

BỘ CÂU HỎI ÔN TẬP MÔN QUẢN LÝ DỰ ÁN PHẦN MỀM

(TỪ NĂM HỌC 2025-2026)

Câu 1: Trình bày khái niệm dự án và các đặc trưng cơ bản của dự án. Dự án phần mềm có gì khác biệt so với các dự án nói chung?

Khái niệm dự án:

Dự án là một nỗ lực tạm thời nhằm tạo ra một sản phẩm, dịch vụ hoặc kết quả duy nhất. Nó có

thời gian bắt đầu và kết thúc rõ ràng, với phạm vi và nguồn lực cụ thể.

Các đặc trưng cơ bản của dự án:

1. **Tính tạm thời:** Có thời điểm bắt đầu và kết thúc xác định.
2. **Tính duy nhất:** Kết quả cuối cùng là độc đáo, không lặp lại.
3. **Giới hạn nguồn lực:** Thời gian, ngân sách, nhân lực đều có giới hạn.
4. **Tính phức tạp và không chắc chắn:** Đặc biệt trong giai đoạn đầu của dự án.

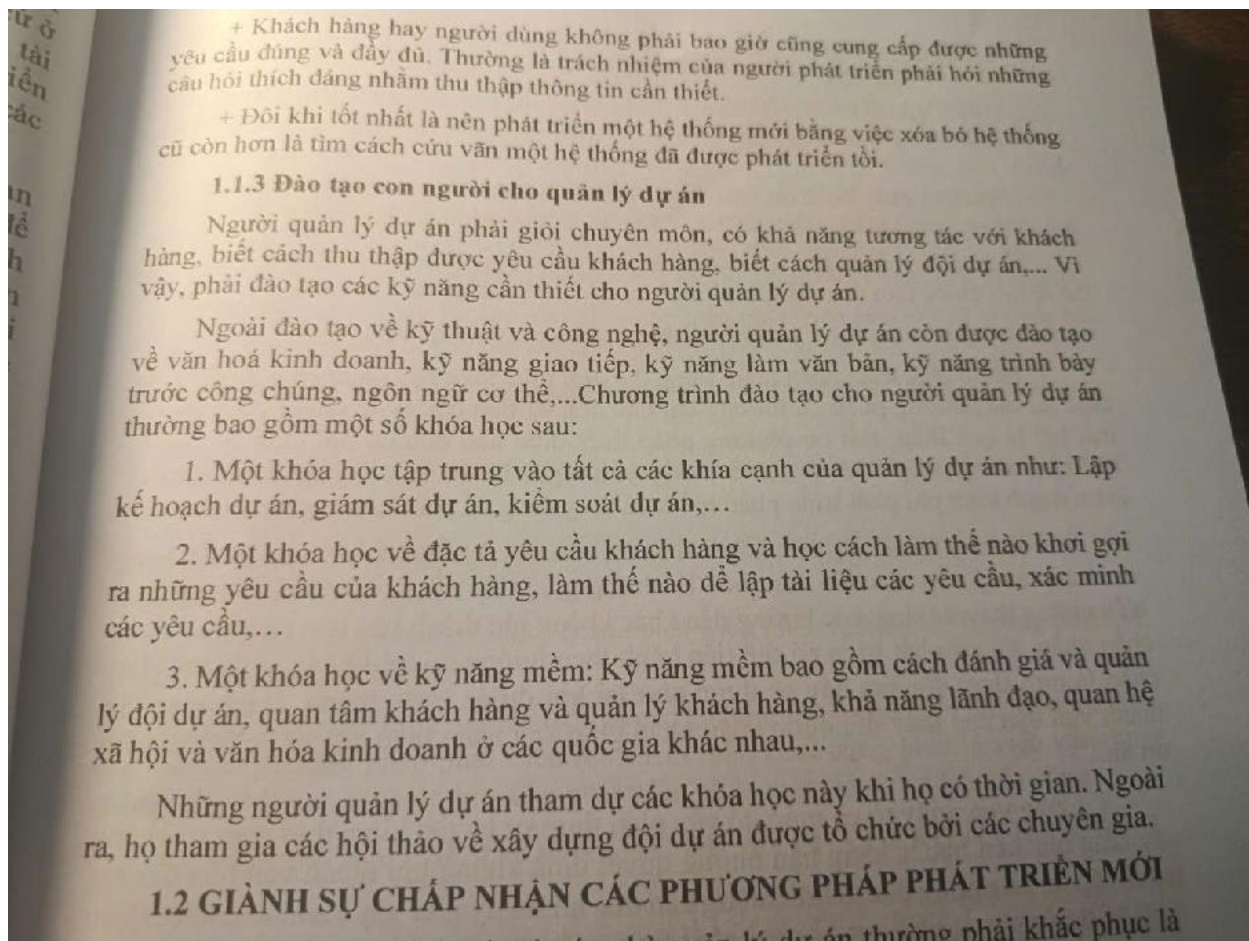
Sự khác biệt của dự án phần mềm:

- **Sản phẩm phi vật lý:** Khó đo lường tiến độ, chất lượng bằng cảm quan.
- **Thường xuyên thay đổi yêu cầu:** Do người dùng thay đổi hoặc hiểu sai ban đầu.
- **Khó kiểm soát và ước lượng:** Chi phí, thời gian và effort phụ thuộc nhiều vào kinh nghiệm.
- **Phụ thuộc vào con người:** Kiến thức, kỹ năng, phối hợp giữa các thành viên rất quan trọng.

Ví dụ:

Dự án xây dựng phần mềm quản lý sinh viên khác với dự án xây dựng tòa nhà vì nó khó thấy được tiến độ rõ ràng, yêu cầu dễ thay đổi, sản phẩm vô hình

Câu 2: Người quản lý dự án cần phải có những phẩm chất gì? Họ thường phải tham gia những khóa học nào?



Câu 3: Nêu các vấn đề cơ bản trong các dự án phần mềm. Trình bày sự ảnh hưởng của vấn đề “Thay đổi yêu cầu không kiểm soát được” tới sự thành công của dự án.

Các vấn đề cơ bản trong dự án phần mềm:

1. **Yêu cầu không rõ ràng/người dùng không xác định được yêu cầu chính xác.**
2. **Thay đổi yêu cầu liên tục:** Thường đến từ người dùng hoặc nhà quản lý.
3. **Ước lượng sai lệch:** Về thời gian, nhân lực, chi phí.
4. **Thiếu tài nguyên hoặc nhân lực không đủ kỹ năng.**
5. **Thiếu quy trình kiểm soát chất lượng hoặc cấu hình.**

Ảnh hưởng của việc thay đổi yêu cầu không kiểm soát được:

- **Gây trễ tiến độ:** Mỗi thay đổi cần thời gian phân tích, điều chỉnh thiết kế, mã hóa và kiểm thử lại.
- **Tăng chi phí:** Phát sinh thêm nhân lực, giờ làm việc, chỉnh sửa tài liệu và phần mềm.
- **Mất kiểm soát phạm vi dự án:** Khi có quá nhiều thay đổi, người quản lý không xác định được đâu là mục tiêu cuối cùng.
- **Chất lượng giảm:** Vì thời gian không đủ để kiểm thử đầy đủ sau khi thay đổi.
- **Gây bất ổn cho nhóm:** Nhóm bị phân tâm, giảm tinh thần khi phải làm đi làm lại.

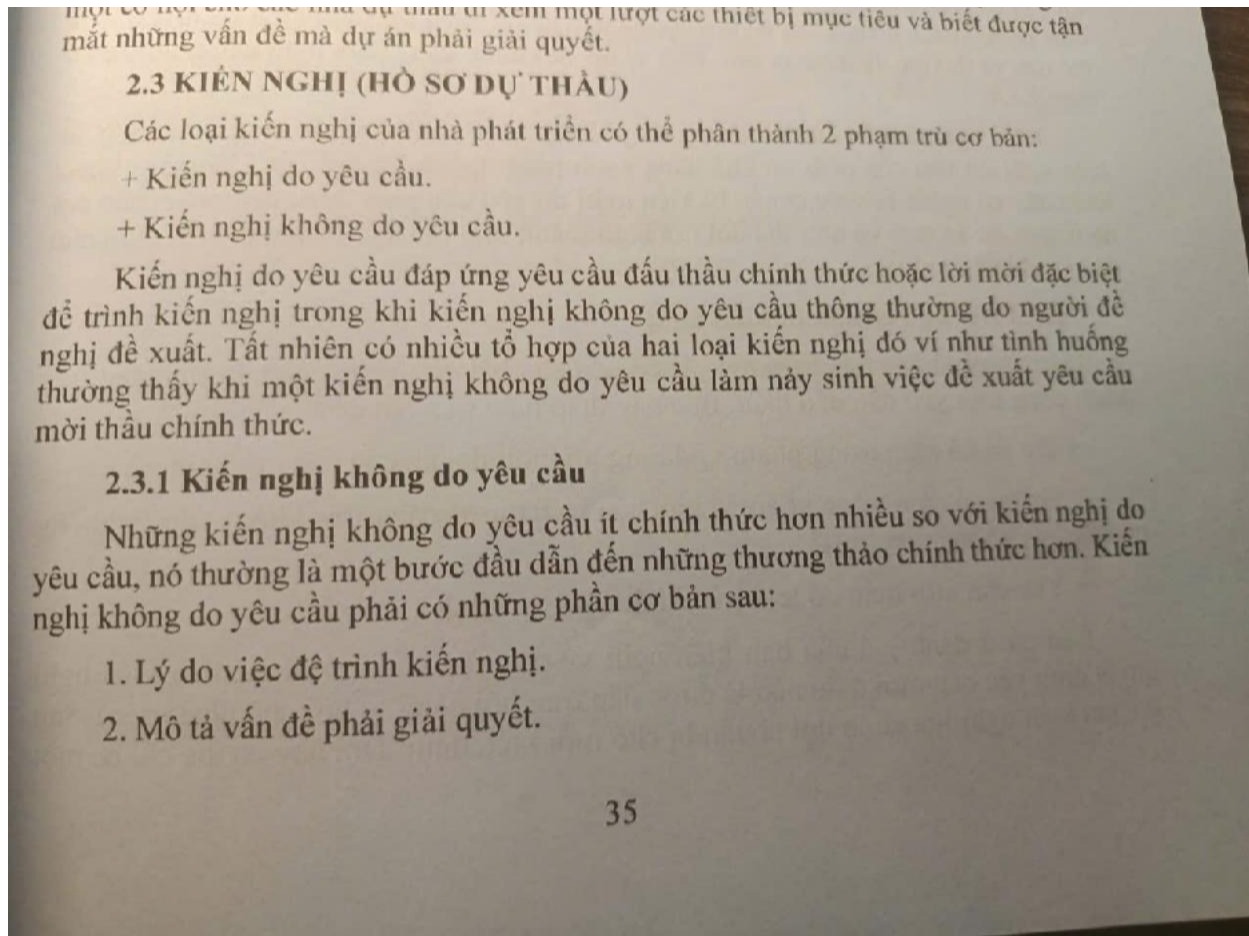
Giải pháp:

- **Áp dụng quản lý yêu cầu chặt chẽ** (quy trình phê duyệt, traceability).

- Thiết lập baseline yêu cầu.
- Sử dụng các mô hình phát triển linh hoạt (Agile) nếu dự án cần thay đổi thường xuyên.

Câu 4: Anh (Chị) hãy trình bày khái niệm về hợp đồng phí cộng thêm và hợp đồng giá cố định. Sự khác nhau cơ bản giữa chúng?

Câu 5: Anh (Chị) hãy trình bày về kiến nghị không do yêu cầu và kiến nghị do yêu cầu. Sự khác nhau cơ bản giữa chúng?



3. Mô tả giải pháp đề nghị.

4. Mô tả tổ chức của người kiến nghị.

5. Tổng quan chung về kinh phí của giải pháp đề nghị.

Phần lý do là quan trọng vì nó giải thích vì sao khách hàng nên đọc tiếp bản kiến nghị. Phần này có thể nêu, chẳng hạn người kiến nghị đã phát triển một công nghệ mới và thành công có lợi cho khách hàng và có thể thích nghi với nhu cầu của tổ chức khách hàng. Mục tiêu chính là cung cấp câu trả lời cho câu hỏi của khách hàng: "Vì sao công ty (nhà phát triển) lại tiếp cận tôi và vì sao đọc kiến nghị này là lợi ích cho tôi vậy?"

Các phần 2 và 3, mô tả cách mà kiến thức chuyên môn hóa của người đề nghị được vận dụng vào các vấn đề của tổ chức khách hàng. Điều này đòi hỏi người đề nghị nghiên cứu tổ chức của khách hàng nhằm đảm bảo là kiến nghị của họ cung cấp một giải pháp thực tế cho vấn đề của khách hàng.

Chi phí chính xác của giải pháp không cần nêu ở giai đoạn này vì một kiến nghị không do yêu cầu hiếm khi được chấp nhận ngay vòng đầu.

Mục tiêu chính của loại kiến nghị không do yêu cầu là tạo dựng một mối quan tâm của tổ chức khách hàng. Nếu kiến nghị tạo đủ mối quan tâm thì người đề nghị sẽ được mời thảo luận kiến nghị đó và sau đó sẽ đệ trình lại cho khách hàng một bản trình bày kiến nghị chi tiết hơn.

2.3.2 Kiến nghị do yêu cầu

Trái với tính chất không chính thức của kiến nghị không do yêu cầu, kiến nghị do yêu cầu (khách hàng yêu cầu) là hoàn chỉnh và chi tiết. Cùng với yêu cầu đệ trình kiến nghị, khách hàng cũng có thể hướng dẫn chính xác kiến nghị phải được chuẩn bị thế nào và thủ tục đệ trình ra sao. Một ví dụ về khuôn dạng kiến nghị chính thức có ở mục 2.3.4.

Kiến nghị do yêu cầu và không do yêu cầu khác nhau ở nhu cầu cạnh tranh. Các kiến nghị do yêu cầu phải có khả năng cạnh tranh thắng lợi với các kiến nghị khác. Điều này có nghĩa là việc chuẩn bị kiến nghị do yêu cầu phải được nhà phát triển coi nó là một dự án nhỏ và như thế đòi hỏi hình thành một đội ngũ chuẩn bị kiến nghị chu đáo và chuyên nghiệp.

Câu 6: Cơ sở dữ liệu tiến trình là gì? Tại sao lại cần phải có nó? Những loại dữ liệu chính thường được lưu giữ trong cơ sở dữ liệu tiến trình là gì?

này sẽ thảo luận các vấn đề tra cứu dữ liệu tiến trình; Baseline về khả năng của tiến trình tri thức.

Cơ sở dữ liệu tiến trình lưu giữ dữ liệu của các dự án đã hoàn thành. Baseline về khả năng của tiến trình tóm tắt hiệu suất của nhiều dự án khác nhau, nó xác định phạm vi của các kết quả được mong đợi nếu loại tiến trình tương tự được thực hiện theo. Các thành phần của tiến trình là các tài liệu như các bản liệt kê những mục cần kiểm tra, các biểu mẫu, các phương pháp, và các bài học kinh nghiệm. Hệ tri thức là những kinh nghiệm quá khứ và giúp những người quản lý dự án và các kỹ sư sử dụng hiệu quả các tiến trình.

3.1 CƠ SỞ DỮ LIỆU TIẾN TRÌNH

Cơ sở dữ liệu tiến trình là một kho lưu trữ lâu dài dữ liệu tiến trình của các dự án đã được thực hiện bởi công ty. Nó có thể được sử dụng để lập kế hoạch, ước lượng, phân tích năng suất và chất lượng, và những mục đích khác cho dự án sẽ được thực hiện. Nó bao gồm dữ liệu từ các dự án đã hoàn thành, mỗi dự án cung cấp một bản ghi (record) trong cơ sở dữ liệu tiến trình. Để được lưu trữ trong cơ sở dữ liệu, dữ liệu phải được thu thập, phân tích, và sau đó tổ chức theo khuôn dạng để nhập vào. Ở chương này chúng ta tập trung vào cách trình bày dữ liệu trong cơ sở dữ liệu tiến trình; chương 8 sẽ thảo luận về việc dữ liệu được thu thập như thế nào.

được ước lượng bằng giai đoạn phát triển kế, pha cài đặt và thu thập những dữ

Kích thước of code), số lượng năng (FP - Function ước lượng, bản chuyển đổi kích chuyển đổi [1]

Dữ liệu và hiện lỗi khác nhau. Do đó, cần ghi yêu cầu, xem

Ngoài ra, trình, bao gồm một chương trình (ví dụ, làm th

3.1.1 Nội dung của cơ sở dữ liệu tiến trình

Để sử dụng các thông tin trong cơ sở dữ liệu tiến trình cho việc lập kế hoạch dự án, người quản lý dự án tìm kiếm thông tin về các dự án tương tự đã thực hiện bởi công ty trong cơ sở dữ liệu tiến trình. Để cho phép kiểm tra sự tương tự, người quản lý dự án nên nắm bắt các thông tin chung trong cơ sở dữ liệu tiến trình về dự án, chẳng hạn như ngôn ngữ được sử dụng, nền tảng, hệ quản trị cơ sở dữ liệu được sử dụng, các công cụ được sử dụng, kích thước và nỗ lực.

▪ **Dữ liệu trong cơ sở dữ liệu tiến trình có thể được phân loại như sau:**

- + Các đặc trưng của dự án.
- + Thời gian biểu của dự án.
- + Nỗ lực của dự án.
- + Kích thước của phần mềm.
- + Lỗi của các sản phẩm.

Dữ liệu về các đặc trưng của dự án bao gồm tên dự án, tên người quản lý dự án và các người lãnh đạo mô-đun, đơn vị kinh doanh, mô hình tiến trình được triển khai, ngôn ngữ lập trình được sử dụng, hệ quản

Câu 7: Anh (Chị) hãy trình bày về baseline của một tiến trình: Khái niệm, vai trò và các loại baseline.

1. Khái niệm

Baseline về khả năng của một tiến trình là một hình ảnh tức thời về khả năng đã được định lượng của tiến trình đó tại một số điểm thời gian nhất định. Nó được xác định dựa trên năng lực thực tế trong quá khứ của tiến trình.

2. Vai trò

- **Dự báo kết quả:** Về bản chất, baseline là phạm vi của các kết quả dự kiến của một dự án nếu tiến trình đó được áp dụng.
- **Cơ sở đánh giá:** Nó cung cấp các tham số định lượng giúp các công ty phần mềm đánh giá và quản lý dự án hiệu quả hơn. Khi một dự án cụ thể được thực hiện thường xuyên, baseline giúp đưa ra kết quả dự kiến có độ sai lệch ít hơn cho các dự án cùng loại.
- **Định lượng các chỉ số:** Baseline cung cấp dữ liệu cụ thể về các tham số quan trọng như:
 - **Chất lượng:** Số lỗi trên mỗi Function Point (FP) hoặc số lỗi theo giờ công.
 - **Năng suất:** Số FP được thực hiện trong một tháng công (ví dụ: 12 FP/tháng cho ngôn ngữ thế hệ 3).

- **Lịch biểu:** Tỷ lệ sản phẩm được giao trong khoảng thời gian thỏa thuận.
- **Nỗ lực:** Phân phối nỗ lực cho các giai đoạn như yêu cầu & thiết kế, xây dựng, thử nghiệm, đào tạo,.
- **Lỗi:** Tỷ lệ tìm lỗi, hiệu quả loại bỏ lỗi và phân phối phát hiện lỗi.

3. Các loại Baseline

Tùy thuộc vào cấu trúc và mục tiêu của tổ chức, baseline thường được phân loại như sau:

- **Baseline theo loại tiến trình:** Các công ty phần mềm thường tạo ra các baseline riêng biệt cho mỗi loại tiến trình trong công ty, ví dụ như: tiến trình bảo trì, tiến trình tái cấu trúc, và tiến trình phát triển.
- **Baseline theo loại dự án:** Một baseline riêng biệt có thể được định nghĩa cho mỗi loại dự án cụ thể. Ví dụ, bảng 3.5 trong tài liệu trình bày baseline tổng quát cho các dự án phát triển sử dụng ngôn ngữ lập trình thể hệ thứ ba.

Câu 8: BOK là gì? Kiến thức trong BOK bao gồm những chủ đề gì? Chúng đến từ đâu?

1. BOK là gì?

BOK (Body of Knowledge), hay còn gọi là **khung kiến thức** hoặc **hệ tri thức**, là một hệ thống được các công ty phần mềm sử dụng để **đóng gói kinh nghiệm** của công ty. Bên cạnh cơ sở dữ liệu tiến trình và các thành phần của tiến trình, BOK giúp ghi nhận và tái sử dụng các hình thức kiến thức khác nhau nhằm quản lý tri thức của nhân viên một cách hiệu quả.

2. Các chủ đề kiến thức trong BOK

Kiến thức trong BOK chủ yếu tồn tại dưới dạng các bài viết và được tổ chức theo các chủ đề chính sau đây:

- Máy tính và các dịch vụ truyền thông.
- Đặc tả yêu cầu.
- Thiết kế.
- Xây dựng.
- Xem xét lại, kiểm tra, kiểm thử.
- Các công cụ.
- Các phương pháp và kỹ thuật.
- Giáo dục và nghiên cứu.
- Đảm bảo chất lượng và năng suất.
- Quản lý dự án.

3. Nguồn gốc của các kiến thức trong BOK

Kiến thức trong BOK đến từ các nguồn và quy trình cụ thể sau:

- **Từ kinh nghiệm nhân viên:** Hệ thống được phát triển để dùng lại một cách hiệu quả các kinh nghiệm và kiến thức của nhân viên trong công ty.
- **Bài học kinh nghiệm và thực hành tốt nhất:** BOK đăng các bài viết liên quan đến những bài học kinh nghiệm thực tế và các phương pháp thực hành tốt nhất (best practices).
- **Đóng góp từ mọi thành viên:** Bất kỳ thành viên nào trong công ty cũng có thể gửi nội dung để đưa vào BOK.
- **Quy trình xét duyệt:** Các bài viết không được đăng tự do mà phải trải qua một **quá trình xét duyệt** nghiêm ngặt về tính hữu dụng, tính tổng quát và các tiêu chuẩn chất lượng trước khi được công bố,.

Câu 9: Anh (Chị) hãy nêu những mô hình công nghệ phần mềm nào đã và đang được sử dụng trên thế giới. Mô hình tiến trình công nghệ phần mềm chuẩn được sử dụng ở Infosys được xây dựng bằng cách nào.

Dựa trên các hình ảnh tài liệu được cung cấp, dưới đây là câu trả lời chi tiết cho các nội dung bạn yêu cầu:

1. Các mô hình công nghệ phần mềm trên thế giới

Hiện nay có nhiều mô hình tiến trình phát triển phần mềm đang tồn tại và được sử dụng rộng rãi, bao gồm:

- **Mô hình thác nước (Waterfall model):** Là mô hình được sử dụng rộng rãi nhất, tổ chức các giai đoạn phát triển theo một trình tự nhất định.
- **Mô hình tăng trưởng theo vòng lặp (Iterative enhancement model):** Phát triển sản phẩm qua nhiều chu kỳ lặp lại.
- **Mô hình tạo bản mẫu (Prototyping model):** Xây dựng các bản dùng thử để làm rõ yêu cầu.
- **Mô hình xoắn ốc (Spiral model):** Tập trung vào việc phân tích rủi ro trong từng giai đoạn.
- **Mô hình RUP (Rational Unified Process):** Được coi là mô hình hiện đại nhất, thường áp dụng cho các dự án lớn và phức tạp.

2. Cách xây dựng mô hình tiến trình chuẩn tại Infosys

Mô hình tiến trình phát triển chuẩn tại công ty Infosys được xây dựng dựa trên các nguyên tắc sau:

- **Dựa trên mô hình thác nước:** Tiến trình này được phát triển dựa trên nền tảng của **mô hình thác nước truyền thống**.

- **Chia nhỏ hoạt động để thực hiện song song:** Điểm khác biệt cốt lõi là các hoạt động truyền thống của mô hình thác nước được **phân chia thành các hoạt động nhỏ hơn**. Cách làm này cho phép một số hoạt động có thể **thực hiện song song** thay vì phải đợi kết thúc hoạt động trước đó.
- **Ví dụ tiêu biểu về tính song song:** Hoạt động "**lập kế hoạch kiểm thử hệ thống**" được tách ra thành một nhiệm vụ riêng biệt. Điều này cho phép các đội dự án lập kế hoạch kiểm thử ngay trong lúc đang thực hiện cài đặt mã (coding), thay vì phải đợi cài đặt mã xong mới bắt đầu như trong mô hình thác nước thuần túy.
- **Các hoạt động chính trong mô hình:** Bao gồm phân tích yêu cầu, thiết kế mức cao, thiết kế chi tiết, mã hóa, kiểm thử đơn vị, lập kế hoạch tích hợp, tích hợp và kiểm thử tích hợp, lập kế hoạch kiểm thử hệ thống, kiểm thử hệ thống, lập tài liệu, kiểm thử chấp nhận, cài đặt và hỗ trợ bảo hành.

Tiến trình cơ sở này có tính linh hoạt cao, có thể được điều chỉnh để phù hợp với các dự án phát triển theo vòng lặp, tạo bản mẫu hoặc các tình huống dự án cụ thể khác

Câu 10; Tại sao lại phải điều chỉnh tiến trình công nghệ phần mềm? Hãy trình bày các đặc trưng của dự án làm cơ sở cho điều chỉnh tóm tắt tiến trình.

1. Tại sao phải điều chỉnh tiến trình công nghệ phần mềm?

Việc điều chỉnh tiến trình là cần thiết vì những lý do sau:

- **Tính không vạn năng của tiến trình:** Không có bất kỳ tiến trình nào được định nghĩa trước (dù là tiến trình chuẩn hay tiến trình từ một dự án tương tự) có thể áp dụng tốt cho **tất cả các dự án**.
- **Phù hợp với đặc thù dự án:** Mỗi dự án có những đặc trưng riêng về kỹ thuật và yêu cầu. Do đó, tiến trình sẵn có cần được điều chỉnh để tạo ra một tiến trình mới **phù hợp, có lợi và hiệu quả hơn** cho dự án hiện tại.
- **Tối ưu hóa nguồn lực:** Việc điều chỉnh giúp sửa đổi hoặc thêm/xóa các hoạt động của tiến trình sao cho tương xứng với nỗ lực bỏ ra và giá trị thu về (ví dụ: điều chỉnh hoạt động "Xem xét lại mã").

2. Các đặc trưng của dự án làm cơ sở cho điều chỉnh ở cấp độ tóm tắt

Trong điều chỉnh ở cấp độ tóm tắt, người quản lý dự án sẽ dựa vào các đặc trưng sau của dự án để áp dụng các hướng dẫn điều chỉnh tiến trình chuẩn,:

1. **Mức độ kinh nghiệm và kỹ năng:** Xét trên cả đội dự án và người quản lý dự án.
 - *Kinh nghiệm cao:* Phần lớn thành viên có trên **2 năm kinh nghiệm** làm việc với công nghệ đang triển khai trong dự án.
2. **Kích thước đội dự án lúc cao điểm:** Quy mô nhân sự lớn nhất tham gia vào dự án tại một thời điểm.
3. **Mức độ rõ ràng của các yêu cầu:** Sự thấu hiểu và tính ổn định của các yêu cầu từ phía khách hàng.
4. **Thời gian thực hiện dự án:** Thời hạn bàn giao sản phẩm.
 - *Thời gian đặc biệt ngắn:* Nếu dự án phải được giao trong vòng **3 tháng**.
5. **Mức độ quan trọng của ứng dụng:** Tầm ảnh hưởng của ứng dụng đối với nghiệp vụ của khách hàng hoặc công ty phát triển.
 - *Quan trọng cao:* Nếu tác động của ứng dụng lên hoạt động kinh doanh là lớn.

Các đặc trưng này giúp thiết lập bối cảnh để định nghĩa một tiến trình mới thích hợp nhất cho dự án, giúp quản lý rủi ro và đảm bảo tiến độ cũng như chất lượng sản phẩm.

Câu 11: Tại sao lại phải quản lý thay đổi yêu cầu? Anh (Chị) hãy trình bày quy trình quản lý thay đổi yêu cầu ở Infosys.

1. Tại sao phải quản lý thay đổi yêu cầu?

Việc quản lý thay đổi yêu cầu là bắt buộc vì những lý do sau:

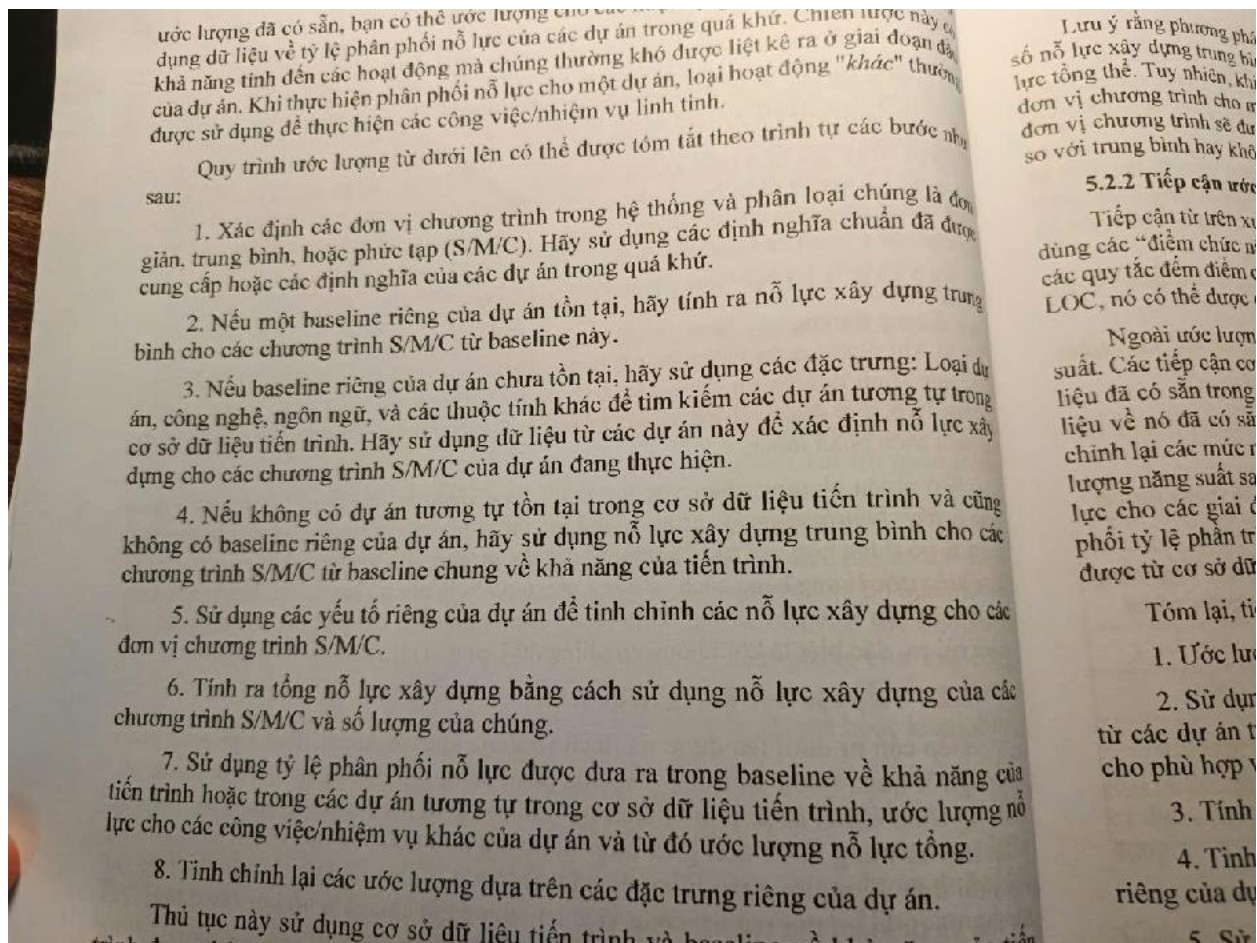
- **Tác động tiêu cực:** Những thay đổi yêu cầu có thể đến bất cứ lúc nào và nếu không được kiểm soát, chúng sẽ gây **ảnh hưởng xấu đến chi phí, tiến độ và chất lượng** của dự án,.
- **Giảm thiểu rủi ro:** Quản lý thay đổi giúp **giảm thiểu tác động** của sự thay đổi lên tiến độ dự án, tránh tình trạng dự án bị chậm trễ hoặc quá tải nguồn lực như trường hợp dự án không được kiểm soát,.
- **Thỏa thuận minh bạch:** Đảm bảo có sự thỏa thuận rõ ràng về thanh toán và nguồn lực bổ sung giữa nhà phát triển và khách hàng khi có thay đổi phát sinh.

2. Quy trình quản lý thay đổi yêu cầu (tại Infosys)

Tiến trình xử lý các thay đổi yêu cầu bao gồm **8 bước cơ bản** sau,:

1. **Ghi nhận các thay đổi:** Tiếp nhận và ghi chép lại nội dung yêu cầu thay đổi.
2. **Phân tích tác động lên các sản phẩm công việc:** Xác định những phần nào của hệ thống (tài liệu, mã nguồn, thiết kế) cần phải sửa đổi.
3. **Ước lượng nỗ lực:** Tính toán khối lượng công việc và nguồn lực cần thiết để thực hiện thay đổi.
4. **Ước lượng lại thời gian giao hàng:** Điều chỉnh lại lịch biểu và thời hạn bàn giao dự án dựa trên khối lượng công việc phát sinh.
5. **Phân tích tác động lên chi phí tích lũy:** Đánh giá sự thay đổi về mặt ngân sách và tài chính.
6. **Xem xét với quản lý cấp cao:** Thảo luận về các tác động nếu các ngưỡng về nỗ lực hoặc chi phí bị vượt quá giới hạn cho phép.
7. **Khách hàng ký duyệt:** Yêu cầu khách hàng xác nhận và ký tên đồng ý với các thay đổi về chi phí và tiến độ đã phân tích.
8. **Thực hiện thay đổi:** Tiến hành làm lại hoặc cập nhật các sản phẩm công việc theo yêu cầu mới đã được phê duyệt.

Câu 12: Anh (Chị) hãy trình bày các bước trong tiến trình ước lượng nỗ lực theo tiếp cận từ dưới lên.



Câu 13: Anh (Chị) hãy trình bày các bước trong tiến trình ước lượng nỗ lực theo tiếp cận từ trên xuống.

chỉnh lại các mức này, nếu cần thiết, để phù hợp với dự án đang được ước lượng. Ước lượng năng suất sau đó được sử dụng để tính ra nỗ lực tổng thể. Từ nỗ lực tổng thể, nỗ lực cho các giai đoạn khác nhau được ước lượng bằng cách sử dụng các bảng phân phối tỷ lệ phần trăm. Như trong tiếp cận từ dưới lên, những phân phối này có thể thu được từ cơ sở dữ liệu tiến trình hoặc baseline về khả năng tiến trình.

Tóm lại, tiếp cận ước lượng từ trên xuống bao gồm các bước sau đây:

1. Ước lượng kích thước tổng cộng của phần mềm bằng các “điểm chức năng”.
2. Sử dụng các dữ liệu về năng suất từ baseline về khả năng của tiến trình, hoặc từ các dự án tương tự trong cơ sở dữ liệu tiến trình, sửa chữa lại các mức năng suất cho phù hợp với dự án đang được ước lượng.
3. Tính nỗ lực tổng thể của dự án từ năng suất và kích thước.
4. Tinh chỉnh lại ước lượng, có xem xét đến mức độ ảnh hưởng của các đặc trưng riêng của dự án.
5. Sử dụng dữ liệu về phân phối nỗ lực từ baseline về khả năng của tiến trình hoặc từ dự án tương tự để ước lượng nỗ lực cho các giai đoạn khác nhau của dự án.

Cũng giống như tiếp cận từ dưới lên, tiếp cận từ trên xuống cũng cho phép tinh chỉnh lại kết quả ước lượng bằng cách sử dụng các yếu tố riêng của dự án. Sự cho phép này (không thực sự định nghĩa các yếu tố nào) cho thấy mỗi dự án là duy nhất và có vài yếu tố đặc trưng mà các dự án khác không có. Không thể liệt kê ra những yếu tố đặc trưng này hoặc lập mô hình một cách hình thức về sự ảnh hưởng của chúng lên năng suất. Do đó, người quản lý dự án tự quyết định các yếu tố nào cần được xem xét và chúng sẽ ảnh hưởng lên dự án như thế nào.

5.2.3 Tiếp cận điểm ca sử dụng - UC

Câu 14: Lịch biểu của dự án phần mềm phụ thuộc vào những yếu tố nào? Kích cỡ đội dự án thay đổi như thế nào trong tiến trình thực hiện dự án? Tại sao lại như vậy?

1. Các yếu tố ảnh hưởng đến lịch biểu dự án

Lịch biểu của một dự án phần mềm phụ thuộc vào các yếu tố chính sau:

- **Mức độ nhân sự:** Người quản lý có thể thay đổi thời gian biểu bằng cách kiểm soát số lượng nhân viên tham gia.
- **Yêu cầu về tiến độ:** Các mốc thời gian thường bị ràng buộc bởi kế hoạch kinh doanh hoặc yêu cầu cụ thể từ khách hàng.
- **Sự sẵn có của nguồn lực:** Nếu số lượng nhân sự cần thiết chưa có sẵn, thời gian hoàn thành dự án buộc phải điều chỉnh.
- **Yếu tố phụ thuộc bên ngoài:** Ví dụ như việc chờ đợi hoàn thành một dự án khác hoặc chờ đợi một phần mềm/linh kiện từ bên thứ ba.
- **Tổng nỗ lực ước tính:** Thời gian hoàn thành thường được tính toán dựa trên căn bậc hai của tổng nỗ lực (quy tắc kiểm tra căn bậc hai).

2. Sự thay đổi kích cỡ đội dự án và lý do

Sự thay đổi: Kích cỡ đội dự án không cố định mà thay đổi theo **đường cong Rayleigh**.

- **Giai đoạn đầu và cuối:** Có ít người làm việc hơn.
- **Giai đoạn giữa:** Số lượng nhân sự đạt mức tối đa (đỉnh của đường cong).

Lý do:

- **Giai đoạn đầu (Phân tích yêu cầu và Thiết kế):** Chỉ cần một vài người có kỹ năng chuyên môn cao để xác định khung dự án và yêu cầu, nên không cần đông nhân sự.
- **Giai đoạn giữa (Cài đặt mã và Kiểm thử đơn vị):** Đây là lúc khối lượng công việc thực thi lớn nhất, cần nguồn nhân lực tối đa để viết mã và kiểm tra các thành phần nhỏ.
- **Giai đoạn cuối (Tích hợp và Kiểm thử hệ thống):** Khi sản phẩm dần hoàn thiện, khối lượng công việc giảm xuống và chỉ cần một đội ngũ ít hơn để thực hiện các công việc kiểm thử cuối cùng và bàn giao.

Câu 15: Khái niệm về lỗi phần mềm? Tại sao phát triển phần mềm lại hay mắc lỗi? Các hoạt động để phát hiện lỗi và loại bỏ lỗi?

Câu 15: Khái niệm về lỗi phần mềm, nguyên nhân và các hoạt động phát hiện/loại bỏ lỗi

- **Khái niệm về lỗi phần mềm:** Mặc dù không có định nghĩa chính xác áp dụng chung, nhưng lỗi phần mềm thường được hiểu là việc **phần mềm có cư xử theo một cách thức không nhất quán với các yêu cầu hoặc nhu cầu của khách hàng**. Trong quản lý chất lượng, lỗi cũng thường xoay quanh khái niệm "**mật độ lỗi được giao**", tức là số lượng lỗi trên mỗi đơn vị kích thước phần mềm.
- **Tại sao phát triển phần mềm hay mắc lỗi:** Phát triển phần mềm là một **hoạt động theo hướng con người**, do đó nó rất dễ bị lỗi. Trong suốt quá trình chuyển đổi từ yêu cầu của khách hàng thành sản phẩm hoàn chỉnh, lỗi có thể được "**tiêm (inject)**" vào tại tất cả các giai đoạn như: đặc tả yêu cầu, thiết kế mức cao, thiết kế chi tiết và cài đặt mã.
- **Các hoạt động để phát hiện và loại bỏ lỗi:** Việc phát hiện và loại bỏ lỗi được thực hiện thông qua các hoạt động kiểm soát chất lượng chính sau,:
 - **Xem xét lại (Review):** Bao gồm xem xét lại yêu cầu, xem xét lại thiết kế và xem xét lại mã nguồn. Đây là hoạt động hướng con người để xác định lỗi.
 - **Kiểm thử (Testing):** Là tiến trình thực thi phần mềm trên máy để xác định lỗi, bao gồm: **Kiểm thử đơn vị (UT)**, **Kiểm thử tích hợp (IT)**, **Kiểm thử hệ thống (ST)** và **Kiểm thử chấp nhận (AT)**.

Câu 16: Anh (Chị) hãy trình bày vắn tắt một số tiếp cận định lượng để quản lý chất lượng của dự án phần mềm.

Câu 17: Anh (Chị) hãy trình bày các bước được thực hiện cho các hoạt động phòng ngừa lỗi ở mức dự án.

mềm đang được xây dựng. Đây chính là động lực để nâng cao chất lượng bằng cách tăng cường các hoạt động phòng ngừa lỗi.

Các hoạt động phòng ngừa lỗi cũng giúp tăng năng suất. Thật vậy, chu kỳ lỗi cơ bản trong xây dựng phần mềm là các nhà phát triển tạo ra lỗi, rồi sau đó xác định và loại bỏ chúng. Rõ ràng, chu kỳ tìm lỗi và loại bỏ lỗi này là một sự lãng phí nỗ lực. Do đó, nếu lúc đầu không tạo ra lỗi thì sau đó nỗ lực cần thiết để xác định và loại bỏ chúng sẽ không được cần đến. Tức là, nếu tìm vào ít lỗi hơn, ít lỗi hơn sẽ được gỡ bỏ, nỗ lực cần thiết để loại bỏ lỗi sẽ được giảm, do đó làm tăng năng suất.

Phòng ngừa lỗi được thực hiện như thế nào? Có một số nguyên nhân đằng sau những lỗi tìm vào. Vì vậy, phòng ngừa lỗi nói chung đòi hỏi phải thu thập dữ liệu về các lỗi được tìm thấy trong quá khứ, phân tích dữ liệu này để tìm ra nguyên nhân gốc rễ của việc tìm lỗi, và sau đó phát triển các giải pháp loại bỏ các nguyên nhân này.

Để thực hiện phòng ngừa lỗi, người quản lý dự án có thể bắt đầu với một tập hợp các gợi ý (nên được thực hiện theo) ở mức công ty và sau đó xây dựng gợi ý cho từng loại dự án cụ thể bằng cách dựa trên phân tích dữ liệu về lỗi của dự án. Ở Infosys, các bước sau đây được thực hiện cho các hoạt động phòng ngừa lỗi ở mức dự án:

1) Xây dựng một nhóm phòng ngừa lỗi trong phạm vi dự án.

2) Mở một cuộc họp đầu tiên của nhóm và xác định các giải pháp hiện có.

3) Lập kế hoạch phòng ngừa lỗi cho dự án:

+ Thiết lập mục tiêu phòng ngừa lỗi cho dự án.

+ Huấn luyện về phòng ngừa lỗi và phương pháp phân tích nguyên nhân cho nhóm nếu cần thiết.

+ Xác định tần số thực hiện các hoạt động phòng ngừa lỗi.

4) Thực hiện phòng ngừa lỗi:

+ Tại các điểm đã xác định, thu thập dữ liệu về lỗi.

+ Xác định các loại lỗi phổ biến nhất bằng cách làm phân tích Pareto.

+ Thực hiện phân tích nguyên nhân và xếp độ ưu tiên các nguyên nhân gốc.

+ Xác định và phát triển các giải pháp cho các nguyên nhân gốc.

+ Thực hiện các giải pháp.

+ Xem lại các trạng thái và lợi ích của phòng ngừa lỗi tại các cột mốc của dự án.

5) Rút ra bài học kinh nghiệm:

+ Trong báo cáo phân tích số liệu, rút ra bài học và lợi ích đã thu được.

+ Gửi tất cả các kết quả đầu ra của phòng ngừa lỗi để góp phần xây dựng các thành phần của tiến trình.

Các hoạt động "Thực hiện phòng ngừa lỗi" hoặc "Rút ra bài học" được thực hiện trong khi dự án đang được thực hiện hoặc khi nó đã hoàn tất. Ở đây, chúng ta tập trung chủ yếu vào các nhiệm vụ liên quan đến lập kế hoạch (từ bước 1 tới bước 3).

Câu 18: Anh (Chị) hãy trình bày khái niệm rủi ro và quản lý rủi ro. Cho một ví dụ minh họa.

7.1 KHÁI NIỆM RỦI RO VÀ QUẢN LÝ RỦI RO

Quản lý dự án phần mềm là một công việc phức tạp. Các sự kiện bất ngờ xảy ra trong quá trình thực thi dự án có thể tác động xấu đến chi phí, tiến độ và chất lượng của dự án. Quản lý rủi ro là một nỗ lực để hạn chế nguy cơ thất bại của dự án do các sự kiện bất ngờ gây ra. Mục đích của quản lý rủi ro là để giảm thiểu tác động của rủi ro trong các dự án được thực hiện. Thách thức đối với một người quản lý dự án không phải là tránh các rủi ro mà là quản lý chúng một cách hiệu quả thông qua chiến lược giảm thiểu rủi ro. Quản lý rủi ro không đúng cách là một trong những nguyên nhân của nhiều dự án bị thất bại.

Kn (Các rủi ro là những sự kiện có thể xảy ra, và sự xảy ra của nó gây thiệt hại hoặc có tác động tiêu cực lên dự án). Vì một sự kiện xác suất có thể hoặc không thể xảy ra nên chúng ta thường có khuynh hướng lạc quan, không nhìn thấy được các rủi ro hoặc mong muốn rằng chúng sẽ không xảy ra. Ngoài ra, các yếu tố xã hội và tổ chức cũng có thể góp phần làm tăng rủi ro và ngăn cản xác định rõ chúng. Điều này làm cho dự án gặp rắc rối nếu các sự kiện rủi ro xảy ra thực sự, nhất là trong những dự án lớn. (Quản lý rủi ro nhằm mục đích để xác định các rủi ro và sau đó có các hành động để giảm thiểu tác động của chúng đến dự án). Quản lý rủi ro lần đầu tiên được đề cập đến bởi Boehm thông qua một hướng dẫn quản lý rủi ro. Kể từ đó, nhiều cuốn sách đã xem quản lý rủi ro là một trong những nhiệm vụ của quản lý dự án phần mềm. Mục đích

Trước khi thảo luận về các rủi ro trong bối cảnh phần mềm, chúng ta hãy xem xét về khái niệm rủi ro và quản lý rủi ro thông qua một ví dụ thực tế:

Một chương trình máy tính mà mục tiêu quan trọng của nó là cung cấp các dịch vụ máy tính không bị gián đoạn.

Đối với mục tiêu này, một trong những rủi ro rõ ràng là mất nguồn điện cung cấp cho máy tính. Nguồn điện cung cấp cho máy tính là điện lưới, nó có thể mất hoặc không bất kỳ lúc nào vì một lý do kỹ thuật nào đó. Nếu mất, thậm chí chỉ trong một giây, các dịch vụ máy tính sẽ bị gián đoạn. Để đối phó với rủi ro này, một bộ lưu điện

(Uninterruptible Power Supply - UPS) có thể được triển khai để đối phó khi mất điện lưới trong thời gian ngắn. Nếu trường hợp điện lưới có thể bị mất một khoảng thời gian dài, một máy phát điện dự phòng có thể được thiết lập để đối phó với sự cố mất điện dài. Với hệ thống quản lý rủi ro này, nếu điện bị mất, ngay cả trong một thời gian dài, mục tiêu cung cấp dịch vụ máy tính liên tục sẽ không bị ảnh hưởng.

Vấn đề đầu tiên là... đòi hỏi chi phí bổ sung

1. Khái niệm Rủi ro

- **Rủi ro** là những sự kiện **có thể xảy ra hoặc không** (mang tính xác suất) trong quá trình thực hiện dự án.
- Nếu rủi ro thực sự xảy ra, nó sẽ gây thiệt hại hoặc có **tác động tiêu cực** đến chi phí, tiến độ và chất lượng của dự án.

2. Khái niệm Quản lý rủi ro

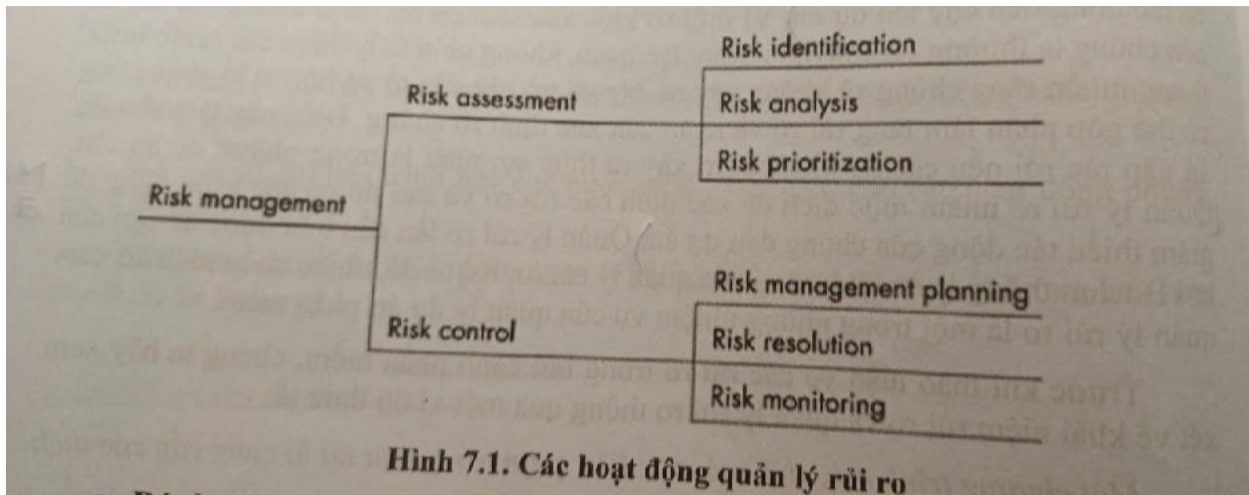
- **Quản lý rủi ro** là toàn bộ nỗ lực nhằm **xác định trước** các rủi ro tiềm ẩn và đưa ra các hành động (chiến lược giảm thiểu) để hạn chế nguy cơ thất bại của dự án.

- **Thách thức:** Người quản lý không thể tránh hoàn toàn rủi ro mà phải chủ động kiểm soát và giảm thiểu tác động của chúng một cách hiệu quả.

3. Ví dụ minh họa thực tế

- **Mục tiêu dự án:** Duy trì một hệ thống máy tính hoạt động liên tục, không bị gián đoạn.
- **Rủi ro xác định:** Hệ thống có thể bị mất nguồn điện lưới bất cứ lúc nào.
- Giải pháp quản lý rủi ro:
 - *Mất điện ngắn hạn:* Chuẩn bị sẵn bộ lưu điện (**UPS**) để đối phó tạm thời.
 - *Mất điện dài hạn:* Lắp đặt thêm **máy phát điện dự phòng**.
- **Kết quả:** Nhờ có hệ thống quản lý rủi ro này, nếu sự cố mất điện thực sự xảy ra (dù ngắn hay dài), mục tiêu hoạt động liên tục của máy tính vẫn được đảm bảo và không bị ảnh hưởng.

Câu 19: Anh (Chị) hãy trình bày các hoạt động trong quản lý rủi ro (theo hình vẽ).



1. Đánh giá rủi ro (Risk Assessment)

Là nhóm hoạt động nhằm nhận diện và định lượng rủi ro, bao gồm 3 bước:

- **Xác định rủi ro (Risk identification):** Tập trung tìm kiếm và liệt kê ra tất cả các tình huống xấu có thể xảy ra làm dự án gặp rắc rối.
- **Phân tích rủi ro (Risk analysis):** Xem xét kỹ lưỡng mọi khía cạnh, tác động và khả năng xảy ra của từng rủi ro.
- **Xếp độ ưu tiên rủi ro (Risk prioritization):** Đánh giá xem rủi ro nào nguy hiểm nhất để ưu tiên tập trung xử lý trước.

2.

Kiểm soát rủi ro (Risk Control)

Là nhóm hoạt động nhằm lên phương án và hành động thực tế để đối phó với rủi ro, bao gồm 3 bước:

- **Lập kế hoạch quản lý rủi ro (Risk management planning):** Xây dựng các kịch bản, phương án dự phòng và chiến lược để giảm thiểu hậu quả nếu rủi ro xảy ra.
- **Giải quyết rủi ro (Risk resolution):** Thực hiện các hành động hoặc giải pháp cụ thể để triệt tiêu nguyên nhân hoặc né tránh rủi ro.
- **Giám sát rủi ro (Risk monitoring):** Theo dõi liên tục trạng thái của dự án để phát hiện kịp thời khi rủi ro xuất hiện và đánh giá xem các giải pháp đối phó có hiệu quả hay không.

Câu 20: Đánh giá rủi ro bao gồm những hoạt động nào? Trình bày về các phương pháp cơ bản để xác định rủi ro.

Quản lý rủi ro có thể được tích hợp vào tiến trình phát triển phần mềm, như trong phát triển phần mềm theo mô hình xoắn ốc. Nếu người quản lý xem quản lý rủi ro như là một tiến trình riêng biệt, họ phải hiểu mối quan hệ của nó với tiến trình thực hiện dự án. Như thể hiện trong Hình 7.2, đánh giá rủi ro và giám sát thông tin từ quá trình thực hiện dự án, cùng với các yếu tố khác, để xác định các rủi ro cần được quản lý. Mặc khác, các hoạt động quản lý rủi ro sẽ ảnh hưởng đến quá trình giảm thiểu hậu quả của rủi ro.

7.2 ĐÁNH GIÁ RỦI RO

Trong một dự án, đánh giá rủi ro bao gồm các hoạt động: Xác định rủi ro, sau đó phân tích và xếp độ ưu tiên rủi ro. Hoạt động xác định rủi ro tập trung vào liệt kê các rủi ro có thể xảy ra cho dự án. Hoạt động này là để cố gắng hình dung ra tất cả các tình huống có thể làm cho dự án gặp rắc rối. Các hoạt động phân tích và xếp độ ưu tiên rủi ro xem xét tất cả các khía cạnh của tất cả các rủi ro và sau đó xếp độ ưu tiên (cho các mục đích của quản lý rủi ro). Mặc dù là hai nhóm hoạt động riêng biệt, nhưng chúng thường được thực hiện cùng một lúc. Tức là, người quản lý dự án có thể xác định và phân tích rủi ro chung với nhau.

7.2.1 Xác định rủi ro

Đối với một dự án, bất cứ sự kiện nào có thể xảy ra và sẽ gây ảnh hưởng cho sự thành công của dự án đều tạo nên rủi ro. Do đó, xác định rủi ro là một công việc hình dung những gì có thể làm cho dự án gặp rắc rối. Phương pháp giúp xác định rủi ro bao gồm: Xem xét danh sách kiểm tra của các rủi ro có thể xảy ra, tiến hành các cuộc khảo sát và các cuộc họp, xem xét lại các kế hoạch, tiến trình và các sản phẩm công việc. Trong đó, danh sách kiểm tra về các rủi ro thường xuyên xảy ra có lẽ là công cụ phổ biến nhất để xác định rủi ro.

Ở các công ty phần mềm, những rủi ro thường gặp đối với các dự án đã được biên soạn ra từ các cuộc khảo sát ở các dự án trước đó. Danh sách này sẽ là điểm khởi đầu để xác định các rủi ro cho dự án hiện tại. Thông thường thì các rủi ro trong dự án hiện tại sẽ xuất hiện trong danh sách.

Người quản lý dự án cũng có thể sử dụng cơ sở dữ liệu tiến trình để có được thông tin về rủi ro và quản lý rủi ro từ các dự án tương tự. Đánh giá và nghĩ về các rủi ro gặp phải từ các dự án trước đây cũng giúp xác định các rủi ro có thể xảy ra cho dự án này, nhưng không xuất hiện trên danh sách kiểm tra.

Người quản lý dự án cũng có thể sử dụng kinh nghiệm của họ để đánh giá tình hình và xác định các rủi ro tiềm năng cho dự án của họ. Một lựa chọn khác là sử dụng việc xem xét lại kế hoạch quản lý dự án và các cuộc họp thảo luận để gợi ra những quan điểm về rủi ro từ những thành viên khác trong đội dự án.

7.2.2 Phân tích tác động và xếp độ ưu tiên rủi ro

Những rủi ro đã được xác định cho một dự án chỉ ra các sự kiện có thể cản trở việc đạt được các mục tiêu của dự án. Tuy nhiên, chúng có thể khác nhau. Tuy nhiên, chúng có thể khác nhau. Tuy nhiên, chúng có thể khác nhau.

1. Các hoạt động của Đánh giá rủi ro (Risk Assessment)

Đánh giá rủi ro là việc cố gắng hình dung ra mọi tình huống có thể làm dự án gặp rắc rối, bao gồm 3 hoạt động thường được thực hiện cùng lúc:

- **Xác định rủi ro:** Liệt kê tất cả các rủi ro có thể xảy ra cho dự án.
- **Phân tích rủi ro:** Xem xét kỹ các khía cạnh của từng rủi ro.

- **Xếp độ ưu tiên rủi ro:** Phân loại rủi ro nào cần tập trung quản lý và xử lý trước.

2. Các phương pháp cơ bản để xác định rủi ro

Để tìm ra các rủi ro tiềm ẩn của dự án, người quản lý có thể áp dụng các phương pháp sau:

- **Sử dụng danh sách kiểm tra (Checklist) của công ty:** Đây là công cụ **phổ biến nhất**. Xem lại danh sách các rủi ro thường gặp đã được tổng hợp từ khảo sát của các dự án quá khứ để làm điểm khởi đầu cho dự án hiện tại.
- **Tra cứu cơ sở dữ liệu tiến trình:** Tìm kiếm thông tin về rủi ro và cách quản lý rủi ro từ các dự án tương tự đã thực hiện trước đây.
- **Dựa vào kinh nghiệm cá nhân:** Người quản lý sử dụng trải nghiệm thực tế của mình để đánh giá tình hình và tự phán đoán các rủi ro mới (không có trong checklist).
- **Họp nhóm và xem xét lại kế hoạch:** Tổ chức các cuộc họp thảo luận, rà soát lại kế hoạch quản lý dự án để thu thập thêm góc nhìn và ý kiến về rủi ro từ các thành viên trong đội.

Câu 21: Anh (Chị) hãy trình bày khái niệm số đo của tiến trình phần mềm hoặc các sản phẩm phần mềm. Những số đo cơ bản nào được sử dụng trong mọi dự án phần mềm?

1. Khái niệm số đo (Metrics)

Số đo của tiến trình hoặc sản phẩm phần mềm là các **tham số được định lượng cụ thể** (tức là có thể đo đếm được bằng con số) để đánh giá hiệu suất, năng lực và chất lượng của dự án.

Thay vì quản lý bằng cảm tính, việc thu thập các số đo giúp nhà quản lý lưu lại dữ liệu thực tế vào cơ sở dữ liệu tiến trình, từ đó làm căn cứ chính xác để lập kế hoạch, ước lượng và giám sát cho các dự án sau này.

2. Những số đo cơ bản dùng trong mọi dự án

Bất kỳ dự án phần mềm nào cũng cần thu thập và lưu giữ 4 nhóm số đo cốt lõi sau:

- **Kích thước (Size):** Đo độ lớn của sản phẩm phần mềm làm ra, thường tính bằng số dòng mã lệnh (LOC) hoặc số điểm chức năng (Function Point - FP).
- **Nỗ lực (Effort):** Đo sức lao động của con người bỏ ra để hoàn thành công việc, thường được tính bằng Giờ công hoặc Tháng công phân bổ cho từng giai đoạn.
- **Thời gian biểu / Lịch trình (Schedule):** Đo lường thời gian thực hiện dự án (ngày bắt đầu, ngày kết thúc) và tỷ lệ phần mềm được giao đúng thời gian thỏa thuận.
- **Lỗi (Defects):** Số đo phản ánh chất lượng sản phẩm, bao gồm số lượng lỗi phát hiện được, tỷ lệ tìm lỗi vào hệ thống và hiệu quả của việc gỡ bỏ lỗi

Câu 22: Trong một dự án, thường việc phát hiện và loại bỏ lỗi được tiến hành theo tiến trình nào? Giải thích các hoạt động của tiến trình đó.

1. Tiến trình phát hiện và loại bỏ lỗi

Trong một dự án phần mềm, việc phát hiện và loại bỏ lỗi thường được tiến hành thông qua quy trình kiểm soát chất lượng với hai nhóm hoạt động chủ đạo là: **Xem xét lại (Review)** và **Kiểm thử (Testing)**.

2. Giải thích các hoạt động trong tiến trình

A. Hoạt động Xem xét lại (Review)

- **Bản chất:** Đây là phương pháp kiểm tra "hướng con người". Thay vì chạy máy móc, con người sẽ dùng kinh nghiệm để trực tiếp đọc, soi chiếu và tìm ra lỗi từ sớm.
- **Các bước cụ thể:** Hoạt động này được áp dụng xuyên suốt từ đầu dự án, bao gồm: xem xét lại tài liệu yêu cầu, xem xét lại các bản thiết kế và xem xét lại mã nguồn (code) của nhau.

B. Hoạt động Kiểm thử (Testing)

- **Bản chất:** Khác với xem xét lại, kiểm thử là tiến trình trực tiếp "thực thi phần mềm trên máy tính" để xác định và tìm ra các lỗi phát sinh khi hệ thống hoạt động thực tế.
- **Các cấp độ cụ thể:** Quá trình này được thực hiện theo thứ tự từ quy mô nhỏ đến lớn, bao gồm:

1. **Kiểm thử đơn vị (Unit Test):** Chạy thử và bắt lỗi trên từng đoạn code hoặc

mô-đun nhỏ nhất.

2. **Kiểm thử tích hợp (Integration Test):** Kiểm tra xem các mô-đun khi ghép nối với nhau có hoạt động trơn tru không.
3. **Kiểm thử hệ thống (System Test):** Chạy thử toàn bộ phần mềm hoàn chỉnh để đánh giá tổng thể.
4. **Kiểm thử chấp nhận (Acceptance Test):** Cho khách hàng dùng thử sản phẩm để họ xác nhận và nghiệm thu phần mềm.

Câu 23: Anh (Chị) hãy trình bày phương pháp kiểm soát tiến độ và nỗ lực trong một dự án phần mềm dùng SPC.

Phương pháp kiểm soát tiến độ và nỗ lực dùng SPC (Kiểm soát tiến trình bằng thống kê)

Về bản chất, áp dụng SPC vào quản lý tiến độ và nỗ lực là việc **thiết lập các "giới hạn kiểm soát" (Control Limits)** để theo dõi xem dự án có đang đi đúng hướng hay không. Phương pháp này gồm 3 bước cốt lõi:

1. Thiết lập giới hạn kiểm soát cho phép

- Dựa trên dữ liệu quá khứ (baseline) của công ty, người quản lý định ra một khoảng sai số (chênh lệch) có thể chấp nhận được giữa thực tế và kế hoạch.
- *Ví dụ:* Ở Infosys, giới hạn chênh lệch nỗ lực thường được đặt là $\pm 20\%$, và giới hạn chênh lệch lịch biểu (tiến độ) là $\pm 10\%$.

2. Đo lường và so sánh tại các cột mốc (Milestones)

- Tại mỗi cột mốc quan trọng của dự án, người quản lý sẽ đo lường nỗ lực thực tế (số giờ công đã tiêu) và tiến độ thực tế (thời gian đã trôi qua).
- Sau đó, lấy số liệu thực tế này so sánh với các con số đã ước lượng/lập kế hoạch ban đầu để tính ra độ chênh lệch.

3. Phân tích và Hành động khắc phục

Dựa vào mức độ chênh lệch so với giới hạn kiểm soát, người quản lý sẽ quyết định hành động:

- **Nằm trong giới hạn:** Dự án đang hoạt động ổn định, không cần can thiệp.
- **Vượt ra ngoài giới hạn (cao hơn hoặc thấp hơn mức cho phép):** Báo động đỏ! Người quản lý bắt buộc phải tìm hiểu nguyên nhân và đưa ra hành động khắc phục ngay.
 - *Nếu nỗ lực thực tế quá cao (tốn nhiều thời gian hơn):* Có thể do nhân viên thiếu kinh nghiệm, phải sửa lỗi nhiều. Cần đào tạo thêm, bổ sung người hoặc đàm phán lùi ngày giao hàng.
 - *Nếu nỗ lực thực tế quá thấp (xong quá nhanh):* Có thể do ước lượng sai, hoặc làm ẩu, bỏ sót việc. Cần rà soát lại các công việc đã làm để đảm bảo chất lượng.

Câu 24: Anh (Chị) hãy cho biết các thành phần của một đội dự án và vai trò của mỗi thành phần này. Một số yếu tố mà một người quản lý dự án cần quan tâm khi xây dựng và phân công các thành viên trong đội là gì?

được khi các thành viên trong đội phải đoàn kết và duy trì động lực, toàn đội phải làm việc một cách trơn tru và hiệu quả. Quản lý đội dự án vượt ra ngoài phạm vi kỹ thuật, nó liên quan đến các vấn đề con người. Nó là một khía cạnh quan trọng của dự án mà nếu bỏ qua nó, có thể gây ra những rắc rối cho dự án.

9.1.1 Cấu trúc đội dự án

Một đội dự án được lãnh đạo bởi một người quản lý dự án, người sẽ báo cáo cho người quản lý kinh doanh hoặc người quản lý quan hệ khách hàng (hoặc cả hai) của công ty. Ngoài ra, một đội dự án điển hình còn gồm có các nhà phát triển, người kiểm soát cấu hình, và người quản trị cơ sở dữ liệu. Tất cả các thành viên này sẽ báo cáo cho người quản lý dự án. Một dự án lớn cũng có thể có những người lãnh đạo mô đun, họ báo cáo cho người quản lý dự án và có một số nhà phát triển dưới quyền. Ngoài ra, một nhóm phòng ngừa lỗi sẽ được thành lập từ các thành viên hiện có của của đội dự án, nhóm này chịu trách nhiệm thực hiện các nhiệm vụ liên quan đến phòng ngừa lỗi.

Như đã thảo luận trước đây, một thành viên của nhóm quy trình công nghệ phần mềm được xem như là cố vấn chất lượng phần mềm cũng có liên quan đến từng dự án. Người này tương tác rất nhiều với người quản lý dự án và với người kiểm soát cấu hình nhưng không có trách nhiệm phải báo cáo với người quản lý dự án. Thay vào đó, anh ta có một kênh báo cáo độc lập với công ty.

Việc xây dựng và phân công vai trò của từng thành viên trong đội là một công việc phức tạp. Mục tiêu của người quản lý dự án là để có được một đội cân bằng về phát triển sự nghiệp và kỹ năng của mỗi thành viên trong đội. Do đó, trong việc xác định cấu trúc đội, người quản lý dự án phải xem xét nhu cầu phát triển cá nhân của từng thành viên trong đội cũng như nhu cầu của dự án. Sau đây là một số yếu tố mà một người quản lý dự án cần quan tâm khi xây dựng và phân công các thành viên trong đội:

- + Kỹ năng, kiến thức nền tảng, và kinh nghiệm của các thành viên.
- + Nguyên vọng cá nhân và con đường sự nghiệp của các thành viên.
- + Các nhu cầu phát triển con người.

Cấu trúc đội ban đầu được ghi chép vào trong kế hoạch quản lý dự án cùng với vai trò và trách nhiệm của mỗi thành viên trong đội. Một thành viên trong đội có thể có nhiều trách nhiệm. Trong suốt quá trình của vòng đời dự án, đặc biệt là đối với một dự án dài hạn, cấu trúc đội có thể bị thay đổi dựa trên hiệu suất của các thành viên khi đảm nhiệm các vai trò khác nhau, nguyên vọng cá nhân, động lực,...

9.1.2 Giao tiếp trong đội dự án

Một đội dự án sẽ làm việc cùng nhau trong vài tháng, thậm chí hàng năm để cùng hướng tới một mục tiêu chung cho sự thành công của dự án. Vì vậy, phải có giao tiếp

Ý 1: Các thành phần của đội dự án và vai trò

Một đội dự án chuẩn bao gồm 7 thành phần sau:

1. **Người quản lý dự án (PM):** "Chỉ huy trưởng" của dự án, chịu trách nhiệm chung và báo cáo cho quản lý cấp cao hoặc khách hàng.
2. **Nhà phát triển (Developer):** Trực tiếp thi công dự án (lập trình, kiểm thử,...), báo cáo cho PM.
3. **Người lãnh đạo mô-đun (Module Lead):** (Chỉ có ở dự án lớn) Quản lý một nhóm nhà phát triển làm một phân hệ riêng, báo cáo cho PM.
4. **Người kiểm soát cấu hình:** Quản lý kho lưu trữ, tài liệu, mã nguồn và các phiên bản phần mềm, báo cáo cho PM.
5. **Người quản trị cơ sở dữ liệu (DBA):** Chịu trách nhiệm quản trị hệ thống dữ liệu, báo cáo cho PM.
6. **Nhóm phòng ngừa lỗi:** Trích ra từ các thành viên hiện có, chuyên phân tích và tìm cách ngăn chặn lỗi.
7. **Cố vấn chất lượng phần mềm:** Hỗ trợ PM kiểm soát chất lượng, nhưng hoạt động khách quan, có quyền **báo cáo độc lập thẳng lên công ty** (không báo cáo cho PM).

Ý 2: Các yếu tố PM cần quan tâm khi phân công công việc

Việc phân công không chỉ để hoàn thành dự án, mà còn để giữ lửa và đào tạo con người.

PM cần xét 3 yếu tố:

1. **Năng lực:** Kỹ năng, kiến thức nền tảng và kinh nghiệm thực tế của nhân viên.
2. **Định hướng:** Nguyên vọng cá nhân và con đường sự nghiệp mà nhân viên muốn hướng tới.

3. **Phát triển:** Nhu cầu được học hỏi, nâng cấp bản thân của nhân viên qua dự án này.

Câu 25: Anh (Chị) hãy trình bày về các chủ đề giao tiếp trong đội dự án.

...an đại nạn, cấu trúc đội có thể bị thay đổi dựa trên hiệu suất của các thành viên khi
đảm nhiệm các vai trò khác nhau, nguyện vọng cá nhân, động lực,...

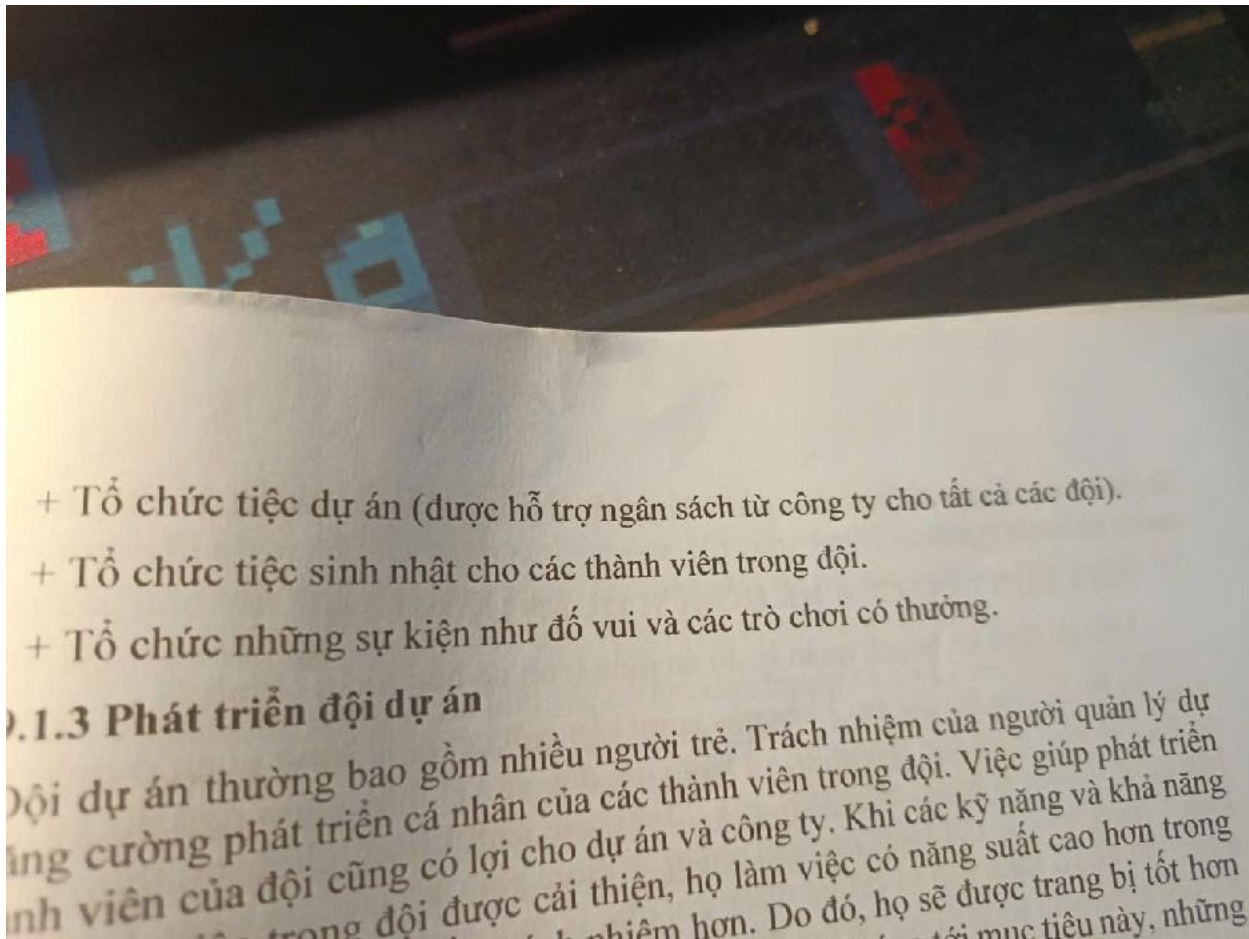
9.1.2 Giao tiếp trong đội dự án

Một đội dự án sẽ làm việc cùng nhau trong vài tháng, thậm chí hàng năm để cùng
hướng tới một mục tiêu chung cho sự thành công của dự án. Vì vậy, phải có giao tiếp
tốt bên trong đội dự án. Các giao tiếp trong đội có thể được phân chia thành hai loại:
Giao tiếp liên quan đến công việc dự án và giao tiếp để giảm căng thẳng. Người quản
lý dự án cần hướng đến cả hai loại giao tiếp này.

Một cách để giúp cho các thành viên trong đội luôn biết được về tiến độ của dự
án và các vấn đề phát sinh trong dự án là cho phép họ truy cập được vào các báo cáo
trạng thái của dự án, ý kiến của khách hàng, và ý kiến của người quản lý kinh doanh.
Ngoài những báo cáo hình thức, tùy thuộc vào kích thước đội dự án và thời gian hoàn
thành dự án, người quản lý dự án có thể sử dụng các phương pháp sau đây để tăng
cường giao tiếp trong đội dự án:

- + Bảng tin trình bày ngắn gọn tin tức cho dự án để đăng các thông báo, chú ý,
báo cáo trạng thái,...
- + Danh sách những địa chỉ nhận thông tin thường xuyên của dự án.
- + Web site của dự án để công bố các tài liệu, các trang chủ của mỗi thành viên
trong đội, các bài viết về kỹ thuật và các ghi chú có liên quan, và tài liệu huấn luyện
dành cho tự học.
- + Các cuộc họp nhanh để giải quyết các vấn đề phát sinh.
- + Các thực hành tốt nhất và các trình bày của các thành viên trong đội về công
việc của họ.

Hơn nữa, bởi vì thời hạn để hoàn thành các công việc thường ngắn, do đó tất cả
mọi người thường bị áp lực về thời gian, căng thẳng có xu hướng gia tăng. Vì vậy,
giao tiếp để giảm căng thẳng là cực kỳ quan trọng để đảm bảo mọi người có động lực
tiếp tục làm việc. Người quản lý dự án cần lập kế hoạch cho các sự kiện để tạo ra giao
tiếp vui vẻ cho đội. Dưới đây là ví dụ về các phương pháp được sử dụng:



1. Giao tiếp liên quan đến công việc

- **Mục đích:** Giúp các thành viên luôn nắm bắt được tiến độ dự án, các vấn đề phát sinh và ý kiến phản hồi từ khách hàng hoặc quản lý.
- **Các phương pháp thực hiện:**
 - Dùng **bảng tin** để đăng thông báo, báo cáo trạng thái.
 - Tạo **danh sách email** để nhận/gửi thông tin thường xuyên.
 - Lập **website dự án** để lưu trữ tài liệu, bài viết kỹ thuật.
 - Tổ chức các **cuộc họp nhanh** để giải quyết vấn đề nóng.
 - Tổ chức các buổi **thuyết trình chia sẻ kinh nghiệm** (thực hành tốt nhất).

2. Giao tiếp để giảm căng thẳng

- **Mục đích:** Do áp lực thời gian (deadline) ngắn, các thành viên rất dễ bị căng thẳng. Giao tiếp vui vẻ giúp giải tỏa áp lực và duy trì động lực làm việc.
- **Các phương pháp thực hiện:**

- Tổ chức liên hoan, **tiệc dự án** (thường được công ty hỗ trợ ngân sách).
- Tổ chức **tiệc sinh nhật** cho các thành viên trong đội.
- Tổ chức các sự kiện giải trí như **đổ vui, trò chơi có thưởng**.

Câu 26: Anh (Chị) hãy trình bày cấu trúc điển hình của bản kế hoạch quản lý dự án và nội dung cơ bản của mỗi phần.

lên hai phía nhanh chóng giải quyết các phát sinh và khi cần thiết có thể thực hiện chúng nhiều hơn nữa.

9.3 CẤU TRÚC CỦA KẾ HOẠCH QUẢN LÝ DỰ ÁN

Một tài liệu kế hoạch quản lý dự án điển hình có bốn phần chính:

1) *Phần tóm tắt dự án*: Phần này cung cấp một cái nhìn tổng quan ở mức cao về dự án. Nó bao gồm các thông tin về ngày bắt đầu và kết thúc, lãnh đạo dự án, địa chỉ liên hệ của khách hàng, mục tiêu của dự án, các cam kết chính được thực hiện cho khách hàng tại các cột mốc, các sản phẩm sẽ được giao, và các giả định. Các giả định được liệt kê rõ ràng bởi vì chúng được coi như là một nguồn của các rủi ro. Thông tin chi tiết về thanh toán cũng có thể được mô tả (nhờ vậy mà người quản lý kinh doanh có thể theo dõi chúng). Các mục tiêu của dự án, từ quan điểm của khách hàng đến quan điểm của nhà phát triển (được đề cập đến để làm cho tất cả mọi người thấy rõ là tại sao dự án sẽ được thực hiện).

2) *Phần lập kế hoạch dự án*: Phần này lập kế hoạch cho tất cả các hoạt động của dự án. Nó bao gồm tiến trình phát triển đang được sử dụng, các ghi chú về điều chỉnh, tiến trình quản lý thay đổi yêu cầu, các kế hoạch để theo dõi dấu vết nguồn gốc yêu cầu, ước lượng nỗ lực và thời gian hoàn thành, và yêu cầu nhân sự theo kỹ năng, vai trò, và kinh nghiệm. Nó cũng quy định môi trường phát triển, các công cụ làm việc, và bất kỳ kế hoạch đào tạo cụ thể nào cho dự án. Kế hoạch chất lượng và kế hoạch quản lý rủi ro cũng được đưa ra trong phần này.

3) *Phần theo dõi dự án*: Phần này xác định các phép đo được thực hiện và các hệ thống được sử dụng để thu thập dữ liệu, theo dõi các hoạt động khác nhau của dự án được thực hiện, tần suất và bản chất của báo cáo tiến độ, và các thủ tục leo thang.

4) *Phần đội dự án*: Phần này xác định đội dự án và cấu trúc của nó, cũng như vai trò và trách nhiệm của từng người trong đội.

Câu 27: Anh (Chị) hãy trình bày các loại chức năng của quản lý cấu hình mà các dự án phần mềm thường yêu cầu.

tính tự nhiên và hoàn cảnh cụ thể của từng dự án, nhưng một số chức năng chung có thể được xác định rõ.

Sau đây là danh sách các chức năng chung của quản lý cấu hình kèm theo các kịch bản cần đến chúng. Các chức năng này thì chi tiết hơn các chức năng quản lý cấu hình được định nghĩa bởi Humphrey [3].

+ *Cho biết các trạng thái của các chương trình:* Người quản lý dự án cần thông tin này để quyết định khi nào bắt đầu kiểm thử hoặc khi nào giao (phát hành) phần mềm.

+ *Cho biết phiên bản mới nhất của một chương trình:* Giả sử rằng một chương trình phải được sửa đổi. Rõ ràng rằng, việc sửa đổi phải được thực hiện trong phiên bản mới nhất, nếu không, các thay đổi trước đó có thể bị mất.

+ *Xử lý các yêu cầu cập nhật đồng thời:* Chẳng hạn, hai lập trình viên xử lý hai thay đổi khác nhau, có thể thay đổi đồng thời cùng một chương trình. Một trong những thay đổi có thể có khả năng ghi đè lên thay đổi khác. Để tránh trạng thái như vậy cần phải kiểm soát truy cập để chỉ có một người có thể thay đổi một chương trình tại một thời điểm. Nếu nhiều thay đổi song song được cho phép, các thủ tục hòa giải cần được thực hiện để đảm bảo rằng tất cả các thay đổi được phản ánh trong phiên bản cuối cùng.

+ *Hủy bỏ thay đổi một chương trình:* Một thay đổi được thực hiện lên một chương trình, nhưng sau đó lại có nhu cầu phát sinh để hủy bỏ việc thay đổi này.

+ *Ngăn chặn các thay đổi trái phép:* Chẳng hạn, một lập trình viên có thể quyết định thay đổi một số chương trình, chỉ để khám phá ra rằng sự thay đổi đó có tác dụng phụ. Cho nên, cần có cơ chế kiểm soát truy cập để các thay đổi trái phép sẽ không được chấp thuận.

+ *Cung cấp khả năng theo dõi dấu vết nguồn gốc giữa các thay đổi yêu cầu và các thay đổi chương trình:* Giả sử một đề nghị thay đổi yêu cầu dẫn đến ba chương trình phải được sửa đổi, và những sửa đổi này đã được phân công cho ba thành viên trong đội. Làm thế nào để người quản lý dự án đảm bảo rằng thay đổi yêu cầu đã được thực hiện đúng cách. Tức là, tất cả các chương trình đã được thay đổi và các chương trình vừa được thay đổi xong đã trải qua chu kỳ sống của chúng và đang ở trạng thái "sẵn sàng để phát hành". Để trả lời câu hỏi này đòi hỏi phải có một cơ chế để theo dõi các thay đổi yêu cầu, cơ chế này có thể xác định tất cả các chương trình được thay đổi cũng như trạng thái của từng chương trình.

+ *Hủy bỏ một thay đổi yêu cầu:* Một thay đổi yêu cầu cần được thực hiện (bằng cách phải thay đổi nhiều chương trình), nhưng sau đó lại cần phải hủy bỏ thay đổi vừa được thực hiện (có lẽ do người sử dụng không thích các tính năng mới).

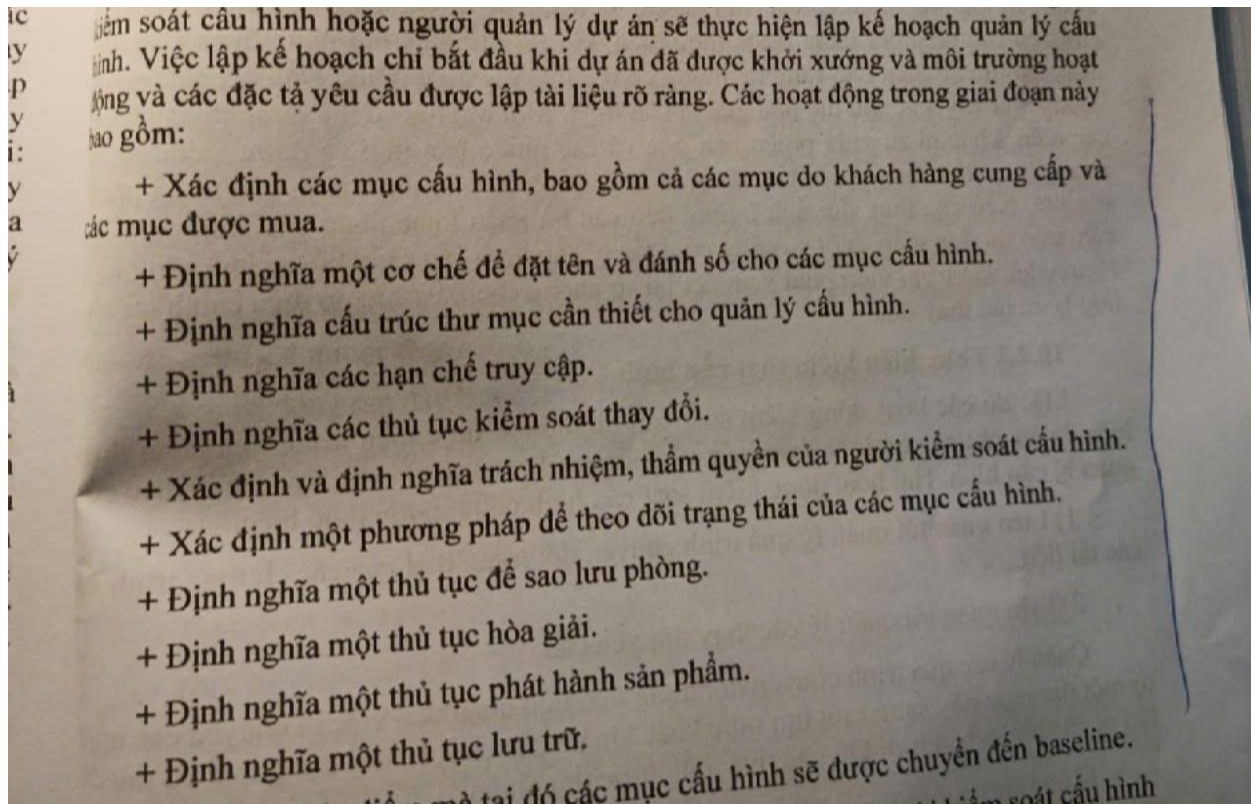
+ *Chỉ ra các thay đổi có liên quan:* Giả sử một lỗi được tìm thấy trong một chương trình, và nó bị nghi ngờ rằng lỗi này đến từ việc thực hiện thay đổi yêu cầu. Do đó, cần phải xem xét lại tất cả các thay đổi đã được thực hiện liên quan đến thay đổi yêu cầu này.

+ *Thu thập tất cả các mã nguồn, tài liệu, và các thông tin khác cho hệ thống hiện tại:* Chúng sẽ cần thiết trong trường hợp sửa đổi làm sai lệch tập tin hay sự sụp đổ hệ thống, cần phải phục hồi lại tất cả các tập tin của trạng thái trước đó. Tương tự như

Các chức năng cơ bản của quản lý cấu hình phần mềm:

1. **Xác định trạng thái:** Cho biết chương trình đang ở giai đoạn nào (ví dụ: đang code, chờ test, hay đã sẵn sàng giao cho khách hàng).
2. **Cung cấp phiên bản mới nhất:** Đảm bảo lập trình viên luôn lấy được bản code cập nhật nhất để làm việc, tránh tình trạng sửa đề làm mất code của nhau.
3. **Xử lý cập nhật đồng thời:** Kiểm soát và "hòa giải" (trộn code) khi có nhiều lập trình viên cùng sửa một file tại cùng một thời điểm.
4. **Hủy bỏ thay đổi (Hoàn tác/Undo):** Cho phép quay ngược một file về phiên bản cũ trước đó nếu bản cập nhật mới bị lỗi.
5. **Ngăn chặn sửa đổi trái phép:** Có cơ chế phân quyền, chỉ những ai được giao nhiệm vụ mới có quyền chỉnh sửa hệ thống.
6. **Theo dõi dấu vết nguồn gốc:** Gắn kết chặt chẽ một "đề nghị thay đổi" với những "file code cụ thể đã được sửa", giúp biết rõ code này sửa vì yêu cầu nào.
7. **Hủy bỏ một yêu cầu thay đổi:** Khôi phục hàng loạt file về trạng thái ban đầu nếu khách hàng đổi ý, không muốn áp dụng yêu cầu thay đổi đó nữa.
8. **Chỉ ra các thay đổi liên quan:** Giúp khoanh vùng nhanh chóng các file bị ảnh hưởng khi tình cờ phát hiện ra một lỗi mới trong hệ thống.
9. **Đóng gói hệ thống:** Thu thập, gom toàn bộ mã nguồn và tài liệu chuẩn xác nhất ở thời điểm hiện tại để chuẩn bị phát hành sản phẩm.

Câu 28: Anh (Chị) hãy trình bày các hoạt động trong giai đoạn lập kế hoạch quản lý cấu hình.



1. Xác định đối tượng cần quản lý

- Xác định các **mục cấu hình** thuộc dự án (bao gồm cả mã nguồn, tài liệu, và các mục do khách hàng cung cấp hoặc mua ngoài).

2. Thiết lập Tổ chức & Định danh

- Định nghĩa cơ chế **đặt tên và đánh số** (phiên bản) cho từng mục.
- Xây dựng **cấu trúc thư mục** để lưu trữ một cách khoa học.

3. Phân quyền & Trách nhiệm

- Định nghĩa các **hạn chế truy cập** (quy định rõ ai được quyền xem, ai được quyền sửa).
- Xác định rõ **trách nhiệm và thẩm quyền** của người làm nhiệm vụ kiểm soát cấu hình.

4. Định nghĩa các Thủ tục / Quy trình (Quan trọng nhất)

Phải xây dựng sẵn các quy trình chuẩn để xử lý mọi tình huống:

- Thủ tục **kiểm soát thay đổi** và phương pháp **theo dõi trạng thái** của phần mềm.
- Thủ tục **hòa giải** (xử lý khi nhiều người sửa code cùng lúc).
- Thủ tục **sao lưu phòng ngừa** (backup) và **lưu trữ** dữ liệu.
- Thủ tục **phát hành sản phẩm**.
- Xác định các điểm (cột mốc) mà tại đó sản phẩm sẽ được chốt thành **Baseline**

(Đường cơ sở).

Câu 29: Anh (Chị) hãy trình bày các hoạt động của việc thực hiện một thay đổi yêu cầu trong kiểm soát cấu hình.

Để thực hiện các thay đổi yêu cầu, cần phải lần lượt kích hoạt thay đổi đến các mục cấu hình, thay đổi yêu cầu trước tiên được phân tích bằng cách thực hiện một phân tích tác động. Phân tích này xác định các chương trình và các tài liệu cần được thay đổi và ước lượng chi phí và thời gian để thực hiện thay đổi. Sau khi thay đổi được sự chấp thuận của khách hàng, người lãnh đạo dự án và người kiểm soát cấu hình, tất cả các chương trình và các tài liệu đã được xác định bởi hoạt động phân tích tác động phải được thay đổi một cách hợp lý.

Các hoạt động sau đây là một phần của thực hiện một thay đổi yêu cầu:

- + Chấp nhận thay đổi yêu cầu (với phân tích tác động).
- + Thiết lập một cơ chế theo dõi.
- + Tách ra các mục cấu hình cần thay đổi.
- + Thực hiện các thay đổi.
- + Đưa các mục cấu hình sau khi thay đổi trở lại môi trường có kiểm soát.
- + Đưa các mục cấu hình qua chu kỳ sống của nó.

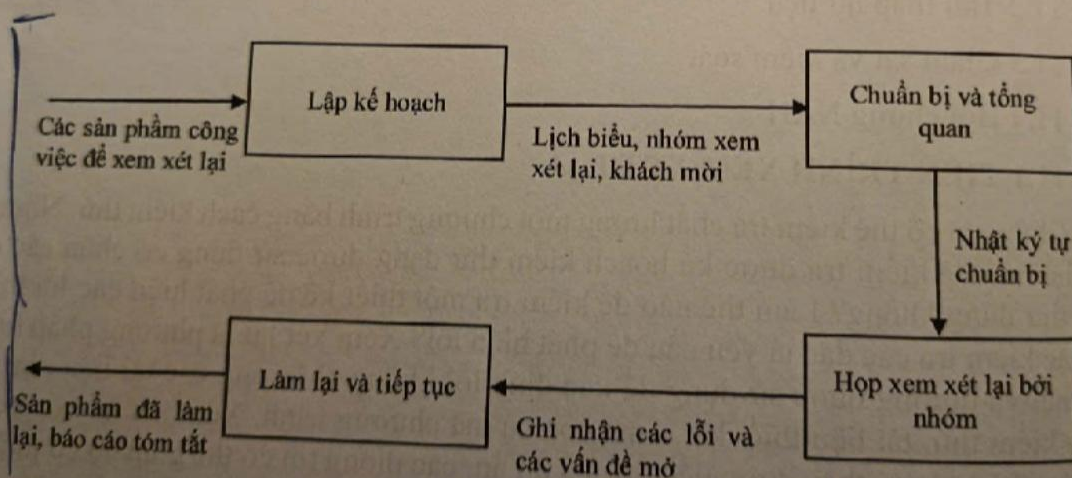
Câu 30: Anh (Chị) hãy trình bày tiến trình xem xét lại bởi nhóm và giải thích tóm tắt các hoạt động của tiến trình này.

+ Xem xét lại giúp phòng ngừa lỗi bằng cách tạo ra nhận thức nhiều hơn về chúng.

Xem xét lại có thể được thực hiện bằng nhiều cách khác nhau. Một xem xét lại bởi nhóm theo đúng hình thức có lẽ là một lựa chọn tốt nhất để xác định lỗi. Xem xét lại phần mềm lần đầu tiên được đề xuất bởi Fagan [1][2]. Các xem xét lại trước đây chỉ tập trung vào mã chương trình, nhưng trong những năm qua kỹ thuật này đã lan rộng đến các sản phẩm công việc khác. Ngày nay, xem xét lại phần mềm đã được công nhận là thực hành tốt nhất của công nghiệp phần mềm. Có nhiều nguồn dữ liệu để chứng minh rằng xem xét lại giúp nâng cao chất lượng và năng suất trong các dự án phần mềm (Gilb và Graham [3] Weller [4] Grady và Slack [5]).

Tiến trình xem xét lại cơ bản tại Infosys là việc xem xét lại bởi nhóm. Xem xét lại bởi nhóm là phân tích sản phẩm công việc được làm bởi một nhóm các đồng nghiệp bằng cách tuân theo một tiến trình đã được định nghĩa rõ ràng. Các mục tiêu của xem xét lại là để nâng cao chất lượng bằng cách tìm ra các lỗi và nâng cao năng suất bằng cách tìm ra các lỗi một cách hiệu quả về mặt chi phí. Ngoài ra, xem xét lại cung cấp khả năng nhìn thấy được trạng thái của dự án để giúp ích cho quản lý dự án ở khía cạnh chất lượng của các sản phẩm công việc. Một nhóm xem xét lại được xây dựng bởi những người kỹ thuật, và tập trung vào việc xác định các vấn đề, nhưng không đi giải quyết chúng.

Tiến trình xem xét lại bởi nhóm bao gồm các giai đoạn: Lập kế hoạch, chuẩn bị và tổng quan, họp xem xét lại bởi nhóm, làm lại công việc và tiếp tục. Như được trình bày trong Hình 11.1, các giai đoạn này thường được thực hiện theo một thứ tự tuyến tính.



Hình 11.1. Tiến trình xem xét lại bởi nhóm

Tiến trình xem xét lại bởi nhóm (Group Review) được thực hiện tuần tự qua 4 bước (giai đoạn) cốt lõi sau:

1. Lập kế hoạch (Planning)

- Xác định sản phẩm công việc nào cần xem xét.
- Phân công vai trò cho các thành viên (Người điều phối, Tác giả, Người đọc, Thư ký).
- Lên lịch biểu chi tiết cho cuộc họp.

2. Chuẩn bị và Tổng quan (Preparation & Overview)

- Tác giả có thể trình bày tổng quan về sản phẩm (nếu nó quá phức tạp).
- Các thành viên nhận tài liệu và **tự đọc, tự nghiên cứu độc lập** trước ở nhà.
- Mỗi người tự ghi lại các lỗi mình phát hiện ra vào *Nhật ký tự chuẩn bị*.

3. Họp xem xét lại bởi nhóm (Group Meeting)

- Cả nhóm ngồi lại với nhau. Người đọc sẽ đọc tuần tự từng phần của tài liệu/mã nguồn.
- Các thành viên đối chiếu với phần chuẩn bị ở nhà để chỉ ra lỗi.
- Thư ký ghi chép lại toàn bộ lỗi vào *Nhật ký cuộc họp*.
- **Nguyên tắc vàng:** Cuộc họp này **chỉ để tìm lỗi**, tuyệt đối không mất thời gian tranh luận tìm giải pháp sửa lỗi.

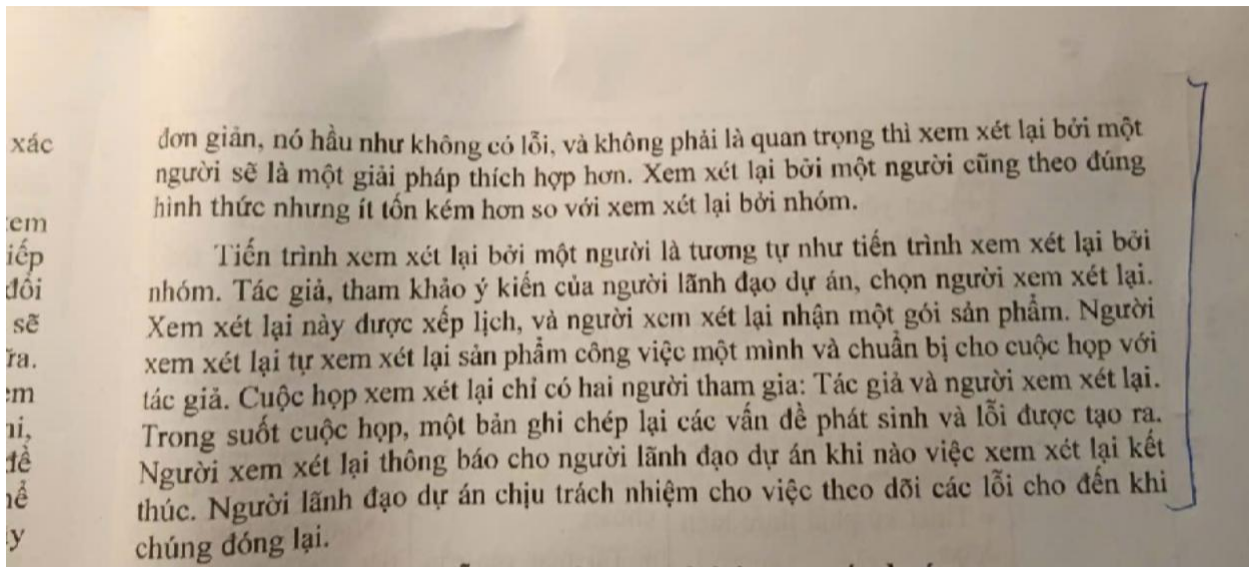
4. Làm lại và Tiếp tục (Rework & Follow-up)

- **Làm lại:** Tác giả mang danh sách lỗi về và tự tiến hành sửa chữa, khắc phục sản phẩm.
- **Tiếp tục (Kiểm tra lại):** Người điều phối kiểm tra lại lần cuối để đảm bảo mọi lỗi trong danh sách đã được tác giả khắc phục hoàn toàn.

Câu 31: Anh (Chị) hãy cho biết khi nào thì thực hiện xem xét lại bởi 1 người. Trình bày tiến trình xem xét lại bởi 1 người.

11.1.5 Xem xét lại bởi một người

Xem xét lại bởi nhóm là cách rất hiệu quả để xác định lỗi. Tuy nhiên, chi phí của nó cũng cao vì nhiều người tham gia xem xét lại. Ngoài ra, việc tổ chức các cuộc họp xem xét lại cũng là một vấn đề phức tạp. Nếu sản phẩm công việc có nhiều lỗi hoặc quan trọng, chi phí này là hợp lý. Trái lại, nếu các sản phẩm công việc là tương đối



1. Khi nào thì thực hiện xem xét lại bởi 1 người?

Thay vì gọi cả nhóm gây tốn kém thời gian và chi phí, việc xem xét lại bởi 1 người được áp dụng khi thỏa mãn các điều kiện:

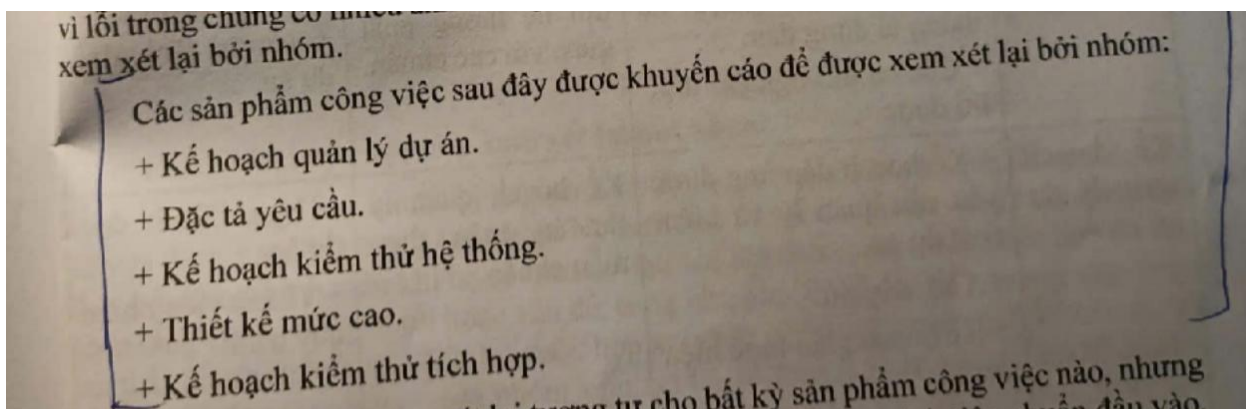
- Sản phẩm công việc tương đối **đơn giản**.
- Sản phẩm **không quá quan trọng** (ít rủi ro).
- Dự kiến sản phẩm **hầu như không có lỗi**.

2. Tiến trình xem xét lại bởi 1 người

Tiến trình này vẫn giữ sự bài bản như xem xét bởi nhóm, nhưng được rút gọn lại qua 4 bước:

- **Bước 1: Lên lịch & Phân công:** Tác giả (sau khi hỏi ý kiến Quản lý dự án - PM) sẽ chọn ra đúng 1 người để xem xét và gửi "gói sản phẩm" cho người đó.
- **Bước 2: Tự chuẩn bị:** Người xem xét tiến hành đọc và rà soát sản phẩm **một mình** để tìm ra các lỗi.
- **Bước 3: Họp xem xét (Chỉ 2 người):** Tác giả và người xem xét gặp nhau. Hai bên cùng thảo luận và lập ra một bản ghi chép danh sách các lỗi / vấn đề phát sinh.
- **Bước 4: Báo cáo & Theo dõi:** Người xem xét báo cho PM biết cuộc họp đã xong. Sau đó, PM sẽ chịu trách nhiệm đôn đốc, theo dõi cho đến khi tác giả sửa xong toàn bộ lỗi (đóng lỗi).

Câu 32: Anh (Chị) cho biết các sản phẩm công việc nào cần được xem xét lại bởi nhóm? Khi xem xét lại đặc tả yêu cầu cần quan tâm đến những khía cạnh nào? Thành phần tham gia nhóm xem xét lại sản phẩm này?



Đặc tả yêu cầu	+ Các yêu cầu đáp ứng đúng nhu cầu khách hàng. + Các yêu cầu phải thực hiện được. + Bỏ sót, không nhất quán, và mơ hồ trong các yêu cầu.	+ Tài liệu yêu cầu tuân theo các chuẩn.	+ Khách hàng. + Người thiết kế mức cao. + Người kiểm thử hệ thống. + Tác giả của tài liệu yêu cầu người dùng.
----------------	--	---	--

1. Các sản phẩm công việc CẦN được xem xét lại bởi nhóm (5 tài liệu lớn)

Đây là những tài liệu xương sống của dự án, nếu sai sót sẽ gây hậu quả nghiêm trọng:

- Kế hoạch quản lý dự án.
- Đặc tả yêu cầu.
- Thiết kế mức cao.
- Kế hoạch kiểm thử hệ thống.
- Kế hoạch kiểm thử tích hợp.

2. Các khía cạnh cần quan tâm khi xem xét "Đặc tả yêu cầu"

Khi review tài liệu yêu cầu, nhóm phải soi kỹ 4 vấn đề sau:

- **Tính chính xác:** Yêu cầu có đáp ứng đúng nhu cầu thực tế của khách hàng không?
- **Tính khả thi:** Các yêu cầu đưa ra có thực sự thực hiện được không?
- **Tính rõ ràng, đầy đủ:** Có lỗi bỏ sót, mơ hồ hay không nhất quán (mâu thuẫn nhau) trong các yêu cầu không?
- **Tính chuẩn mực:** Hình thức tài liệu có tuân theo đúng các tiêu chuẩn quy định không?

3. Thành phần tham gia nhóm xem xét Đặc tả yêu cầu

Gồm 4 đối tượng liên quan trực tiếp nhất đến tài liệu này:

- **Khách hàng:** Người đưa ra yêu cầu và nghiệm thu.
- **Tác giả của tài liệu:** Người chấp bút viết ra bản đặc tả yêu cầu.
- **Người thiết kế mức cao:** Người sẽ dùng tài liệu này làm đầu vào để thiết kế hệ thống.
- **Người kiểm thử hệ thống:** Người sẽ dựa vào tài liệu này để lên kịch bản test (kiểm tra xem phần mềm có chạy đúng yêu cầu không).

(Ghi nhớ logic: Người viết (Tác giả) → Người duyệt (Khách hàng) → Người dùng tài liệu để làm bước tiếp theo (Thiết kế & Kiểm thử). Đi thi bạn cứ nhớ theo luồng này là không bao giờ quên!)

Câu 33: Nếu trong phân tích tại cột mốc, nỗ lực thực tế thấp hơn so với nỗ lực ước lượng quá giới hạn thì những nguyên nhân gì có thể và cách kiểm soát như thế nào?

khoảng 20% và giới hạn chênh lệch lịch biểu là 10%. Các giới hạn thực tế cho một dự án được quy định cụ thể trong bản kế hoạch của nó.

Bảng 12.4. Các hướng dẫn cho cho hoạt động phân tích và kiểm soát

Lý do có thể	Các hoạt động kiểm soát
Nếu nỗ lực thực tế nhỏ hơn ước lượng một lượng lớn hơn giới hạn cho phép	
Các ước lượng cho các đơn vị chương trình quá cao.	Ước lượng lại các mô đun còn lại.
Đội dự án có kiến thức và kinh nghiệm chuyên môn cao hơn dự kiến.	Ước lượng lại nhân công.
Các nhiệm vụ đến thời điểm hiện tại vẫn chưa được thực hiện triệt để và kỹ lưỡng.	+ Xem xét lại các nhiệm vụ đã được làm đến thời điểm hiện tại. + Xem xét lại lịch của các sản phẩm công việc vẫn chưa được xem xét lại. + Kiểm tra lại nhật ký vấn đề.
Nếu nỗ lực thực tế lớn hơn ước lượng một lượng lớn hơn giới hạn cho phép	
Kiến thức chuyên môn thấp.	Xếp lịch để đào tạo.
Kinh nghiệm phần mềm/viết mã của tác giả thấp.	Phân công lại nhân sự để tận dụng kinh nghiệm sẵn có.
Lĩnh vực công nghệ mới.	+ Ước lượng lại hoặc yêu cầu về thêm nhân sự. + Đàm phán lại ngày giao sản phẩm.
Các ước lượng đã quá lạc quan.	+ Xác định các thành phần cần bổ sung thêm nỗ lực, ước lượng lại cho các hoạt động còn lại. + Yêu cầu về nhân sự. + Đàm phán để thu hẹp quy mô dự án lại.
Tối ưu hóa nguồn nhân sự thấp.	Xếp lại lịch và xếp lại độ ưu tiên cho các nhiệm vụ, xác định và loại bỏ "thời gian chết".
Một thành viên quan trọng vắng mặt.	+ Lấy nhân sự dự phòng. + Xếp lại lịch, luôn nghĩ tới (các) nhân sự quan trọng trong đầu.
Quá nhiều công việc bị làm lại do chất lượng kém của các giai đoạn trước.	+ Xác định nguồn gốc của vấn đề và sửa lại chúng. + Thay đổi lịch của dự án.

Khi phân tích tại một cột mốc, nếu thấy **nỗ lực (thời gian/giờ công) thực tế thấp hơn ước lượng vượt qua cả giới hạn cho phép** (tức là xong quá nhanh một cách bất thường), người quản lý (PM) cần xem xét các ý sau:

1. Các nguyên nhân có thể xảy ra (Tại sao lại nhanh như vậy?)

- **Làm ẩu, bỏ bước (Nguy hiểm nhất):** Nhân viên có thể đã cắt xén quy trình, làm qua loa, bỏ qua các bước kiểm tra (review, test) hoặc code cầu thả để đẩy phó tiến độ.
- **Năng lực nhân sự xuất sắc:** Công việc vô tình được giao cho một chuyên gia có quá nhiều kinh nghiệm, họ giải quyết vấn đề nhanh hơn rất nhiều so với mức trung bình.
- **Ước lượng ban đầu bị sai:** Kế hoạch ban đầu đã dự tính thời gian quá dư thừa (overestimate) so với độ khó thực tế của công việc.

2. Cách kiểm soát và khắc phục (PM cần làm gì?)

- **Kiểm tra chất lượng (Audit):** Ngay lập tức tổ chức rà soát, kiểm tra lại (review) các sản phẩm vừa làm xong để đảm bảo không bị bỏ sót công việc và chất lượng vẫn đạt chuẩn. Bắt buộc nhân viên làm lại nếu phát hiện làm ẩu.
- **Điều chỉnh lại ước lượng:** Nếu nguyên nhân là do ước lượng ban đầu bị sai, PM cần tính toán và điều chỉnh lại nỗ lực (thời gian) cho các giai đoạn tiếp theo để bám sát với thực tế hơn.
- **Cập nhật dữ liệu:** Nếu do nhân sự làm việc năng suất cao, PM cần ghi nhận vào cơ sở dữ liệu tiến trình để có số đo chính xác hơn khi giao việc cho người này ở các dự án sau.

Câu 34: Nếu trong phân tích tại cột mốc, số lỗi thực tế thấp hơn so với số lỗi ước lượng quá giới hạn thì những nguyên nhân gì có thể và cách kiểm soát như thế nào?

get kế các hướng
có thể xem xét

lượng quá thấp
hành động
quá lạc quan,
các thành viên
kiểm soát đã
n có thể xem
trán lại. Hầu

được giảm
như mong
sau khi các
không hiệu

phần còn
sẵn có và
số nhân
các hành
kịp thời

được
ử dụng
u và số
o sánh
ng của
hành
quan

đề này đã được thảo luận trong chương 11.

Bảng 12.5. Các hướng dẫn khi số lỗi được tìm thấy lệch với ước lượng

Lý do có thể	Các hoạt động kiểm soát
Nếu lỗi thực tế thấp hơn ước lượng	
Sản phẩm công việc có chất lượng cao.	Xác định nguyên nhân và rút ra bài học kinh nghiệm cho dự án hoặc cho tiến trình.
Kiểm thử chưa đủ.	+ Xem lại nỗ lực đã tiêu phí cho kiểm thử. + Xếp lịch cho kiểm thử thêm.
Các hoạt động kiểm soát chất lượng trước đây đã được thực hiện rất kỹ lưỡng.	+ Kiểm tra lại tất cả các nhật ký về xem xét lại và về kiểm thử. + Kiểm tra xem có bài học nào được rút ra cho dự án hoặc cho tiến trình hay không.
Các ước lượng về số lỗi khi lập kế hoạch thì quá cao.	Xác định nguyên nhân và sửa chữa lại các ước lượng trong bản kế hoạch.
Nếu lỗi thực tế cao hơn ước lượng	
Cho đến thời điểm hiện tại, các hoạt động kiểm soát chất lượng chưa được thực hiện đầy đủ.	+ Kiểm tra lại tất cả các nhật ký về kiểm thử và về xem xét lại. + Xếp lịch để xem xét lại các mô đun quan trọng trước khi tiếp tục kiểm thử.
Việc xem xét lại và kiểm thử đơn vị đã được lập kế hoạch chưa đủ.	+ Mở rộng kế hoạch kiểm thử và xếp lịch để kiểm thử thêm. + Mở rộng kế hoạch xem xét lại, xếp lịch xem xét lại thêm.
Ước lượng về số lỗi quá thấp.	Xác định nguyên nhân và sửa chữa lại các ước lượng.

134

2.2. Giám sát và kiểm soát rủi ro tại cột mốc

như đã

Khi phân tích tại một cột mốc, nếu thấy **số lỗi thực tế thấp hơn ước lượng vượt qua cả giới hạn dưới** (nghĩa là tìm được quá ít lỗi so với mức bình thường), đừng vội mừng. Quản lý dự án (PM) cần xem xét các điểm sau:

1. Các nguyên nhân có thể xảy ra (Tại sao lại ít lỗi thế?)

- **Kiểm tra hời hợt, bỏ sót lỗi (Phổ biến & Nguy hiểm nhất):** Người xem xét (reviewer) hoặc người kiểm thử (tester) làm việc cẩu thả, không chuẩn bị kỹ, hoặc thiếu kinh nghiệm nên không nhìn ra lỗi.
- **Không ghi nhận lỗi:** Có tìm ra lỗi nhưng các thành viên lại tự sửa ngầm với nhau, không ghi chép (log) lại vào hệ thống báo cáo.
- **Mức độ phức tạp thấp:** Sản phẩm công việc thực tế quá đơn giản, không khó và nhiều rủi ro như lúc ước lượng ban đầu.
- **Chất lượng công việc xuất sắc (Rất hiếm):** Lập trình viên/tác giả làm việc quá hoàn hảo, code cực kỳ sạch và chuẩn (nhưng thực tế ít khi đạt được mức này).

2. Cách kiểm soát và khắc phục (PM cần làm gì?)

- **Yêu cầu làm lại khâu kiểm tra (Rework):** Bắt buộc nhóm rà soát hoặc kiểm thử lại sản phẩm đó một lần nữa. Yêu cầu phải bám sát **danh sách kiểm tra (checklist)** để đảm bảo không bỏ sót.

- **Đào tạo lại nhân sự:** Nếu tester hoặc người xem xét năng lực yếu, PM cần tổ chức hướng dẫn, đào tạo lại kỹ năng tìm lỗi cho họ.
- **Chấn chỉnh quy trình báo cáo:** Nhắc nhở, ép kỷ luật đội dự án phải ghi nhận (log) đầy đủ 100% số lỗi tìm được vào cơ sở dữ liệu.
- **Điều chỉnh ước lượng:** Nếu nguyên nhân thực sự là do hệ thống đơn giản hoặc code quá tốt, PM cần ghi nhận dữ liệu để cập nhật lại mức ước lượng lỗi (baseline) cho các dự án sau này.

Câu 35: Anh (Chị) hãy trình bày các bước xây dựng sơ đồ nhân quả để phân tích và phòng ngừa lỗi.

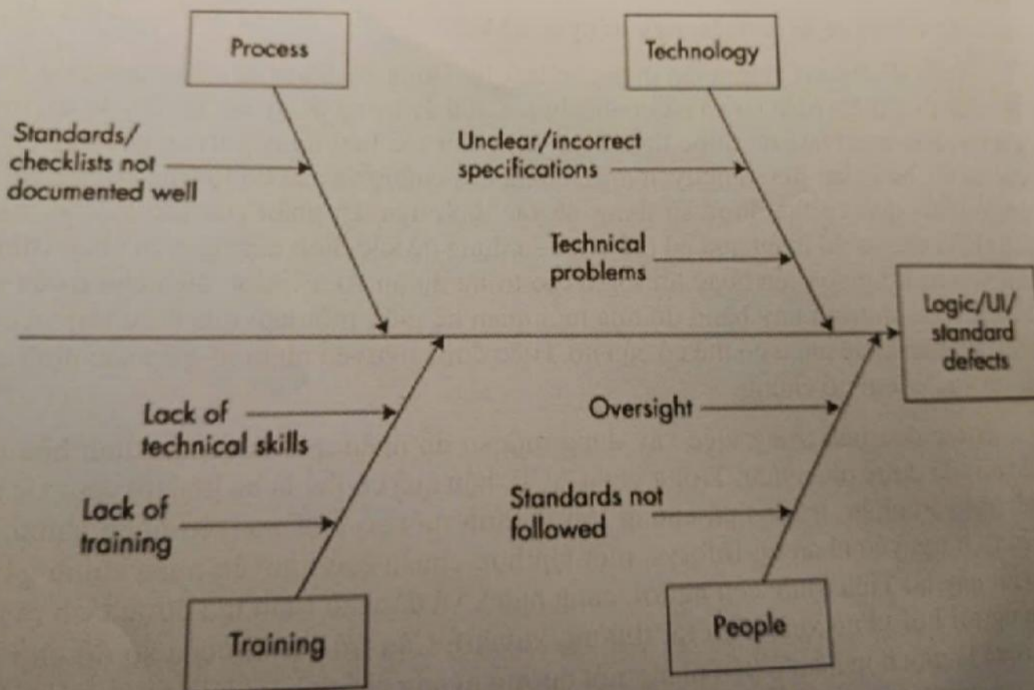
con người bởi vì nó xuất hiện rất thường xuyên). Cấu trúc chính của sơ đồ cho thấy hậu quả là một hộp nằm ở bên phải, một đường ngang chính kéo dài từ bên trái tới hộp này, và các đường kết nối mỗi nguyên nhân chính vào đường ngang chính.

Một nguyên nhân chính có thể chứa nhiều nguyên nhân con. Các nguyên nhân con của mỗi nguyên nhân chính được biểu diễn như là các đường ngang phụ nối vào đường đang trình bày nguyên nhân chính. Các bước chính để vẽ một sơ đồ nhân quả (còn gọi là sơ đồ xương cá) như sau:

1. Xác định các lỗi chính đưa vào hộp "hậu quả".
2. Vẽ một mũi tên từ trái tới hộp chứa "hậu quả" ở đầu bên phải. Đây là xương sống của sơ đồ.
3. Xác định các loại nguyên nhân chính. Đây có thể là các loại nguyên nhân chuẩn đã được xây dựng bởi công ty.
4. Viết những nguyên nhân chính này vào trong các hộp và kết nối chúng bằng các mũi tên chéo đến xương sống của sơ đồ. Những nguyên nhân này hình thành các xương sườn của sơ đồ.
5. Thu thập ý kiến của nhóm phòng ngừa lỗi về các nguyên nhân con của các nguyên nhân chính.
6. Thêm các nguyên nhân con vào sơ đồ xung quanh xương sườn của nguyên nhân chính. Nếu cần thiết, chia nhỏ hơn nữa những nguyên nhân này.

Khi sơ đồ xương cá được vẽ xong, bạn đã xác định được tất cả các nguyên nhân của hậu quả đang được phân tích. Tuy nhiên, nhiều khả năng sơ đồ xương cá ban đầu

sẽ có quá nhiều nguyên nhân, một số nguyên nhân có tác động lớn hơn so với những nguyên nhân khác. Do đó, trước khi hoàn tất việc phân tích nguyên nhân gốc rễ, bạn hãy xác định những nguyên nhân hàng đầu, chủ yếu là thông qua thảo luận trong nhóm. Để phòng ngừa lỗi, bạn có thể thực hiện các bước thực hành này cho một vài loại lỗi đầu đã được tìm thấy trong phân tích Pareto.



Hình 12.3. Sơ đồ nhân quả cho dự án ACE

Sơ đồ nhân quả (hay Sơ đồ xương cá / Ishikawa) là một công cụ trực quan rất nổi tiếng, giúp nhóm dự án "mổ xẻ" vấn đề để tìm ra tận gốc rễ lý do vì sao một lỗi lại xảy ra, từ đó có biện pháp phòng ngừa.

Quá trình xây dựng sơ đồ này gồm **5 bước chuẩn**:

1. Xác định vấn đề (Vẽ "Đầu cá")

- Thống nhất và viết rõ ràng vấn đề/lỗi đang gặp phải vào một hộp ở mép phải của bảng.
- Vẽ một đường trục chính nằm ngang nổi thẳng vào hộp đó (như xương sống con cá).

2. Xác định các nhóm nguyên nhân chính (Vẽ "Xương lớn")

- Xác định các nhóm yếu tố bao quát nhất có thể gây ra lỗi và vẽ các nhánh chéo đâm vào trục chính.
- *Gợi ý 4 nhóm phổ biến nhất trong phần mềm*: Con người (Thiếu kinh nghiệm?), Quy trình (Làm tắt bước?), Công cụ (Máy móc/Phần mềm hỏng?), Môi trường (Ồn ào, áp lực?).

3. Động não tìm nguyên nhân chi tiết (Vẽ "Xương nhỏ")

- Tổ chức họp nhóm (Brainstorming) để mọi người cùng đưa ra ý kiến.

- Liên tục đặt câu hỏi "**Tại sao?**" (phương pháp 5 Whys) để đào sâu vấn đề, sau đó gắn các nguyên nhân chi tiết này vào các nhánh "xương lớn" tương ứng.

4. Xác định nguyên nhân gốc rễ (Root Cause)

- Sau khi vẽ xong một sơ đồ chi chít các nguyên nhân, nhóm sẽ cùng thảo luận, đánh giá và khoanh vùng lại: Đây là nguyên nhân cốt lõi, mấu chốt nhất thực sự sinh ra lỗi này?

5. Đề xuất hành động phòng ngừa

- Khi đã bắt đúng "bệnh" (nguyên nhân gốc), lập tức đưa ra kế hoạch hành động cụ thể để khắc phục tận gốc, đảm bảo lỗi này sẽ bị chặn đứng và không bao giờ xuất hiện lại ở các giai đoạn sau hoặc dự án sau.

*(Mẹo đi thi: Bạn chỉ cần nhớ hình ảnh con cá, đi từ **Đầu cá (Vấn đề)** → **Xương lớn (Nhóm)** → **Xương nhỏ (Chi tiết)** → **Bắt đúng bệnh (Gốc rễ)** → **Chữa bệnh (Phòng ngừa)** là sẽ viết được trôi chảy trọn vẹn câu này!)*

Câu 36: Anh (Chị) hãy trình bày vai trò của phân tích kết thúc dự án.

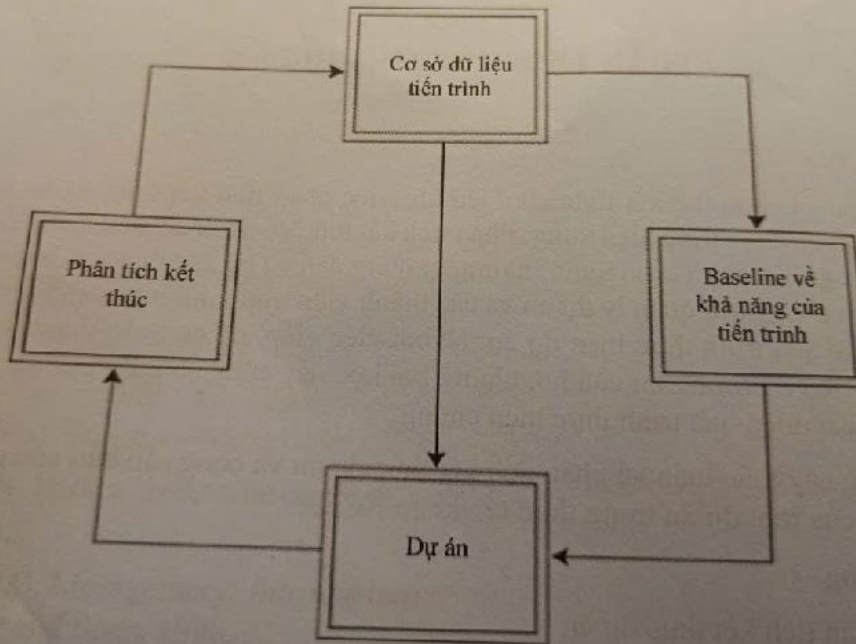
[1][2][3][4]. Trước khi chúng ta mô tả chi tiết về báo cáo phân tích kết thúc của một dự án thực tế, chúng ta thảo luận ngắn gọn về vai trò của phân tích kết thúc và quá trình thực hiện nó.

13.1.1 Vai trò của phân tích kết thúc dự án

Mục tiêu của một phân tích kết thúc dự án là rút ra bài học kinh nghiệm, phân tích kết thúc là để "xác định những gì đã đi đúng, những gì đã đi sai, những gì đã hoạt động tốt, những gì không tốt, và từ đó trả lời cho câu hỏi làm thế nào để có thể thực hiện tốt hơn trong tương lai?". Thông tin quan trọng phải được thu thập từ dự án vừa thực hiện, chủ yếu là để sử dụng cho các dự án trong tương lai. Tức là, mục đích của hoạt động phân tích không chỉ đơn giản chứng tỏ "Dự án đã được làm xong", cũng không phải để giúp cho dự án vừa làm xong này mà là để giúp công ty bằng cách rút ra các bài học kinh nghiệm. Phân tích này cũng cần làm rõ hiệu suất của tiến trình phần mềm đã được dùng trong dự án.

Như đã đề cập trước đây, các dữ liệu thu được trong quá trình phân tích kết thúc được lưu giữ vào cơ sở dữ liệu tiến trình. Các dữ liệu từ cơ sở dữ liệu tiến trình có thể

được sử dụng trực tiếp bởi các dự án tiếp theo cho mục đích lập kế hoạch. Thông tin này cũng được sử dụng trong việc tính toán Baseline về khả năng của tiến trình, được sử dụng để lập kế hoạch và phân tích định hướng cho các dự án khác trong tương lai. Hình 13.1 minh họa vai trò của phân tích kết thúc dự án.



Phân tích kết thúc dự án (Post-mortem Analysis) là bước "tổng kết và nhìn lại" sau khi phần mềm đã hoàn thành và giao cho khách hàng. Nhóm dự án sẽ ngồi lại với nhau không phải để "đổ lỗi", mà để thực hiện **4 vai trò cốt lõi** sau:

1. Đánh giá hiệu suất thực tế của dự án

- Trực tiếp so sánh các con số thực tế đạt được (nỗ lực, chi phí, tiến độ, chất lượng) với bản kế hoạch ban đầu.
- Giúp trả lời câu hỏi: *Dự án này thành công hay thất bại? Lệnh kế hoạch bao nhiêu phần trăm?*

2. Thu thập và đóng góp dữ liệu (Quan trọng nhất)

- Tổng hợp toàn bộ các số đo thực tế cuối cùng (Số dòng code, tổng giờ công, tổng số lỗi, thời gian hoàn thành...).
- Đưa các số liệu này lưu trữ vào **Cơ sở dữ liệu tiến trình** của công ty, làm tài sản (baseline) để giúp việc ước lượng cho các dự án trong tương lai chính xác hơn.

3. Rút ra bài học kinh nghiệm (Lessons Learned)

- Tổng kết lại những **điểm tốt** (công cụ hay, cách làm hiệu quả) để tiếp tục phát huy.
- Phân tích những **điểm xấu, sai lầm** đã vấp phải để các đội dự án khác trong công ty "không giẫm lại vết xe đổ".

4. Phản hồi để cải tiến quy trình chung

- Dựa trên những bài học rút ra, đề xuất lên cấp quản lý để cập nhật, nâng cấp các quy trình và tiêu chuẩn làm việc của công ty ngày càng hoàn thiện và chuyên nghiệp hơn.

*(Mẹo đi thi: Bạn cứ nhớ logic của việc tổng kết là: **Nhìn lại quá khứ (Đánh giá)** → **Đóng gói số liệu (CSDL)** → **Rút kinh nghiệm** → **Cải tiến tương lai**. Viết theo luồng này là chắc chắn trọn vẹn điểm!)*

Câu 37: Anh (Chị) hãy cho biết trong báo cáo phân tích kết thúc dự án bao gồm những loại dữ liệu chính nào?

- + Thông tin chung và thông tin liên quan đến tiến trình
- + Thông tin về quản lý rủi ro
- + Thông tin về kích thước
- + Thông tin về năng suất
- + Thông tin về nỗ lực
- + Dữ liệu lỗi
- + Phân tích nhân quả

Báo cáo kết thúc dự án giống như một cuốn "nhật ký tổng kết", trong đó bắt buộc phải chứa **7 nhóm dữ liệu cốt lõi** sau để lưu vào cơ sở dữ liệu tiến trình của công ty:

1. **Thông tin chung và thông tin liên quan đến tiến trình:** Các thông tin cơ bản về dự án, loại dự án và mô hình tiến trình đã được áp dụng.
2. **Thông tin về quản lý rủi ro:** Các dữ liệu liên quan đến những rủi ro đã dự tính, rủi ro thực tế xảy ra và cách dự án đã kiểm soát chúng.
3. **Thông tin về kích thước:** Quy mô của sản phẩm phần mềm làm ra (thường được đo bằng số dòng mã lệnh LOC hoặc điểm chức năng FP).
4. **Thông tin về năng suất:** Hiệu quả làm việc thực tế của đội dự án (ví dụ: hoàn thành được bao nhiêu lượng công việc trong một tháng).
5. **Thông tin về nỗ lực:** Tổng thời gian và công sức (giờ công/tháng công) thực tế đã tiêu tốn so với bản ước lượng ban đầu.
6. **Dữ liệu lỗi:** Tổng số lỗi được tìm thấy, tỷ lệ tìm lỗi và hiệu quả của các khâu gỡ bỏ lỗi (review, kiểm thử).
7. **Phân tích nhân quả:** Việc mổ xẻ tìm ra nguyên nhân gốc rễ của các vấn đề hoặc các lỗi lớn, từ đó đưa ra bài học kinh nghiệm cho các dự án sau.

*(Mẹo đi thi: Bạn chỉ cần học thuộc luồng logic của một dự án từ tổng quan đến chi tiết: **Thông tin chung** → **Rủi ro** → **Kích thước** → **Năng suất** → **Nỗ lực** → **Lỗi** → **Tìm nguyên nhân (Nhân quả)** là sẽ viết được trôi chảy mà không sợ bị sót ý nhé!)*

Câu 38: Bài học kinh nghiệm rút ra từ dự án ACIC là gì?

7	Nhất quán	2	1,7
8	Khả năng tái sử dụng	1	0,9
	Tổng	116	

10. PHÂN TÍCH NHÂN QUẢ VÀ BÀI HỌC KINH NGHIỆM

Có rất ít các chênh lệch lớn ở hiệu suất tiến trình, hiệu suất thực tế đã gần bằng với những gì đã được dự kiến. Những lý do chênh lệch đối với những chênh lệch lớn được đưa ra kèm theo. Một số bài học kinh nghiệm quan trọng là:

(1) Phát triển tăng dần hoặc phát triển theo từng giai đoạn là vô cùng hữu ích để đạt được chất lượng và năng suất cao hơn bởi vì dữ liệu từ giai đoạn đầu tiên có thể được sử dụng để cải thiện các giai đoạn còn lại thông qua việc phòng ngừa lỗi.

(2) Phòng ngừa lỗi có thể làm giảm đáng kể tỷ lệ tiềm lỗi. Phòng ngừa lỗi giúp tiết kiệm nhiều nỗ lực vì giảm nỗ lực làm lại.

(3) Nếu thay đổi yêu cầu có ảnh hưởng lớn, việc thảo luận với khách hàng thông qua phân tích chi tiết các tác động có thể là rất hữu ích. Làm một phân tích lợi ích về mặt chi phí đúng cách có thể dẫn đến việc phải trì hoãn các thay đổi, như đã xảy ra trong dự án này.

(4) Hiệu quả loại bỏ lỗi của hoạt động xem xét lại mã và kiểm thử đơn vị còn thấp. Quy trình cho cả hai cần phải được xem xét lại để cải thiện tính hiệu quả. Trong dự án này, kiểm thử tích hợp và hệ thống đã bù đắp cho sự kém hiệu quả của các hoạt động xem xét lại và kiểm thử đơn vị. Tuy nhiên, đối với các dự án lớn, điều này thì không thể và sự kém hiệu quả trong các hoạt động xem xét lại và kiểm thử đơn vị có thể có ảnh hưởng xấu đến chất lượng của dự án.

11. CÁC THÀNH PHẦN ĐƯỢC NỘP

Bản kế hoạch quản lý dự án, lịch biểu của dự án, kế hoạch quản lý cấu hình, các tiêu chuẩn cài đặt mã Java, danh sách kiểm tra để xem xét lại mã, danh sách kiểm tra để xem xét lại kế hoạch tích hợp, danh sách kiểm tra để phân tích tác động, các báo

Sau khi dự án ACIC hoàn thành, đội ngũ quản lý đã rút ra được **3 bài học "xương máu"** rất thực tế như sau:

1. Giá trị của việc phòng ngừa lỗi (Defect Prevention)

- Dữ liệu thu thập từ các giai đoạn đầu của dự án cực kỳ hữu ích để cải thiện và phòng ngừa lỗi cho các giai đoạn sau.
- Việc chủ động phòng ngừa giúp **giảm đáng kể tỷ lệ tiềm lỗi** vào hệ thống. Lỗi sinh ra ít hơn đồng nghĩa với việc dự án tiết kiệm được rất nhiều nỗ lực và thời gian phải bỏ ra để sửa chữa (làm lại).

2. Khôn khéo khi đối mặt với Thay đổi yêu cầu

- Khi khách hàng yêu cầu một thay đổi có ảnh hưởng lớn, đội dự án không nên cảm đầu vào làm ngay.
- Cần lập một bản **phân tích tác động và chi phí - lợi ích** thật chi tiết để ngồi lại đàm phán với khách hàng. Thực tế ở dự án ACIC, việc phân tích chỉ ra rủi ro rõ ràng đã thuyết phục được khách hàng đồng ý **trì hoãn yêu cầu thay đổi** đó, giúp dự án không bị vỡ kế hoạch.

3. Đừng ỷ lại vào khâu Kiểm thử ở cuối (Bài học về Chất lượng)

- Trong dự án ACIC, khâu bắt lỗi sớm là *Xem xét lại mã nguồn (Code Review)* và *Kiểm thử đơn vị (Unit Test)* hoạt động kém hiệu quả. Rất may là khâu *Kiểm thử hệ thống (System Test)* ở cuối dự án đã phát hiện bù lại được lượng lỗi bị sót.
- Bài học rút ra:** Đối với các dự án lớn, việc ỷ lại, dồn rủi ro xuống giai đoạn kiểm thử hệ thống ở cuối là vô cùng nguy hiểm và có thể làm hỏng chất lượng dự án. Bắt buộc phải rà soát và nâng cao năng lực cho các khâu review và kiểm thử đơn vị ngay từ đầu.

(Mẹo đi thi: Bạn chỉ cần nhớ 3 từ khóa: **Phòng ngừa lỗi** → **Phân tích chi phí khi đổi yêu cầu** → **Không ỷ lại vào kiểm thử hệ thống** là sẽ viết trôi chảy toàn bộ câu này!)