

Trường Đại Học Công Nghiệp TP.HCM

Khoa Công Nghệ Thông tin



Báo cáo môn học

**ĐẢM BẢO CHẤT LƯỢNG VÀ
KIỂM THỬ PHẦN MỀM**

Tên đề tài:

Các thành phần quản lý của chất lượng phần mềm

Nhóm: 8

1. Nguyễn Thị Nga
2. Nguyễn Đa Nghiêm
3. Nguyễn Đăng Nguyên
4. Phan Nguyễn Khôi Nguyên
5. Phạm Thị Thùy Nhi

MỤC LỤC

Chương 1. MỞ ĐẦU.....	1
Chương 2. NỘI DUNG.....	2
1. Kiểm soát tiến độ dự án.....	2
1.1. Mục tiêu của kiểm soát tiến độ dự án	2
1.2. Các thành phần khi kiểm soát tiến độ dự án	2
1.2.1. Kiểm soát các hoạt động quản lý rủi ro:.....	2
1.2.2. Kiểm soát tiến độ dự án:.....	2
1.2.3. Kiểm soát tài nguyên hệ thống:	3
1.2.4. Kiểm soát ngân sách dự án:	3
1.3. Kiểm soát tiến độ của các dự án nội bộ và người tham gia bên ngoài:	4
1.4. Triển khai các quy trình kiểm soát tiến độ dự án.....	5
1.5. Công cụ vi tính cho kiểm soát tiến độ phần mềm.....	6
2. Các độ đo chất lượng phần mềm	7
2.1. Mục tiêu chung khi thu thập các độ đo	7
2.1.1. Hỗ trợ kiểm soát quản lý, lập kế hoạch và thực hiện các can thiệp quản lý thích hợp:	7
2.1.2. Xác định các tình huống cần hoặc có thể cải thiện quy trình phát triển hoặc bảo trì: 8	
2.2. Phân loại các độ đo chất lượng phần mềm	8
2.2.1. Độ đo quy trình (Process Metrics).....	8
2.2.2. Độ đo sản phẩm (Product Metrics).....	9
2.3. Giải thích công thức độ đo	9
2.3.1. Error Density Metrics	9
2.3.2. Development Productivity Metrics.....	10

3.	Chi phí chất lượng phần mềm	10
3.1.	Mục tiêu của chi phí chất lượng phần mềm.....	11
3.2.	Mô hình cổ điển về chi phí chất lượng phần mềm.....	11
3.2.1.	Chi phí kiểm soát	11
3.2.2.	Chi phí do sai sót trong kiểm soát	12
3.3.	Mô hình mở rộng về chi phí chất lượng phần mềm.....	13
3.3.1.	Chi phí chuẩn bị và kiểm soát quản lý.....	15
3.3.2.	Chi phí thất bại trong quản lý	15
Chương 3. KẾT LUẬN		17
Chương 4. TÀI LIỆU THAM KHẢO.....		18

BẢNG PHÂN CÔNG

MSSV	Họ và tên	Công việc thực hiện	Thời gian thực hiện	Đánh giá
21130791	Nguyễn Thị Nga	1. Mở đầu + Kết luận 2. Phân chia công việc 3. Duyệt nội dung tiểu luận 4. Định dạng file cho tiểu luận	Từ 02/09/2024 Đến 15/10/2024	10/10
21057901	Nguyễn Đa Nghiêm	1. Tìm hiểu một công cụ trong quản lý tiến độ dự án 2. Soạn nội dung: Kiểm soát tiến độ dự án <ul style="list-style-type: none">- Các thành phần khi kiểm soát tiến độ dự án (Bổ sung)- Kiểm soát tiến độ của các dự án nội bộ và người tham gia bên ngoài- Triển khai các quy trình kiểm soát tiến độ dự án- Công cụ vi tính cho kiểm soát tiến độ phần mềm	Từ 02/09/2024 Đến 15/10/2024	10/10
21082231	Nguyễn Đăng Nguyên	1. Soạn nội dung: Kiểm soát tiến độ dự án <ul style="list-style-type: none">- Mục tiêu của kiểm soát dự án- Các thành phần khi kiểm soát tiến độ dự án	Từ 02/09/2024 Đến 15/10/2024	10/10
21068021	Phan Nguyễn Khôi Nguyên	1. Soạn nội dung: Các độ đo chất lượng phần mềm <ul style="list-style-type: none">- Mục tiêu chung khi thu thập các độ đo- Phân loại các độ đo chất lượng phần mềm- Giải thích công thức độ đo	Từ 02/09/2024 Đến 15/10/2024	10/10
21054781	Phạm Thị Thùy Nhi	1. Soạn nội dung: Chi phí chất lượng phần mềm <ul style="list-style-type: none">- Mục tiêu của chi phí chất lượng phần mềm- Mô hình cổ điển về chi phí chất lượng phần mềm- Mô hình mở rộng về chi phí chất lượng phần mềm	Từ 02/09/2024 Đến 15/10/2024	10/10

CHƯƠNG 1. MỞ ĐẦU

Trong sự phát triển mạnh mẽ của ngành công nghiệp phần mềm, quản lý tiến độ dự án (Project Progress Control) trở thành một yếu tố quan trọng giúp đảm bảo tính khả thi và thành công của bất kỳ dự án nào. Việc kiểm soát tiến độ không chỉ đơn thuần là việc theo dõi các công việc đã hoàn thành và những hạng mục cần thực hiện, mà còn giúp nhà quản lý dự đoán trước các rủi ro, điều chỉnh tài nguyên hợp lý, và đảm bảo rằng dự án được thực hiện đúng theo kế hoạch đã định. Đây là một khía cạnh thiết yếu trong quản lý dự án, đặc biệt trong các dự án phần mềm phức tạp, nơi có rất nhiều yếu tố bất định và sự thay đổi thường xuyên về yêu cầu của khách hàng.

Kiểm soát tiến độ giúp dự án đạt được các mục tiêu quan trọng về thời gian, ngân sách và chất lượng, đồng thời nâng cao tính minh bạch và hiệu quả trong quá trình làm việc nhóm. Đặc biệt, trong những dự án yêu cầu mức độ phối hợp cao giữa các nhóm chức năng khác nhau, kiểm soát tiến độ đóng vai trò như một phương tiện giúp kết nối các bộ phận, tối ưu hóa sự tương tác và phối hợp công việc.

Song song với việc kiểm soát tiến độ, chất lượng phần mềm luôn là tiêu chí hàng đầu được các tổ chức đặt ra khi phát triển sản phẩm. Thu thập và phân tích các độ đo chất lượng phần mềm không chỉ giúp nhà phát triển đánh giá hiệu quả của quá trình phát triển mà còn giúp phát hiện sớm các lỗi và sự thiếu sót để khắc phục kịp thời. Các độ đo này đóng vai trò như những chỉ số đánh giá sự thành công của sản phẩm, đảm bảo rằng phần mềm cuối cùng không chỉ đáp ứng các yêu cầu chức năng mà còn đảm bảo tính ổn định và hiệu quả trong vận hành.

Bên cạnh đó, chi phí chất lượng phần mềm cũng là một yếu tố quan trọng cần được quản lý cẩn thận. Các tổ chức không chỉ cần đảm bảo rằng chất lượng sản phẩm đạt yêu cầu, mà còn phải tối ưu hóa chi phí liên quan đến phát triển, kiểm thử và bảo trì phần mềm. Việc quản lý chi phí chất lượng giúp giảm thiểu rủi ro phát sinh chi phí ngoài dự kiến, đồng thời nâng cao khả năng kiểm soát tài chính của dự án. Các mô hình quản lý chi phí chất lượng, như mô hình cổ điển và mô hình mở rộng, cung cấp nền tảng vững chắc cho việc phân tích và điều chỉnh chi phí hợp lý trong quá trình phát triển phần mềm.

Trong tiểu luận này, chúng ta sẽ cùng tìm hiểu sâu hơn về các khía cạnh quan trọng trong kiểm soát tiến độ dự án, thu thập độ đo chất lượng phần mềm và quản lý chi phí chất lượng. Cụ thể, tiểu luận sẽ giải thích mục tiêu và các thành phần của kiểm soát tiến độ, trình bày các phương pháp thu thập độ đo chất lượng, và giới thiệu các mô hình chi phí chất lượng phần mềm. Qua đó, người đọc sẽ có cái nhìn toàn diện về các yếu tố quyết định sự thành công của một dự án phần mềm, từ việc duy trì tiến độ cho đến đảm bảo chất lượng và tối ưu hóa chi phí.

CHƯƠNG 2. NỘI DUNG

1. Kiểm soát tiến độ dự án

1.1. Mục tiêu của kiểm soát tiến độ dự án

Mục tiêu chính của kiểm soát tiến độ dự án là đảm bảo rằng dự án được thực hiện đúng theo kế hoạch đã định, bao gồm việc hoàn thành các nhiệm vụ đúng thời gian và đạt được các mốc quan trọng đã đề ra. Cụ thể, các mục tiêu của kiểm soát tiến độ dự án bao gồm:

- Phát hiện sớm các sự kiện bất thường: Giúp nhận các vấn đề tiềm tàng trong quản lý dự án như trễ tiến độ hoặc vượt ngân sách ngay khi chúng xuất hiện
- Khởi động các phản ứng giải quyết vấn đề kịp thời: Từ việc phát hiện sớm, các biện pháp khắc phục và điều chỉnh có thể được thực hiện để giảm thiểu tác động tiêu cực.
- Đề xuất các hành động điều chỉnh lâu dài: Cung cấp thông tin về các thành công và thất bại cực đoan để xác định và thực hiện các biện pháp điều chỉnh cho các dự án trong tương lai.

1.2. Các thành phần khi kiểm soát tiến độ dự án

Kiểm soát tiến độ dự án (CMM sử dụng thuật ngữ "theo dõi dự án phần mềm") có một mục tiêu trước mắt: phát hiện sớm các sự kiện bất thường. Việc phát hiện sớm giúp khởi động kịp thời các biện pháp giải quyết vấn đề. Thông tin tích lũy về kiểm soát tiến độ cũng như những thành công và thất bại nghiêm trọng cũng phục vụ một mục tiêu dài hạn: khởi xướng các hành động khắc phục.

Các thành phần chính của kiểm soát tiến độ dự án bao gồm:

1.2.1. *Kiểm soát các hoạt động quản lý rủi ro:*

Điều này liên quan đến các hạng mục rủi ro phần mềm được xác định trong giai đoạn trước dự án, những hạng mục được liệt kê trong tài liệu xem xét hợp đồng và kế hoạch dự án, cùng với các hạng mục rủi ro khác được xác định trong suốt quá trình thực hiện dự án. Nhóm phát triển phần mềm xử lý các hạng mục rủi ro phần mềm bằng cách áp dụng các hoạt động quản lý rủi ro có hệ thống. Việc kiểm soát tiến độ quản lý rủi ro bắt đầu bằng việc chuẩn bị các đánh giá định kỳ về tình trạng của các hạng mục rủi ro phần mềm và kết quả dự kiến của các hoạt động quản lý rủi ro đã thực hiện. Dựa trên các báo cáo này, các nhà quản lý dự án được kỳ vọng sẽ can thiệp và giúp tìm ra giải pháp trong những trường hợp nghiêm trọng. Một số tiêu chuẩn và nhiều sách báo đã đề cập đến rủi ro trong dự án phần mềm, chẳng hạn như IEEE (2001) và Jones (1994).

1.2.2. *Kiểm soát tiến độ dự án:*

Phần này liên quan đến việc tuân thủ của dự án với các thời gian biểu đã được phê duyệt và ký kết. Theo dõi chủ yếu dựa trên các mốc quan trọng, được thiết lập một phần để phát hiện sự chậm trễ trong việc hoàn thành các hoạt động đã lập kế hoạch. Các mốc quan trọng trong hợp đồng, đặc biệt là các ngày giao sản phẩm phần mềm được quy định hoặc hoàn thành một hoạt động phát triển, thường được nhấn mạnh đặc biệt. Mặc dù có thể dự đoán được một số sự chậm trễ, nhưng quản lý sẽ tập trung các hoạt động kiểm soát vào các chậm trễ quan trọng, những chậm trễ có thể ảnh hưởng đáng kể đến việc hoàn thành cuối cùng của dự án. Nhiều thông tin cần thiết cho việc kiểm soát tiến độ dự án được truyền tải thông qua các báo cáo mốc và các báo cáo định kỳ khác. Để đối phó với thông tin này, quản lý có thể can thiệp bằng cách phân bổ thêm nguồn lực hoặc thậm chí đàm phán lại thời gian biểu với khách hàng.

1.2.3. Kiểm soát tài nguyên hệ thống:

Kiểm soát nguồn lực chủ yếu tập trung vào nguồn nhân lực chuyên môn nhưng cũng có thể liên quan đến các tài sản khác. Đối với các hệ thống phần mềm thời gian thực và phần mềm nhúng, các tài nguyên phát triển và kiểm thử phần mềm thường đòi hỏi kiểm soát nghiêm ngặt nhất. Ở đây, quản lý dựa trên các báo cáo định kỳ về việc sử dụng nguồn lực để so sánh giữa thực tế và kế hoạch. Cần nhấn mạnh rằng mức độ sai lệch thực sự trong việc sử dụng nguồn lực chỉ có thể được đánh giá từ quan điểm tiến độ của dự án. Nói cách khác, một dự án chỉ có vẻ như có sự sai lệch nhỏ trong việc sử dụng nguồn lực (ví dụ 5%) có thể thực sự gặp phải sai lệch nghiêm trọng tích lũy (ví dụ 25%) nếu tiến độ bị chậm trễ nghiêm trọng.

Một khía cạnh khác của kiểm soát nguồn lực là sự phân bổ hoặc thành phần nội bộ. Ví dụ, quản lý có thể nhận thấy không có sự sai lệch nào xảy ra trong tổng số tháng công lao động dành cho các nhà phân tích hệ thống. Tuy nhiên, việc xem xét chi tiết có thể phát hiện rằng thay vì sử dụng 25% tháng công dành cho các nhà phân tích hệ thống cao cấp, thực tế đã sử dụng 50%, điều này có thể làm phá vỡ kế hoạch ngân sách ban đầu.

1.2.4. Kiểm soát ngân sách dự án:

Kiểm soát ngân sách dựa trên sự so sánh giữa chi tiêu thực tế và kế hoạch. Giống như kiểm soát nguồn lực, để có bức tranh chính xác hơn về sự sai lệch ngân sách, cần xem xét các sự chậm trễ liên quan đến việc hoàn thành các hoạt động. Các hạng mục ngân sách chính cần kiểm soát bao gồm:

- Nguồn nhân lực
- Cơ sở phát triển và kiểm thử
- Mua phần mềm COTS
- Mua phần cứng
- Thanh toán cho các nhà thầu phụ.

Kiểm soát ngân sách là ưu tiên cao nhất của quản lý vì tác động trực tiếp của nó đến lợi nhuận của dự án. Tuy nhiên, việc chỉ tập trung vào kiểm soát ngân sách có thể dẫn đến bỏ sót các công cụ kiểm soát tiến độ khác như kiểm soát rủi ro và tiến độ, khiến việc xử lý vấn đề trở nên chậm trễ và tốn kém hơn trong dài hạn.

1.3. Kiểm soát tiến độ của các dự án nội bộ và người tham gia bên ngoài:

Kiểm soát tiến độ dự án được khởi xướng nhằm cung cấp cho ban quản lý một cái nhìn tổng quan về tất cả các hoạt động phát triển phần mềm đang được thực hiện trong tổ chức. Tuy nhiên, trong hầu hết các tổ chức, việc kiểm soát dự án thường chỉ cung cấp một cái nhìn hạn chế về tiến độ của các dự án phát triển phần mềm nội bộ và thậm chí còn ít thông tin hơn về tiến độ của các đối tác bên ngoài. Kiểm soát các dự án nội bộ và các đối tác bên ngoài thường gặp phải những khiếm khuyết, như sẽ được mô tả dưới đây.

Các dự án nội bộ, chẳng hạn như những dự án được thực hiện cho các phòng ban khác hoặc các dự án phát triển các gói phần mềm cho thị trường phần mềm chung, không bao gồm sự tham gia của khách hàng bên ngoài theo định nghĩa. Do đó, các dự án này thường được xếp vào thứ tự ưu tiên thấp hơn trong danh sách của ban quản lý. Việc thiếu sự chú ý đầy đủ này thường đi kèm với việc theo dõi không phù hợp hoặc lỏng lẻo từ phía khách hàng nội bộ. Tình trạng này thường dẫn đến việc nhận diện muộn các sự chậm trễ nghiêm trọng và việc vượt quá ngân sách đáng kể, dẫn đến việc khắc phục hạn chế các vấn đề gặp phải. Giải pháp không thể tránh khỏi cho tình trạng này là áp dụng đầy đủ các biện pháp kiểm soát tiến độ dự án cho các dự án nội bộ.

Các đối tác bên ngoài bao gồm các nhà thầu phụ, nhà cung cấp phần mềm COTS (Commercial Off-the-Shelf) và các module phần mềm tái sử dụng, và trong một số trường hợp là chính khách hàng. Dự án càng lớn và phức tạp thì càng có khả năng các đối tác bên ngoài sẽ được yêu cầu tham gia, và tỷ lệ công việc phân bổ cho họ càng cao. Quản lý tìm đến các đối tác bên ngoài vì nhiều lý do khác nhau, từ lợi ích kinh tế đến kỹ thuật và nhân sự, và xu hướng này ngày càng tăng trong các hợp đồng và thầu phụ dự án. Hơn nữa, các thỏa thuận ký kết với các đối tác trong dự án đã trở nên phức tạp

đến mức việc giao tiếp và phối hợp đã trở thành vấn đề đối với cả nhóm dự án lẫn ban quản lý. Để đối phó, cần có những nỗ lực đáng kể hơn để đạt được mức độ kiểm soát chấp nhận được. Do đó, kiểm soát tiến độ dự án đối với các đối tác bên ngoài phải tập trung chủ yếu vào tiến độ của dự án và các rủi ro đã được xác định trong các hoạt động dự án đã lên kế hoạch.

1.4. Triển khai các quy trình kiểm soát tiến độ dự án

Kiểm soát tiến độ dự án thường dựa trên các quy trình nhằm xác định:

- Phân bổ trách nhiệm cho việc thực hiện các nhiệm vụ kiểm soát quy trình phù hợp với đặc điểm của dự án, bao gồm kích thước dự án:
 - + Cá nhân hoặc đơn vị quản lý chịu trách nhiệm thực hiện các nhiệm vụ kiểm soát tiến độ
 - + Tần suất báo cáo được yêu cầu từ mỗi đơn vị của dự án và từng cấp quản lý
 - + Các tình huống yêu cầu lãnh đạo dự án phải báo cáo ngay lập tức cho ban quản lý
 - + Các tình huống yêu cầu quản lý cấp dưới phải báo cáo ngay lập tức cho quản lý cấp trên.
- Các cuộc kiểm toán của ban quản lý về tiến độ dự án, chủ yếu tập trung vào: (1) mức độ hiệu quả của việc truyền đạt các báo cáo tiến độ bởi lãnh đạo dự án và từ quản lý cấp dưới lên cấp trên, và (2) các hoạt động kiểm soát cụ thể mà ban quản lý cần khởi xướng.
- Trong các tổ chức phát triển phần mềm lớn, kiểm soát tiến độ dự án có thể được thực hiện trên nhiều cấp độ quản lý, chẳng hạn như quản lý phòng ban phần mềm, quản lý bộ phận phần mềm và ban quản lý cấp cao. Mặc dù mỗi cấp quản lý đều cần xác định riêng chế độ kiểm soát tiến độ dự án của mình, phản ánh các thông số được coi là phù hợp để đánh giá tiến độ của dự án từ góc độ của cấp đó, sự phối hợp giữa các cấp độ khác nhau là điều bắt buộc để kiểm soát tiến độ đạt hiệu quả.

- Toàn bộ chuỗi báo cáo sẽ truyền đạt thông tin được thu thập từ cấp quản lý thấp nhất báo cáo tiến độ định kỳ của lãnh đạo dự án – tóm tắt tình trạng các rủi ro của dự án, tiến độ dự án và việc sử dụng nguồn lực, tức là ba thành phần chính đầu tiên của kiểm soát tiến độ. Lãnh đạo dự án sẽ dựa trên thông tin thu thập được từ các trưởng nhóm để lập báo cáo tiến độ của mình.

1.5. Công cụ vi tính cho kiểm soát tiến độ phần mềm

Các công cụ vi tính cho kiểm soát tiến độ dự án phần mềm là điều cần thiết rõ ràng khi kích thước và độ phức tạp của các dự án ngày càng tăng lên, và những lợi ích mà chúng mang lại cũng rất đáng kể. Các công cụ quản lý dự án toàn diện đã có mặt trên thị trường từ nhiều năm nay có thể phục vụ hầu hết các thành phần kiểm soát của các dự án phần mềm một cách hiệu quả và chính xác. Phần lớn các gói phần mềm đa mục đích này sử dụng phân tích PERT/CPM để các báo cáo kết quả có thể xem xét sự tương tác giữa các hoạt động và mức độ quan trọng của từng hoạt động. Các gói phần mềm này thường dễ dàng thích ứng với các trường hợp cụ thể nhờ sự đa dạng lớn về các tùy chọn mà chúng cung cấp.

Các ví dụ về các dịch vụ mà các công cụ vi tính hóa có thể cung cấp bao gồm:

- Kiểm soát các hoạt động quản lý rủi ro
 - + Danh sách các mục rủi ro phần mềm theo danh mục và ngày dự kiến giải quyết.
 - + Danh sách các ngoại lệ của các mục rủi ro phần mềm – các ngày giải quyết bị trễ có thể ảnh hưởng đến ngày hoàn thành dự án.
- Kiểm soát tiến độ dự án
 - + Danh sách phân loại các hoạt động bị trễ.
 - + Danh sách phân loại các hoạt động quan trọng bị trễ – những hoạt động này nếu không được khắc phục, có thể ảnh hưởng đến ngày hoàn thành dự án.
 - + Lịch trình hoạt động cập nhật được tạo ra theo báo cáo tiến độ và các biện pháp khắc phục đã được áp dụng – cho các nhóm, đơn vị phát triển, v.v.
 - + Danh sách phân loại các mốc bị trễ.

- + Lịch trình mốc cập nhật được tạo ra theo báo cáo tiến độ và các biện pháp khắc phục đã được áp dụng – cho các nhóm, đơn vị phát triển, v.v.
- Kiểm soát nguồn lực dự án
 - + Kế hoạch phân bổ nguồn lực cho các hoạt động và các module phần mềm, cho các nhóm, đơn vị phát triển, giai đoạn thời gian được chỉ định, v.v.
 - + Báo cáo sử dụng nguồn lực dự án – theo giai đoạn hoặc tích lũy – như đã nêu trên.
 - + Ngoại lệ về việc sử dụng nguồn lực dự án – theo giai đoạn hoặc tích lũy – như đã nêu trên.
 - + Kế hoạch phân bổ nguồn lực cập nhật được tạo ra theo báo cáo tiến độ và các biện pháp khắc phục đã được áp dụng.
- Kiểm soát ngân sách dự án
 - + Kế hoạch ngân sách dự án – theo hoạt động và module phần mềm, cho các nhóm, đơn vị phát triển, giai đoạn thời gian được chỉ định, v.v.
 - + Báo cáo sử dụng ngân sách dự án – theo giai đoạn hoặc tích lũy – như đã nêu trên.
 - + Các sai lệch về việc sử dụng ngân sách – theo giai đoạn hoặc tích lũy – như đã nêu trên.
 - + Kế hoạch ngân sách cập nhật được tạo ra theo báo cáo tiến độ và các biện pháp khắc phục đã được áp dụng.

2. Các độ đo chất lượng phần mềm

2.1. Mục tiêu chung khi thu thập các độ đo

Mục tiêu chung khi thu thập các độ đo chất lượng phần mềm (Software Quality Metrics) cho thấy rằng việc thu thập các độ đo chất lượng phần mềm không chỉ đơn giản là để đánh giá hiện trạng mà còn để tạo ra các cơ hội cải tiến, từ đó nâng cao chất lượng sản phẩm và hiệu quả của toàn bộ quy trình phát triển phần mềm.

2.1.1. Hỗ trợ kiểm soát quản lý, lập kế hoạch và thực hiện các can thiệp quản lý thích hợp:

Mục tiêu đầu tiên của việc thu thập các độ đo chất lượng phần mềm là để hỗ trợ quản lý trong việc kiểm soát, lập kế hoạch và thực hiện các biện pháp can thiệp cần thiết trong quá trình phát triển phần mềm. Điều này được thực hiện bằng cách sử dụng các thước đo để so sánh hiệu suất thực tế với các kế hoạch và tiêu chuẩn đã được đề ra trước đó.

Cụ thể, các thước đo sẽ cho phép quản lý xác định sự chênh lệch giữa hiệu suất thực tế và hiệu suất dự kiến trong hai khía cạnh chính:

- Hiệu suất chức năng (chất lượng) thực tế so với những gì đã được lên kế hoạch ban đầu. Ví dụ, nếu một phần mềm dự kiến phải đáp ứng một số yêu cầu chức năng cụ thể, các thước đo sẽ giúp đánh giá xem phần mềm có thực sự đáp ứng được các yêu cầu đó hay không.
- Hiệu suất về thời gian và ngân sách thực tế so với kế hoạch. Điều này bao gồm việc xem xét xem dự án có được hoàn thành trong thời gian và ngân sách đã định trước hay không, và nếu có sự chênh lệch, quản lý cần xác định nguyên nhân và điều chỉnh kế hoạch để tối ưu hóa các nguồn lực.

2.1.2. Xác định các tình huống cần hoặc có thể cải thiện quy trình phát triển hoặc bảo trì:

Mục tiêu thứ hai là để nhận diện những tình huống hoặc điều kiện mà việc cải thiện quy trình phát triển hoặc bảo trì phần mềm là cần thiết hoặc có thể thực hiện. Việc cải thiện này có thể dưới dạng các hành động phòng ngừa (ngăn chặn các vấn đề trước khi chúng xảy ra) hoặc hành động khắc phục (sửa chữa các vấn đề đã xảy ra). Để đạt được mục tiêu này, việc thu thập các độ đo tập trung vào việc tích lũy thông tin liên quan đến hiệu suất của các đội ngũ, đơn vị trong tổ chức. Ví dụ, bằng cách theo dõi các thước đo chất lượng của nhiều đội phát triển khác nhau, quản lý có thể nhận ra những điểm mạnh và yếu của từng đội, từ đó đưa ra các biện pháp để nâng cao hiệu suất chung của tổ chức.

Ngoài ra, việc so sánh các thành tựu chất lượng trung bình của tổ chức với các thực hành tốt nhất trong ngành cũng là một phần quan trọng trong việc xác định những điểm cần cải thiện. Điều này giúp tổ chức không chỉ giữ vững chất lượng hiện tại mà còn tiếp tục phát triển và cải tiến để đạt được các tiêu chuẩn cao hơn trong ngành.

2.2. Phân loại các độ đo chất lượng phần mềm

Phân loại các độ đo chất lượng phần mềm được chia thành hai cấp độ chính. Việc phân loại và đo lường theo các đối tượng này giúp các nhà quản lý và đội ngũ phát triển phần mềm hiểu rõ hơn về các khía cạnh khác nhau của quy trình phát triển và bảo trì, từ đó đưa ra các quyết định cải tiến phù hợp.

2.2.1. Độ đo quy trình (Process Metrics)

Độ đo quy trình liên quan đến quy trình phát triển phần mềm và bao gồm ba phân loại con:

- Độ đo chất lượng quy trình phần mềm (Software Process Quality Metrics): Các thước đo này đánh giá chất lượng của chính quy trình phần mềm.
- Độ đo thời gian quy trình phần mềm (Software Process Timetable Metrics): Các thước đo này đánh giá thời gian liên quan đến các giai đoạn khác nhau của quy trình phát triển phần mềm.
- Độ đo năng suất quy trình phần mềm (Software Process Productivity Metrics): Các thước đo này tập trung vào năng suất của nỗ lực phát triển phần mềm.

2.2.2. Độ đo sản phẩm (Product Metrics)

Độ đo sản phẩm liên quan đến bảo trì phần mềm và được phân loại thành bốn phân loại con:

- Độ đo chất lượng dịch vụ HD (HD Quality Metrics): Các thước đo này đánh giá chất lượng của các dịch vụ HD (độ phân giải cao).
- Độ đo năng suất và hiệu quả HD (HD Productivity and Effectiveness Metrics): Các thước đo này đo lường năng suất và hiệu quả liên quan đến các dịch vụ HD.
- Độ đo chất lượng bảo trì sửa lỗi phần mềm (Software Corrective Maintenance Quality Metrics): Các thước đo này đánh giá chất lượng của việc bảo trì sửa lỗi thực hiện trên phần mềm.
- Độ đo năng suất và hiệu quả bảo trì sửa lỗi phần mềm (Software Corrective Maintenance Productivity and Effectiveness Metrics): Các thước đo này tập trung vào năng suất và hiệu quả đặc biệt trong bảo trì sửa lỗi.

2.3. Giải thích công thức độ đo

2.3.1. Error Density Metrics

a. Code Error Density (CED)

Là một chỉ số mật độ lỗi mã nguồn, được tính bằng cách chia số lượng lỗi mã phát hiện được cho số lượng hàng nghìn dòng mã. Chỉ số này cho biết số lượng lỗi trung bình trên mỗi KLOC, giúp đánh giá mức độ chất lượng của mã nguồn. Giá trị CED càng thấp, chất lượng mã càng cao.

Công thức : $CED = NCE / KLOC$

Trong đó : NCE (Number of Code Errors) là số lượng lỗi mã được phát hiện trong quá trình kiểm tra và thử nghiệm mã nguồn. KLOC (Thousands of Lines of Code): Số lượng hàng nghìn dòng mã.

b. Weighted Code Error Density (WCED)

Là chỉ số mật độ lỗi mã nguồn có trọng số, được tính bằng cách chia số lượng lỗi mã đã được trọng số hóa cho số lượng hàng nghìn dòng mã. Trọng số được xác định dựa trên mức độ nghiêm trọng của lỗi, do đó, WCED cung cấp một đánh giá chính xác hơn về chất lượng mã nguồn so với CED thông thường. Nếu WCED cao hơn một mức nhất định, điều đó cho thấy rằng chất lượng mã cần được cải thiện.

Công thức: $WCED = WCE / KLOC$

Trong đó WCE (Weighted Code Errors) là số lượng lỗi mã đã được tính trọng số dựa trên mức độ nghiêm trọng của lỗi và KLOC (Thousands of Lines of Code) là số lượng hàng nghìn dòng mã.

2.3.2. *Development Productivity Metrics*

a. Development Productivity (DevP)

Công thức: $DevP = Total Output / Total Input$

Giải thích: Công thức này định lượng năng suất trong phát triển phần mềm. Tổng Sản lượng có thể được đo lường bằng số điểm chức năng hoàn thành, và Tổng Đầu vào có thể được đại diện bởi số giờ công hoặc tài nguyên tiêu tốn trong dự án. Bằng cách tính toán DevP, quản lý có thể đánh giá hiệu quả sử dụng tài nguyên phát triển.

b. Function Point Corrective Maintenance Productivity (FCMP)

Công thức: $FCMP = CMaiYH / NMFP$

Giải thích: FCMP đo lường năng suất của các hoạt động bảo trì sửa chữa, trong đó *CMaiYH* là tổng số giờ công hàng năm đầu tư vào bảo trì sửa chữa, và *NMFP* đại diện cho số điểm chức năng được chỉ định cho phần mềm được bảo trì. Chỉ số này là một chỉ báo về hiệu quả của các nỗ lực bảo trì so với độ phức tạp của phần mềm đang được bảo trì.

Tóm tắt những công thức này cung cấp những thông tin quan trọng về chất lượng phần mềm và hiệu quả hoạt động. Các Chỉ số Mật độ Lỗi như CED và WCED giúp xác định chất lượng của mã bằng cách định lượng số lỗi so với kích thước mã. Ngược lại, các Chỉ số Năng suất Phát triển như DevP và FCMP rất quan trọng để đánh giá hiệu quả của quá trình phát triển và bảo trì, đảm bảo rằng các tài nguyên đang được sử dụng một cách khôn ngoan trong các dự án phần mềm. Mỗi chỉ số phục vụ cho các mục đích riêng biệt nhưng bổ sung lẫn nhau trong việc giám sát chất lượng và năng suất phần mềm.

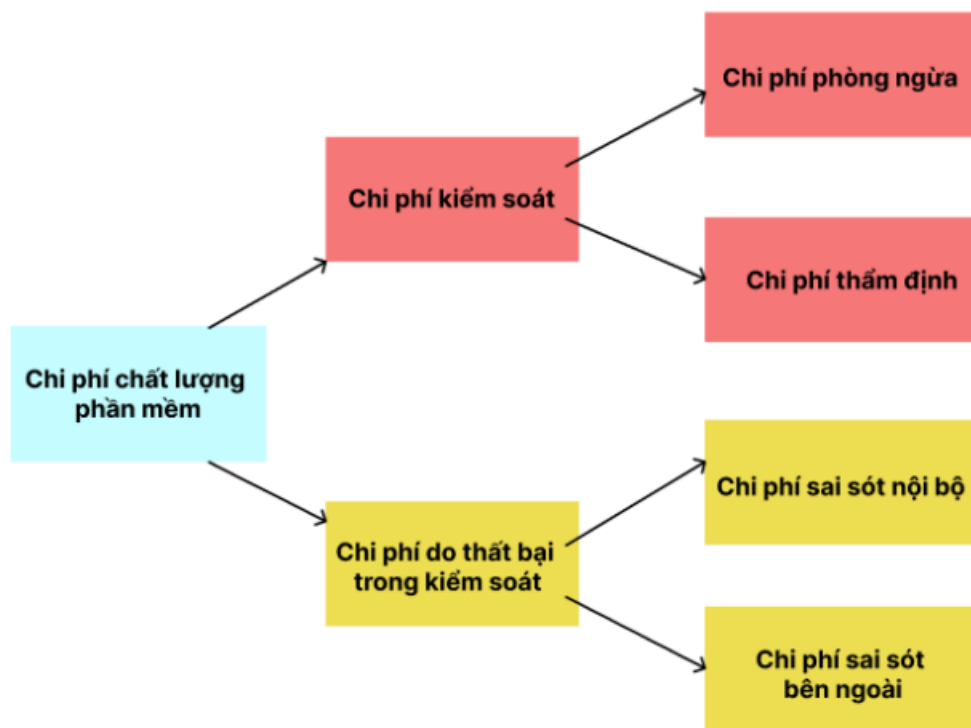
3. Chi phí chất lượng phần mềm

3.1. Mục tiêu của chi phí chất lượng phần mềm

Việc áp dụng các số liệu về chất lượng phần mềm cho phép ban quản lý đạt được sự kiểm soát kinh tế và kết quả của SQA. Các mục tiêu cụ thể là:

- Điều khiển chi phí do tổ chức khởi xướng để ngăn ngừa và phát hiện lỗi phần mềm.
- Đánh giá thiệt hại kinh tế do lỗi phần mềm làm cơ sở điều chỉnh ngân sách SQA.
- Đánh giá các kế hoạch tăng hoặc giảm các hoạt động SQA hoặc đầu tư vào cơ sở hạ tầng SQA mới hoặc cập nhật trên cơ sở hiệu quả kinh tế trong quá khứ. Mô hình cổ điển về chi phí chất lượng phần mềm

3.2. Mô hình cổ điển về chi phí chất lượng phần mềm



Hình 1: Mô hình cổ điển về chi phí chất lượng phần mềm

Mô hình chi phí chất lượng cổ điển được phát triển vào đầu những năm 1950 bởi Feigenbaum và những người khác, cung cấp một phương pháp để phân loại chi phí liên quan đến việc đảm bảo chất lượng sản phẩm xét theo khía cạnh kinh tế của quan điểm. Được phát triển để phù hợp với các tình huống chất lượng được tìm thấy trong sản xuất các tổ chức, mô hình này đã được triển khai rộng rãi.

Mô hình phân loại chi phí liên quan đến chất lượng sản phẩm thành hai nhóm chung:

3.2.1. Chi phí kiểm soát

a. Chi phí phòng ngừa

Đầu tư phát triển cơ sở hạ tầng SQA mới hoặc cải tiến

- Quy trình và hướng dẫn công việc
- Thiết bị hỗ trợ: mẫu, danh sách kiểm tra, v.v.
- Hệ thống quản lý cấu hình phần mềm
- Thước đo chất lượng phần mềm

Thực hiện thường xuyên các hoạt động phòng ngừa SQA:

- Hướng dẫn nhân viên mới về các môn SQA
- Chứng nhận của nhân viên
- Tư vấn về các vấn đề SQA cho trưởng nhóm và những người khác

Kiểm soát hệ thống SQA thông qua việc thực hiện:

- Đánh giá chất lượng nội bộ
- Đánh giá chất lượng bên ngoài của khách hàng và tổ chức chứng nhận hệ thống SQA
- Đánh giá chất lượng quản lý

b. Chi phí thẩm định

Chi phí thẩm định được dành cho việc phát hiện lỗi phần mềm trong các dự án hoặc hệ thống phần mềm cụ thể. Chi phí thẩm định điển hình bao gồm:

Chi phí đánh giá:

- Đánh giá thiết kế chính thức (DR)
- Đánh giá ngang hàng (kiểm tra và hướng dẫn)
- Đánh giá của chuyên gia

Chi phí kiểm thử phần mềm:

- Kiểm tra đơn vị, tích hợp và hệ thống phần mềm
- Kiểm tra chấp nhận (do khách hàng thực hiện)

Chi phí đảm bảo chất lượng của các bên tham gia bên ngoài

- Nhà thầu phụ, nhà cung cấp hệ thống phần mềm COTS và các mô-đun phần mềm có thể tái sử dụng, khách hàng
- Khách hàng là người tham gia thực hiện dự án

3.2.2. Chi phí do sai sót trong kiểm soát**a. Chi phí do sai sót nội bộ**

- Là những chi phí phát sinh lỗi đã được phát hiện sau trong quá trình xem xét thiết kế, kiểm tra phần mềm và kiểm tra chấp nhận được thực hiện trước khi phần mềm được cài đặt tại địa điểm của khách hàng hay nói cách khác, chi phí lỗi nội bộ thể hiện chi phí sửa lỗi sau khi kiểm tra chính thức phần mềm trong quá trình phát triển phần mềm, trước khi cài đặt hệ thống tại địa điểm của khách hàng

- Chi phí thiết kế lại hoặc chỉnh sửa thiết kế sau khi xem xét thiết kế và kết quả thử nghiệm
- Chi phí lập trình lại hoặc sửa chữa các chương trình để đáp ứng các kết quả kiểm tra
- Chi phí xem xét thiết kế lặp đi lặp lại và thử nghiệm lại (kiểm tra hồi quy)
- Điều quan trọng là mặc dù chi phí cho việc xem xét lại thiết kế và kiểm tra phần mềm thường xuyên được coi là chi phí thẩm định, nhưng bất kỳ việc xem xét lại thiết kế hoặc kiểm tra phần mềm lặp lại nào trực tiếp phát sinh từ thiết kế kém chất lượng đều được coi là chi phí do lỗi nội bộ.

b. Chi phí sai sót bên ngoài

Đòi hỏi chi phí khắc phục các lỗi do khách hàng hoặc đội bảo trì phát hiện sau khi hệ thống phần mềm được cài đặt tại địa điểm của khách hàng.

Trong hầu hết các trường hợp, mức độ của chi phí ẩn lớn hơn nhiều so với chi phí công khai.

Chi phí sai sót bên ngoài điển hình (chi phí “công khai”)

- Giải quyết khiếu nại của khách hàng: Trong hầu hết các trường hợp, việc này liên quan đến việc xem xét khiếu nại hoặc chuyển hướng dẫn. Trong hầu hết các trường hợp, khiếu nại là do chức năng “trợ giúp” hoặc các hướng dẫn trong sách hướng dẫn không hoạt động.
- Sửa lỗi phần mềm được phát hiện trong quá trình hoạt động thường xuyên: Những việc liên quan đến sửa mã (bao gồm kiểm tra phần mềm đã sửa) sau đó là cài đặt mã sửa hoặc thay thế các phiên bản lỗi bằng các phiên bản đúng thường được thực hiện tại trang web của khách hàng.
- Khắc phục các lỗi phần mềm sau thời gian bảo hành
- Bồi thường thiệt hại cho khách hàng
- Hoàn trả chi phí mua hàng của khách hàng
- Bảo hiểm đối với các khiếu nại của khách hàng
- Các hạng mục được liệt kê chỉ phản ánh chi phí sai sót bên ngoài, chi phí chiếm một tỷ lệ nhỏ trong toàn bộ chi phí sai sót bên ngoài. Những chi phí này phát sinh trực tiếp do lỗi phần mềm được phát hiện và ghi lại trong quá trình vận hành thường xuyên của phần mềm.

Ví dụ điển hình về chi phí sai sót bên ngoài tiềm ẩn

- Giảm doanh số bán hàng cho khách hàng bị lỗi phần mềm
- Doanh số bán hàng sụt giảm nghiêm trọng do danh tiếng của công ty bị tổn hại
- Tăng cường đầu tư vào xúc tiến bán hàng để chống lại tác động của lỗi phần mềm trong quá khứ

3.3. Mô hình mở rộng về chi phí chất lượng phần mềm

Phân tích chi phí chất lượng phần mềm được xác định theo mô hình cổ điển cho thấy một số chi phí có tầm quan trọng đáng kể đã bị loại trừ. Những chi phí này bao gồm là

duy nhất đối với ngành công nghiệp phần mềm hoặc không đáng kể đối với ngành công nghiệp khác.

Ví dụ: Chi phí do lỗi chất lượng phần mềm điển hình bao gồm:

- Những thiệt hại được trả cho khách hàng do việc hoàn thành công việc chậm trễ dự án do kế hoạch không thực tế.
- Thiệt hại do khách hàng phải bồi thường do dự án chậm hoàn thành do không tuyển đủ nhân viên.

Yếu tố chung của hai thất bại này là chúng không xuất phát từ bất kỳ hành động cụ thể của nhóm phát triển hoặc bất kỳ sự thiếu chuyên nghiệp nào, chúng thực sự là kết quả của sự thất bại trong quản lý.

Ban quản lý có thể thực hiện một số hoạt động để ngăn ngừa hoặc giảm chi phí do các loại sai sót cụ thể đối với chức năng của mình:

- Rà soát hợp đồng: Rà soát dự thảo đề xuất và rà soát dự thảo hợp đồng.
- Kiểm tra tiến độ dự án phần mềm một cách thích hợp và triệt để

Theo mô hình mở rộng, hai phân lớp được thêm vào để hoàn thiện phạm vi của mô hình: chi phí chuẩn bị và kiểm soát quản lý và chi phí thất bại trong quản lý.



Hình 2: Chi phí mở rộng của mô hình chất lượng phần mềm

3.3.1. Chi phí chuẩn bị và kiểm soát quản lý

Chi phí này gắn với các hoạt động thực hiện nhằm ngăn chặn những thất bại trong quản lý hoặc giảm thiểu khả năng xảy ra những thất bại đó. Chi phí chuẩn bị kiểm soát quản lý điển hình bao gồm:

- Chi phí thực hiện rà soát hợp đồng
- Chi phí chuẩn bị kế hoạch dự án, bao gồm cả kế hoạch chất lượng
- Chi phí cập nhật định kỳ kế hoạch dự án và chất lượng
- Chi phí thực hiện kiểm soát tiến độ thường xuyên
- Chi phí thực hiện kiểm soát tiến độ thường xuyên đối với sự đóng góp của những người tham gia bên ngoài vào dự án

3.3.2. Chi phí thất bại trong quản lý

Chi phí này có thể phát sinh trong toàn bộ quá trình phát triển phần mềm, bắt đầu từ giai đoạn tiền dự án. Chúng thường xuất hiện do những nỗ lực thất bại trong việc ước tính tiến độ và ngân sách phù hợp của dự án cũng như phát hiện kịp thời những sai lệch và cần đề cần sự can thiệp của ban quản lý.

Chi phí thất bại trong quản lý điển hình bao gồm:

- **Chi phí ngoài kế hoạch** cho các nguồn lực chuyên môn và các nguồn lực khác, do đánh giá thấp các nguồn lực trong giai đoạn đề xuất
- **Những thiệt hại** được trả cho khách hàng như là sự đền bù cho việc hoàn thành dự án chậm, là kết quả của việc không thực tế lịch trình trong đề xuất của Công ty
- **Những thiệt hại** được trả cho khách hàng như một khoản bồi thường cho việc hoàn thành dự án chậm trễ do ban quản lý không tuyển được thành viên trong nhóm
- **Hiệu ứng Domino** : Thiệt hại đối với các dự án khác dự kiến được thực hiện bởi chính các nhóm tham gia vào các dự án bị trì hoãn. Hiệu ứng domino có thể gây ra chi phí thất bại bên ngoài đáng kể.

CHƯƠNG 3. KẾT LUẬN

Kiểm soát tiến độ dự án, thu thập độ đo chất lượng phần mềm, và quản lý chi phí chất lượng là những yếu tố then chốt quyết định sự thành công của bất kỳ dự án phần mềm nào. Việc kiểm soát tiến độ giúp các nhà quản lý dự án đảm bảo rằng mọi công việc được thực hiện đúng theo kế hoạch, các nguồn lực được phân bổ một cách hiệu quả và các rủi ro tiềm tàng được nhận diện và xử lý kịp thời. Qua quá trình giám sát chặt chẽ, các tổ chức có thể duy trì sự ổn định trong việc thực hiện dự án, hạn chế tối đa những sai lệch không mong muốn về thời gian và tài chính.

Đồng thời, việc thu thập và phân tích các độ đo chất lượng phần mềm đóng vai trò quan trọng trong việc xác định hiệu quả của quá trình phát triển, từ đó giúp phát hiện sớm những lỗi phát sinh và khắc phục kịp thời. Các độ đo này không chỉ là cơ sở để đánh giá chất lượng phần mềm mà còn là công cụ hỗ trợ quá trình cải tiến liên tục, đảm bảo sản phẩm phần mềm đạt được các tiêu chuẩn về hiệu suất, độ tin cậy và tính bảo trì.

Bên cạnh đó, quản lý chi phí chất lượng là một yếu tố không thể thiếu để đảm bảo rằng dự án được thực hiện trong giới hạn ngân sách, đồng thời mang lại giá trị cao nhất cho các bên liên quan. Các mô hình chi phí chất lượng phần mềm, bao gồm mô hình cổ điển và mô hình mở rộng, cung cấp cho các nhà quản lý một phương pháp cụ thể để dự đoán, kiểm soát và tối ưu hóa chi phí liên quan đến phát triển, kiểm thử và bảo trì phần mềm.

Tiểu luận này đã cung cấp cái nhìn toàn diện về ba yếu tố cốt lõi trong quản lý dự án phần mềm: kiểm soát tiến độ, thu thập độ đo chất lượng và quản lý chi phí chất lượng. Từ việc đảm bảo tiến độ dự án, quản lý hiệu quả các chỉ số chất lượng, đến tối ưu hóa chi phí phát triển, các yếu tố này kết hợp với nhau tạo nên sự thành công bền vững cho các dự án phần mềm. Những kiến thức và công cụ đã được trình bày không chỉ hỗ trợ các nhà quản lý trong việc ra quyết định mà còn giúp đảm bảo rằng sản phẩm cuối cùng không chỉ đáp ứng được các yêu cầu kỹ thuật mà còn mang lại giá trị cao cho tổ chức và khách hàng.

CHƯƠNG 4. TÀI LIỆU THAM KHẢO

1. Galin, Daniel. *Software quality assurance: from theory to implementation*. Pearson education, 2004.