật độ lỗi phát triển	$DED = \frac{NDE}{KLOC}$
WCED	Mật độ lỗi code đánh trọng số	$WCED = \frac{WCE}{KLOC}$
WDED	Mật độ lỗi phát triển đã đánh trọng số	$WDED = \frac{WDE}{KLOC}$
WCEF	Lỗi code đã đánh trọng số cho mỗi điểm chức	$WCEF = \frac{WCE}{NFP}$
WDEF	Lỗi phát triển đánh trọng số cho mỗi điểm chức năng	$WDEF = \frac{WDE}{NFP}$

Từ khóa:

- NCE = số lỗi code được phát hiện trong code phần mềm bởi quá trình test và duyệt. Dữ liệu cho phép đo này được chọn lọc từ các báo cáo test và duyệt code.
 - KLOC = hàng nghìn dòng code
 - NDE = tổng số lỗi phát triển (thiết kế và code) được phát hiện trong quy trình phát triển phần mềm. Dữ liệu cho phép đo này được tìm ra trong rất nhiều báo cáo test rà soát code và thiết kế.
 - WCE = lỗi code đánh trọng số được phát hiện. Nguồn dữ liệu cho độ đo này là tương tự như nguồn của NCE.
 - WDE = tổng lỗi phát triển (thiết kế vào code) đánh trọng số được phát hiện trong phát triển của phần mềm. Các nguồn dữ liệu cho độ đo này giống với NDE.
 - NFP: số điểm chức năng được yêu cầu cho phát triển phần mềm. Nguồn gốc số điểm chức năng là sự khảo sát chuyên nghiệp của phần mềm liên quan.
- ❖ Độ đo tính nghiêm trọng của lỗi

Code	Tên	Công thức tính
ASCE	Trung bình nghiêm trọng lỗi code	$ASCE = \frac{WCE}{NCE}$
ASDE	Trung bình nghiêm trọng của lỗi phát triển	$ASDE = \frac{WDE}{NDE}$

Bảng Các độ đo tính nghiêm trọng của lỗi

❖ Độ đo thời gian biểu của quy trình phần mềm.

Code	Tên	Công thức tính
TTO	Tuân theo thời gian biểu	$TTO = \frac{MSOT}{MS}$
ADMC	Trung bình trễ hoàn thành mốc quan trọng	$ADMC = \frac{TCDAM}{MS}$

Bảng Các độ đo thời gian biểu tiến trình phần mềm

Từ khóa:

- MSOT= Những mốc quan trọng(milestones) đã hoàn thành đúng thời gian
- MS= tổng số mốc quan trọng
- TCDAM= tổng các trễ hoàn thành (hàng ngày, hàng tuần...) cho tất cả mốc quan trọng. Để tính toán phép đo này, các trễ đã báo cáo cho tất cả các mốc quan trọng liên quan được tổng kết lại. Các mốc quan trọng hoàn thành đúng thời gian hoặc trước lịch biểu được xem là trễ „O“. Một số chuyên gia xem các mốc quan trọng hoàn thành trước lịch biểu như các trễ “âm”. Chúng được xét để cân bằng hiệu quả (hiệu lực) của các trễ “dương”. Trong trường hợp này, giá trị của ADMC có thể thấp hơn giá trị đạt được theo độ đo đề nghị ban đầu.

❖ Độ đo hiệu quả loại bỏ lỗi.

Code	Tên	Công thức tính
DERE	Hiệu quả loại bỏ lỗi phát triển	$DERE = \frac{NDE}{NDE + NYF}$
DWERE	Hiệu quả loại bỏ lỗi phát triển có đánh	$DWERE = \frac{WDE}{WDE + WYF}$

Bảng Các độ đo hiệu quả loại bỏ lỗi.

Từ khóa:

- NYF: số thất bại(failures) phần mềm được phát hiện trong suốt 1 năm bảo trì dịch vụ.
- WYF: số thất bại phần mềm đã đánh trọng số trong suốt 1 năm bảo trì dịch vụ.

❖ Độ đo hiệu suất quy trình phần mềm.

4 độ đo năng suất tiến trình, trực tiếp và gián tiếp, được biểu diễn trong bảng 21.5.

Table 21.5: Process productivity metrics

Code	Name	Calculation formula
DevP	Development Productivity	$DevP = \frac{DevH}{KLOC}$
FDevP	Function point Development Productivity	$FDevP = \frac{DevH}{NFP}$
CRe	Code Reuse	$CRe = \frac{ReKLOC}{KLOC}$
DocRe	Documentation Reuse	$DocRe = \frac{ReDoc}{NDoc}$

Trong đó:

- DevH= tổng thời gian làm việc dành cho phát triển của hệ thống phần mềm.
- ReKLOC = số dòng lệnh sử dụng lại trong hàng nghìn dòng code
- ReDoc = số trang sử dụng lại của tài liệu Ndoc= số trang của tài liệu

13.2.3. Các độ đo sản phẩm

Độ đo sản phẩm (product metrics) bao gồm:

- Độ đo chất lượng HD (HelpDesk)
- Độ đo năng suất và hiệu quả HD
- Độ đo chất lượng bảo trì sửa chữa
- Độ đo năng suất và hiệu quả bảo trì sửa chữa phần mềm

❖ Độ đo chất lượng HD

Các kiểu của độ đo chất lượng HD được thảo luận ở đây đề cập đến:

- Độ đo mật độ cuộc gọi HD- phạm vi của yêu cầu khách hàng cho dịch vụ HD được đo bởi số cuộc gọi.
- Độ đo tính nghiêm trọng của HD được tăng lên.
- Độ đo thành công HD - Mức độ của sự thành công trong sự phản ứng lại tới những cuộc gọi này. Thành công đạt được bằng sự hoàn thành dịch vụ yêu

cầu trong thời gian nhất định của hợp đồng dịch vụ.

❖ Độ đo năng suất và hiệu quả HD

Độ đo năng suất liên quan tới tổng tài nguyên đầu tư trong suốt một giai đoạn cụ thể, trong khi độ đo hiệu quả liên quan tới tài nguyên đầu tư trong đáp ứng lại tới 1 cuộc gọi HD.

❖ Độ đo năng suất HD

Độ đo năng suất HD làm cho việc sử dụng của phép đo KLMC dễ dàng áp dụng cho kích thước hệ thống phần mềm bảo trì hoặc theo ước lượng điểm chức năng của hệ thống phần mềm. Hai độ đo năng suất HD được miêu tả trong hình sau:

Table 21.7: HD productivity metrics

Code	Name	Calculation formula
HDP	HD Productivity	$HDP = \frac{HDYH}{KLMC}$
FHDP	Function point HD Productivity	$FHDP = \frac{HDYH}{NMFP}$

Trong đó:

- HDYH = Tổng thời gian làm việc hàng năm đầu tư trong dịch vụ HD của hệ thống phần mềm.
- KLMC và NMFP được định nghĩa ở phần trước.

❖ Độ đo hiệu quả HD

$$HDE = \frac{HDYH}{NHYC}$$

Độ đo trong nhóm này liên quan tới tài nguyên được đầu tư trong đáp ứng lại cuộc gọi HD của khách hàng. Một độ đo thông dụng được miêu tả ở đây, hiệu quả HD (HDE):

Độ đo bảo trì sửa chữa phần mềm xử lý hàng loạt khía cạnh chất lượng của dịch vụ bảo trì. Một sự khác biệt cần thiết giữa thất bại hệ thống phần mềm được giải quyết bởi nhóm bảo trì và thất bại do dịch vụ bảo trì liên quan tới trường hợp bảo trì không cung cấp một sự sửa chữa thỏa mãn những tiêu chuẩn hoặc yêu cầu hợp đồng. Do đó, độ đo bảo trì phần mềm được phân loại như sau:

- Độ đo mật độ thất bại hệ thống phần mềm: Giải quyết phạm vi của yêu cầu tới bảo trì sửa chữa, dựa trên bản ghi của các thất bại được xác định trong

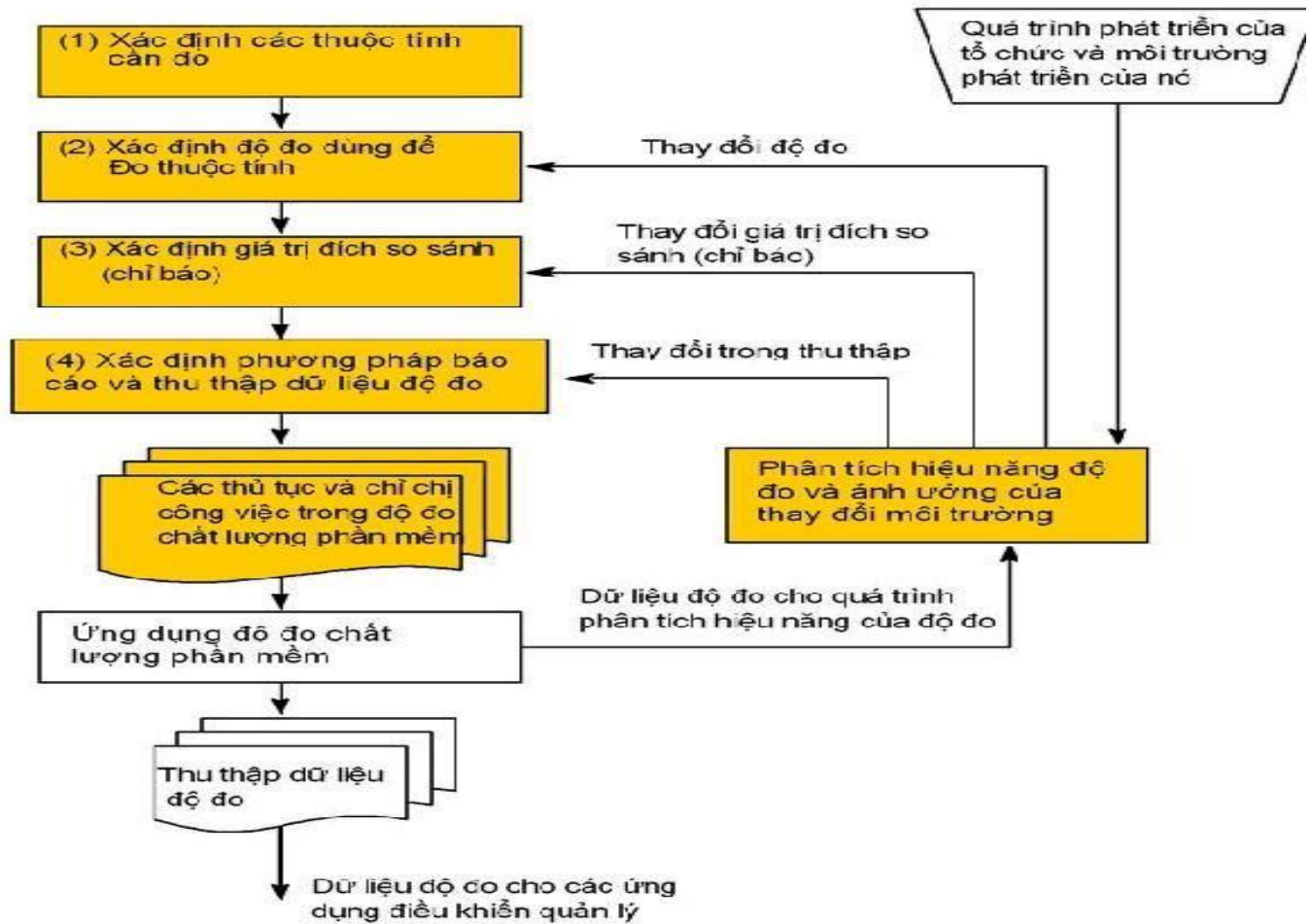
hoạt động bình thường của hệ thống phần mềm.

- Độ đo nghiêm trọng thất bại hệ thống phần mềm: Giải quyết sự nghiêm trọng của thất bại hệ thống phần mềm quan tâm bởi nhóm bảo trì sửa chữa.
- Độ đo thất bại của dịch vụ bảo trì: Giải quyết trường hợp mà dịch vụ bảo trì không thể hoàn thành sửa chữa thất bại đúng thời gian hoặc việc sửa lỗi thất bại.
- Độ đo sự sẵn sàng hệ thống phần mềm: Giải quyết phạm vi của sự xáo trộn được gây ra tới khách hàng như thực hiện bởi một giai đoạn thời gian nơi dịch vụ của hệ thống phần mềm không sẵn sàng hoặc chỉ sẵn sàng một phần.

13.2.4. Thực hiện đo chất lượng phần mềm

Để thực hiện đo chất lượng phần mềm cần:

- Xác định các độ đo chất lượng phần mềm
- Áp dụng thông thường bởi đơn vị...
- Phân tích thống kê dữ liệu độ đo đã thu thập được.
- Các hành động tiếp theo bao gồm:
 - Các thay đổi trong tổ chức và các phương pháp của các đơn vị bảo trì và phát triển phần mềm và/hoặc bất kỳ cá nhân nào thu thập dữ liệu độ đo.
 - Thay đổi về độ đo và thu thập dữ liệu độ đo.
 - Áp dụng dữ liệu và phân tích dữ liệu nhằm lên kế hoạch cho các hoạt động sửa lỗi cho tất cả các đơn vị liên quan.



Quá trình xác định các độ đo chất lượng phần mềm

13.2.5. Những giới hạn của các độ đo phần mềm

Ứng dụng của độ đo chất lượng có nhiều cản trở. Những cản trở này có thể nhóm theo sau:

- Ngân sách ràng buộc trong chỉ định tài nguyên cần thiết (nhân sự, tài chính, vv..) cho phát triển của hệ thống độ đo chất lượng và ứng dụng bình thường của nó.
- Yếu tố con người, đặc biệt là sự chống lại của người lao động để ước lượng những hoạt động này.
- Không chắc chắn về giá trị của dữ liệu, đã ăn sâu vào các bản ghi thiên vị và thành kiến.

13.3. Giá thành của chất lượng phần mềm

13.3.1. Các mục tiêu tính giá thành các độ đo chất lượng phần mềm

Mục tiêu của các phép đo chi phí của chất lượng phần mềm

Việc áp dụng các phép đo chi phí cho đảm bảo chất lượng phần mềm cho phép có được sự quản lý tài chính đối với các hoạt động SQA và các kết quả. Các mục tiêu cụ thể là:

- Quản lý tổ chức – xây dựng chi phí để ngăn ngừa và dò tìm lỗi phần mềm
- Ước tính thiệt hại kinh tế của thất bại phần mềm để làm cơ sở xét duyệt ngân sách cho SQA
- Ước lượng các kế hoạch để tăng hoặc giảm các hoạt động SQA hoặc để đầu tư mới hoặc cập nhật cơ sở hạ tầng SQA dựa trên cơ sở hiệu năng kinh tế trong quá khứ

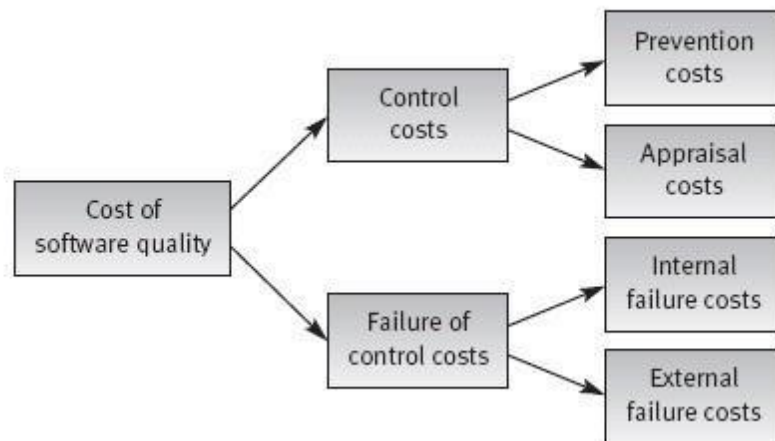
13.3.2. Mô hình truyền thống tính giá chất lượng phần mềm

Mô hình chi phí chất lượng phần mềm được phát triển vào đầu những năm 1950 bởi Feigenbaum và một số người khác (Xem trong Feigenbaum, 1991), cung cấp một phương pháp luận cho việc phân loại chi phí kết hợp với đảm bảo chất lượng sản phẩm từ quan điểm về kinh tế. Được phát triển để phù hợp hoàn cảnh chất lượng dựa trên các tổ chức sản xuất, mô hình đã được thực hiện một cách rộng rãi.

Mô hình phân loại chi phí liên quan đến chất lượng sản phẩm vào hai lớp chung:

- Chi phí quản lý: bao gồm chi phí phải trả cho ngăn ngừa và dò tìm lỗi phần mềm để
- giảm thiểu chúng đến một mức độ được chấp nhận.
- Chi phí thất bại của quản lý: bao gồm chi phí của những thất bại xảy ra bởi vì thất bại trong việc ngăn ngừa và dò tìm lỗi phần mềm.
- Mô hình chia nhỏ hơn nữa thành những lớp con. Chi phí quản lý được chia cho lớp con ngăn ngừa hoặc lớp con chi phí đánh giá:
- Chi phí ngăn ngừa bao gồm các khoản đầu tư vào cơ sở hạ tầng chất lượng và hoạt động chất lượng không được nhằm vào một dự án hoặc hệ thống riêng biệt, là chung cho cả tổ chức.
- Chi phí đánh giá bao gồm chi phí các hoạt động được thực hiện cho những

- dự án cụ thể hoặc hệ thống phần mềm cho mục đích dò tìm lỗi phần mềm.
- Chi phí thất bại của quản lý được chia nhỏ hơn thành chi phí thất bại bên trong và chi phí thất bại bên ngoài:
 - Chi phí thất bại bên trong bao gồm chi phí sửa chữa những lỗi được tìm thấy khi xem xét lại thiết kế, kiểm tra phần mềm và kiểm tra chấp nhận (được thực hiện bởi khách hàng) và được hoàn thành trước khi phần mềm được cài đặt ở phía khách hàng.
 - Chi phí thất bại bên ngoài bao gồm tất cả các chi phí sửa chữa những lỗi được tìm thấy bởi khách hàng hoặc đội bảo trì sau khi phần mềm đã được cài đặt.
 - Mô hình cổ điển của chi phí chất lượng phần mềm được biểu diễn ở hình 22.1. Mặc dù sự cố gắng áp dụng mô hình cổ điển vào việc phát triển và bảo



trì phần mềm đã được ghi nhận là thành công nhưng vẫn không hoàn chỉnh. Nguyên nhân cho những khó khăn phải đối mặt sẽ được thảo luận phần sau của chương này. Nhưng trước khi tiếp tục, ta xem lại mô hình.

Mô hình tính chi phí truyền thống

13.3.3. Mô hình mở rộng tính giá chất lượng phần mềm

Các phân tích chi phí cho đảm bảo chất lượng phần mềm theo mô hình cổ điển cho thấy rằng có một số chi phí rất lớn đã bị bỏ qua. Những chi phí này hoặc là chi

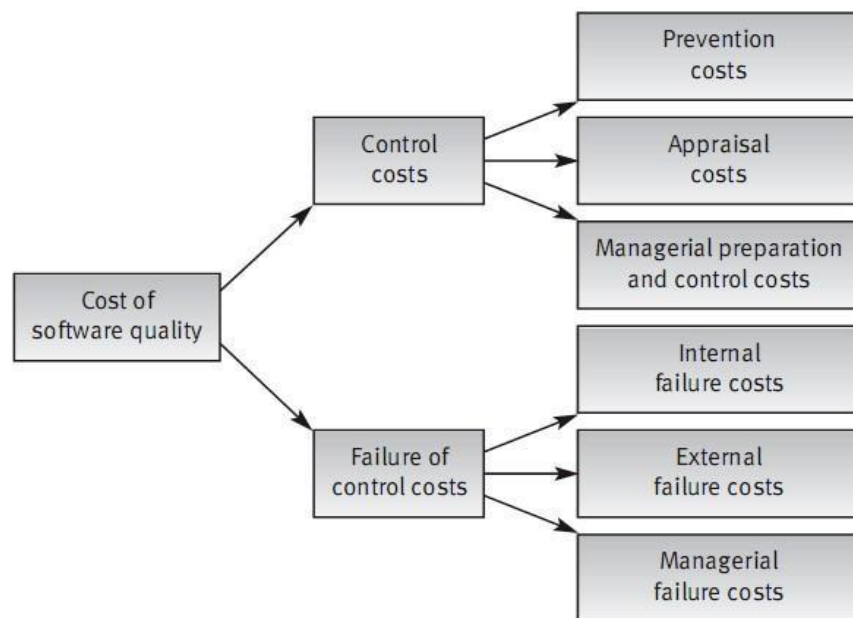
có trong công nghiệp phần mềm, hoặc không đáng kể trong các ngành công nghiệp khác. Ta xét ví dụ về hai chi phí tiêu biểu phải mất do các lỗi trong đảm bảo chất lượng phần mềm như sau:

- Chi phí đền bù do hoàn thành dự án muộn, nguyên nhân xuất phát từ việc lập lịch biểu một cách không thực tế.
- Chi phí đền bù vì hoàn thành dự án muộn do thất bại trong việc tuyển nhân viên mới có đủ khả năng.

Hai thất bại trên đây đều không phải do bất cứ một hoạt động cụ thể nào của nhóm phát triển hay do thiếu chuyên môn nghiệp vụ, mà là do việc quản lý gây ra. Do đó, những người quản lý dự án có thể thực hiện một vài hoạt động để ngăn ngừa hoặc chí ít là giảm thiểu chi phí mà các loại thất bại trên gây ra:

- Xem xét lại hợp đồng (xem xét lại các đề xuất sơ bộ và hợp đồng sơ bộ). Chi phí của các hoạt động này thường không đáng kể đối với các hợp đồng trong công nghiệp sản xuất. Tuy nhiên trong công nghiệp phần mềm, người ta cần một lượng công việc chuyên môn đáng kể để đảm bảo rằng các đề xuất của dự án được dựa trên các đánh giá cẩn thận và các ước lượng dễ hiểu (rõ ràng). Sự khác biệt về yêu cầu tài nguyên cho cùng công việc xem xét lại hợp đồng giữa công nghiệp phần mềm và công nghiệp sản xuất xuất phát từ bản chất của sản phẩm và tiến trình sản xuất được đề cập đến trong hợp đồng. Trong khi hợp đồng trong công nghiệp sản xuất xử lý việc tạo ra nhiều lần các sản phẩm trong danh sách cố định, thì hợp đồng trong công nghiệp thường chỉ xử lý việc phát triển một hệ thống phần mềm mới, duy nhất.

- Điều khiển tiến độ dự án phần mềm một cách kỹ lưỡng và thích hợp. Trong khi việc điều khiển quá trình sản xuất trong công nghiệp sản xuất là một công việc lặp đi lặp lại và thường được thực hiện tự động bằng máy móc, thì việc điều khiển tiến độ phát triển phần mềm quản lý thiết kế tác vụ và các



hoạt động coding được thực hiện bởi nhóm phát triển dự án.

Mô hình tính chi phí mở rộng

13.3.4. Các vấn đề trong áp dụng tính giá các độ đo chất lượng phần mềm

BÀI 14. BÀI TẬP VỀ ĐỘ ĐO CHẤT LƯỢNG PHẦN MỀM

Bài 1. Phần độ đo mật độ lỗi mô tả 3 mật độ lỗi sau đây: CED, WCED và WCEF

1. So sánh CED và WCED về đặc điểm ứng dụng quản lý cũng như tính hợp lệ của chúng.
2. So sánh WCED và WCEF về đặc điểm thực hiện quản lý của họ cũng như tính hợp lệ của chúng.
3. Bạn muốn số liệu nào ở trên? Liệt kê các lập luận của bạn.

Bài 2. Phần độ đo hiệu suất tiến trình mô tả hai chỉ số năng suất phát triển sau đây: DevP và FDevP.

1. So sánh DevP và FDevP về đặc điểm thực hiện quản lý cũng như tính hợp lệ của chúng.
2. Bạn muốn số liệu nào ở trên? Liệt kê các lập luận của bạn.

Bài 3. Phần độ đo chất lượng các chỉ số cho các dịch vụ bảo trì HD và sửa chữa.

1. Giải thích sự khác biệt giữa các dịch vụ này.
2. Giải thích các loại số liệu riêng biệt và các hành động dựa trên sự khác biệt của chúng.

Bài 4 Mô tả hai chỉ số mật độ thất bại bảo trì - WSSFD và WSSFF.

1. Đánh giá từng chỉ số trên về mức độ đáp ứng các yêu cầu về các chỉ số chất lượng phần mềm như liệt kê.
2. Cho biết chiều hướng mong muốn của biến dạng cho từng số liệu.

Bài 5 Các dịch vụ HD là rất quan trọng để sử dụng thành công một hệ thống phần mềm.

1. Đề nghị các tình huống mà dịch vụ HD là một sự thất bại.
2. Những thước đo nào có thể được áp dụng cho các tình huống thất bại được đề cập trong câu trên?

Bài 6. Mô tả một số biện pháp được sử dụng để xây dựng số liệu phát triển phần mềm được trình bày trong phần này.

Dựa trên các biện pháp được liệt kê, đề xuất hai chỉ số chất lượng quy trình mới và hai chỉ số năng suất quy trình mới.

Bài 7 Mô tả một số biện pháp được sử dụng để xây dựng các chỉ số bảo dưỡng HD và bảo dưỡng sửa đổi được trình bày trong phần này.

Dựa trên các biện pháp được liệt kê, đề xuất ba chỉ số chất lượng sản phẩm mới và hai chỉ số năng suất sản phẩm mới.

Bài 8 Chọn một trong các số liệu sản phẩm được mô tả trong Phái 13 bao gồm NYF như một trong các biện pháp của nó.

1. Xem xét năm yếu tố ảnh hưởng đến các biện pháp bảo trì được liệt kê trong bài 13, chỉ ra theo hướng mà mỗi người trong số họ có thể thiên vị các chỉ số bạn đã chọn và chỉ ra sự thiên vị đó ảnh hưởng đến tính hợp lệ của chỉ số.
2. Xem xét năm yếu tố trên và chỉ ra cách mỗi người trong số họ có thể hạn chế tính toàn diện của số liệu mà bạn đã chọn.

Bài 9 Hệ thống phần mềm nguồn nhân lực yêu cầu 15000 dòng mã Visual Basic và 5000 dòng mã SQL.

1. Ước tính số điểm chức năng cần thiết cho hệ thống phần mềm.
2. Ước tính số dòng mã C yêu cầu cho hệ thống phần mềm.

Bài 10 Phân tích các yêu cầu kỹ thuật cho một cuộc đấu thầu cho sự phát triển của hệ thống CRM Club người mua đã được công bố trong một tạp chí chuyên nghiệp.

ABC Software Lab đang cân nhắc tham gia đấu thầu. Nhóm được chỉ định để chuẩn bị hồ sơ thầu đã phân tích các yêu cầu kỹ thuật và đạt được các kết quả sau:

- Số đầu vào người sử dụng - 28
- Số lượng đầu ra của người dùng - 36
- Số truy vấn trực tuyến của người dùng - 24
- Số tệp logic - 8
- Số giao diện bên ngoài - 12.

Nhóm ước tính rằng 50% các thành phần là đơn giản, trung bình 25% và phức tạp 25%. Nhóm nghiên cứu cũng đánh giá mức độ phức tạp của dự án, với ước tính là $RCAF = 57$.

1. Tính các ước tính điểm chức năng cho dự án.
2. Ông Barnes, Trưởng Lập trình, ước tính có 3500 dòng mã C++ sẽ được yêu cầu cho dự án. Dựa trên kết quả cho câu 1, bạn có đồng ý với ước tính của mình?

Bài 15. CHUẨN QUẢN LÝ CHẤT LƯỢNG ISO, CMM và CMMI

15.1. Chuẩn quản lý chất lượng

15.1.1. Phạm vi của các chuẩn quản lý chất lượng

Các tiêu chuẩn chứng chỉ phần lớn dựa trên sự đánh giá nội dung hay mức độ quan trọng. Phạm vi của các tiêu chuẩn chứng chỉ dựa trên mục đích của các chứng chỉ. Đó là:

- Cho phép tổ chức phát triển phần mềm chứng minh năng lực phù hợp để đảm bảo sản phẩm phần mềm của mình hay các dịch vụ bảo trì tuân theo đúng yêu cầu chất lượng.
- Giống như sự giao ước cơ bản của khách hàng và nhà cung cấp nhằm đánh giá hệ thống quản lý chất lượng của nhà cung cấp, điều này có thể được hoàn thành bởi việc kiểm toán chất lượng của khách hàng. Việc kiểm toán này dựa trên các yêu cầu chứng chỉ tiêu chuẩn.
- Hỗ trợ tổ chức phát triển phần mềm để cải thiện hiệu năng của hệ thống quản lý chất lượng và nâng cao sự hài lòng của khách hàng thông qua việc tuân theo các tiêu chuẩn yêu cầu

Phạm vi của các tiêu chuẩn đánh giá cũng xác định trên mục đích của việc đánh giá, đó là

- Các tổ chức phát triển và duy trì phần mềm sẽ sử dụng như 1 công cụ cho việc
- đánh giá bản thân về kỹ năng của họ trong đảm nhận các dự án phát triển phần mềm
- Như 1 công cụ để cải thiện quá trình phát triển và duy trì, các tiêu chuẩn có thể đưa ra các chỉ dẫn cho việc cải thiện tiến trình.
- Giúp các tổ chức chi trả hay đầu tư xác định được tiềm năng của nhà cung cấp
- Hướng dẫn tập luyện đánh giá bằng cách mô tả chất lượng và các khóa học lập trình.

15.1.2. Giới thiệu các chuẩn quản lý chất lượng

15.2. ISO 9001 và ISO 9000-3

ISO 9000-3, được đưa ra bởi tổ chức ISO, trình bày cách cài đặt thực hiện phương pháp luận chung của tiêu chuẩn quản lý chất lượng ISO 9000 trong các trường hợp đặc biệt của phát triển và duy trì phần mềm. Cả ISO 9001 và ISO 9000-3 được xem xét lại và cập nhật liên tục khoảng 5 – 8 năm 1 lần, với những các cuộc thảo luận riêng rẽ.

Giống như ISO 9000-3 dựa trên cơ sở những gì mà tiêu chuẩn ISO 9001 đưa ra, việc công bố các sửa chữa trong các nguyên tắc cũng dựa trên sự công bố các tiêu chuẩn đã được xem xét lại sau khoảng 1 vài năm. Chẳng hạn như, phiên bản 1997 của ISO 9000-3 dựa trên phiên bản 1994 của ISO 1994. Phiên bản 1997 của ISO 9000-3 kết hợp ISO 9001 cùng các đặc trưng của ISO 9000-3 thành tiêu chuẩn “tất cả trong một” cho ngành công nghiệp phần mềm.

Nguyên lý hướng dẫn tiêu chuẩn ISO 9000 – 3 mới, có nguồn gốc từ tiêu chuẩn ISO9000:2000 như sau:

- Tập trung khách hàng : Tổ chức dựa trên khách hàng của họ và từ đó nên hiểu được các yêu cầu hiện tại và trong tương lai của khách hàng
- Lãnh đạo : Những người lãnh đạo thiết lập tầm nhìn của tổ chức. Họ nên tạo ra và duy trì môi trường bên trong mà mọi người có thể phát triển nhằm đạt được mục tiêu của tổ chức qua các lộ trình định sẵn
- Sự phát triển của con người : Con người là cốt lõi của tổ chức. Tại mọi mức độ của tổ chức, sự phát triển của nhân lực cho phép kỹ năng của họ góp phần làm tăng lợi nhuận của tổ chức
- Tiếp cận tiến trình : Kết quả mong muốn giành được 1 cách hiệu quả hơn khi các hoạt động và tài nguyên được quản lý như 1 tiến trình
- Tiếp cận hệ thống để quản lý : Xác định, hiểu được và quản lý tiến trình, nếu nhìn như 1 hệ thống sẽ góp phần nâng cao hiệu năng tổ chức.
- Tiếp tục cải thiện : Liên tục cải thiện tất cả các hiệu suất nên được đề cao trong các cuộc họp của tổ chức

- Tiếp cận sự thực để ra quyết định : Các quyết định hiệu quả dựa trên việc phân tích các thông tin
- Mỗi quan hệ hỗ trợ lẫn nhau giữa những nhà cung cấp : 1 tổ chức và nhà cung cấp của nó là độc lập với nhau .Tuy nhiên mỗi quan hệ hỗ trợ lẫn nhau có thể nâng cao kỹ năng của cả 2 để tạo thêm ra giá trị gia tăng

Tiêu chuẩn ISO 9000-3 (ISO 1997) gồm 20 yêu cầu liên quan đến rất nhiều khía cạnh của hệ thống quản lý chất lượng phần mềm. Phiên bản mới hơn ISO 2001 đề nghị 1 cấu trúc mới với 22 yêu cầu được phân thành 5 nhóm cơ bản :

- Hệ thống quản lý chất lượng:
 - Những yêu cầu chung
 - Các tài liệu yêu cầu
- Trách nhiệm quản lý :
 - Sự tận tâm trong quản lý
 - Đối tượng khách hàng cần tập trung
 - Điều khoản về chất lượng
 - Lập kế hoạch
 - Trách nhiệm , thẩm quyền và cách truyền đạt
 - Xem xét lại về quản lý
- Quản lý tài nguyên :
 - Tài nguyên cung cấp
 - Nhân lực
 - Các thiết bị cơ bản
 - Môi trường làm việc
- Hiểu rõ sản phẩm
 - Hiểu rõ kế hoạch của sản phẩm
 - Quá trình quan hệ với khách hàng có liên quan
 - Thiết kế và phát triển
 - Khả năng chi trả
 - Dự liệu về sản phẩm và các dịch vụ
 - Điều khiển thiết bị đo lường và kiểm tra
 - Quản lý , phân tích và cải thiện :

- Kiểm tra và đo lường
- Điều khiển các sản phẩm không phù hợp
- Phân tích dữ liệu
- Cải thiện chất lượng

ISO 9001 – từ ứng dụng đến phần mềm : Sáng kiến TickIT

Được bắt đầu vào cuối thập kỷ 1980 , ngành công nghiệp phần mềm Vương quốc Anh cộng tác với Bộ công thương UK đã nâng phương pháp luận phát triển cho ISO 9001 thành các đặc điểm riêng cho công nghiệp phần mềm được biết đến như sáng kiến TickIT. Tại thời điểm nó bắt đầu, ISO 9001 đã được áp dụng thành công vào công nghiệp chế tạo tuy nhiên không có phương pháp luận nào có ý nghĩa cho các đặc điểm riêng biệt của công nghiệp phần mềm. Vài năm sau đó, TickIT cùng với sự nỗ lực nghiên cứu trong phát triển ISO 9000-3 đã đạt được thành quả này .

15.3. Các mô hình tăng trưởng khả năng – phương pháp đánh giá CMM và CMMI

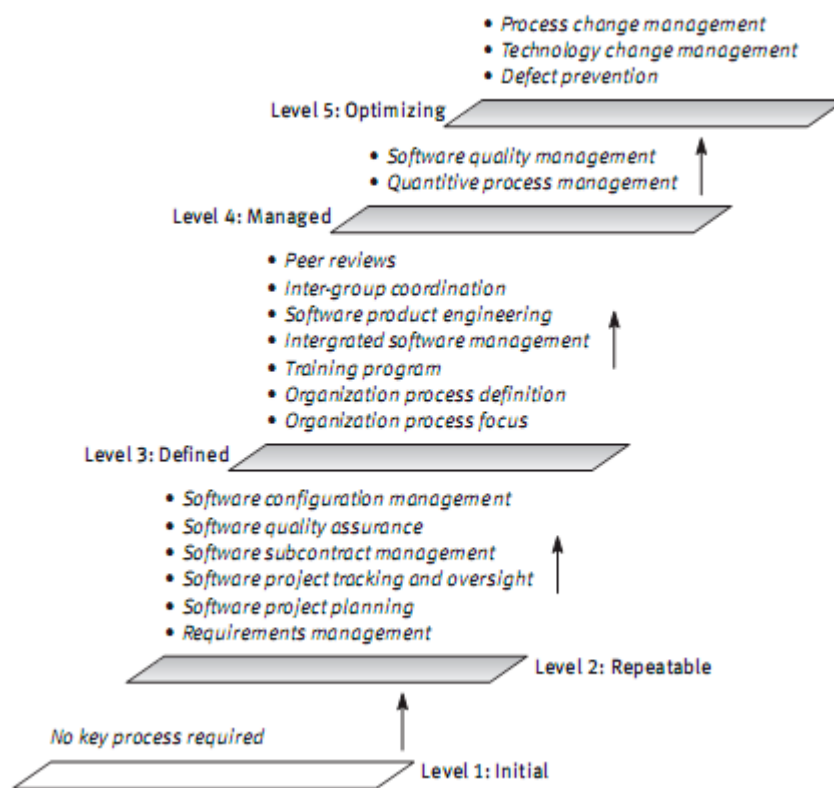
Đánh giá CMM dựa trên các khái niệm và nguyên lý sau:

- Ứng dụng của nhiều phương thức quản lý phức tạp dựa trên cách tiếp cận định lượng nhằm tăng khả năng của tổ chức để điều khiển chất lượng và cải thiện hiệu quả của quá trình phát triển phần mềm.
- Phương tiện để đánh giá việc phát triển phần mềm bao gồm 5 mức của mô hình đánh giá khả năng hoàn thành công việc đúng thời hạn. Mô hình cho phép một tổ chức đánh giá hiệu quả hoạt động của nó và xác định các yêu cầu cần thiết để có thể đạt tới mức tiếp theo bằng cách xác định các vùng xử lý được yêu cầu cho việc cải thiện.

- Vùng xử lý được dùng chung, chúng định nghĩa “cái gì” chứ không phải “bằng cách nào”. Đây là cách tiếp cận cho phép mô hình có thể được áp dụng cho một cho một miền rộng lớn của việc thực thi tổ chức, bởi vì:
 - Nó cho phép sử dụng một số mô hình vòng đời.
 - Nó cho phép sử dụng một số phương pháp luận, công cụ để phát triển phần mềm và ngôn ngữ lập trình.
 - Nó không chỉ rõ một tài liệu chuẩn cụ thể nào.

Mô hình CMMI cũng giống như mô hình gốc CMM bao gồm 5 mức. Các mức của CMMI cũng giống như các mức trong bản gốc của nó ,chỉ có một sự thay đổi nhỏ ở mức 4, được đặt tên như sau:

- Capability maturity level 1: khởi tạo
- Capability maturity level 2: Lập lại
- Capability maturity level 3: Xác định
- Capability maturity level 4: Quản lí
- Capability maturity level 5: Tối ưu hóa



Các mức của mô hình CMM và các miền tiến trình then chốt

BÀI 16. CÁC CHUẨN TIẾN TRÌNH DỰ ÁN SQA

- Các chuẩn tiến trình dự án SQA tập trung vào phương pháp luận để thực hiện việc phát triển phần mềm và bảo trì phần mềm trong đó:

- Mô tả mỗi bước của một quá trình
- Yêu cầu kèm theo: tài liệu thiết kế, các nội dung thiết kế, xem xét lại thiết kế và các vấn đề xem xét lại, kiểm tra phần mềm và các mục tiêu của nó...
- Các tổ chức phát triển chuẩn: IEEE, EIA, ISO....
- Các chuẩn IEEE có thể được chia thành 3 lớp chính
- Các chuẩn khái niệm (conceptual standards)
 - IEEE 610.12, IEEE 1061, IEEE 1320.2, IEEE 1420.1a, IEEE/EIA 12207.0
- Các chuẩn yêu cầu cho sự tương thích (Prescriptive standards of conformance)

PTIT

182

- IEEE 828, IEEE 829, IEEE 1012, IEEE 1028, IEEE 1042.1
- Các chuẩn hướng dẫn (Guidance standards):
 - IEEE 1233, IEEE/EIA 12207.1, IEEE/EIA 12207.2

16.1. IEEE/EIA Std 12207- các tiến trình vòng đời phần mềm

Chuẩn IEEE/EIA Std 12207 cung cấp framework chung cho phát triển và quản lý phần mềm

- Tổ chức phát triển chuẩn:
 - US Department of Defense (MIL-STD-498:1994)
 - ANSI, IEEE và EIA (Joint Standard 016 (J-Std-016-1995))
 - Tổ chức tiêu chuẩn quốc tế (ISO) và IEC – ISO/IEC 12207 Standard
- Chuẩn gồm 3 phần – three-part standard
- IEEE/EIA Std 12207.0-1996 (IEEE/EIA, 1996): bao gồm ISO/IEC 12207 gốc và các phần thêm vào
- IEEE/EIA Std 12207.1-1997 (IEEE/EIA, 1997a): Hướng dẫn – dữ liệu vòng đời
- IEEE/EIA Std 12207.2-1997 (IEEE/EIA, 1997b): Hướng dẫn – Xem xét cài đặt

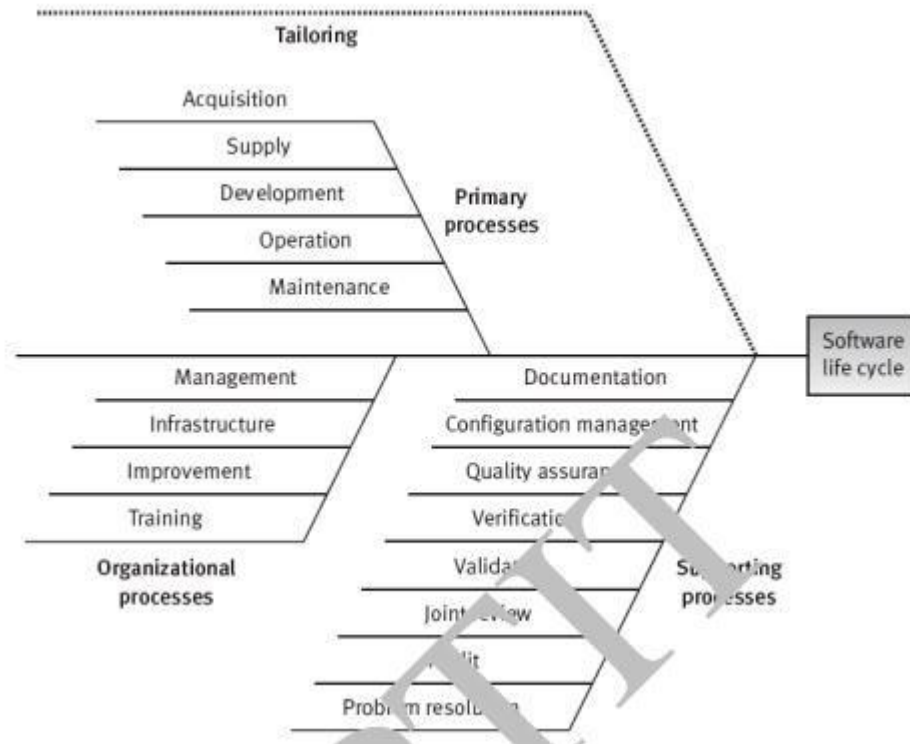
Mục tiêu của IEEE/EIA Std 12207 là:

- Thiết lập mô hình các quá trình vòng đời phần mềm chung được chấp nhận trên toàn cầu
- Đẩy mạnh sự am hiểu giữa các nhóm nghiệp vụ bằng việc áp dụng các quá trình, các hoạt động và các nhiệm vụ đã được chấp nhận chung

Kiến trúc vòng đời phần mềm IEEE/EIA 12207 bao gồm:

- 4 mức
- Process classes: Các lớp quy trình
- Processes: Các tiến trình
- Activities: Các hoạt động
- Tasks: Các công việc
- 3 quy trình
- Primary life cycle processes

- Supporting life cycle processes
- Organizational life cycle processes



Biểu đồ xương cá- Những tiến trình vòng đời phần mềm theo chuẩn IEEE/EIA 12207

- **Các khái niệm chung – general concepts**
 - Khả năng áp dụng chuẩn nói chung và sự thích ứng của nó bằng việc điều chỉnh cho một nhu cầu cụ thể
 - Khả năng áp dụng cho tất cả các bên tham gia trong vòng đời phần mềm
 - Tính mềm dẻo và tính đáp ứng với thay đổi công nghệ
 - Phần mềm liên kết với hệ thống
 - Tính nhất quán của quản lý chất lượng toàn phần mềm
 - Không có các yêu cầu xác minh
 - Baselining
- **Các khái niệm liên quan tới công việc – task-related concepts**
 - Gán trách nhiệm cho mỗi hoạt động và mỗi công việc
 - Modun của các thành phần trong vòng đời phần mềm
 - Các mức phù hợp được yêu cầu
 - Bản chất của công việc ước lượng

16.2. IEEE Std 1012 – xác minh và thẩm định

Mục tiêu của chuẩn IEEE Std 1012 là:

- Thiết lập framework chung cho các hoạt động xác minh và thẩm định(V & V)
- Xác định đầu vào – đầu ra của việc V & V

- Xác định các mức toàn vẹn phần mềm, công việc V & V cho mỗi mức
- Xác định nội dung của tài liệu lập kế hoạch V&V

Các khái niệm cơ bản của IEEE 1012

- Định nghĩa rộng về các hoạt động V&V
 - Cho phép chuẩn bao quát tất cả các hoạt động V&V thực hiện xuyên suốt vòng đời phần mềm
- Tuân thủ và tương thích với các chuẩn quốc tế
- Các mức toàn vẹn của phần mềm và các yêu cầu V&V cho từng mức
 - Độc lập của các hoạt động V&V
 - Tính độc lập về quản lý: nhóm V&V tự quyết định phương thức V&V nào được áp dụng.
 - Tính độc lập về công nghệ: các thành viên trong nhóm V&V không liên quan đến việc phát triển phần mềm
 - Tính độc lập về tài chính: quyền điều hành về ngân sách của V&V không được trao cho nhóm phát triển

Quy trình xác minh và thẩm định bao gồm:

- Quy trình quản lý
- Quy trình thu thập
- Quy trình cung cấp
- Quy trình phát triển
- Quy trình vận hành
- Quy trình bảo trì

16.3. IEEE Std 1028 – rà soát

- Mục tiêu của IEEE Std 1028 là

Định nghĩa các thủ tục của rà soát một cách hệ thống để:

- Thích hợp cho việc xem xét lại được thực hiện trong suốt vòng đời phần mềm
- Phù hợp với những yêu cầu việc xem xét lại được định nghĩa bởi những tiêu chuẩn khác

Có 5 kiểu rà soát trong IEEE Std 1028:

- Xem xét lại quản lý.
- Xem xét lại kỹ thuật
- Kiểm tra.
- Walkthroughs
- Kiểm toán

Ba khái niệm cơ bản được đặc trưng

- Hình thức cao - High formality
- Follow-up
- Sự tuân theo tiêu chuẩn quốc tế và IEEE IEEE Std 1028-. 1997

Phần chính của IEEE Std 1028-1997 đòi hỏi:

- Trình bày định nghĩa chi tiết của những yêu cầu của việc xem xét lại

- Phụ lục trình bày những mối quan hệ của tiêu chuẩn với những quá trình vòng đời
- Thành phần của yêu cầu rà soát:
 - Giới thiệu:
 - Những mục đích của mỗi kiểu xem xét lại.
 - Những ví dụ tiêu biểu của mỗi kiểu sản phẩm phần mềm
 - Các trách nhiệm: Chuẩn cung cấp một danh sách những người tham gia, có thể là:
 - Người tham gia ủy quyền (ví dụ: lãnh đạo, nhân viên kỹ thuật...)
 - Hoặc những người tham gia tùy chọn khác (ví dụ: nhà quản lý, khách hàng...)
 - Đầu vào dữ liệu được chia thành
 - những mục bắt buộc: tập trung vào mục đích của việc xem xét lại (vd: phát biểu mục tiêu, các tiêu chuẩn...)
 - tùy chọn (vd: những điều chỉnh, những bất thường)
 - Tiêu chuẩn: Là những điều kiện ban đầu để cho phép và thực hiện việc xem xét lại, bao gồm:
 - Phát biểu tất cả các mục tiêu của việc xem xét lại
 - Tính sẵn sàng của dữ liệu đầu vào được yêu cầu
 - Thủ tục giao gồm:
 - Chuẩn bị quản lý
 - Lập kế hoạch để xem xét lại
 - Sắp xếp thành viên trong nhóm
 - Kiểm tra sản phẩm phần mềm
 - Theo dõi các hoạt động chỉnh sửa
 - Các tiêu chuẩn đã tồn tại xác định những việc phải được thực hiện trước khi việc xem xét lại kết thúc hoàn toàn, bao gồm:
 - Hoàn thành các hoạt động theo thủ tục
 - Theo dõi và phê chuẩn của các mục hành động đã được hoàn thành hoặc các hoạt động sửa chữa và ngăn chặn
 - Hoàn thành tài liệu của việc xem xét lại.
 - Đầu ra: gồm:
 - Các tiêu chuẩn xác định các đầu ra cho mỗi loại xem xét lại
 - Các mục bổ sung có thể được yêu cầu bởi tổ chức, bởi các thủ tục bộ phận
 - khác, hoặc trong các trường hợp cụ thể.
 - Các khuyến nghị thu thập dữ liệu:

Nhóm kiểm tra và walkthrough thu thập dữ liệu liên quan tới những bất thường đã gặp mỗi trường hợp được phân loại và xếp hạng theo mức độ nghiêm trọng. Dữ liệu này được sử dụng để nghiên cứu tính hiệu quả của quá trình thực hiện hiện tại chúng cũng được sử dụng để khuyến khích cải tiến các phương pháp và thủ tục.

 - Cải tiến: dữ liệu kiểm tra và walkthrough sẽ được phân tích theo:
 - Hoàn thiện các thủ tục

- Cập nhật danh sách kiểm tra (checklists) được sử dụng bởi những người tham gia
- Cải thiện các quy trình phát triển phần mềm

16.4. Thảo luận về các chuẩn quản lý chất lượng

BÀI 17. TỔ CHỨC ĐỂ ĐẢM BẢO CHẤT LƯỢNG

17.1. Giới thiệu

17.1.1. Cơ cấu tổ chức phát triển phần mềm

Bất cứ tổ chức phát triển phần mềm nào cũng có thể được tạo ra bởi 3 mức cấu

trúc quản lý: người quản lý cấp cao(top management), người quản lý mức trung bình (middle management) và người quản lý dự án (project management).

Người quản lý cấp cao là những người quản lý chung trong tổ chức. Người quản lý mức trung bình có thể có nhiều loại, tùy thuộc vào quy mô của từng tổ chức, bao gồm một số trách nhiệm và một số mức như : các giám đốc bộ phận và các trưởng phòng... Người

quản lý dự án cũng có nhiều loại, phụ thuộc vào dự án và quy mô của dự án, bao gồm các

giám đốc dự án, trưởng nhóm...

Với mục đích thảo luận, chúng ta chỉ xem xét 3 mức cơ bản trong tổ chức phát triển phần mềm :

- **Người quản lý cấp cao** (top management): quản lý và thực thi chung, ví dụ như các CEO.
- **Giám đốc bộ phận** (department management): những người quản lý việc phát triển, bảo trì và kiểm thử phần mềm
- **Người quản lý dự án** và trưởng nhóm phát triển dự án và các dịch vụ bảo trì

17.1.2. Khung tổ chức phát triển phần mềm

Để hệ thống quản lý chất lượng phần mềm thực thi thành công, thì rất nhiều, thậm chí toàn bộ các nhân viên của tổ chức phải có trách nhiệm với sản phẩm và dịch vụ phần mềm của tổ chức. Kết quả này được hoàn thành nhờ sự tuân thủ theo các thủ tục và chỉ thị công việc SQA (đã được đề cập trong chương 14), và nhờ thực hiện một cách chuyên nghiệp các tác vụ(task).

Khung tổ chức bảo đảm chất lượng thuộc cơ cấu tổ chức phần mềm, bao gồm các thành viên (các tác nhân) được chia ra thành các nhóm những người quản lý, tester, và các chuyên gia SQA:

- Những người quản lý:
- Người quản lý cấp cao quản lý và thực thi chung, đặc biệt là việc thực thi có liên quan đến đảm bảo chất lượng phần mềm một cách chính xác
- Giám đốc bộ phận phát triển phần mềm và bảo trì
- Giám đốc bộ phận kiểm thử phần mềm
- Người quản lý dự án và trưởng nhóm phát triển và bảo trì phần mềm
- Trưởng đội kiểm thử dự án
- Tester: những thành viên trong đội kiểm thử dự án
- Các chuyên gia SQA
 - Những ủy viên quản trị SQA. (SQA trustees)
 - Những thành viên của ủy ban SQA.(SQA committee members)

- Những thành viên của diễn đàn SQA.(SQA forum members)
- Những thành viên của đơn vị SQA.(SQA unit team members)

17.2. Quản lý và vai trò của quản lý trong đảm bảo chất lượng phần mềm

17.2.1. Các hoạt động đảm bảo chất lượng của quản lý mức cao nhất

Bên cạnh các trách nhiệm khác, ban quản lý cấp cao cũng có trách nhiệm về chất lượng phần mềm. Toàn bộ những trách nhiệm này được tóm tắt như sau:

- Đảm bảo chất lượng sản phẩm phần mềm của công ty và các dịch vụ bảo trì phần mềm
- Truyền đạt sự quan trọng về chất lượng của sản phẩm và dịch vụ cùng với sự hài lòng của khách hàng tới các nhân viên ở mọi mức (truyền đạt tới toàn bộ nhân viên)
- Đảm bảo sự hài lòng về chức năng và toàn bộ sự bằng lòng với yêu cầu của khách hàng
- Đảm bảo những mục tiêu về chất lượng được thiết lập cho hệ thống SQA của tổ chức và những mục tiêu này được thực hiện
- Bắt đầu lên kế hoạch và quan sát việc thực thi các thay đổi cần thiết để tích hợp vào hệ thống SQA các thành phần chính bên trong cũng như các thay đổi bên ngoài có liên quan đến khách hàng, sự cạnh tranh và kỹ thuật của tổ chức.
- Can thiệp 1 cách chính xác để giải quyết những lúc gặp khủng hoảng và giảm thiểu thiệt hại.
- Đảm bảo tính sẵn sàng của các tài nguyên được yêu cầu bởi hệ thống SQA.

3 công cụ chính sẵn có để người quản lý cấp cao hoàn thành trách nhiệm:

- (1) Thiết lập và cập nhật các chính sách đảm bảo chất lượng của tổ chức
- (2) Chỉ định công việc thực thi các vấn đề về chất lượng phần mềm cho 1 người (ví dụ Phó giám đốc chuyên về SQA)
- (3) Điều khiển việc rà soát quản lý về hiệu suất liên quan đến các vấn đề chất lượng phần mềm.

Chúng ta sẽ xem xét lần lượt từng công cụ này.

a. Các chính sách chất lượng phần mềm

Chính sách chất lượng chỉ ra mục đích và mục tiêu của một tổ chức về vấn đề

chất lượng được quy định bởi người quản lý.(Pressman 2004)

Theo chuẩn ISO-9003, chính sách chất lượng xác định chính sách mô tả quan điểm của tổ chức về vấn đề chất lượng.

Chính sách về chất lượng phần mềm của tổ chức, thông qua nội dung chung và các phát biểu của họ, phải truyền đạt đầy đủ các yêu cầu sau:

- (1) Phù hợp với mục đích và mục tiêu của tổ chức
- (2) Cam kết các khái niệm khung về đảm bảo chất lượng phần mềm
- (3) Cam kết các chuẩn chất lượng thông qua bởi tổ chức

b. Việc thực thi đảm bảo chất lượng phần mềm

Trách nhiệm thực hiện các vấn đề về chất lượng phần mềm có thể được phân chia như sau:

(5) Trách nhiệm chuẩn bị chương trình hoạt động và ngân sách cho SQA hàng năm

(6) Trách nhiệm chuẩn bị kế hoạch phát triển hệ thống SQA

(7) Kiểm soát toàn bộ việc cài đặt chương trình hoạt động SQA và lên kế hoạch cho dự án phát triển SQA

(8) Đưa ra và ủng hộ các vấn đề SQA để quản lý việc thực thi

Chi tiết từng trách nhiệm:

(1) Trách nhiệm chuẩn bị các chương trình hoạt động và ngân sách cho SQA hàng năm

Yêu cầu thực hiện:

(5) Đưa ra mục tiêu SQA của hệ thống trong năm tới

(6) Xem xét lại những đề xuất được chuẩn bị bởi các đơn vị SQA cho các hoạt

động hàng năm và xác minh những đề xuất có khả năng hoàn thành các mục tiêu cho hệ thống SQA

(7) Xác định chương trình hoạt động thích hợp với những đặc điểm và phạm vi

của các dịch vụ của nhà thầu phụ và lên kế hoạch mua sắm trong năm tới

(8) Xác định nguồn lực và những tài nguyên khác để lên kế hoạch cài đặt các chương trình SQA

(9) Phê chuẩn phiên bản cuối cùng của chương trình hoạt động và ngân sách cho SQA

(2) Trách nhiệm chuẩn bị các kế hoạch phát triển hệ thống SQA

(1) Xem xét lại những xu hướng được mong đợi cải thiện chất lượng phần mềm của tổ chức trong tương lai gần

(2) Xem xét lại những đề xuất về tích hợp SQA. Ví dụ, việc cố gắng để thử những mua sắm mới dẫn đến những công cụ phát triển phần mềm mới và cần tuân theo các chuẩn chất lượng phần mềm mà không bao giờ được công ty chấp nhận trước đó. Việc tích hợp vào hệ thống SQA bao gồm:

1. Chuẩn bị những thủ tục mới thích hợp với các công cụ mới và các chuẩn SQA

2. Chuẩn bị những chương trình tập huấn cho các nhóm phát triển phần mềm kỳ

cụ (veteran) và các thành viên mới

3. Phát triển việc đo lường chất lượng phần mềm thích hợp với sự phát triển các

công cụ mới và các chuẩn cũng như sự thành công của các chương trình tập huấn

1. Phê chuẩn các phiên bản cuối cùng của các dự án phát triển SQA đã được lên kế hoạch, bao gồm cả lịch biểu và ngân sách.

(3) Kiểm soát toàn bộ việc cài đặt chương trình và các dự án được lên kế hoạch hàng năm

2. Giám sát chung chương trình hoạt động hàng năm

3. Xem xét lại tiến trình tích hợp dự án SQA
4. Giám sát chung các hoạt động được tin tưởng sẽ hoàn thành các mục tiêu của nhóm (dựa trên cơ sở những báo cáo định kỳ)
5. Tiếp tục tuân theo lịch biểu và ngân sách phát triển dự án phần mềm
6. Tiếp tục tuân theo những điều khoản về bảo trì chất lượng dịch vụ cho khách hàng

(4) Đưa ra và ủng hộ các vấn đề SQA

Để đẩy mạnh chất lượng và giải quyết những khó khăn trong hệ thống SQA thì yêu cầu:

- Đưa ra phê chuẩn cuối cùng của chương trình hoạt động và ngân sách hàng năm
- Đưa ra sự phê chuẩn cuối cùng về những dự án tích hợp SQA đã được lên kế hoạch cùng với ngân sách.
- Tạo ra và chỉ đạo những cuộc họp “rà soát quản lý” định kỳ về các chính sách chất lượng phần mềm của tổ chức và các vấn đề của hệ thống SQA, tóm tắt lại thành báo cáo.
- Tạo ra những cuộc thảo luận các cấp quản lý về những vấn đề chất lượng phần mềm đặc biệt, ví dụ như các lỗi và các cảnh báo về chất lượng ảnh hưởng tới sự thành công của dự án cũng như việc thiếu các chuyên gia, thiếu những nhà quản lý trong các đơn vị SQA...

c. Rà soát quản lý

Rà soát quản lý là cách gọi cuộc nhóm họp định kì cho phép người điều hành có

được một cái nhìn khái quát về sự quản lý chất lượng phần mềm của tổ chức của họ. Rà soát quản lý được thực hiện 1 hoặc 2 lần trong năm.

Một báo cáo rà soát quản lý, được chuẩn bị bởi đơn vị quản lý chất lượng phần mềm, quyết định giai đoạn để thảo luận bằng cách cung cấp các điều khoản, nó được thể hiện trên đề tài thảo luận của cuộc họp.

Những mục đích chính của việc rà soát quản lý để đánh giá sự chấp nhận của hệ

thống SQA với chính sách chất lượng của tổ chức, là để:

- Đánh giá thành tựu của những mục đích chất lượng xây dựng cho hệ thống quản lý chất lượng phần mềm của tổ chức.
- Khởi đầu những sửa đổi và cải tiến của hệ thống quản lý chất lượng phần mềm và những mục tiêu của nó.
- Phác thảo những sự hướng dẫn để bù đắp những thiếu hụt SQA chính và những vấn đề quản lý chất lượng phần mềm.
- Phân bổ những nguồn bổ sung cho hệ thống quản lý chất lượng phần mềm.

Những sự giải quyết làm cho việc rà soát quản lý hiện thời đạt được những kinh nghiệm để chỉ dẫn và điều khiển hoạt động của hệ thống quản lý chất lượng phần mềm cho giai đoạn tiếp theo, kết thúc ở việc rà soát kế tiếp.

17.2.2. Những trách nhiệm quản lý phòng ban

Những trách nhiệm đảm bảo chất lượng của giám đốc quản lý bộ phận bao gồm:

Quản lý hệ thống SQA(các công việc liên quan đến hệ thống chất lượng) và những công việc liên quan đến dự án và các dịch vụ được thực hiện bởi các đơn vị hay các đội dưới

quyền hạn xác định của nhà quản lý (các công việc liên quan đến dự án)

Những trách nhiệm liên quan đến hệ thống chất lượng:

Bao gồm những hoạt động đảm bảo chất lượng:

- Chuẩn bị chương trình các hoạt động SQA và ngân sách hàng năm dựa trên chương trình được đưa ra và được chuẩn bị bởi bộ phận SQA.
- Chuẩn bị các kế hoạch phát triển hệ thống SQA dựa vào kế hoạch được đưa ra và được chuẩn bị bởi bộ phận SQA.
- Điều khiển hiệu năng của chương trình hoạt động SQA hàng năm và các dự án phát triển.
- Trình bày những vấn đề SQA cho người quản lý cấp cao-người chịu trách nhiệm thực thi đối với chất lượng phần mềm.

Những trách nhiệm liên quan đến dự án

Những trách nhiệm này thay đổi theo các thủ tục của tổ chức và sự phân quyền, gồm:

1. Điều khiển công việc để thỏa mãn những thủ tục đảm bảo chất lượng trong các đơn vị của bộ phận này, bao gồm CAB, SCM và SCCA.
2. Chi tiết những kết quả rà soát hợp đồng và tán thành đề xuất.
3. Rà soát hiệu năng của các hoạt động rà soát đã được lên kế hoạch, tán thành những tài liệu của dự án và hoàn thành pha dự án.
4. Xác định việc kiểm tra phần mềm và kết quả kiểm tra; chấp nhận sản phẩm phần mềm.
5. Xác định sự tiến triển của các kế hoạch lên lịch biểu của dự án phát triển phần mềm và những chênh lệch về ngân sách.
6. Ý kiến đóng góp và hỗ trợ những người quản lý dự án trong việc quyết định lên lịch biểu, ngân sách và những khó khăn có liên quan đến khách hàng.
7. Xác định chất lượng của các dịch vụ bảo trì.
8. Chi tiết các rủi ro của dự án và các biện pháp để giải quyết những rủi ro đó.
9. Sự phù hợp của dự án với yêu cầu của khách hàng và thỏa mãn sự hài lòng của khách hàng.
10. Chấp nhận những yêu cầu thay đổi lớn trong phần mềm và sự chênh lệch từ đặc tả dự án.

17.2.3. Những trách nhiệm quản lý dự án

Trong chỉ thị công việc. Người quản lý dự án là người có trách nhiệm đảm bảo rằng tất cả các thành viên của đội tuân theo các thủ tục và các tài liệu đã được xây dựng. Công việc của người quản lý dự án bao gồm những tác vụ thực hành chuyên nghiệp và quản lý các công việc.

Những tác vụ thực hành chuyên nghiệp

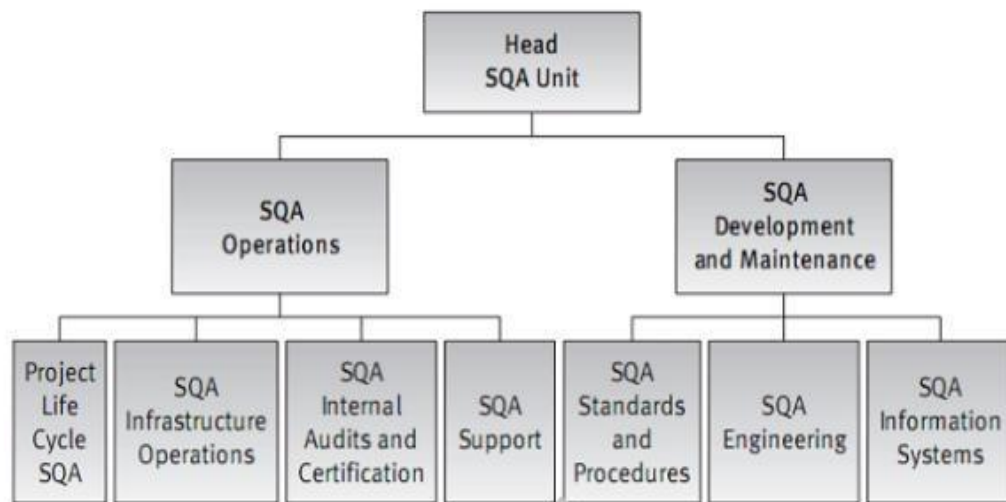
1. Chuẩn bị, cập nhật dự án và các kế hoạch chất lượng.
2. Tham gia trong ban kết nối giữa khách hàng-nhà cung cấp
3. Chốt lại vấn đề nhân sự trong đội dự án, bao gồm việc tham gia vào tuyển dụng, huấn luyện và truyền đạt kiến thức

Những tác vụ quản lý:

Người quản lý dự án phải chỉ ra được:

- Việc thực thi các hoạt động rà soát và những điều chỉnh, bao gồm việc tham gia vào một vài rà soát.
- Sự phát triển phần mềm và bảo trì ở khía cạnh phát triển, tích hợp và các hoạt động kiểm thử hệ thống như những hoạt động sửa chữa và kiểm thử hồi quy.
- Thực hiện acceptance tests.
- Cài đặt phần mềm cho khách hàng và chạy hệ thống phần mềm
- Huấn luyện và chỉ dẫn đảm bảo chất lượng phần mềm cho các thành viên của dự án.
- Lên lịch biểu và các tài nguyên phân phối cho các hoạt động của dự án.
- Các yêu cầu của khách hàng và sự hài lòng của khách hàng
- Luận ra những rủi ro trong sự phát triển của dự án, áp dụng những giải pháp và điều khiển những rủi ro.

17.3. Đơn vị SQA và các tác nhân khác trong hệ thống SQA



Mô hình cấu trúc tổ chức đơn vị SQA

17.3.1. Đơn vị SQA

Nhiệm vụ cơ bản của trưởng bộ phận SQA có thể được phân thành các loại sau:

■ Lập kế hoạch

- Chuẩn bị các chương trình hoạt động và chi phí dự toán thường niên cho bộ phận
- Lập kế hoạch và cập nhật hệ thống quản lý chất lượng phần mềm của tổ chức
- Chuẩn bị các chương trình hoạt động SQA thường niên được khuyến nghị cho các bộ phận phát triển và bảo trì phần mềm; yêu cầu các sự hỗ trợ từ các bộ phận SQA con khi thực hiện nhiệm vụ này.

■ Quản lý bộ phận:

- Quản lý các hoạt động của đội SQA
- Giám sát thực thi chương trình hoạt động SQA
- Chỉ định, bổ nhiệm các thành viên nhóm, các thành viên ủy ban SQA và các ủy viên SQA
- Chuẩn bị các báo cáo đặc biệt và định kỳ, ví dụ: tình trạng của các vấn đề chất lượng phần mềm trong tổ chức và các báo cáo thi hàng tháng.

■ Các nhiệm vụ liên quan đến giao dịch với khách hàng các tổ chức bên ngoài khác:

- Các địa chỉ khách hàng cho các vấn đề trong chất lượng phần mềm
- Liên lạc với khách hàng, quan tâm đến chất lượng các sản phẩm và dịch vụ phần mềm đã cung cấp.

- Đại diện cho tổ chức trước các tổ chức bên ngoài khác có quan tâm đến các vấn đề về chất lượng phần mềm
- Dự thảo các báo cáo về quản lý cho các cuộc họp đánh giá công tác quản lý.
- Các vấn đề về tổ chức SQA, chuẩn bị các tài liệu cần thiết... cho các quyết định mức cao về quản lý (được thực hiện bởi trưởng bộ phận chất lượng phần mềm).
- Các công việc SQA chuyên ngành:
 - Tham gia vào dự án joint committees
 - Tham gia đánh giá thiết kế chính thức
 - Đánh giá và phê chuẩn sự sai khác so với đặc tả (khi các thủ tục yêu cầu)
 - Bàn bạc với các nhà quản lý dự án và các trưởng nhóm.
 - Tham gia vào các diễn đàn và ủy ban SQA

17.3.2. Những ủy viên SQA và nhiệm vụ

Nhiệm vụ cơ bản của SQA trustee có thể được phân thành các loại sau:

Ủy viên SQA là những nhân viên, là những người được quan tâm rất nhiều trong chất lượng của phần mềm, họ tình nguyện dành một phần thời gian của họ để nâng cao chất lượng. Họ thường xuyên hướng dẫn các chủ đề được quan tâm của SQA. Như là các đại diện SQA, ủy viên được kỳ vọng đưa các hỗ trợ nội bộ cần thiết cho việc cài đặt thành công các thành phần của SQA.

Các nhiệm vụ của SQA ủy khác nhau với mỗi tổ chức.

Những nhiệm vụ có thể liên hệ giữa các đơn vị và/hoặc các tổ chức, và bao gồm các hoạt động sau đây

17.3.2.1. Các công việc liên quan đến đơn vị

- Hỗ trợ các đồng nghiệp để giải quyết những khó khăn phát sinh trong việc triển khai các thủ tục phần mềm chất lượng cũng như các chỉ thị công việc
- Trợ giúp các đơn vị quản lý trong hoạt động SQA của mình (ví dụ như chuẩn bị 1 dự án hướng dẫn làm việc , tập hợp các dữ liệu để tính toán độ đo SQA)
- Thúc đẩy việc tuân thủ và giám sát công việc triển khai thực hiện các thủ tục SQA, hướng dẫn và làm cùng đồng nghiệp
- Báo cáo sự không tuân thủ có hệ thống các sự kiện tới đơn vị SQA
- Báo cáo chất lượng phần mềm thất bại nghiêm trọng tới đơn vị SQA.

17.3.2.2. Các công việc liên quan để cơ cấu tổ chức

- Tiến hành thay đổi và cập nhật các thủ tục và chỉ thị công việc của organization-wide SQA
- Tiến hành cải tổ các chức năng, phát triển và cải tiến quy trình và bảo trì các ứng dụng vào CAB cho các giải pháp giả quyết những thất bại quan sát được trong đơn vị của họ

- Xác định tổ chức SQA cần sự huấn luyện phù hợp và các đề xuất 1 chương trình huấn luyện hay chỉ dẫn thích hợp được thực hiện bởi đơn vị SQA.

17.3.3. Hội đồng SQA và nhiệm vụ

Các uỷ ban SQA có thể được hoặc thường trực hoặc đặc biệt. Các chủ đề cần được xử lý thẩm quyền cũng như phân chia nhiệm vụ giữa các uỷ ban thường trực và đặc biệt hoặc giữa các tổ chức khác nhau và theo thời gian

17.3.3.1.Ủy ban thường trực:

- Thường giải quyết các SCC (điều khiển thay đổi phần mềm), CA (các công việc sửa chữa), các thủ tục các công cụ phát triển phần mềm, các thước đo chất lượng.
- Là các bộ phận kết hợp của khung làm việc của tổ chức SQA
- Công việc của uỷ ban thường trực thường được quy định trong các thủ tục SQA của tổ chức

17.3.3.2.Ủy ban không chính thức (ad-hoc committee):

- Được thành lập dựa trên mỗi vấn đề ngắn hạn
- Các thành viên được chỉ định bởi người điều hành có trách nhiệm giải quyết các vấn đề về chất lượng phần mềm (trưởng bộ phận SQA, các bộ phận SQA con, các uỷ viên SQA...)

17.3.4. Nhiệm vụ và phương thức hoạt động của diễn đàn SQA

Đặc điểm

Những diễn đàn SQA là thành phần không chính thức của khung tổ chức SQA, chúng được thành lập bởi những người tự nguyện và thể hiện những đặc điểm của một cộng đồng.

Các diễn đàn thường hoạt động một cách tự do mà không bị lệ thuộc vào bất cứ yêu cầu hoặc thủ tục tiêu chuẩn nào. Những chủ đề, hoạt động và người tham gia vào diễn

đàn rất đa dạng. Thông thường một tổ chức thu lợi từ những hoạt động của diễn đàn SQA

mà diễn đàn có thể hoạt động độc lập hoặc có liên hệ cộng tác với tổ chức.

Những thành viên của diễn đàn SQA thường định ra phạm vi và chức năng hoạt động (hạn chế hoặc rộng lớn), gặp gỡ thường xuyên hoặc không thường xuyên, và có thể định ra những cách thức liên lạc ưa thích (Internet, intranet, ...)

Những diễn đàn SQA tập trung chính vào:

- Cải thiện và triển khai thủ tục SQA
- Những độ đo chất lượng
- Hoạt động sửa lỗi – phân tích những trường hợp thất bại và thành công
- Những vấn đề về chất lượng hệ thống – phát triển và triển khai những công cụ mới
- Những vấn đề về quản lý chất lượng – vấn đề chất lượng phần mềm hoạt động thường ngày được bàn đến trước bởi người quản lý chất lượng ở mỗi mức độ.

Sự tham gia và các diễn đàn SQA có thể là đóng (ví dụ giới hạn với những người quản lý chất lượng) hoặc mở với tất cả mọi người. Thành viên của một diễn đàn mở có thể bao gồm:

- Thành viên đơn vị SQA
- Ủy viên quản trị SQA
- Nhân viên phát triển và bảo trì phần mềm
- Tư vấn/ chuyên gia SQA và công nghệ phần mềm
- Đại diện khách hàng.

Diễn đàn cũng duy trì sự lựa chọn công bố. Sự công bố có thể bao gồm thư đến các thành viên, định kỳ xem xét lại những vấn đề SQA, báo cáo của những nhóm chuyên gia hoặc thành viên diễn đàn đặc biệt. Thêm vào việc mô tả và phân tích một vấn đề chất lượng, bản báo cáo có thể bao gồm sự tư vấn cho hoạt động sửa lỗi. Diễn đàn cũng quyết định dựa trên một danh sách phân tán, và sự công bố được giới hạn trong những thành viên hay mở rộng đến những thành viên khác trong tổ chức hay không?

BÀI 18. THỰC HÀNH KHẢO SÁT SQA PLAN

18.1. Mục tiêu :

- Giúp SV hiểu và nắm vững cấu trúc của SQA plan và các nội dung cần có của 1 SQA plan.
- Giúp SV lập được SQA plan đơn giản

18.2. Nội dung :

Đọc SQAplan giáo viên cung cấp, ghi nhận lại các thông tin được trình bày trong từng testplan theo dạng bảng như sau (dùng Excel) :

	A	B	C	D	E	F	G	H	I
1	STT	Nội dung	TP1	TP2	TP3	TP4	TP5	TP6	TP7
2	1	Nội dung 1	x	x		x		x	
3	2	Nội dung 2		x	x		x	x	x
4	3	Nội dung 3	x		x	x		x	
5	4	Nội dung 4		x		x	x		x
6	5	Nội dung 5	x	x	x	x	x	x	x
7	6	...							

18.3. Bình luận kết quả thu được.

- Dựa vào kết quả thu được, xây dựng SQA plan cho Project 3 mà các em đang thực hiện.

18.4. Chuẩn đầu ra :

- Sinh viên nắm vững và chủ động xây dựng SQA plan cho hệ thống phần mềm cần đảm bảo chất lượng

18.5. Qui trình :

1. Mở từng file miêu tả SQAplan cho dự án phần mềm; duyệt từ đầu file đến cuối file. Nhận dạng từng mục thông tin mà SQAplan đề cập, đưa vào danh sách Excel nếu mục thông tin này chưa có, đánh dấu x vào cell tương ứng ở cột miêu tả SQAplan đang khảo sát (*các file đính kèm*).
 2. Khảo sát lại bảng kết quả xây dựng được đưa ra bình luận về cấu trúc và nội dung của từng SQAplan.
 3. Xây dựng SQAplan cho Project 3, hoặc hệ thống quản lý phòng trọ trong bài tập lớn đã miêu tả
- Tham khảo :
SQA Planning
☐ Planning is one of the most important aspects of Software Quality Assurance. The entire operation of the SQA team depends on how well their planning is done. In smaller businesses, planning might not really dictate the flow of SQA but in larger businesses, SQA Planning takes on center stage. Without it, each component or department that works on the application will be affected and will never function.

- In gist, SQA Planning tackles almost every aspect of SQA's operation. Through planning, each member and even non-member of the SQA team is clearly defined. The reason for this is very simple: when everyone knows their role and boundaries, there is no overlapping of responsibilities and everyone could concentrate on their roles.
- But SQA Planning is not only a document that tells who gets to do the specific task. The stages in are also detailed. The whole SQA team will be very busy once the actual testing starts but with SQA, everyone's work is clearly laid out. Through planning, the actual state of the application testing is known.
- Again in smaller businesses, the planning maybe limited to the phase of the application testing but when outlined for corporations, the scenario changes and only through planning that everyone will know where they are and where they are going in terms of SQA.
- SQA Planning is not just a simple document where objectives are written and stages are clearly stated. Because of the need to standardize software development ensuring the limitation of error, a scientific approach is recommended in developing an SQA plan. Certain standards such as IEEE Std 730 or 983.

[SQA Plan Content](#)

- An SQA Plan is detailed description of the project and its approach for testing. Going with the standards, an SQA Plan is divided into four sections:
 - Software Quality Assurance Plan for Software Requirements;
 - Software Quality Assurance Plan for Architectural Design;
 - Software Quality Assurance Plan for Detailed Design and Production and;
 - Software Quality Assurance Plan for Transfer
- In the first phase, the SQA team should write in detail the activities related for software requirements. In this stage, the team will be creating steps and stages on how they will analyze the software requirements. They could refer to additional documents to ensure the plan works out.
- The second stage of SQA Plan or the SQAP for AD (Architectural Design) the team should analyze in detail the preparation of the development team for detailed build-up. This stage is a rough representation of the program but it still has to go through rigorous scrutiny before it reaches the next stage.
- The third phase which tackles the quality assurance plan for detailed design and actual product is probably the longest among phases. The SQA team should write in detail the tools and approach they will be using to ensure that the produced application is written according to plan. The team should also start planning on the transfer phase as well.
- The last stage is the QA plan for transfer of technology to the operations. The SQA team should write their plan on how they will monitor the transfer of technology such as training and support.
- {mospagebreak title=SQA Planning Standards}

[SQA Planning Standards](#)

- A special section for standards should be dedicated in SQA Planning. Before anything else the “rules” for checking the application should be laid out. When the standards are laid out, everyone will have a reference for checking the application.
- The following are the standards that should be laid out in SQA Planning:
- **Documentation Standards** – The SQA team should establish the standards on how the documents should be built.
- **Design Standards** – The SQA team should identify the design and approach the developers will be using in developing the application. Since they are not the once who will be determining this standards, they have to work closely with developers.
- **Coding Standards** – The developers will inform the SQA team regarding the coding. Using this, the SQA will know if the application was built according to the actual code.
- **Commentary Standards** – A special section should be dedicated to the comments inserted in the code.
- **Testing Standards** – The software and tools that will be used to test the application is written here. Aside from the software and tools, the approach on how the tests will be conducted is also laid out.
- **Metrics** – The measurement that will be sought after the application is detailed here. The SQA team should clearly lay out what are the metrics they will be using in SQA.
- **General Approach** – Lastly, the SQA team will write in general how they will monitor the development of the application to ensure compliance with the expected standards.

SQL Plan Style, Responsibility and Medium

- The SQA Plan is basically a report that tells the clients who will receive the application what the SQA team will do to ensure that the application will live up to their expectations or maybe more. The output of SQA Planning is a written report distributed to the client made by the SQA team of the software development company.
- The responsibility of creating the report is of course made by the SQA team. Depending on who handles the SQA for the application, the client should be able to communicate effectively to the SQA team so that they can update them regarding their expectations from the application.
- The written report should be distributed either through paper or electronic transmission. The client and the development team should have a copy of the SQA plan. Because of this, the client will have a clear view of what to expect from the application. Developers on the other hand will also have an idea on what will be the actual benchmark in developing the right application.
- Through the detailed SQA Plan, everyone involved in the project will know in advance how the application will work in the actual environment. From the planning to the technology transfer stage, the SQA team will set the benchmark

on what the application should be. Through the detailed plan, everyone will know what to do and what to expect from the application if they are done according the written plan.

BÀI 19. THỰC HÀNH QUẢN LÝ CẤU HÌNH PHẦN MỀM (1)

19.1. Mục tiêu :

- Trình bày được cách quản lý cấu hình phần mềm (SCM) cho một dự án
- Sử dụng được công cụ TortoiseSVN và Visual SVN để quản lý được code của một dự án

19.2. Nội dung :

1. Cài đặt VisualSVN lên server và cách tạo Project

Download VisualSVN để cài đặt lên server: <http://www.visualsvn.com/server/>

2. . Cài đặt và cách get code trên máy Client

Download Tortoise SVN: <http://tortoisesvn.net/downloads.html>

19.3. Chuẩn đầu ra :

Sinh viên nắm vững và chủ động được kho chứa và quản lý được sự thay đổi của các phiên bản trong một dự án phần mềm

19.4. Qui trình :

1. Thực hiện các bước trong tài liệu hướng dẫn sử dụng SVN
2. Xây dựng được một kho chứa và thực hiện các thao tác thay đổi cho một dự án cụ thể

*Tham khảo

Hướng dẫn sử dụng Subversioning (SVN) dựa trên **TortoiseSVN** client.

- Mục tiêu chính của SVN :

Quản lý các phiên bản của một file nào đó (bất kỳ là file text, binary,...) vào server SVN. Quá trình kiểm tra sẽ được SVN sẽ kiểm tra tại 2 nơi

1. Thư mục dự án (Nơi thực hiện dự án).
2. Thông tin dự án được lưu trữ trên server.

Khi kiểm tra file hiện hành có sự thay đổi so với trên server SVN là SVN sẽ tạo một phiên bản mới lưu lại(khi người dùng update -> commit). Việc quản lý phiên bản sẽ giúp việc phục hồi lại các version trước đó dễ dàng mà không cần phải lưu backup tùm lum mà không biết cái nào trước cái nào sau ... Khi lưu từng phiên bản sẽ cho phép lưu kèm theo các thông tin liên quan như ...ai là người thay đổi , thay đổi chỗ nào, thay đổi cái gì ... giúp cho việc quản lý source nguồn một cách rõ ràng. Việc lưu trữ version SVN sự **dụng** các thuật toán so sánh và chỉ lưu trữ các thành phần khác nhau của mỗi version nên dung lượng lưu trữ sẽ được tối ưu so với việc backup từng file.

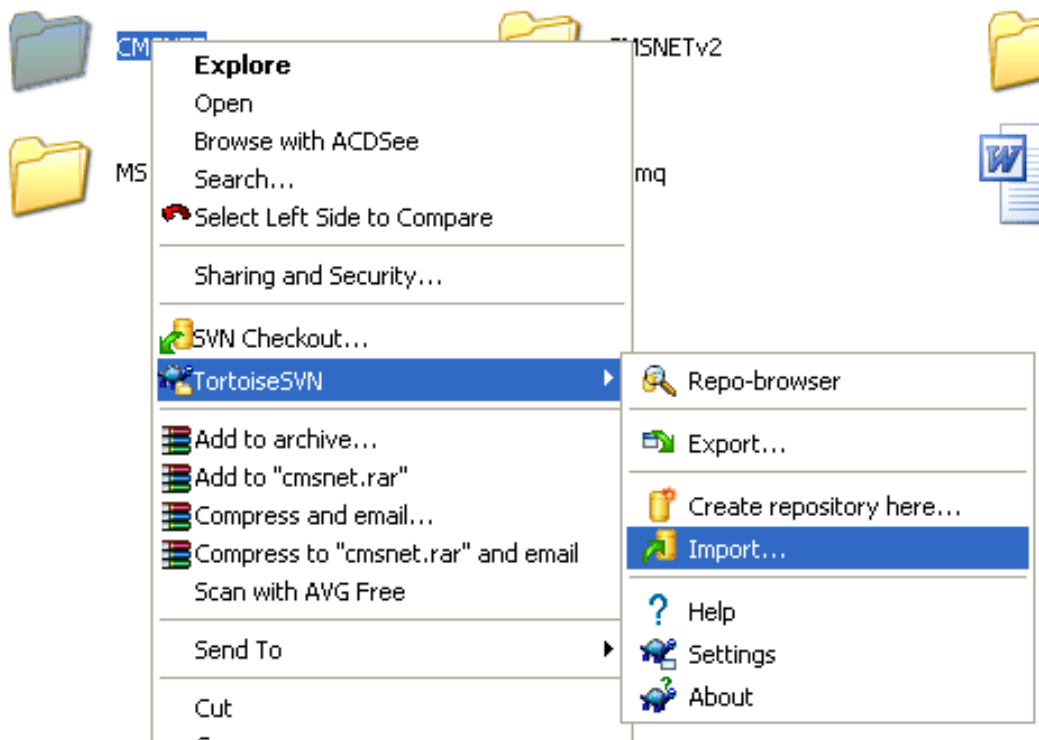
Các quy định khi làm việc với SVN:

- **Khi thực hiện mới dự án :** Nếu là một dự án mới thì quá trình thực hiện dự án như thường lệ, sau đó "Import" dự án vào SVN . Cách thực hiện như sau : right click trên thư mục dự án chọn **TortoiseSVN** → Import...

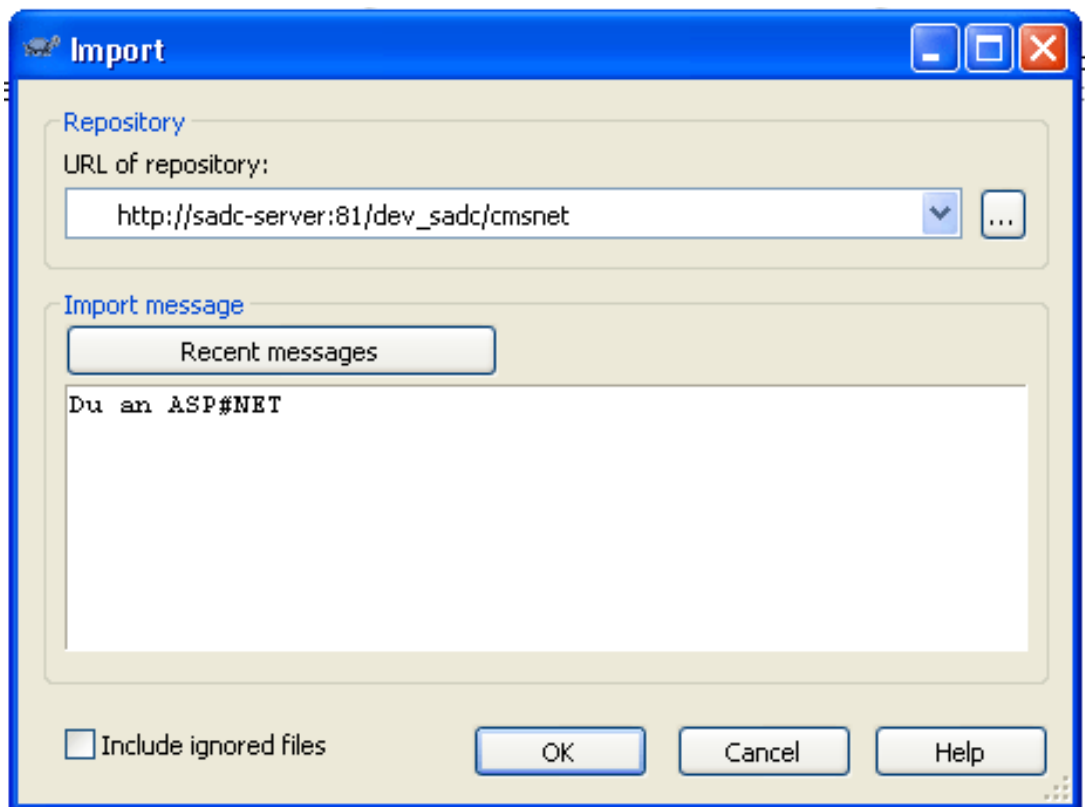
- **Khi thực hiện upgrade từ một dự án có sẵn trong SVN :** Nếu phải thực hiện lại một dự án có sẵn thì phải "checkout" dự án đó từ SVN về thư mục thực hiện dự án. để SVN khởi tạo các thông tin quản lý source nguồn trên thư mục làm việc của dự án. Cách thực hiện như sau : Right click trên thư mục (rỗng) , nơi dự định để thực hiện dự án **TortoiseSVN** → Checkout...

Một số thao tác cơ bản với SVN:

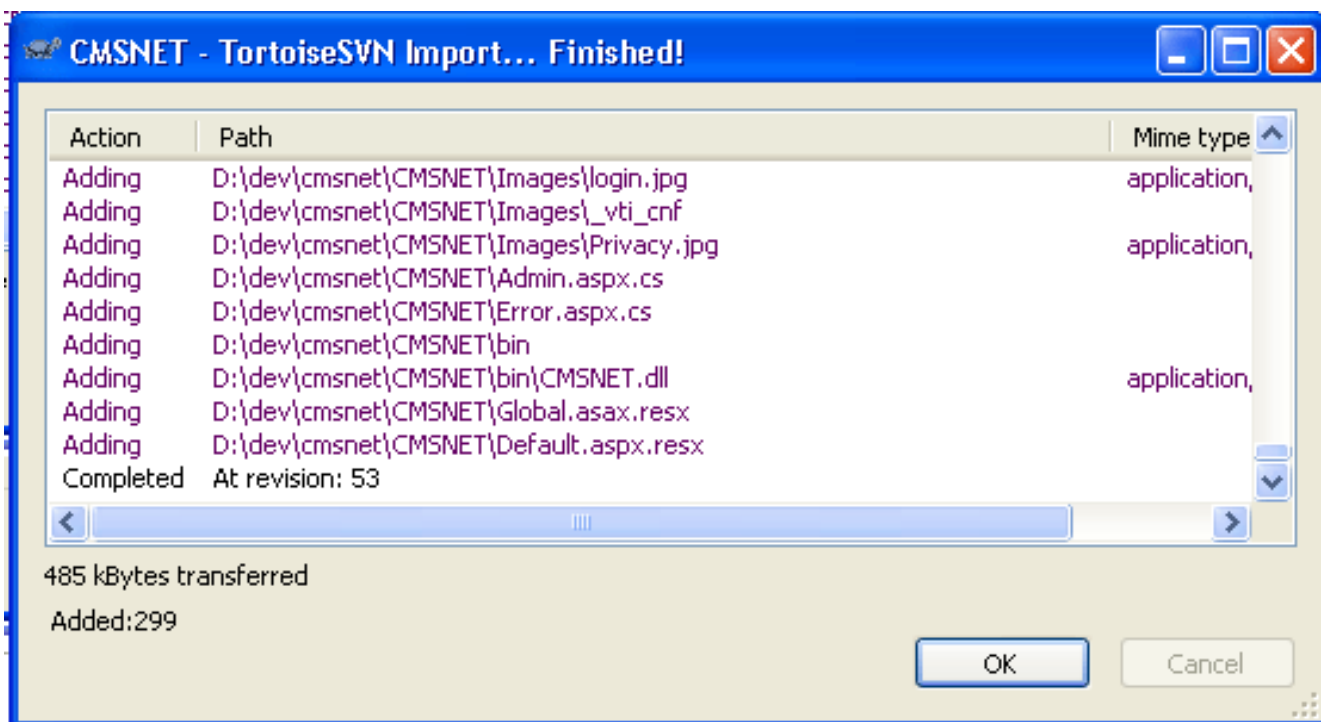
Đưa một project vào SVN server Mở windows explorer lên, chọn thư mục của dự án thực hiện "right click" và chọn **TortoiseSVN** → Import... Các bước như sau :
Bước 1: Chọn chức năng import 3



Bước 2: Điền các thông tin cần thiết (khi truy cập vào url server thì sẽ có dialog yêu cầu nhập thông tin username + password đã gửi)



Bước 3 : Ngồi đợi và chờkết thúc ...xong

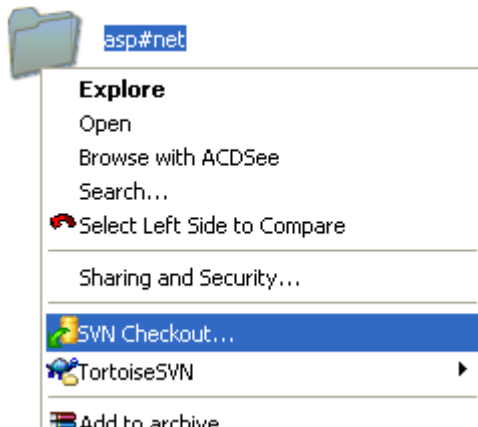


Lấy một project từ SVN

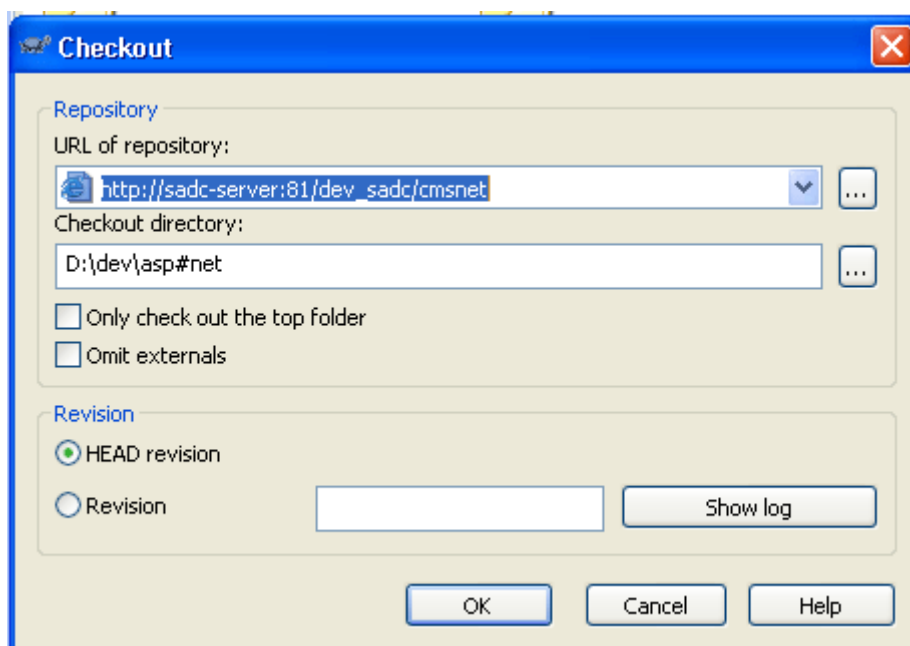
Khi cần lấy các source code từ server SVN thì thực hiện các bước như sau :

Mở windows explorer lên, chọn (hoặc tạo mới) một thư mục để chứa các source code của dự án (có thể là thư mục của webserver) @chú ý : thư mục phải rỗng .

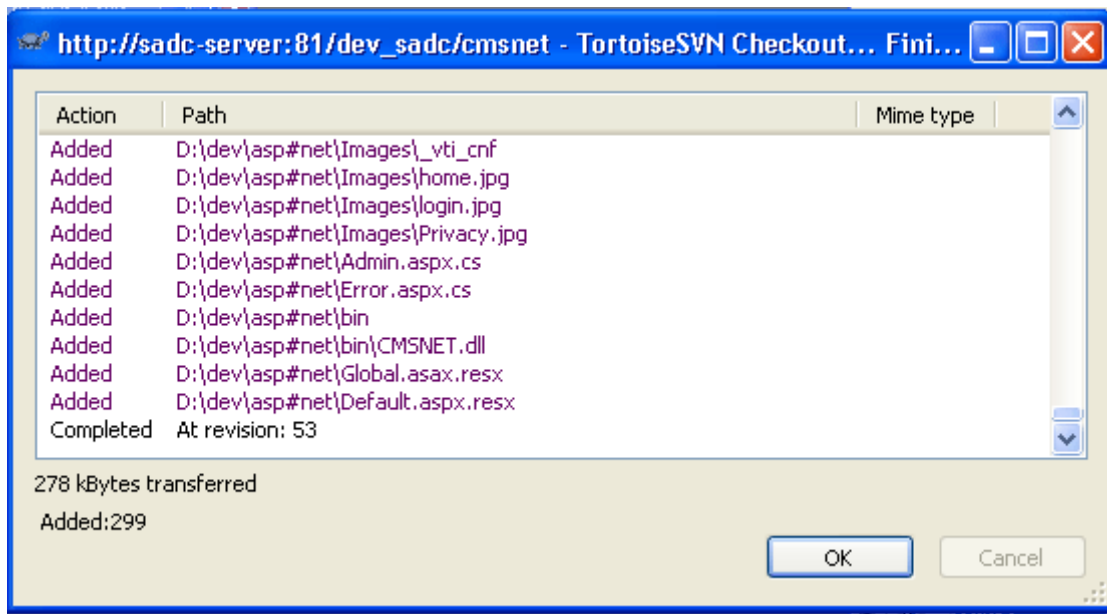
Bước 1 : chọn thư mục chứa dự án



Bước 2 : chọn thông tin dự án trên server:



Bước 3 : Ngồi đợi và ...ok thế là xong , bạn đã có các thông tin của dự án trên máy của mình



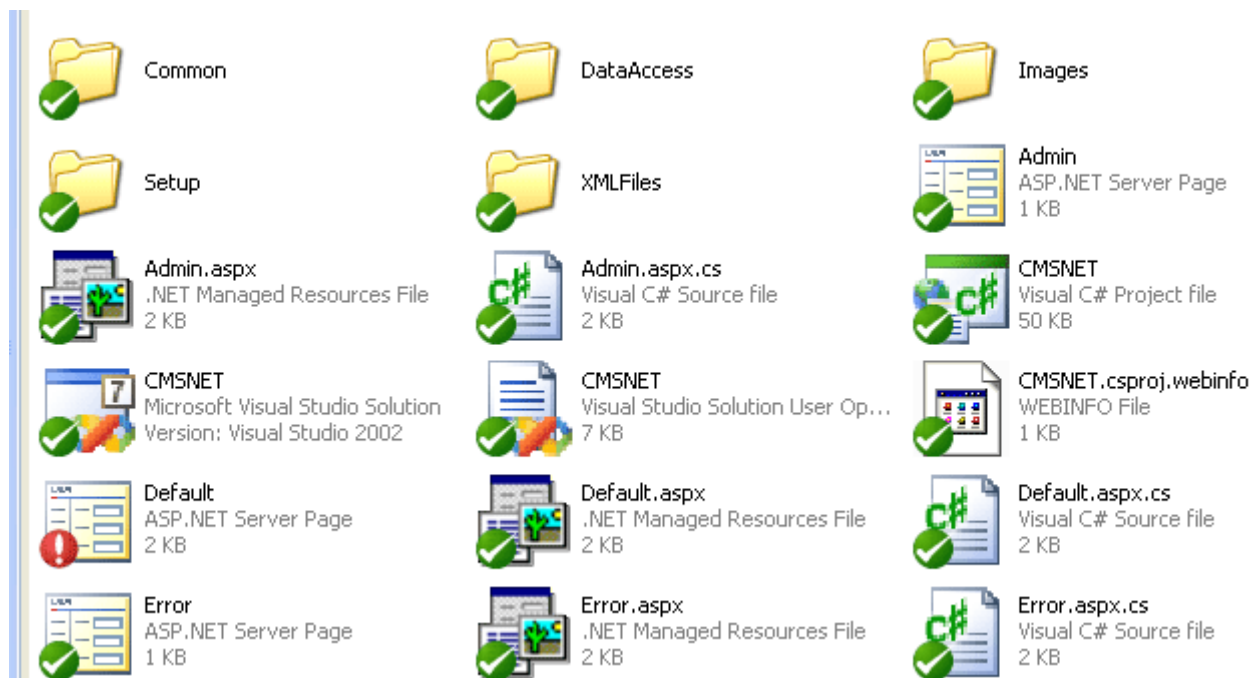
Kiểm tra các file bị thay đổi với server SVN Kiểm tra sự thay đổi của source nguồn
Nếu chưa có bất kỳ sự thay đổi nào thì icon của thư mục dự án như sau



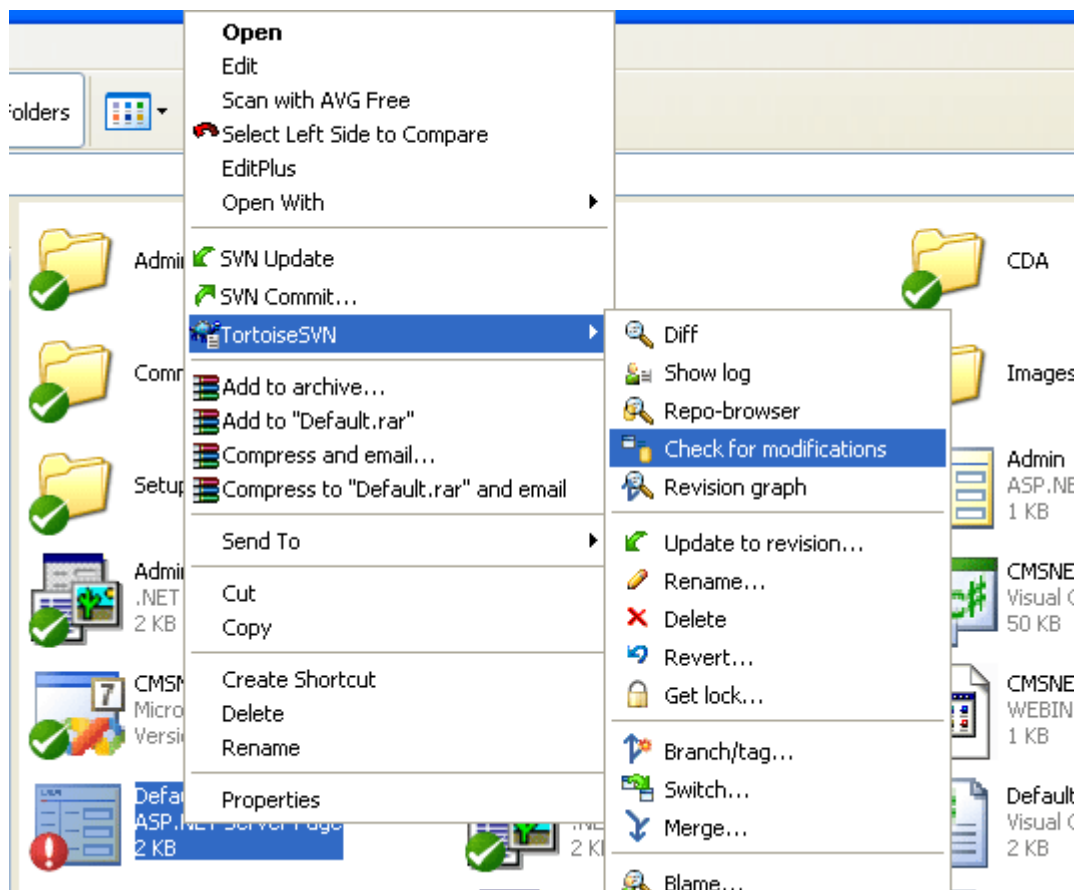
Nếu có một file, hay một thư mục nào có sự thay đổi thì icon của thư mục như sau



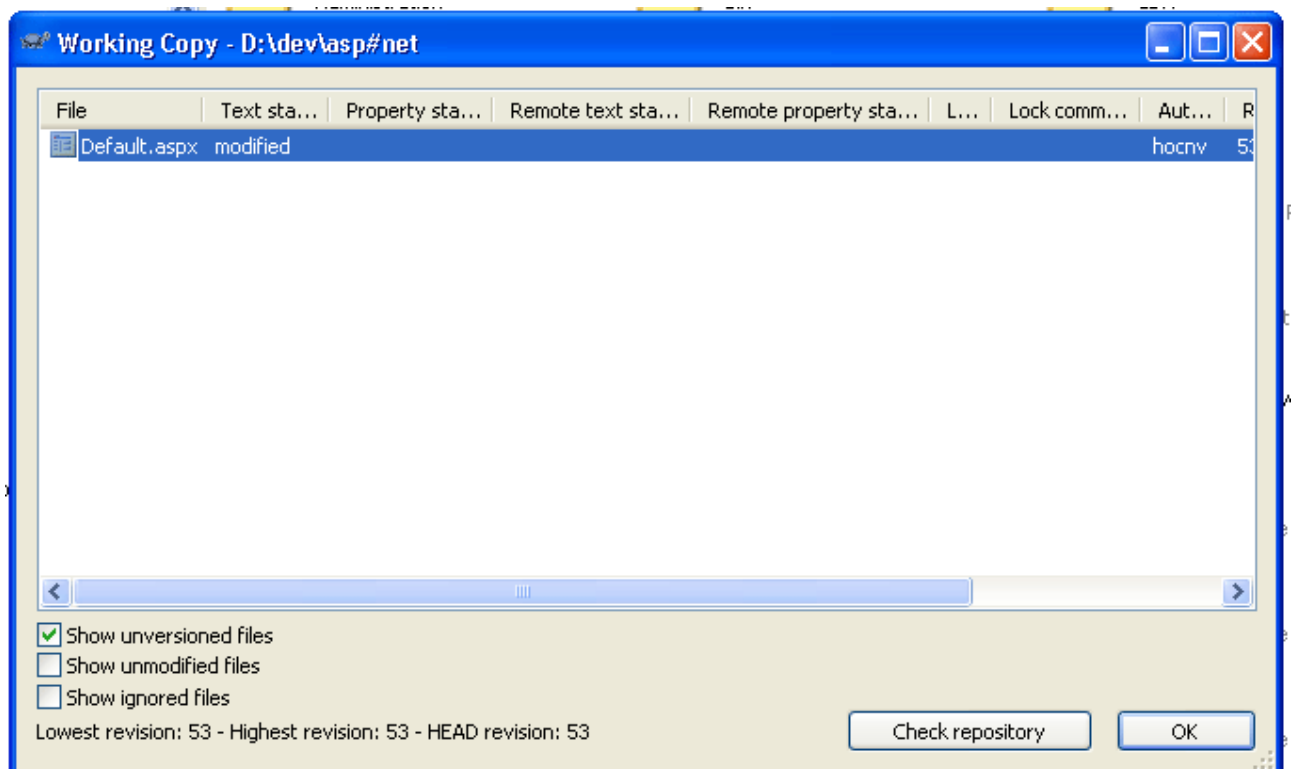
Cho biết file bị thay đổi



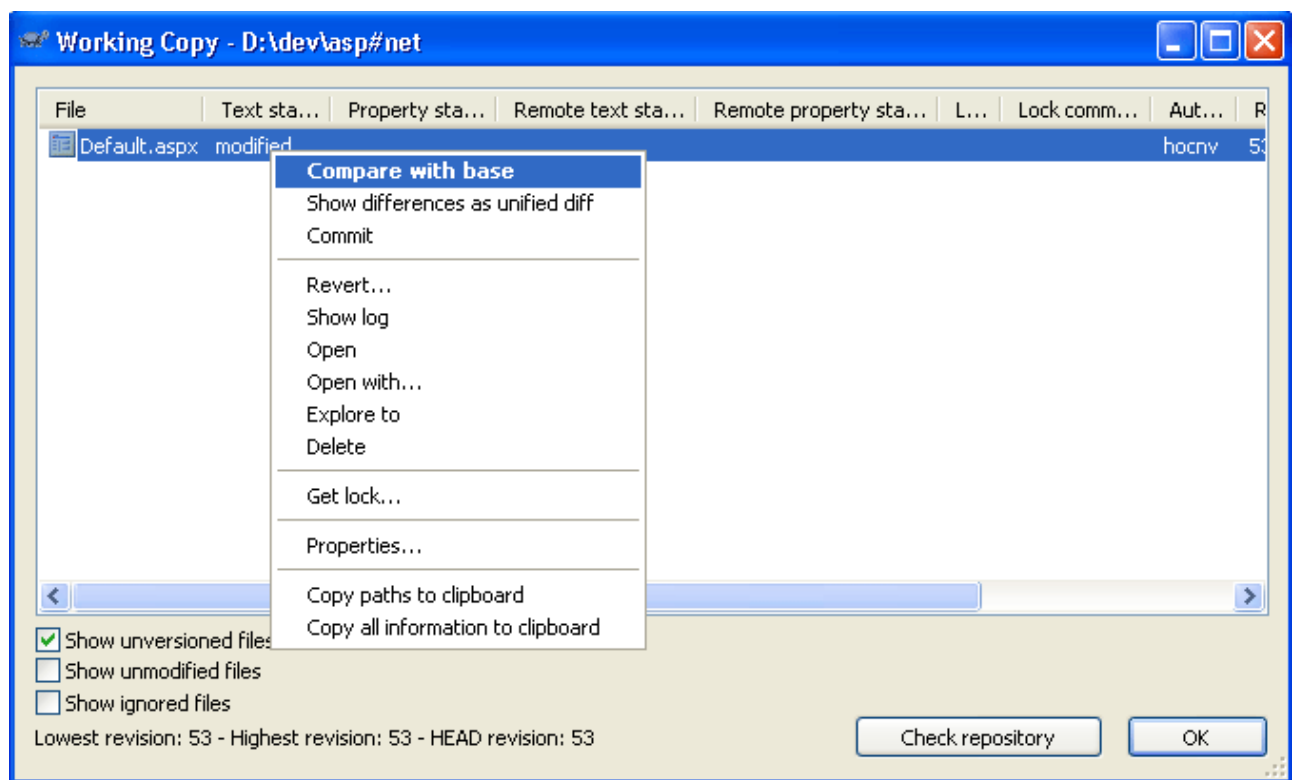
Xem file đó thay đổi nội dung gì



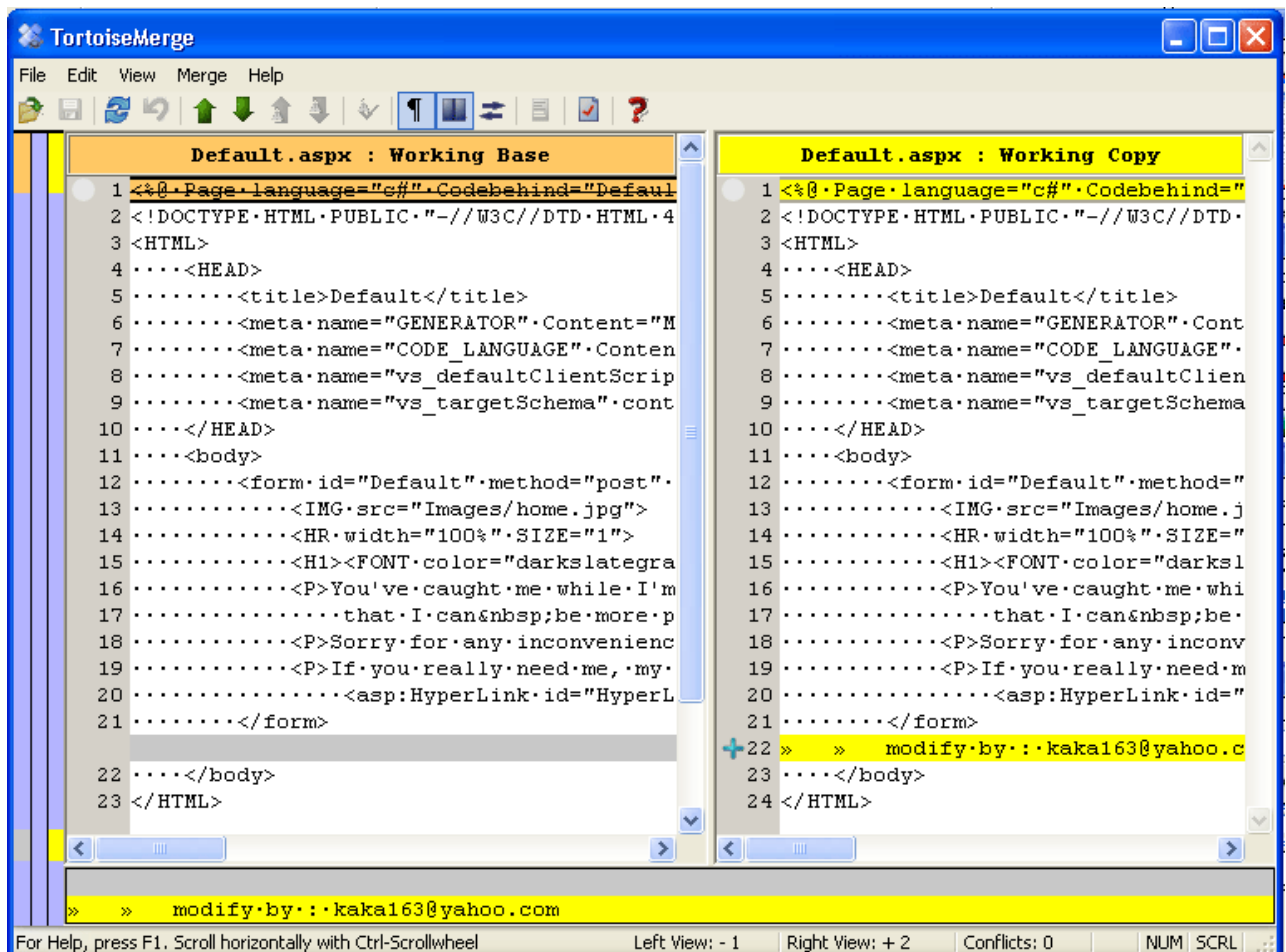
Thông tin thay đổi



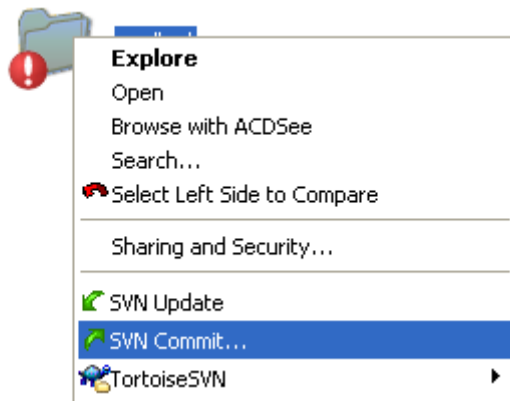
Right click ... tìm chức năng "Compare with base"



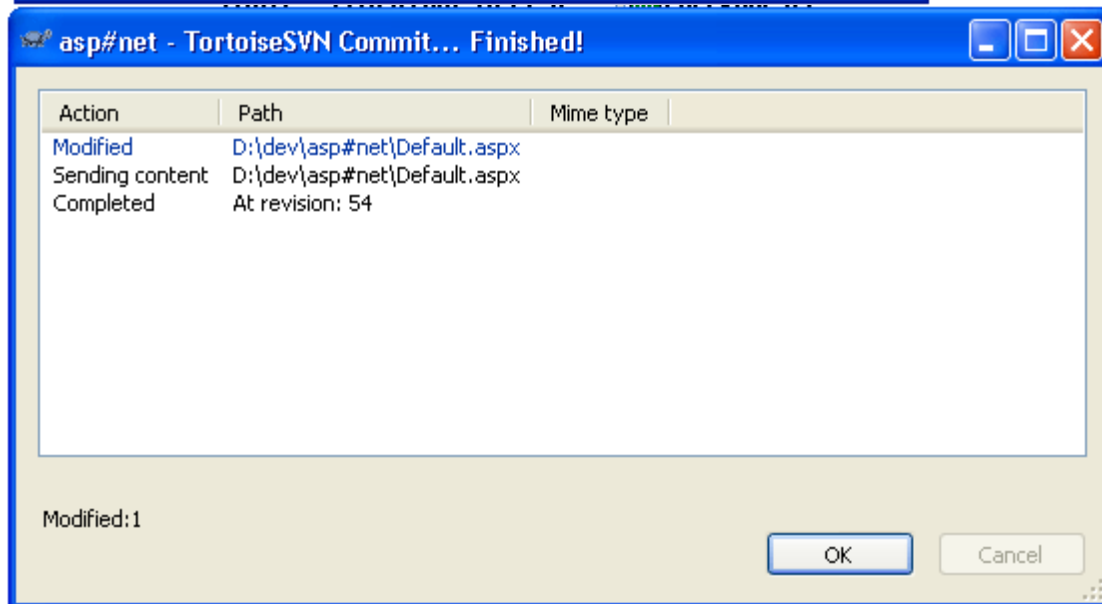
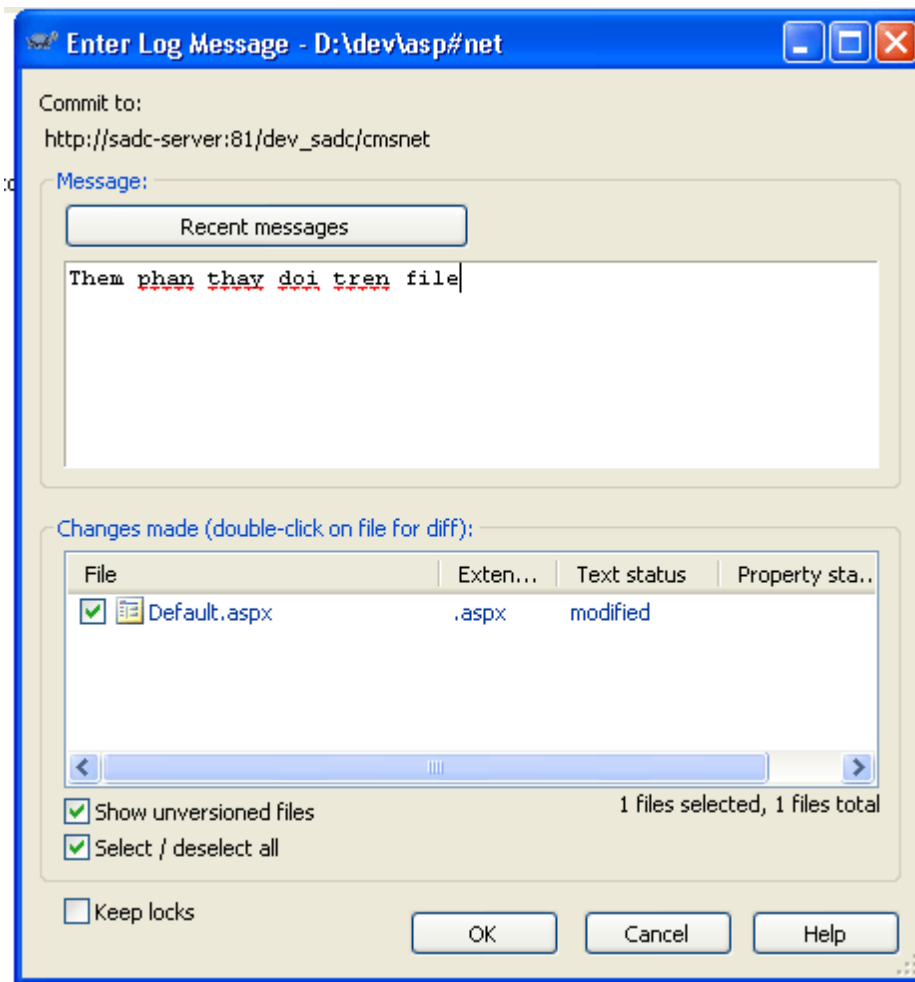
Tìm xem thử coi chi tiết là nó thay đổi cái gì ?



Vậy là biết cái file đó đã bị thay đổi như thế nào Giờ làm sao đưa thông tin thay đổi vào server SVN đây Dùng chức năng "Commit ..." để đưa thông tin vào server



Check vào các file cần commit, hoặc...bỏ bớt một số file chưa cần commit (nếu là nhiều file)



Sau khi commit các thông tin giữa clien và version mới nhất tại server hoàn toàn giống nhau thì icon của thư mục thư mục ..trở về như cũ.



shell



asp#net

BÀI 20. THỰC HÀNH QUẢN LÝ CẤU HÌNH PHẦN MỀM (2)

20.1 Mục tiêu :

- Trình bày được cách quản lý cấu hình phần mềm (SCM) cho một dự án
- Sử dụng được công cụ TortoiseSVN và Visual SVN để thêm mới các file vào dự án, lấy lại được các phiên bản, xử lý các tình huống hay phát sinh

20.2. Nội dung :

- Cài đặt VisualSVN lên server và cách tạo Project
Download VisualSVN để cài đặt lên server: <http://www.visualsvn.com/server/>
- Cài đặt và cách get code trên máy Client
Download Tortoise SVN: <http://tortoisesvn.net/downloads.html>
- Thêm file mới vào dự án
- Lấy lại các phiên bản
- Xử lý các tình huống

20.3 Chuẩn đầu ra :

Sinh viên nắm vững và chủ động được kho chứa và quản lý được sự thay đổi của các phiên bản trong một dự án phần mềm

20.4 Qui trình :

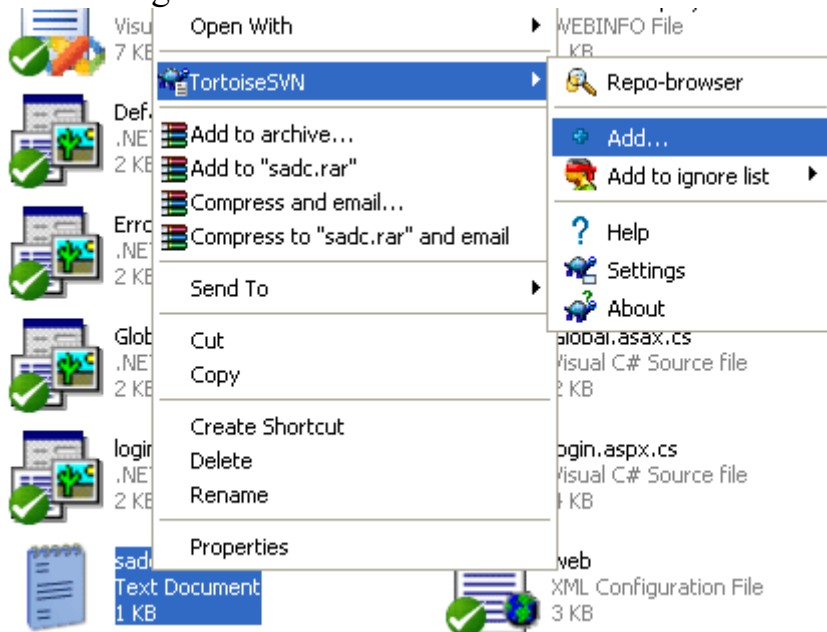
- Thực hiện các bước trong tài liệu hướng dẫn sử dụng SVN
- Xây dựng được một kho chứa và thực hiện các thao tác thay đổi cho một dự án cụ thể

*Tham khảo

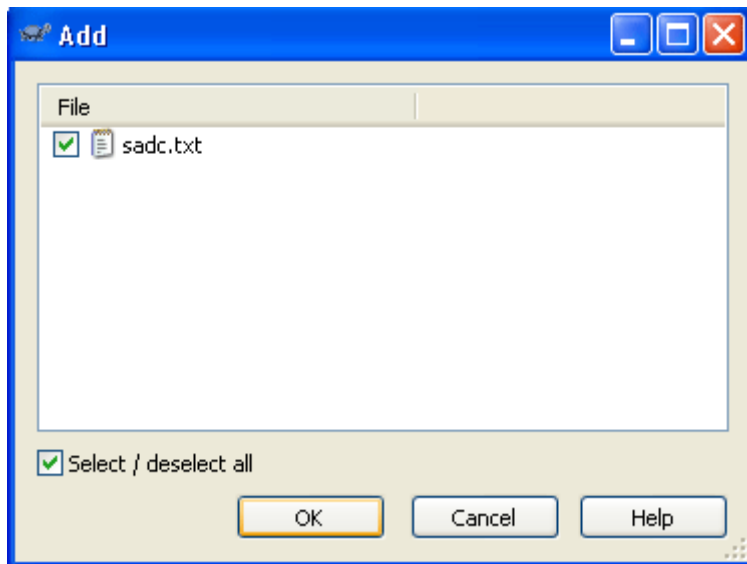
Đưa file mới vào SVN

Nếu trong quá trình phát triển dự án cần thêm một số file vào dự án thì sau khi thêm vào các bạn chọn chức năng "add" để đưa các file này vào SVN

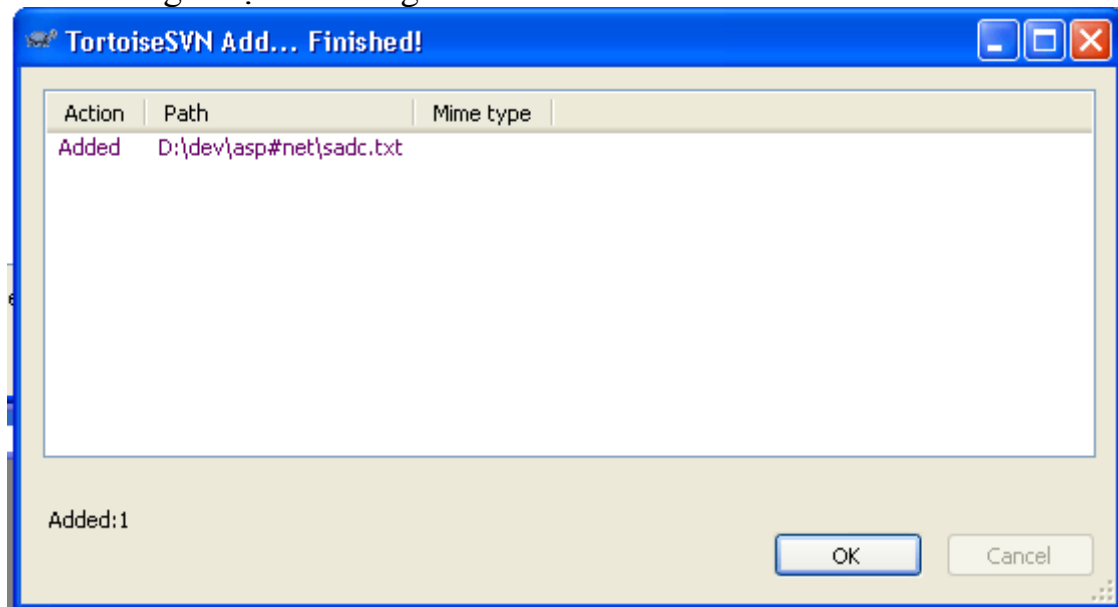
Bước 1: right click vào file chưa có icon SVN



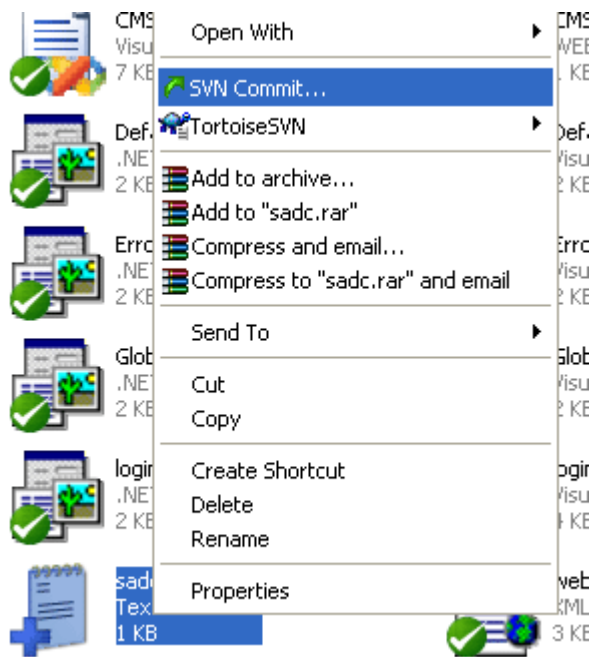
Bước 2: Chọn các file cần đưa nếu trogn bước 1 là một thư mục



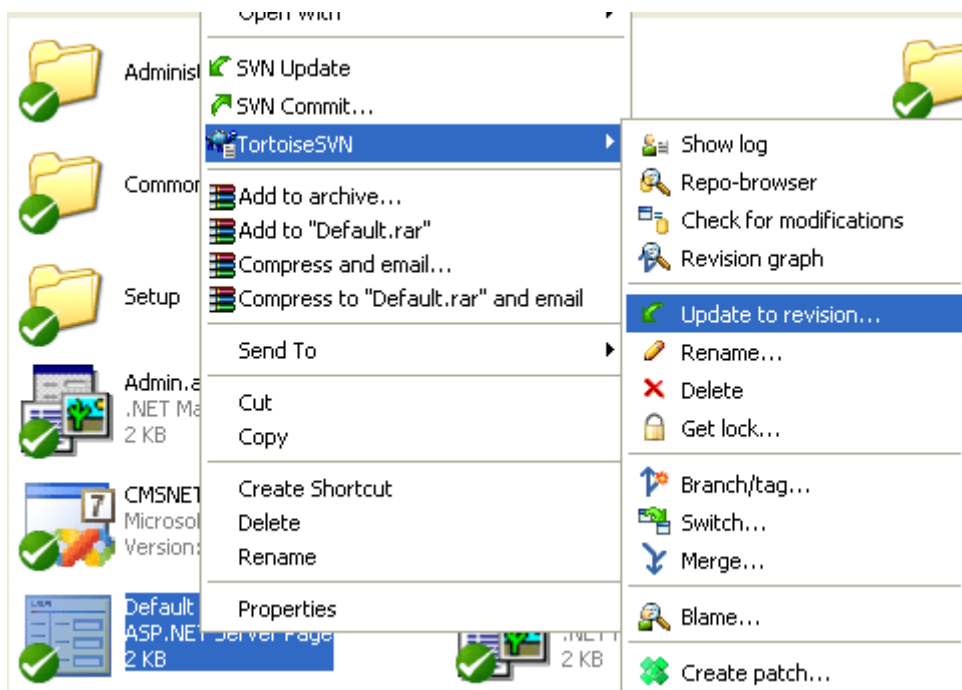
Bước 3: Ngồi đợi ...và xong .



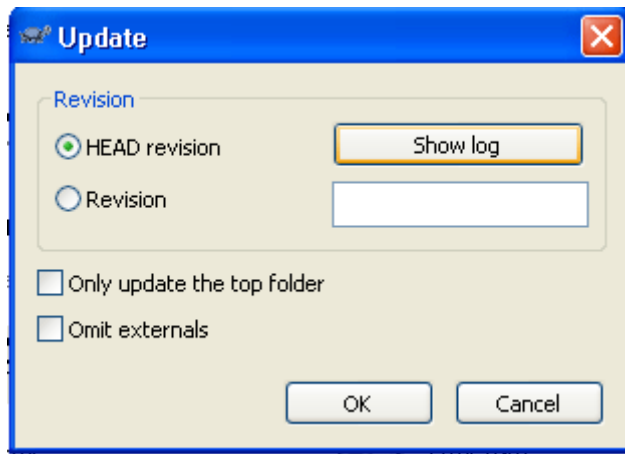
Bước 4: sau khi đưa lên thì phải ... commit các thông tin đó quá trình commit thì tương tự như các bước trên (cập nhật file có thông tin thay đổi).



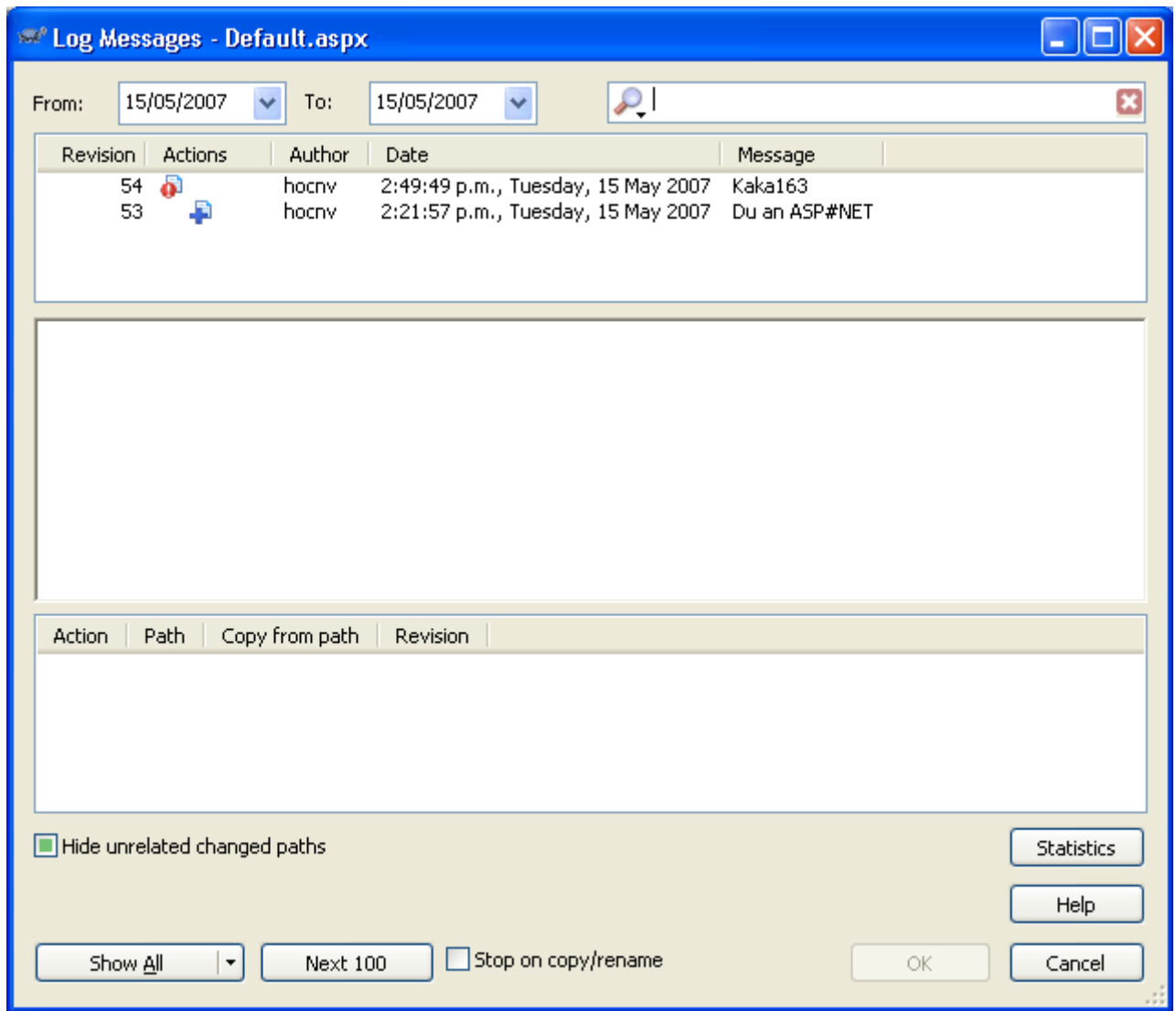
Lấy lại phiên bản trước đó về client (thư mục thực hiện dự án)
Chọn file cần muốn lấy lại các version (phiên bản) trước đây



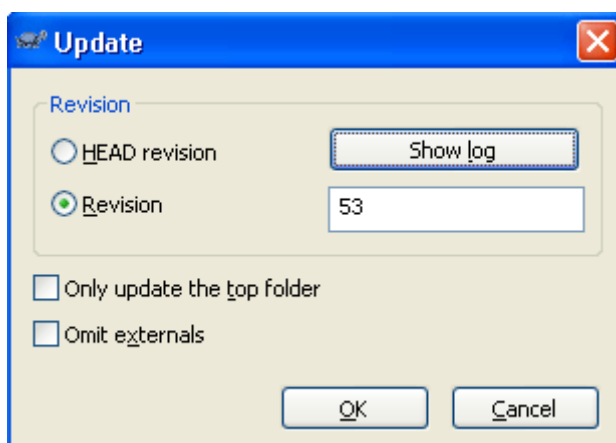
Dialog yêu cầu thông tin version nào



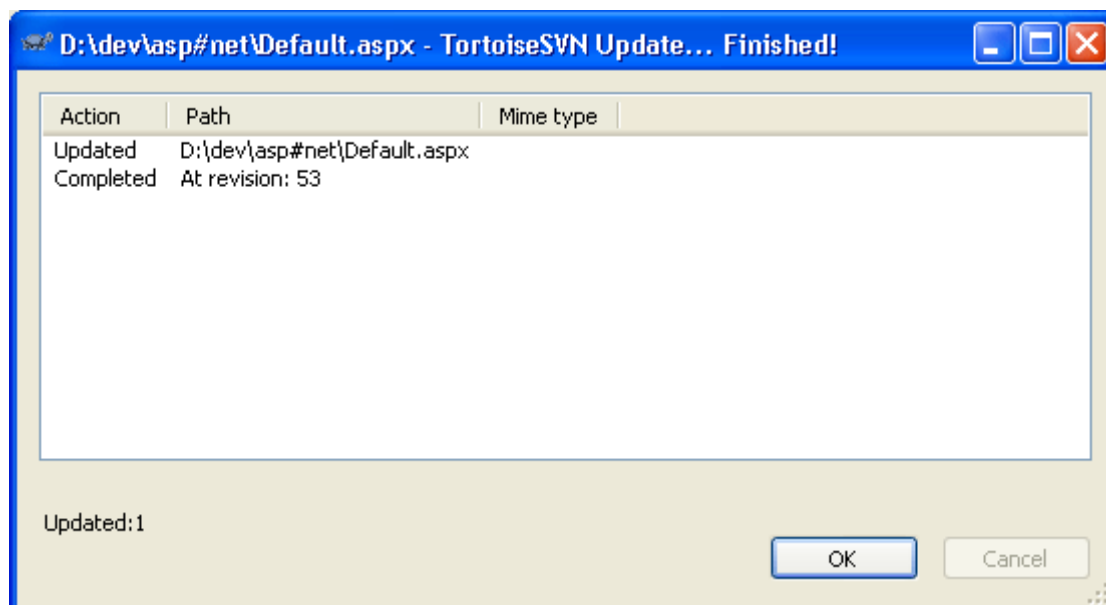
Không biết thì click vào "Show log"



Chọn được ...một version ưng ý (con số 53)



Thế là xong ... file trên thư mục local đã được lấy lại các thông tin của version 53 trước đó



BÀI 21. THỰC HÀNH QUẢN LÝ LỖI (1)

21.1. Mục tiêu :

- Trình bày được quy trình quản lý lỗi và các trạng thái lỗi
- Sử dụng được Bugzilla để quản lý được lỗi cho dự án

21.2. Nội dung :

- Tạo user và login được vào trang chủ Bugzilla
- Tạo dự án và tạo các chức năng cho dự án
- Quản lý được lỗi cho dự án bằng các công cụ của Bugzilla

21.4. Chuẩn đầu ra :

- Sử dụng và quản lý lỗi Bug cho dự án bằng công cụ Bugzilla

Tham khảo:

1.1 Quy trình quản lý lỗi phần mềm của tool tự động Bugzilla

1.1.1 Chức năng của Bugzilla

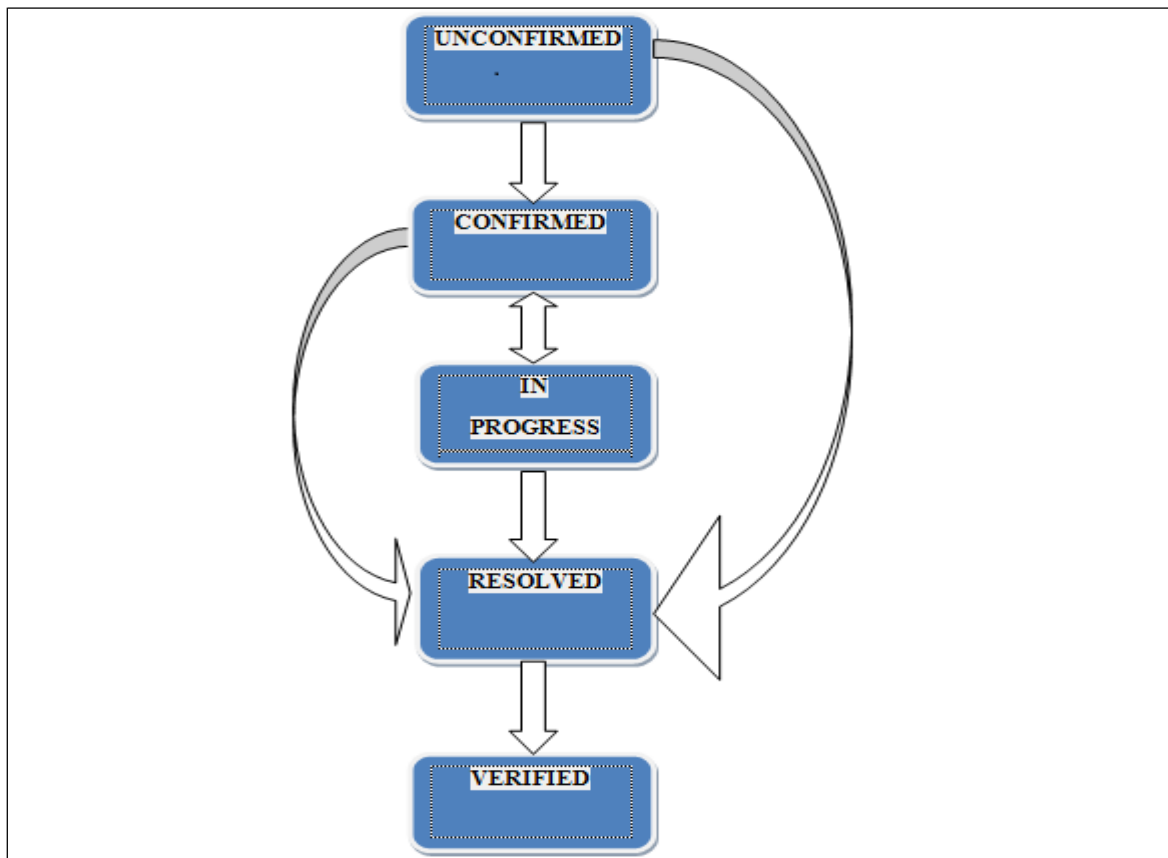
Bugzilla là một mã nguồn mở cung cấp cho chúng ta những chức năng sau:

- Là hệ thống miễn phí quản lý quy trình sửa lỗi phần mềm.
- Cho phép quản lý quá trình hoạt động cũng như tiến độ test lỗi từng dự án..
- Cho phép nhiều user làm việc cùng lúc, dễ tìm kiếm và phân bổ công việc cho từng thành viên. Cập nhật thông tin bug cho thành viên tham gia dự án thông qua chức năng gửi thư điện tử.

1.1.2 Bug và vòng đời của bug

Bug: là thuật ngữ của những người làm công việc kiểm tra phần mềm gọi các lỗi của phần mềm là bug.

Vòng đời của bug: Là thời gian của 1 bug tồn tại từ lúc phát sinh cho tới lúc bug được sửa xong.



Hình 2- 1: Quy trình quản lý lỗi của Bugzilla

Bước 1: Lỗi đã được cập vào cơ sở dữ liệu nhưng chưa được xác định là có phải là lỗi hay không và được gắn với trạng thái UNCONFIRMED.

Bước 2: Xác nhận lỗi

Lỗi cần được xác nhận là có phải lỗi hay không. Nếu là lỗi thì chuyển xuống bước 3, còn không phải lỗi thì gắn trạng thái lỗi resolved là invalid.

Bước 3: Khi lỗi đã được xác nhận sẽ được phân công để sửa lỗi và được gắn với trạng thái in-progress. Nếu lỗi sửa được sẽ chuyển xuống bước 4 và ngược lại quay về bước 2 để xác nhận lại.

Bước 4: Lỗi đã được sửa và đang đợi tester kiểm tra lại.

Bước 5: Lỗi đã được QA xác nhận, tiến hành đóng lỗi.

1.1.3 Các trạng thái của bug

UNCONFIRMED

Unconfirmed (chưa được xác nhận) - Trạng thái này là người test lỗi (một user bình thường) post những phát hiện (được cho là bug) lên hệ thống. Khi ở trạng thái này, bug còn

chưa được xác định nó có phải bug hay không, cần phải có người kiểm tra lại xem đó có phải là bug hay không.

Ở trạng thái này bug có thể chuyển sang các trạng thái như: CONFIRMED, RESOLVED.

CONFIRMED

Trạng thái CONFIRMED là bug mới vừa được post lên hệ thống quản lý bug. Sau khi post bug thành công thì hệ thống Bugzilla sẽ gửi mail tới những thành viên liên quan (người được phân công fix bug này).

Từ trạng thái CONFIRMED, có thể chuyển sang trạng thái INPROGRESS hoặc RESOLVED.

IN - PROGRESS

Trạng thái IN - PROGRESS là bug được phân công cho DEV nào đó fix và đang trong tiến trình fix lỗi, lúc này bug vẫn chưa được fix.

Từ trạng thái này, bug có thể được chuyển sang trạng thái CONFIRMED (để xác nhận lại lỗi và chuyển cho người khác fix bug) hoặc RESOLVED (đã fix xong bug).

RESOLVED

Resolved (giải quyết) - Trạng thái này là bug đã được sửa xong, kết quả có thể là FIXED, INVALID, WONTFIX, DUPLICATE, WORKSFORME.

Ở trạng thái này, bug có thể chuyển sang trạng thái VERIFIED

- Các kết quả của RESOLVED bao gồm:
 - + FIXED: Bug đã fix xong.
 - + INVALID: Vấn đề này không phải là bug.
 - + WONTFIX: Vì lý do nào đó, bug này sẽ không fix (có thể do không có thời gian hoặc bug không quan trọng – cải tiến hoặc không sửa được).
 - + DUPLICATE: Post bug bị trùng với một bug nào đó đã post trước đây. Nếu chọn trạng thái này thì phải nhập thêm bug id của bug bị trùng.
 - + WORKSFORME: bug không thể khắc phục, nếu có thêm thông tin thì có thể mở lại bug.

Chỉ có tester/QC mới có quyền thay đổi trạng thái từ RESOLVED sang các trạng thái khác sau khi đã test lại.

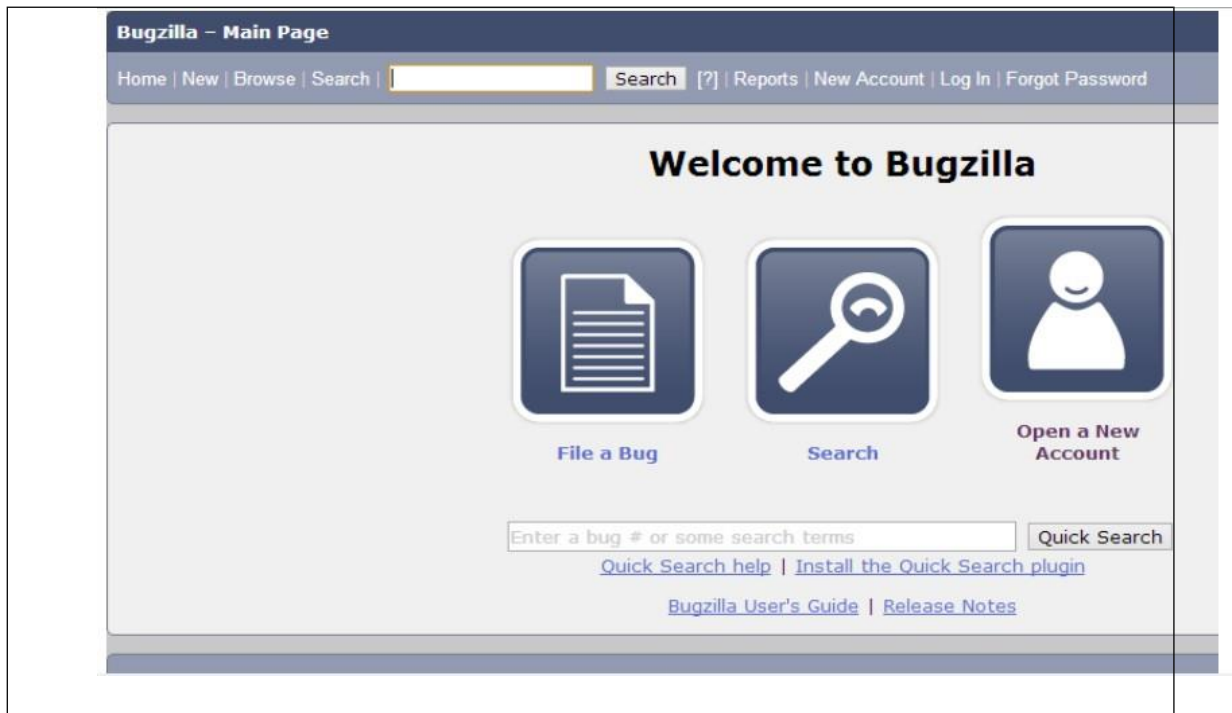
VERIFIED

Trạng thái này là do TESTER/QC chuyển từ trạng thái RESOLVED sang. Trạng thái này là QC xác nhận bug này đã được fix xong. Đây là trạng thái cuối cùng của lỗi.

1.1.4 Cách sử dụng và quản lý phần mềm Bugzilla

a) Tạo user mới và login

Vào trang chủ của Bugzilla:

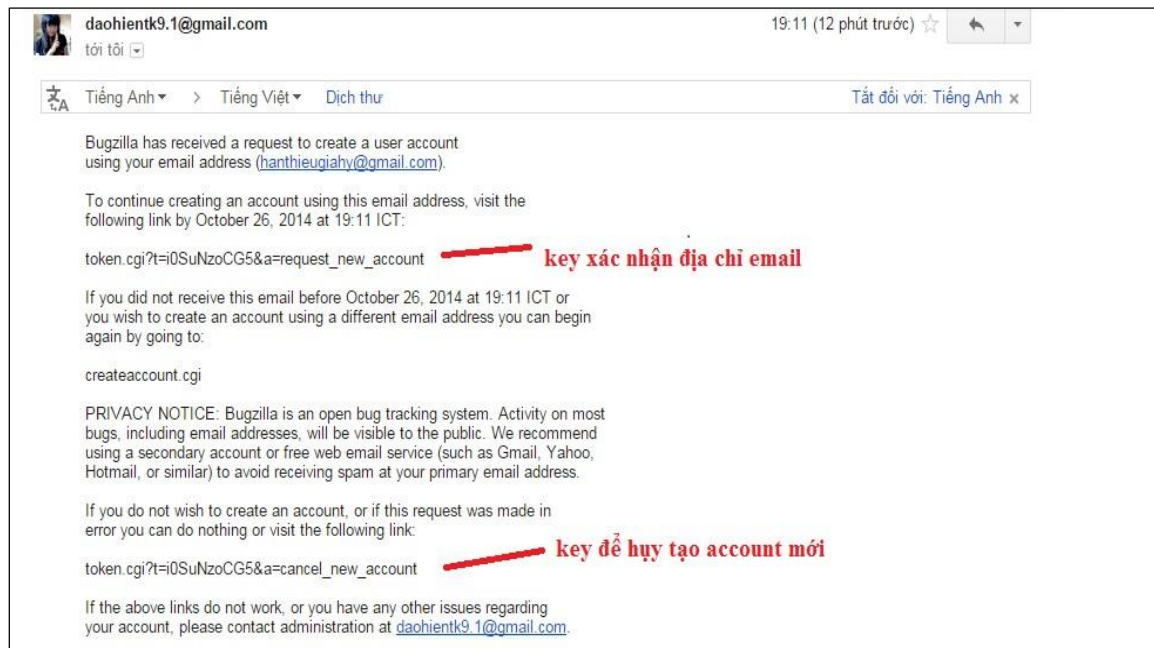


Hình 2- 2: Giao diện trang chủ của Bugzilla

Sau khi click vào “New account” sẽ xuất hiện màn hình

- + Nhập địa chỉ email vào “email address”.
- + Sau đó click nút “Send” (Bạn sẽ nhận được một email thông báo về việc tạo user bugzilla mới.)
- Theo thông tin trên màn hình thì link được gửi tới email chỉ có giá trị trong vòng 3 ngày

Vào địa chỉ email mà ta vừa send:



Hình 2- 3: Email mà được send đến

- Bạn có 2 lựa chọn: dùng key để xác nhận việc tạo account mới hoặc hủy lệnh tạo account.

Sau khi dùng key để xác nhận việc tạo account mới ta được:

The screenshot shows the Bugzilla account creation page with the following details:

- Page Title: Bugzilla – Create a new user account for 'kogapmay@gmail.com'
- Navigation: Home | New | Browse | Search | [Search] | Reports | daohientk9.1@gmail | [Password] | Log in | Forgot Password
- Text: "To create your account, you must enter a password in the form below. Your email address and Real Name (if provided) will be shown with changes you make."
- Form Fields:
 - Email Address: kogapmay@gmail.com (Annotated with "nhập tên")
 - (OPTIONAL) Real Name: kogapmay
 - Type your password: [Redacted] (minimum 6 characters)
 - Confirm your password: [Redacted]
- Buttons: "Create" (Annotated with "tạo account") and "Cancel Account" (Annotated with "hủy tạo account").
- Text: "This account will not be created if this form is not completed by October 22, 2014 at 13:17 ICT."
- Text: "If you do not wish to create an account with this email click the cancel account button below and your details will be forgotten."
- Footer: Home | New | Browse | Search | [Search] | Reports | daohientk9.1@gmail | [Password] | Log in | Forgot Password

Hình 2- 4: Giao diện tạo một account mới

Sau khi nhập tên, pass và click vào nút “Create” thì hiển thị thông báo xác nhận:



Hình 2- 5: Giao diện thông báo việc tạo user thành công

(đã tạo user mới thành công, từ bây giờ bạn có thể login vào Bugzilla)

Click vào link “Log in” trên header hoặc dưới footer để đăng nhập vào Bugzilla. Sau khi click vào Log in thì hiển thị màn hình như bên dưới, nhập email vào ô “Login” và pass đã tạo vào “Password”, sau đó click nút “Log in”.



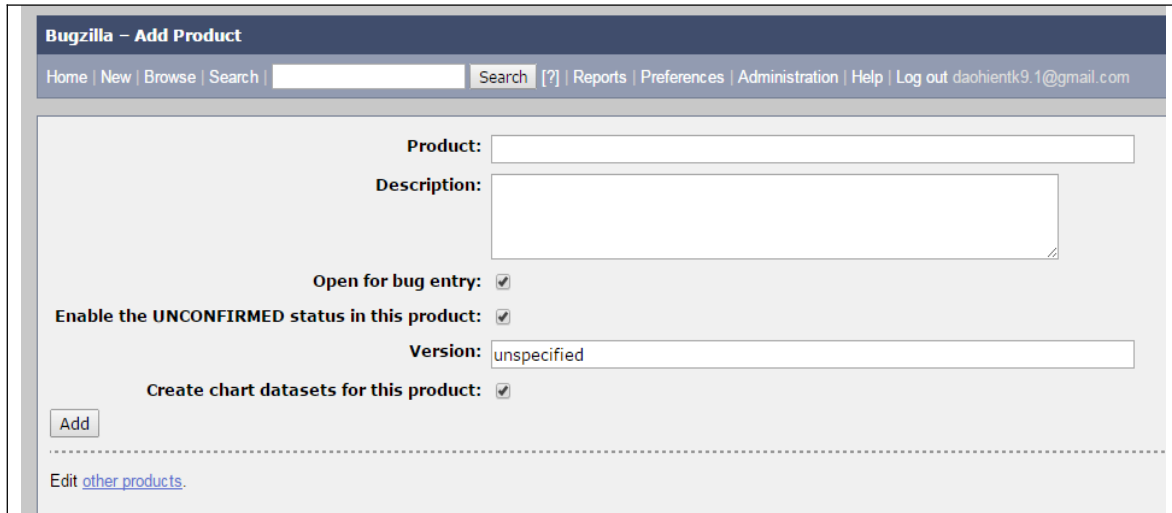
Hình 2- 6: Giao diện trang chủ Bugzilla khi đăng nhập bằng user mới tạo

Trường hợp bạn đã có account nhưng quên pass thì nhập email vào ô trước nút “Submit Request” sau đó click vào nút “Submit Request”

- Khi đó Bugzilla sẽ gửi tới email của bạn một đường link để tạo pass mới (giống như bước thứ 2 trong quá trình tạo account mới).

b) Tạo dự án

- + Bước 1: Đăng nhập vào Bugzilla với quyền admin
- + Bước 2: Administration → Products → add a product sẽ hiện ra màn hình sau:



The screenshot shows the 'Bugzilla - Add Product' web form. At the top, there is a navigation bar with links: Home | New | Browse | Search | [input field] | Search | [?] | Reports | Preferences | Administration | Help | Log out daohientk9.1@gmail.com. The main form area contains the following fields and options:

- Product:** A text input field.
- Description:** A large text area.
- Open for bug entry:** A checkbox that is checked.
- Enable the UNCONFIRMED status in this product:** A checkbox that is checked.
- Version:** A text input field containing the value 'unspecified'.
- Create chart datasets for this product:** A checkbox that is checked.

At the bottom left of the form is an 'Add' button. Below the form area, there is a link that says 'Edit [other products](#)'.

Hình 2- 7: Giao diện trang tạo dự án trên Bugzilla

Tiến hành điền đầy đủ thông tin:

- Product: Website bán mỹ phẩm Naris
- Description: Sản phẩm đồ án 3

Rồi click vào add

BÀI 22. THỰC HÀNH QUẢN LÝ LỖI (2)

22.1. Mục tiêu :

- Trình bày được quy trình quản lý lỗi và các trạng thái lỗi
- Sử dụng được Bugzilla để quản lý được lỗi cho dự án

22.2. Nội dung :

- Tạo user và login được vào trang chủ Bugzilla
- Tạo dự án và tạo các chức năng cho dự án
- Quản lý được lỗi cho dự án bằng các công cụ của Bugzilla

22.3. Chuẩn đầu ra :

- Sử dụng và quản lý lỗi Bug cho dự án bằng công cụ Bugzilla

Tham khảo:

1. Tạo các chức năng của dự án

Administration → (components) Products → Website bán mỹ phẩm Thiên Hương
→ sẽ hiện ra màn hình sau, tiến hành nhập các thông tin cần thiết:

The screenshot shows the Bugzilla web interface for adding a new component to an existing product. The page title is 'Bugzilla - Add component to the Website bán mỹ phẩm Thiên Hương product'. The breadcrumb trail is 'Home | New | Browse | Search | [Search] | Reports | Preferences | Administration | Help | Log out daohientk9.1@gmail.com'. The form contains the following fields and labels:

- Component:** A text input field containing 'Đăng nhập'. To its right is a red label 'tên chức năng'.
- Component Description:** A text area containing 'Màn hình đăng nhập vào hệ thống'. To its right is a red label 'mô tả cho chức năng đó'.
- Default Assignee:** A text input field containing 'kogapmay@gmail.com'. To its right is a red label 'thành viên chịu trách nhiệm sửa lỗi của chức năng này, giá trị mặc định sẽ được hiển thị khi new bug'.
- Default CC List:** A text input field containing 'daohientk9.1@gmail.com, kogapmay@gmail.com, |'. Below the field is the instruction 'Enter user names for the CC list as a comma-separated list.' To the right is a red label 'thành viên liên quan có thể theo dõi lỗi này'.

At the bottom of the form is an 'Add' button. Below the form, there is a link: 'Edit other components of product 'Website bán mỹ phẩm Thiên Hương', or edit product 'Website bán mỹ phẩm Thiên Hương'.' The footer of the page includes the same breadcrumb trail and the text 'My Bugs'.

Hình 2- 8: Giao diện trang tạo chức năng cho dự án trên Bugzilla

2. Post bug vào dự án

Test case cho chức năng đăng nhập vào hệ thống

ID	Test Case Description	Test Case Procedure(tiểu chỉ test)	Expected Output(kết quả mong đợi)
[Module Name]			
Function Name			
TC01	1-Nhập [Tài khoản] = 10 ký tự. 2-Nhập [Mật khẩu]=15 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] và [Mật khẩu] hợp lệ	Đăng nhập thành công
TC02	1-Nhập [Tài khoản] = 10 ký tự. 2-Nhập [Mật khẩu]=6 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] hợp lệ và [Mật khẩu] <8 ký tự	Đăng nhập không thành công
TC03	1-Nhập [Tài khoản] = 10 ký tự. 2-Nhập [Mật khẩu]=30 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] hợp lệ và [Mật khẩu] >25 ký tự	Đăng nhập không thành công
TC04	1-Nhập [Tài khoản] = 3 ký tự. 2-Nhập [Mật khẩu]=15 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] <6 ký tự và [Mật khẩu] hợp lệ	Đăng nhập không thành công
TC05	1-Nhập [Tài khoản] = 25 ký tự. 2-Nhập [Mật khẩu]=15 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] >20 ký tự và [Mật khẩu] hợp lệ	Đăng nhập không thành công
TC06	1-Nhập [Tài khoản] = 3 ký tự. 2-Nhập [Mật khẩu]=6 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] <6 ký tự và [Mật khẩu] <8 ký tự	Đăng nhập không thành công
TC07	1-Nhập [Tài khoản] = 3 ký tự. 2-Nhập [Mật khẩu]=30 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] <6 ký tự và [Mật khẩu] > 25 ký tự	Đăng nhập không thành công
TC08	1-Nhập [Tài khoản] = 25 ký tự. 2-Nhập [Mật khẩu]=6 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] >20 ký tự và [Mật khẩu] <8 ký tự	Đăng nhập không thành công
TC09	1-Nhập [Tài khoản] = 25 ký tự. 2-Nhập [Mật khẩu]=30 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] >20 ký tự và [Mật khẩu] >25 ký tự	Đăng nhập không thành công
TC010	1-Nhập [Tài khoản] = 6 ký tự. 2-Nhập [Mật khẩu]=8 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =6 ký tự và [Mật khẩu] =8 ký tự	Đăng nhập thành công
TC011	1-Nhập [Tài khoản] = 6 ký tự. 2-Nhập [Mật khẩu]=25 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =6 ký tự và [Mật khẩu] = 25 ký tự	Đăng nhập thành công
TC012	1-Nhập [Tài khoản] = 20 ký tự. 2-Nhập [Mật khẩu]=8 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =20 ký tự và [Mật khẩu] = 8 ký tự	Đăng nhập thành công
TC013	1-Nhập [Tài khoản] = 20 ký tự. 2-Nhập [Mật khẩu]=25 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =20 ký tự và [Mật khẩu] = 25 ký tự	Đăng nhập thành công
TC014	1-Nhập [Tài khoản] = 5 ký tự. 2-Nhập [Mật khẩu]=7 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =5 ký tự và [Mật khẩu] = 7 ký tự	Đăng nhập không thành công
TC015	1-Nhập [Tài khoản] = 21 ký tự. 2-Nhập [Mật khẩu]=26 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =21 ký tự và [Mật khẩu] = 26 ký tự	Đăng nhập không thành công
TC016	1-Nhập [Tài khoản] = 5 ký tự. 2-Nhập [Mật khẩu]=26 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =5 ký tự và [Mật khẩu] = 26 ký tự	Đăng nhập không thành công
TC017	1-Nhập [Tài khoản] = 7 ký tự. 2-Nhập [Mật khẩu]=21 ký tự. 3-Click vào [Đăng nhập]	[Tài khoản] =7 ký tự và [Mật khẩu] = 21 ký tự	Đăng nhập không thành công

Hình 2- 9: Danh sách các test case cho chức năng đăng nhập

Theo bản test case ta có: pass với 5 test case là: TC01, TC010, TC011, TC012, TC013; còn các test case còn lại đều Fail. Vì vậy ta tiến hành post lỗi với 12 test case Fail này

- Các bước thực hiện
- + Test case có mã TC02

Sau khi nhận được danh sách test case, tester tiến hành post lỗi vào dự án:

Đăng nhập vào Bugzilla → New: sẽ hiển thị ra màn hình post bug mới, tiến hành nhập thông tin cần thiết:

Bugzilla – Enter Bug: Website bán mỹ phẩm Thiên Hương

Home | New | Browse | Search | Search | [?] | Reports | Preferences | Administration | Help | Log out daohientk9.1@gmail.com

Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug

[Hide Advanced Fields](#) (* = Required Field)

* **Product:** Website bán mỹ phẩm Thiên Hương **Reporter:** daohientk9.1@gmail.com

* **Component:** Đăng nhập **Component Description:** Màn hình đăng nhập vào hệ thống

* **Version:** unspecified **Severity:** critical

Hardware: PC **OS:** Windows

Priority: High

Status: CONFIRMED

Assignee: kogapmay@gmail.com

CC:

Default CC: daohientk9.1@gmail.com, kogapmay@gmail.com

Orig. Est.:

Deadline:

URL: http://myphamthienhuong.vn/

* **Summary:** [Tai khoan] hop le va [Mat khau] < 8 ky tu nhung dang nhap thanh cong

Description:

1. Nhap [Tai khoan]=10 ky tu
2. Nhap [Mat khau]=6 ky tu
3. Click vao [Dang nhap]

Attachment: Don't add an attachment

Add an attachment

File: Enter the path to the file on your computer.
Chon tệp test case form dang nhap.xls

Description: Describe the attachment briefly.
testcase cho form Đăng nhập

Content Type: If the attachment is a patch, check the box below.
☐ patch

Otherwise, choose a method for determining the content type.
☐ auto-detect
☒ select from list: plain text (text/plain)
☐ enter manually:

Depends on:

Blocks:

Submit Bug Remember values as bookmarkable template

Home | New | Browse | Search | Search | [?] | Reports | Preferences | Administration | Help | Log out dao

My Bugs

Hình 2- 10: Giao diện post lỗi mã test case là TC02

Sau đó click vào Submit Bug thì lỗi này sẽ được lưu lại và thêm vào danh sách lỗi của dự án đồng thời lỗi này cũng được gửi tới mail của người được phân công sửa lỗi này

3. Cập nhật bug

User sẽ tiến hành đăng nhập vào Bugzilla và tiến hành sửa Status từ CONFIRMED sang IN_PROGRESS (đang trong tiến trình sửa lỗi)

The screenshot displays a Bugzilla bug report interface. At the top, there is an 'Attachments' section with a table listing an attachment named 'testcase cho form Đăng nhập' (176.50 KB, text/plain) with a 'Details' link. Below this is a link to 'Add an attachment' (proposed patch, testcase, etc.). The 'Additional Comments' section features a large text area for comments. Below the text area, the 'Status' is set to 'IN_PROGRESS' with a dropdown arrow, and there is a 'Save Changes' button and a 'Mark as Duplicate' link. At the bottom, a comment by 'DaoHien' on '2014-11-03 17:32:09 ICT' is shown, containing a list of steps: '1.Nhập [Username] = 10 ký tự', '2.Nhập [Password] =15 ký tự', and '3.Kích vào [OK]'. To the right of the comment are links for 'Description', '[reply]', and '[-]'. Further right are links to 'Collapse All Comments' and 'Expand All Comments'.

Hình 2- 11: Giao diện cập nhập lỗi được post lên

Sau đó click Save changes để lưu lại

Việc sửa đổi này sẽ được gửi đến mail của các thành viên liên quan

4. Kết thúc bug

Khi bug được sửa xong, tester sẽ tiến hành kiểm tra lại và cập nhật lại trạng thái cho bug sang RESOLVED và chọn FIXED INVALID DUPLICATE WORKSFORME

Các test case khác chúng ta tiến hành làm tương tự.

➤ Danh sách bug đã được post lên Bugzilla:

Bugzilla - Bug List							
Home New Browse Search <input type="text"/> Search [?] Reports Preferences Administration Help Log out daohientk9.1@gmail.com							
Fri Nov 28 2014 12:41:40 ICT							
<i>Bugzilla would like to put a random quip here, but no one has entered any.</i>							
Resolution: --- Assignee: daohientk9.1@gmail.com Reporter: daohientk9.1@gmail.com							
12 bugs found.							
ID ▲	Sev	Pri ▼	OS	Assignee ▲	Status ▲	Resolution ▼	Summary
21	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] hop le va [Mat khau] < 8 ky tu nhung dang nhap thanh cong
22	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] hop le va [Mat khau] > 25 ky tu nhung dang nhap thanh cong
23	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] < 6 ky tu va [Mat khau] hop le nhung dang nhap thanh cong
24	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] > 20 ky tu va [Mat khau] hop le nhung dang nhap thanh cong
25	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] < 6 ky tu va [Mat khau] < 8 ky tu nhung dang nhap thanh cong
26	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] < 6 ky tu va [Mat khau] > 25 ky tu nhung dang nhap thanh cong
27	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] > 20 ky tu va [Mat khau] < 8 ky tu nhung dang nhap thanh cong
28	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] > 20 ky tu va [Mat khau] > 25 ky tu nhung dang nhap thanh cong
29	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] = 5 ky tu va [Mat khau] = 7 ky tu nhung dang nhap thanh cong
30	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] = 21 ky tu va [Mat khau] = 26 ky tu nhung dang nhap thanh cong
31	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] = 5 ky tu va [Mat khau] = 26 ky tu nhung dang nhap thanh cong
32	cri	Hig	Wind	kogapmay@gmail.com	CONF	---	[Tai khoan] = 7 ky tu va [Mat khau] = 21 ky tu nhung dang nhap thanh cong
12 bugs found.							
Long Format							
XML CSV Feed iCalendar Change Columns Change Several Bugs at Once Edit Search Remember search as <input type="text"/>							
Time Summary							
File a new bug in the "Website bán mỹ phẩm Thiên Hương" product							

Hình 2- 12: Danh sách các lỗi đã được post lên Bugzilla

BÀI 23. THỰC HÀNH ÁP DỤNG CHUẨN

23.1. Mục tiêu :

- Trình bày được chuẩn ISO, ISO 9126 , Phương pháp đánh giá ISO 9126
- Trình bày được mô hình CMM, CMMI

23.2. Nội dung :

- Xây dựng được các tiêu chí đánh giá theo chuẩn ISO 9126
- Xây dựng được quy trình cho các giai đoạn trong quy trình dự án phần mềm

23.3. Chuẩn đầu ra :

- Áp dụng được các bảng tiêu chí đánh giá theo chuẩn và mô hình chất lượng nhằm đảm bảo chất lượng cho dự án phần mềm
- Gợi ý:

Tiêu chí	Thang điểm
I. Tính chức năng	
1. Trình duyệt đáp ứng đầy đủ các chức năng cơ bản như: xem phim, nghe nhạc, chơi game, mở nhiều tab cùng lúc	
2. Sự khác nhau giữa các kết quả thực và các kết quả mong đợi không chênh lệch ít	
3. Mức độ hài lòng của người dùng với kết quả mà trình duyệt đạt được	
4. Tỷ lệ người sử dụng trao đổi dữ liệu thất bại với phần mềm khác	
5. Trình duyệt có khả năng đồng bộ hóa dữ liệu	
6. Bạn cảm thấy an toàn về tính riêng tư và bảo mật khi sử dụng trình duyệt	
7. Trình duyệt có cảnh báo khi vào web chứa virus độc hại	
8. Trình duyệt có chế độ duyệt web riêng tư	
II. Tính tin cậy	
1. Mức độ trình duyệt gây ra hỏng hóc cho toàn bộ môi trường sản xuất	
2. Mức độ lỗi khi sử dụng trình duyệt	
3. Tần suất hệ thống có thể khởi động lại để cung cấp dịch vụ cho người sử dụng trong một khoảng thời gian yêu cầu.	
4. Trình duyệt có khả năng tự khôi phục lại của trình duyệt sau khi có sự kiện bất thường xảy ra hoặc khi có yêu cầu.	
5. Khả năng khôi phục lại dữ liệu nguyên vẹn như ban đầu	
III. Tính khả dụng	
1. Các chức năng con của trình duyệt rõ ràng, người sử dụng dễ dàng hiểu được các chức năng con	
2. Người sử dụng mất ít thời gian để học sử dụng các chức năng	
3. Trình duyệt có hỗ trợ, tài liệu hướng dẫn sử dụng, phản hồi từ phía người sử dụng	

4. Giao diện với người sử dụng thân thiện, dễ đọc, sắp xếp hợp lý, đồng nhất.	
5. Các thành phần của giao diện có thể được tùy biến theo sở thích người sử dụng	
IV. Tính hiệu quả	
1. Thời gian cần để tìm kiếm một website nhanh	
2. Thời gian hệ thống đáp ứng người sử dụng nhanh chóng.	
3. Chiếm ít tài nguyên hệ thống khi sử dụng.	
V. Khả năng bảo trì	
1. Cho phép sửa đổi, nâng cấp, sẵn sàng đáp ứng các hỗ trợ về mặt kỹ thuật	
2. Bản cập nhật được update thường xuyên và miễn phí	
3. Người sử dụng có thể sử dụng trình duyệt sau khi được bảo trì không	
4. Người sử dụng có thể dễ dàng nhận biết phiên bản sửa lại.	
VI. Tính khả chuyển	
1. Hệ thống phần mềm có đủ khả năng tự tương thích với môi trường hoạt động	
2. Trình duyệt cho phép tích hợp với một số tiện ích khác	
3. Người sử dụng có thể dễ dàng cài đặt phần mềm trong môi trường hoạt động mới	
4. Người sử dụng có thể dễ dàng cài đặt phần mềm ở các hệ điều hành khác nhau	
5. Người sử dụng có thể dễ dàng tiếp tục sử dụng cùng dữ liệu sau khi thay thế phần mềm này thay cho phần mềm trước	

Bảng các tiêu chí đánh giá

Thiết kế kế hoạch đánh giá sản phẩm

Tiêu chí	Phép đánh giá ngoài được áp dụng
I. Tính chức năng	
1. Trình duyệt đáp ứng đầy đủ các chức năng cơ bản như: xem phim, nghe nhạc, chơi game, mở nhiều tab cùng lúc	Tính đầy đủ chức năng
2. Sự khác nhau giữa các kết quả thực và các kết quả mong đợi không chênh lệch ít	Độ chính xác mong đợi
3. Mức độ hài lòng của người dùng với kết quả mà trình duyệt đạt được	
4. Tỷ lệ người sử dụng trao đổi dữ liệu thất bại với phần mềm khác	Khả năng trao đổi dữ liệu
5. Trình duyệt có khả năng đồng bộ hóa dữ liệu (Có đồng bộ: thỏa mãn; Không đồng bộ: không thỏa	Không áp dụng

mãn)	
6. Bạn cảm thấy an toàn về tính riêng tư và bảo mật khi sử dụng trình duyệt (an toàn: thỏa mãn; không an toàn: không thỏa mãn)	Không áp dụng
7. Trình duyệt có cảnh báo khi vào web chứa virus độc hại (Có cảnh báo: thỏa mãn; Không có cảnh báo: không thỏa mãn)	Không áp dụng
8. Trình duyệt có chế độ duyệt web riêng tư (Có: thỏa mãn; Không: Không thỏa mãn)	Không áp dụng
II. Tính tin cậy	
1. Mức độ trình duyệt gây ra hỏng hóc cho toàn bộ môi trường sản xuất	Tránh hỏng hóc
2. Mức độ lỗi, treo máy khi sử dụng trình duyệt (ít : thỏa mãn; thường xuyên: không thỏa mãn)	Không áp dụng
3. Tần suất hệ thống có thể khởi động lại để cung cấp dịch vụ cho người sử dụng trong một khoảng thời gian yêu cầu.	Khả năng khởi động lại
4. Trình duyệt có khả năng tự khôi phục lại của trình duyệt sau khi có sự kiện bất thường xảy ra hoặc khi có yêu cầu.	Khả năng khôi phục lại
5. Khả năng khôi phục lại dữ liệu nguyên vẹn như ban đầu	Tính hiệu quả của việc khôi phục lại
III. Tính khả dụng	
1. Các chức năng con của trình duyệt rõ ràng, người sử dụng dễ dàng hiểu được các chức năng con (các chức năng rõ ràng: thỏa mãn; các chức năng không rõ ràng: không thỏa mãn)	Không áp dụng
2. Người sử dụng mất ít thời gian để học sử dụng các chức năng	Khả năng dễ dàng học chức năng
3. Trình duyệt có hỗ trợ, tài liệu hướng dẫn sử dụng, phản hồi từ phía người sử dụng (Có tài liệu hướng dẫn: thỏa mãn; không có tài liệu: không thỏa mãn)	Không áp dụng
4. Giao diện với người sử dụng thân thiện, dễ đọc, sắp xếp hợp lý, đồng nhất. (thỏa mãn; không thỏa mãn)	Không áp dụng
5. Các thành phần của giao diện có thể được tùy biến theo sở thích người sử dụng (có thể tùy biến: thỏa mãn; không thể tùy biến: không thỏa mãn)	Không áp dụng
IV. Tính hiệu quả	

1. Thời gian cần để tìm kiếm một website nhanh	Thời gian đáp ứng
2. Thời gian hệ thống đáp ứng người sử dụng nhanh chóng.	Thời gian hoàn thành
3. Chiếm ít tài nguyên hệ thống khi sử dụng.	Không áp dụng
V. Khả năng bảo trì	
1. Cho phép sửa đổi, nâng cấp, sẵn sàng đáp ứng các hỗ trợ về mặt kỹ thuật (cho phép nâng cấp: thỏa mãn; không cho phép nâng cấp: không thỏa mãn)	Không áp dụng
2. Bản cập nhật được update thường xuyên và miễn phí (cập nhật miễn phí: thỏa mãn; mất phí cập nhật: không thỏa mãn)	Không áp dụng
3. Người sử dụng có thể sử dụng trình duyệt sau khi được bảo trì không	Tỉ lệ thay đổi thành công
4. Người sử dụng có thể dễ dàng nhận biết phiên bản sửa lại.	Không áp dụng
VI. Tính khả chuyển	
1. Hệ thống phần mềm có đủ khả năng tự tương thích với môi trường hoạt động	Khả năng tương thích môi trường phần cứng
2. Trình duyệt cho phép tích hợp với một số tiện ích khác	Không áp dụng
3. Người sử dụng có thể dễ dàng cài đặt phần mềm trong môi trường hoạt động mới	Không áp dụng
4. Người sử dụng có thể dễ dàng cài đặt phần mềm ở các hệ điều hành khác nhau	Không áp dụng
5. Người sử dụng có thể dễ dàng tiếp tục sử dụng cùng dữ liệu sau khi thay thế phần mềm này thay cho phần mềm trước	Không áp dụng

Bảng kế hoạch đánh giá

BÀI 24 THỰC HÀNH ÁP DỤNG ĐỘ ĐO CHẤT LƯỢNG

23.1. Mục tiêu :

- Trình bày được các cách đo chất lượng phần mềm
- Xây dựng được bảng các độ đo

23.2. Nội dung :

- Xây dựng được bảng các độ đo
- Lựa chọn một ứng dụng và áp dụng bảng các độ đo đã xây dựng để

23.3. Chuẩn đầu ra :

- Áp dụng được các bảng tiêu chí đánh giá theo chuẩn và mô hình chất lượng nhằm đảm bảo chất lượng cho dự án phần mềm
- Tham khảo:

ID	N	Objective	U	D e
Scope				
2.2	Requirement Completeness	Measures percentage of requirement completed against the committed and predict the amount of remain work	%	$= ((\text{Requirement Size} * \text{Requirement Completed Rate}) / \text{Total committed (Requirement Size)}) * 100\%$
2.3	Requirement Stability	Measures stability of requirements, frequency of requirements changed by customer Provides the visibility into whether requirements changes are responsible for cost overruns, schedule delays, and decreased product quality.	%	$= (\text{Size of Requirement Changed} / \text{Total Requirement Size}) * 100\%$ Note: - Requirement changed: Added, deleted, changed requirement from the 1st baseline of Requirement
7.2	Size Deviation	Provides information on how well the size estimation is performed. It is basis for estimation, scheduling, effort & work allocation	%	$= ((\text{Actual size} - 1\text{st estimated size}) / 1\text{st estimated size}) * 100\%$
	Test Case Density	To provide a measurement on the coverage of project test case on the software	Test case/KLOC	= No of Test Cases/KLOC for Productivity
Schedule				
3.1	Timelines	Measures the ability to satisfy customer in timeliness.	%	$= (\text{No of deliverable delivered on time} / \text{Total \# of deliverables delivered}) * 100\%$
3.4	Schedule Deviation	Provides information on project performance with respect to its schedule commitment	%	$= ((\text{Actual duration} - \text{Last committed duration}) / \text{Last committed duration}) * 100\%$

	Schedule Status	Monitor project schedule	%	= (Stage schedule deviation/ Fsoft norm of project schedule deviation)*100% Note: - Blue: =<80% - Yellow: <100% & >81% - Red: =>100%
	Time ETC	To forecast the Finished date of the project.	date	= Current day + (Actual duration/Actual requirement completeness)*Committed requirement completeness – Actual duration
3.2	Response Time	Measure the readiness of Support service	hour	Not applicable
Cost				
4.1	Effort Effectiveness	Provides tracking of actual effort against budgeted effort and predict future effort	%	= (Actual Effort Usage/ Last Committed Effort Usage)*100% Note: - Actual Effort Usage is calculated based on Timesheet registered for the project
4.2	Effort Deviation	1. Indicates effectiveness of project planning and management 2. Predicts future effort	%	= ((Actual Effort Usage- Last Committed Effort Usage)/ Last Committed Effort Usage)*100%
	Effort Efficiency	Measure efficiency of project effort usage	%	= (Billable effort/Calendar effort)*100%
	Effort Utilization	To evaluate resource utilization at project level To evaluate Timesheet adequacy of projects	%	= (Effort usage/Calendar effort)*100% Note: - Actual Effort Usage: effort spent on projects as recorded in TimeSheet system
4.3	Effort Distribution	1. Indicates proportion of Effort to activity/product/subprocess/process in Project life cycle 2. Helps in scheduling and work distribution	%	= (Effort Usage for the activity or product or subprocess or process/Total Effort Usage or Product Effort Usage)*100%
4.2	Effort Status	Indicates performance of effort usage in comparison with requirement completeness	%	= (Effort effectiveness of Engineering activities/Requirement completeness)*100% Note: - Blue: =<80%

BÀI 25. KIỂM TRA THỰC HÀNH