

ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

Bài giảng cho sinh viên ngành Công nghệ thông tin

Phan Thị Hoài Phương

Hà Nội - 2010

Giới thiệu

Trước những thách thức trong quá trình phát triển phần mềm, việc đảm bảo chất lượng phần mềm (Software Quality Assurance-SQA) là hết sức quan trọng, đòi hỏi phải nghiên cứu một cách nghiêm túc để thực thi hiệu quả. Tài liệu này cung cấp những kiến thức cơ bản về chất lượng phần mềm, đảm bảo chất lượng trong một dự án phát triển phần mềm. Quy trình xây dựng hệ thống đảm bảo chất lượng phần mềm cũng được trình bày trong nội dung bài giảng. Qua đó, sinh viên hiểu được cách thức xây dựng một hệ thống đảm bảo chất lượng phần mềm và vai trò của những thành viên trong hệ thống. Một số chuẩn đảm bảo chất lượng cũng được giới thiệu trong những chương cuối. Thông qua nội dung bài giảng sinh viên cũng sẽ nắm được kỹ năng rà soát và kiểm thử phần mềm.

Tài liệu được soạn phần lớn dựa trên cuốn sách *Software Quality Assurance From Theory to Implementation* của Daniel Galin và một số tài liệu về kỹ nghệ phần mềm, nhằm hỗ trợ cho sinh viên gặp khó khăn khi đọc các tài liệu nguyên gốc tiếng Anh.

Nội dung bài giảng được xây dựng trong bảy chương.

Chương 1. Khái niệm về chất lượng phần mềm và các yếu tố chất lượng phần mềm

Những khái niệm mở đầu của tài liệu được giới thiệu trong chương 1. Bắt đầu với khái niệm phần mềm, chất lượng phần mềm và đảm bảo chất lượng phần mềm, phần tiếp theo phân tích các yếu tố chất lượng phần mềm.

Chương 2. Các thành phần chất lượng phần mềm tiền dự án

Chương này trình bày những nội dung liên quan đến những thành phần đảm bảo chất lượng phần mềm tiền dự án bao gồm việc rà soát hợp đồng, kế hoạch phát triển dự án phần mềm và kế hoạch chất lượng phần mềm.

Chương 3. Các thành phần SQA trong vòng đời dự án

Chương 3 đề cập đến các thành phần đảm bảo chất lượng phần mềm trong vòng đời dự án phần mềm. Những nội dung được trình bày trong chương này bao gồm : phân tích một số mô hình phát triển phần mềm phổ biến, các phương pháp rà soát, bảo trì phần mềm và các công cụ CASE. Riêng kiểm thử phần mềm là bước quan trọng sẽ được trình bày riêng ở chương 4.

Chương 4. Kiểm thử phần mềm

Chương 4 đề cập đến kiểm thử phần mềm. Những nội dung được trình bày trong chương này bao gồm : khái niệm cơ bản, các mức kiểm thử, các kỹ thuật kiểm thử, và quá trình kiểm thử.

Chương 5. Phân loại các phần mềm phục vụ kiểm

Chương 5 đề cập đến các loại thành phần được dùng trong kiểm thử phần mềm. Những nội dung được trình bày trong chương này bao gồm : các phần mềm phục vụ kiểm thử và thư viện JUnit được sử dụng rộng rãi trong kiểm thử đơn vị cho ngôn ngữ lập trình Java.

Chương 6. Các thành phần cơ bản của chất lượng phần mềm

Các thành phần cơ bản của chất lượng phần mềm bao gồm các thủ tục (procedure), chỉ dẫn (instruction), khuôn mẫu (templates), checklists (danh mục kiểm tra). Đó chính là nội dung được trình bày trong phần đầu của chương 6. Phần tiếp theo sẽ trình bày các hoạt động đảm bảo chất lượng phần mềm khác như : đào tạo và cấp chứng chỉ, ngăn ngừa và sửa lỗi, quản lý cấu hình và kiểm soát tài liệu.

Chương 7. Các thành phần quản lý chất lượng phần mềm

Ngoài yếu tố kỹ thuật, trong các dự án phát triển phần mềm hiện đại, yếu tố quản lý đóng vai trò hết sức quan trọng. Chương 7 trình bày các vấn đề liên quan đến quản lý chất lượng phần mềm như : điều khiển tiến độ dự án, độ tin cậy chất lượng phần mềm, chi phí chất lượng phần mềm.

Chương 8. Các chuẩn, chứng chỉ và hoạt động đánh giá

Chương này đề cập tới các chuẩn quản lý chất lượng như ISO 9001 và ISO 9000-3, CMM và CMMI và các chuẩn tiêu trình dự án như IEEE/EIA Std 12207, IEEE Std 1012, IEEE Std 1028.

Chương 9. Tổ chức để đảm bảo chất lượng

Trong những tổ chức lớn, quản lý nguồn nhân lực là một yếu tố quyết định sự thành công. Chương 9 đề cập đến các tác nhân tham gia vào hệ thống đảm bảo chất lượng phần mềm, vai trò, trách nhiệm của mỗi tác nhân được phân tích cụ thể trong từng đề mục của chương.

Phụ lục

Trình bày về các lỗi thường gặp khi viết chương trình.

MỤC LỤC

Giới thiệu	2
Chương 1. Khái niệm về chất lượng phần mềm và các yếu tố chất lượng phần mềm	8
1.1. Đặc điểm của phần mềm và môi trường phát triển phần mềm	8
1.2. Khái niệm phần mềm	11
1.3. Lỗi phần mềm và phân loại nguyên nhân gây ra lỗi phần mềm	12
1.3.1. Lỗi phần mềm	12
1.3.2. Nguyên nhân gây ra lỗi phần mềm	12
1.4. Định nghĩa chất lượng phần mềm và đảm bảo chất lượng phần mềm	15
1.5. Những mục tiêu đảm bảo chất lượng phần mềm	15
1.6. Phân loại yêu cầu phần mềm ứng với các yếu tố chất lượng phần mềm	16
Chương 2. Các thành phần chất lượng phần mềm tiền dự án	20
2.1. Rà soát hợp đồng	20
2.1.1. Tiến trình rà soát hợp đồng và các bước thực hiện	20
2.1.2. Các mục tiêu rà soát hợp đồng	21
2.1.3. Thực thi rà soát hợp đồng	24
2.1.4. Những khó khăn của thực hiện xem lại hợp đồng cho các đề xuất chính	25
2.1.5. Khuyến cáo cho việc thực hiện duyệt lại những hợp đồng chính	26
2.1.6. Các đối tượng rà soát hợp đồng	27
2.1.7. Rà soát hợp đồng cho các dự án nội bộ	27
2.2. Các kế hoạch phát triển và kế hoạch chất lượng	30
2.2.1. Những mục tiêu của kế hoạch phát triển và kế hoạch chất lượng	31
2.2.2. Các thành phần của kế hoạch phát triển	31
2.2.3. Các thành phần của kế hoạch chất lượng	35
2.2.4. Các kế hoạch phát triển và kế hoạch chất lượng cho các dự án nhỏ và các dự án nội bộ	38
Chương 3. Các thành phần SQA trong vòng đời dự án	41
3.1. Tích hợp các hoạt động chất lượng trong vòng đời dự án	41
3.1.1. Phương pháp phát triển phần mềm truyền thống và các phương pháp khác	41
3.1.2. Các yếu tố ảnh hưởng hoạt động đảm bảo chất lượng phần mềm	51
3.1.3. Xác minh, thẩm định và đánh giá chất lượng	52
3.2. Rà soát	53
3.2.1. Mục tiêu rà soát	53
3.2.2. Những rà soát thiết kế hình thức	54
3.2.3. Các rà soát ngang hàng (peer review)	56
3.2.4. Các ý kiến của chuyên gia	57
3.3. Đảm bảo chất lượng của các thành phần bảo trì phần mềm	59
3.3.1. Giới thiệu	59
3.3.2. Cơ sở cho chất lượng bảo trì cao	61
3.3.3. Các thành phần chất lượng phần mềm tiền bảo trì	64
3.3.4. Các công cụ đảm bảo chất lượng bảo trì phần mềm	68
3.4. Các CASE tool và ảnh hưởng của nó lên chất lượng phần mềm	77
3.4.1. Khái niệm CASE tool	77
3.4.2. Đóng góp của CASE tool cho chất lượng sản phẩm phần mềm	79
3.4.3. Đóng góp của CASE tool cho chất lượng bảo trì phần mềm	81
3.4.4. Đóng góp của CASE tool cho quản lý dự án	82

3.5. Đảm bảo chất lượng phần mềm của các yếu tố bên ngoài cùng tham gia	82
3.5.1. Những thành phần bên ngoài đóng góp vào dự án phần mềm.....	82
3.5.2. Rủi ro và lợi ích của giới thiệu người tham dự ngoài	83
3.5.3. Những mục tiêu đảm bảo chất lượng về sự đóng góp người tham gia bên ngoài	84
3.5.4. Các công cụ đảm bảo chất lượng những đóng góp của các thành viên đóng góp bên ngoài.....	85
Chương 4. Kiểm thử phần mềm.....	86
4.1. Một số khái niệm cơ bản.....	86
4.1.1. Ví dụ về lỗi phần mềm.....	86
4.1.2. Đặc tả và lỗi phần mềm:	87
4.1.3. Kiểm thử và tiến trình kiểm thử.....	88
4.1.4. Các mức kiểm thử.....	90
4.1.5. Một số thuật ngữ	91
4.2. Các cấp độ kiểm thử	94
4.2.1. Kiểm thử đơn vị - Unit Testing.....	95
4.2.2. Kiểm thử tích hợp - Integration Testing	95
4.2.3. Kiểm thử hệ thống - System Testing	100
4.2.4. Kiểm thử chấp nhận - Acceptance Testing	101
4.3. Các kỹ thuật kiểm thử.....	102
4.3.1. Kiểm thử hộp đen - Black-box testing.....	102
4.3.2. Kiểm thử hộp trắng - White box testing (WB)	108
4.3.3. Kiểm thử gia tăng - Incremental Testing	116
4.3.4. Thread Testing	116
4.3.5. Bảng tóm tắt Testing Levels, Techniques.....	116
4.4. Quá trình kiểm thử.....	116
4.4.1. Xác định tiêu chuẩn chất lượng phần mềm phù hợp	116
4.4.2. Lập kế hoạch kiểm thử.....	119
4.4.3. Thiết kế kiểm thử (Test design)	121
4.4.4. Tiến trình test.....	124
4.4.5. Thiết kế trường hợp kiểm thử (Test Case Design).....	125
Chương 5. Phân loại các phần mềm phục vụ kiểm thử	127
5.1. Phần mềm phục vụ kiểm thử	127
5.1.1. Phần mềm hỗ trợ viết tài liệu.....	127
5.1.2. Phần mềm quản lý lỗi	127
5.1.3. Công cụ kiểm thử tự động.....	129
5.2. Unit test và thư viện JUnit	133
5.2.1. Tổng quan về Unit Testing	133
5.2.2. Tổng quan thư viện Junit	135
Chương 6. Các thành phần cơ bản của chất lượng phần mềm.....	139
6.1. Thủ tục, chỉ dẫn và các thiết bị hỗ trợ chất lượng.....	139
6.1.1. Các thủ tục và chỉ dẫn.....	139
6.1.2. Chuẩn bị, thực thi và cập nhật các thủ tục và chỉ dẫn.....	142
6.1.3. Khuôn mẫu (templates).....	143
6.1.4. Danh mục kiểm tra (Checklists)	146
6.2. Đào tạo đội ngũ và cấp chứng chỉ.....	148

6.2.1.	Mục tiêu của đào tạo và cấp chứng chỉ.....	149
6.2.2.	Tiến trình đào tạo và cấp chứng chỉ.....	149
6.2.3.	Xác định yêu cầu kiến thức chuyên môn và sự cần thiết của đào tạo và cập nhật.....	150
6.2.4.	Xác định những nhu cầu đào tạo và cập nhật (updating).....	150
6.2.5.	Lên kế hoạch đào tạo và chương trình cập nhật.....	151
6.2.6.	Định nghĩa các vị trí yêu cầu cấp chứng chỉ.....	151
6.2.7.	Lên kế hoạch các tiến trình cấp chứng chỉ.....	152
6.2.8.	Phân phối các chương trình đào tạo và cấp chứng chỉ.....	153
6.2.9.	Những công việc tiếp theo của việc đào tạo và cấp chứng chỉ.....	153
6.3.	Các hành động sửa lỗi và phòng ngừa.....	154
6.3.1.	Định nghĩa hoạt động sửa lỗi và phòng ngừa.....	154
6.3.2.	Tiến trình hành động sửa lỗi và phòng ngừa.....	154
6.3.3.	Thu thập, phân tích thông tin.....	155
6.3.4.	Phát triển các giải pháp và thực thi.....	155
6.3.5.	Tổ chức các hành động phòng ngừa và sửa lỗi.....	156
6.4.	Quản lý cấu hình.....	157
6.4.1.	Các thành phần cấu hình phần mềm.....	157
6.4.2.	Quản lý cấu hình phần mềm.....	157
6.4.3.	Kiểm soát sự thay đổi phần mềm.....	157
6.4.4.	Kiểm soát quản lý cấu hình phần mềm.....	158
6.4.5.	Các công cụ máy tính quản lý cấu hình phần mềm.....	159
6.5.	Kiểm soát tài liệu.....	159
6.5.1.	Các tài liệu kiểm soát và ghi chép chất lượng.....	159
6.5.2.	Danh sách các tài liệu được kiểm soát.....	160
6.5.3.	Chuẩn bị, phê chuẩn, lưu trữ và truy hồi tài liệu kiểm soát.....	160
Chương 7.	Các thành phần quản lý chất lượng phần mềm.....	163
7.1.	Điều khiển tiến độ dự án.....	163
7.1.1.	Các thành phần điều khiển tiến độ dự án.....	163
7.1.2.	Điều khiển tiến độ của các dự án nội bộ và các thành phần bên ngoài.....	163
7.1.3.	Thực thi kiểm soát tiến độ dự án.....	163
7.1.4.	Các công cụ kiểm soát tiến độ phần mềm.....	164
7.2.	Độ đo chất lượng phần mềm.....	165
7.2.1.	Các mục tiêu đo lường phần mềm và phân loại các độ đo.....	166
7.2.2.	Các độ đo tiến trình.....	167
7.2.3.	Các độ đo sản phẩm.....	170
7.2.4.	Thực hiện đo chất lượng phần mềm.....	172
7.2.5.	Những giới hạn của các độ đo phần mềm.....	173
7.3.	Giá thành của chất lượng phần mềm.....	174
7.3.1.	Các mục tiêu tính giá thành các độ đo chất lượng phần mềm.....	174
7.3.2.	Mô hình truyền thống tính giá chất lượng phần mềm.....	174
7.3.3.	Mô hình mở rộng tính giá chất lượng phần mềm.....	175
7.3.4.	Các vấn đề trong áp dụng tính giá các độ đo chất lượng phần mềm.....	177
Chương 8.	Các chuẩn, chứng chỉ và hoạt động đánh giá.....	178
8.1.	Các chuẩn quản lý chất lượng.....	178
8.1.1.	Phạm vi của các chuẩn quản lý chất lượng.....	178

8.1.2. ISO 9001 và ISO 9000-3.....	178
8.1.3. Các mô hình tăng trưởng khả năng – phương pháp đánh giá CMM và CMMI 180	
8.2. Các chuẩn tiến trình dự án SQA	181
8.2.1. IEEE/EIA Std 12207- các tiến trình vòng đời phần mềm.....	182
8.2.2. IEEE Std 1012 – xác minh và thẩm định.....	184
8.2.3. IEEE Std 1028 – rà soát.....	185
Chương 9. Tổ chức để đảm bảo chất lượng.....	187
9.1. Giới thiệu	187
9.1.1. Cơ cấu tổ chức phát triển phần mềm	187
9.1.2. Khung tổ chức phát triển phần mềm.....	187
9.2. Quản lý và vai trò của quản lý trong đảm bảo chất lượng phần mềm	188
9.2.1. Các hoạt động đảm bảo chất lượng của quản lý mức cao nhất.....	188
9.2.2. Những trách nhiệm quản lý phòng ban.....	190
9.2.3. Những trách nhiệm quản lý dự án.....	191
9.3. Đơn vị SQA và các tác nhân khác trong hệ thống SQA.....	192
9.3.1. Đơn vị SQA	192
9.3.2. Những ủy viên SQA và nhiệm vụ.....	193
9.3.3. Hội đồng SQA và nhiệm vụ.....	194
9.3.4. Nhiệm vụ và phương thức hoạt động của liên đoàn SQA	194
Tài liệu tham khảo	197
Phụ lục	198

Chương 1. Khái niệm về chất lượng phần mềm và các yếu tố 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 của 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) Mối 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 thiết kế là cần thiết 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 các 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

1.3.1. 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. Chính 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
9. Lỗi tài liệu

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 trong 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 thiết kế logic

Lỗi phần mềm có thể đi vào hệ thống khi các chuyên gia thiết kế hệ thống-các kiến trúc sư hệ thống, kỹ sư phần mềm, các nhà phân tích, vv - Xây dựng phần mềm yêu cầu. Các lỗi điển hình bao gồm:

- + Định nghĩa các yêu cầu phần mềm bằng các thuật toán sai lầm.
- + Quy trình định nghĩa có chứa trình tự lỗi.
- + Sai sót trong các định nghĩa biên
- + Thiếu sót trong các trạng thái hệ thống phần mềm được yêu cầu
- + Thiếu sót trong định nghĩa các hoạt động trái pháp luật trong hệ thống 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.

1.4. 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 hoặc phần hoặc một tiến trình đạt được yêu cầu đã đặc tả*
- *Mức độ mà một hệ thống hoặc 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.

1.5. 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.

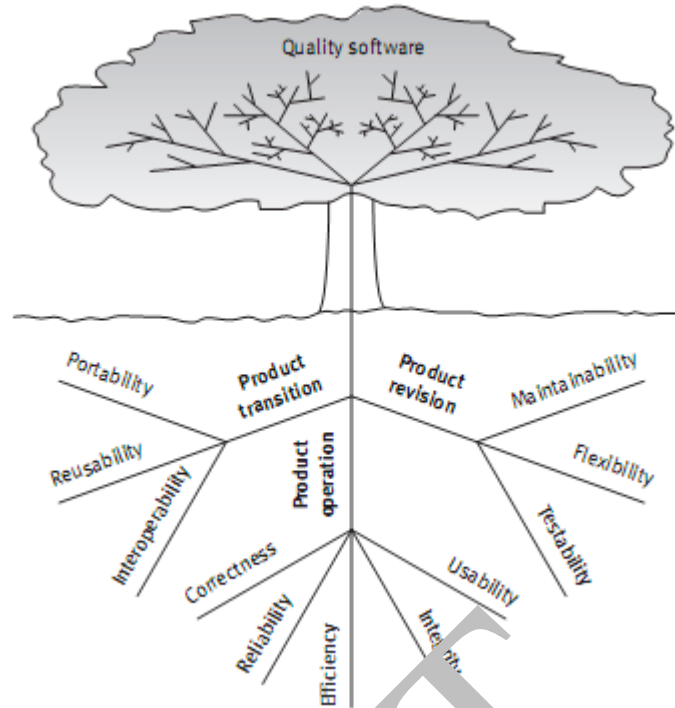
- Bảo trì phần mềm (hướng sản phẩm)

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.

1.6. Phân loại yêu cầu phần mềm tương 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 nhận 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 do 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 Cây mô hình yếu tố chất lượng trong 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 :
 - Nhiệm vụ đầu ra (ví dụ : bản in hóa đơn bán hàng, hay đèn báo động đo khi nhiệt độ tăng lên trên 250 độ F)
 - 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.
 - Tính đầy đủ của thông tin đầu ra; chúng có thể bị ảnh hưởng bất lợi bởi dữ liệu không đầy đủ.
 - Up-to-dateness của thông tin (xác định bằng thời gian giữa sự kiện và việc xem xét hệ thống phần mềm.
 - 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)

- 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 không 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 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 tố 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 chuyển 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.

Chương 2. Các thành phần chất lượng phần mềm tiền dự án

2.1. 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).

2.1.1. 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 bộ phận 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 kí (“rà soát bản dự thảo hợp đồng”). Giai đoạn này rà soát lại bản dự thảo hợp đồng dựa trên đề