

CHƯƠNG 5: QUẢN LÝ CÁC GIAI ĐOẠN XÂY DỰNG HTTT

Chương này trình bày các nội dung trọng tâm:

- Chương này trình bày tóm tắt một số vấn đề về *những việc phải làm trong các giai đoạn xây dựng hệ thống thông tin* và về tổ chức, kinh nghiệm khi thực hiện một dự án tin học.

- Cung cấp một số kiến thức được chắt lọc từ bộ môn Quản lý dự án phần mềm có liên quan đến kiến thức môn Phân tích thiết kế hệ thống thông tin nhằm hỗ trợ tốt hơn khi học sinh, sinh viên tốt nghiệp và công tác tại đơn vị phát triển phần mềm.

5.1 QUẢN LÝ GIAI ĐOẠN LẬP KẾ HOẠCH, XÁC ĐỊNH YÊU CẦU

Về các mục tiêu dự án cần đạt được: Phân tích những mục tiêu phải đạt được của dự án được thực hiện dựa trên sơ đồ hệ thống các vấn đề đã được lập trước đó, gồm các việc:

- Mô tả tình hình trong tương lai một khi mọi vấn đề đã được giải quyết;
- Những trạng thái mong muốn và khả thi của các trạng thái có vấn đề trước đây.
- Lập sơ đồ phân cấp hệ thống các mục đích.

Sau các bước trên, mục đích trọng tâm cũng như các mục đích ở các mức cao hơn của dự án đã bước đầu được xác định.

Về các giải pháp: Các giải pháp thay thế hiện trạng được xác lập dựa trên sơ đồ hệ thống các vấn đề và sơ đồ hệ thống các mục đích của dự án vừa xây dựng nên. Việc lựa chọn ra giải pháp tối ưu được thực hiện bằng cách so sánh chúng về một số phương diện.

Về các yếu tố có thể gây rủi ro cho dự án: Phân tích các yếu tố nhạy cảm không thể giúp ta tránh được hay làm yếu đi các yếu tố tương lai chưa biết cũng như các yếu tố có thể gây rủi ro đối với dự án. Nhưng nó cho phép xác định (bằng tính toán) sự thay đổi của các chỉ số hiệu quả kinh tế hay các chỉ số sinh lợi của dự án gây ra bởi sự thay đổi của một hay một vài thông số liên quan. Đó là những yếu tố có thể gây rủi ro cho dự án. Tất nhiên, các thông số đầu vào được lựa chọn để khảo sát phải là các thông số mà giá trị của chúng có khả năng thay đổi thực sự trong thực tế. Mặt khác, việc phân tích tác động của cùng một yếu tố rủi ro đối với cùng một chỉ số kinh tế của các giải pháp khác nhau sẽ cho phép ta chọn ra được một giải pháp ít rủi ro nhất.

Cách thức quản trị dự án hệ thống thông tin Triển khai dự án hệ thống thông tin (HTTT) trong doanh nghiệp là một quá trình rất phức tạp, lôi cuốn nhiều người tham gia và huy động nhiều nguồn lực, phải xử lý nhiều tình huống phát sinh...

Nói chung, quản trị dự án bao gồm các quá trình: Lập kế hoạch; Giám sát việc thực hiện dự án; đánh giá dự án. Trong đó: Lập kế hoạch dự án gồm 2 công đoạn chính: Phân tích/Chuẩn bị lập kế hoạch và lập kế hoạch chi tiết.

Các yếu tố quan trọng khi lập kế hoạch dự án: Có 5 yếu tố cần quan tâm, đó là: *Con người, vấn đề của dự án, mục tiêu dự án, các giải pháp và các yếu tố rủi ro.*

Về con người: Con người luôn là yếu tố rất quan trọng của dự án. Các đối tượng con người ở đây được hiểu là rất cả các cá nhân, nhóm người hay tổ chức... có liên quan dưới mọi hình thức tới dự án. Để phân tích các đối tượng này nên được phân chia thành các nhóm tùy theo đặc điểm mối liên quan của họ với dự án. Phải làm rõ sự phân nhóm này và mối liên quan của các nhóm cũng như cá nhân đối với dự án.

Về các vấn đề của dự án: Bất cứ dự án nào được thực hiện cũng nhằm giải quyết một (hay nhiều) vấn đề đang tồn tại của một thực trạng nào đó. Phân tích vấn đề nhằm đánh giá được đúng đắn và đầy đủ các vấn đề của dự án, đặt trong mối quan hệ nhân quả với tình hình thực tại, để xác định được chính xác mục tiêu mà dự án định đạt tới cũng như giải pháp thực hiện. Mặt khác, trong phân tích vấn đề cũng cần làm rõ các hạn chế về thời gian, nhân lực và ngân sách thực hiện dự án. Những dự án nhỏ (số người sản xuất từ 6 đến ít hơn 20) thì những dự án phần mềm được tổ chức tốt nhất là thành những các đội phát triển nhỏ. Qui mô lý tưởng của một đội phát triển là khoảng bốn đến sáu người sản xuất. Hầu hết các dự án được tổ chức thành đội ngũ với mỗi đội ngũ được giao phó những chức năng đặc thù trong phạm vi dự án. Các loại dự án khác nhau đòi hỏi những loại cơ cấu đội ngũ khác nhau như chẳng hạn đội ngũ các nhà lập trình trung cấp đòi hỏi lãnh đạo đội ngũ kỹ thuật trong khi đội ngũ chuyên gia có thể chỉ đòi hỏi lãnh đạo đội ngũ hành chính. Đó là trách nhiệm của người quản lý dự án trong việc chọn lựa cơ cấu thích ứng nhất cho dự án.

Giám sát và Đánh giá dự án: Bao gồm xác định phương pháp đánh giá (hiệu quả kinh tế, kết quả thực hiện so với kế hoạch...) và tổ chức công việc đánh giá. Giám sát việc thực hiện dự án là một khâu rất quan trọng của quá trình quản trị dự án, có ảnh hưởng và tác dụng trực tiếp nhất tới sự thành công của dự án. Vì vậy, nó đòi hỏi sự tham gia không chỉ của những người ở cương vị quản lý, mà còn của những người thực hiện công việc trong dự án, thậm chí có thể của tất cả mọi thành viên tham gia thực hiện dự án.

Ba điểm mấu chốt nhất để công việc giám sát dự án thực hiện được hiệu quả là:

- Thông nhất được phương thức thực hiện công việc giám sát và trao đổi thông tin thích hợp và thực tế, sao cho mọi thành viên tham gia công việc này có thể thực hiện được dễ dàng, nhanh chóng và chính xác nhất.
- Phát hiện được càng sớm càng tốt sai lệch so với kế hoạch của những nhiệm vụ quan trọng, chủ chốt và tìm ra đúng các nguyên nhân của những sai lệch đó.
- Có biện pháp điều chỉnh thích hợp và khả thi để đảm bảo đạt được các mục tiêu của dự án.

Còn việc đánh giá dự án là nhằm đánh giá một cách định lượng về hiệu quả của dự án (được quy ra hiệu quả kinh tế); Mức độ thành công/Thất bại của dự án.

Quá trình đánh giá dự án gồm:

- Xác định các chỉ tiêu và chỉ số cần đánh giá;
- Xác định các phương pháp đánh giá sẽ được sử dụng;
- Kiểm tra nguồn cung cấp các dữ liệu đầu vào cần thiết;
- Thu thập dữ liệu và thực hiện tính toán;
- Đánh giá kết quả thu được và rút ra kết luận cần thiết;
- Bổ sung thêm các chỉ số đánh giá khác (nếu cần).

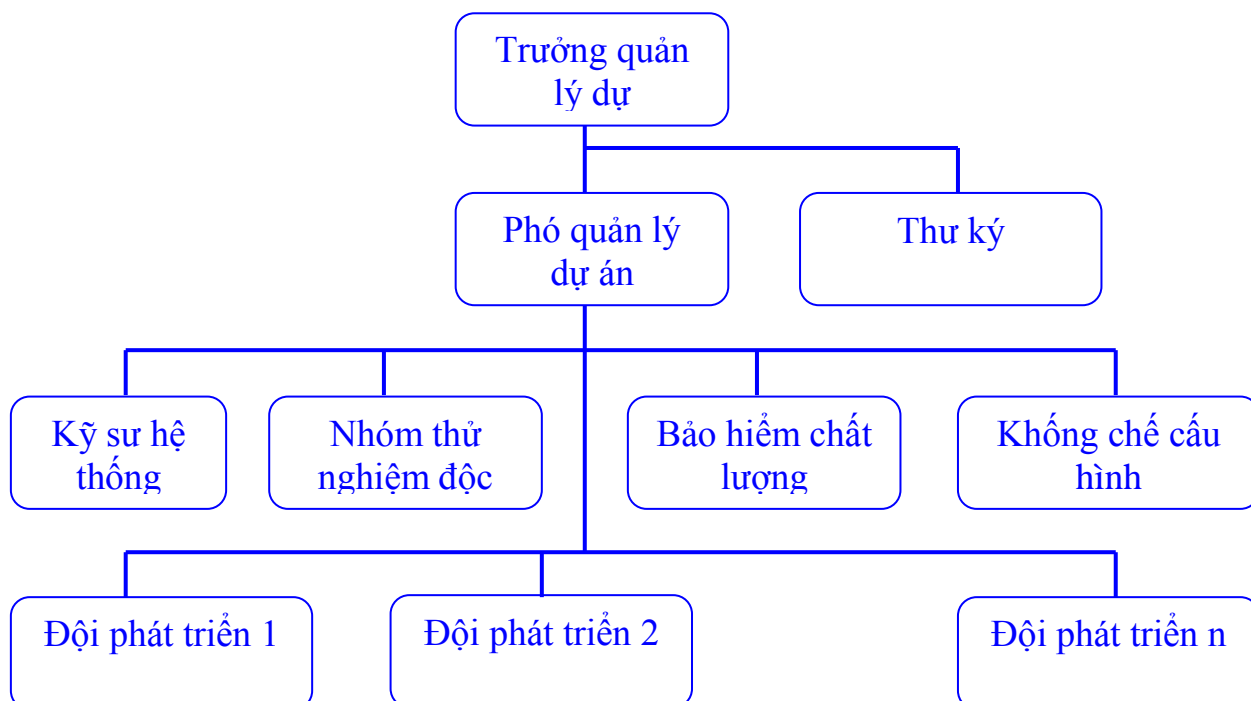
Các phương pháp đánh giá dự án chủ yếu dựa trên các chỉ tiêu kinh tế. Bốn phương pháp thường dùng là:

- Phân tích chi phí - lợi nhuận;
- Tương quan thu - chi;
- Giá trị tư bản;
- Lãi suất (tỉ suất thu hồi) nội tại;

Nội dung và trình tự tiến hành ở trên là tổng quát cho đánh giá dự án nói chung. Tuy nhiên, tùy theo yêu cầu cụ thể và đặc trưng của từng dự án có thể chỉ cần tiến hành một phần trong số các nội dung đó. Ngoài ra, cũng có thể tiến hành công việc đánh giá tại từng giai đoạn nhất định của dự án

5.1.1 Xây dựng nhóm

Phần này đề cập đến những vấn đề đó về quản lý con người chủ yếu những vấn đề liên quan đến dự án phần mềm. Về thảo luận khái quát hơn các phương pháp quản lý và động viên con người theo mô hình tổ chức đề xuất dưới đây:



Hình 5.1: Mô hình cơ cấu tổ chức để triển khai một dự án tin học.

Những tổ chức khối tháp như trên tạo nên một trật tự rõ ràng xác định cụ thể trong đó mọi cá nhân biết được cương vị của chính mình và cương vị của những người trên và dưới họ. Khi đề bạt và qui chế đóng vai trò chủ yếu trong hình thành động cơ. Do đó tổ chức khối tháp là có hiệu quả nhất. Nhiều những yếu tố khác

hình thành động cơ ý thức hoàn thành khen ngợi và quý trọng. Mặc dù đề bạt và qui chế không phải luôn luôn là động cơ hiệu quả nhất, người quản lý dự án rất hiếm khi từ bỏ bất cứ công cụ quản lý nào nhưng tổ chức khối tháp thường là tốt nhất.

Cơ cấu đội ngũ: Những đội ngũ lớn hạn chế khả năng của lãnh đạo đội được hoạt động là người sản xuất và do đó tăng tổng phí quản lý và hạn chế dính líu kỹ thuật của lãnh đạo đội vào dự án Các đội ngũ đem lại cho người quản lý dự án nhiều lợi điểm, bao gồm :

- Quản lý dễ dàng và tốt hơn : Cơ cấu đội ngũ hỗ trợ việc uỷ nhiệm thẩm quyền.
- Trao đổi thông tin và ý kiến hiệu quả hơn do làm quen được rộng hơn trong đội ngũ với nhiệm vụ của mỗi thành viên.

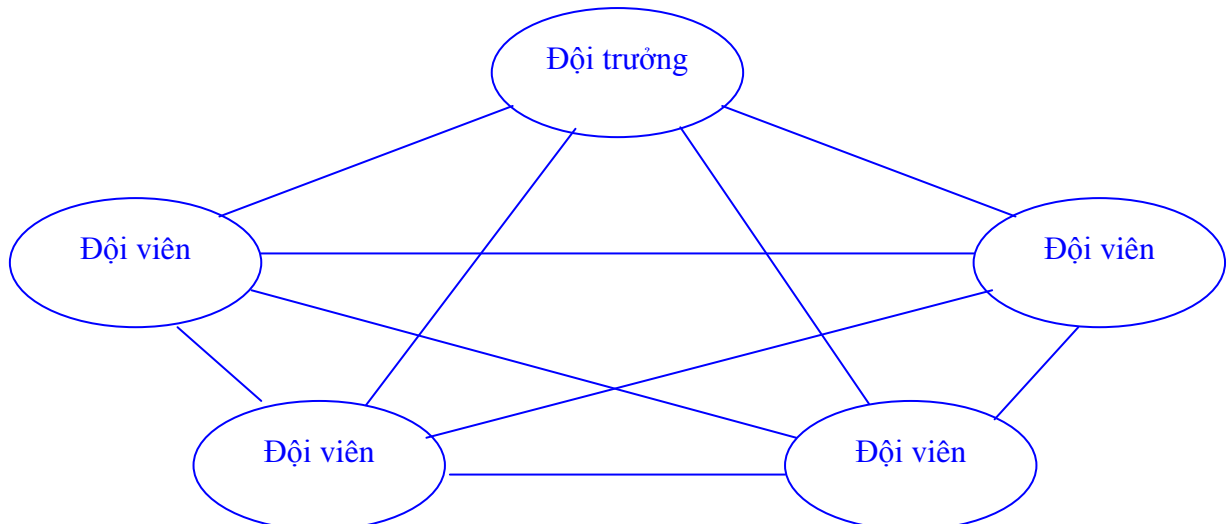
Trong các dự án nhỏ, có sự đồng nhất mạnh hơn với dự án lớn, ở những dự án lớn những người sản xuất có xu hướng cảm thấy họ chính là một trong số rất nhiều người mà đóng góp của họ cho dự án đang tiến hành không ai hay. Dự án nhỏ thì điều đó là ngược lại do đó đội ngũ gắn bó hơn, tận tụy hơn.

Lãnh đạo đội: Lãnh đạo đội ngũ được coi là kênh thông tin chính giữa người quản lý dự án với các thành viên đội. Điều này không có nghĩa là không có thông tin trực tiếp giữa người quản lý cấp trên của dự án với các thành viên đội. Dù sao nếu mọi thông tin là trực tiếp thì như thế điều này hẳn làm cho những mục đích chính của cơ cấu đội bị thất bại: Uỷ nhiệm thẩm quyền và trách nhiệm được hiệu quả.

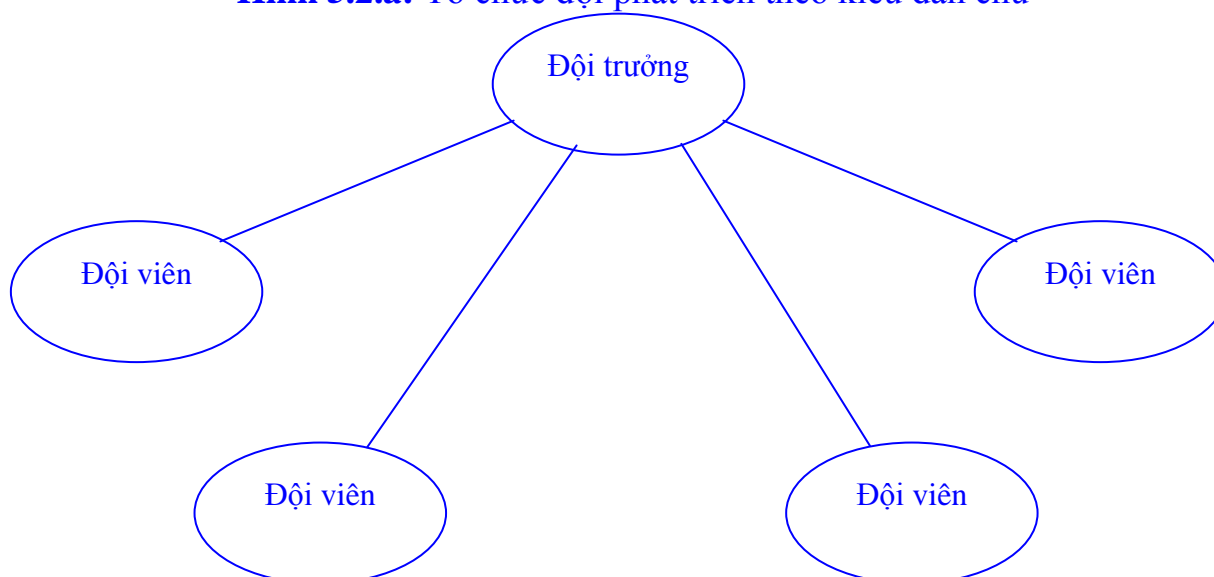
Vai trò của lãnh đạo đội là :

- Đại diện quản lý dự án thông qua uỷ nhiệm thẩm quyền
- Đại diện đội trước người quản lý dự án
- Đại diện đội trước các đội dự án khác và các chức năng tổ chức.

Người lãnh đạo đội cũng có thể có những trách nhiệm khác, tùy theo thể loại cơ cấu đội Hình 5.2 cho những thí dụ hai cơ cấu đội khác nhau (*những cơ cấu đội này được bàn sau*). Mọi lãnh đạo đội lãnh đạo hành chính không phải tất cả lãnh đạo đội lãnh đạo kỹ thuật đội ngũ kỹ sư trưởng đòi hỏi khả năng lãnh đạo kỹ thuật theo đó đòi hỏi thành viên kỹ thuật. Cả đội ngũ dân chủ và đội ngũ kỹ sư trưởng đòi hỏi khả năng lãnh đạo đội về hành chính.



Hình 5.2.a: Tổ chức đội phát triển theo kiểu dân chủ



Hình 5.2.b: Tổ chức đội phát triển theo kiểu kỹ sư trưởng

Các đội dân chủ: Nghiêm túc mà nói, các đội dân chủ không có lãnh đạo, chức năng của vai trò lãnh đạo đội là điều phối viên nhiều hơn, ở các đội dân chủ, các lãnh đạo đội giành một phần nhỏ thời gian của họ cho những việc sau:

- Việc đại diện đội trong thông tin với quản lý dự án và các đội khác.
- Phối hợp hoạt động trong đội
- Sử lý các nhiệm vụ hành chính khác như báo cáo, lập trình và giám sát hoạt động.

Mọi quyết định kỹ thuật trong một đội dân chủ được cả đội thực hiện lãnh đạo đội triệu tập các cuộc họp trong đó những vấn đề gay cần và cấp thiết được đưa ra thảo luận. Nhưng mọi thành viên đội tham gia trong quá trình ra quyết định và chịu trách nhiệm về đầu ra. Các đội dân chủ thường thích ứng cho các nhóm nhà sản xuất thâm niên có kinh nghiệm. Theo đó vai trò của lãnh đạo đội giảm tổng phí hành chính bằng cách giao nhiệm vụ hành chính của đội cho một thành viên duy nhất. Cơ cấu đội dân chủ đặc biệt không thích hợp cho các nhóm hỗn hợp, hay nhóm bao gồm chủ yếu những người sản xuất thanh niên. Trong cả hai trường hợp đó, vai trò lãnh đạo rõ ràng là cần thiết.

Các đội kỹ sư trưởng: Đội kỹ sư trưởng (cũng gọi là đội các lập trình) tiến hành lãnh đạo đội phát triển. Vai trò của lãnh đạo đội vừa là điều phối viên (như trong trường hợp đội dân chủ) vừa là hướng dẫn trong những dự án phức tạp, lãnh đạo đội có thể được yêu cầu giành đến 50% thời gian vào những hoạt động kỹ thuật và hành chính.

Hoạt động chính của lãnh đạo đội kỹ sư trưởng là:

- Giao nhiệm vụ và trách nhiệm cho các thành viên đội
- Giám định công việc của các thành viên đội.
- Góp ý và hướng dẫn các thành viên đội.
- Hoạt động hành chính và phối hợp (tương tự như lãnh đạo đội dân chủ).

Các đội kỹ sư trưởng là thích ứng cho đội hỗn hợp và đội chủ yếu có những người sản xuất có thâm niên hay không có kinh nghiệm. Chức năng của lãnh đạo đội với tư cách người quản lý hàng đầu và do đó phải được đào tạo thích hợp về kỹ thuật quản lý cơ bản. Các đội kỹ sư trưởng cũng có thể thành công trong những đội người sản xuất đàn anh và có kinh nghiệm nhưng vai trò thường không cần thiết khi cơ cấu này được vận dụng cho một đội các kỹ sư có kinh nghiệm thì việc người lãnh đạo đội có những kỹ năng quản lý cơ bản lại quan trọng gấp hai nếu không đụng chạm có thể phát triển giữa thành viên đội và người lãnh đạo đội. Về mặt này, lãnh đạo đội ngũ người chuyên môn có kinh nghiệm lại khó hơn là lãnh đạo đội ngũ người mới vào nghề.

Các đội chuyên gia: Đội ngũ chuyên gia là những đội ngũ nhỏ được thành lập để giải quyết những vấn đề đặc thù trong một dự án. Một đội ngũ chuyên gia có thể được thành lập trong quá trình phát triển dự án khi có vấn đề phức tạp nảy sinh và sau đó đội có thể được giải tán khi vấn đề được giải quyết (*thường thì những dự án gia công phần mềm người ta hay áp dụng mô hình này*). Trong một số trường hợp, các đội chuyên gia có thể hỗ trợ dự án suốt chu kỳ phát triển. Mục tiêu của đội chuyên gia là tập trung giám định trong một lĩnh vực riêng của dự án.

Lấy thí dụ, ta hãy xét một hệ thống thủ quỹ ngân hàng tự động có hai hệ thống phụ chủ yếu : *Computer trung tâm của ngân hàng và thủ quỹ tự động từ xa. Kế hoạch phát triển cho hai hệ thống phụ đó đã giao việc phát triển của mỗi hệ phụ cho một đội riêng nhưng qui mô của dự án không thể đảm bảo có một đội hợp nhất riêng. Cả hai đội đã sử dụng bộ mô phỏng thực hiện thử nghiệm và hợp nhất hệ thống phụ ban đầu. Kết quả là khi hai hệ thống phụ được hợp nhất với nhau, thông tin giữa hai hệ thống không có.*

Trong những tình huống như thế bao giờ cũng có nguy cơ là một trong hai đội tìm kiếm vấn đề trong công việc của đội kia. Cho dù hai đội hợp tác với nhau tốt, khác biệt trong thực hiện (*hay trong thiết kế*) có thể làm cho vấn đề khó được giải quyết và vì kịch trình bắt đầu trở nên phức tạp, điều này trở thành mối quan ngại chủ yếu cho người quản lý dự án.

Với những trường hợp như thế, người quản lý dự án nên quyết định thành lập đội chuyên gia để giải quyết vấn đề thông tin. Đội dự án và hai kỹ sư dự án chung hai đội. Sau đó đội chuyên gia có thể tập trung cố gắng của mình vào việc giải quyết vấn đề càng nhanh càng tốt. Trong khi hai đội kia tiếp tục các hoạt động phát triển khác. Sau khi vấn đề thông tin giữa hai hệ thông phụ đã được giải quyết, đội chuyên gia thể giải tán về một số mặt, các đội thử nghiệm độc lập và đội bảo hiểm chất lượng được coi là đội chuyên gia. Các đội thử nghiệm độc lập lúc đầu hoạt động trong các giai đoạn hợp nhất và thử nghiệm của chu kỳ phát triển. Đội bảo hiểm chất lượng là một thí dụ về đội chuyên gia hoạt động xuyên suốt chu kỳ phát triển của dự án.

Đội chuyên gia thường có những kỹ sư có kinh nghiệm cao trong những trường hợp như thế điều chắc chắn nhất là đội được tổ chức như là đội dân chủ. Trong thí dụ trước, đội có thể là đội dân chủ hay có thể do một chuyên gia thông tin đàn anh lãnh đạo.

5.1.2 Bố trí nhân sự, phân công nhiệm vụ

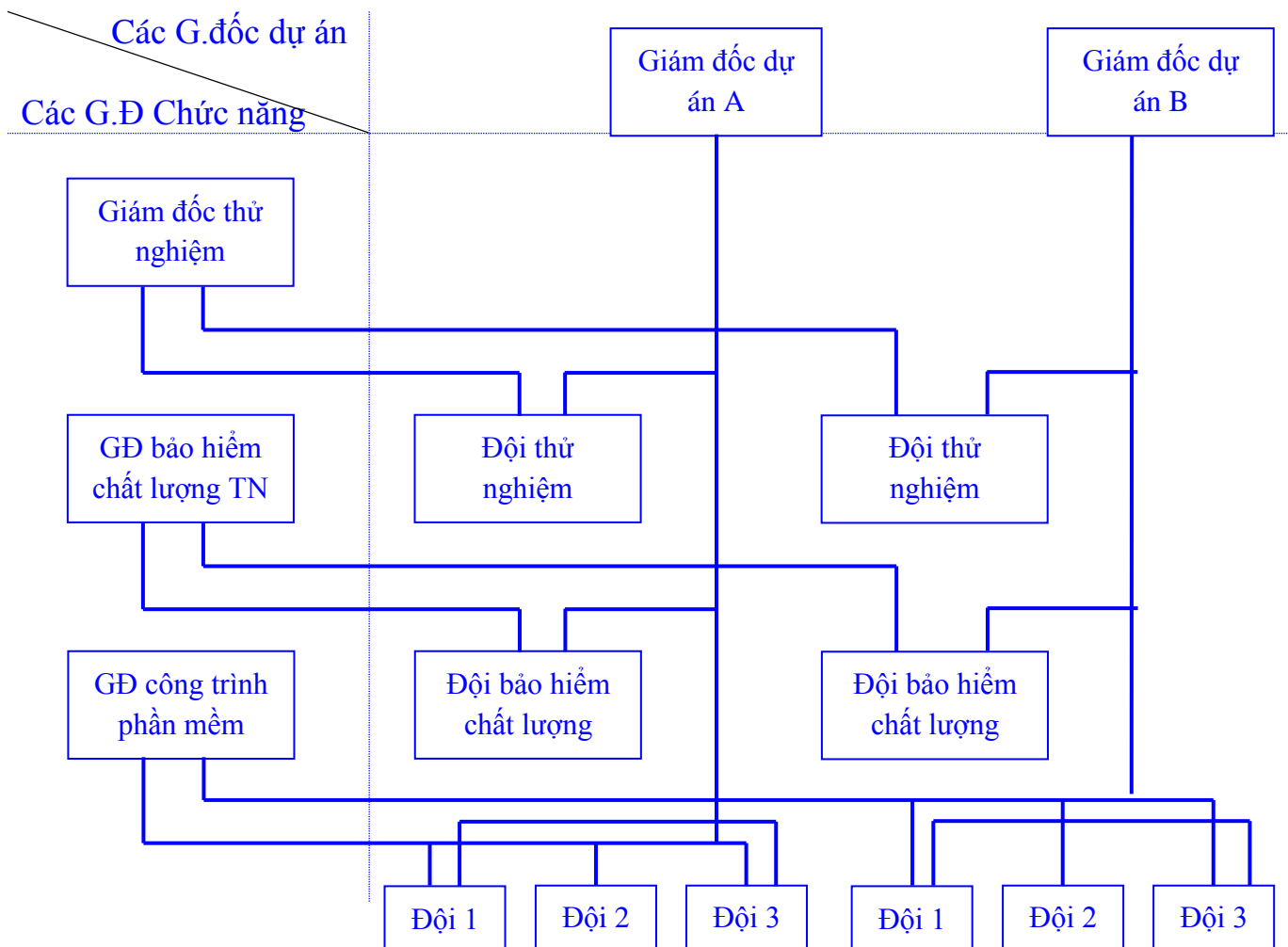
Rõ ràng cơ cấu tổ chức của dự án tùy thuộc ở loại dự án được phát triển. Một số vấn đề phải xem xét là :

Qui mô dự án : Dự án càng lớn, tổ chức mô hình càng quan trọng. Những dự án lớn có tổng phí thông tin và phối hợp đáng kể về con người do đó đòi hỏi nhiều chức năng hỗ trợ hơn.

Dự án phát triển phần mềm/phần cứng: Việc phát triển liên tiếp phần cứng và phần mềm không dễ dàng. Việc lập kế hoạch, hợp nhất và thử nghiệm lại càng phức tạp hơn và đòi hỏi những nhóm hỗ trợ tận tụy.

Các hệ thống tin cậy cao: Bất cứ hệ thống nào nhạy cảm với các vấn đề độ tin (như hệ thống quân sự hay cứu nạn) đòi hỏi cố gắng chủ yếu về bảo hiểm chất lượng. Chất lượng cũng là một nhân định quan trọng trong nhiều sản phẩm phần mềm đưa ra thị trường (thí dụ bộ chương trình thông tin). Những loại dự án đó đòi hỏi một tổ chức bảo hiểm chất lượng riêng.

Cơ cấu tập đoàn: Tổ chức dự án phụ thuộc rất nhiều vào cơ cấu tổng thể của công ty trong đó dự án được phát triển. Nhiều những chức năng hỗ trợ dự án có thể được các nhóm tập trung hoá trong công ty cung cấp. Trên thực tế, những dịch vụ cơ bản như dịch vụ tài chính văn phòng pháp lý thường do tổ chức bà con hay của tập đoàn cung cấp.

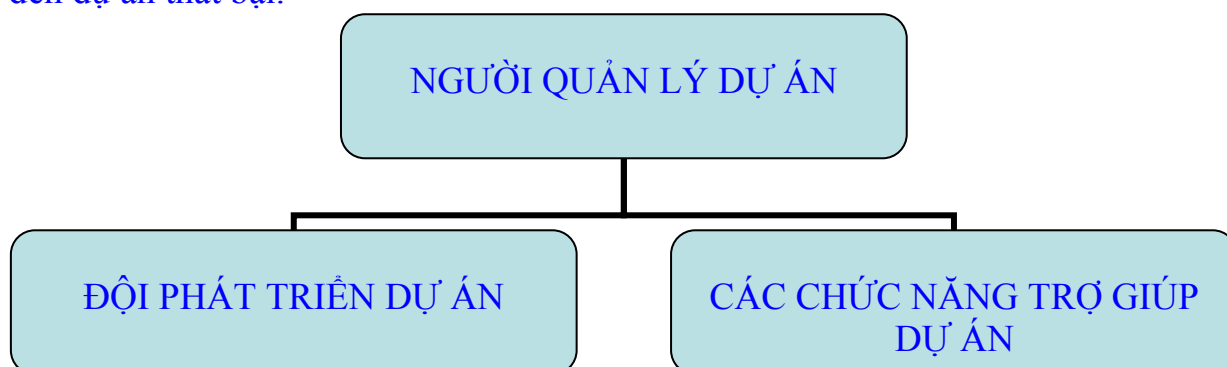


Hình 5.3: Biểu đồ tổ chức

Cơ cấu tập đoàn thường định ra một trong hai loại tổ chức dự án cơ bản: Ma trận hay khối tháp. Hình 5.3 mô tả cơ cấu của một tổ chức ma trận (so sánh với cơ

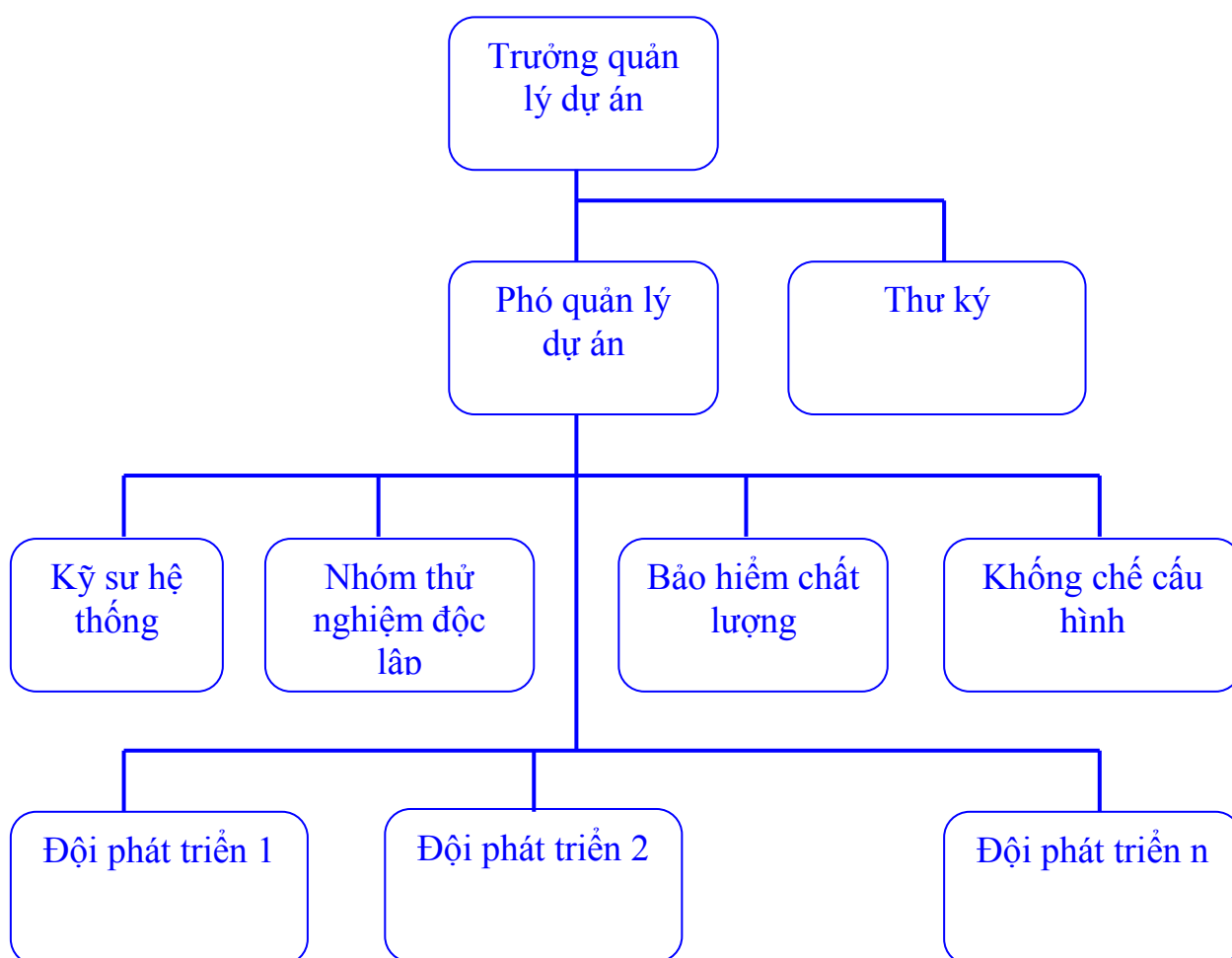
cấu khối tháp trong hình 5.1 hoặc sơ đồ rút gọn hình 5.4). Trong một tổ chức ma trận tập đoàn, người quản lý dự án quản lý những hoạt động kỹ thuật của đội ngũ dự án trong khi dính líu của anh hay chị ấy trong những vấn đề nhân sự phi kỹ thuật (thí dụ duyệt lương, đề bạt, đào tạo) thì rất ít.

Có nhiều cách tổ chức dự án phần mềm. Dự án càng lớn thì cơ cấu tổ chức càng trở nên gay gắt hơn. Những dự án tổ chức tồi gieo rắc lộn xộn và lộn xộn dẫn đến dự án thất bại.



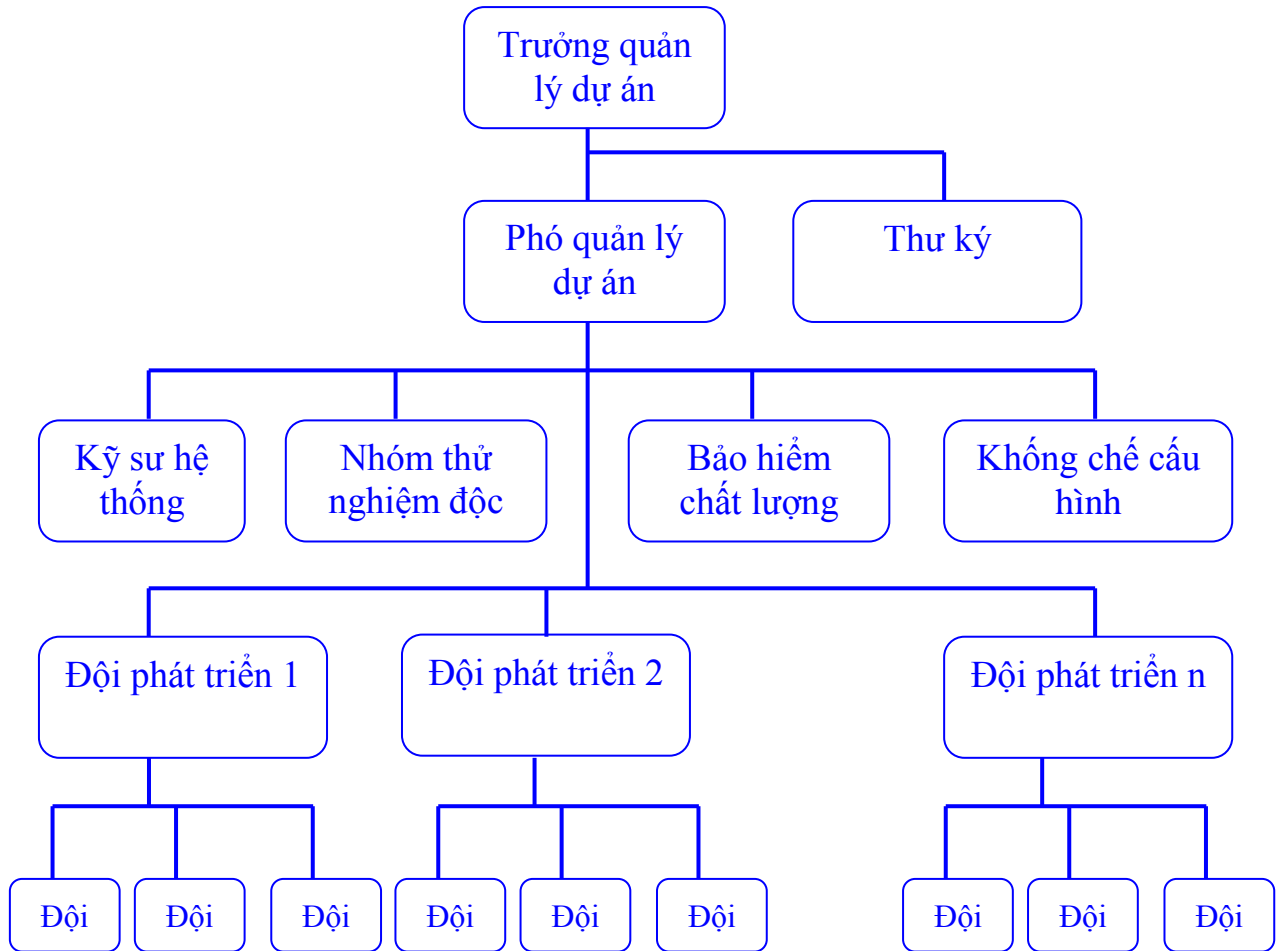
Hình 5.4: Biểu đồ tổ chức dự án lớn phần mềm/ phần cứng

Hình 5.5 mô tả cơ cấu cơ bản của dự án trong đó bên dưới người quản lý dự án đúng là có 2 chức năng tổng quát phát triển và hỗ trợ.



Hình 5.5: Biểu đồ tổ chức dự án phần mềm

Hình 5.6 Mô tả biểu đồ tổ chức chi tiết kể cả những chức năng hỗ trợ chủ yếu. Cơ cấu tổ chức này thích ứng với những dự án lớn (với nhân sự trên 20). Những dự án nhỏ có thể không đòi hỏi phó quản lý dự án hay các nhóm kiểm tra cấu hình và bảo hiểm chất lượng riêng biệt. Những dự án rất lớn (nhân sự từ 40 người trở lên) thường có thể được quản lý dễ dàng hơn khi phân dự án thành những dự án phụ.



Hình 5.6: Biểu đồ tổ chức dự án phần mềm/phần cứng lớn

Bên cạnh việc phát triển một dự án phần mềm còn có quá trình qui hoạch, tổ chức, tuyển nhân sự, giám sát, kiểm tra và lãnh đạo dự án phần mềm. Hiếm khi mọi nhiệm vụ đều được quản lý một cách chặt chẽ. Trên thực tế nhiều hoạt động kiểm tra và giám sát đều có thể được ủy thác cho các nhóm hỗ trợ dự án. Những nhóm hỗ trợ này không chỉ giám định cho quản lý dự án và kỹ sư phát triển bằng những nhiệm vụ hỗ trợ. Họ cũng thực hiện những nhiệm vụ đó tốt hơn bằng cách tập trung mọi cố gắng của họ vào những chức năng hỗ trợ đặc trưng. Có nhiều loại chức năng hỗ trợ dự án.

Dịch vụ thư ký, hỗ trợ hành chính xuất bản và cung cấp tài liệu là những thí dụ về chức năng hỗ trợ không kỹ thuật;

Thử nghiệm, kiểm tra cấu hình, công nghệ hệ thống quản lý hội nhập và bảo hiểm chất lượng là những thí dụ về chức năng hỗ trợ kỹ thuật.

Dự án càng lớn và càng phức tạp lại sẽ đòi hỏi chức năng hỗ trợ nhiều hơn. Chẳng hạn, một dự án lớn thường có tổ chức kiểm tra chất lượng của nó trong khi một dự án nhỏ có thể chia sẻ chức năng đó với các dự án khác. Tương tự, nhiều tổ chức duy trì nhóm thử nghiệm độc lập mà vai trò là thử nghiệm một sản phẩm phần

mềm trước khi đưa ra ở những dự án lớn, nhóm thử nghiệm độc lập là một bộ phận của đội dự án và tham gia trong thử nghiệm và qui hoạch thử nghiệm xuyên suốt chu trình phát triển.

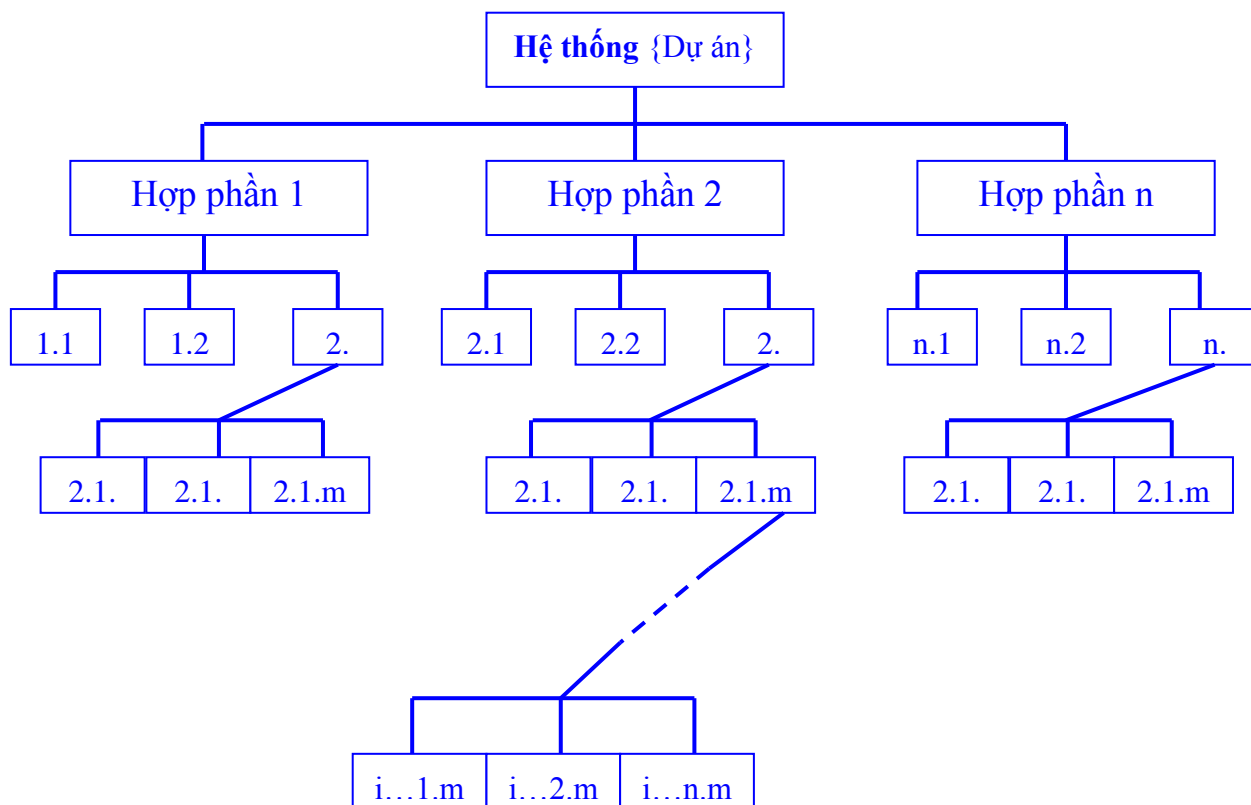
5.1.3 Lập kế hoạch tiếp cận vấn đề

Nhiều đối tượng phức tạp có thể được xem là một bộ vô vàn đối tượng đơn giản hơn hợp thành. Một thí dụ thích hợp là một hoá chất hình thành từ nhiều phân tử khác nhau mỗi phân tử được tạo thành khi phối hợp nhiều nguyên tử khác nhau. Các nguyên tử mặc dầu bản thân phân chia được có thể được coi là phân tử nhỏ nhất của một hoá chất. Theo cách đó, tương tự cho các dự án phức tạp có thể được phân thành những thành phần đơn giản hơn. Trong khi toàn bộ dự án có thể khó quản lý thì mỗi thành phần sẽ dễ xử lý hơn những dự án phần mềm có thể được phân thành những thành phần nhỏ hơn nhằm dự tính tốt hơn về khối lượng công việc hoặc nhằm điều khiển các hoạt động của các đội phát triển khác nhau.

Việc phân giải dự án phần mềm là một trong những nhiệm vụ đầu tiên của người quản lý dự án phần mềm. Các phương pháp phân giải có thể khác nhau, tùy theo mục tiêu thực sự của người quản lý dự án. Việc phân tích dự án theo chức năng có thể không như là phân tích thiết kế. Phân tích chức năng chia dự án thành những thành phần cơ bản của nó theo cách nhìn của người dùng trong khi phân tích thiết kế chia dự án thành thành phần hay mô đun lập trình cơ bản. Nói tóm lại Phương pháp chính là “**Chia để trị**”.

Tinh chế từng bước một:

Theo trực giác có vẻ không hợp lý khi tìm cách minh định mọi thành phần dự án trong một bước duy nhất. Rõ ràng một qui trình lập có thể dần dần cung cấp nhiều chi tiết hơn, hẳn là dễ sử dụng hơn. Những phương pháp lập thuộc loại này được gọi là tinh chế từng bước một vì sự phân giải được tiếp tục tinh chế ở mỗi bước kế tục



Hình 5.7: Phân giải phần mềm bằng tinh lọc từng bước

Hình 5.7 trình bày minh họa tổng quát về tinh chế từng bước một. Hệ thống ban đầu được phân thành ba thành phần cấp cao và rồi mỗi thành phần cấp cao lại được phân tiếp thành các thành phần cấp thấp hơn, và cứ thế cho đến khi đạt được cấp thành phần thấp nhất.

Trong phân giải dự án từng bước, mỗi thành phần phân giải thành những thành phần ở trực tiếp dưới nó sao cho mỗi bước phân giải đều mô tả toàn bộ hệ thống nhưng ở mức chi tiết khác trong hình 5.5 các hợp phần 1,2 và 3 tạo nên hệ thống hoàn chỉnh. Để được chi tiết hơn, chúng ta lấy bước phân giải tiếp và thấy các thành phần 1.1,1.2,2.1,2.2,2.3,3.1 và 3.1 biểu thị toàn bộ hệ thống.

Sơ đồ tinh chế từng bước một tương tự như biểu đồ một hệ thống tôn ti trật tự. Dù sao điều quan trọng là phải hiểu rằng tinh chế từng bước là cơ bản khác nhau vì các khối cấu thành của biểu đồ khác nhau. Sơ đồ hệ thống tôn ti mô tả mối quan hệ tôn ti giữa các thành phần khiến cho mỗi thành phần trong sơ đồ tương ứng thực sự với một thành phần thực trong hệ thống. Dù sao sơ đồ tinh lọc từng bước, thành phần cấp cao hơn chỉ là tên đặt theo qui ước cho một nhóm các thành phần thực xuất hiện ở ngay dưới nó.

Phân giải chức năng:

Việc phân giải theo chức năng một dự án phần mềm là sự phân chia hệ thống thành những thành phần theo hoạt động của nó như người dùng nhận định. Việc phân giải theo chức năng là một bộ phận của pha yêu cầu của dự án. Mục tiêu của pha này là xác định mọi đặc điểm của hệ thống theo cách nhìn của người dùng. Ở các quyết định thực hiện có thể từng lúc được đưa ra trong giai đoạn yêu cầu và được coi là yêu cầu thực hiện. Điều này có thể bao gồm đặc điểm như là loại máy tính đối tượng, ngôn ngữ lập trình được sử dụng hay phương pháp truyền thông

được dùng tốt hơn hết là nên hoãn, các quyết định thực hiện càng chậm càng hay cho đến pha thiết kế.

Phân giải thiết kế:

Phân giải thiết kế của một hệ thống phần mềm là phân chia hệ thống thành những thành phần cấp thấp khớp với thành phần phần mềm thực sự của hệ thống. Trong phân giải thiết kế hoàn toàn của một hệ thống phần mềm, các thành phần thấp nhất tương ứng với các mô đun lập trình (*thông thường là các thủ tục, chương trình con hay chức năng chương trình*).

Đúng như pha yêu cầu đi trước giai đoạn thiết kế việc phân giải chức năng của một hệ thống phần mềm thông thường đi trước việc phân giải thiết kế. Phân giải chức năng thường cung cấp nhiều thông tin cần thiết cho việc phân chia tiếp hệ thống đó thành những thành phần thực hiện.

Trên thực tế, phân giải chức năng thường là nơi tốt để khởi sự thiết kế một hệ thống phần mềm vì thành phần chức năng chủ yếu của một hệ thống thường tương ứng với phân chia ban đầu hệ thống đó thành những hệ thống con hay thành phần mức cao. Về cơ bản phân giải thiết kế tạo ra hai loại thành phần hệ thống thành phần cấp cao và mô đun cấp thấp hơn. Các tiêu chuẩn phát triển phần mềm khác nhau sử dụng thuật ngữ khác nhau để nhận biết các cấp phân giải khác nhau.

Một hệ thống phân giải hoàn toàn với mọi thành phần cấp thấp của nó, không phải bao giờ cũng dễ nắm bắt, điều này đặc biệt đúng trong sự trình bày hệ thống lúc duyệt dự án khi hệ thống cần được những người không dính líu vào việc thiết kế nó nhanh chóng hiểu được. Trong những trường hợp như thế, kỹ thuật tinh lọc từng bước là phương pháp thuận lợi để trình bày tuần tự chi tiết tăng dần bằng cách lúc đầu chọn thấy cấp phân giải đầu tiên và rồi chậm chậm phát hiện những cấp tiếp theo.

5.1.4 Xây dựng các mối quan hệ

Việc phát triển phần mềm có thể khống chế được. Có những phương pháp, những kỹ thuật, những tiêu chuẩn và các công cụ khi được vận dụng đúng thì chúng thúc đẩy việc phát triển thắng lợi dự án phần mềm với ba mục tiêu không thể quên để phát triển phần mềm.

- ***Theo đúng lịch trình.***
- ***Trong phạm vi ngân sách.***
- ***Theo yêu cầu.***

Như đã trình bày trong chương 2 thì ngay lúc đầu tiếp cận nhà phát triển chưa thể có ngay bản hợp đồng kinh tế được ký kết trong tay. Phía nhà phát triển dự án phải tự bỏ chi phí để khảo sát tạm thời nhằm kiểm tra hiện trạng, ít nhất là về mặt tổ chức, trang thiết bị và trình độ, cũng như khả năng sẵn sàng cho việc tin học hóa và cuối cùng là khả năng tài chính của đơn vị có nhu cầu tin học hóa. Rõ ràng để làm đúng ***Theo yêu cầu*** mà chỉ ***Trong phạm vi ngân sách*** thì cần phải thương thảo khá nhiều, điều đó cũng tức là cần xây dựng các mối quan hệ bao gồm:

Xây dựng các nguyên tắc làm việc bao gồm: Nguyên tắc giữa các đối tác; Nguyên tắc giữa cá nhân thuộc nhà phát triển với tổ chức hoặc đơn vị có nhu cầu và ngược lại; Nguyên tắc làm việc của các cá nhân của các phía.

Xây dựng kế hoạch làm việc giữa các bên: Thống nhất các cuộc hẹn làm việc trong tương lai gần; Phác thảo sơ bộ kế hoạch phát triển dự án và các cuộc hội thảo chia sẻ thông tin nếu cần.

Thắng lợi hay thất bại của dự án không chỉ liên quan đến ba mục tiêu phát triển cơ bản trên nhưng cũng liên quan cả đến kỳ vọng của khách hàng: Một khách hàng có thể hài lòng với đầu ra của một dự án trong khi khách hàng khác lại có thể không. Do vậy, thành công của dự án được xác định tối hậu ở sự hài lòng của khách hàng.

Một trong những trở ngại mà các nhà quản lý dự án thường phải khắc phục là thiếu những phương pháp phát triển hiện đại. Mọi lập luận chống lại các phương pháp luận phát triển mới chỉ có thể bác bỏ sau khi có sự chuẩn bị đầy đủ. Bên cạnh đó cũng phải trù liệu một số vấn đề tranh cãi là đặc biệt cho phát triển phần mềm. Quan tâm đúng mức đến các rủi ro, bởi vì phát triển phần mềm mang đến nhiều rủi ro hơn bất cứ các lĩnh vực khác của công nghệ.

Phí cộng thêm và giá cố định là hai trong số mối quan hệ theo hợp đồng cổ truyền giữa người phát triển và khách hàng. Có nhiều biến thức của hai mối quan hệ cơ bản đó kể cả các ghép nối phù hợp với các dự án đặc trưng.

Một số trong những quan hệ đó liên kết với vai trò của khách hàng và người phát triển nhằm tạo nhiều yếu tố kích thích hơn cho người phát triển hỗ trợ mục tiêu của khách hàng ngoài những trách nhiệm theo hợp đồng.

Những loại khác của quan hệ khách hàng - người phát triển bao gồm:

- ***Phối hợp giá cố định và phí cộng thêm***
- ***Liên doanh***
- ***Thỏa thuận bản quyền***
- ***Cam kết lâu dài***

Liên doanh là những trường hợp mà ranh giới giữa khách hàng và người phát triển có thể trở nên mờ ảo và phiên thay những thuận lợi và bất lợi thảo luận trước đây có thể không vận dụng được. Có nhiều trường hợp mà một số hình thức liên doanh có thể là mong muốn cho hai bên như khi người phát triển muốn duy trì quyền về sản phẩm, hay khi người phát triển chung sức với khách hàng tài trợ một phần của cố gắng phát triển.

Có một cách mà khách hàng có thể chào người phát triển tham gia vừa phải vào mặt kinh doanh của dự án là bằng cách thay thế bản quyền coi như thanh toán một phần. Điều này tạo nên qui mô bổ sung cho lợi ích của người phát triển trong thành công của dự án. Bản quyền thông thường là ở chỗ thất bại của dự án có thể tạo nên ít lợi nhuận cho người phát triển hơn là một hợp đồng giá cố định bằng thừng và thắng lợi của dự án sẽ làm tăng lợi nhuận của người phát triển.

Mối quan hệ lâu dài thường là quan trọng cho người phát triển. Trong nhiều trường hợp, những cam kết dài hạn cũng nằm trong lợi ích của khách hàng. Điều

này xảy ra khi người phát triển do gấn bó ở hợp đồng ban đầu, giành được lợi thế chủ yếu, qua kiến thức thu lượm, đối với những người khác cho công việc phát triển tiếp sau. Rõ ràng khi người phát triển hoàn thành thắng lợi một dự án lớn và phức tạp, anh ta có được một lợi thế đáng kể so với các công ty khác về các tăng cường trong tương lai của dự án đó. Cam kết lâu dài theo đó có thể có lợi ích tương hỗ cho cả hai bên trong đó khách hàng đảm bảo các dịch vụ sau này của người phát triển và người sản xuất đảm bảo cam kết thu nhập lâu dài.

5.1.5 Ký kết các thỏa thuận tài chính, kỹ thuật

Hợp đồng phí cộng thêm:

Phí cộng thêm là mối quan hệ theo hợp đồng theo đó người phát triển được trả cho chi phí dịch vụ đã làm và thêm vào đây được phép hưởng mức lời thỏa thuận. Điều này thực ra giống như cho thuê ô tô: khách hàng trả cho số thời gian xe được sử dụng (*theo giờ, ngày, tuần v.v...*) và cho mọi chi phí khác như bảo hiểm và xăng. Theo thế trong một hợp đồng phí cộng thêm, tổng phí của một dự án chỉ được biết sau khi dự án đã hoàn thành.

Hợp đồng phí cộng thêm phải bao gồm các điều khoản sau:

- Danh sách những người được giao làm dự án.
- Xác định công việc.
- Tỷ lệ phần trăm giao việc cho mỗi người.
- Mức công việc hàng giờ hay hàng ngày cho mỗi người.
- Tổng phí hành chính.
- Chi phí được phép để được bồi hoàn.
- Thủ tục làm hóa đơn.
- Thủ tục thanh toán.
- Thủ tục kết thúc.

Hợp đồng giá cố định:

Hợp đồng giá cố định là một cam kết của người phát triển sẽ cung cấp sản phẩm hay dịch vụ thỏa thuận với phí thỏa thuận trong phạm vi lịch trình thỏa thuận. Điều này tương tự với mua tic kê xe buýt theo đó công đi xe buýt thỏa thuận đưa khách hàng đến nơi nhất định trong phạm vi thời gian biểu đã công bố với phí thỏa thuận. Tất nhiên, du khách có thể chọn thuê xe chứ không mua tic kê xe buýt và rời tự mình lái đến nơi của mình. Dù sao, điều này có thể trở nên tốn kém hơn và đòi hỏi ở người du khách đôi chút kỹ năng và kiến thức trước như kỹ năng lái xe và kiến thức về hành trình đến nơi. Nếu du khách (*hay khách hàng*) phải quyết định giữa việc tự mình tạo ra dịch vụ và việc hợp đồng với ai đó để cung cấp dịch vụ.

Hợp đồng giá cố định chỉ có thể được vận dụng cho một dự án xác định rõ. Cả khách hàng và người sản xuất phải có khả năng xác định sản phẩm hay dịch vụ cuối cùng mong muốn. Một khi điều này để thực hiện được, một trong những yếu kém chính của hợp đồng cố định sẽ được khắc phục.

Lợi ích của hợp đồng giá cố định cho người phát triển là:

- Không chế đầy đủ quá trình phát triển.

- Lợi ích kinh doanh có thể cao hơn.
- Cam kết cho dự án trọn vẹn.
- Cam kết cho dự án trọn vẹn là ưu việt đáng kể so với hợp đồng phí cộng thêm ở đó nó có thể kết thúc bất cứ lúc nào tùy sự xét đoán của khách hàng.

Tất nhiên, hợp đồng giá cố định cũng có một số bất lợi cho người phát triển, bao gồm:

- Đảm nhận rủi ro kinh doanh và phát triển.
- Bất đồng tiềm ẩn với khách hàng do:
 - Thay đổi yêu cầu liên tiếp.
 - Tiêu chuẩn hoàn thành dự án.
 - Giải thích yêu cầu.

Một tổ chức phần mềm thắng lợi sẽ thường chuộng hợp đồng giá cố định. Đó thường là những dự án tạo dựng danh tiếng chuyên môn của công ty và phát sinh lợi tức đảm bảo tăng trưởng. Bất hạnh là những dự án này cũng gây ra lỗi và thường tác hại nghiêm trọng cho công ty cạnh tranh gay gắt để có hợp đồng quan trọng đôi khi làm cho công ty nhận thầu giá thấp cuối cùng gây ra lỗi cho người phát triển.

Hầu như không thể tránh khỏi trong bất kỳ dự án nào người phát triển được đòi hỏi thay đổi yêu cầu trong quá trình phát triển. Những thay đổi như thế thường đi liền với chi phí bổ sung đòi khách hàng và bao giờ cũng là nguyên nhân bất đồng giữa người sản xuất và khách hàng. Điều này thường là do yêu cầu không rõ hay mơ hồ và nó lại dẫn đến bất đồng về chỉ tiêu trong việc hoàn thành dự án. Về cơ bản, điều này làm cho hợp đồng trở lại trạng thái không được xác định đầy đủ.

Theo quan điểm của khách hàng, ưu việt của hợp đồng giá cố định bao gồm:

- Ngân sách cố định cho dự án.
- Hầu hết các rủi ro phát triển được chuyển sang người phát triển.
 - Tham gia tối thiểu trong quá trình phát triển.

Bất lợi cho khách hàng là:

- Rủi ro chậm giao sản phẩm.
- Giảm sự khống chế quá trình phát triển
- Bất đồng tiềm ẩn với người sản xuất do:
 - Chi phí cao về thay đổi yêu cầu.
 - Chỉ tiêu hoàn thành dự án không rõ ràng.
- Giải thích yêu cầu.

Ngay dù quyền lợi của người sản xuất và khách hàng có thể khác nhau, với hợp đồng giá cố định vẫn thường được cả hai bên ưa chuộng. Nếu dự án là chi tiết đủ mức và rõ ràng và nếu quan hệ giữa hai bên được xác định rõ thì các hợp đồng giá cố định có thể có lợi cho cả người phát triển lẫn khách hàng.

5.1.6 Xây dựng dự toán cho các giai đoạn

Dự toán và các vấn đề liên quan:

Dự toán tốt là quan trọng vì chúng tạo thành nền móng của kế hoạch phát triển dự án tốt. Kế hoạch đó, do nhà quản lý dự án chuẩn bị được lập thành trong các giai đoạn ban đầu của dự án và bao gồm dự toán liên quan đến:

- Ngân sách phát triển dự án.
- Lịch trình phát triển dự án.
- Tài nguyên phát triển cần đến (*đội ngũ phát triển thiết bị phát triển v.v...*).

Những dự toán kỹ thuật cũng được hình thành trong pha thiết kế và bao gồm:

- Các đặc tính của phần mềm (*dự toán về cỡ bộ nhớ, cơ sở dữ liệu v.v...*).
- Các đặc tính của phần cứng về cỡ mục tiêu cần đến (*dự toán tốc độ CPU, khả năng đầu vào, đầu ra; khả năng điều khiển đĩa v.v...*).

Dự toán là cơ sở cho nhiều quyết định kỹ thuật và quản lý dự toán tôi dẫn đến quyết định dở. Dự toán tôi có thể hiểu là quá cao hoặc là quá thấp và quyết định theo đó hoặc là tạo ra lãng phí hoặc thiếu tài nguyên phát triển. Điều này hình thành sai lầm trong qui hoạch, như:

- Lịch trình quá ngắn hoặc cao thái quá.
- Ngân sách quá thấp hay quá tăng giả tạo.
- Quá thiếu hay quá thừa người làm và những sai lầm trong thiết kế kỹ thuật, như: Những máy tính trong mục tiêu quá nhiều (*và đắt hơn*) hơn cần thiết hoặc không thể hỗ trợ ứng dụng được phát triển.

Những vấn đề nảy sinh từ dự toán thấp thường gay gắt hơn là những vấn đề nảy sinh từ dự toán cao. Hiểu rõ điều này, các nhà dự toán thường thêm vào một số yếu tố bất trắc (*cho rằng đến 30%*) trong dự toán của mình, giả định rằng quá cao còn hơn quá thấp.

Dù sao dự toán cao có thể không gây ra thất bại của dự án nhưng có thể ngăn cản dự án chẳng bao giờ được khởi công. Đã có nhiều phương pháp được phát triển nhằm tạo ra các loại dự toán khác nhau ở các giai đoạn khác nhau của dự án.

Những dự toán chuẩn bị tốt có thể dẫn đến những vấn đề nếu chúng không được cập nhật trên một cơ sở định kỳ và đều đặn. Rõ ràng là thông tin tốt hơn và đầy đủ hơn tạo nên dự toán tốt hơn. Do đó khi dự án tiến triển và có nhiều thông tin hơn, dự toán cần được xem xét lại và hiệu chỉnh. Điều này dẫn đến việc giám định lại các quyết định phát triển, tạo cho các vấn đề tiềm ẩn được đề cập sớm trước khi chúng trở thành vấn đề rủi ro.

Dự toán của dự án:

Với người quản lý dự án phần mềm, có nhiều định lượng chưa biết phải dự toán. Những định lượng này liên quan đến các lĩnh vực như:

- Chi phí phát triển dự án.
- Lịch trình phát triển dự án
- Qui mô đội phát triển dự án
- Khối lượng phần mềm phải phát triển.
- Nguồn phân cứng yêu cầu.

Dự toán những định lượng đó như thế nào không chỉ là vấn đề cần được đề cập: Các đơn vị sử dụng để đo những định lượng đó cũng phải cần được xem xét. Chi phí phát triển dự án được đo lường tốt nhất bằng đơn vị tiền tệ, như đồng đôla Mỹ hay bảng (Anh). Dù sao với dự toán ban đầu chấp nhận được việc chuẩn bị bằng một đo lường trung gian sau đó được chuyển đổi sang đơn vị tiền tệ. Đơn vị trung gian phổ thông cho chi phí phát triển dự án là *tháng công* (hay *tháng nhân công* hay *tháng kỹ sư v.v.*).

Lịch trình phát triển dự án rõ ràng được dự toán bằng đơn vị thời gian như ngày, tuần, tháng hay năm. Lịch trình giống như phần lớn các kế hoạch, phải có khởi đầu và kết thúc. Thường khi một dự án được quy hoạch, khởi đầu chính thức không rõ, và do đó kết thúc cũng không rõ nốt. Trong những trường hợp đó, tên tắt ARO (*sau khi nhận đơn đặt*) thông thường được dùng và lịch trình được xây dựng sử dụng những chỉ định như: *Kết thúc thiết kế mức cao - 4 tháng ARO*.

Dự toán nhân sự phát triển có yêu cầu cho dự án rõ ràng theo đơn vị người. Dù sao, nhân sự phát triển phải được nhóm theo một số phân loại phổ thông như kỹ sư phần mềm, người lập trình, nhân lực hỗ trợ v.v. và mỗi nhóm phải được dự toán riêng rẽ. Thời gian về dự án của mỗi người cũng phải được dự toán bằng cách nhận ra việc họ tham gia vào dự án và rút ra khỏi dự án.

Khối lượng phần mềm được phát triển thường được dự toán bằng cả hai cách đo: Bằng dòng mã hay kilô Bai (Kbyte) của bộ nhớ. Cả hai phương pháp có điểm lợi và bất lợi. Số dòng mã là mức đo tiêu biểu nhiều hơn về mức độ cố gắng phát triển nhưng cũng là ngôn ngữ tùy thuộc ở chỗ hàng trăm dòng mã ngôn ngữ mức cao không đòi hỏi cũng mức độ cố gắng như hàng trăm dòng mã hình thành ngôn ngữ mức thấp.

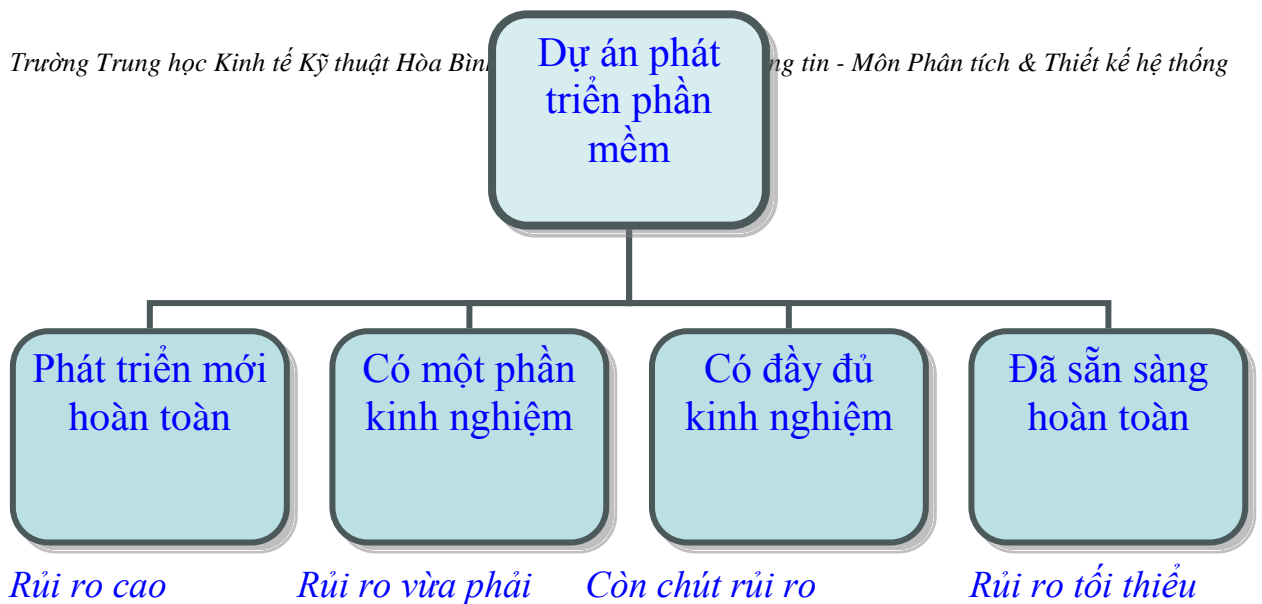
Dự toán từng bước:

Phương cách “*Phân chia và Khuất phục*” thường được áp dụng trong nhiều lĩnh vực công nghệ phần mềm. Phương pháp này (*phân chia vấn đề lớn thành vô vàn vấn đề nhỏ hơn*) cũng được sử dụng trong hầu hết kỹ thuật dự toán. Tiếp cận cơ bản là phân giải dự án thành những hợp phần được xác định rõ ràng và rồi lặp lại từng bước đến khi chỉ còn những đơn vị nhỏ sau đó có thể được dự toán dễ dàng hơn.

Bước đầu tiên trong phân giải dự án là phân chia dự án thành bốn loại sau - Hình 5.8.

- Hợp phần đã có sẵn sàng.
- Hợp phần đầy kinh nghiệm.
- Hợp phần kinh nghiệm 1 phần.
- Phát triển mới.

Mỗi loại biểu thị một lớp hoạt động phát triển có thể được dự toán riêng rẽ sử dụng các phương pháp dự toán khác nhau. Như chúng ta sẽ thấy, mỗi loại cũng gắn với mức rủi ro liên quan trong phát triển phần mềm.



Hình 5.8: Phân loại phát triển dự án theo mức độ rủi ro

5.2 QUẢN LÝ GIAI ĐOẠN PHÂN TÍCH

5.2.1 Tiếp cận hiện trạng

Tiến hành tìm hiểu sơ bộ về hệ thống hiện hành và đưa ra cho được các điểm yếu của hệ thống đó. Trên cơ sở đó, nêu lên các phương pháp cải tiến cho hệ thống bao gồm:

- ✧ Đánh giá hiện trạng.
- ✧ Các phạm vi, khả năng, mục tiêu của dự án.

5.2.1.1 Tìm hiểu, đánh giá hiện trạng của hệ thống hiện hành:

- ✧ Điều tra, thu thập thông tin về hệ thống hiện hành.
- ✧ Biên tập, biểu diễn, phê phán, đề xuất ý kiến.

5.2.1.2 Phương pháp khảo sát:

Khảo sát hệ thống ở cả bốn mức:

- ✧ Mức thao tác thừa hành: tiếp cận các người thừa hành trực tiếp để xem họ làm việc.
- ✧ Mức điều phối quản lý.
- ✧ Mức quyết định lãnh đạo.
- ✧ Mức chuyên gia cố vấn.

Hình thức tiến hành:

- ✧ Quan sát và theo dõi:
 - ✓ Một cách chính thức: cùng làm việc với họ.
 - ✓ Một cách không chính thức: tìm hiểu cách làm việc qua các hồ sơ, sổ sách, v.v...
- ✧ Cố vấn: Bằng nhiều cách:
 - ✓ Đặt câu hỏi trực tiếp: Yes / No
 - ✓ Đặt câu hỏi chọn lựa: a, b, c, d ..., đánh ✓ để thống kê.
 - ✓ Đặt câu hỏi gián tiếp có tính gợi mở cho câu trả lời
 - ✓ Bảng câu hỏi, phiếu điều tra.

5.2.1.3 Phát hiện các yếu kém của hiện trạng và yêu cầu cho tương lai:

a. Yếu kém:

- ✧ Thiếu sót:
 - √ Thiếu người xử lý thông tin.
 - √ Bỏ sót công việc xử lý thông tin.
- ✧ Kém hiệu lực, quá tải:
 - √ Phương pháp xử lý không chặt chẽ.
 - √ Cơ cấu tổ chức không hợp lý.
 - √ Con đường lưu chuyển các thông tin không hợp lý. Ví dụ: Giấy tờ, tài liệu trình bày kém, cấu trúc không hợp lý, v.v...
- ✧ Tồn phí cao, gây lãng phí.

b. Yêu cầu mới:

Trong tương lai:

- ✧ Thỏa đáng các thông tin chưa được đáp ứng.
- ✧ Đáp ứng các nguyện vọng của nhân viên.
- ✧ Dự kiến kế hoạch phát triển.

5.2.1.4. Lập dự trù và kế hoạch triển khai dự án:

5.2.1.4.1 Lập hồ sơ khảo sát:

a. Lập dự trù về thiết bị:

* Tìm hiểu và dự kiến các vấn đề sau:

- ✧ Khối lượng dữ liệu lưu trữ.
- ✧ Các dạng làm việc với máy tính (máy đơn, máy mạng), xử lý trực tuyến (Online), v.v...
- ✧ Số lượng người dùng tối thiểu và tối đa của hệ thống.
- ✧ Khối lượng thông tin cần thu thập.
- ✧ Khối lượng thông tin cần kết xuất, cần in ra giấy, v.v...
- ✧ Thiết bị ngoại vi đặc biệt như: Scanner, máy vẽ, máy cắt, v.v...

* Điều kiện mua và lắp đặt:

- ✧ Nên chọn nhà cung cấp nào, chi phí vận chuyển.
- ✧ Mua nguyên bộ, mua rời, v.v...
- ✧ Sơ đồ lắp đặt mức sơ bộ.

b. Công tác huấn luyện sử dụng chương trình:

- ✧ Thời gian huấn luyện bao lâu.
- ✧ Chia làm bao nhiêu nhóm huấn luyện.

c. Công việc bảo trì:

- ✧ Đội ngũ bảo trì.
- ✧ Chi phí bảo trì.
- ✧ Thời gian bảo trì.

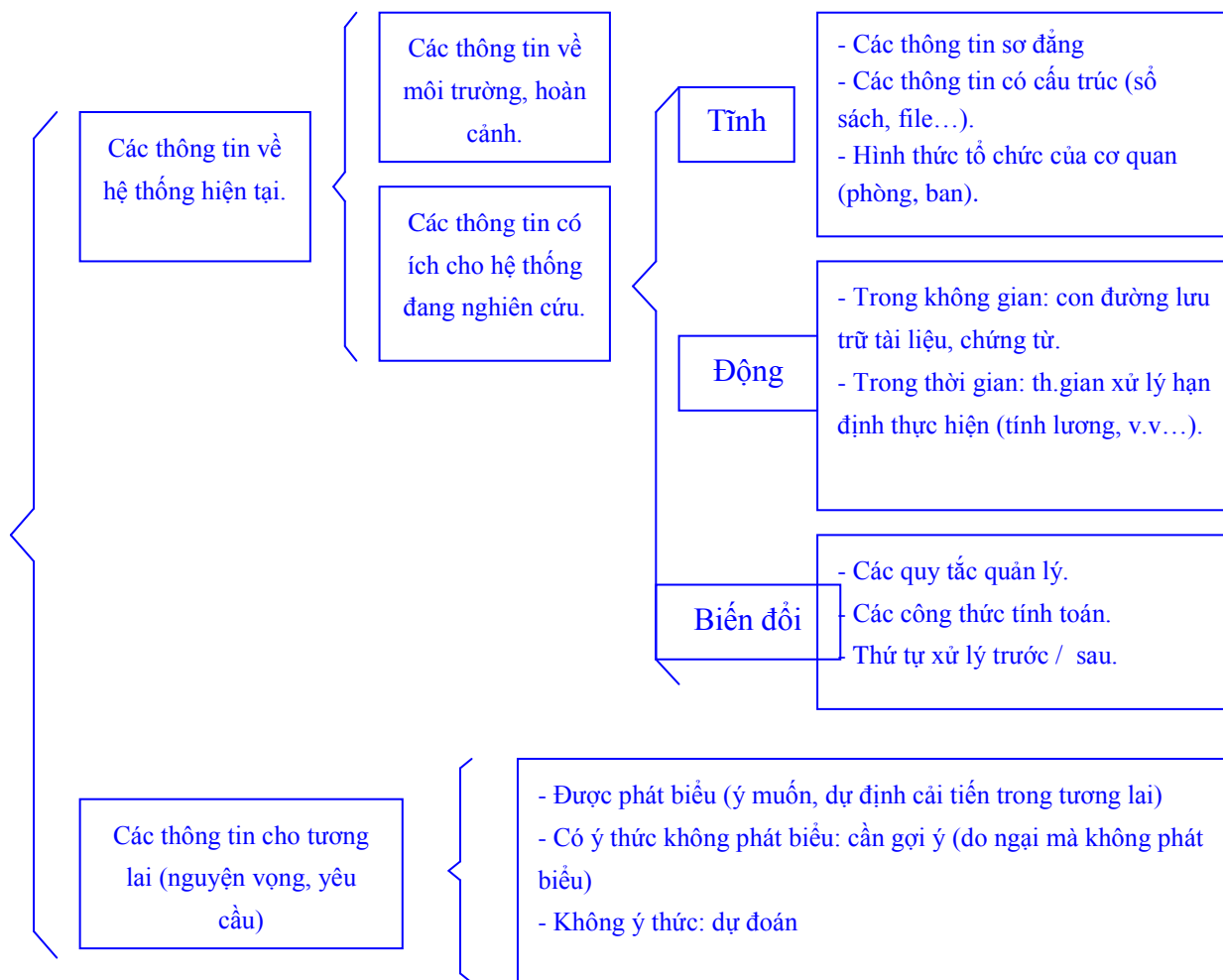
5.2.1.4.2 Lập kế hoạch triển khai dự án:

- ✧ Về mặt nhân sự: Có mặt tất cả các chuyên viên, người sử dụng, lãnh đạo cơ quan, phân tích viên hệ thống (có thể có cả các lập trình viên).
- ✧ Lập tiến độ triển khai dự án.
- ✧ Phân tích tài chính dự án.
- ✧ Lập mối quan hệ với các dự án khác.

5.2.2 Thu thập các tài liệu có liên quan đến hệ thống sẽ tin học hóa

Đây là một nhiệm vụ tương chừng như đơn giản đến mức nhóm phát triển thấy điều đó là hiển nhiên không đánh giá đúng mức tuy nhiên trên thực tế lại không phải vậy vì “*tài liệu có liên quan*” không hẳn chỉ là những gì có liên quan trực tiếp đến hệ thống mà thôi. Tóm lại nên thu thập tất cả các tài liệu có thể có được và chỉ loại khi mà dự án kết thúc giai đoạn thử nghiệm bao gồm:

- ✧ Thông tin về hiện tại hay tương lai.
- ✧ Thông tin về trạng thái tĩnh, động hay biến đổi.
 - √ Tĩnh: Thông tin về tổ chức hồ sơ và sổ sách.
 - √ Động: Thông tin về sự tăng hay giảm lưu chuyển của các chứng từ, giấy tờ, v.v...
 - √ Biến đổi: Thông tin được biến đổi ra sao, sử dụng những công thức tính toán nào?
- ✧ Thông tin thuộc nội bộ hay môi trường của hệ thống, thông thường thì người ta tổ chức sắp xếp, tổ hợp những vấn đề thông tin trên lại như sau:



5.2.3 Triển khai theo kế hoạch đã xác lập ở giai đoạn trước

Các hạng mục kế hoạch phát triển dự án phần mềm:

1. Tổng quan hệ thống.
2. Quản lý phát triển phần mềm: Tổ chức dự án và nguồn lực dự án, Phương tiện phát triển, Cơ cấu tổ chức dự án, Nhân sự
3. Lịch và cột mốc: Hoạt động theo lịch, Đường mốc và cột mốc, Sơ đồ mạng hoạt động, Nguồn lực thành tố hệ thống, Quản lý ngân sách: Thanh toán cột mốc, chi phí ngân sách chủ yếu, thủ tục cho phép chi phí.
4. Phân tích rủi ro.
5. An toàn.
6. Giao diện với các nguồn bên ngoài.
7. Thủ tục rà soát lại hợp thức.
8. Quá trình tiến hành hiệu chỉnh.
9. Báo cáo về các thay đổi/vấn đề.
10. Công nghệ phần mềm: Tiêu chuẩn và thủ tục, Phương pháp luận phát triển, nguồn lực phát triển, nhân sự - trình độ và chức trách
11. Thủ tục thử nghiệm
12. Quản lý cấu hình phần mềm
13. Bảo đảm chất lượng phần mềm.

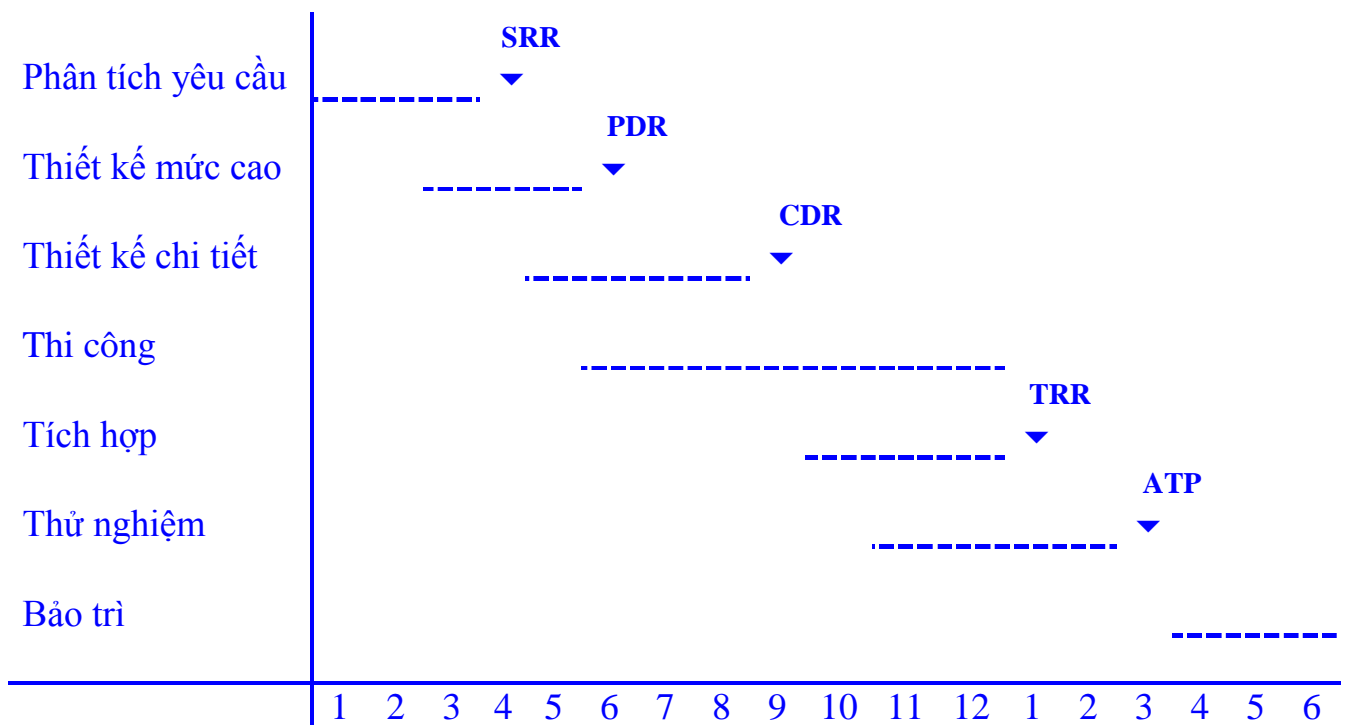
Độc giả có thể tham khảo tài liệu Quản lý dự án phần mềm – Đại học Quốc Gia Hà Nội để tìm hiểu đầy đủ và sâu hơn.

Các biểu đồ GANTT: Theo [QLDAPM.ĐHQG].

Trước khi có máy tính ra đời rất nhiều, Henry L. Gantt đã lấy tên mình đặt cho biểu diễn đồ thị đơn giản và rất có ích về lịch phát triển dự án. Biểu đồ GANTT cho thấy hầu hết thông tin có trong danh sách hoạt động của lịch theo một cách dễ hiểu hơn. Thông tin theo lịch dễ nắm bắt và hiểu và các hoạt động có thể so sánh dễ dàng. Biểu đồ GANTT giúp chúng ta nhìn thấy vào bất cứ lúc nào có những hoạt động gì hẳn là đang diễn ra trong dự án đó. Hình 5.9 là một thí dụ điển hình về biểu đồ GANTT. Các ký hiệu được dùng trong biểu đồ là được chấp nhận một cách rộng rãi, mặc dù là chúng chưa được chuẩn hoá. Chẳng hạn, mỗi tam giác ngược thường được dùng để chỉ một sự kiện đáng kể, ví như là các cột mốc.

Biểu đồ GANTT trong hình 5.9 thuyết minh sự dễ dàng nhanh chóng nhận thức được các thông tin lập lịch quan trọng. Chúng ta có thể trực tiếp thấy ngay rằng không kể đến pha bảo trì, mọi pha khác đều gối đầu nhau, và rằng trong khoảng thời gian từ tháng 11 đến giữa tháng 12 năm xx thì ba hoạt động mức cao là gối đầu nhau.

Một biểu đồ chi tiết hơn cũng có thể bao gồm các tên của các kỹ sư gắn với từng hoạt động và bao gồm các thiết bị cần thiết cho từng hoạt động một. Thông tin này có thể được thêm vào bên cạnh đường thời gian hoạt động trong đồ thị đó, hoặc là một bảng tham khảo lồng vào bên (*trông tự như danh sách các cột mốc chủ yếu trong hình 59*). Một vài biến thể của biểu đồ GANTT thực cũng đã bao gồm kiểu thông tin đó trên biểu đồ, nhưng chúng có thể gây ra sự lộn xộn, mà như thế lại mâu thuẫn với chính mục tiêu của biểu đồ đó: Làm cho người ta có thể dễ dàng nhanh chóng nắm bắt các thông tin lịch trình.



Các cột mốc chủ yếu

SRR = Rà soát yêu cầu phần mềm

PDR = Rà soát thiết kế sơ khởi

CDR = Rà soát thiết kế tới hạn

TRR = Rà soát thử nghiệm

ATP = Thủ tục thử nghiệm chấp thuận

Hình 5.9: Biểu đồ GANTT mức cao - Lịch phát triển dự án

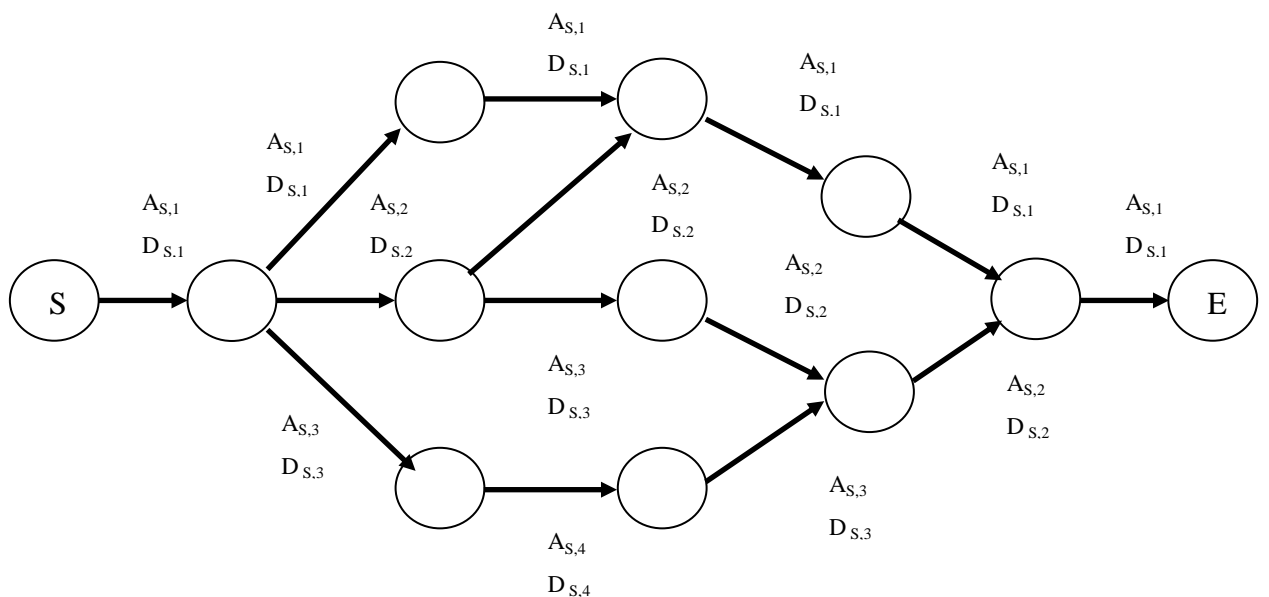
Các biểu đồ PERT và đường tới hạn:

Kỹ thuật rà soát lại và đánh giá chương trình (PERT) sử dụng mạng quan hệ trước sau để lập kế hoạch các hoạt động dự án và khống chế sự thực hiện chúng một cách hiệu quả. Giống như biểu đồ GANTT có nhiều loại biến thể của biểu đồ PERT. Kỹ thuật PERT quy ước cơ bản mô tả một mạng với các node (mẫu) là các sự kiện và các cung nối là các hoạt động.

Mỗi hoạt động liên kết với hai sự kiện liên quan đầu và cuối. Node kết thúc của một hoạt động có thể trùng với node đầu hoạt động thứ hai, khi mà sự thực hiện của hoạt động thứ hai tùy thuộc ở sự thực hiện của hoàn thành của hoạt động thứ nhất. Điều này có nghĩa một hoạt động chỉ có thể được thực hiện khi mọi hoạt động khác kết thúc ở node khởi điểm của chúng đã được hoàn thành.

Hình 5.10 cho thấy một thí dụ của biểu đồ mạng PERT biểu thị dòng hoạt động của dự án từ node khởi điểm tới node kết thúc. Mỗi sự kiện được biểu thị bằng một vòng tròn có đánh số mạng bắt đầu bằng sự kiện khởi điểm, gọi là node nguồn và kết thúc bằng sự kiện kết thúc gọi là node chìm. Mỗi đường liên kết biểu thị một hoạt động dự án. Hoạt động $A_{i,j}$ mô tả hoạt động bắt đầu với sự kiện i và kết thúc ở sự kiện j . Thuộc số $D_{i,j}$ biểu thị lượng thời gian trôi theo lịch, giữa khởi đầu và kết thúc của hoạt động $A_{i,j}$.

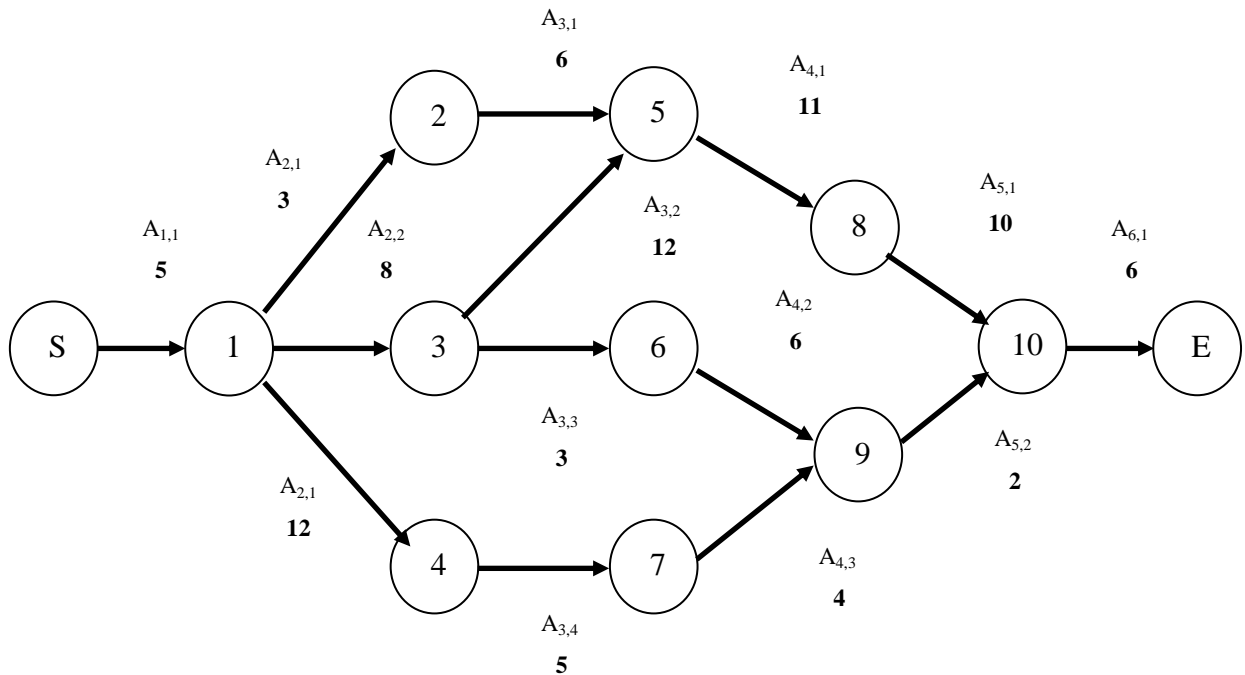
Một khía cạnh quan trọng của phương pháp biểu đồ PERT là quan niệm các hoạt động song hành. Mỗi node sự kiện bắt nhánh vào một số hoạt động có thể thực hiện song hành. Trong hình 5.11 các hoạt động $A_{1,2}$, $A_{1,3}$ và $A_{1,4}$ có thể thực hiện song hành. Dù sao mỗi một hoạt động $A_{S,1}$ và $A_{10,E}$ không thể thực hiện song song với bất cứ hoạt động nào khác. Chúng ta cũng có thể thấy ở biểu đồ là hoạt động $A_{3,6}$ có thể thực hiện song song với hoạt động $A_{5,8}$ hay hoạt động $A_{8,10}$ nhưng không với cả hai. Tương tự, ở một thời điểm đã cho nào đó hoạt động $A_{3,6}$ có thể thực hiện song song với chỉ một trong ba hoạt động $A_{1,4}$, $A_{4,7}$ và $A_{7,9}$.



$A_{i,j}$: Hoạt động bắt đầu tại node i và kết thúc tại node j.

$D_{i,j}$: Thời gian của hoạt động $A_{i,j}$.

Hình 5.10: Một biểu đồ PERT điển hình.



$A_{i,j}$: Hoạt động bắt đầu tại node i và kết thúc tại node j.

$D_{i,j}$: Thời gian của hoạt động $A_{i,j}$.

Hình 5.11: Một biểu đồ PERT điển hình.

Con đường tới hạn:

Khủng hoảng của các lịch phát triển dự án là phổ biến mà những nhà quản lý dự án phải tiên liệu. Những dự án lớn có nhiều phần liên quan trong việc lên lịch như quản lý tập đoàn, khách hàng chủ thầu phụ, người bán, người dùng, bộ phận tiếp thị v.v. Một trong những khủng hoảng thường gặp nhất là về nhu cầu rút ngắn lịch. Một sai lầm chung về phía người quản lý dự án là cho rằng bất cứ một cố gắng phụ cũng sẽ rút ngắn được lịch. Dù sao trong một số trường hợp những hoạt động rút ngắn sẽ tuyệt đối không tác động gì đến thời hạn tổng thể của lịch.

Nhằm xem xét hiện tượng này trước hết chúng ta phải hiểu là trong mọi mạng không tầm thường có nhiều đường chuyển tin node nguồn tới node chìm. chẳng hạn trong hình 5.11 con đường khả dĩ chạy từ node S tới 1 tới 3 tới 6 tới 9 tới 10 tới E. Một con đường khả dĩ khác chạy từ nút S tới 1 tới 2 tới 5 tới 8 tới 10 tới E. Mỗi con đường có thể được đặc trưng bằng con số biểu thị tổng thời hạn của con đường, được tính bằng cách lấy tổng số các thời hạn cho mọi hoạt động dọc con đường.

Bảng 5.12 có danh mục mọi con đường khả dĩ từ nút S tới nút E trong biểu đồ PERT ở hình 5.11 cùng với độ dài của mỗi con đường. Con đường 2 là dài nhất, 52 tuần. Con đường dài nhất được coi là con đường tới hạn xác định thời hạn của dự án.

Khi rút ngắn thời hạn hoạt động của một hoạt động dọc con đường tới hạn, thường chúng ta có thể rút ngắn được thời hạn của toàn dự án. có ít trường hợp thái cực mà điều này sẽ không diễn ra chủ yếu khi có hai con đường tới hạn. Dù sao có một điều là chắc chắn rút ngắn hoạt động không có trên con đường tới hạn sẽ không rút ngắn được chiều dài của toàn bộ dự án.

Bảng 5.12 Mọi con đường khả dĩ từ S tới E (căn cứ ở Hình 5.11)

Con đường	Độ dài	Con đường tới hạn
1. S.1.2.5.8.10.E	41	
2. S. 1.3.5.8.10.E	52	xxxxxxx
3. S. 1.3.6.9.10.E	30	
4. S. 1.4.7.9.10.E	34	

5.2.4 Tổng hợp các yêu cầu, phân loại theo nhóm, loại bỏ các yêu cầu trùng lặp, lập hồ sơ cho từng nhóm việc.

Phần kiến thức này đã được trình bày ở chương 2 tuy nhiên cũng không thể không nhắc lại bởi vì hồ sơ nghiên cứu hiện trạng là mức bất biến cao nhất và không thể thay đổi của một dự án tin học hóa.

Thứ nhất: Quá trình tổng hợp với nhiều lý do chủ quan, khách quan mà các yêu cầu có thể bị bỏ sót hoặc là thiếu vì thế việc bổ sung các yêu cầu mặc dù không mong muốn nhưng trong dự án tin học sẽ lại phải bắt đầu từ việc duyệt lại hồ sơ nghiên cứu hiện trạng.

Thứ hai: Hơn thế nữa hồ sơ phân tích thiết kế dự án cần tin học hóa được gọi là tốt chỉ khi hồ sơ đó độc lập với việc sử dụng công cụ để phát triển cũng tương tự như hồ sơ thiết kế nhà một tòa nhà vậy.

Thứ ba: Hồ sơ nghiên cứu hiện trạng là ngôn ngữ để ký kết hợp đồng giữa nhà đầu tư và nhà phát triển. Việc nhà đầu tư cũng có thể thay đổi yêu cầu cũng là điều nên cân nhắc cho các chi phí sau này.

Thứ tư: Quá trình phân đã thành các tiểu dự án hoặc chỉ là lập hồ sơ cho từng nhóm việc là quá trình tam sao thất bản ảnh hưởng rất lớn đến quá trình tích hợp và thử nghiệm hệ thống. Đây là là một rủi ro thường có của tất cả các dự án tin học hóa cần phải đánh giá mức độ nghiêm trọng dựa trên kỳ vọng và tác động đồng thời xây dựng kế hoạch đối phó.

Những việc phải thực hiện chính là:

5.2.4.1 Xác định khả năng, mục tiêu dự án của hệ thống mới:

- ✧ Phạm vi của hệ thống mới giải quyết vấn đề gì?
- ✧ Nhân lực sử dụng. Ví dụ: đội ngũ nhân viên điều khiển hệ thống cần bao nhiêu?
- ✧ Tài chính (*Chi phí bao nhiêu cho dự án. Ví dụ: Phí viết chương trình, phí bảo trì, v.v...*).
- ✧ Khắc phục các điểm yếu kém của hệ thống hiện tại.
- ✧ Thể hiện chiến lược lâu dài. Dự án phải có hướng mở, ví dụ: trong tương lai dự án có thể được phát triển thêm, giải quyết thêm những vấn đề gì?

5.2.4.2 Phác họa giải pháp và cân nhắc tính khả thi:

Đưa ra giải pháp đề thuyết phục người dùng (ở mức sơ bộ). Từ đó, định hướng cho việc phân tích và thiết kế hệ thống thông tin. Ta nên đưa ra nhiều giải pháp:

- ✧ Giải pháp cho máy đơn.
- ✧ Giải pháp máy mạng, v.v...

Với từng giải pháp phải mang tính khả thi:

- ✧ Khả thi về mặt nghiệp vụ: phải đáp ứng được các yêu cầu của công việc.
- ✧ Khả thi về mặt kỹ thuật: sử dụng phù hợp với hệ thống máy hiện có, tương lai, v.v...
- ✧ Khả thi về mặt kinh tế: chi phí viết chương trình có thể chấp nhận được, chi phí bảo trì không quá cao, v.v...

5.2.5 Lập kế hoạch phân công, bố trí nhân sự tiếp cận đối tác làm rõ các yêu cầu, khái niệm chuyên môn chưa rõ.

Đây là vấn đề nằm trong pha yêu cầu, Chúng ta thường nhận thức đây là giai đoạn quan trọng nhất của chu trình phát triển phần mềm. Chắc chắn rằng nó là một trong những phần khó khăn nhất, một phần do khó khăn trong việc lập tư liệu một mô tả được chấp thuận cho yêu cầu phần mềm. Bao giờ cũng có mâu thuẫn cơ bản về quyền lợi giữa khách hàng và người phát triển. Khách hàng miễn cưỡng kết thúc yêu cầu vì nhận biết rằng một khi đã làm điều này, mọi thay đổi sau này có thể phải trả giá. Mặt khác, người phát triển cần kết thúc yêu cầu càng sớm càng tốt vì tiến triển sẽ chậm đi chừng nào sản phẩm còn chưa hoàn toàn được xác định và cũng chính điều này là tổn kém. Nên đặc điểm chính của pha yêu cầu có thể được đặc trưng như là một hình thức của trò kéo co giữa khách hàng và người phát triển.

Có nhiều vấn đề liên quan đến việc thành lập đội ngũ phát triển dự án ban đầu. Đặt đúng người vào đội ngũ dự án hiếm khi là một việc dễ dàng. Thật vậy, bố trí người quản lý dự án thích hợp không phải bao giờ cũng là một việc dễ dàng.

- Vấn đề thông thường hoặc là thiếu hoặc là thừa lãnh đạo dự án. Trong nhiều trường hợp, pha quan niệm do nhiều người chỉ đạo, người này có thể chịu trách nhiệm về phân tích thị trường, người kia về dự toán và lại người khác nữa về soạn thảo các tư liệu quan niệm. Điều này có thể gây ra tình trạng thiếu phối hợp giữa những người có trách nhiệm về các hoạt động khác nhau, tạo nên nhiều mâu thuẫn và những quyết định ban đầu tồi.

- Khi đội ngũ ban đầu đã yên vị và kế hoạch phát triển sơ bộ đã được soạn thảo, không phải không phổ biến có chuyện nhiều ý kiến khác nhau về việc sản phẩm nên phải thực sự như thế nào tạo ra tư liệu quan niệm được thoả thuận thường có thể trở nên một vấn đề chủ yếu. Vấn đề này thường được giải quyết thông qua vận dụng các hệ thống trình diễn hay nguyên mẫu.

- Bố trí nhân sự : Việc sắp xếp thành viên đội ngũ phát triển thích hợp có thể là một nhiệm vụ khó. Việc bố trí các thành viên đội ngũ phải được hoàn tất trong pha yêu cầu.

Những vấn đề nhân sự thường có thể giải quyết bằng cách:

- Sử dụng nhân lực tạm thời.
- Tăng dân đội ngũ sau này.
- Huấn luyện nhân lực hiện có.

5.2.6 Hiệp thương với đối tác để giải quyết các vấn đề mới nảy sinh về mặt kỹ thuật và tài chính.

Như chúng ta đã thấy, có được sự phê duyệt chính thức về đặc tả yêu cầu là nhiệm vụ khó khăn nhất của giai đoạn này. Thường các yêu cầu tiến triển dần dần với nhiều phương án dự thảo của đặc tả tập trung vào tư liệu được phê duyệt cuối cùng. Dù sao nếu những thay đổi chủ yếu được đưa ra liên tục thì sẽ khó lên kế hoạch. Trong những tình hình như thế, việc tạo nguyên mẫu nhanh có thể có ích nhất trong việc giúp khách hàng xác định yêu cầu.

Mâu thuẫn giữa người phát triển và khách hàng trong việc kết thúc đặc tả yêu cầu. Đôi khi xảy ra các tranh chấp, bất đồng và lẫn lộn biểu thị không khí của giai đoạn yêu cầu chính là nguồn gốc chủ yếu của các vấn đề. Danh sách thì dài nhưng những vấn đề thông thường nhất là :

- Thay đổi yêu cầu luôn : Trong pha này, thường có luồng thay đổi có vẻ vô tận làm cho việc thu thập đặc điểm yêu cầu thêm khó khăn.
- Chấp thuận yêu cầu và ký: pha này không thể hoàn tất nếu không có tư liệu yêu cầu được phê duyệt chính thức.
- Tính khả thi của yêu cầu: đôi khi điều này khó xác định trước giai đoạn thực hiện.
- Các thay đổi phút chót : Đây là một vấn đề chỉ có thể được giải quyết bằng cách đảm bảo thủ tục khống chế các thay đổi một cách nghiêm ngặt và có trình tự. Luồng thay đổi là tai họa của công trình nói chung và của phát triển phần mềm nói riêng. Rõ ràng thay đổi phải được phép trong một chừng mức nhưng luồng thay đổi không kiểm soát được có thể làm cho dự án sụp đổ.
- Tranh chấp với khách hàng. Cuối cùng khách hàng có thể nhìn thấy phiên bản ban đầu của cái mà sản phẩm sẽ có dáng vẻ thế nào. Về điều này, có những lý giải yêu cầu khác nhau nảy sinh ra và thường cần được giải quyết ở cấp quản lý cao hơn.

5.3. QUẢN LÝ GIAI ĐOẠN THIẾT KẾ

Những việc phải làm khi thiết kế một hệ thống tin học hóa:

1. *Tài liệu dùng cho thiết kế.*
2. *Các loại sơ đồ.*
3. *Cơ sở dữ liệu.*
4. *Giao diện – Báo cáo.*

5.3.1 Phân công nhóm thiết kế tổng thể, nhóm thiết kế chi tiết

Trong pha thiết kế, các yêu cầu được phân tích và phương pháp thực hiện được xác định. Pha yêu cầu trước đây đặt câu hỏi “cái gì?”, pha thiết kế đặt câu hỏi “thế nào?” Việc trả lời cho câu hỏi này được có trong tư liệu đặc tả thiết kế phần mềm.

Pha thiết kế thường được chia làm hai pha riêng biệt: *Thiết kế mức đỉnh* và *thiết kế chi tiết*. Đường phân chia giữa hai pha này được căn cứ ở mức độ phân chia hệ thống thành các phần nhỏ chi tiết theo nhóm các nhiệm vụ.

Một trong những lợi thế trong việc sử dụng hai pha thiết kế so với thiết kế một pha là ở chỗ thiết kế mức đỉnh thứ nhất cung cấp một mốc bổ sung. Ở mốc đó có thể đánh giá và phê duyệt cách tiếp cận thiết kế. Điều này là đặc biệt quan trọng ở các dự án vừa và lớn mà do đó những sai lầm thiết kế chủ yếu phải được xác định càng sớm càng tốt. Khi tìm được sai lầm chủ yếu ở cuối pha thiết kế mức đỉnh thì sẽ dễ dàng sửa nó hơn là nếu tìm thấy nó sau khi toàn bộ thiết kế đã hoàn thành. Qua ví dụ trên ta dễ thấy pha thiết kế tạo cơ sở cho :

- Đường mốc, hệ thống chủ yếu thứ hai của hệ thống
- Thực hiện hệ thống (giai đoạn sau).
- Chương trình phát triển được cập nhật.

Pha thiết kế tạo nên những văn bản sau:

- Đặc tả thiết kế (với những dự án lớn : đặc tả thiết kế mức đỉnh và đặc tả thiết kế chi tiết).
- Kế hoạch tích hợp.
- Đặc tả trường hợp thử nghiệm, mô tả chi tiết mỗi thử nghiệm mức thấp cho từng cá thể.

Văn bản đặc tả thiết kế tạo lập đường mốc chủ yếu thứ hai của dự án. Trường hợp hai pha thiết kế thì đặc tả thiết kế chi tiết được coi là đường mốc chủ yếu của thiết kế và đặc tả thiết kế mức đỉnh được coi là đường mốc thứ yếu.

Vào cuối pha này, nhiều cái chưa biết của dự án lại trở nên đã biết, do đó tạo nên cải tiến đáng kể trong dự toán kế hoạch phát triển các thông số phát triển dự án khác nhau như tài nguyên và lịch tích hợp và các trường hợp thử nghiệm thực cho pha thử nghiệm, giờ đây có thể được lập kế hoạch được kế hoạch phát triển dự án được cập nhật do đây có thể coi rằng ở giai đoạn này kế hoạch đó là đáng tin cậy hơn nhiều. Theo thế, song song với thiết kế của hệ thống, các hoạt động sau đây cũng tiến triển :

- Nền tảng, tích hợp và phát triển được thiết lập bao gồm mọi thiết bị yêu cầu cho phát triển và tích hợp hệ thống .
- Dự toán được cải tiến đáng kể.
- Phân tích rủi ro của dự án được duyệt xét lại và cập nhật.
- Lịch phát triển của hệ thống được cập nhật.

Mọi thông tin trên được đưa vào trong việc sửa đổi mới chủ yếu của kế hoạch phát triển dự án. Pha thiết kế kết thúc với việc ký kết tư liệu đặc tả thiết kế. Điều này thông thường xảy ra trong việc duyệt xét thiết kế chính thức, được gọi là duyệt xét thiết kế có phê phán. Nếu một đặc tả thiết kế mức đỉnh trung gian được chuẩn bị, thì tư liệu này được dừng ở việc duyệt xét thiết kế sơ bộ ban đầu.

Những vấn đề chính của giai đoạn thiết kế liên quan đến:

- Những tranh cãi về thiết kế kỹ thuật
- Khó khăn về nhân sự

- Cung cấp nguồn lực phát triển
- Các quan hệ khách hàng

Những vấn đề thiết kế kỹ thuật liên quan cả đến những gì do các yêu cầu bất khả thi cũng như những gì do những quyết định thiết kế và thực hiện phức tạp. Chính trong pha thiết kế tất cả những vấn đề đó phải được giải quyết vì việc giải quyết chúng sẽ trở nên tốn kém hơn khi dự án tiến triển.

Đôi khi, những vấn đề nhân sự được chuyển từ giai đoạn yêu cầu trước đây. Việc tuyển người cho một số nhiệm vụ là khó hay chậm do đây quan trọng cho người quản lý dự án là phải tìm cách bố trí nhân viên vào mọi vị trí dự án càng sớm càng tốt thậm chí trước những người hiện nay được yêu cầu. Trên thực tế, đây là một thói quen tốt trong việc bắt đầu bố trí nhân viên vào những vị trí trước ngày mà nhân viên đó thực sự tham gia dự án. Nhiều vấn đề liên quan đến việc cung cấp nguồn lực phát triển được tiến hành từ giai đoạn trước song ở pha này của dự án những vấn đề này thường được phối hợp hoặc với những người cung cấp nhân lực bên ngoài và những người cung cấp hoặc thiết bị công ty hiện có mà nó vẫn còn ràng buộc với các dự án khác. Như đã nêu trước đây, nhiều những vấn đề đó thường có thể được giải quyết bằng cho vay hay thuê thiết bị. Dù sao tác động không lợi của những giải pháp tạm thời như thế lại tăng lên khi dự án chuyển sang những giai đoạn phát triển tiên tiến hơn. Trên thực tế, những giải pháp đó, nếu không được xử lý thận trọng, có thể dẫn đến những vấn đề mới :

- Thuê có thể tốn kém và có thể hao phí những số tiền đáng kể ở ngân sách phát triển.
- Vay nợ (hay nợ vay như thường được quan niệm) có thể bị thu lại, làm cho thành viên của đội ngũ phát triển không có phương tiện phát triển cần thiết.
- Những giải pháp thay thế tạm thời có thể dẫn đến chất lượng sút giảm và thời gian phát triển kéo dài. Rút cục điều này lại dẫn đến tình trạng lịch trình bị trượt

Một số vấn đề liên quan khách hàng, người sản xuất vẫn còn có thể xảy ra trong pha thiết kế. Với những yêu cầu giờ đây được chấp thuận. Phần lớn căng thẳng đã giảm đi nhưng khi pha thiết kế kết thúc với việc tổng duyệt chính thức, khách hàng chia xẻ trách nhiệm về thiết kế đó. Điều này tăng thêm dính líu của khách hàng trong pha thiết kế và có thể dẫn đến những bất đồng về hệ thống phải được thực hiện thế nào.

Những loại vấn đề đó được giải quyết tốt nhất bằng cách chỉ định một thành viên đội ngũ (có thể là người quản lý dự án) hành động như mối liên lạc với khách hàng. Điều này giảm những điểm tiếp xúc giữa đội ngũ và khách hàng trong khi vẫn đảm bảo rằng khách hàng có một địa chỉ duy nhất cho mọi yêu cầu và nhận xét.

5.3.2 Lập văn bản đặc tả thiết kế mức tổng thể, mức chi tiết

Hãy xét thí dụ lấy hai chức năng chính trong việc quản lý khách sạn thể hiện dạng văn bản đặc tả thiết kế mức tổng thể, mức chi tiết:

a. Phiếu thuê phòng

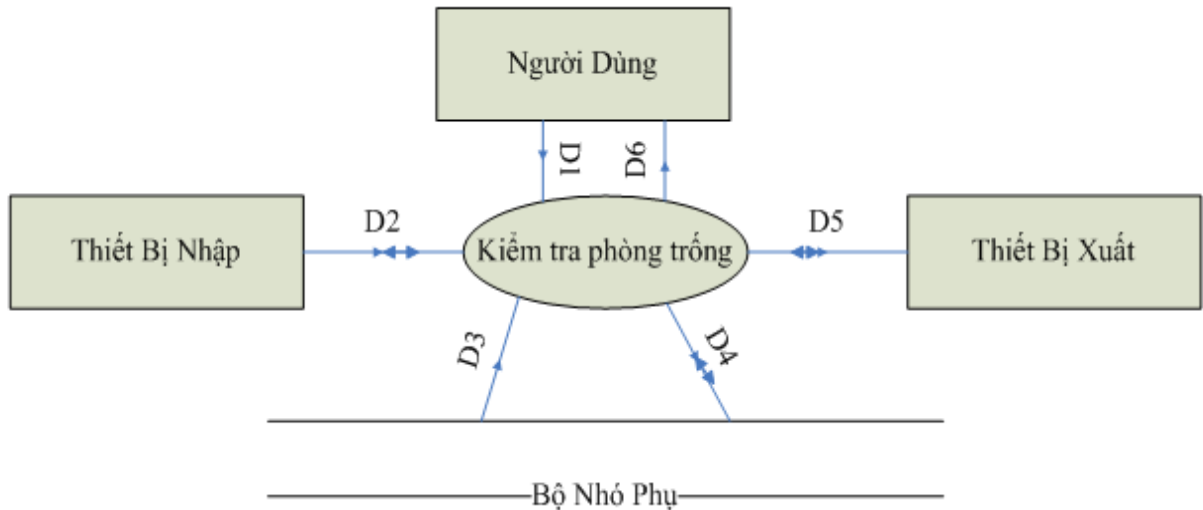
Qui định liên quan: Có 2 loại khách (Nội địa ,nước ngoài). Mỗi phòng có tối đa ba khách.

Biểu mẫu liên quan:

PHIẾU THUÊ PHÒNG				
Phòng:			Ngày bắt đầu thuê Phòng:	
TT	Khách hàng	Loại khách	CMND/Hộ chiếu	Địa chỉ

Sơ đồ luồng dữ liệu:

Sơ đồ luồng dữ liệu cho yêu cầu kiểm tra phòng trống



D1: Mã phòng, Mã loại phòng, Thời gian muốn kiểm tra phòng

D2: Không có

D3: Tình trạng phòng: Ngày bắt đầu thuê, ngày kết thúc, tình trạng phòng (Phòng có người ở, phòng đã được đặt trước).

D4: Không có.

D5: Không có.

D6: Phòng có thể cho thuê , hoặc phòng có thể cho thuê.

* Thuật toán

Bước 1: Nhận D1

Bước 2: Mở kết nối cơ sở dữ liệu

Bước 3: Đọc D3 từ bộ nhớ phụ

Bước 4: Với D1 nhận vào, và D3 nhận được thực hiện kiểm tra: Nếu trong thời gian kiểm tra tình trạng phòng khách với: Phòng có người ở, phòng được đặt trước thì trả về D6 phòng có thể cho thuê. Ngược lại D6 là phòng không thể cho thuê.

Bước 5: Xuất D6 ra màn hình.

Bước 6: Đóng kết nối cơ sở dữ liệu.

Bước 7: Kết thúc

D1 : Phòng, Tên khách hàng, Loại khách hàng, CMND, Địa chỉ, Điện thoại, Giới tính.

D2: Không có.

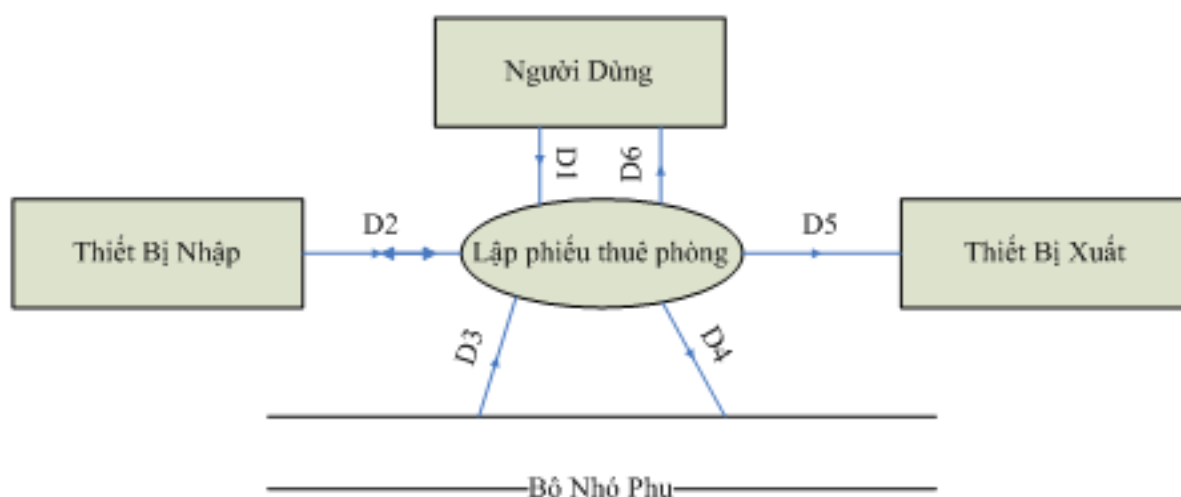
D3: Danh sách các loại khách, và số người tối đa trong 1 phòng.

D4: D1+Ngày thuê phòng.

D5: D4

D6: D4

Sơ đồ luồng dữ liệu cho yêu cầu lập phiếu thuê phòng



* Thuật toán:

Bước 1: Nhận D1 từ người dùng.

Bước 2: Kết nối cơ sở dữ liệu

Bước 3: Đọc D3 từ bộ nhớ phụ

Bước 4: Kiểm tra phòng có còn trống hay không

Bước 5: Tính xem số người tối đa trong 1 phòng.

Bước 6: nếu thỏa mãn bước 3 và 4 thì sang bước 7 ngược lại thì theo bước nào đó.

Bước 7: Lưu D4 xuống bộ nhớ phụ.

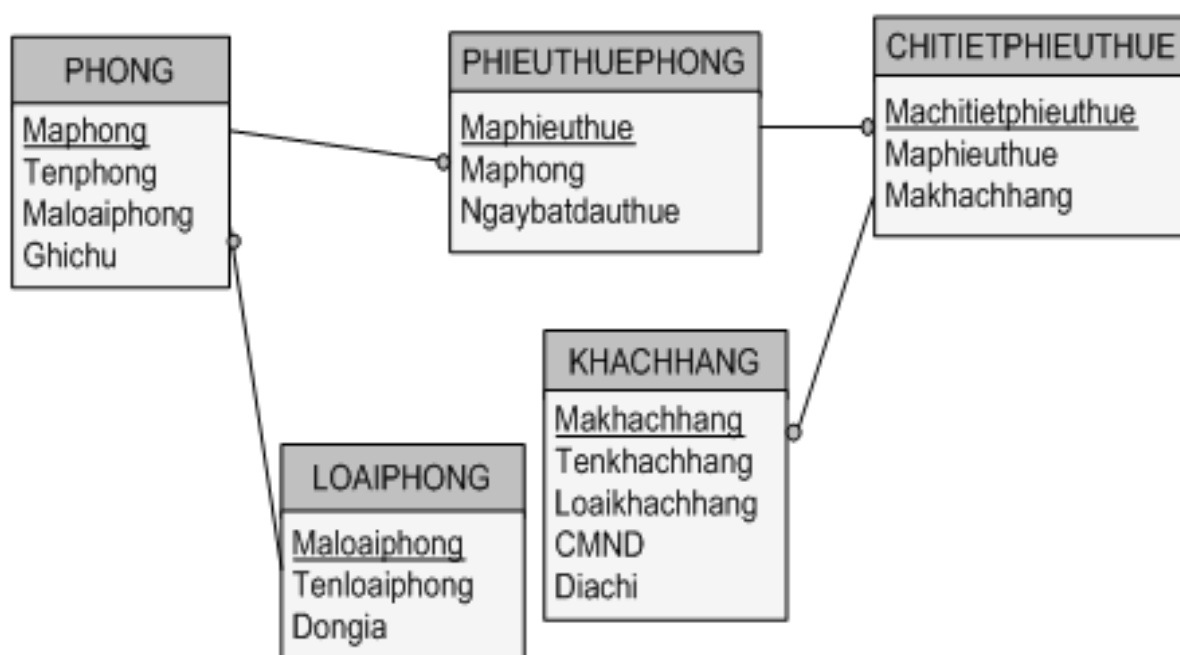
Bước 8: Xuất D5 ra máy in

Bước 9: Xuất D6 về phía người dùng

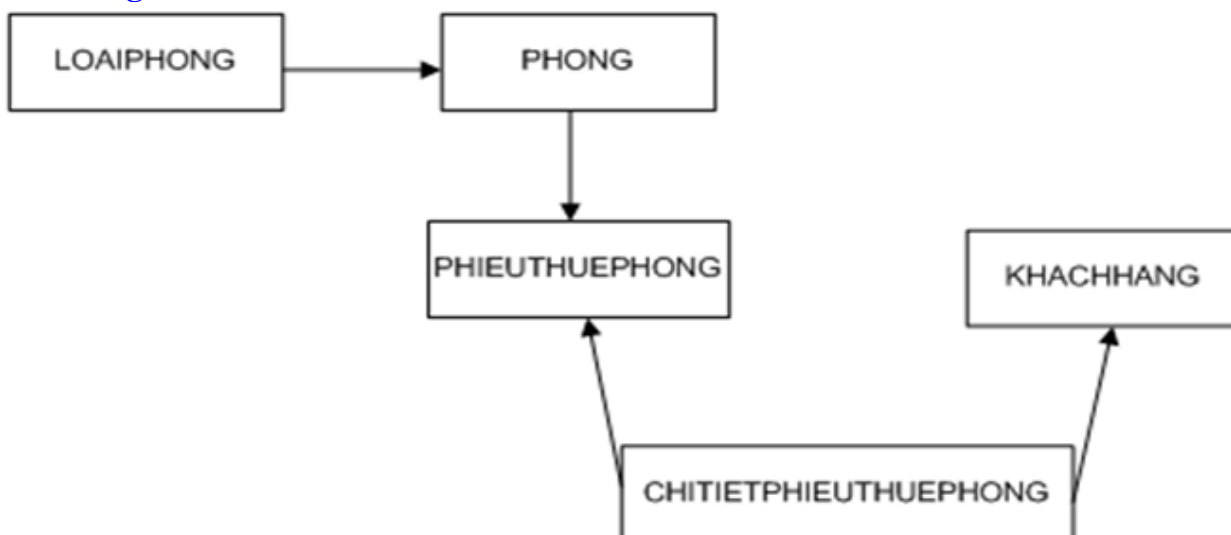
Bước 10: Đóng kết nối cơ sở dữ liệu

Bước 11: Kết thúc

Sơ đồ lớp



Sơ đồ logic



Các thuộc tính mới

Ngaybatdauthue, Tenphong, Loaiphong, CMND, Diachi

Các thuộc tính trừu tượng

Maphieuthue, Machitietphieuthue, Makhachhang

Thiết kế giao diện cho yêu cầu tiếp nhận khách

The interface is titled "TIẾP NHẬN KHÁCH TRỢ". It contains a form with the following fields and controls:

- 1 cmbMaLoaiPhong: Mã Loại Phòng (dropdown)
- 2 txtTenLoaiPhong: Tên Loại Phòng (text)
- 3 bdGiatien: Giá Tiền Phòng (text)
- 4 cmbMaPhong: Mã Phòng (dropdown)
- 5 TenPhong: Tên Phòng (text)
- 6 dtpNgayo: Ngày ở: (date)
- 7 txtSongayo: Số Ngày ở: (text)
- 8 txtDiachi: Họ Tên: (text)
- 9 txtNghenghiep: Địa Chỉ (text)
- 10 txtCMND: Nghề Nghiệp (text)
- 11 cmbLoaiKhach: CMND: (dropdown)
- 11 cmbLoaiKhach: Loại Khách: (dropdown)
- Buttons: Đồng ý (12 cmdDangy), Huỷ Bỏ (13 cmdHuybo), Thoát (14 cmdDang)

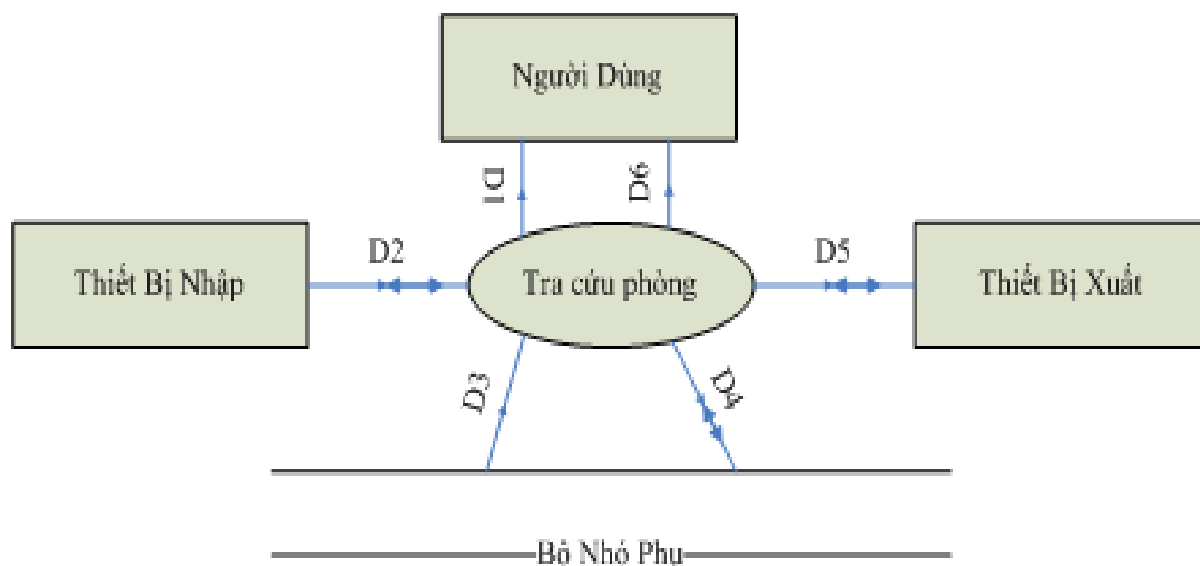
On the right, there is a table titled "DANH SÁCH PHÒNG TRỢ" (15) with columns: Phòng, Loại Phòng, Ngày bắt đầu, Ngày Hết Hạn.

b. Tra cứu phòng

Biểu mẫu liên quan:

TRA CỨU PHÒNG				
TT	Phòng	Loại phòng	Đơn giá	Tình trạng

Sơ đồ luồng dữ liệu:



- D1: Tiêu chuẩn tra cứu phòng trọ: loại phòng, và thời gian ,phong
- D2: không có
- D3: Danh sách các khách trọ thoả mãn đối tượng tìm kiếm (D1)
- D4: không có
- D5: không có
- D6: trả kết quả tìm kiếm cho người dùng biết.

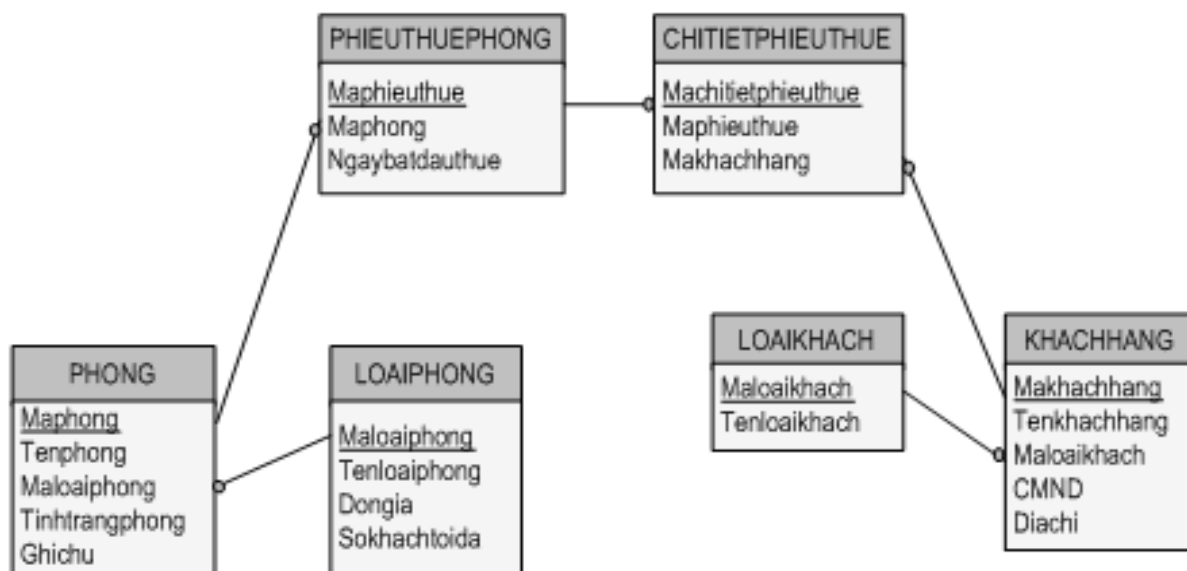
* Thuật toán:

- Bước 1: Đọc D1 từ phía người dùng
- Bước 2: Mở kết nối cơ sở dữ liệu
- Bước 3: Đọc D3 từ bộ nhớ phụ.
- Bước 4: Xuất kết quả tìm kiếm(D6) ra màn hình cho người dùng
- Bước 5: Đóng kết nối cơ sở dữ liệu
- Bước 6: Kết thúc.

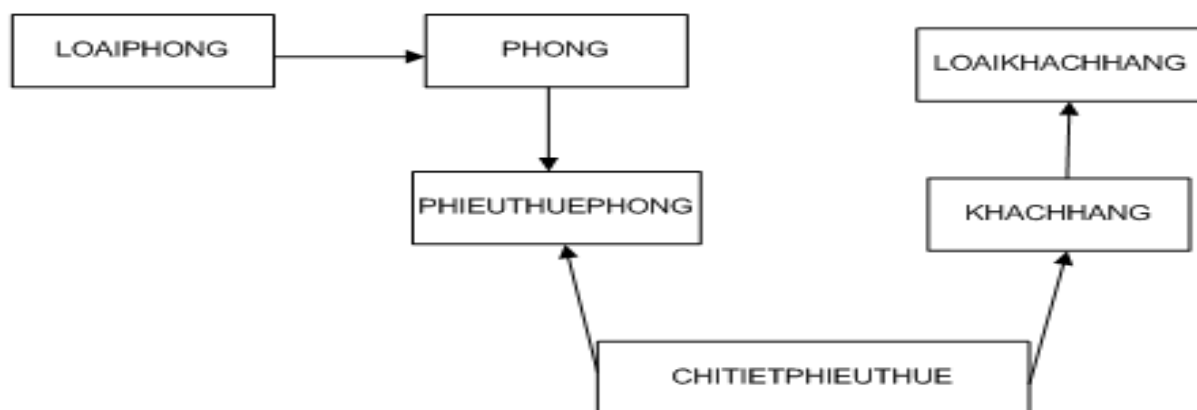
Các thuộc tính mới

Loaiphong, Thoigian, Tinhtrangphong

Sơ đồ lớp:



Sơ đồ logic:



Thiết kế giao diện cho yêu cầu tra cứu phòng.

Màn hình tra cứu theo thời gian (tab tra cứu theo thời gian)

The interface is titled "TRA CỨU PHÒNG". It features two date selection fields: "Từ ngày" (labeled 1) with a dropdown menu (labeled 2) and "Đến ngày" (labeled 3) with a dropdown menu. Below these are three buttons: "TRA CỨU" (labeled 4), "XOÁ", and "ĐÓNG" (labeled 5). A table titled "KẾT QUẢ TÌM KIẾM" displays search results with columns: STT, LOẠI PHÒNG, PHÒNG, NGÀY Ở, NGÀY TRẢ, and TEN KH.

Danh sách các biến cố

STT	Biến Cố	Xử Lý Các Biến Cố
1	Click vào Combobox Từ ngày (dtpTungay) để chọn ngày	Khi người dùng click vào Combobox chọn ngày (dtpTungay), ta ràng buộc chỉ cho phép người dùng chọn ngày nhỏ hơn ngày hiện tại, nếu chọn ngày > ngày hiện tại, thì sẽ tự trả về ngày hiện tại
2	Click vào Combobox Đến ngày (dtpDenngay) để chọn ngày	Khi người dùng click vào Combobox chọn ngày (dtpDenngay), ta ràng buộc chỉ cho phép người dùng chọn ngày \geq Đến ngày, nếu người dùng chọn ngày nhỏ hơn dtpTungay thì ta trả dtpDenngay bằng ngày hiện tại
3	Click vào Button TRA CỨU	Chương trình sẽ dựa vào dtpTungay và dtpDenngay để duyệt vào Cơ sở dữ liệu để

		tìm dữ liệu thoả và xuất ra như lưới như màn hình được thiết kế trên
4	Click vào Button XÓA	Chương trình sẽ xuất hiện 1 message “Bạn có chắc chắn muốn xóa không ?” gồm 2 <i>button Yes/No</i> , nếu nhấn Yes thì lưới bên dưới sẽ trả về rỗng (lưới bây giờ không còn dữ liệu), nếu người dùng nhấn No sẽ không làm gì cả, <i>chú ý ở đây Default là button No</i>
5	Click vào button ĐÓNG	Chương trình sẽ xuất hiện 1 message “Bạn có chắc chắn muốn đóng không ?” gồm 2 <i>button Yes/No</i> , nếu nhấn Yes thì Màn hình tra cứu sẽ được đóng, ngược lại nhấn <i>button No</i> thì chương trình sẽ không làm gì cả, <i>chú ý ở đây Default là button No</i>

Màn hình tra cứu theo Loại phòng (tab tra cứu theo loại phòng)

STT	Biến Cố	Xử Lý Các Biến Cố
1	Click vào combobox mã loại phòng (cmbMaloaiphong)	Khi click vào Combobox mã loại phòng sẽ phát sinh ra sự kiện <code>cmbMaloaiphong_change()</code> sự kiện này có chức năng add tên loại phòng tương ứng vào textbox tên loại phòng (<code>txtTenloaiphong</code>)
2		Text box (<code>txtTenloaiphong</code>) ở trạng thái ReadOnly
3	Click vào Button TRA CỨU	Chương trình sẽ dựa vào <code>cmbMaloaiphong</code> để duyệt vào Cơ sở dữ liệu để tìm dữ liệu thoả và xuất ra như lưới (6) như màn hình được thiết kế trên

4	Click vào Button XÓA	Chương trình sẽ xuất hiện 1 message “Bạn có chắc chắn muốn xóa không ?” gồm 2 <i>button Yes/No</i> , nếu nhấn Yes thì lưới bên dưới (6) sẽ trả về rỗng (lưới bây giờ không còn dữ liệu), nếu người dùng nhấn No sẽ không làm gì cả, <i>chú ý ở đây Default là button No</i>
5	Click vào button ĐÓNG	Chương trình sẽ xuất hiện 1 message “Bạn có chắc chắn muốn đóng không ?” gồm 2 <i>button Yes/No</i> , nếu nhấn Yes thì Màn hình tra cứu sẽ được đóng, ngược lại nhấn <i>button No</i> thì chương trình sẽ không làm gì cả, <i>chú ý ở đây Default là button No</i>

Màn hình tra cứu theo phòng (tab tra cứu theo phòng)

The screenshot shows a form titled "TRA CỨU PHÒNG". It contains two dropdown menus labeled "Mã Loại Phòng" and "Mã Phòng", both with "Please select" text and a dropdown arrow. To their right are two text boxes labeled "Tên Loại Phòng" and "Tên Phòng". Below the form are three buttons: "TRA CỨU", "XÓA", and "ĐÓNG". At the bottom, there is a table titled "KẾT QUẢ TÌM KIẾM" with columns: "STT", "LOẠI PHÒNG", "PHÒNG", "NGÀY Ở", "NGÀY TRẢ", and "TEN KH". The table is currently empty.

STT	Biên Cố	Xử Lý Biên Cố
1	Chọn vào combobox Mã loại phòng (cmbMaloaiphong)	Mã Loại phòng được Load lên từ cơ sở dữ liệu , khi bạn chọn vào cmbMaloaiphong thì sẽ phát sinh ra sự kiện cmbMaloaiphong_Change() sự kiện này sẽ lấy tất cả các Mã phòng tương ứng với loại phòng đó đưa vào cmbMaphong, Tên loại phòng tương ứng đưa vào txtTenphong.
2		Text Box (txtTenloaiphong) ở trạng thái ReadOnly, Tên loại Phòng được Load lên từ cơ sở dữ liệu ứng với mỗi mã loại phòng tương ứng.
3	Chọn vào combobox	Loại phòng được Load lên từ cơ sở dữ liệu

	phòng (cmbMaphong)	ứng với Mã loại phòng, khi bạn chọn Mã loại phòng thì sẽ phát sinh ra sự kiện cmbMaphong_Change() sự kiện này sẽ lấy Tên phòng của mã phòng tương ứng với loại phòng đó đưa vào textbox (txtTenphong)
4		Text Box (txtTenphong) ở trạng thái ReadOnly, Tên Phòng được Load lên từ cơ sở dữ liệu ứng với mỗi mã phòng tương ứng
5	Người dùng click vào Button tra cứu	Chương trình sẽ kiểm tra các Combobox, dựa vào các giá trị trên Combox để tìm kiếm, kết quả sẽ tìm kiếm sẽ trả về vào bảng bên dưới (bảng 8).
6	Người dùng click vào Button Xóa	Người dùng bấm vào nút xóa sẽ xuất hiện 1 message thông báo bạn có chắc chắn muốn xóa không gồm 2 button Yes/No, nếu nhấn Yes thì các Combobox và lưới bên dưới (8) sẽ được xóa. Ngược lại thì không làm gì cả, lưu ý ở đây Default là button No
7	Người dùng click vào Button Đóng .	Người dùng bấm vào nút xóa sẽ xuất hiện 1 message thông báo bạn có chắc chắn muốn đóng không gồm 2 button Yes/No, nếu nhấn Yes thì các màn hình tra cứu sẽ đóng., Ngược lại thì không làm gì cả, lưu ý ở đây Default là button No

5.3.3 Kế hoạch nhất thể hóa

Nên cho tích hợp và thử nghiệm là tích hợp. Chuẩn bị một kế hoạch tích hợp các modul phần mềm sao cho tích hợp có thể bắt đầu càng sớm càng tốt. Các phương pháp xây dựng kế hoạch tích hợp là:

- Xây dựng kế hoạch tích hợp từ trên xuống
- Xây dựng kế hoạch từ dưới lên
- Xây dựng kế hoạch từ trong ra ngoài

Hầu hết các dự án kế hoạch nhất thể hóa đều được xây dựng ngay trong cuối giai đoạn phân tích bởi vì cuối giai đoạn này dự án đã có hồ sơ nghiên cứu hiện trạng và các yêu cầu đã được phân tích sơ bộ và được chỉ số hóa theo nội dung từ điển. Tuy nhiên việc tích hợp các modul và nhất thể hóa chương trình nằm xen kẽ trong giai đoạn thiết kế, thực hiện và thử nghiệm. Điều quan trọng là kết quả mong đợi của kế hoạch cho hoạt động này là:

- Sự kết dính
- Sự ghép nối

Nói chung sự kết dính và sự ghép nối có liên quan mật thiết với nhau điều này phụ thuộc vào việc thiết kế hướng đối tượng hay thiết kế hướng chức năng. Độc giả có thể tham khảo chi tiết ở giáo trình Công nghệ phần mềm ĐHQG Hà Nội.

5.3.4 Đặc tả các thử nghiệm cho từng cá thể

Những con số ở bảng 5.12 lấy từ sơ đồ chung có nhân mạnh đến qui hoạch phần mềm (*yêu cầu và thiết kế*) và thử nghiệm. Những hệ thống xử lý dữ liệu thương mại, trừ một số ngoại lệ, vẫn còn phải tiêu phí một khối lượng đáng kể thời gian phát triển trong pha lập trình và thử nghiệm đơn vị. Các hệ thống thời gian thực thường phức tạp hơn nhiều và có thể bao gồm sự tích hợp bao quát phần cứng/phần mềm. Điều này thông thường đòi hỏi qui hoạch nhiều hơn và tích hợp cùng thử nghiệm nhiều hơn.

Bảng 5.12 Phần trăm dự tích về thời gian hao phí trong pha phát triển phần mềm chủ yếu.

	Qui hoạch	mã và thử nghiệm đơn vị	nhật thể và thử nghiệm
Xử lý dữ liệu thương mại	25%	40%	35%
Các hệ thống thời gian thực	35%	25%	40%
Các hệ thống quân sự	40%	20%	40%

Đặc tả trường hợp thử nghiệm, mô tả chi tiết mỗi thử nghiệm mức thấp cho từng cá thể. Vào cuối pha này, nhiều cái chưa biết của dự án lại trở nên đã biết, do đó tạo nên cải tiến đáng kể trong dự toán kế hoạch phát triển các thông số phát triển dự án khác nhau như tài nguyên và lịch tích hợp và các trường hợp thử nghiệm thực cho pha thử nghiệm, giờ đây có thể được lập kế hoạch được kế hoạch phát triển dự án được cập nhật do đây có thể coi rằng ở giai đoạn nay kế hoạch đó là đáng tin cậy hơn nhiều.

Theo thế, song song với thiết kế của hệ thống, các hoạt động sau đây cũng tiến triển :

- Nền tảng, tích hợp và phát triển được thiết lập bao gồm mọi thiết bị yêu cầu cho phát triển và tích hợp hệ thống.
- Dự toán được cải tiến đáng kể.
- phân tích rủi ro của dự án được duyệt xét lại và cập nhật.
- Lịch phát triển của hệ thống được cập nhật.

Mọi thông tin trên được đưa vào trong việc sửa đổi mới chủ yếu của kế hoạch phát triển dự án. Thử nghiệm đơn vị được người lập trình tiến hành trên mỗi modul cá thể ngay sau khi modul đó được mã hoá.

5.3.5 phân tích các rủi ro

Xử lý rủi ro trong đầu tư cho CNTT: Có nhiều dự án CNTT được dự báo là thành công khi thẩm định bằng phân tích kinh tế như trên, nhưng khi triển khai dự án lại có thể gặp nhiều trục trặc, gọi là các rủi ro. Vì vậy, ngoài phân tích về kinh tế, cần thực hiện quản lý rủi ro đối với các dự án.

Quản lý rủi ro bao gồm: Xác định và phân tích các dạng rủi ro, đưa ra giải pháp để giảm thiểu rủi ro và điều chỉnh ước toán về lợi ích và chi phí cho phù hợp.

Có hai cách ứng phó với rủi ro: Khắc phục hậu quả khi đã xảy ra và dự báo rủi ro để có các biện pháp phòng ngừa trước. Cần kết hợp các biện pháp này. Phòng ngừa rủi ro là cách hiệu quả hơn, nếu như thực hiện được.

Để phòng ngừa rủi ro: Đầu tiên phải có nhận thức khái lược về chúng, dựa trên thông tin thu được từ các bước thực hiện trước đó. Kết quả, đưa ra một bảng liệt kê và phân loại sơ bộ các rủi ro có thể gặp của dự án trong các giai đoạn triển khai khác nhau. Tiếp đến, các rủi ro này được phân tích “*nhạy cảm*”, tức xác định xác suất xảy ra và phạm vi ảnh hưởng nếu xảy ra của chúng, để điều chỉnh giải pháp cũng như các dự báo ảnh hưởng kinh tế của dự án. Vì vậy, phân tích rủi ro cần được đưa vào các kết quả đánh giá dự án.

Phân tích rủi ro có tầm quan trọng đặc biệt trong triển khai các dự án HTTT, nhất là các dự án lớn và phức tạp. Việc phân tích rủi ro cần chuẩn bị trước những vấn đề dự liệu đòi hỏi đánh giá mỗi vấn đề nhằm:

1. Dự toán xác suất vấn đề sẽ xảy ra.
2. Dự toán tác động của vấn đề tới dự án.
3. Quy cho mức độ nghiêm trọng của vấn đề.

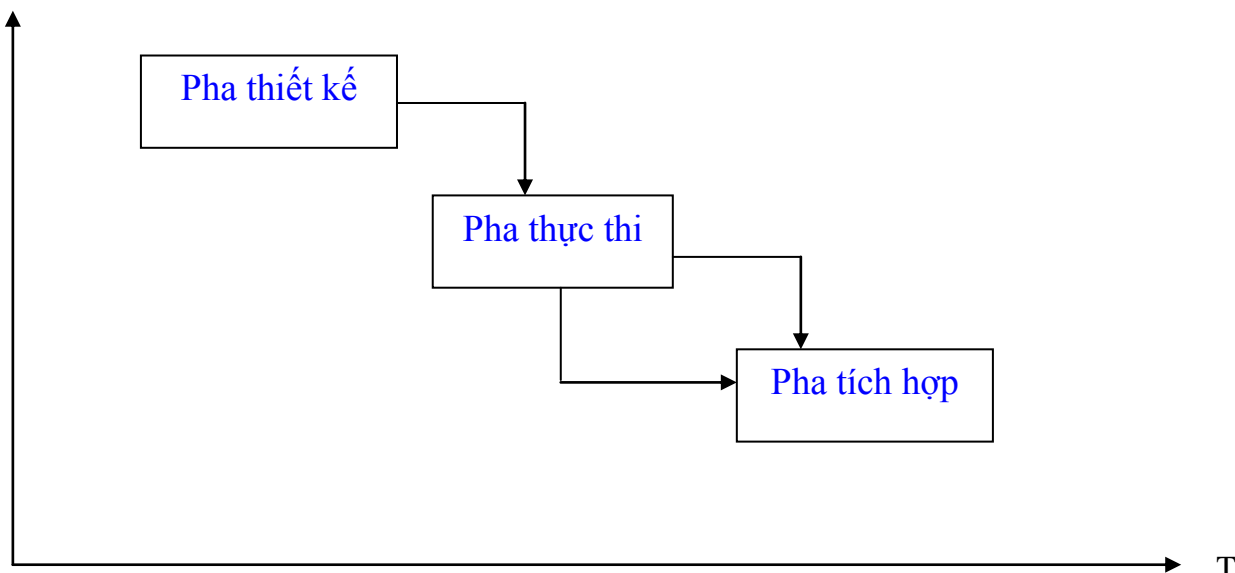
Một cách đơn giản và hiệu quả để tính mức độ nghiêm trọng của mỗi vấn đề được dự liệu là:

1. Gán một số kỳ vọng giữa 1 và 10 căn cứ ở xác suất là vấn đề sẽ xảy ra, với 10 điểm xác suất cao, và 1 xác suất thấp nhất (thí dụ nhân xác suất với 10).
2. Gán một số giữa 1 và 10 căn cứ vào tác động của vấn đề với dự án trong đó 10 biểu thị tác động cao và 1 tác động thấp.
3. Nhân trị giá có được ở bước (1) với giá trị có được ở bước (2) để tính mức độ nghiêm trọng cho vấn đề. Dưới đây là một ví dụ:

Bảng 5.13: Ví dụ về cách tính mức độ nghiêm trọng.

TT	Vấn đề	Kỳ vọng	Tác động	Nghiêm trọng
1	Chậm giao máy tính để phát triển	6	5	30
2	Bộ nhớ không đủ	4	2	8
3	Không có chuyên gia hệ điều hành	5	2	25
4	Thời gian đáp ứng của hệ thống quá chậm	5	3	15
5	Thuyên chuyên nhân sự cao	5	8	40
6	Truyền thông quá chậm	2	8	16
7	Chậm giao hệ thống phụ cơ sở dữ liệu	3	9	27

5.4 QUẢN LÝ GIAI ĐOẠN THỰC HIỆN



Hình 5.14 : Sự gối đầu của các pha

5.4.1 Phát triển mã nguồn

Trong pha thực hiện, các mô đun phần mềm được mã hoá và các thử nghiệm đơn vị ban đầu được tiến hành. Thử nghiệm đơn vị được người lập trình tiến hành trên mỗi mô đun cá thể ngay sau khi mô đun đó được mã hoá. Sau đó các mô đun được bộ phận kiểm tra chất lượng phần mềm phê chuẩn và được đệ trình bộ phận kiểm tra cấu hình. Sau đó bộ phận kiểm tra cấu hình chuyển các mô đun để hợp nhất.

Đặc tả thiết kế chi tiết và cấu trúc tốt dẫn đến việc lập mã tương đối trơn chu và thẳng băng (Điều này được thuyết minh qua nhiều cố gắng để phát triển các máy sinh mã tự động tạo lập mã từ đặc tả thiết kế, điều này đúng như vậy hầu hết các công cụ phân tích thiết kế đều có thể sinh mã nguồn). Theo thể việc lập mã (hay lập trình) thoát đầu được nhận thức là đồng nghĩa với phát triển phần mềm, nay đã trở thành một pha riêng rẽ trong chu trình phát triển phần mềm. Trên thực tế pha thực hiện lại cũng không là pha dài nhất. Một kinh nghiệm thông thường đề dự tính các pha của chu trình phát triển phần mềm sử dụng cách phân chia theo tỉ lệ 40 – 20 - 40 công sức và thời gian (bảng 5.12).

Điều này có nghĩa khoảng 40% thời gian được giành cho đặc tả (*yêu cầu và thiết kế*) 20% cho thực hiện (*lập trình và thử nghiệm đơn vị*) và 40% cho tích hợp và thử nghiệm xu hướng chung là tìm cách giảm 20% dành cho thực hiện trong khi tăng 40% dành cho đặc tả? Một khía cạnh của lập luận đằng sau tiếp cận đó là các giai đoạn phát triển sớm hơn ít tốn kém hơn các giai đoạn sau. Các pha đặc tả thường có nhân lực thấp và ít thiết bị phát triển hơn pha thực hiện. Cũng vậy như thường đã chứng minh khi công sức được giành nhiều hơn cho yêu cầu và thiết kế, pha tích hợp dễ dàng hơn và hiệu quả hơn.

Pha thực hiện là cầu nối các pha thiết kế và tích hợp hệ thống và thường gối đầu rõ rệt với mỗi một trong hai pha đó (hình 5.14). Tình trạng gối đầu sẽ thường xảy ra khi nhiều bộ phận của thiết kế hệ thống được thực hiện tương đối nhanh để

lại một số vấn đề thiết kế bỏ ngỏ một thời gian khá. Trong những trường hợp đó gói đầu có thể rút ngắn đáng kể lịch phát triển.

Tình trạng gói đầu của các pha thiết kế và thực hiện đòi hỏi phải rất thận trọng trong việc đảm bảo chỉ những mô đun thiết kế hoàn chỉnh được phê chuẩn để thực hiện sớm. Có rủi ro là mọi thay đổi chậm sau này về thiết kế của những mô đun đó có thể đòi hỏi phải lập mã lại. Theo thế lãng phí tài nguyên. Cũng có rủi ro là thiết kế thay đổi và mã không thay đổi. Dù sao những rủi ro đó thường vẫn nắm bắt được tốt. Với qui hoạch và không chế kiểm tra cấu hình tốt những vấn đề đó có thể được khắc phục

5.4.2 Xây dựng kế hoạch bảo trì

Mặt khác của pha thực hiện, tình trạng gói đầu của lập mã và ích hợp thường ít rủi ro và nếu được qui hoạch đúng đắn có thể là một sự tiết kiệm thời gian tuyệt hảo. Trình tự thực hiện của các mô đun phải được qui hoạch tốt đảm bảo chúng được đưa ra theo trình tự yêu cầu để tích hợp. Điều này cũng có nghĩa những sai lầm trong thực hiện có thể định vị trong quá trình tích hợp và được hồi tiếp lại pha thực hiện.

Pha thực hiện bao gồm những hoạt động chính sau đây :

- Phát triển mã phần mềm.
- Chuẩn bị cho sự tích hợp và thử nghiệm hệ thống (giai đoạn sau).
- Phát triển kế hoạch bảo trì.

Ngoài mã hiện tại được viết một số tư liệu khác được phát triển trong pha này bao gồm :

- Sổ ghi của người lập trình dẫn ghi lại các quyết định lập mã những thử nghiệm đơn vị và giải pháp cho những vấn đề thực hiện .
- Kế hoạch bảo trì và thu thập tư liệu, kể cả tư liệu cốt yếu cần thiết cho việc bảo trì hệ thống.
- Các bản ban đầu về tư liệu cho người dùng, kể cả sổ tay tham khảo và hướng dẫn cho người điều hành. Trong pha thực hiện, dự án bước vào thời kỳ hoạt động khẩn trương.

Nhiều hoạt động khác diễn ra bao gồm:

- Pha thiết kế được hoàn tất sớm trong quá trình pha thực hiện.
- Nền tích hợp được định vị và việc tích hợp bắt đầu.
- Nền của thử nghiệm (*môi trường và thiết bị thử nghiệm*) được định vị để chuẩn bị cho việc thử nghiệm hệ thống
- Các tình huống rủi ro có thể được trở thành hiện thực và khi đó các kế hoạch dự phòng bắt buộc được đưa vào hoạt động.
- Kế hoạch dự án được duyệt lại và cập nhật.

5.4.3 Lập sổ ghi các giải pháp cho các mô đun thử nghiệm

Lấy lại ví dụ quản lý khách sạn ở trên điều dễ thấy mỗi các modul có được trình bày bao gồm:

1. Tài liệu dùng cho thiết kế

Các qui định liên quan.
Văn bản đặc tả sơ bộ/Chi tiết.

2. Các loại sơ đồ

Sơ đồ luồng dữ liệu
Sơ đồ lớp
Sơ đồ logic

3. CSDL

Các thuộc tính.
Các thuộc tính mới.
Thuộc tính trừu tượng.

4. Giao diện – Báo cáo

Giao diện - Biểu mẫu
Danh sách các biến cố hoặc hành vi.

Đây chính là quá trình thử nghiệm đầu tiên, điều quan trọng ở đây là nhà thiết kế thường bây giờ cũng là một lập trình viên ở mức đỉnh và thường thì bỏ quên việc ghi chép tỉ mỉ kết quả của quá trình thiết kế: Đầu vào, đầu ra, tên modul và những giới hạn cho phép thực hiện modul để làm tài liệu cho bàn giao cho giai đoạn tiếp.

5.4.4 Xây dựng trợ giúp cho người dùng cuối

Xây dựng trợ giúp có một tầm quan trọng không thể không quan tâm, trong phân tích rủi ro thì đây cũng được coi là một rủi ro kế hoạch đôi phó bất ngờ được lập từ trước. Đối với những dự án nhỏ hoặc thiết kế không chuyên thì phần này chẳng bao giờ được quan tâm, mà thường nhầm lẫn rằng quá trình chuyển giao và đào tạo nhà phát triển đã thực hiện rồi. Hầu hết các phần mềm do các công ty của Việt Nam hiện nay đều rất yếu phần này và đây là một lý do mà các phần mềm này không tồn tại được lâu trên thị trường. Có một vài cách xây dựng trợ giúp cho người dùng cuối;

- Trợ giúp ngữ cảnh.
- Trợ giúp chức năng.
- Tạo kênh thông tin trợ giúp.

5.4.5 Nhất thể hóa từng phần, tất cả, rà soát lỗi mã khi nhất thể hóa

Đề tạo ra chất lượng cho một hệ thống. Pha này thường được phân làm hai pha riêng biệt và khá song song :

Pha tích hợp và pha thử nghiệm. trong những dự án lớn, sự phân chia này thường là cần thiết đặc biệt khi việc thử nghiệm được một nhóm riêng biệt độc lập tiến hành (*đội ngũ thử nghiệm độc lập*) làm việc sát cánh với những người sản xuất. trong những dự án nhỏ, điều hợp lý đằng sau sự phối hợp của hai hoạt động vào một pha là sự tích hợp không thể thành công nếu không có thử nghiệm rộng rãi được tiến hành song song. Do vậy nếu hai hoạt động đó được cùng một đội ngũ thực hiện, thường tốt nhất là phối hợp chung trong một pha.

Có nhiều kỹ thuật và phương pháp tích hợp:

- *Từ trên xuống*

Tiếp cận trên xuống đòi hỏi cốt lõi của hệ thống (*thường là các modul điều hành trung tâm*) phải được thực hiện trước. Rồi chúng được phối hợp một hệ thống cực nhỏ sử dụng những thủ tục chưa có nội dung thay cho các modul cha được thực hiện. Những những thủ tục rỗng đó của mã trả lại những giá trị cố định và chẳng có tính logic gì cả, thường được gọi là cuồng. Sau đó các cuồng được thay dần bằng các modul thực, theo một cách xây dựng tăng tiến có kế hoạch tốt của hệ thống sao mỗi lần từ hệ thống đưa ra mới, lại tăng thêm nhiều chức năng hơn.

- *Từ dưới lên*

Tiếp cận từ dưới lên bắt đầu từ các modul cá thể ở cấp thấp nhất (ví dụ : - bộ điều khiển đầu vào đầu ra, bộ định khuôn, dạy thao tác, dữ liệu, bộ đối thoại người dùng, máy v.v..) và dần dần gộp chúng thành những nhóm mỗi lúc mỗi lớn hơn cho đến khi toàn bộ hệ thống được ghép lại. Cách tiếp cận dưới lên ít khi được khuyến dùng làm tích hợp dễ hiểu, trong hầu hết trường hợp cách tiếp cận trên xuống dễ hơn và tự nhiên hơn. Dù sao trên thực tế, hầu hết các chiến lược tích hợp hệ thống thành công là một phối hợp của cách tiếp cận trên xuống và lác đác đôi chỗ theo cách tiếp cận từ dưới lên.

Một cách tiếp cận lý thú khác được gọi là

- *Từ trong ra*

Tích hợp từ trong ra (*indide out integration*) là thông thường trong sự phát triển của các hệ cơ sở dữ liệu lớn khi các cấu trúc file nội bộ được xây dựng trước sau đó là logic xử lý dữ liệu và cuối cùng giao diện với con người. Tiếp cận này là tốt nhất khi hệ thống về mặt logic là các tầng chức năng kế tiếp nhưng điểm yếu chính của nó là giao diện với con người lại thường là được tích hợp cuối cùng. Điều này có thể đòi hỏi viết mã thử nghiệm tạm thời để đầu ra có thể được duyệt xét lại. Do đây thử nghiệm thường chậm hơn và khó hơn so với tiếp cận từ trong ra. Việc thử nghiệm bắt đầu bằng việc tích hợp và tiếp tục tới khi cuối cùng hệ thống được bàn giao cho khách hàng.

5.4.6 Xây dựng kế hoạch dự phòng cho các rủi ro

Bảng 5.15. Ví dụ về bảng ngẫu nhiên

TT	Vấn đề	Nghiêm trọng	Kế hoạch đối phó bất ngờ	Người theo dõi
1	Thuyên chuyển nhân sự cao	40	Tiền thưởng hoàn thành dự án thắng lợi	Hoàng Tùng
2	Chậm giao máy tính để phát triển	30	Yêu cầu làm ca đêm về hệ thống phát triển của dự án khác	Bùi Thị Phương
3	Chậm giao hệ thống phụ cơ sở dữ liệu	27	Thiết kế một mô phỏng hệ thống phụ cơ sở dữ liệu dùng để tích hợp	Lê Văn Nghiệp
4	Không có chuyên gia hệ điều hành	25	Bố trí chuyên gia hệ điều hành ngoài công ty và thuê làm tư vấn	Nguyễn Văn Thiện

5	Truyền thông quá chậm	16	Hợp đồng với công ty đã phát triển gói truyền thông nhị phân thích nghi cho gói với dự án này	Quách công Nghĩa
6	Thời gian đáp ứng của hệ thống quá chậm	15	Cho vào điều khoản thoả thuận nâng cấp CPU trong hợp đồng mua máy tính	Trần Duy Khang
7	Bộ nhớ không đầy đủ	8	(không xem xét)	

Thực hiện các kế hoạch đối phó bất ngờ.

Các kế hoạch đối phó bất ngờ được thực hiện ở một trong những trường hợp sau:

1. Vấn đề dữ liệu diễn ra hay sắp xảy đến đến nơi.
2. Kế hoạch đối phó bất ngờ đòi hỏi được chuẩn bị trước.

Nói chung, các kế hoạch đối phó bất ngờ có thể được nhìn nhận theo như các kế hoạch, hành động được xếp vào ngăn kéo để phòng khi dùng đến sau này. Dù sao, trong vài trường hợp, kế hoạch được thực hiện trước khi vấn đề dự liệu xảy ra như phát triển một bộ mô phỏng trường hợp việc giao một hợp phần quyết định bị chậm trễ. Sau đó, nếu hợp phần được giao đúng hạn thì bộ mô phỏng đó có thể bị bỏ đi.

5.4.7 Quyết toán các khâu đã hoàn thiện.

Đối với khách hàng: Đến giai đoạn này nhà phát triển dự án cần phải yêu cầu được thanh toán dự án theo hợp đồng (*một phần của hợp đồng*) để trả chi phí cho các hạng mục phát triển.

Đối với nhà phát triển: Thực hiện công tác thẩm định đối với các hạng mục đã phát triển. Phương pháp xác minh các vấn đề có lẽ là được đề nghị như sau:

Các hoạt động SQA (*nhóm đảo bảo chất lượng phần mềm*) như việc giám định duyệt lại, chọn lựa và kiểm tra cấu hình tùy thuộc ở cách mà SQA được thích ứng với dự án đặc trưng. Qui mô của dự án thường là yếu tố quyết định. Những hướng dẫn sau bàn đến một số thông số cần xem xét cho các loại dự án khác nhau khi lập kế hoạch SQA.

⇒ Trong những dự án nhỏ, có nhiều hoạt động SQA có thể do người quản lý dự án thực hiện. Điều này bao gồm tổ chức và giám định duyệt lại và kiểm toán, đánh giá và lựa chọn các công cụ phát triển, lựa chọn và vận dụng các tiêu chuẩn.

⇒ Các thủ tục thử nghiệm và thử nghiệm luôn là tốt nhất khi được một đội riêng biệt độc lập tiến hành (*xin tham khảo chi tiết các tiêu chuẩn có trong các giáo trình Công nghệ phần mềm và Quản lý dự án phần mềm ĐHQG Hà Nội*). Quyết định là chỗ giám định các hoạt động thử nghiệm nào có thể được ủy thác cho SQA tùy thuộc ở nhiều yếu tố, kể cả sự độc lập của đội SQA, qui mô của dự án và tính phức tạp của dự án.

Khi việc thử nghiệm được đội thử nghiệm độc lập tiến hành thì đỉnh lú của SQA sẽ là tối thiểu. Trong hầu hết các trường hợp khác, trách nhiệm của đội SQA sẽ là lập kế hoạch và giám định thử nghiệm của hệ thống.

⇒ Về hướng dẫn chung, thường không nên để một thành viên của đội phát triển thực hiện SQA. Dù sao, các dự án nhỏ thường khi không thể xác minh chi phí của một kỹ sư được bố trí SQA. Vấn đề này có thể được giải quyết khi có một kỹ sư SQA duy nhất chịu trách nhiệm cho hai hay ba dự án nhỏ (*với mỗi dự án tài trợ phần đóng góp của mình về dịch vụ SQA*).

Khả năng kiểm tra chất lượng phần mềm hiệu quả trực tiếp với chất lượng của đặc điểm yêu cầu của phần mềm. Kiểm tra chất lượng đòi hỏi chi tiết không mơ hồ của càng nhiều đặc điểm yêu cầu của chất lượng phần mềm càng tốt.

5.5 QUẢN LÝ GIAI ĐOẠN THỬ NGHIỆM

5.5.1 Thực hiện việc nhất thể hóa

⇒ **Thử nghiệm tích hợp** : Thực hiện bằng các máy tích hợp của hệ thống.

⇒ **Thử nghiệm độc lập** : Thực hiện bởi nhóm thử nghiệm bên ngoài để đảm bảo thử nghiệm khách quan không thiên lệch.

⇒ **Thử nghiệm lắp đặt**: Điều này bao gồm thử nghiệm hoàn thiện chung. Đòi hỏi khi mà một phiên bản của hệ thống đầu tiên được lắp đặt trong một môi trường điều hành. Nhiều hệ thống bao gồm một bộ thử nghiệm lắp đặt để đảm bảo hệ thống được lắp đặt thành công.

5.5.2 Thực hiện các thử nghiệm An pha, Bê ta, Thử nghiệm thu

Các thử nghiệm này tiến hành trong hệ thống trong một môi trường thực (không phải thử nghiệm trong phòng thí nghiệm test). Các thử nghiệm phần mềm bao gồm:

⇒ **Thử nghiệm đơn vị**

⇒ **Thử nghiệm hợp nhất**

⇒ **Thử nghiệm hệ thống phụ**

⇒ **Thử nghiệm hệ thống**

⇒ **Thử nghiệm suy thoái**

Và các thử nghiệm quan trọng nhất

⇒ **Thử nghiệm alpha**: Thử nghiệm hệ thống không có dữ liệu sống.

⇒ **Thử nghiệm beta**: Thử nghiệm hệ thống có sử dụng dữ liệu sống có giám sát thường xuyên để chỉnh lý mọi vấn đề có thể phát sinh.

⇒ **Thử nghiệm nghiệm thu**: Đây là cột mốc cuối cùng của dự án và sự hoàn tất thắng lợi của nó có nghĩa là sự nghiệm thu của khách hàng về sản phẩm được phát triển.

Vào lúc kết thúc pha tích hợp và thử nghiệm mọi tư liệu phải đầy đủ và sẵn sàng để giao bao gồm :

⇒ Tài liệu bảo trì.

⇒ Tài liệu của người dùng cuối cùng

⇒ Mọi tài liệu phát triển cập nhật.

⇒ Tài liệu thử nghiệm và báo cáo thử nghiệm.

5.5.3 Lập các báo cáo thử nghiệm

Người quản lý dự án cần thông tin khách quan. thông tin như thế thường có thể được thu thập từ những báo cáo của các đội thử nghiệm độc lập như:

⇒ Các báo cáo bảo hiểm chất lượng có các tiêu chí sau:

☞ Toàn bộ khía cạnh và đặc điểm của một sản phẩm phần mềm có khả năng đáp ứng nhu cầu đề ra: Chẳng hạn thích hợp với các đặc điểm kỹ thuật.

☞ Mức độ mà phần mềm có được sự phối hợp mong muốn của các thuộc tính.

☞ Mức độ mà khách hàng hay người dùng nhận thức là phần mềm đáp ứng được các kỳ vọng hợp thể của họ.

☞ Các đặc điểm hợp thể của phần mềm xác định mức độ mà phần mềm sử dụng sẽ đáp ứng được các kỳ vọng của khách hàng.

☞ Khả năng của sản phẩm phần mềm để đáp ứng những yêu cầu đặc dụng của nó. Chất lượng phải được đo theo nghĩa kỳ vọng của khách hàng. Người quản lý dự án phải coi việc thực hiện kiểm tra chất lượng là một quá trình khách quan, mang tính hệ thống.

⇒ Báo cáo thử nghiệm gồm bốn thành phần:

☞ Qui hoạch thử nghiệm.

☞ Mô tả thử nghiệm.

☞ Tiến hành thử nghiệm.

☞ Kết quả thử nghiệm.

☞ Tiêu chuẩn thử nghiệm.

☞ Các yếu tố đó về mức độ chi tiết và thông tin tư liệu yêu cầu.

Mặc dù báo cáo đúng là những nguồn thông tin có ích, nhưng không có gì có thể thay thế tiếp xúc trực tiếp giữa người quản lý dự án và nhân lực phát triển. Nhưng cuộc nói chuyện thường xuyên không chính thức với những người sản xuất là những nguồn thông tin tuyệt hảo đặc biệt khi được tiến hành trong bầu không khí không chính thức (*và không phải trong văn phòng của người quản lý dự án*).

Người quản lý dự án phải duy trì cảnh giác thường xuyên với sai lầm thường được gọi là "hội chứng 90/50" khẳng định là "phải mất 50% thời gian để hoàn thành 90% công việc và 50% thời gian phụ thêm để hoàn thành số 10% công việc còn lại". Điều này có nghĩa là những người sản xuất dự án sẽ bắt đầu khoe khoang sớm là họ đã "hầu như xong" nhiệm vụ của họ, bất hạnh là có sự khác biệt lớn giữa "hầu như xong" với "xong".

Xong nhiệm vụ - viết tài liệu và thanh toán một vài vấn đề cuối cùng - thường lại lâu hơn dự kiến của những người phát triển. Có điều này do những hoạt động đó tạo ra một số kết quả nhãn tiền và những Người phát triển có xu hướng (sai lầm) kết hợp công việc với kết quả. Do vậy, những người quản lý có thể có được nhiều thông tin hơn ở những người sản xuất khi hỏi họ dự tính bao lâu nữa thì xong công việc chứ không hỏi họ đã hoàn thành được bao nhiêu công việc.

5.5.4 Đánh giá mức độ do các các lỗi sinh ra. Giải pháp khắc phục

Khi ủy thác SQA (*nhóm đảo bảo chất lượng phần mềm*) bao gồm các hoạt động kiểm tra cấu hình, các nguồn yêu cầu cũng bao gồm cả các nguồn yêu cầu cho kiểm tra cấu hình. Việc hòa đồng SQA và kiểm tra cấu hình không phải không

phổ biến và có thể loại trừ một số ủy nhiệm và hoạt động phải nhân lên. Nên nhớ là với những dự án nhỏ, việc hòa đồng hai nhóm đơn giản có thể có nghĩa là phó thác cả hai nhiệm vụ cho cùng một người.

Mặc dù nhiều công cụ là phổ biến cả cho kiểm tra chất lượng và kiểm tra cấu hình, chỉ có ít công cụ được đặc biệt chỉ định cho kiểm tra chất lượng. Sau đây là một số công cụ hỗ trợ chung có thể có ích trong việc hỗ trợ các hoạt động của SQA:

- ⇒ Tiện ích tư liệu.
- ⇒ Công cụ thiết kế phần mềm.
- ⇒ Hỗ trợ tìm sai (Chương trình gỡ rối).
- ⇒ Các bộ xử lý trước được cấu thành.
- ⇒ Bộ so sánh hồ sơ.
- ⇒ Những bộ phân tích cơ cấu.
- ⇒ Những bộ kiểm toán tiêu chuẩn.
- ⇒ Những bộ mô phỏng.
- ⇒ Những bộ phân tích thi công.
- ⇒ Những bộ giám sát thực hiện.
- ⇒ Những công cụ hợp chất CASE hoặc một số công cụ khác hiện đang có trên thị trường.
- ⇒ Những bộ kích thích thử nghiệm.
- ⇒ Những bộ phát trường hợp thử nghiệm.

Những công cụ đó hỗ trợ kiểm tra chất lượng trong mọi giai đoạn phát triển phần mềm. Các bộ hỗ trợ tư liệu có thể cung cấp phần nào các bộ viết tư liệu tự động, các bộ kiểm tra chính tả và các sưu tập chuyên ngành (Thesaurus) v.v... các bộ xử lý trước được cấu thành có ích cả trong việc tiêu chuẩn hóa liệt kê mã lẫn cung cấp cảnh báo bổ sung thời gian biên dịch mà các bộ biên dịch thường bỏ qua. Những cảnh báo sớm liên quan đến các vấn đề thời gian thực hiện có thể được các mô phỏng, các bộ phân tích thời gian thực hiện và các bộ giám sát thực hiện cung cấp. Thử nghiệm thấu đáo hệ thống phần mềm thường có thể được tự động thực hiện nhờ các bộ phát hệ thử nghiệm và các bộ thực hiện thử nghiệm tự động.

5.5.5 Lập ngân sách cuối cùng của dự án

Song song với tích hợp và thử nghiệm, các hoạt động quản lý và không phát triển sau đây diễn ra :

- ⇒ Lập ngân sách cuối cùng của dự án, phí tổn cho các thay đổi (phát minh) được xác định hoạt động dự phòng rủi ro được đánh giá và ngân sách được cập nhật.
- ⇒ Việc huấn luyện được tiến hành cho người dùng, người vận hành, khách hàng, người lắp đặt, đội ngũ bảo trì và đội tiếp thị.
- ⇒ Địa điểm lắp đặt được chuẩn bị và hạ tầng có sở cho phần cứng các thiết bị đặc biệt được lập kế hoạch và lắp đặt.
- ⇒ Qui mô đội ngũ phát triển được giảm đi.

5.5.6 Lắp đặt, tích hợp hệ thống và đào tạo người vận hành

Hai hoạt động khác song song với bảo trì đó là:

- ⇒ Duy trì thông tin cập nhật
- ⇒ Cập nhật những lớp huấn luyện người dùng

Không giống như bảo trì phần cứng, không có một hoạt động nào trong đó lại làm cho phần mềm trở lại trạng thái ban đầu của nó. Điều trái lại mới hoàn toàn đúng: Mục tiêu ở đây là sửa đổi phần mềm sửa đổi phần mềm bao gồm mọi đặc điểm của phát triển phần mềm. Những điều sửa đổi cần được mô tả, thiết kế, thực hiện hoà nhập và thử nghiệm chính thức và tất nhiên, những sửa đổi cần được tài trợ bởi nhân lực và vật lực. Do vậy để thực hiện giai đoạn bảo trì coi như một loạt những dự án phần mềm nhỏ.

Kiểm tra cấu hình là đặc biệt quan trọng trong thời gian bảo trì để quản lý những thay đổi với phần mềm và kiểm tra những đợt phát hành và phiên bản của hệ thống. Điều này đảm bảo những đợt phát hành định kỳ có trật tự của phần mềm và tránh được hỗ trợ phiêu lưu và không cố định có thể trở thành cơn ác mộng công trình.

Quản lý dự án không phải bao giờ cũng được tiến hành từ phát triển sản phẩm phần mềm sang giai đoạn bảo trì bảo trì đòi hỏi một đội nhỏ hơn nhiều và quản lý kiểu khác. Trên thực tế, một nhóm bảo trì duy nhất có thể được thiết lập để bảo trì nhiều sản phẩm với quản lý chung, không chế cấu hình kỹ thuật lắp đặt tại hiện trường và bảo trì tư liệu.

Tư liệu cần được cập nhật trong giai đoạn này bao gồm :

- ⇒ Tư liệu phát hành phiên bản
- ⇒ Các báo cáo vấn đề
- ⇒ Mọi tư liệu phát triển
- ⇒ Mọi tư liệu của người dùng
- ⇒ Nhật ký bảo trì và báo cáo dịch vụ khách hàng

Bảo trì được tiếp đọc chừng nào sản phẩm phần mềm được lắp đặt và hoạt động. Khi hệ thống già đi, các kế hoạch được chuẩn bị cho một dự án phát triển mới thay thế cho hệ thống lão hoá. Vào lúc đó, các cố gắng bảo trì được giảm tới mức tối thiểu do kỳ vọng là mọi sửa đổi yêu cầu và hiệu chỉnh sai số kỹ thuật sẽ được có trong hệ thống mới.

5.6 QUẢN LÝ GIAI ĐOẠN BẢO TRÌ

Khái quát chung:

Giai đoạn bảo trì bắt đầu khi khách hàng đã chấp nhận sản phẩm và cần có các thay đổi trên sản phẩm như mã nguồn, tài liệu, hướng dẫn sử dụng... hay còn gọi là sự tiến triển.

Sự cần thiết của bảo trì:

Trong chu trình sống của phần mềm thì bảo trì là giai đoạn cuối cùng của một chu trình và ta cần phải nói bảo trì là giai đoạn vô cùng cần thiết vì:

Khi thực hiện một quá trình nghiệm cứu trên một số lượng phần mềm lớn khó tránh khỏi sai sót. Vì vậy, ta cần phải hiệu chỉnh các lỗi đặt tả, thiết kế, tài liệu, mã nguồn hay các dạng khác. Trong đó, hiệu chỉnh chiếm khoảng 17.5%. Về khâu hoàn thiện cũng cần cần tới sự bảo trì vì khoảng 60.5% sai sót. Nên ta cần phải thay đổi về mã lệnh nhằm hoàn thiện hiệu năng của sản phẩm. Sau khi hoàn thiện và đưa vào sử dụng để đạt được hiệu quả cao ta cũng cần bảo trì để phần mềm thích ứng được khoảng 18% khi các thay đổi nhằm tác động lại những thay đổi của

môi trường mà sản phẩm đang vận hành. Khách hàng phải chịu khoảng chi phí đó. Ngoài các dạng kể trên còn có khoảng 4% các dạng khác.

Ví dụ: Khách hàng yêu cầu thêm một số chức năng hay sửa đổi sản phẩm để tăng tốc độ xử lý nhằm để hoàn thiện. Hay thay đổi trình biên dịch, hệ điều hành hoặc phần cứng để thích ứng.

Những quan tâm đối với các nhà lập trình bảo trì:

Do quá trình bảo trì thật sự cần thiết như thế nên đã đặt ra những đòi hỏi đối với các nhà lập trình bảo trì như sau:

Hoàn thành thuật ngữ nhà lập trình bảo trì (maintenance programmer – MP) để chỉ những nhóm lập trình viên thực hiện các công tác bảo trì phần mềm. Bảo trì là giai đoạn cuối cùng của một chu kỳ phần mềm nên nó được coi là khía cạnh khó khăn nhất, nhiều thách thức của một sản phẩm phần mềm vì đụng chạm đến tất cả các giai đoạn trong tiến trình này dựng phần mềm. Đòi hỏi từ thực tế cho thấy các công ty hiện nay thường xem nhẹ công tác bảo trì và hay giao các công đoạn bảo trì cho các lập trình viên mới. Khi đó sẽ dễ nảy sinh trường hợp MP không đủ kinh nghiệm và kỹ năng lẫn vết tốt trong việc xác định chính xác vị trí lỗi. Bởi vì trong một số trường hợp lỗi có thể nảy sinh ra do người sử dụng hoặc hướng dẫn sử dụng không chính xác và cũng có thể do lỗi mã nguồn gây ra. Lỗi hồi qui cũng là một đòi hỏi khả quan vì khi sửa chữa lỗi có quan tâm đến các lỗi khác trong sản phẩm.

Các nhà lập trình bảo trì cần phải chuẩn bị tài liệu chi tiết cho toàn bộ sản phẩm cũng như từng mô-đun riêng biệt sau khi sửa chữa xong để khi công tác bảo trì cho lần sau có thể sử dụng lại. Và trong thực tế bảo trì được xem như một dịch vụ hậu thuẫn nhằm giữ khách hàng bằng cách cung cấp những dịch vụ bảo trì tốt nhất và nhanh nhất. Đó cũng là chuẩn mực cần có của tất cả các công ty kinh doanh phần mềm.

Quản lý bảo trì:

Xây dựng cơ chế cho phép có những thay đổi trên sản phẩm khi thực hiện việc bảo trì.

Lãnh đạo nhóm SQA - Nhóm đảm bảo chất lượng phần mềm [QLDAPM.ĐHQG] và nhóm phát triển phần mềm phải độc lập nhau.

Quản lý chặt trong việc báo cáo lỗi tránh việc sai sót các thông tin ban đầu bằng cách:

- Yêu cầu người sử dụng điền các thông tin về lỗi trên các chức năng.
- Các báo cáo lỗi phải đủ thông tin để những MP có thể tái tạo lại những lỗi đã xảy ra.

Quản lý sự uỷ quyền thay đổi tên sản phẩm:

- Xác định lỗi, thay đổi mã nguồn, cố định mã nguồn.
- Khó khăn thử qui hồi trên sản phẩm.
- Cập nhật các tài liệu để phản ánh các thay đổi.
- Có thể cập nhật về dữ liệu về đặt tả cũng như thiết kế.
- Tạo phiên bản mới khi cần thiết.
- Chuyển đến nhóm SQA để xác nhận lại (nhưng không được can thiệp vào công việc của các lập trình viên).

Bảo đảm công tác bảo trì:

- Việc bảo trì cần phải được thực hiện nhiều lần.
- Tạo nhiều phiên bản để việc thẩm tra và kiểm thử được chính xác hơn.
- Có kế hoạch bảo trì trong suốt trình phần mềm.
- Ghi nhận cẩn thận các thông tin kỹ thuật để có thể tái tạo sử dụng khi cần
- Tài liệu phải được hoàn tất và hiệu chỉnh chu đáo, phản ánh chính xác mọi thành phần của phiên bản hiện hành.

Vấn đề về sự lặp lại công tác bảo trì:

- Do khách hàng thường xuyên thay đổi các yêu cầu nên việc bảo trì cũng phải thay đổi thường xuyên tùy theo yêu cầu.
- Nên đưa ra các mô hình làm việc cụ thể khi có sự thay đổi càng phải trả thêm chi phí phát sinh.

Bảo trì phần mềm hướng đối tượng:

Công tác bảo trì phần mềm hướng đối tượng có ưu và nhược điểm sau:

Ưu điểm: Dễ dàng bảo trì các đối tượng do:

- Các khái niệm độc lập với nhau nên dễ dàng xác định vị trí lỗi, nhằm thực hiện sự hiệu chỉnh hay nâng cao tùy theo từng yêu cầu.
- Các thay đổi tác động bên trong các đối tượng nên giảm thiểu các lỗi hồi quy.

Nhược điểm:

- MP phải nghiên cứu toàn bộ cây thừa kế.
- Khó khăn khi cài đặt trên ngôn ngữ lập trình hướng đối tượng vì vấn đề đa hình và động.
- Khi lần vết các nhà thừa kế liên tục nhau khi có một lớp nào đó có sự thay đổi tạo ra sự khó khăn đáng kể.

So sánh kỹ năng bảo trì và kỹ năng phát triển:

Khả năng	Kỹ năng bảo trì	Kỹ năng phát triển
Xác định nguyên nhân gây ra lỗi	Hiệu chỉnh	Kiểm thử tích hợp và kiểm thử phát triển
Thực hiện hiệu quả các chức năng mà không có tài liệu thích hợp	Hoàn thiện, thích ứng	Đặc tả, thiết kế, cài đặt và tích hợp, kiểm thử
Nắm vững các vấn đề liên quan trên các giai đoạn	Đòi hỏi như nhau	Đòi hỏi như nhau

Kiểm thử giai đoạn bảo trì:

Khó khăn khi phải nắm vững toàn bộ sản phẩm.

Cách tiến hành:

- Sử dụng các tình huống kiểm thử để đảm bảo sản phẩm vẫn còn vận hành tốt sau khi đã có cập nhật.
- Thay đổi một số tình huống kiểm thử.

- Lưu trữ toàn bộ các tình huống kiểm thử với kết quả cần đạt được tương ứng.
- Sử dụng kiểm thử hồi quy.

Đánh giá giai đoạn bảo trì:

Sử dụng cách đánh giá cho các giai đoạn liên quan như trong quá trình phát triển.

Ngoài ra còn có:

- Số lượng báo cáo lỗi.
- Phân loại lỗi theo độ khó và kiểu lỗi.
- Thông tin về trạng thái hiện hành của báo cáo lỗi.

CÂU HỎI CHƯƠNG 5 :

Câu 5.6.1.1: Thuật ngữ “nhà lập trình bảo trì” có ý nghĩa như thế nào?

Câu 5.6.1.2: Bảo trì phần mềm hướng đối tượng là gì? Tại sao bảo trì phần mềm hướng đối tượng được cho là dễ?

Câu 5.6.1.3: Sử dụng kiểm thử hồi qui là gì ? Nếu không có quá trình này được không ?

Câu 5.6.1.4: Giai đoạn bảo trì là giai đoạn cuối cùng trong chu trình sống của phần mềm vậy nói bảo trì có ảnh hưởng đến các giai đoạn khác là đúng hay không ?

Câu 5.6.1.5: Trong phần đánh giá tại sao phải đánh giá theo độ khó và kiểu lỗi ?

Câu 5.6.1.6: Nếu 1 phần mềm game sau khi mua về sử dụng 1 thời gian xảy ra lỗi ở phần đồ họa thì có tốn chi phí bảo trì hay không ?

Câu 5.6.1.7: Làm sao xác định điểm dừng cho bảo trì ?

GỢI Ý TRẢ LỜI CÁC BÀI TẬP

Trả lời câu 5.6.1.1: Nhà lập trình bảo trì là những lập trình viên thực hiện công tác bảo trì khi có lỗi xảy ra hay có yêu cầu từ khách hàng.

Trả lời câu 5.6.1.2: Bảo trì phần mềm hướng đối tượng là công tác bảo trì trên các phần mềm mang tính chất hướng đối tượng. Đây là vấn đề được cho là dễ vì có thuận lợi nhất định khi xảy ra lỗi dễ tìm kiếm cũng như khắc phục không sợ các lỗi hồi qui.

Trả lời câu 5.6.1.3: Kiểm thử hồi qui là khi thực hiện công tác bảo trì nếu phát hiện ra lỗi ở một giai đoạn nào đó thì ta phải sửa chữa lúc đó ta cũng phải kiểm tra lại các giai đoạn khác vì lỗi ở giai đoạn trước đó có thể gây ra lỗi cho các phần tiếp theo. Nếu không có quá trình này thì công tác bảo trì không đảm bảo được chất lượng phần mềm đã hoàn hảo như yêu cầu đặt ra.

Trả lời câu 5.6.1.4: Theo mô hình thác nước hình 3.2 chương 3(chu trình sống của phần mềm) đã thấy rõ bảo trì có sự quan hệ với các giai đoạn khác.Vd:Có 1 khách hàng đã mua phần mềm do ta viết và sử dụng sau 1 thời gian họ yêu cầu ta sửa đổi cấu trúc cho phù hợp hơn với thời điểm hiện hành khi đó công tác bảo trì được thực hiện và khi thay đổi theo yêu cầu của khách hàng là đã phải làm việc sửa đổi đối với các giai đoạn khác.

Trả lời câu 5.6.1.5: vì khi ta thực hiện bảo trì cần phải xác định được lỗi xảy ra là do nguyên nhân từ đâu nếu là lỗi của khách hàng sử dụng thì cần xác định thêm độ khó để biết được chi phí khắc phục mà khách hàng phải chịu là bao nhiêu.

Trả lời câu 5.6.1.6: Vì đây là phần mềm bạn mua và đã sử dụng sau 1 khoảng thời gian nên các MP sẽ phải xác định lại đây là lỗi do mã nguồn hay do người sử dụng

nếu do mã nguồn gây ra lỗi thì bạn sẽ được cấp 1 phần mềm mới không lỗi và đáp ứng được yêu cầu của bạn. Ngược lại bạn sẽ phải trả chi phí cho hoạt động bảo trì.

Trả lời câu 5.6.1.7: Đối với tất cả phần mềm có bản quyền thì nhà cung cấp sẽ ký với người sử dụng 1 hợp đồng về thời gian bảo hành và các điều khoản của công tác bảo hành. Vì vậy có thể nói điểm dừng của bảo hành chính là thời gian bảo hành và các vấn đề cần phải biết trong điều khoản bảo hành.

TÀI LIỆU THAM KHẢO

1. [GTPPTK.ĐHQG] - Đào Kiến Quốc, Giáo trình phân tích và thiết kế hệ thống thông tin tin học hóa, Hà Nội 2000.
2. [QLDAPM.ĐHQG] - Quản lý dự án phần mềm, Khoa Công nghệ Thông tin Đại học Quốc gia Hà Nội.
3. [TĐQ&NMS.PTTK] – Trần Đình Quế, Nguyễn Mạnh Sơn, Phân tích thiết kế hệ thống thông tin, Học viện Công nghệ Bưu chính Viễn Thông.
4. Tham khảo nhiều tác giả - www.moet.gov.vn - Phân tích và thiết kế hệ thống tin học.
5. Tham khảo nhiều tác giả - www.quantrimang.com - Phân tích và thiết kế hệ thống tin học.