

Mục lục

1	Phát triển hệ thống	1
	Giới thiệu	2
1.1	Phương pháp luận phát triển	2
1.1.1	Vai trò của tổ chức phát triển hệ thống	2
1.1.2	Mô hình phát triển phần mềm	5
1.1.3	Vòng đời phần mềm	10
1.1.4	Dùng lại phần mềm	19
1.2	Phân tích yêu cầu và phương pháp thiết kế	22
1.2.1	Phương pháp lập biểu đồ	22
1.2.2	Lập biểu đồ phân tích/thiết kế	25
1.2.3	Phương pháp thiết kế	32
1.3	Ngôn ngữ lập trình	54
1.3.1	Thuộc tính chương trình	54
1.3.2	Kiểu dữ liệu	55
1.3.3	Cấu trúc điều khiển	56
1.3.4	Phân tích cú pháp	58
1.3.5	Phân loại về ngôn ngữ lập trình	63
1.3.6	Kiểu và đặc trưng của ngôn ngữ lập trình	67
1.4	Kỹ thuật lập trình	75
1.4.1	Lập trình thủ tục	75
1.4.2	Lập trình hàm	75
1.4.3	Lập trình logic	76
1.4.4	Lập trình hướng đối tượng	77
1.5	Phương pháp kiểm thử và kiểm điểm	78
1.5.1	Phương pháp kiểm thử	78
1.5.2	Phương pháp kiểm điểm	85
1.5.3	Thiết kế kiểm thử và phương pháp quản lý	88
1.6	Môi trường phát triển	92
1.6.1	Công cụ hỗ trợ phát triển	92
1.6.2	EUC và EUD	99
1.7	Quản lý phát triển	101
1.7.1	Lập kế hoạch dự án	101
1.7.2	Lập kế hoạch, quản lý và đánh giá chất lượng	102
1.7.3	Quản lý tiến trình	104
1.7.4	Năng suất phần mềm	108
1.7.5	Tổ chức phát triển	113
1.8	Gói phần mềm	119

1.8.1	Đại cương về gói phần mềm	119
1.8.2	Phân loại các gói phần mềm	120
1.8.3	Công cụ quản lí sản xuất	122
1.8.4	Ví dụ về việc dùng hiệu quả gói phần mềm	123
1.8.5	Phần mềm nhóm - Groupware	124
1.8.6	Công cụ OA	125
	Bài tập	131
2	Vận hành và bảo trì hệ thống	139
	Giới thiệu	140
2.1	Vận hành hệ thống	140
2.1.1	Quản lí tài nguyên	140
2.1.2	Quản lí vấn đề	142
2.1.3	Quản lí tiện nghi	144
2.1.4	Quản lí an ninh	147
2.1.5	Việc quản lí vận hành khác	150
2.2	Bảo trì hệ thống	151
2.2.1	Bảo trì là gì?	151
2.2.2	Tầm quan trọng của công việc bảo trì	152
2.2.3	Chi phí bảo trì	153
2.2.4	Nhiệm vụ bảo trì	153
2.2.5	Tổ chức bảo trì	158
2.2.6	Các kiểu bảo trì	161
2.2.7	Bảo trì phần cứng và bảo trì phần mềm	162
	Bài tập	164
3	Cập nhật xử lí thông tin và an ninh	168
	Giới thiệu	169
3.1	Lĩnh vực ứng dụng của hệ thống máy tính	169
3.1.1	Tổng quan về kinh doanh điện tử (e-business)	169
3.1.2	Tổng quan về e-learning	172
3.1.3	Tổng quan về chính phủ điện tử (e-government)	177
3.2	An ninh	183
3.2.1	An ninh Internet	183

Giới thiệu

- Loạt sách giáo khoa này đã được xây dựng dựa trên Chuẩn kỹ năng kỹ sư Công nghệ thông tin được đưa ra công cộng vào tháng 7 năm 2000. Bốn tập sau đây bao quát toàn bộ nội dung của tri thức và kỹ năng cơ bản cần cho việc phát triển, vận hành và bảo trì hệ thống tin:
- Tập 1: Giới thiệu về hệ thống máy tính
- Tập 2: Phát triển và vận hành hệ thống
- Tập 3: Thiết kế trong và lập trình -- Tổng lượng tri thức cốt lõi và thực hành
- Tập 4: Công nghệ mạng và cơ sở dữ liệu
- Cuốn sách này nêu những giải thích dễ một cách có hệ thống, để những người đang học về phát triển, vận hành và bảo trì hệ thống lần đầu tiên có thể dễ dàng thu được những tri thức trong các lĩnh vực này. Cuốn sách này bao gồm các chương sau:
- Chương 1: Phát triển hệ thống
- Chương 2: Vận hành và bảo trì hệ thống

1 Phát triển hệ thống

Mục đích của chương

Với các công ti, điều bản chất là thu được lợi nhuận. Tuy nhiên tình hình kinh tế hiện nay là hỗn loạn và rất khó dự đoán xu hướng tương lai. Trong tình huống như vậy, các hệ thống xử lý thông tin trở thành quan trọng thiết yếu cho hoạt động nghiệp vụ, và do đó là mấu chốt cho sự tồn tại của công ti. Các kĩ sư xử lý thông tin, tất cả các hệ thống thiết kế và phát triển xử lý thông tin đã trở nên ngày một quan trọng.

Chương này dự định nói về luồng phát triển dùng mô hình thác đổ, cung cấp một cơ sở cho việc phát triển hệ thống, môi trường phát triển, hệ thống quản lí, và việc dùng các gói phần mềm. Về chi tiết, các mục sau sẽ được giảng dạy:

- ① Kiểu và đại cương về các phương pháp luận phát triển hệ thống, và các tiến trình phát triển dùng mô hình thác đổ.
 - ② Cách lập biểu đồ phân tích yêu cầu, và các công nghệ thiết kế khác nhau.
 - ③ Các ngôn ngữ lập trình và kĩ thuật lập trình.
 - ④ Tầm quan trọng của kiểm thử, và các phương pháp thiết kế các trường hợp kiểm thử.
 - ⑤ Các công cụ được dùng trong môi trường phát triển, và các kĩ thuật và hệ thống cho việc quản lí phát triển.
- Các kiểu gói phần mềm và cách dùng chúng.

Giới thiệu

- Năm mươi năm đã trôi qua kể từ khi cái máy được gọi là "máy tính" đi vào hiện hữu. Ban đầu, không có lí thuyết hay phương pháp luận gì cho việc phát triển hệ thống cả. Cho nên việc phát triển tùy thuộc chủ yếu vào "tài nghệ" của kĩ sư. Nó hoàn toàn tương tự như tình huống trong những ngày xưa khi thợ mộc làm nhà theo "trực giác" và "kinh nghiệm" của họ. Tuy nhiên, các hệ thống đã dần trở nên ngày một lớn hơn, và cùng với những phát triển mới, sự mở rộng của việc thay đổi cho hệ thống hiện có đã trở thành cần thiết. Thêm vào đó, năng suất phát triển đã trở nên cao hơn. Tình huống này đã đạt tới điểm mà việc dùng "tài nghệ" là hoàn toàn không đủ, đưa tới việc mặc nhiên công nhận cần có các lí thuyết phát triển hệ thống, và các phương pháp kĩ nghệ để thực hiện chúng.
- Thêm vào với các phương pháp qui ước, những phương pháp phát triển và lí thuyết mới đã được đưa vào để cung cấp năng suất cao hơn và thân thiện hơn với người dùng.
- Trong chương này, các lí thuyết cơ sở cho việc phát triển hệ thống và các phương pháp luận của chúng được giảng dạy. Việc dùng có hiệu quả các gói phần mềm gần đây và các phương pháp kiểm điểm là bản chất cho công việc phát triển cũng được trình bày.

1.1 Phương pháp luận phát triển

- Mục này giới thiệu về mối quan hệ giữa các hệ thống doanh nghiệp nghiệp, công nghệ thông tin và các phương pháp phát triển chính.

-

1.1.1 Vai trò của tổ chức phát triển hệ thống

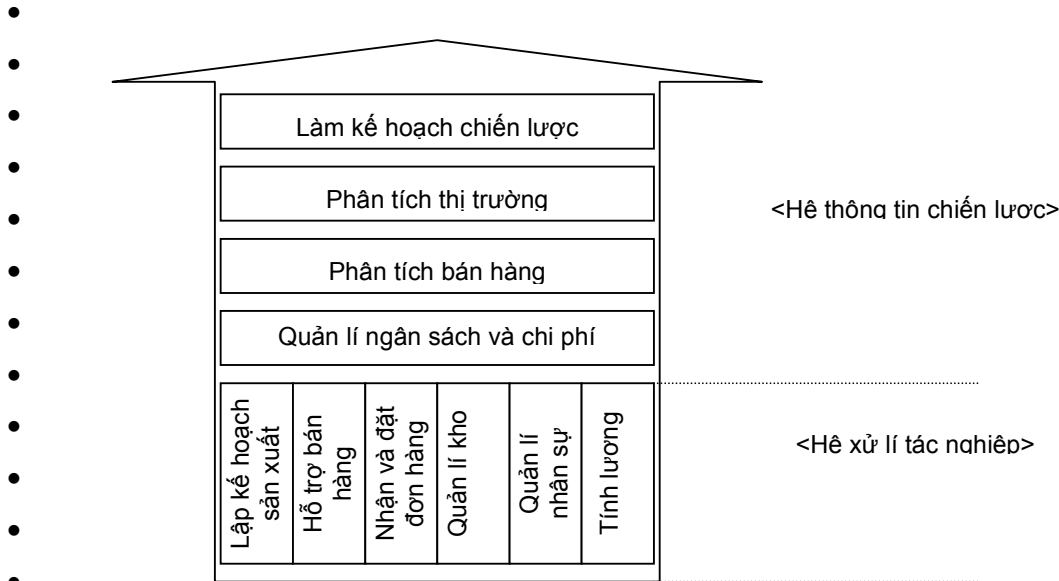
(1) Hoạt động xí nghiệp và hệ thông tin

- Cần nhiều nỗ lực để tạo ra sự phát triển liên tục của xí nghiệp. Trong đó việc sử dụng Công nghệ Thông tin (CNTT) là điều chủ chốt.
- Với các xí nghiệp, có hai kiểu thông tin:
 - Thông tin nội bộ : Thông tin được phát sinh qua các hoạt động nghiệp vụ, và bao gồm cả các hoá đơn, biểu mẫu và tài liệu quản lí được dùng trong bán hàng, ra lệnh sản xuất và kế toán.
 - Thông tin ngoài : Thông tin được phát sinh ra tùy thuộc vào tình trạng kinh tế bao quanh công ti, kể cả việc bán sản phẩm, xu hướng các ngành công nghiệp có liên quan, động thái của các công ti cạnh tranh, và các giao tác với các công ti có liên quan.
- Doanh nghiệp dùng các thông tin trên trong hoạt động hàng ngày của mình một cách hiệu quả. Hệ thông tin được phân thành hai loại:
 -
 - Hệ xử lí tác nghiệp : Được dùng để hỗ trợ cho các hoạt động hàng ngày và cung cấp dữ liệu quản lí doanh nghiệp. Việc xử lí thông tin đều kì được tiến hành để làm tăng hiệu suất và cải

tiến hiệu quả vận hành.

- Hệ thống tin chiến lược : Để đạt tới các hoạt động nghiệp vụ đa dạng, điều quan trọng là dùng tài nguyên sẵn có, con người, vật tư và tiền bạc theo cách hiệu quả nhất. Hệ thống cung cấp thông tin cần cho việc quản lí các tài nguyên này. Hệ thống này chủ yếu được dùng để sinh ra các báo cáo theo quan điểm quản lí.

Hình 1-1-1 Hệ thống tin chiến lược và hệ xử lí tác nghiệp



- Mục đích của từng hệ là như sau.
- <Hệ thống xử lí tác nghiệp>
 - Làm giảm nhân công cho những thao tác có liên quan
 - Làm giảm việc xử lí nghiệp vụ
 - Làm giảm thời gian giao hàng
 - Làm giảm việc quản lí kho
 - Thực hiện các thao tác không giấy tờ
- <Hệ thông tin chiến lược>
 - Làm tăng bán hàng
 - Cải thiện hiệu quả bán hàng
 - Cải thiện sự thoả mãn của khách hàng
 - Tạo ra thị trường mới
- Nói chung, một tổ chức chịu trách nhiệm phát triển các hệ thống xử lí thông tin được gọi là tổ chức phát triển hệ thống hay cái gì đó tương tự, và các kĩ sư hệ thống (SE) là các lập trình viên thuộc tổ chức này.
- Mặt khác, một tổ chức dùng các hệ thống đã phát triển được gọi là tổ chức người dùng.

(2) Tiến bộ trong CNTT và tổ chức phát triển hệ thống

- ① Tiến bộ trong công nghệ thông tin
- Những tiến bộ mới đây trong công nghệ là rất đáng ngạc nhiên. Trong số nhiều tiến bộ đa dạng, những điều sau đây được xem như các nhân tố ảnh hưởng rất đáng kể tới việc phát triển hệ thống.
- a. Tiến bộ trong công nghệ máy tính
 - Việc cải tiến hiệu năng của cả máy tính cá nhân và trạm làm việc, và việc giảm giá đáng kể của chúng là điều rất đáng ngạc nhiên. Những thao tác mà có thời đã phải được thực hiện chỉ với máy tính lớn thì bây giờ có thể được thực hiện bằng những máy tính nhỏ hơn.
- b. Sử dụng rộng rãi các gói phần mềm
 - Bởi vì sự phát triển nhanh chóng của các gói phần mềm, nên những gói phần mềm cho quản trị cơ sở dữ liệu RDBMS (Relational Data Base Management System) và bảng tính nay đã sẵn có, và nó cũng có thể được tích hợp vào một số hệ thống.
- c. Tiến bộ trong công nghệ mạng
 - Đổi mới công nghệ trong lĩnh vực thông tin và truyền thông, kể cả việc tổ hợp của mạng cục bộ LAN (Local Area Network) và WAN (Wide Area Network) và việc xây dựng các mạng nội bộ intranets và mạng ngoại bộ extranets (mạng được tạo ra bằng việc mở rộng mạng nội bộ ra bên ngoài công ti), là rất đáng chú ý.
- d. Tiến bộ trong công nghệ xây dựng hệ thống
 - Công nghệ đã dịch chuyển từ cách tiếp cận hướng xử lý sang cách tiếp cận hướng dữ liệu (DOA). Kết quả là các kĩ pháp như Biểu đồ luồng dữ liệu - Data Flow Diagram (DFD), Biểu đồ thực thể quan hệ - Entity-Relationship Diagram (ERD), và Cấp bậc cộng với Vào Xử lý Ra - Hierarchy plus Input Process Output (HIPO) được tạo ra để biểu diễn cho các thiết kế có cấu trúc, đã được chấp nhận rộng rãi. Thêm vào đó, các công cụ CASE (Computer Aided Software Engineering) đã trở nên sẵn có để hỗ trợ cho những nỗ lực phát triển, và đã trở nên được dùng rộng rãi.
- ② Hiện trạng của các tổ chức phát triển hệ thống
- Với việc tăng qui mô hệ thống và việc đưa vào multimedia, các tổ chức phát triển hệ thống đang phải đương đầu với những vấn đề sau:
- a. Tăng khối lượng công việc tồn đọng lại
 - Số lượng việc tồn đọng lại, điều chỉ ra hệ thống không thể được bắt đầu ngay lập tức theo yêu cầu phát triển từ tổ chức người dùng, đã tăng lên. Mọi công ti trung bình đều tồn đọng 2 đến 3 năm công việc.
- b. Đưa vào hệ multimedia và tăng số các hệ thống qui mô lớn
 - Các hệ thống multimedia, trong đó các dữ liệu đa dạng, kể cả tiếng nói, video và văn bản đều được dùng, đã đi vào thực tế. Thêm vào đó, WAN và LAN đã được tổ hợp và mạng nội bộ intranets đã được xây dựng. Do đó, ngày nay con người phải giải quyết các hệ thống qui mô ngày càng lớn và phức tạp.
- c. Tăng công việc bảo trì
 - Với qui mô ngày càng mở rộng hệ thống, khối lượng công việc bảo trì cũng tăng lên. Khối lượng này tăng lên bởi vì các yêu cầu sửa đổi hệ thống từ các phòng ban người dùng tăng lên, sửa đổi các hệ thống hiện có trở thành cần thiết, và việc sửa đổi toàn bộ hệ thống do gặp nhiều lỗi cũng

tăng lên.

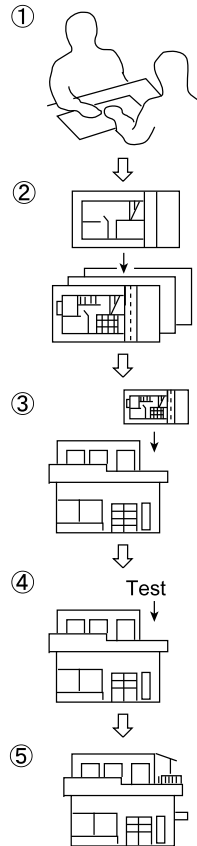
- ③ **Vai trò mới của tổ chức phát triển hệ thống**
- Bên cạnh những công việc qui ước về việc phát triển hệ thống và bảo trì, các tổ chức phát triển hệ thống hiện nay được trông đợi thực hiện những công việc có liên quan như sau:
 - a. Phát triển hệ thống cùng sự vận hành và bảo trì chúng
 - Bên cạnh các công việc qui ước, các tổ chức phải chấp nhận công nghệ và phương pháp mới một cách tích cực.
 - b. Xây dựng và vận hành các cơ sở dữ liệu và mạng
 - Các cơ sở dữ liệu khác nhau là bản chất cho sự vận hành và quản lý nghiệp vụ. Bên cạnh đó, bây giờ không thể xem xét các hoạt động nghiệp vụ mà không dùng mạng. Việc xây dựng, vận hành và quản lý các cơ sở dữ liệu và mạng này được coi như công việc bản chất của tổ chức.
 - c. Lập kế hoạch và điều phối tin học hoá toàn công ti
 - Việc lập kế hoạch cho các hệ thống lớn bao quát toàn bộ công ti, và phản ánh cái nhìn của người dùng và ban điều hành công ti, cũng là việc quan trọng của tổ chức.
 - d. Gắn chặt với người dùng
 - Người ta trông đợi rằng tổ chức chấp nhận các yêu cầu của người dùng một cách tích cực và phát triển các hệ thống thực tế hơn. Tổ chức cũng cần sẵn sàng cung cấp sự hỗ trợ kỹ thuật cần thiết để hướng dẫn môi trường tính toán người dùng cuối (EUC), huấn luyện từ xa và huấn luyện tại chỗ về xử lý thông tin trong công ti.

1.1.2 Mô hình phát triển phần mềm

- Với việc phát triển hệ thống, người ta dùng nhiều mô hình khác nhau tùy theo qui mô của công ti và cách thức làm việc chủ đạo trong công ti. Tại đây, những phương pháp này được mô tả một cách vắn tắt. Đặc biệt, chúng tôi sẽ nêu chi tiết về mô hình thác đổ trong mục 1.1.3.
- Như được vẽ trong Hình 1-1-2, việc xem xét cách xây nhà sẽ làm cho chúng ta dễ hiểu hơn về cách thức tiến triển việc xây dựng hệ thống.
- Nếu chỉ riêng các biểu đồ lược đồ khách hàng khó có thể có được hình ảnh rõ ràng về ngôi nhà và ngôi nhà trông sẽ như thế nào khi được hoàn tất. Do đó, hãy kiểm tra với ngôi nhà mô hình, hay các ảnh 3 chiều trên máy tính mới được cung cấp gần đây.
- Việc giải quyết tương tự cũng được dùng trong phát triển hệ thống.
-

Hình 1-1-2

Quy trình
xây dựng
nhà



- Yêu cầu của khách về một ngôi nhà được đưa cho nhà xây dựng. Dựa trên các yêu cầu này, người xây dựng đưa ra giá thành ước lượng, các bản vẽ sơ đồ và lịch biểu, và nói điều đó với khách hàng. Thêm vào đó, người chủ nhà còn phải đi xin phép các cơ quan chính phủ có liên quan.

- Dựa trên thiết kế sơ đồ, người xây dựng tạo ra các bản thiết kế chi tiết có tính tới những ràng buộc cần thiết như các điều kiện địa lí và xã hội của vị trí này và giá thành. Cuối cùng, việc thiết kế được chia xuống mức cho phép việc xây dựng thực.

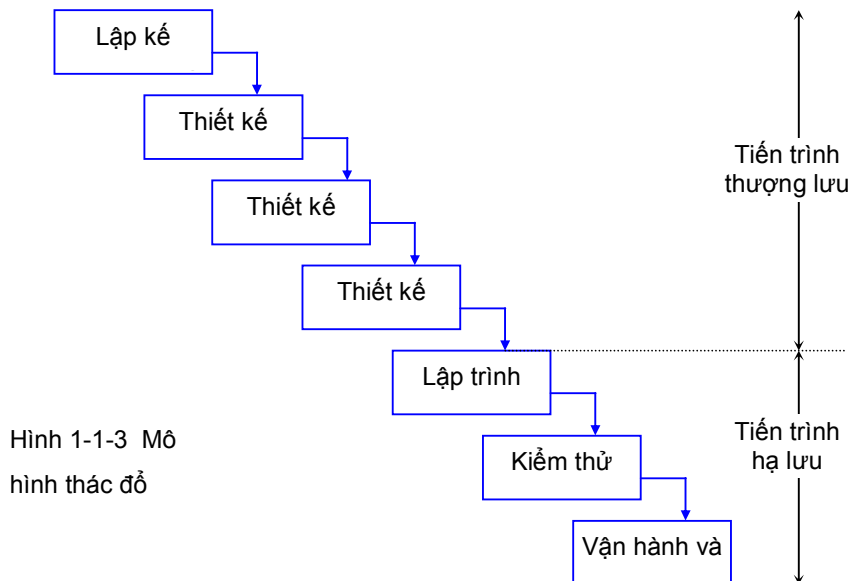
- Việc lắp ráp được thợ mộc thực hiện trên các thiết kế.

- Việc kiểm tra được tiến hành mỗi khi hoàn tất một bộ phận. Đến cuối, việc kiểm tra được thực hiện với sự có mặt của khách hàng. Sau đó ngôi nhà được chuyển giao cho khách hàng.

- Nhiều loại bảo trì sẽ được tiến hành tương ứng theo yêu cầu của khách..

(1) Mô hình thác đổ

- Mô hình thác đổ, công nghệ phát triển hệ thống, vẫn là mô hình được sử dụng rộng rãi nhất. Trong mô hình này công việc được phân chia thành một số pha, và việc quản lí được tiến hành cho từng pha.



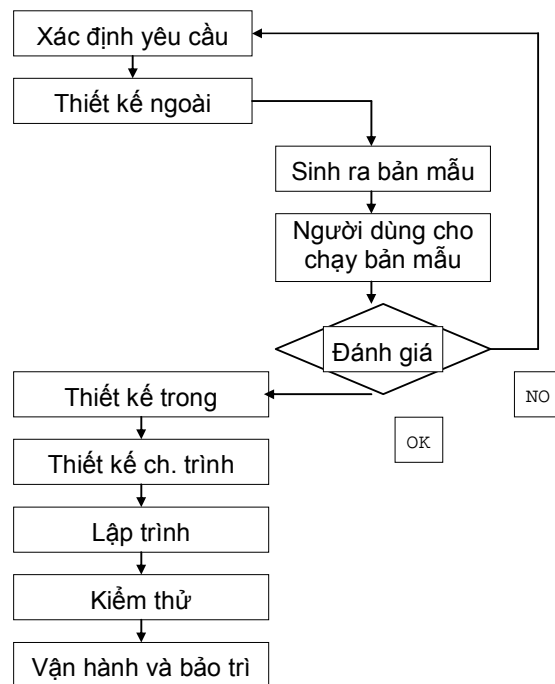
Hình 1-1-3 Mô
hình thác đổ

- Như được chỉ ra bởi cái tên "thác đổ", công việc tiến triển trong mô hình này từ luồng trên

cao (lập kế hoạch cơ sở) tới luồng dưới thấp (kiểm thử), không bao giờ đi ngược lại.

(2) Mô hình bản mẫu

- Mô hình thác đổ bao gồm những vấn đề sau.
 - Với mô hình thác đổ cực kì khó hiểu thấu yêu cầu của người dùng trong pha lập kế hoạch cơ sở cho hệ thống. Đôi khi ngay cả khách hàng cũng không biết tới những yêu cầu đó.
 - Các biểu đồ thiết kế và giải thích miệng đôi khi không đủ.
- Để giải quyết những vấn đề này, mô hình bản mẫu đã được đề xuất ra. Với mô hình bản mẫu, hệ thống được xây dựng sẽ được làm mô hình thô trong các ngôn ngữ lập trình đơn giản như SQL (Structured Query Language) để giúp cho khách hàng hiểu. Sau đó, công việc phát triển dự định sẽ được bắt đầu.
- Bản mẫu bao gồm nhiều mô hình đa dạng.
 - <Phân lớp theo phương pháp xây dựng>
 - "Kiểu vớt đi": Các mảnh mẫu thử bị vớt đi sau khi đã đạt tới mục đích của chúng.
 - "Kiểu khung xương": Các chi tiết được thêm dần vào cho các mảnh mẫu thử để mở rộng dần nó thành hệ thống dự định.
 - <Phân lớp theo mức độ bao quát>
 - "Kiểu dùng bộ phận": Mô hình này được dùng trong các pha xác định yêu cầu và thiết kế ngoài.
 - "Kiểu dùng toàn bộ": Mô hình này được xây dựng cho tất cả các pha.
- Hình 1-1-4 chỉ ra một ví dụ lưu đồ của mô hình bản mẫu theo kiểu dùng bộ phận.



Hình 1-1-4 Ví dụ
lưu đồ về mô hình
bản mẫu theo kiểu

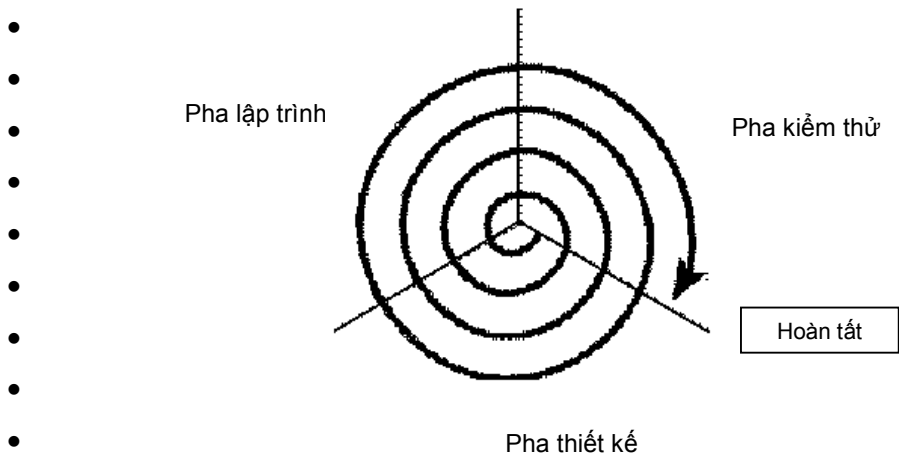
- Việc dùng mô hình bản mẫu đem tới cảm giác về sự tham gia của người dùng, ngăn ngừa các lỗi trong các pha thượng lưu vốn ảnh hưởng đáng kể tới công việc về sau.

- Tuy nhiên, mô hình bản mẫu bao gồm các vấn đề sau cần được giải quyết.
- - Chi phí phát triển vượt quá mô hình thác đổ.
- - Khó điều chỉnh lịch.

(3) Mô hình xoắn ốc

- Trong mô hình xoắn ốc, một loạt các tiến trình bao gồm thiết kế, lập trình và kiểm thử được lặp lại cho từng đơn vị con của hệ thống, với việc phát triển được lặp lại nhiều lần (xem Hình 1-1-5).

- <Các đặc trưng>
- - Cách dùng mô hình này là thích hợp cho những trường hợp mà đơn vị con của hệ thống được phát triển là tương đối độc lập lẫn nhau.
- - Nó tuân theo mô hình thác đổ theo bộ phận.
- - Nó cho phép mô hình bản mẫu được sử dụng khi có nhu cầu nảy sinh.
- - Được sử dụng trong việc phát triển kiểu hướng đối tượng và theo những cách khác.



Hình 1-1-5 Mô hình xoắn ốc

(4) Phát triển hướng đối tượng

- Gần đây sự chú ý đã dồn vào cách phát triển hướng đối tượng. Theo mô hình này, hệ thống được xét như một tập các đối tượng, và việc phát triển được tiến hành trên cơ sở đối tượng. Trong phát triển đối tượng, tiến trình phân tích tới thiết kế tới thực hiện được thực hiện lặp đi lặp lại thiết lập nên một loại mô hình xoắn ốc.

(5) Cấu trúc phân việc

- Để đạt tới mục đích của mình, dự án phát triển hệ thống được phân ra thành các mức theo thứ tự sau, tùy theo tiến trình phát triển.

1. Mức quyết định cấu trúc chính của dự án
2. Mức công việc tạo nên khuôn khổ cho từng pha
3. Mức công việc hiện hành chi tiết

•

•

•

• Hình 1-1-6 Cấu trúc phân

• việc

•

•

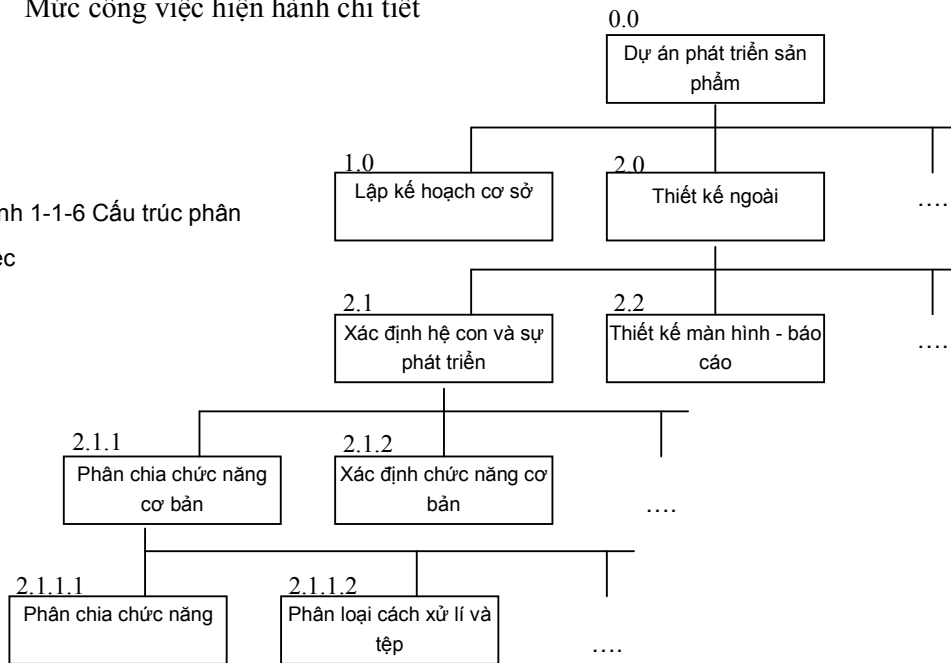
•

•

•

•

•



- Cấu trúc phân việc - Work Breakdown Structure (WBS) được suy ra bằng cách bổ sung thêm các mục đích cụ thể, lịch công việc, và việc quản lý tiến trình xác định các chi tiết ở mức mịn nhất có thể được cho điều thu được bởi những thao tác phân chia này. WBS được biểu diễn bằng cấu trúc phân cấp như được vẽ trong Hình 1-1-6.

- Việc dùng WBS đưa ra những ích lợi sau:

Cung cấp các ước lượng chi phí và dữ liệu cho việc phân tích chi phí.

Làm sáng tỏ cấu trúc công việc và bao quát công việc dự án cùng trách nhiệm về công việc.

Việc hiểu thấu tiến trình thực tại cho từng đơn vị công việc, và lập kế hoạch công việc được dễ dàng hơn.

Tên của các đơn vị công việc được phân ra và hệ thống phân lớp là một phần của tri thức chuyên gia về cách làm.

- Với WBS, mục đích của công việc, như chất lượng, chi phí và thời gian được cho trên cơ sở đơn vị công việc. Cho nên, công việc được thực hiện với các mục tiêu như tham chiếu.

•

(6) Tiến trình và mô hình tiến trình

- Tiến trình được định nghĩa như các đơn vị công việc nào, như phân tích, thiết kế và chế tạo,

được cần tới trong việc tạo ra sản phẩm (kể cả sản phẩm phần mềm), và được bố trí theo chuỗi thời gian. Mỗi cấu phần của tiến trình được gọi là "giai đoạn tiến trình". Với WBS được mô tả ở trên, công việc được biểu diễn theo cấp bậc, nhưng không theo chuỗi thời gian. Điều này tạo ra sự khác biệt lớn.

- Tiến trình được thiết kế đại thể trong pha lập kế hoạch cơ sở. Trong thiết kế này, mô hình tiến trình được nêu trong Hình 1-1-7 có thể được dùng như tham chiếu.

<Các giai đoạn tiến trình>	Lập kế hoạch	Xác định yêu cầu	Thiết kế			Chế tạo, xây dựng và kiểm thử				Chuyển dịch	Vận hành và bảo trì
<Mô hình tiến trình 1>	Thiết kế quan niệm		Thiết kế cơ sở		Thiết kế chi tiết	Mã hoá		Kiểm thử		Chuyển dịch	Bảo trì
<Mô hình tiến trình 2>	Phân tích hệ thống		Kế hoạch hệ thống hoá		Thiết kế chương trình	Sinh chương trình		Kiểm thử		Chuyển dịch	Bảo trì
<Mô hình tiến trình 3>	Lập kế hoạch		Thiết kế			Chế tạo		Kiểm thử hệ thống		Chuyển dịch sang vận hành thực tế	Bảo trì
<Mô hình tiến trình 4>	Kế hoạch hệ thống hoá	Xác định yêu cầu	Thiết kế ngoài	Thiết kế trong	Thực hiện phát triển					Kiểm thử hệ thống	Bảo trì
					Thiết kế mô đun	Mã hoá	Kiểm thử đơn vị	Kiểm thử môc nối	Chuyển dịch		
<Mô hình tiến trình 5>	Lập kế hoạch hệ thống hoá	Phân tích hệ thống	Thiết kế hệ thống đại cương	Thiết kế hệ thống chi tiết		Chế tạo		Kiểm thử, chuyển dịch			Vận hành và bảo trì
<Mô hình tiến trình 6>	Lập kế hoạch hệ thống	Phân tích hệ thống	Thiết kế giao diện người dùng	Thiết kế cấu trúc hệ thống	Thiết kế cấu trúc chương trình	Lập trình	Kiểm thử chương trình	Kiểm thử môc nối	Kiểm thử hệ thống	Kiểm thử vận hành	Bảo trì/ đánh giá hệ thống

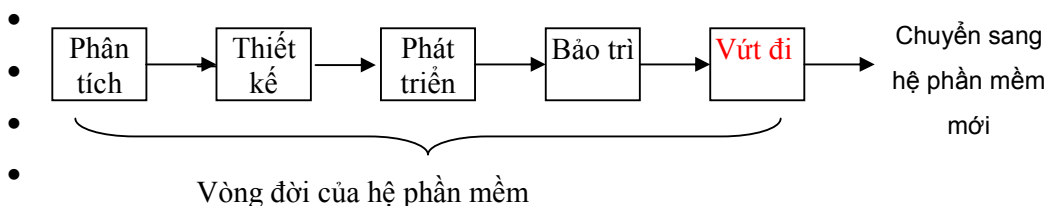
Hình 1-1-7 Mô hình tiến trình

- Các sản phẩm phần mềm được tạo ra qua từng tiến trình phát triển. Do đó, thiết kế của từng tiến trình, cho cơ sở của tiến trình, chủ yếu ảnh hưởng tới chất lượng và chi phí của sản phẩm phần mềm.

1.1.3 Vòng đời phần mềm

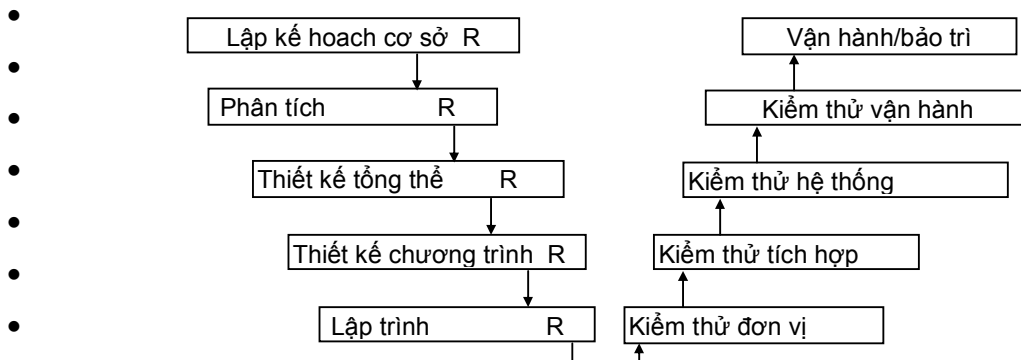
- Vòng đời là tiến trình từ khi sinh đến khi chết hay khoảng sống của vật sống và sản phẩm. Với khái niệm về vòng đời phần mềm (SLC) cũng vậy, đó là khoảng thời gian từ lúc bắt đầu dự án phát triển hệ thống, và thời gian khi việc cập nhật hệ thống kết thúc, được xem như cuộc sống của hệ thống. Vậy, các hoạt động xuất hiện trong thời kì đó được biểu diễn bằng cuộc sống thực được dùng, xem như mô hình diễn tả cho mối quan hệ giữa các tiến trình.

- Hình 1-1-8 Vòng đời của hệ thống phần mềm



- Trong phần sau đây sẽ giải thích về mô hình thác đổ, là mô hình điển hình nhất.

(1) Đặc trưng của mô hình thác đổ

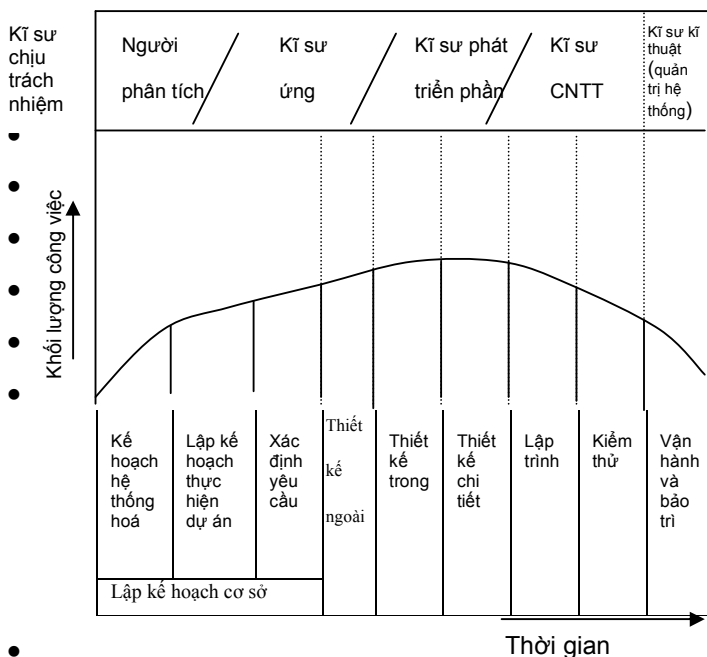


R: họp xét duyệt

Hình 1-1-9 Hình ảnh toàn thể của mô hình thác nước (cấu trúc hình chữ V)

Trong mô hình thác đổ, các kỹ thuật sau được sử dụng. Do đó, có thể trực quan hoá mô hình này như cấu trúc hình chữ V trong Hình 1-1-9.

- Pha lập kế hoạch cơ sở tới pha lập trình: phương pháp làm mịn từng bước (tiếp cận trên xuống)
- Pha kiểm thử đơn vị tới pha kiểm thử vận hành: phương pháp tích hợp từng bước (cách tiếp cận dưới lên)



Hình 1-1-10 Mối quan hệ giữa từng pha trong mô hình thác đổ và khối lượng công việc của nó

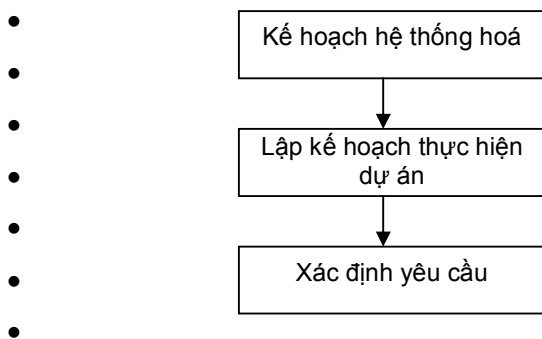
- Các đặc trưng của Mô hình thác đổ được tóm tắt sau đây:
- Công việc phát triển hệ thống được chia thành một số pha cho việc quản lí.
- Khi công việc của một pha hoàn tất thì sản phẩm công việc (kể cả đủ loại tài liệu thiết kế) của pha này được xét duyệt để kiểm tra tính đúng đắn.
- Các sản phẩm công việc (kể cả đủ loại tài liệu thiết kế) từ pha này được chuyển tiếp sang tiến trình tiếp. Theo cách này, sự nhất quán trong việc phát triển hệ thống được duy trì.

- Về cơ bản, không được phép trở lại công việc của pha trước.
- Cách tổ chức dự án có tầm quan trọng chủ chốt
- Như đã mô tả ở trên, việc phân chia tiến trình phát triển hệ thống thành một số pha cho quản lý là một trong những đặc trưng của mô hình thác nước. Hình 1-1-10 chỉ ra mối quan hệ giữa từng pha và khối lượng công việc ở đó.
- Trong các mục (2) tới (8) dưới đây, sẽ mô tả đại cương từng pha công việc trong phát triển hệ thống.
-

(2) Lập kế hoạch cơ sở

- Lập kế hoạch cơ sở là bước đầu tiên của việc phát triển hệ thống. Cần phải có tri thức thấu đáo về hoạt động hiện tại nhằm đạt mục đích tin học hoá. Bằng không, không thể nào phát triển được hệ thống có thể thoả mãn cho người dùng. Do vậy, việc lập kế hoạch cơ sở bắt đầu bằng phân tích về hệ thống hiện tại, rồi nhận diện ra vấn đề của nó.

- Hình 1-1-11 Lập kế hoạch cơ sở



- Thủ tục chi tiết trong lập kế hoạch cơ sở được mô tả sau đây:

1. Kế hoạch hệ thống hoá

- Việc lập kế hoạch hệ thống hoá là công việc soạn thảo ra các kế hoạch cơ sở cho một hệ thống.

- <Nhiệm vụ>

- Điều tra và phân tích vấn đề trong các hoạt động nhằm tới việc hệ thống hoá.

- Dựa trên kết quả của mục a, khảo sát giải pháp tốt nhất và sự cần thiết phát triển hệ thống được xem xét lại. Nếu thấy rằng có một giải pháp tốt hơn, thì chấp nhận giải pháp đó.

- Nếu từ công việc của mục b thấy rằng việc phát triển hệ thống mới là thích hợp thì bản kế hoạch hệ thống hoá sẽ được tạo ra và được đề đạt cho người có trách nhiệm.

- <Tài liệu>

- Bản kế hoạch hệ thống hoá.

2. Kế hoạch thực hiện dự án

- Sau khi người có trách nhiệm chấp thuận kế hoạch hệ thống hoá, thì bản kế hoạch thực hiện (bản kế hoạch thực hiện dự án) được viết ra.

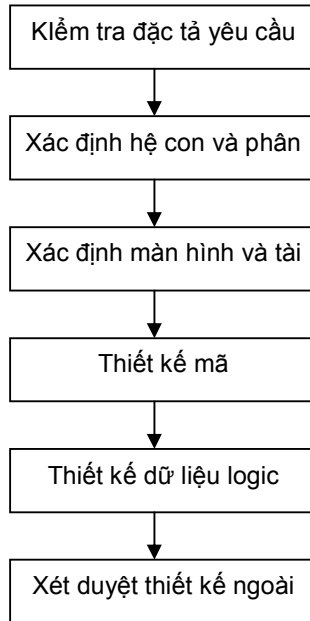
- <Thủ tục>

- a. Dự án được tổ chức (kể cả bổ nhiệm người phụ trách)
- b. Bản kế hoạch tài nguyên hệ thống (ước lượng) được soạn thảo ra.
- Kế hoạch nhân sự
- Phần cứng cho phát triển hệ thống
- Ước lượng qui mô phát triển (kể cả nhân lực và ngân sách)
- Tài chính
- Và những thứ khác
- c. Bản kế hoạch tiến trình công việc và lịch biểu mức cao nhất được soạn ra.
- Có các kiểu lịch sau:
 - Lịch biểu mức cao nhất: lịch cho toàn bộ hệ thống
 - Lịch biểu mức trung: Lịch cho từng pha trong việc phát triển hệ thống.
 - Lịch biểu mức thấp nhất: Lịch biểu cho từng người có liên quan.
- Nếu lập được tất cả các loại lịch biểu trên tại pha này là tốt nhất nhưng thực tế thì khó làm được. Cho nên ít nhất cần soạn ra lịch biểu mức cao nhất tại pha này.
- <Tài liệu>
- Bản kế hoạch phát triển.
- 3. Xác định yêu cầu
 - Trong xác định yêu cầu, các chức năng đích thiết lập nên cái vào cho việc phát triển hệ thống, và các yêu cầu cho hệ thống tin được phân tích và định nghĩa chi tiết hơn trong các bản kế hoạch phát triển, bằng các phương pháp có cấu trúc như DFD (biểu đồ luồng dữ liệu) và ERD (biểu đồ thực thể quan hệ)
 - <Nhiệm vụ>
 - a. Thông tin về hệ thống, như công việc trong các chức năng đích (mô hình công việc được sinh ra tại đây được gọi là mô hình logic hiện tại), các khuôn mẫu được dùng và các mục đích, được thu thập.
 - b. Các yêu cầu về hệ thống xem như một tổng thể, bao gồm các chức năng, các yêu cầu hiệu năng và thao tác được xác định.
 - c. Các yêu cầu cho cả phần cứng và phần mềm được làm sáng tỏ.
 - <Tài liệu>
 - Bản đặc tả yêu cầu
- 4. Tóm tắt về việc lập kế hoạch cơ sở
 - Việc lập kế hoạch cơ sở là pha thiết kế ra đại cương về hệ thống tin, kể cả việc phân tích hệ thống cần phát triển và vạch ra lịch biểu mức cao nhất. Kết quả của nó ảnh hưởng rất nhiều tới các tiến trình đi sau. Bên cạnh đó, nó là pha đầu tiên trong việc phát triển hệ thống. Do đó, điều cần thiết là công việc được thực hiện với sự xác nhận của người dùng, công việc được người phân tích hệ thống có kỹ năng cao và có tri thức cốt lõi thực hiện.
 -
 -

(3) Thiết kế ngoài

- Thiết kế ngoài là để xác định phần thấy được bên ngoài hay những phần dành cho giao tiếp với người dùng. Do đó, hệ thống được xét thuần túy theo quan điểm của người dùng mà không đề ý tới những ràng buộc về phần cứng (như máy tính). Bên cạnh đó, cấu trúc hệ thống cần được đạt tới cũng được làm sáng tỏ.

-
- Hình 1-1-12 Thiết kế ngoài
-



- <Nhiệm vụ>
- a. Kiểm tra đặc tả yêu cầu
 - Sau khi kiểm tra bản đặc tả yêu cầu có trong bản kế hoạch cơ sở, thì việc xét tổng quan về hệ thống được biểu diễn bởi việc dùng các biểu đồ sao cho việc xử lý và luồng dữ liệu có thể được hiểu dễ dàng. Dựa trên các biểu diễn này, thực hiện việc phân chia thành các hệ con và thiết kế vào/ra. Các DFD hay HIPO được dùng để vẽ biểu đồ.
- b. Xác định các hệ con và phân chia thêm nữa
 - Toàn bộ hệ thống được chia thành một số hệ con trên cơ sở chức năng, rồi hệ con lại được phân chia thêm nữa thành các đơn vị nhỏ hơn.
 -
- c. Thiết kế tài liệu và màn hình.
 - Trong thiết kế màn hình và tài liệu, các thiết kế thô cho màn hình và việc chuyển đổi màn hình, thiết kế thô cho tài liệu vào/ra được tạo ra.
- d. Thiết kế mã
 - Tại đây, việc thiết kế mã, như việc xác định hệ thống mã được tiến hành.
- e. Thiết kế dữ liệu logic
 - Trong thiết kế dữ liệu logic, mối quan hệ giữa dữ liệu được phân tích, và ứng cử viên cho cơ sở dữ liệu và tệp được rút ra. (công việc chi tiết cho điều này được tiến hành trong thiết kế

ngoài).

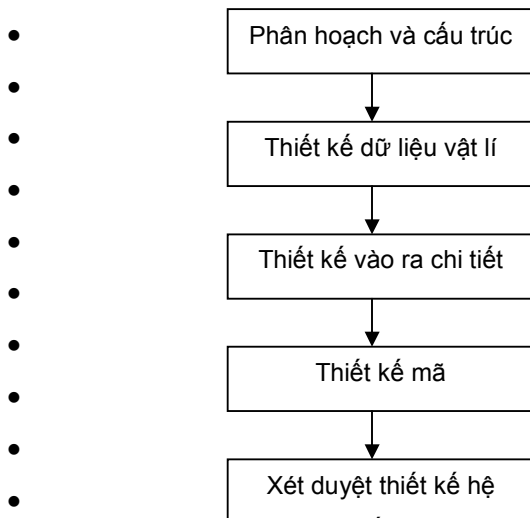
- f. Xét duyệt thiết kế ngoài
- Tài liệu thiết kế ngoài được xét duyệt.
- <Tài liệu>
- Tài liệu thiết kế ngoài
- Báo cáo xét duyệt tài liệu thiết kế ngoài

(4) Thiết kế trong

- Thiết kế trong dành cho phần không thấy được của hệ thống, và xử lý với những thiết kế được xem xét từ phía máy tính hay phía phát triển hệ thống. Trong thiết kế ngoài, hệ thống được nhìn từ quan điểm của người dùng. Tuy nhiên trong thiết kế trong, các chi tiết được thiết kế bằng việc xem xét tới các thiết kế ngoài này được cài đặt hiệu quả thế nào trên máy tính, hay bởi việc tính tới những ràng buộc phần cứng cũng như phần mềm.

- Trong thiết kế trong, công việc được chỉ ra trong Hình 1-1-13 sẽ được tiến hành.

- Hình 1-1-13 Thiết kế trong



- <Nhiệm vụ>

- a. Phân hoạch và cấu trúc chức năng (thiết kế cấu trúc)
 - Trong phân hoạch và cấu trúc chức năng (thiết kế cấu trúc), mỗi hệ con đều phân hoạch thành các đơn vị lập trình, và luồng dữ liệu và xử lý giữa các chương trình được làm sáng tỏ.
- b. Thiết kế dữ liệu vật lí
 - Trong thiết kế dữ liệu vật lí (thiết kế tệp), để dùng hiệu quả các đặc trưng phần cứng, việc thiết kế vật lí về tệp và cơ sở dữ liệu được dựa trên thiết kế dữ liệu logic đã được thực hiện trong pha phân tích hệ thống.
- c. Thiết kế vào-ra chi tiết

- Trong thiết kế vào ra chi tiết, các chi tiết về màn hình và tài liệu vào-ra được thiết kế bằng việc dùng mẫu đặc biệt.

- d. Xét duyệt thiết kế trong

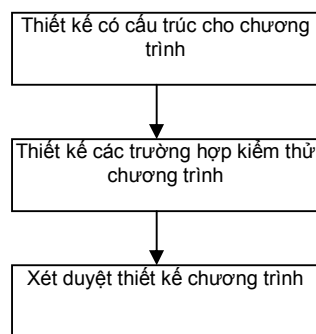
- Việc xét duyệt được tiến hành với tài liệu thiết kế trong.
- <Tài liệu>
- Tài liệu thiết kế trong
- Báo cáo xét duyệt thiết kế trong

(5) Thiết kế chương trình

- Trong thiết kế chương trình, các cấu trúc trong của từng chương trình được thiết kế. Mỗi chương trình được dẫn ra qua việc phân chia trong pha thiết kế hệ thống lại được phân chia thêm nữa một cách có phân cấp thành các đơn vị chức năng được gọi là mô đun. Sau đó, các giao diện (dữ liệu vào/ra) giữa các modun được thiết kế.

- Bên cạnh đó, các kế hoạch cho việc kiểm thử chương trình (tích hợp) cũng được chuẩn bị, và các trường hợp kiểm thử được thiết lập. Trong thiết kế chương trình, công việc được chỉ ra trong Hình 1-1-14 sẽ được tiến hành.

- Hình 1-1-14 Thiết kế chương trình



- <Nhiệm vụ>

- a. Thiết kế có cấu trúc cho chương trình (Phân hoạch mô đun)

- Trong thiết kế có cấu trúc, mỗi chương trình lại được phân hoạch thành các đơn vị chức năng được gọi là mô đun để cho phép dễ dàng bảo trì, còn luồng dữ liệu và xử lý giữa các chương trình thì rõ ràng, mạch lạc. Bên cạnh đó, các chức năng của từng mô đun và giao diện giữa các mô đun cũng được xác định.

- b. Thiết kế các trường hợp kiểm thử (tích hợp) chương trình

- Các kế hoạch cho kiểm thử chương trình được chuẩn bị và các trường hợp kiểm thử được thiết kế.

- c. Xét duyệt thiết kế chương trình

- Trong xét duyệt thiết kế chương trình, tiến hành xét duyệt tài liệu thiết kế chương trình.
- <Tài liệu>
- Tài liệu thiết kế chương trình
- Báo cáo xét duyệt thiết kế chương trình
- Kế hoạch kiểm thử (tích hợp) chương trình

(6) Lập trình

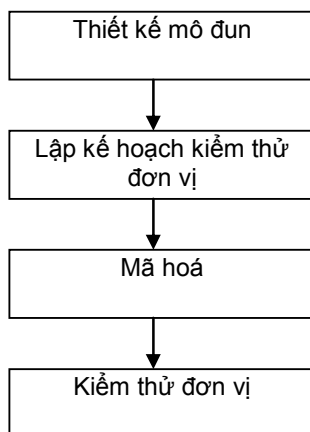
- Trong lập trình, việc thiết kế các cấu trúc logic của mô đun được định nghĩa trong pha thiết kế chi tiết và việc mã hoá cho các mô đun được thực hiện. Bên cạnh đó, các kế hoạch cho kiểm thử (đơn vị) mô đun được chuẩn bị, và các trường hợp kiểm thử được thiết lập.

- <Nhiệm vụ>

a. Thiết kế mô đun

- Trong thiết kế mô đun, các cấu trúc logic trong của mô đun (các thủ tục xử lý chi tiết bên trong từng mô đun) được thiết kế bằng việc dùng các kỹ thuật có cấu trúc khác nhau.

- Hình 1-1-15 Lập trình



b. Lập kế hoạch kiểm thử đơn vị

- Các kế hoạch kiểm thử đơn vị được chuẩn bị. (Tạo ra dữ liệu kiểm thử thích hợp, xác định lịch kiểm thử.)

c. Mã hoá

- Từng mô đun được mã hoá trong ngôn ngữ lập trình

d. Kiểm thử đơn vị

- Tiến hành kiểm thử đơn vị cho từng mô đun

- <Tài liệu>

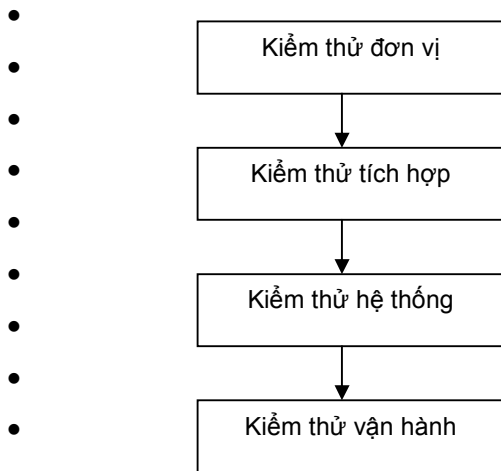
- Tài liệu thiết kế mô đun
- Báo cáo xét duyệt thiết kế mô đun
- Kế hoạch kiểm thử đơn vị

- Danh sách chương trình gốc
- Báo cáo xét duyệt chương trình gốc
- Báo cáo kiểm thử đơn vị

(7) Kiểm thử

- Công việc kiểm thử được tiến hành để phát hiện lỗi trong hành vi và cấu trúc của mô đun, chương trình hay hệ thống xem như một tổng thể (xem Hình 1-1-16). Nếu lỗi được tìm ra thì cần làm phân hồi, nếu cần, cho pha lập trình hay thiết kế để sửa chữa. Rồi lại tiến hành kiểm thử để kiểm tra liệu lỗi trên thực tế đã được sửa hay chưa.

- Hình 1-1-16 Các loại kiểm thử



- <Nhiệm vụ>

- a. Kiểm thử đơn vị (được tiến hành trong pha lập trình)

- Trong kiểm thử đơn vị, mỗi mô đun đều được kiểm tra xem liệu nó có thực hiện đúng đắn hay không.

-

- b. Kiểm thử tích hợp

- Trong kiểm thử tích hợp, các kiểm thử được tiến hành cho từng chương trình được tạo ra bằng việc móc nối các mô đun. Kiểm tra việc vận hành của chương trình và giao diện giữa các mô đun.

- c. Kiểm thử hệ thống

- Trong kiểm thử hệ thống, sự vận hành của hệ thống xem như một tổng thể được kiểm tra toàn bộ theo quan điểm của mục đích và hiệu năng được yêu cầu. Rồi việc bắt đầu vận hành thực tế được quyết định dựa trên kết quả này.

- d. Kiểm thử vận hành

- Trong kiểm thử vận hành, các nhóm vận hành từ phía người dùng tiến hành các kiểm thử trong điều kiện và môi trường như trong vận hành thực tế.

- <Tài liệu>

- Báo cáo kiểm thử đơn vị
- Báo cáo kiểm thử tích hợp
- Báo cáo kiểm thử hệ thống
- Báo cáo kiểm thử vận hành

(8) Vận hành và bảo trì

- Việc sản xuất theo hệ thống đã phát triển được bắt đầu. Các hoạt động bảo trì được tiến hành khi khiếm khuyết (như khó dùng) hay lỗi được phát hiện ra, hay khi thay đổi là không thể tránh khỏi. Trong một số trường hợp, hệ thống cần phải được thay đổi.

-

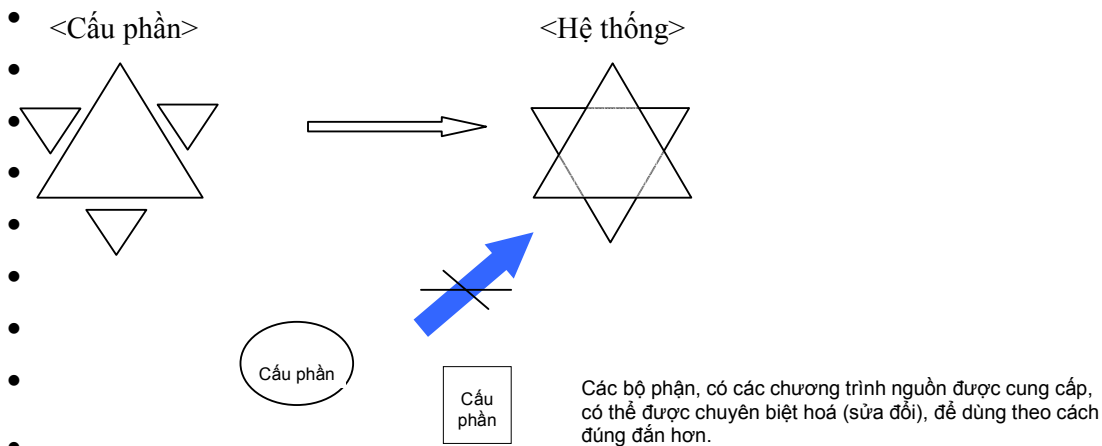
1.1.4 Dừng lại phần mềm

- Hệ thống đã được phát triển trên cơ sở xây dựng theo đơn hàng. Dựa trên yêu cầu của người dùng, người phát triển phân tích các thao tác đích, và thiết kế, lập trình, và kiểm thử hệ thống trước khi hoàn thành việc phát triển của nó. Sau đó người dùng làm quen với thao tác của mình bằng việc dùng hệ thống đã được xây dựng đáp ứng cho các yêu cầu riêng của người đó.
- Tuy nhiên bây giờ nhiều công ti bị phiền hà bởi việc tồn đọng của họ (các khoản mục phát triển còn chưa được bắt đầu). Việc sản xuất dựa theo xây dựng theo đơn hàng là nguyên nhân làm tăng thêm việc tồn đọng. Với sản xuất dựa theo việc xây dựng theo đơn hàng, việc phát triển hệ thống phải mất nhiều tháng và nhiều năm. Bên cạnh đó, việc tham dự của các chuyên gia như kĩ sư hệ thống (SE) là cần thiết để phát triển hệ thống. Tuy nhiên, trong những nỗ lực thực tế của các kĩ sư hệ thống tốn nhiều vào việc bảo trì hệ thống hiện có, điều này nghĩa là số nhân viên có thể dự phòng cho các dự án phát triển mới là nhỏ. Tình huống này đã tạo ra ý tưởng về việc dừng lại phần mềm hiện có.
- Trong việc dừng lại phần mềm, các bộ phận của toàn bộ phần mềm được tạo ra từng phần, hay phần mềm hiện tại được phân tích, và sửa đổi, và kết quả là thu được phần mềm mới. Theo cách đó, một phần mềm có thể được dùng lặp lại trong các hệ thống khác nhau.
- Có những cách sau đây để dừng lại phần mềm:
 - - Dừng lại như các bộ phận
 - - Dừng lại qua tái kĩ nghệ

(1) Dừng lại như các cấu phần

- Phương pháp tạo ra các cấu phần từ phần mềm hiện có và lắp ráp chúng để xây dựng hệ thống mới làm giảm chi phí và tăng chất lượng hệ thống. Bên cạnh đó, chu kì phát triển có thể được làm ngắn lại.
- ① Phương pháp để tạo ra các cấu phần
 - Các bộ phận được chính người dùng tạo ra hay được người bán phần mềm cung cấp. Các bộ phận do người bán cung cấp đều được chuẩn hoá. Tuy nhiên, việc chuẩn hoá theo định dạng đặc biệt, và việc biểu diễn cũng cần cho các bộ phận do người dùng tạo ra. Lí do là ở chỗ các cấu phần mà không được chuẩn hoá thì không thể được lắp ráp đúng. Hơn nữa, trong khi ngôn ngữ nào được dùng về cơ bản không phải là vấn đề, thì tốt hơn cả là nên dùng cùng một ngôn ngữ để tính tới mã nguồn có thể phải được thay đổi.

Hình 1-1-17 Chuẩn hoá các cấu phần



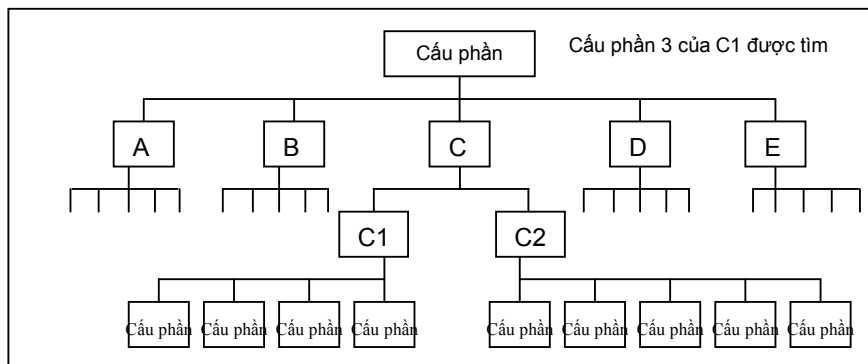
② Thư viện cấu phần và hệ thống tìm kiếm

Để xây dựng hệ thống bằng việc dùng các cấu phần phần mềm, các cấu phần được cất giữ và duy trì trong thư viện cấu phần. Số các cấu phần được cất giữ trong thư viện càng lớn, thì kết cấu của hệ thống càng trở nên linh hoạt hơn.

Hơn nữa, khi hệ thống tìm được cải tiến, thì thư viện cấu phần sẽ được dùng thường xuyên hơn, cải tiến cơ sở dữ liệu cấu phần và tăng cơ hội dùng thư viện.

•

Hình 1-1-18 Thư viện cấu phần



•

•

• Sự tồn tại của thư viện cấu phần rất phong phú và công cụ để dùng tạo ra hiệu quả tổng hợp

•

③ Cải tiến các cấu phần - tạo ra công cụ và công cụ tìm cấu phần

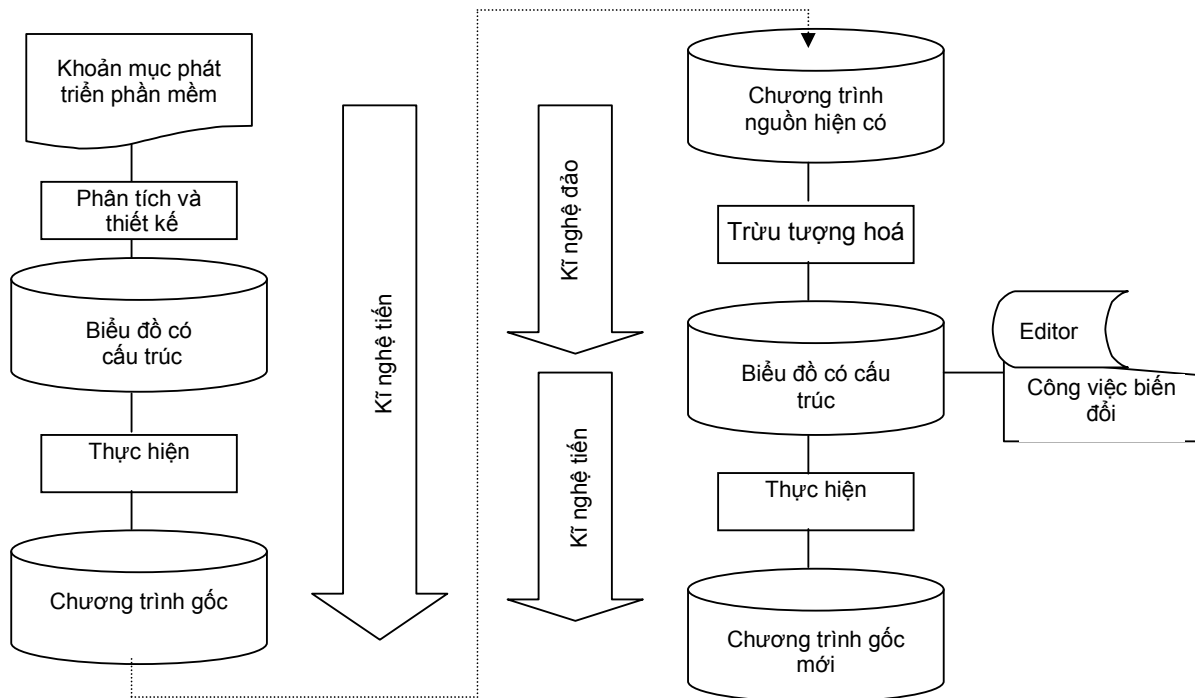
Khi cấu phần phần mềm được tạo ra và nó cung cấp cách thức dễ dàng để tìm kiếm cấu phần trong thư viện, thì việc dùng lại phần mềm sẽ được tăng tốc bởi vì dễ dùng. Hệ quả là sẽ có thể cung cấp các công cụ tạo cấu phần đã được cải tiến và những công cụ tìm cấu phần.

Các phương pháp hay công cụ để phát triển ứng dụng bằng việc lắp ráp các cấu phần phần mềm được tạo ra, dựa trên các đặc tả chuẩn để có khả năng dùng lại, được gọi là "componentware."

(2) Dùng lại qua tái kĩ nghệ

- Việc tạo ra phần mềm mới từ phần mềm hiện có được gọi là "tái kĩ nghệ". Tái kĩ nghệ là công nghệ được dùng để tạo ra hệ thống mới bằng việc sử dụng hệ thống đang trong sử dụng.
- Để thực hiện tái kĩ nghệ, các công cụ CASE (như công cụ CASE bảo trì) thường được dùng.
- Trong tái kĩ nghệ thông thường, các đặc tả phần mềm hiện có được dẫn ra ngay bước đầu tiên. Công nghệ được dùng trong phần này của công việc được gọi là "kĩ nghệ đảo". Sau đó, đặc tả cho phần mềm mới được tạo ra bằng việc sửa đổi đặc tả suy dẫn. Dựa trên các đặc tả mới, một hệ thống mới được tạo ra. Công nghệ phát triển truyền thống được dùng ở đây được gọi là "kĩ nghệ tiến" tương phản với "kĩ nghệ đảo" (xem Hình 1-1-19).

Hình 1-1-19 Tái kĩ nghệ



① Tiến → ② Đảo → ③ Tiến = Tái kĩ nghệ

1.2 Phân tích yêu cầu và phương pháp thiết kế

- Sau khi bản kế hoạch thực hiện dự án phát triển cho một sản phẩm đã được người dùng chấp thuận, thì điều đầu tiên cần được tiến hành trong dự án phát triển là phân tích các yêu cầu. Sau đây sẽ trình bày về các phương pháp phân tích và thiết kế điển hình.

-

1.2.1 Phương pháp lập biểu đồ



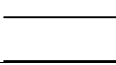

- Trong phân tích yêu cầu, các chức năng của hệ thống ứng dụng đích được rút ra bằng việc tạo nên một mô hình đại cương về hệ thống, để xây dựng ra sản phẩm. Sau đây các biểu đồ DFD, ERD, biểu đồ chuyển trạng thái và UML sẽ được trình bày để giúp cho công việc phân tích.

(1) DFD (Biểu đồ luồng dữ liệu)

- DFD (Biểu đồ luồng dữ liệu) là phương pháp phân tích bằng việc dùng cách vẽ biểu đồ, và biểu diễn các tiến trình nghiệp vụ một cách trực quan để trợ giúp cho việc hiểu bằng cách chú ý tới luồng dữ liệu.

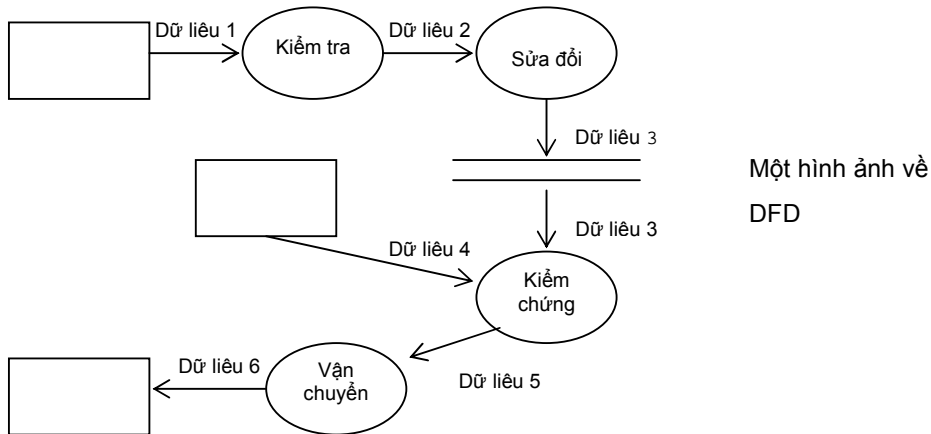
-

Hình 1-2-1 Ký pháp DFD

Kí hiệu	Tên	Ý nghĩa
	Luồng dữ liệu (mũi tên có tên)	Chỉ ra luồng dữ liệu
 	Tiến trình (Bong bóng)	Chỉ ra xử lý/ chuyển đổi dữ liệu
	Kho dữ liệu (Hai đường song song)	Chỉ ra kho dữ liệu
	Tác nhân ngoài (hình vuông)	Chỉ ra nguồn hay nơi nhận dữ liệu (thường là người hay tổ chức ngoài việc phân tích)

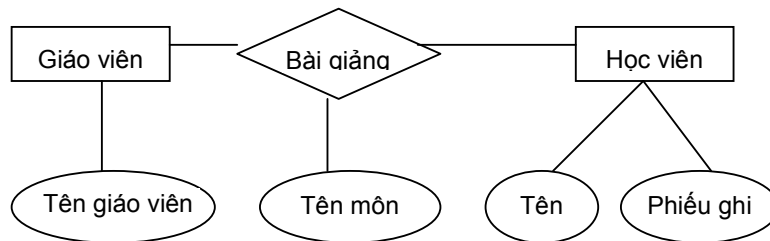
-

-



(2) ERD (Biểu đồ thực thể quan hệ)

- ERD đưa ra một mô hình dữ liệu trong đó thế giới đích được biểu diễn bằng hai khái niệm "thực thể" và "quan hệ" giữa các thực thể. ERD cũng còn được gọi là "mô hình vận hành nghiệp vụ" vì nó được dùng để tạo ra mô hình vận hành nghiệp vụ.

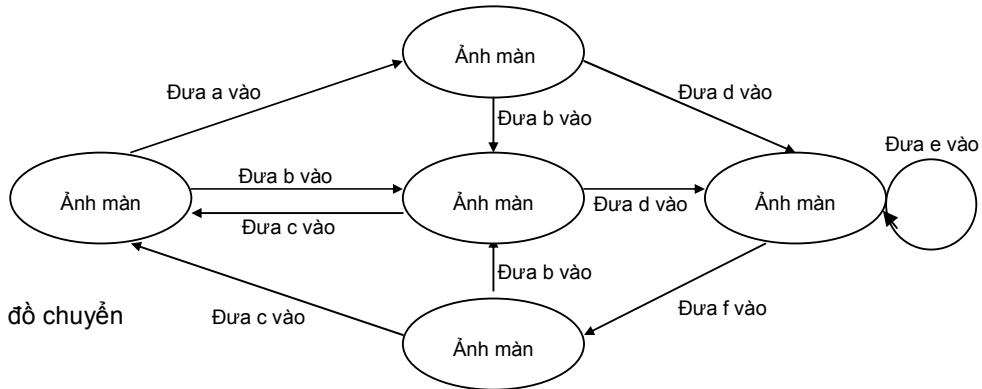


- Hình 1-2-2 Biểu đồ thực thể quan hệ
- ERD bao gồm ba phần tử sau:
 - Thực thể:** Chỉ ra một thực thể cho quản lý; được biểu diễn bằng hình chữ nhật.
 - Quan hệ:** Chỉ ra mối quan hệ giữa các thực thể, và mỗi quan hệ bao gồm cả giữa thực thể và quan hệ, được biểu diễn bằng hình thoi.
 - Thuộc tính:** Chỉ ra các đặc trưng và tính chất của thực thể hay quan hệ, được biểu diễn bằng hình elip.
- ERD trong Hình 1-2-2 chỉ ra điều sau:
 - "Giáo viên" và "Học viên" được nối qua quan hệ có tên "Bài giảng."
 - "Giáo viên" có "Tên giáo viên."
 - "Học viên" có "Tên" và "Phiếu ghi."
 - "Bài giảng" có "Tên chủ đề."
- Ba kiểu quan hệ tồn tại: "1 - 1," "1 - N" và "N - N" với N là số nguyên lớn hơn 1. Trong thí

dụ này, nếu giáo viên đọc bài giảng cho nhiều học viên và học viên dự nhiều bài giảng khác nhau do một giáo viên giảng tương ứng, thì mỗi quan hệ là N-N.

(3) Biểu đồ chuyển trạng thái

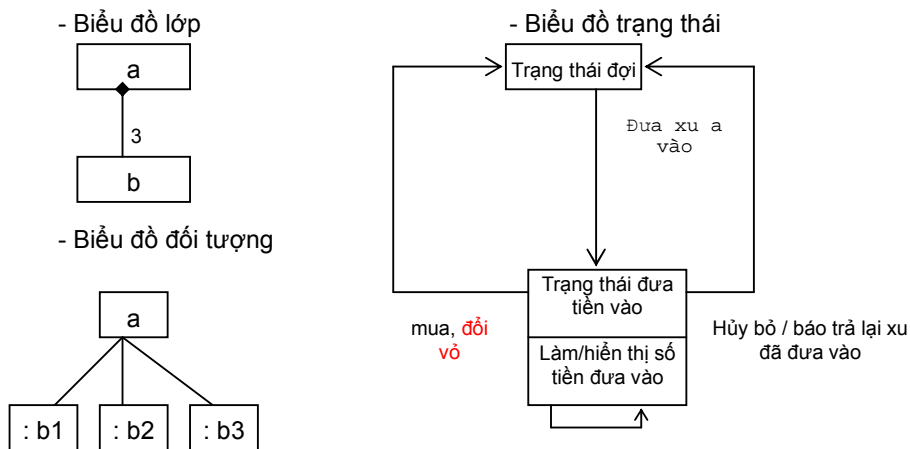
Biểu đồ chuyển trạng thái được dùng để sơ đồ hoá những tình huống có trạng thái biến thiên theo thời gian và hành vi. Biểu đồ chuyển trạng thái của Hình 1.2.3 chỉ ra cách các màn hình thay đổi bởi các thao tác trên màn hình (như đưa vào dữ liệu và ấn phím chức năng).



Hình 1.2.3 Biểu đồ chuyển trạng thái

(4) UML (Ngôn ngữ mô hình hoá thống nhất)

UML là ngôn ngữ mô hình hoá được dùng trong phân tích hướng đối tượng. Đã có nhiều phương pháp biểu diễn, mỗi phương pháp lại tùy thuộc vào người đề nghị nó, và chúng đã không được thống nhất. UML đã được phát minh ra để chuẩn hoá chúng.



Hình 1.2.4 UML

1.2.2 Lập biểu đồ phân tích/thiết kế

- Các chức năng hệ thống được xác định trong các tiến trình thượng lưu được đạt tới bằng một số chương trình. Lưu đồ là phương pháp lập biểu đồ tiêu biểu nhất được dùng để phân tích các cấu trúc logic. Sau đây, lưu đồ các sơ đồ có cấu trúc và bảng quyết định được mô tả như các phương pháp lập biểu đồ cho việc phân tích và thiết kế.

•

(1) Lưu đồ

- Dựa trên việc xác định/phân tích các vấn đề phức tạp và thủ tục xử lý cũng như phạm vi công việc, lưu đồ là một trong các phương pháp và là phương pháp tiêu biểu nhất, được dùng để biểu diễn cách thức giải quyết chúng và luồng thao tác cần thiết, với các kí hiệu thống nhất.
- Hình 1-2-5 chỉ ra các kí hiệu được dùng trong lưu đồ, được xác định trong chuẩn JIS X 0121.

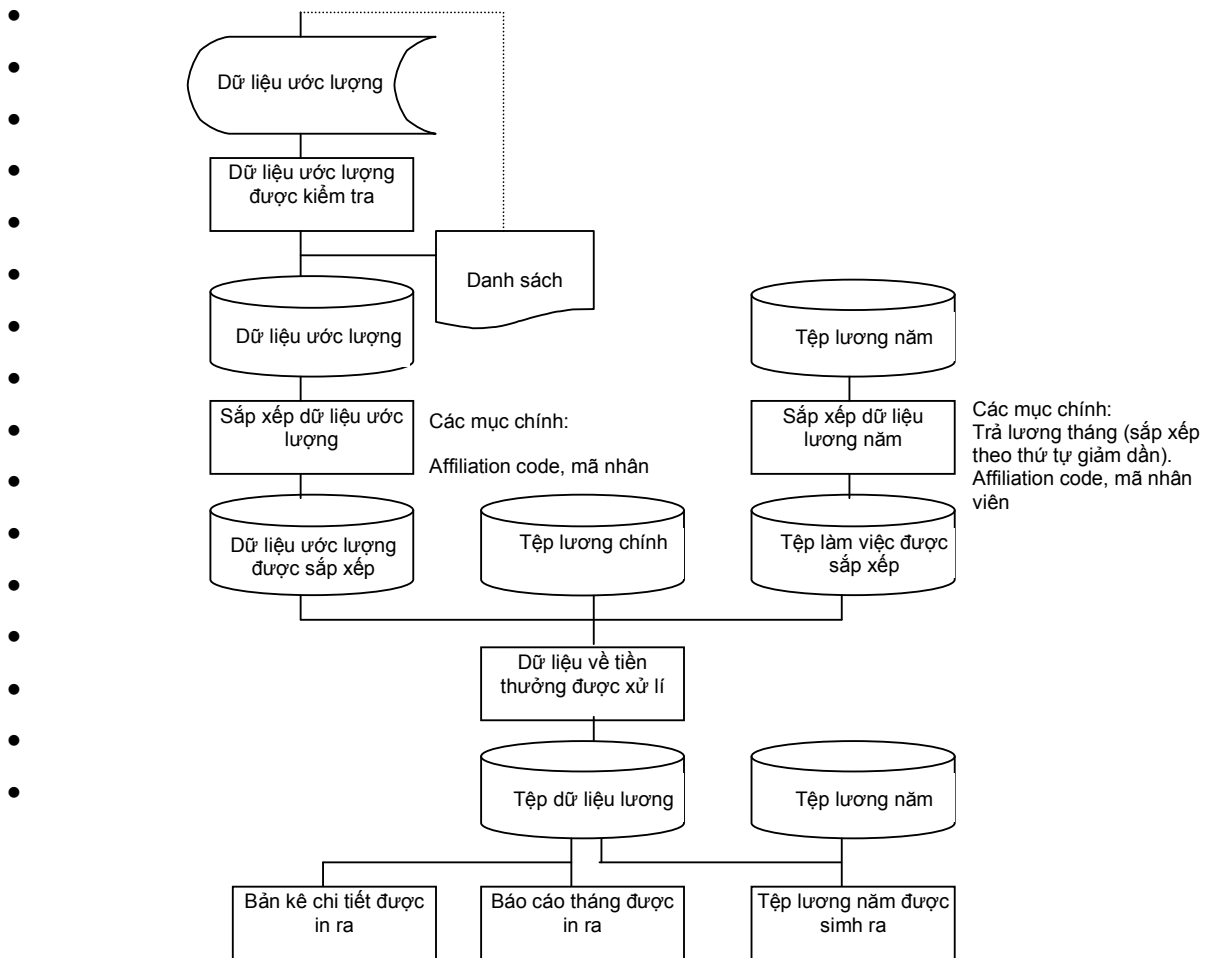
Kí hiệu	Tên	Mô tả
	Dữ liệu	Chỉ ra dữ liệu mà với nó phương tiện không được xác định.
	Dữ liệu được cất giữ	Chỉ ra dữ liệu được giữ theo định dạng thích hợp cho xử lý (phương tiện không được xác định).
	Bộ nhớ trong	Chỉ ra dữ liệu được cất giữ trong phương tiện bộ nhớ trong.
	Bộ nhớ truy nhập tuần tự	Chỉ ra dữ liệu truy nhập được theo cách tuần tự, như trên băng từ và băng cassette.
	Bộ nhớ truy nhập trực tiếp	Chỉ ra dữ liệu có thể truy nhập trực tiếp, như dữ liệu trên đĩa từ và đĩa mềm.
	Tài liệu	Chỉ ra dữ liệu trên phương tiện con người có thể đọc được.
	Cái vào thủ công	Chỉ ra dữ liệu trên bất kì phương tiện nào được đưa vào bằng các phương thức thủ công, như bàn phím trực tuyến, chuyển mạch, nút bấm hay mã vạch.
	Hiển thị	Chỉ ra dữ liệu trên bất kì phương tiện nào cho phép hiển thị thông tin con người dùng, như trên màn hình hiển thị hay bộ chỉ báo trực tuyến.
	Tiến trình	Chỉ ra bất kì kiểu chức năng xử lý nào.
	Tiến trình định trước	Chỉ ra tiến trình bao gồm một hay nhiều thao tác hay lệnh được xác định, chẳng hạn, một trình con.
	Thao tác thủ công	Chỉ ra bất kì thao tác nào được giải quyết thủ công.
	Chuẩn bị	Chỉ ra việc thay đổi của một lệnh hay nhóm lệnh, giống như việc thiết lập chuyển mạch hay thiết lập ban đầu cho một trình, mà ảnh hưởng tới các thao tác về sau.
	Quyết định	Chỉ ra một chức năng quyết định mà với một cổng vào và các cổng ra, chỉ một trong chúng có thể được lựa tại một thời điểm, việc lựa một cổng ra tùy thuộc vào kết quả của việc tính điều kiện được mô tả trong kí hiệu.
	Giới hạn chu trình	Bao gồm hai phần; phần điều kiện cho việc kết thúc và tên chu trình được viết trong kí hiệu chỉ ra chỗ bắt đầu và kết thúc của chu trình tương ứng.

• —	• Đường	• Chỉ ra luồng dữ liệu hay điều khiển.
• ↗	• Móc nối truyền thông	• Chỉ ra rằng dữ liệu được truyền trên đường truyền thông.
• ○	• Bộ ghép nối	• Chỉ ra một cổng đưa ra tới một chỗ khác trên lưu đồ hay một cổng đưa vào từ một chỗ khác.
• ○	• Thiết bị đầu cuối	• Chỉ ra một cổng đưa ra, hay một cổng đưa vào từ môi trường ngoài.
• ┌ • —	• Bình luận, chú thích	• Được dùng để thêm lời giải thích hay lưu ý để làm sáng tỏ.
• —····—	• E líp	• Được dùng thay cho kí hiệu đường, và chỉ ra rằng một kí hiệu hay nhóm kí hiệu bị bỏ qua.
•	•	• *JIS X0121
•	•	•

- Lưu đồ bao gồm hai kiểu sau:
- Lưu đồ hệ thống (sơ đồ tiến trình)
- Lưu đồ chương trình

Hình 1-2-5 Các kí hiệu được dùng trong lưu đồ

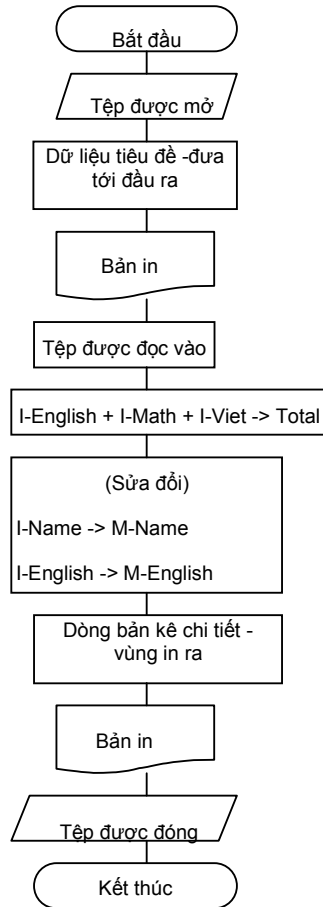
- 1. Lưu đồ hệ thống (sơ đồ tiến trình)
- Lưu đồ hệ thống (sơ đồ tiến trình) chỉ ra lưu đồ cho hệ thống cần nhắm tới như một tổng thể.
- Hình 1-2-6 Ví dụ về lưu đồ hệ thống



-
-
-
-
-
-

• 2. Lưu đồ chương trình

- Khi chương trình được phát triển, thì lưu đồ chương trình được dùng để mô tả cho các thủ tục xử lý dựa trên kết quả phân tích bằng lưu đồ.



Hình 1.2.7 Lưu đồ chương trình

(2) Sơ đồ có cấu trúc

- Với các lưu đồ, có thể mô tả các thuật toán dựa trên định lý cấu trúc. Tuy nhiên, việc dùng các kí hiệu mũi tên trong lưu đồ (nói cách khác, cho phép dùng các câu lệnh GOTO) có thể dẫn tới việc sinh ra thuật toán phi cấu trúc. Mặt khác, trong các sơ đồ có cấu trúc, các thuật toán được mô tả không dùng kí hiệu mũi tên nào (hay câu lệnh GOTO). Do đó, các mô tả dựa trên định lý cấu trúc là có thể dùng được.

- Các sơ đồ có cấu trúc bao gồm các kiểu khác nhau sau đây:
- Sơ đồ NS
- PAD
- SPD
- HCP

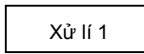
• YAC II

• 1. Sơ đồ NS (sơ đồ Nassi-Shneiderman)

• ND sơ đồ dùng các kí hiệu đặc biệt có các cấu trúc điều khiển cho việc biểu diễn.

•

• <Cấu trúc tuần tự>



•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

•

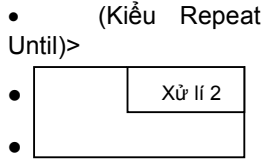
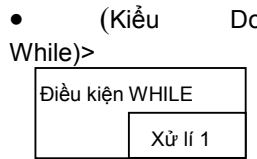
•

•

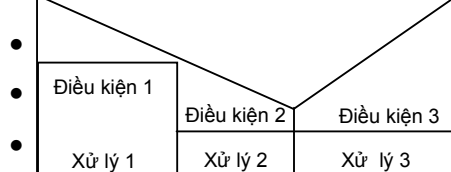
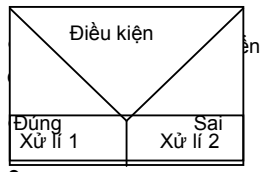
• <Cấu trúc lặp (Kiểu Do While)>

• <Cấu trúc lặp (Kiểu Repeat Until)>

• <Cấu trúc lặp (Kiểu Repeat Until)>



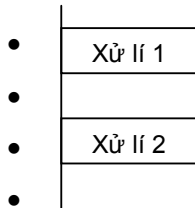
Hình 1-2-8 Ví dụ về sơ đồ NS



• 2. PAD (Problem Analysis Diagram - Biểu đồ phân tích vấn đề)

• PAD mô tả cho cấu trúc logic của thuật toán bằng cấu trúc cây.

• <Cấu trúc tuần tự>



•

•

•

•

•

•

•

•

•

•

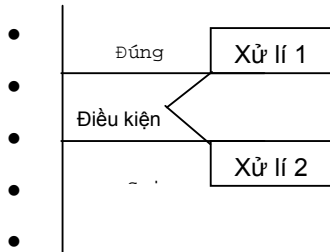
•

•

•

•

• <Cấu trúc tuyển chọn>



•

•

•

•

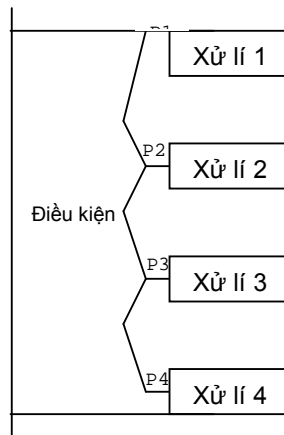
•

•

•

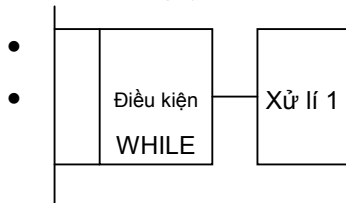
•

<Cấu trúc CASE>



Hình 1-2-9 Ví dụ về PAD

• <Cấu trúc lặp (kiểu Do While)>



•

•

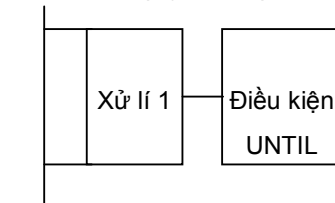
•

•

•

•

• <Cấu trúc lặp (kiểu Repeat Until)>



•

•

•

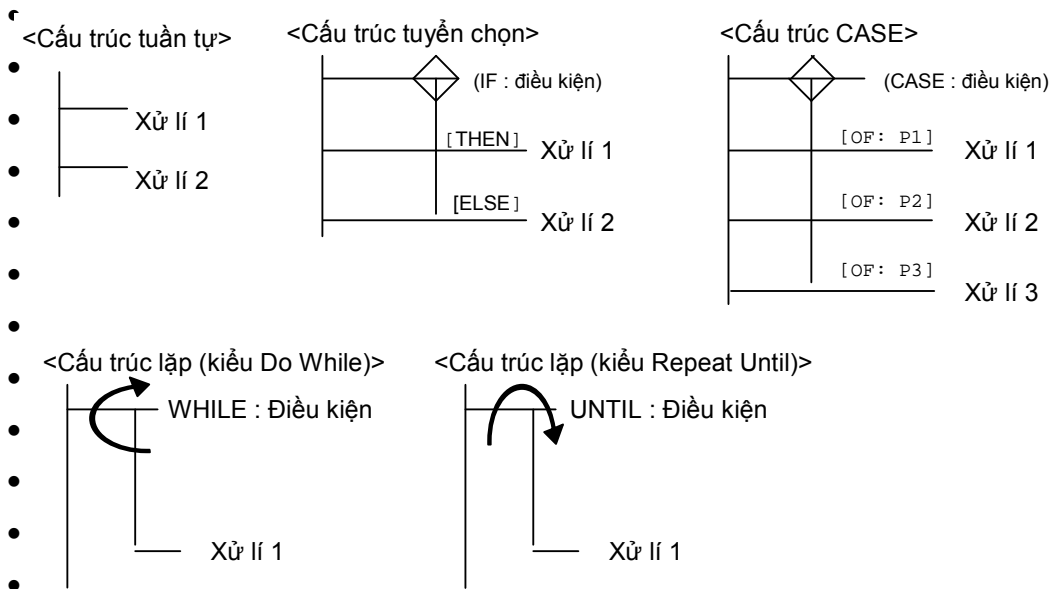
•

•

•

-
-
-
-

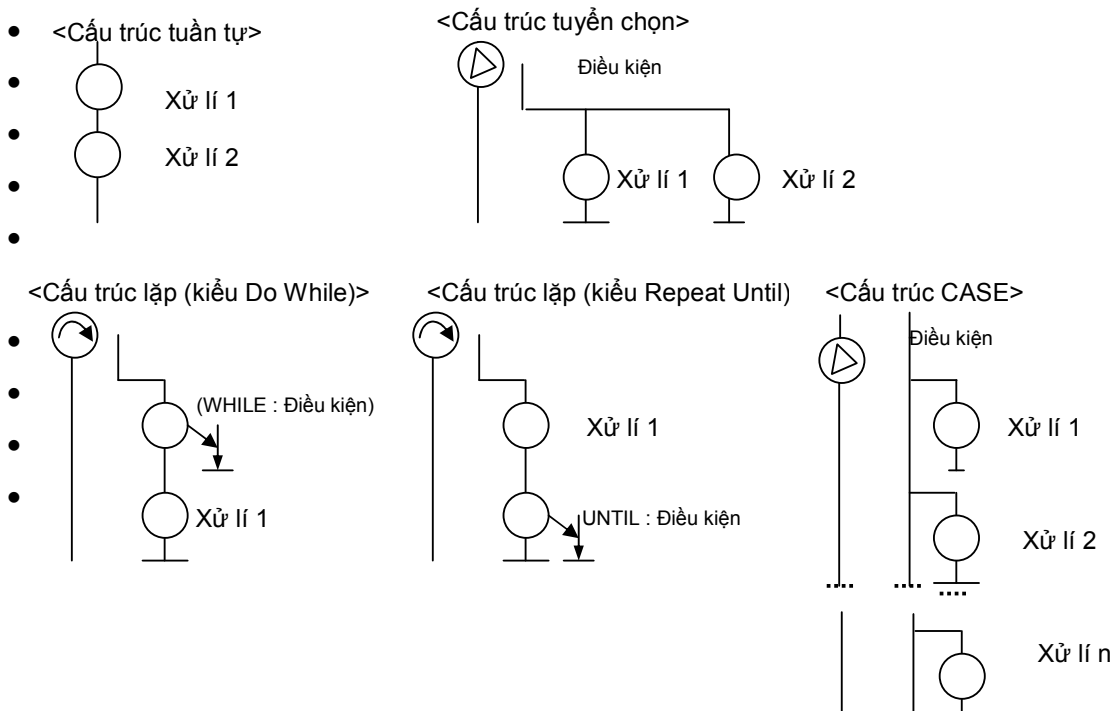
- 3. SPD (Structured Programming Diagram) Biểu đồ lập trình có cấu trúc
- SPD mô tả cấu trúc logic của thuật toán bằng sơ đồ cây có cấu trúc.



Hình 1-2-10 Ví dụ về SPD

- 4. HCP (Sơ đồ mô tả gọn và có thứ bậc - Hierarchical and Compact description chart)
- HCP mô tả theo cùng cách như SPD, cấu trúc logic của thuật toán bằng sơ đồ cấu trúc cây.

Hình 1-2-11 Ví dụ về HCP

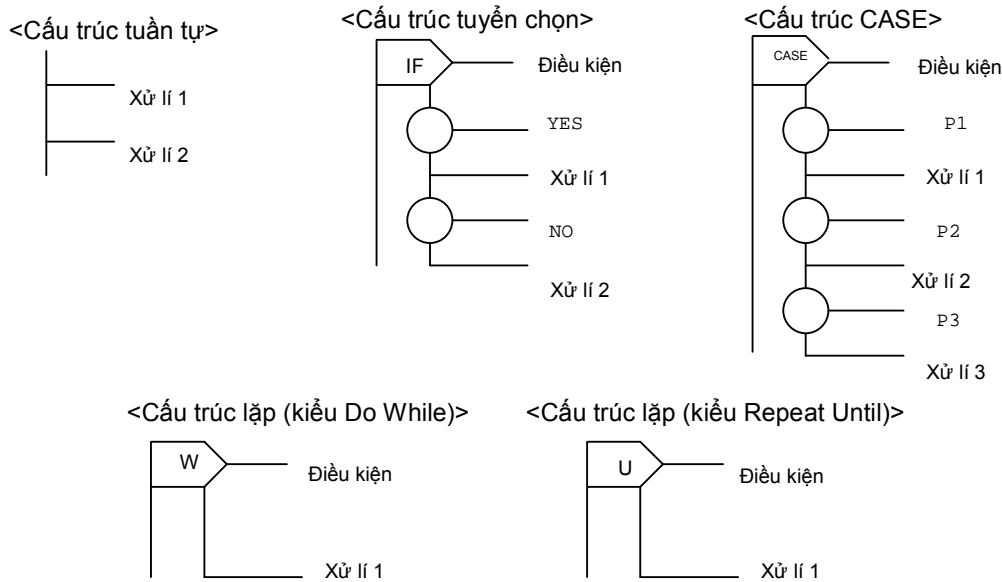


-
-
-
-
-

5. YAC II (Yet Another Control chart II Sơ đồ điều khiển khác II)

- YAC II cũng mô tả cấu trúc logic của thuật toán có sơ đồ có cấu trúc.

Hình 1-2-12 Ví dụ về YAC II



(3) Bảng quyết định

- Bảng quyết định được dùng để phân tích và thu xếp cấu trúc logic của các mô đun, và mô tả tất cả các nhân tố từ việc xác định vấn đề cho tới giải pháp của chúng bằng bảng 2 chiều đơn giản. Bảng quyết định cho phép thu xếp đơn giản các mối quan hệ giữa các điều kiện và xử lí, thậm chí cho những vấn đề phức tạp và bao gồm nhiều nhân tố. Nó tạo khả năng chuyển dịch trơn tru sang mã hoá dựa trên định lí cấu trúc.

<ul style="list-style-type: none"> • Điều kiện • Các điều kiện cần được kiểm tra cho những vấn đề xuất hiện được làm thành dạng câu hỏi 	<ul style="list-style-type: none"> • Ô điều kiện • Trả lời cho câu hỏi từ một cuống điều kiện được mô tả bằng Y (có) hay N (không)
<ul style="list-style-type: none"> • Cuống hành động • Tất cả những hành động có thể được tiến hành cho vấn đề đã được xác định trong cuống điều kiện được mô tả rõ ràng 	<ul style="list-style-type: none"> • Ô hành động • Thay vì Y hay N cho ô điều kiện, X được viết chỉ khi hành động được tiến hành
<ul style="list-style-type: none"> • Làm việc vào ngày làm việc 	<ul style="list-style-type: none"> • Y • Y • N • N
<ul style="list-style-type: none"> • Làm việc vào ngày nghỉ 	<ul style="list-style-type: none"> • N • N • Y • Y

• Làm việc từ: 09:00 tới 21:00	• Y	• N	• Y	• N
• Làm việc từ: 21:00 tới 09:00	• N	• Y	• N	• Y
• Trả theo giờ: 1 000	• X	• X		•
• Trả theo giờ: 1 200			• X	• X
• Phụ cấp: 1 000 cho việc tham dự		• X		• X

- Hình 1.2.13 Bảng quyết định
-

1.2.3 Phương pháp thiết kế

(1) Phân tích và thiết kế có cấu trúc

• Phương pháp có cấu trúc là một thuật ngữ tổ hợp cả phân tích có cấu trúc và thiết kế có cấu trúc. Với phân tích có cấu trúc (SA), các yêu cầu về hệ thống đích cần phát triển được phân tích bằng việc chú ý đặc biệt tới chức năng của hệ thống, và luồng dữ liệu giữa các chức năng. Phương pháp phân tích này dùng ba tài liệu sau biểu diễn về mặt sơ đồ cho thế giới nghiệp vụ ứng dụng với luồng dữ liệu cho phân tích.

- - Biểu đồ luồng dữ liệu (DFD)
 - • Cho phép mô tả theo phân cấp, tạo khả năng phân chia phân cấp về các chức năng hệ thống.
- - Từ điển dữ liệu
 - • Được dùng để định nghĩa các cấu trúc dữ liệu của những dữ liệu có tên được bổ sung vào luồng dữ liệu trong DFD.
- - Đặc tả mini
 - • Việc phân chia có phân cấp tiếp tục cho các chức năng dẫn tới các thao tác cơ sở. Những thao tác cơ sở này được định nghĩa trong đặc tả mini.

• Phân tích có cấu trúc nguyên thủy chỉ ngụ ý cho việc mô tả dữ liệu và tiến trình bằng biểu đồ. Tuy nhiên De Marco đã bổ sung thêm việc dùng từ điển dữ liệu và đặc tả mini vào phương pháp nguyên thủy. Kết quả là việc phân tích có cấu trúc bây giờ đã được dùng như một phương pháp luận.

• Mục đích của phương pháp thiết kế có cấu trúc (SD) là để chuyển các tiến trình trong DFD được sinh ra bởi phân tích có cấu trúc thành các mô đun chương trình, và tiến hành phân chia thành các mô đun bằng cách tiếp cận trên xuống. Phương pháp phân chia STS (Source Transform Sink), hay phương pháp phân chia giao tác, được dùng để phân chia thành các mô đun.

(2) Thiết kế hướng tiến trình

• Phần mềm bao gồm hai yếu tố, chức năng và thông tin (dữ liệu). Trong việc phát triển phần mềm truyền thống, phân tích và thiết kế đã được tiến hành đặt trọng tâm vào chức năng do phần mềm thực hiện. Điều này được gọi là thiết kế hướng tiến trình hay cách tiếp cận hướng thiết kế. Trong phương pháp này, không có chuẩn rõ ràng để định nghĩa đơn vị chức năng, và do vậy việc định nghĩa này có thể dễ dàng bị ảnh hưởng bởi cách nghĩ riêng của người thiết kế. Bên cạnh đó, khi các chức năng đáp ứng cho yêu cầu của người dùng được thực hiện, thì khó điều chỉnh các yêu cầu cho nhiều người dùng, và việc sử dụng các chức năng chồng chéo nhau là không thể tránh khỏi. Kết quả là hệ thống bao gồm nhiều chức năng chồng chéo nhau, là một trong những nhân tố làm cho việc bảo trì thành khó khăn.

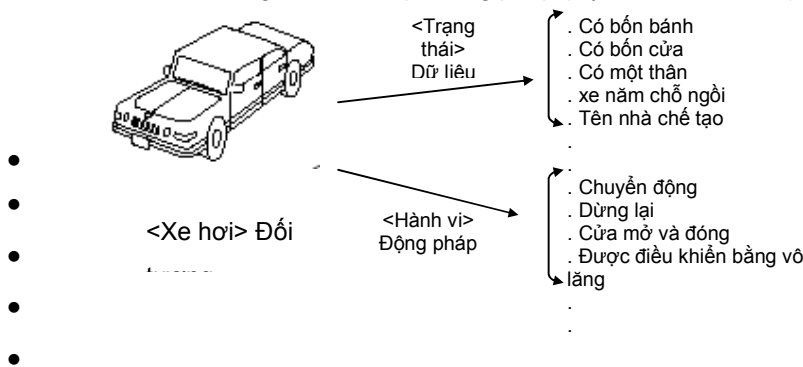
(3) Thiết kế hướng dữ liệu

- Dữ liệu không thay đổi bởi vì các yêu cầu hay đòi hỏi của người dùng về thao tác nghiệp vụ, nhưng tồn tại trong thế giới thực bất kể đến kiểu, hay mối quan hệ, của các thao tác. Vậy, trong thiết kế tập trung vào dữ liệu, sự chú ý được nhằm vào dữ liệu, và hệ thống được thiết kế dựa trên cấu trúc tiến trình dữ liệu. Hơn nữa, phương pháp thiết kế này đã tiến hoá thành Cách tiếp cận hướng dữ liệu (DOA). Việc phân tích và thiết kế được tiến hành cho dữ liệu, được tách bạch với yêu cầu hay đòi hỏi của người dùng về thao tác, và do vậy các tiến trình được xác định và tích hợp vào trong các thủ tục chuyên dụng dữ liệu.

(4) Thiết kế hướng đối tượng

- Thiết kế hướng đối tượng được suy ra bằng việc nâng cao hơn cách tiếp cận hướng dữ liệu.
- ① Khái niệm thiết kế hướng đối tượng
- Chúng ta hãy xem xét khái niệm hướng đối tượng qua xe hơi lấy làm ví dụ.

Hình 1-2-14 Đối tượng, dữ liệu và phương pháp (lấy xe hơi làm ví dụ)



- Ngay cả việc dùng một từ "xe hơi," điều đó cũng bao hàm mỗi xe có trạng thái đặc biệt của nó, hoàn toàn khác với bất kì xe nào khác, đã bao hàm có kiểu xe, năm sản xuất, mô tô, hình dáng cửa, lượng phụ tùng thay thế. Bên cạnh đó, xe bao gồm các hành vi đa dạng như "chuyển động," "dừng," "mở cửa" và "đóng cửa." Vậy, việc dùng các cấu trúc và hành vi nền tảng của một "đối tượng" cho việc phát triển hệ thống là một đặc trưng chủ chốt của thiết kế hướng đối tượng.

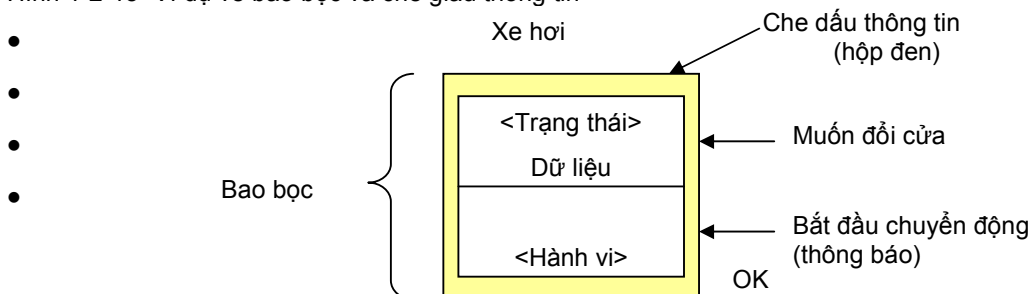
- Các mối quan hệ sau đây thu được khi các thuật ngữ được dùng trong thiết kế hướng đối tượng được áp dụng ví dụ trong Hình 1-2-14.

- Tiêu bản của xe hơi: Class
- Một xe riêng lẻ: Thể hiện hay đối tượng
- Trạng thái: Dữ liệu (Thuộc tính)
- Hành vi: Động pháp (Thủ tục)

- Việc tích hợp "trạng thái" và "hành vi" được gọi là "bao bọc."

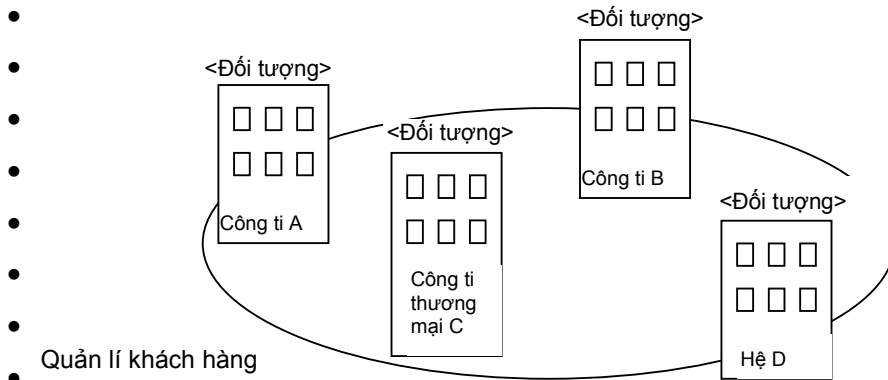
- Hơn nữa, việc che giấu "trạng thái" và "hành vi" bằng việc chứa chúng trong hộp đen được gọi là "che giấu thông tin" Điều này dẫn tới việc cấm truy nhập trực tiếp vào dữ liệu, làm tăng sự độc lập của đối tượng và làm cho việc dùng lại được dễ dàng hơn.

Hình 1-2-15 Ví dụ về bao bọc và che giấu thông tin



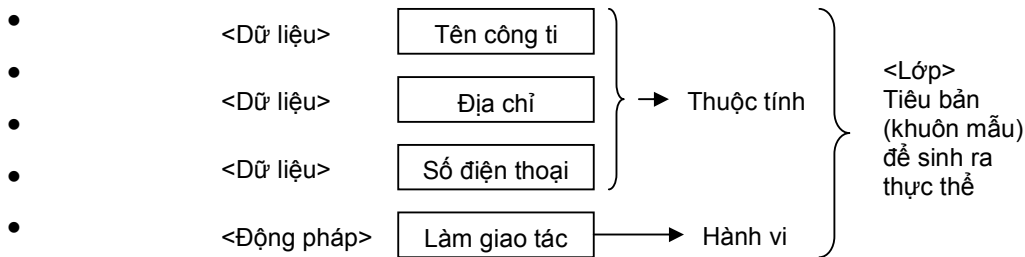
-
- ② Đối tượng/lớp/thể hiện
- Chẳng hạn, khi một ứng dụng quản lý khách hàng được phát triển bằng thiết kế hướng đối tượng, từng khách hàng được xử trí giống như một đối tượng.
-
-

Hình 1-2-16 Ví dụ về đối tượng (khách hàng)



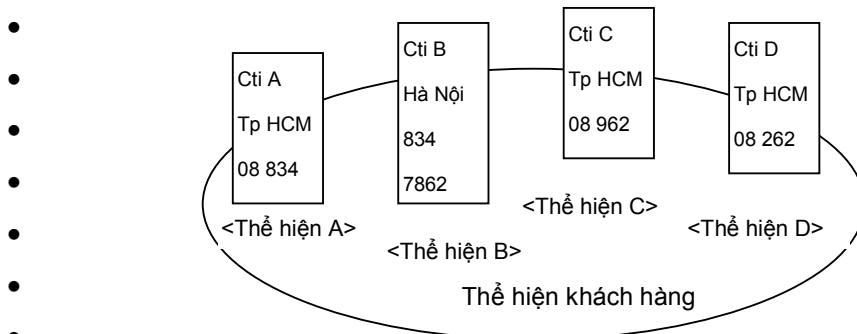
- Cái được định nghĩa như thuộc tính và hành vi tổng quát của khách hàng được gọi là "Lớp." Lớp được định nghĩa bằng việc lấy ra các phần (các đặc trưng) mà nhiều đối tượng tương tự có chung. Cho nên, một lớp có thể được gọi là một loại tiêu bản (khuôn đúc). Mặt khác, một hành vi riêng cho một đối tượng được gọi là "Động pháp."
-

Hình 1-2-17 Ví dụ về lớp



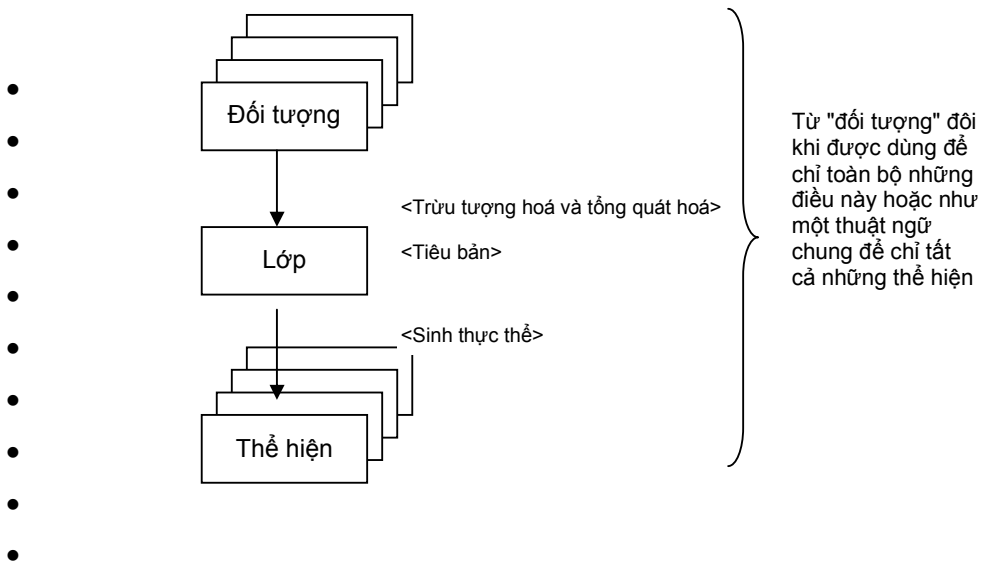
- Từng thực thể được sinh ra từ một lớp được gọi là "thể hiện."

Hình 1-2-18 Ví dụ về thể hiện (thực thể)



-
-
-
-
-
-
-
-

- Hình 1-2-19 chỉ ra mối quan hệ giữa các đối tượng và lớp, và thể nghiệm được mô tả ở trên.



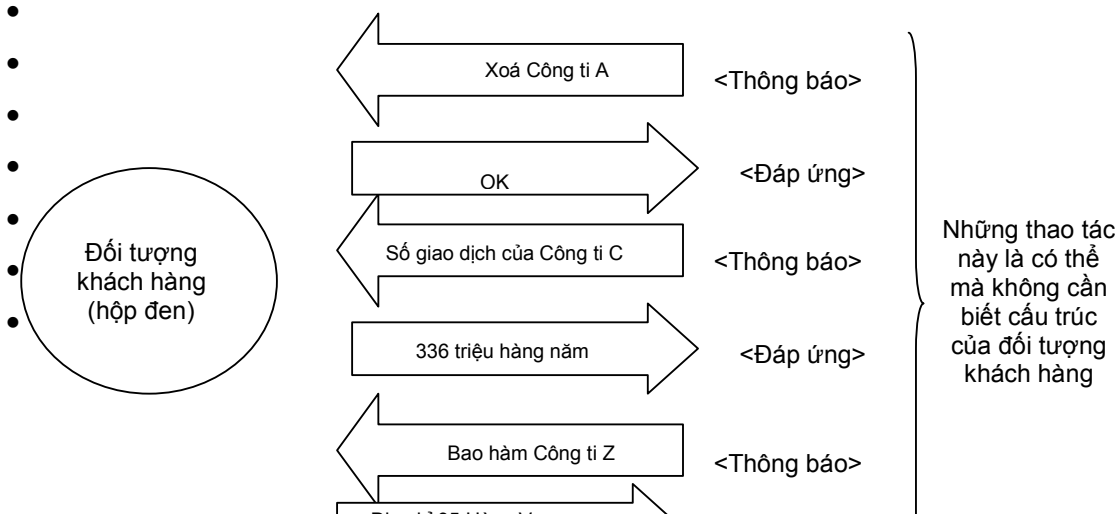
- Hình 1-2-19 Mối quan hệ giữa các đối tượng và lớp và thể nghiệm

③ Thông báo

- Thông báo là phương tiện duy nhất sẵn có để ra lệnh cho đối tượng. Trong thiết kế hướng đối tượng, việc truy nhập vào đối tượng bằng phương tiện khác hơn thông báo là bị cấm, do vậy làm tăng sự độc lập và tính tin cậy của đối tượng. Bên cạnh đó, có thể xem xét một đối tượng như hộp đen, có khả năng thực hiện những thao tác cần thiết mà không biết tới nội dung của đối tượng.

-

Hình 1-2-20 Ví dụ về thông báo



-
-
-
-
-
-

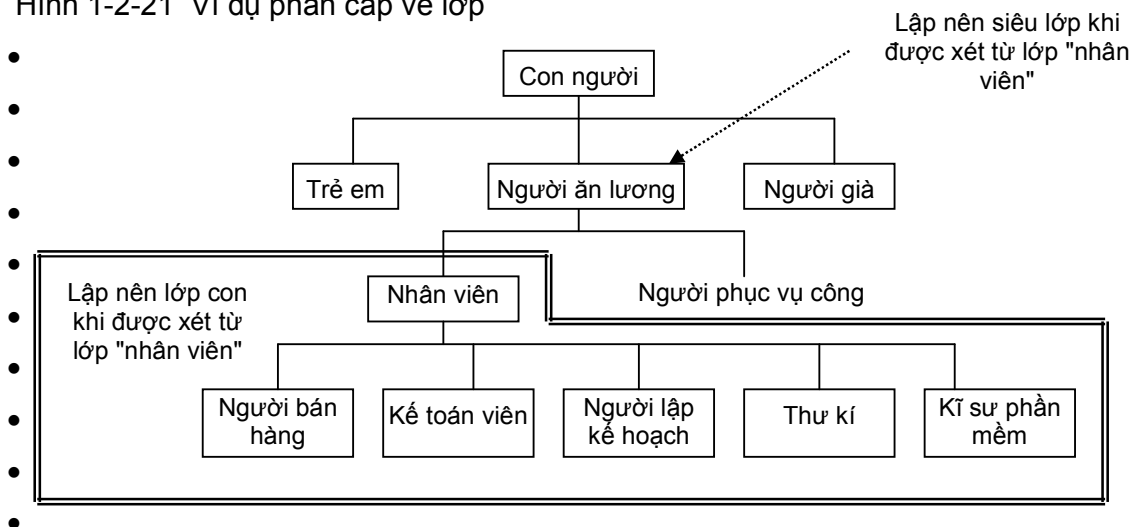
④ Kế thừa

Nếu việc tăng năng suất được coi như một mục đích trong khi dùng thiết kế hướng đối tượng, thì "Kế thừa" đưa ra một phương tiện hữu dụng. Kế thừa chỉ ra việc kế thừa dữ liệu (trạng thái) và thủ tục (hành vi) từ các lớp khác (chủ yếu là lớp cấp trên).

Lớp được đặt trên một lớp được gọi là "Siêu lớp," trong khi lớp đặt bên dưới là "Lớp con".

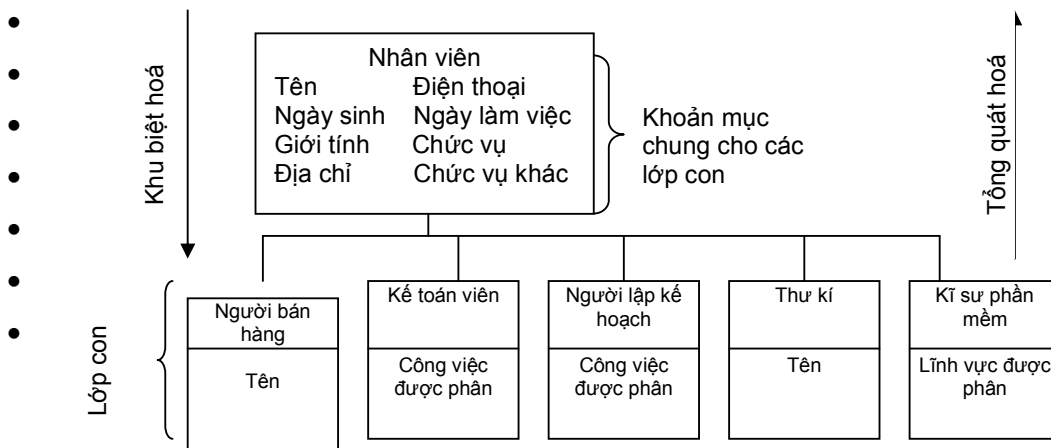
-

Hình 1-2-21 Ví dụ phân cấp về lớp



Trong ví dụ của Hình 1-2-21, lớp người hưởng lương được đặt trên lớp nhân viên tạo nên một siêu lớp, trong khi người bán hàng, người lập kế hoạch kế toán, thư kí và các lớp SE được định vị dưới lớp nhân viên lại tạo nên các lớp con. Mỗi một trong các lớp con này lại bao hàm cùng các khoản mục như tên, giới tính, và địa chỉ, chỉ vài khoản mục còn phải được xác định tách biệt. Dùng kế thừa là để nhằm làm tăng năng suất bằng việc chỉ xác định các khoản mục cần được xác định tách biệt trong từng lớp (việc lập trình cho điều này được gọi là lập trình khác biệt) và bằng việc kế thừa các khoản mục chung cho các lớp con từ siêu lớp "nhân viên."

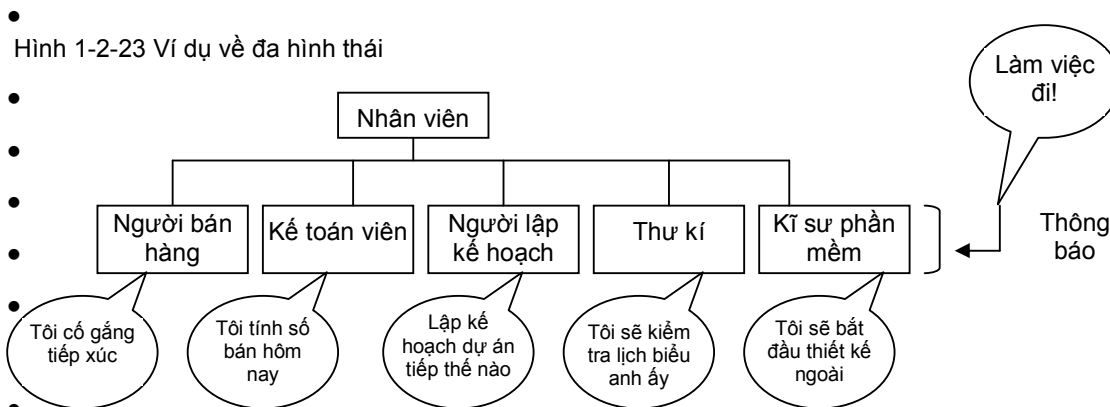
Hình 1-2-22 Ví dụ kế thừa



-
-
- Khoản mục chung cho những lớp con này được rút ra và định nghĩa trong lớp nhân viên làm tăng các khoản mục được xác định tách biệt và tăng năng suất
- Việc xác định những khoản mục chung cho các lớp con trong siêu lớp được gọi là "tổng quát hoá", trong khi việc phân chia một lớp và xác định tiếp các lớp con được gọi là "khu biệt hoá." Những quan hệ này được gọi là "các quan hệ tổng quát hoá và khu biệt hoá" hay "quan hệ là_một" (Hình 1-2-21 đưa ra một ví dụ như vậy).

⑤ Đa hình thái

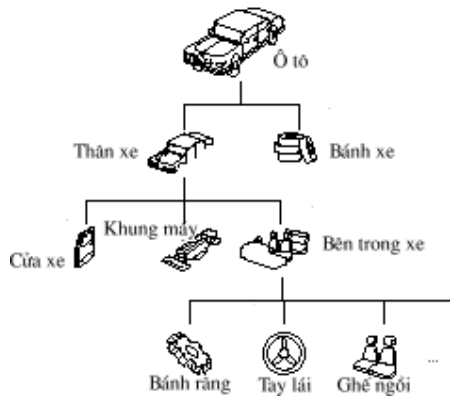
- Đa hình thái chỉ ra rằng từng đối tượng nhận cùng thông báo sẽ đáp ứng theo cách hành xử khác nhau.



- Mỗi nơi nhận thông báo "Làm việc đi!"
- sẽ đáp ứng hay hành động khác nhau
- Đa hình thái có quan hệ với bao bọc các đối tượng. Trong ví dụ trong Hình 1-2-23, điều người gửi thông báo phải làm là ban ra một thông báo, "Hãy làm đi!," bất kể tới sự phản đối của nơi nhận. Bên cạnh đó, nếu kiểu công việc khác được thêm vào cho phía nhận thông báo, thì những sửa đổi cần thiết bị giới hạn vào phía nhận, không thay đổi nào được yêu cầu từ người gửi. Vậy, đa hình thái, trong khi đảm bảo sự độc lập cho cả hai phía, lại cung cấp những hiệu quả có ý nghĩa

⑥ Đối tượng hợp thành

- Một đối tượng được tạo nên bằng cách tổ hợp nhiều đối tượng được gọi là "đối tượng hợp thành".



Hình 1-2-24
Ví dụ về đối tượng hợp thành

- Trong ví dụ của Hình 1-2-24, chiếc xe hơi được hợp thành từ những bộ phận đa dạng. Nếu từng bộ phận lại được coi là đối tượng, thì xe hơi có thể được gọi là đối tượng hợp thành. Một cấu trúc như vậy được gọi là "quan hệ kết tập và phân tách," hay "bộ phận-của quan hệ"

⑦ Thư viện lớp

- Thư viện lớp theo đúng từng từ nghĩa là một thư viện các lớp. Mục đích của thiết kế hướng đối tượng là để làm tăng năng suất. Do đó, cần khởi xướng việc dùng lại bằng cách chuẩn hoá thiết kế lớp nhiều nhất có thể được. Hệ quả là, thư viện lớp cung cấp các lớp chất lượng cao trở thành quan trọng.

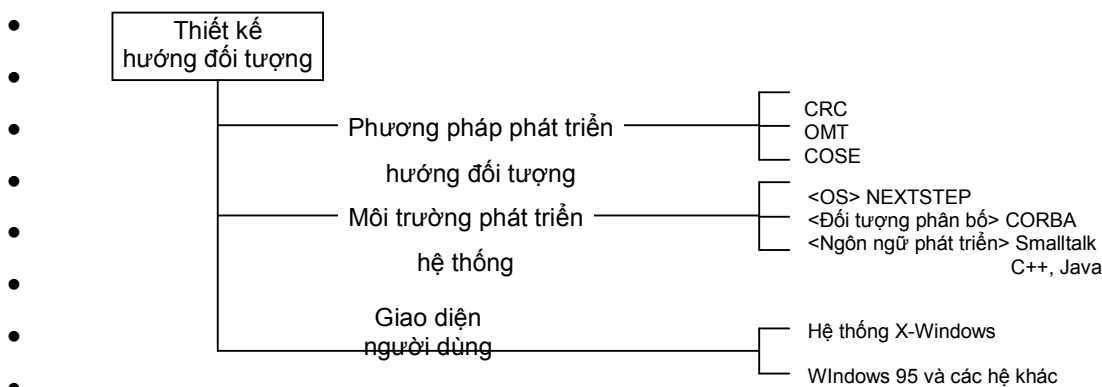
⑧ Ủy quyền

- Nếu một đối tượng nhận thông báo không thể giải quyết được điều được xác định bởi thông báo này, thì đối tượng này gửi thông báo tới đối tượng khác để yêu cầu xử lý. Một hành động như vậy được gọi là "Ủy quyền". Trong ngôn ngữ hướng đối tượng không dùng cấu trúc kế thừa, ủy quyền được dùng vào chỗ của cấu trúc kế thừa. Vì nơi nhận ủy quyền có thể được thay đổi động trong khi xử lý, nên ủy quyền tổng quát hơn cấu trúc kế thừa.

⑨ Ví dụ về dùng có hiệu quả thiết kế hướng đối tượng

- Gần đây, các hệ thống trong nhiều lĩnh vực khác nhau đã được phát triển bằng việc dùng thiết kế hướng đối tượng. Kết quả là, cái ra từ những phát triển này đã được dùng có hiệu quả trong nhiều lĩnh vực khác nhau. Hệ thống của sở là ví dụ quen thuộc về việc dùng có hiệu quả thiết kế hướng đối tượng. Điều này là vì thiết kế hướng đối tượng được làm phù hợp để xây dựng các hệ thống "điều khiển theo biến cố".

- Hình 1-2-25 Ví dụ về dùng thiết kế hướng đối tượng có hiệu quả



- Trong Hình 1-2-25, CORBA (Common Object Request Broker Architecture) nêu ra các đặc tả chuẩn để dùng công nghệ hướng đối tượng điều phối hoạt động của các ứng dụng trên nhiều máy

tính (phục vụ), được đặt và phân bố trên toàn mạng để thực hiện một phần xử lí. Điều này có thể được gọi là một loại hệ thống khách phục vụ.

(5) Thiết kế mô đun

- Trong thiết kế có cấu trúc, việc phân hoạch chức năng được thực hiện dựa trên cơ sở phân tích đặc tả yêu cầu, và từng chức năng được phân hoạch thì được xác định như một chương trình. Một chương trình đã được xác định sẽ được phân hoạch thêm nữa thành một số mô đun trong thiết kế mô đun. Mô đun được xác định như sau:
- Mô đun là tuyển tập các lệnh được dùng để đạt tới chức năng và có thể được dịch một cách độc lập.
- Các mô đun đã dịch có thể được gọi bởi các mô đun khác, và các đối được dùng làm giao diện.
- Hơn nữa, tiêu chí để phân hoạch chương trình thành mô đun được tạo ra để làm tăng sự độc lập giữa các mô đun. Các hiệu quả sau đây được trông đợi để giảm mức độ phụ thuộc giữa các mô đun và làm tăng mức độ độc lập tương ứng của chúng.
- Một cấu trúc được đưa vào trong một nhóm các mô đun, và mối quan hệ giữa chúng trở thành dễ hiểu.
- Mức độ theo đó việc thay đổi hay sửa chữa mô đun ảnh hưởng tới các mô đun khác bị rút đi, làm tối thiểu phạm vi tác động kết quả.
- Nó có thể được dùng như một nguyên tắc hướng dẫn có hiệu quả để tạo ra phần mềm chất lượng cao.
- Hai biện pháp sau đây được dùng làm tiêu chuẩn để đánh giá mức độ độc lập của phần mềm (xem Hình 1.2.26):

1. Tính cố kết mô đun

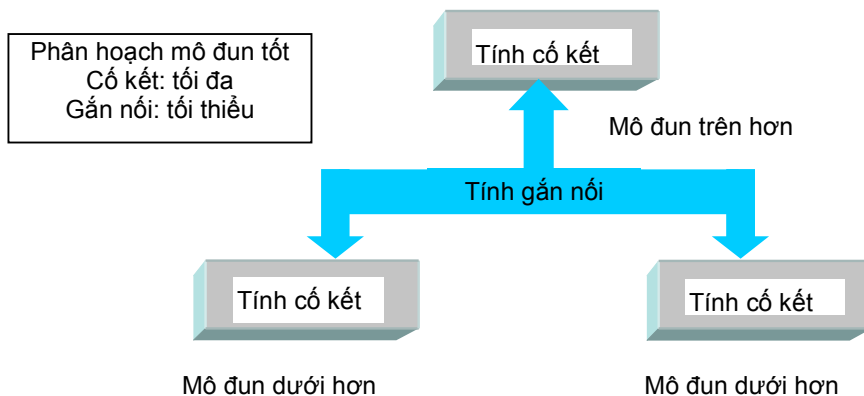
- Tính cố kết mô đun chỉ ra sức mạnh theo đó các cấu phần tạo nên mô đun được gắn với nhau, và các mô đun với mối quan hệ nội bộ mạnh hơn được coi là tốt hơn. Cấu hình của các tiến trình được trừ tính ở đây.

2. Tính gắn nối mô đun

- Tính gắn nối mô đun chỉ ra cường độ theo đó các mô đun gắn với nhau. Mối quan hệ này càng yếu thì các mô đun càng tốt. Các cách giải quyết đối được trừ tính ở đây.

Hình 1-2-26

Sự độc lập của mô đun

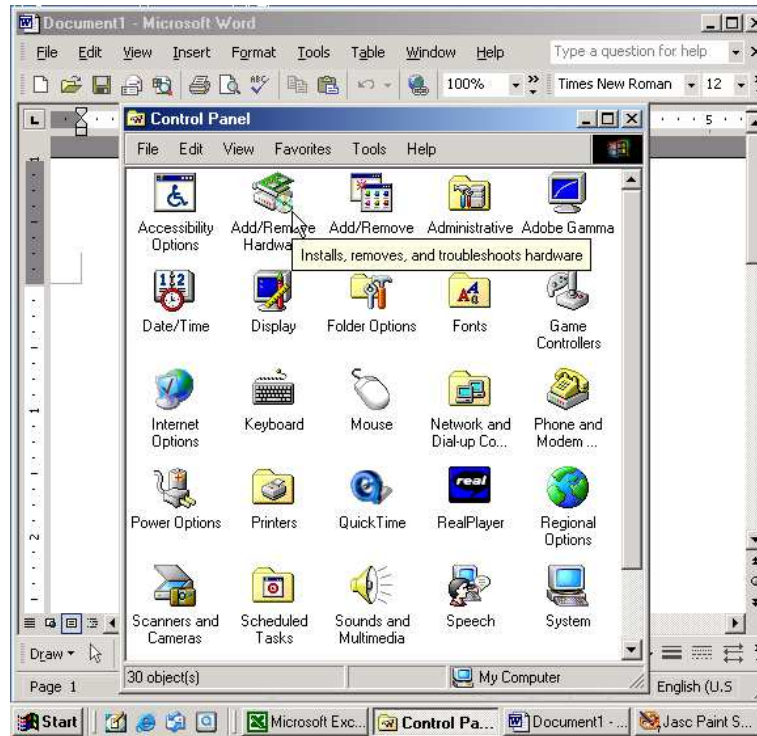


(6) Thiết kế màn hình

Với thiết kế giao diện con người thì thiết kế màn hình là đặc biệt quan trọng. Thiết kế màn hình quan trọng cho người dùng tới mức chất lượng hệ thống đôi khi được đánh giá chỉ theo mức độ dễ dùng màn hình thế nào. Sau đây sẽ đề cập tới định nghĩa về thiết kế màn hình và thủ tục của nó.

Hình 1-2-27

Ví dụ về GUI



1. Thiết kế màn hình là gì?

Trong một hệ thống, màn hình (nơi đưa dữ liệu vào) là phần quen thuộc nhất với người dùng, bởi vì họ thường xuyên tiếp xúc với nó. Gần đây, việc thiết kế màn hình đã được thực hiện bằng việc dùng GUI (Giao diện người dùng đồ họa) rất nhiều. GUI cung cấp một giao diện cho phép xử lý tương tác bằng việc dùng các biểu tượng dễ nhận dạng, các menu thả xuống và bật ra. Với thiết bị trở như chuột, mũi tên hay các kí hiệu khác, có thể di chuyển tới bất kì chỗ nào trên màn hình, việc xử lý vậy có thể được thực hiện bằng cách trở vào biểu tượng hay khoản mục trên menu.

Trong máy tính cá nhân, các chức năng GUI được xây dựng thành hệ điều hành, trong khi, với UNIX, lại cần cung cấp các chức năng GUI một cách tách biệt. Bản thân các biểu tượng, ý nghĩa của chúng và cách giải quyết chúng có thay đổi theo cách nào đó từ GUI nọ sang GUI kia.

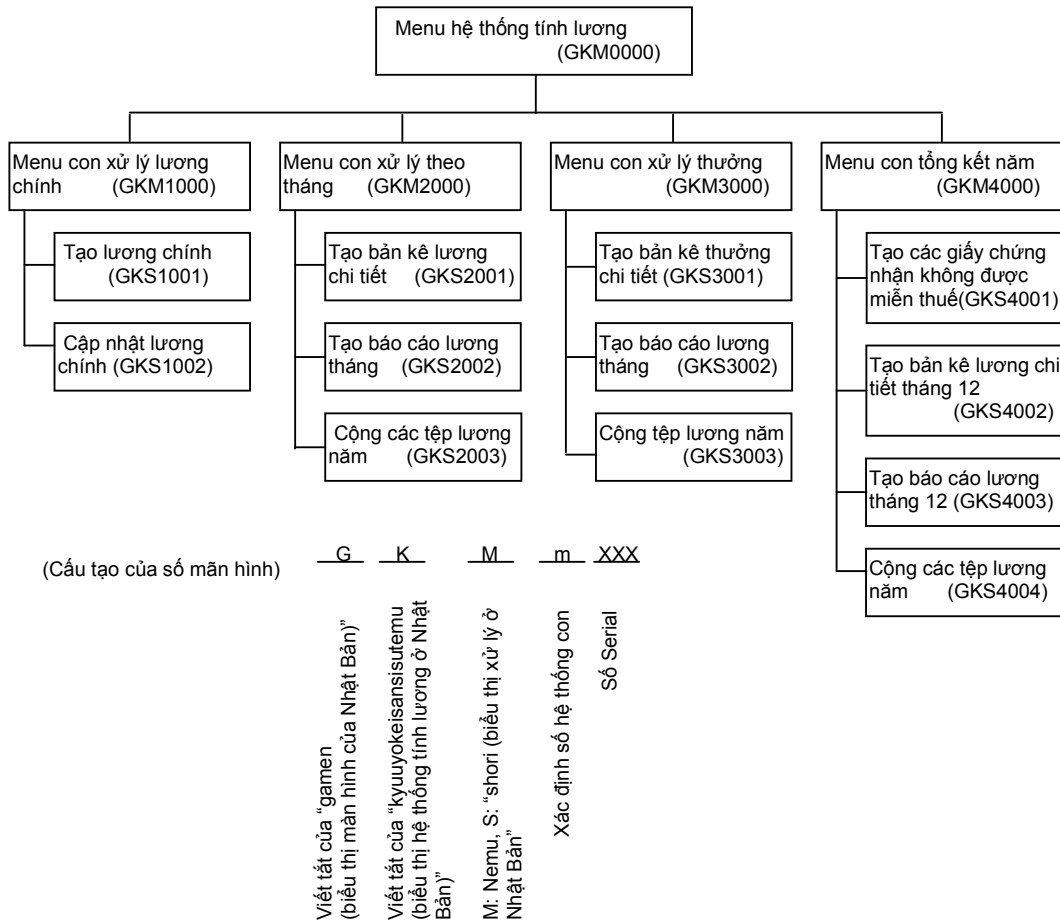
2. Các thủ tục thiết kế màn hình và công việc thiết kế thực tế

- Việc thiết kế màn hình được tiến hành bằng việc dùng các thủ tục sau:
- Một bức tranh toàn cảnh được tạo ra (toàn bộ biểu đồ tổ chức của tất cả các màn hình được dùng được tạo ra).
- Các thiết kế màn hình được chuẩn hoá (cách bố trí và luồng màn hình được chuẩn hoá).
- Luồng màn hình được thiết kế (trình tự theo đó thiết kế ra việc xử lý tương tác được giải quyết)
- Cách hiển thị các khoản mục trên màn hình được nghiên cứu (tần suất của từng màn hình được dùng, và mức độ kinh nghiệm của người dùng được xét tới)

- Cách bố trí màn hình được thiết kế (cách tổ chức và thu xếp trên màn hình được thiết kế)

Hình 1-2-28

Tạo ra bức tranh tổng thể



- a. Tạo ra bức tranh tổng thể

Tất cả các màn hình được dùng trong hệ thống định phát triển sẽ được lấy ra để xem xét mối quan hệ của chúng và cần tạo ra một sơ đồ tổ chức toàn bộ cho các màn hình này. Nói chung, sơ đồ này bao gồm hầu hết các menu, menu con và các tiến trình. Việc đánh số cho từng màn hình giúp cho việc quản lý chúng được dễ dàng hơn.

- b. Chuẩn hoá thiết kế màn hình

Việc chuẩn hoá cách bố trí màn hình (vị trí đặt các khoản mục) và luồng màn hình tạo khả năng làm tăng hiệu suất của cả phía người phát triển và người dùng.

- <Các khoản mục được chuẩn hoá>
- Các vị trí hiển thị (kể cả vị trí tiêu đề và khoản mục nội dung)
- Cách hiển thị (kể cả việc thống nhất cách dùng màu sắc, từ và câu)
- Cách đưa vào (phương pháp chọn, đưa vào trực tiếp v.v..)
- Dùng phím chức năng chương trình

- c. Thiết kế luồng màn hình

- Việc xử lý ứng dụng cần nhiều màn hình. Trong thiết kế luồng màn hình, trật tự hiển thị từng màn hình được thiết kế để thực hiện việc xử lý.

Hình 1-2-29

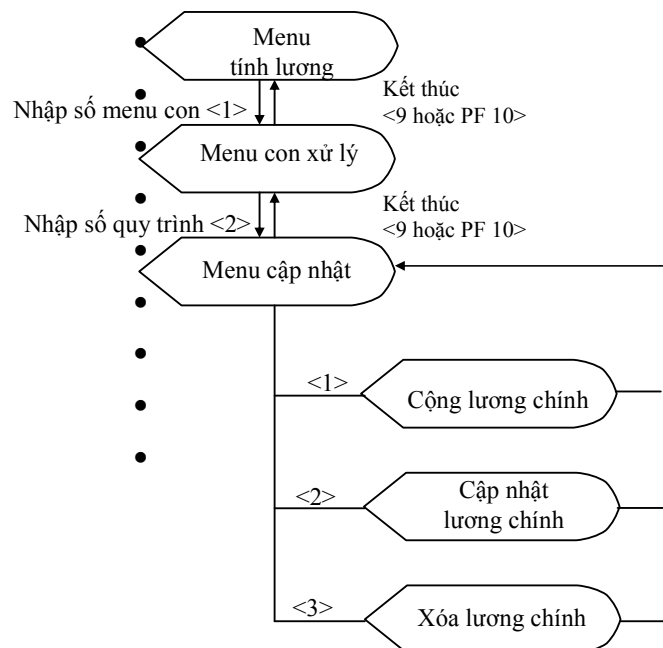
Chuẩn hoá bố trí màn hình (ví dụ)

(Số hiệu màn hình)	(Số hiệu ứng dụng)	Không gian chung cho hệ thống	(Số hiệu kết thúc)
Tiêu đề			
AAAA	BBBB	C	DDDD EEEE
Không gian vào/ra cho ứng dụng			
XX mã []			
XX tên []			
Thông báo hướng dẫn vận hành			
Thông báo lỗi (thông báo của ứng dụng)			(Ngày tháng và thời gian)

- Trong tiến trình thiết kế, năng suất của cả người phát triển và người dùng có thể được tăng lên nếu có sử dụng các hình mẫu thống nhất cho luồng màn hình, cho xử lý và cho những điều như đăng kí, tìm kiếm, cập nhật, xoá và những việc khác.

Hình 1-2-30

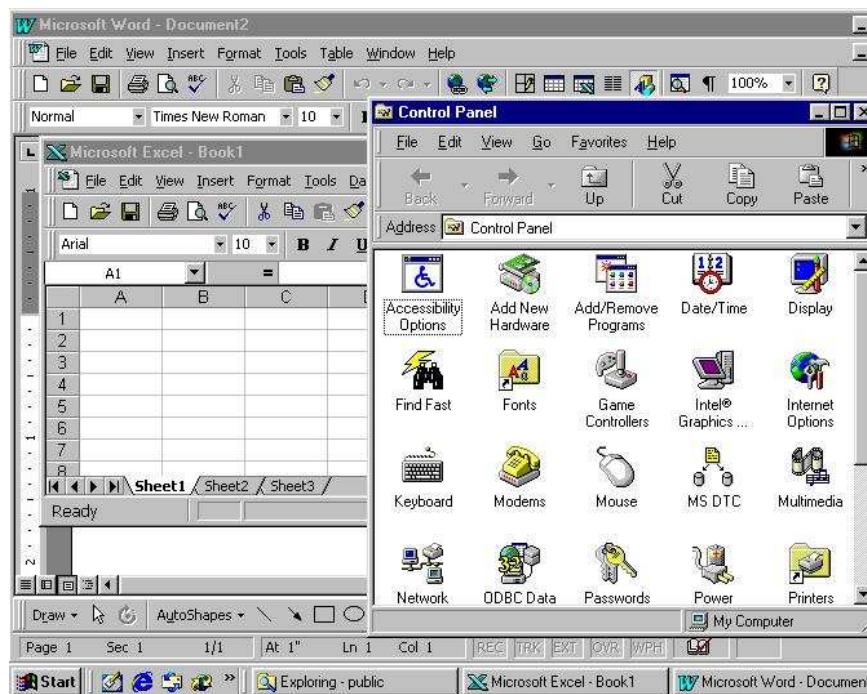
Ví dụ về luồng màn hình



-
-
- d. Nghiên cứu cách hiển thị trên màn hình
- Cách hiển thị các khoản mục trên màn hình được xác định bởi việc xem xét tần suất các màn hình được dùng và mức độ kinh nghiệm của người dùng. Chẳng hạn, nếu người mới tập sự vận hành các màn hình, thì việc dùng phương pháp đưa vào lệnh (đưa lệnh trực tiếp vào trên màn hình) sẽ không làm tăng năng suất. Thiết kế dựa trên quan điểm của người dùng là bản chất trong mọi trường hợp.
- Gần đây, nhiều hệ thống đa cửa sổ, trong đó nhiều màn hình được hiển thị đồng thời tạo ra khả năng xử lý song song cũng đã được sử dụng. Kiểu hệ thống này dễ dùng. Tuy nhiên, vì nó vận hành với thiết bị trở nên cần huấn luyện cho người dùng sử dụng thiết bị trở trong khi sử dụng hệ thống.

Hình 1-2-31

Ví dụ về đa cửa sổ



-
- e. Thiết kế bố trí màn hình
- Bố trí màn hình được thiết kế trong pha này. Hình 1-2-32 chỉ ra một ví dụ như vậy. Trong một số trường hợp, các ảnh màn hình được kiểm tra trong thiết kế ngoài, còn thiết kế thực tế được tiến hành về sau trong thiết kế trong dùng các mẫu đã chuẩn bị đặc biệt.
- Ngày nay, mô hình bản mẫu được tạo ra bằng các công cụ GUI và ngôn ngữ trực quan cho việc đánh giá, và các cách bố trí được quyết định qua việc thu được sự chấp thuận của người dùng.

Hình 1-2-32

Ví dụ về bố trí màn hình

Bố trí màn hình		Ngày	/	/	Chuẩn bị	Được chấp thuận bởi	
Tên màn hình	Menu cập nhật tệp lương chính				Screen ID	G K S 1 0 0 2	

	1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	
1	GKS1002	***Menu cập nhật tệp lương***						TTT010	1
2									2
3									3
4									4
5									5
6		Cập nhật							6
7		Bổ sung							7
8		Xoá							8
9									9
10		Kết thúc							10
1									1
2									2
3									3
4									4
5									5
6		Đưa vào một số từ danh sách trên, rồi							6
7		nhấn phím thực hiện							7
8									8
9									9
20									20
1	PF10	Vào menu con xử lý lương chính							1
2									2
3									3
4									4
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	

Lưu ý:

- Đưa vào một số tương ứng với việc xử lý được yêu cầu. Rồi, màn hình sẽ được chuyển sang màn hình dành cho xử lý đặc biệt.

(7) Thiết kế báo cáo

Khi người sử dụng dùng hệ thống thì các thiết kế báo cáo cũng quan trọng như thiết kế màn hình. Thiết kế này phải dễ hiểu và dễ dùng.

Thiết kế báo cáo có thể được dùng để thiết kế tài liệu in ra, nhưng cũng nên nhớ rằng kết quả hiển thị trên màn hình cũng là các báo cáo.

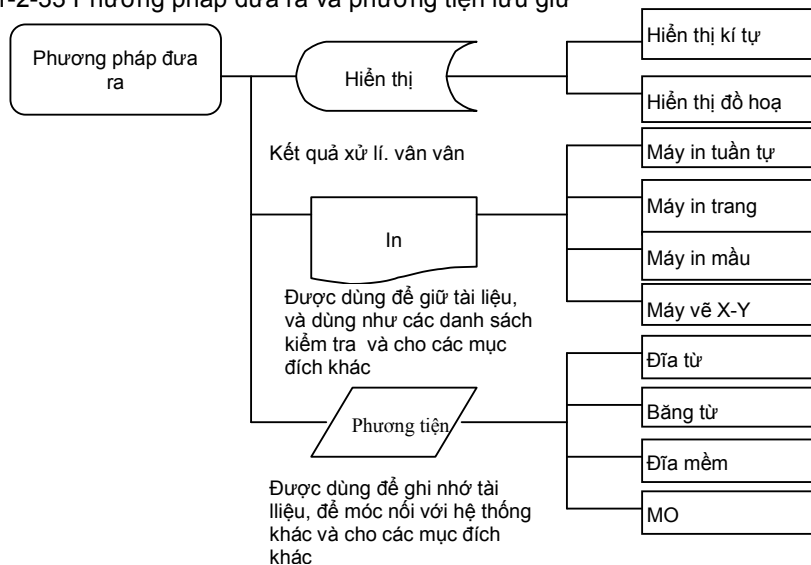
1. Thiết kế báo cáo là gì?

Báo cáo được dùng để trình bày kết quả xử lý của hệ thống theo một cách cụ thể và hiểu được. Do đó các định dạng và những điều khác phải được xem xét theo quan điểm của người dùng. Trong pha thiết kế ngoài, những hình ảnh đưa ra, như kích cỡ báo cáo, và cách bố trí các khoản mục trong báo cáo được thiết kế ra, trong khi thiết kế cho những điều đó lại phụ thuộc vào những ràng buộc phần cứng (như phương pháp in) và những điều chi tiết được thực hiện trong pha thiết kế trong.

2. Thủ tục thiết kế báo cáo và công việc cần thiết

- Các báo cáo được thiết kế theo các thủ tục sau:
- Khảo sát cái ra theo quan điểm tổng thể
- Xác định phương pháp đưa ra và phương tiện lưu giữ
- Tạo sơ đồ trình bày
- Chi tiết cho từng công việc này là như sau:
 - a. Khảo sát cái ra theo quan điểm tổng thể: Nhiều khoản mục, như mục đích và thời gian, cần cho việc tạo ra báo cáo, được nghiên cứu theo quan điểm tổng thể. Các khoản mục cho nghiên cứu bao gồm: tiêu đề báo cáo, mục đích sử dụng, chu trình sản xuất và thời gian, hạn chót, địa chỉ phân phối, số lượng, định dạng, và vị trí của các khoản mục được đưa ra.
 - Tiêu đề báo cáo
 - Cần cho tiêu đề thích hợp với nội dung báo cáo.
 - Mục đích
 - Cần phải xác định xem liệu mỗi báo cáo được dùng bên ngoài hay nội bộ. Nếu báo cáo để trình ra bên ngoài, thì có thể dùng các khuôn mẫu đã thiết kế sẵn. Cho nên, cần có sự chú ý cẩn thận.
 - Vòng đời và thời gian sản xuất

Hình 1-2-33 Phương pháp đưa ra và phương tiện lưu giữ



- Phải nghiên cứu khi nào thì cần tới báo cáo (hàng ngày, hàng tuần, hàng tháng, vào thời gian làm bản kê nghiệp vụ hay các tình huống khác).
- Hạn chót: Được nghiên cứu tới khi phải đưa ra các báo cáo.
- Địa chỉ phân phối: Phải xác định rõ ràng địa chỉ phân phối báo cáo.
- Số lượng: Số trang hay màn hình được ước lượng cho từng báo cáo.
- Định dạng: Xác định số các chữ số và kiểu của các khoản mục được in ra (hiển thị ra) trong báo cáo.
 - Vị trí của khoản mục ra: góc nhìn cái ra như vị trí của khoản mục, cần làm sáng tỏ.
- b. Xác định phương pháp đưa ra và phương tiện lưu giữ
- Phương pháp đưa ra thích hợp nhất và trung gian được xác định cho từng báo cáo bằng việc

xem xét tới bản chất của báo cáo và mục đích của chúng. Có nhiều trường hợp báo cáo được ghi lại trên các phương tiện lưu giữ như dữ liệu.

- c. Tạo ra bố trí cái ra
- Các đặc tả cái ra được soạn thảo dựa trên kết quả của các khoản mục a và b đã được mô tả ở trên, và cách bố trí cái ra (ảnh đưa ra) được tạo ra dựa trên các đặc tả (xem Hình 1-2-34 và 1-2-35). Các thiết kế dựa trên quan điểm người dùng cũng cần cho cách bố trí cái ra (ảnh đưa ra), chẳng hạn như khoảng cách giữa các kí tự, cách bố trí khoản mục, việc chèn thêm dữ liệu năm, tháng và ngày một cách tự động.

Hình 1-2-34

Ví dụ về đặc tả cái ra

Tên hệ con	Tên tài liệu	Ngày chuẩn bị
Xử lí tệp lượng chính	Đặc tả cái ra	

Tên thông tin	Tên xử lí
Danh sách cập nhật lượng chính	Cập nhật lượng chính

Mục đích		
Danh sách kiểm tra cho việc cập nhật lượng chính		
Chu trình và thời gian Khi nhu cầu này sinh	Mode đưa ra Theo lô	Địa chỉ phân phối
Thiết bị đưa ra Máy in trang	Phương tiện lưu trữ Giấy in (A4)	Hạn chót
Khối lượng thông tin		
Lưu ý		
- Dữ liệu đưa ra 7 người trên 1 trang (55 dòng/trang) Dữ liệu được in theo thứ tự số hiệu nhân viên		

NO	Tên mục	Kiểu kí tự	Số chữ số	Lưu ý
1	Mã phòng	N	4	
2	Mã nhân viên	N	5	
3	Mã chức vụ	N	2	
4	Tên	K	10	
5	Địa chỉ	K	40	
6	Ngày sinh	N	8	
7	Mã vợ chồng	N	1	
8	Số ăn theo	N	1	
9	Mã giới tính	N	1	
10	Lương cơ sở	N	8	
11	Phụ cấp dịch vụ	N	6	
12	Phụ cấp gia đình	N	6	
13	Phụ cấp nhà	N	6	
14	Phụ cấp đi lại	N	6	
15	Thuế dân cư	N	6	
16				
17				

(8) Thiết kế bộ mã

- Chúng ta thấy có một số bộ mã, như mã tỉnh thành, mã bưu điện, biển số xe cộ và số hiệu sản phẩm. Điều này là vì việc dùng đúng các bộ mã tạo điều kiện thuận lợi cho việc xử lí và quản lí thông tin trong máy tính.

- Tại đây chúng ta xem xét về các khoản mục mã hoá.

- ① Thiết kế bộ mã là gì?

- a. Mục đích của việc lập mã

- Thông tin mã hoá làm cho việc nhận diện, phân loại và thu xếp dữ liệu được dễ dàng hơn. Tuy nhiên, việc làm các bộ mã không đơn giản. Trong thiết kế bộ mã cần bao hàm cả tính mềm dẻo như khả năng mở rộng và sửa đổi nữa. Nếu dùng thiết kế bộ mã sai, thì tất cả các hệ thống (chương trình) có liên quan và dữ liệu sẽ phải bị sửa đổi. Do đó, cần có thái độ thận trọng trong thiết kế bộ mã.

Hình 1-2-35

Ví dụ bố trí cái ra (ảnh đưa ra)

Bảng tập lương chính				P. 999
<u>Mã nhân viên 9 ... 9</u>	<u>Tên N...N</u>	<u>Giới tính</u>		
<u>Địa chỉ N.....N</u>				
<u>Mã phòng ban</u>	<u>Tên phòng ban</u>			
<u>Mã chức vụ</u>	<u>Tên chức vụ</u>	<u>Bậc</u>	<u>Bậc con</u>	
<u>Lương cơ bản</u>	<u>Phụ cấp quản lí</u>	_____	_____	
_____	_____	_____	_____	
<u>Mã nhân viên 9 ... 9</u>	<u>Tên N...N</u>	<u>Giới tính</u>		
<u>Địa chỉ N.....N</u>				
<u>Mã phòng ban</u>	<u>Tên phòng ban</u>			
<u>Mã chức vụ</u>	<u>Tên chức vụ</u>	<u>Bậc</u>	<u>Bậc con</u>	
<u>Lương cơ bản</u>	<u>Phụ cấp quản lí</u>	_____	_____	

Các bộ mã vào/ra cho dữ liệu được dùng trong thiết kế ngoài được gọi đặc biệt là các bộ mã ngoài.

- b. Ý nghĩa của bộ mã
- Nói chung, bốn loại ý nghĩa sau cần phải được tính tới trong thiết kế mã:
 - Nghĩa nhận diện
 - Nghĩa để phân biệt dữ liệu này với dữ liệu khác (tính duy nhất).
 - Chẳng hạn, khách hàng có cùng họ và tên có thể được nhận diện tách biệt bằng việc cho từng người một số hiệu khách hàng khác nhau.
 - Nghĩa phân loại
 - Nghĩa để phân loại dữ liệu.
 - Chẳng hạn, dữ liệu có thể được thu xếp và phân loại một cách hệ thống, như theo cơ sở nhóm tuổi hay theo giới tính.
 - Nghĩa thu xếp
 - Nghĩa để xác định (thu xếp lại) trật tự dữ liệu.
 - Chẳng hạn, dữ liệu có thể được phân tích và thu xếp theo trật tự ngày sinh hay theo trật tự số chứng minh thư.
 - Nghĩa kiểm tra
 - Nghĩa để kiểm tra xem liệu các mã có được đưa vào đúng không.
 - Thông thường, một phép toán như lấy phần dư theo 11 được áp dụng cho từng mã gốc và dữ liệu kết quả (kiểm tra dữ liệu chữ số) được thêm vào cho chữ số thấp nhất của mã này.

-
-

② Những điểm cần xét tới trong thiết kế bộ mã

Thiết kế bộ mã sai có thể dẫn tới những vấn đề lớn ảnh hưởng tới toàn thể hệ thống. Do đó, thiết kế mã phải được làm một cách thận trọng bằng việc tính tới những mục sau:

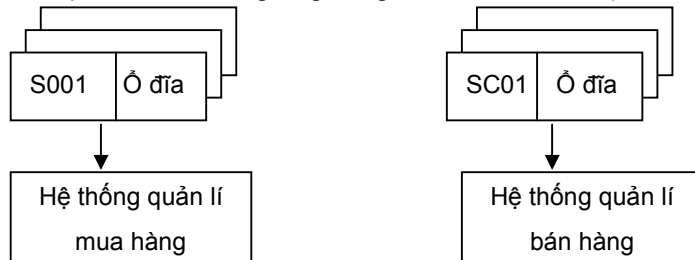
- a. Miền bao phủ miền và thời kì sử dụng bộ mã
- • Miền bao phủ

Hệ thống mã tùy thuộc vào miền nào mã được sử dụng. Chẳng hạn, khi cần được móc nối với hệ thống ngoài (được nối ghép với các hệ thống của các công ti khác qua mạng), thì cần phải sử dụng những hệ thống mã chuẩn trong công nghiệp. Ngay cả khi được dùng trong cùng tổ chức, thì điều mong muốn là thiết kế ra các bộ mã dùng được trong toàn bộ công ti.

-

Hình 1-2-36

Ví dụ về mã sản phẩm (ví dụ về hệ thống mua quản lí mua và hệ thống quản lí bán tương ứng dùng các mã khác nhau)



Nếu mã khác nhau được dùng cho cùng sản phẩm thì các thao tác không cần thiết tăng lên

- • Thời kì sử dụng
- Nếu các mã được thiết kế dựa trên những ước lượng sai về thời kì dùng bộ mã và/hoặc làm tăng thêm khối lượng công việc cần giải quyết trong tương lai, thì có thể gây ra việc thiếu hụt mã. Nếu những trường hợp như vậy xuất hiện, thì bộ mã phải được xem xét lại, làm cho toàn bộ hệ thống bị thay đổi. Việc thiết kế bộ mã yêu cầu đủ khối lượng mã.

Để giải quyết cho tình huống như vậy, có thể cung cấp mã phụ cho việc mở rộng ở cuối mỗi bộ mã.

-

Hình 1-2-37

Ví dụ thiết kế mã với khả năng mở rộng

X:X:X:9:9:9

X: Kí tự chữ

• Số hiệu Số Mã

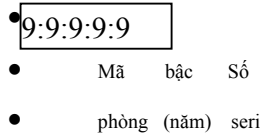
9: Kí tự số

• nhận diện se ri phụ

- b. Tính hiệu được
- Một điểm quan trọng cần được xét tới trong thiết kế bộ mã là tính hiệu được.
- - "Tính hiệu được" nghĩa là: dễ giải quyết (ngắn hơn và đơn giản hơn)
- - Có hệ thống (cho phép phân loại thành từng nhóm)
- - Rõ ràng (về cơ bản các kí tự số được dùng còn các kí tự chữ chỉ dùng khi cần).

Hình 1-2-38

Ví dụ hệ thống hoá về mã



-
- ③ Thủ tục thiết kế bộ mã và công việc cần thiết
- Các bộ mã được thiết kế theo những thủ tục sau:
 - 1. Lựa ra các khoản mục cần được mã hoá.
 - 2. Phân loại các mục đích mã hoá.
 - 3. Ước lượng thời kì sử dụng và khối lượng dữ liệu.
 - 4. Xác định miền sử dụng.
 - 5. Làm việc mã hoá và tạo ra bảng mã.
 - 6. Tạo ra các tệp bộ mã.
- Chi tiết cho từng công việc là như sau.
 - a. Lựa các khoản mục cần được mã hoá
 - Các khoản mục ứng cử viên cần được mã hoá được lựa ra qua việc xem xét và phân tích dữ liệu vào/ra và từ cách bố trí màn hình đã được tạo ra. Rồi các khoản mục cần được mã hoá sẽ được xác định theo cái nhìn về tiến trình nghiệp vụ và xử lý máy tính.
 - b. Phân loại các mục đích mã hoá
 - Hệ thống mã được dùng tùy thuộc vào mục đích mã hoá. Do đó, các mục đích mã hoá phải được làm sáng tỏ trước hết. Chẳng hạn, việc mã hoá có tham chiếu tới (hay tìm) dữ liệu hay phân loại một lượng dữ liệu lớn hay không?
 - c. Ước lượng thời kì sử dụng và khối lượng dữ liệu
 - Số dữ liệu cho từng khoản mục cần được mã hoá được ước lượng theo khối lượng hiện tại, và ước lượng tăng trong tương lai. Ước lượng này rất quan trọng. Nếu số dữ liệu tăng quá số ước lượng này trong cuộc đời hệ thống, thì việc sửa đổi toàn bộ hệ thống, như việc sửa các tệp và chương trình, trở thành cần thiết. Cũng cần có một thiết kế có tính tới thời kì sử dụng.
 - d. Xác định miền sử dụng (ứng dụng)
 - Miền sử dụng các bộ mã nên được xác định bằng việc xác định công việc mà chúng dùng và bằng việc xem xét liệu chúng có được dùng trong công việc khác hay không. Điều này là vì, để dùng trong một số miền, số chữ số phải được tăng lên. Như đã mô tả trên, tốt hơn cả là dùng các bộ mã chuẩn khi có ghép nối với hệ thống của các công ti khác (chẳng hạn qua ghép nối mạng).
 - e. Làm việc mã hoá và tạo ra bảng mã
 - Các bộ mã thực tế nên được thiết kế với tri thức về kiểu và đặc trưng của các bộ mã thường, dựa trên kết quả của các mục a tới d trên đây, và bằng việc xem xét số các chữ số (được xác định trong việc xem xét một số lớn dữ liệu) được dùng trong hệ thống mã và các chữ số kiểm tra. Kết quả là cần chuẩn bị các tài liệu thiết kế mã. Sau đó, các bảng mã được soạn ra để thực tế gán mã cho dữ liệu. Tài liệu gán các bảng này lại với nhau được gọi là "Sách bộ mã"
 - f. Tạo ra tệp bộ mã

- Dữ liệu trong bảng mã được lưu giữ trong thiết bị nhớ (như đĩa từ) để dùng trong công việc thực tế. Vậy tệp được tạo ra được gọi là tệp mã.
-
-

- ④ Ví dụ về các bộ mã hay được dùng
Hình 1-2-39 Ví dụ về hệ thống mã

Tên mã	Mô tả	Ví dụ ứng dụng
Mã tuần tự	Số hiệu được gán tuần tự. Mặc dầu công việc này được thực hiện đơn giản, nhưng các mã không thể được hệ thống hoá.	Mã tỉnh thành: • 04 - Hà Nội • 08- Thành phố Hồ Chí Minh
Mã theo khối	Số hiệu được gán cho từng khối. Rồi số hiệu được gán tuần tự trong khối. Việc dùng mã này thuận tiện cho phân loại, nhưng bất tiện cho việc thêm dữ liệu, hay khi số lượng khoản mục lớn.	Mã ngành nghề • Mã các trường đại học
Mã thập phân	Các đối tượng để mã hoá đầu tiên được mã từ 0 đến 9. Rồi các mã 0 đến 9 lại được cho từng số hiệu được mã này. Tiến trình này cứ tiếp tục cho việc mã hoá mịn hơn.	Ví dụ: • 000 Chung • 010 Thư viện • 020 Sách, tiểu sử • ... • 100 Triết học • 110 Chung về triết
Mã theo chữ số (mã phân lớp theo nhóm)	Mỗi chữ số của mã đều có một nghĩa đã gán cho. Việc dùng mã này là thuận tiện cho từng dữ liệu phân loại, nhưng cần tăng thêm nhiều con số để giải quyết cho số lớn dữ liệu.	<Mã phân loại các ngành công nghiệp ở JIS> • ① ② ③ ④ • 12 84 1 1 Công nghiệp phần mềm • 12 84 1 2 Công nghiệp dịch vụ xử lý thông tin • 12 84 1 3 Công nghiệp dịch vụ cung cấp thông tin • 12 84 1 9 Công nghiệp dịch vụ thông tin khác • ① Mã loại chính(công nghiệp dịch vụ) • ② Mã loại giữa(dịch vụ thông tin, các ngành công nghiệp quảng cáo và nghiên cứu) • ③ Mã loại nhỏ(công nghiệp dịch vụ thông tin) • ④ Mã phụ
Mã tượng trưng	Chữ viết tắt hay các kí hiệu sản phẩm được dùng như mã. Vậy chúng dễ nhớ, nhưng việc dùng chúng có thể bất tiện cho phân loại	18CTV (máy thu tivi 18 inches) • YY, MM, DD(năm, tháng và ngày)
Mã số kiểm tra	Chỉ ra các mã với chữ số kiểm tra cho bản thân mã này. Được dùng để mã cho những đối tượng có bao hàm nhiều lỗi không được phép xảy ra.	<Số hiệu tài khoản> • 1 2 6 5 3 • X) 1 4 2 3 Số kiểm tra

		<ul style="list-style-type: none"> 1 8 12 15 $(1+8+12+15) / 11 = 3$ Số dư 3
--	--	--

-
- ⑤ Phương pháp kiểm tra mã
- Phương pháp chữ số kiểm tra được dùng để tìm ra lỗi trong mã, nhưng không cung cấp khả năng sửa lỗi.
- Trong bộ nhớ máy tính, nói chung người ta dùng một mã sửa lỗi error correcting code (ECC), cung cấp cả việc phát hiện lỗi bit và khả năng sửa lỗi.
- Lỗi mã gần như được sinh ra khi dữ liệu mã được đưa vào. Chẳng hạn, lỗi có thể xuất hiện giữa 3 và 8 có hình dáng nhìn tương tự hay đưa vào nhầm trật tự, như đưa vào 123 thay vì 132.
- Hình 1-2-40 mô tả cách tính chữ số kiểm tra.

Hình 1-2-40

Cách tính chữ số kiểm tra

- - < Tính chữ số kiểm tra bằng phép modulus 10>
- - ① áp dụng tính trọng số (mã được giả thiết là "1011").
- - 1 0 1 1
- - 5 4 3 2 < Từng số nguyên chỉ ra trọng số được cho chữ số tương ứng.>
 - * Trọng số nhỏ nhất được cho chữ số thấp nhất.
- - ② Các phép nhân được thực hiện cho từng chữ số, và dữ liệu kết quả .
- - 1
 - 0
 - 1
 - 1
 - - ×
 - ×
 - ×
 - ×
 - - 5
 - 4
 - 3
 - 2
 - - 5+0+3+
 - 2= 10
- - * Nếu phép nhân cho kết quả hai chữ số, thì từng chữ số được tích lũy tách biệt.
- - ③ Chia (bằng việc dùng phép toán modulus)
 - $10 \div 10 = 1$ Số dư là 0, sẽ trở thành chữ số kiểm tra.
 - (M: modulus) (Dữ liệu được suy ra bằng việc trừ phần dư từ dữ liệu modulus có thể được dùng làm chữ số kiểm tra).

-
- ④ Đưa chữ số kiểm tra vào
- 1 0 1 1 0
- Chữ số kiểm tra

(9) Thiết kế giao diện con người

Trong thiết kế hệ thống người-máy một cách hợp lí, thì kĩ nghệ con người, như trước đây đã đặt con người làm hệ con của toàn thể hệ thống. Nói cách khác, hệ thống con người và hệ thống máy móc được giải quyết với trọng lượng ngang nhau. Tuy nhiên, gần đây người ta mới thấy rằng thiết kế mang trọng tâm con người, dựa trên bản chất linh động và mềm dẻo của con người, là cần thiết. Điều này có nghĩa là bây giờ người ta nhằm tới việc làm ra các giao diện người-máy của hệ thống tự nhiên nhất có thể được cho con người sử dụng. Hậu quả là thuật ngữ "giao diện con người" đã được dùng thay cho "giao diện người-máy."

Nói cụ thể, giao diện con người là một phần mà người dùng có tiếp xúc với hệ thống. GUI, trợ giúp cho việc vận hành bằng việc dùng các màn hình và cách bố trí tài liệu, cả hai đều đã được mô tả ở trên, đều là những ví dụ về giao diện con người. Nói riêng, trong các thao tác vào/ra qua màn hình máy tính cá nhân, điển hình như bởi Windows, các giao diện người dùng dùng biểu tượng và chuột đã được sử dụng rộng rãi. Do đó, hệ thống bây giờ phải được thiết kế bằng việc dùng các chức năng này một cách tương ứng.

- Hình 1-2-41 Các nhân tố giao diện con người điển hình

•	• Windows	
•	• Icons	• Menus
•	Nút bấm	Menu bật ra
•	Hộp kiểm	Menu kéo xuống
•	Nút radio	Menu thả xuống
•		Thanh công cụ

1.3 Ngôn ngữ lập trình

1.3.1 Thuộc tính chương trình

(1) Dùng lại được (reusable)

- Để thực hiện chương trình, thông thường chương trình phải được lưu lại vào bộ nhớ chính bằng bộ nạp (thao tác này được gọi là "nạp lại"). Chương trình, một khi đã được nạp vào, có thể được dùng lặp đi lặp lại mà không cần thao tác cất giữ lại, được gọi là "chương trình dùng lại được". Trong chương trình dùng lại được, việc khởi đầu và những thao tác khác có liên quan đều được thực hiện trước việc kích hoạt chương trình để duy trì tính toàn vẹn của xử lý. Do đó, chương trình này không thể được nhiều tiến trình sử dụng đồng thời.

(2) Đồng dụng (reentrant)

- Tương phản với chương trình dùng lại mà không cho phép việc dùng đồng thời, những chương trình có thể tạo ra kết quả đúng ngay cả khi được nhiều tiến trình sử dụng đồng thời thì được gọi là "chương trình đồng dụng."
- Trong các chương trình đồng dụng, phần cho các biến và phần cho các lệnh được đặt tách biệt nhau. Vậy phần cho biến được cung cấp cho từng tiến trình một cách tách biệt, trong khi phần cho lệnh lại được dùng chung bởi tiến trình liên quan tới xử lý đồng thời.

(3) đệ qui (recursive)

- Một chương trình có khả năng tự gọi (tự thực hiện) chính nó được gọi là "chương trình đệ qui." Việc có khả năng tự gọi đến chính mình không đảm bảo rằng các tiến trình khác có thể gọi nó. Do đó, chương trình đệ qui không phải bao giờ cũng là chương trình dùng lại được.

(4) Khả năng tái định vị (re-locatable)

- Chương trình định cho thực hiện được lưu giữ trong bộ nhớ chính bằng bộ nạp. Các chương trình có thể được lưu giữ tại bất kì vị trí nào trong không gian bộ nhớ chính để thực hiện được gọi là "chương trình khả tái định vị."
- Chương trình có vị trí trong bộ nhớ chính có thể được thay đổi ngay cả khi nó đang được thực hiện được gọi là "chương trình khả tái định vị động."

(5) Chèn lấp (Overlay)

- "Chèn lấp" là phương pháp được dùng để thực hiện một chương trình chiếm bộ nhớ lớn hơn dung lượng bộ nhớ chính. Theo phương pháp này, chương trình được chia thành một số đoạn, mỗi đoạn không được thực hiện đồng thời. Do đó, trong khi thực hiện, đoạn gốc nạp vào một trong những đoạn loại trừ lẫn nhau này từ thiết bị bộ nhớ phụ (xem Hình 1-3-1).

- Hình 1-3-1 Phương pháp chèn lấp

• 873	• Song Tùng	• 5	• Hải Hưng	• Đọc thơ cổ
• 245	• Chí Hiếu	• 1	• Hà Nội	• Du hành

•

(3) Kiểu dữ liệu trừu tượng

• Kiểu dữ liệu trừu tượng được định nghĩa cho các thực thể có chứa dữ liệu và các thao tác trên dữ liệu. Các thao tác là các hàm hay thủ tục. Trong việc dùng kiểu dữ liệu trừu tượng, chỉ cho phép dùng tham chiếu tới tên kiểu và việc gọi tới thao tác.

• Thao tác	• Dữ liệu
• Đẩy xuống	• Giá trị con trở ngăn xếp
• Bật ra	

•

1.3.3 Cấu trúc điều khiển

• Cấu trúc điều khiển của chương trình, hay của thuật toán, được thiết kế dựa trên định lý cấu trúc, bởi vì thiết kế dùng định lý này là có hiệu quả trong việc phát triển chương trình với các đặc trưng sau, làm năng suất và chất lượng phần mềm tăng lên.

- - Cấu trúc logic rõ ràng
- - Chính xác
- - Dễ đọc
- - Dễ duy trì
- "Định lý cấu trúc" được xác định như sau:

• "Với các chương trình (gọi là chương trình đúng) được thiết kế để bao gồm các đơn vị cấu trúc mà mỗi một trong chúng đều có một lối vào và lối ra, thì bất kì logic nào cũng có thể được mô tả bằng việc tổ hợp ba đơn vị cấu trúc cơ sở."

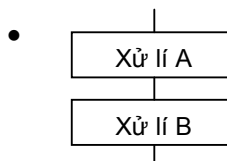
- Các chi tiết được mô tả dưới đây:

(1) Cấu trúc cơ sở

- ① Cấu trúc tuần tự (kiểu trình tự)

• Trong cấu trúc trình tự, các hàm (lệnh trong chương trình) được thực hiện tuần tự theo một chiều.

• Hình 1-3-2
Ví dụ về cấu trúc
tuần tự



•

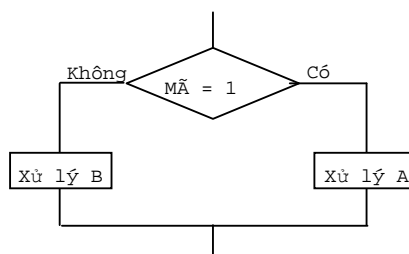
• ② Cấu trúc tuyến chọn (kiểu If-then-else)

• Trong cấu trúc tuyến chọn, một trong hai chức năng được chọn lựa tùy theo liệu điều kiện có được đáp ứng hay không.

•

• Hình 1-3-3

• Cấu trúc tuyến chọn



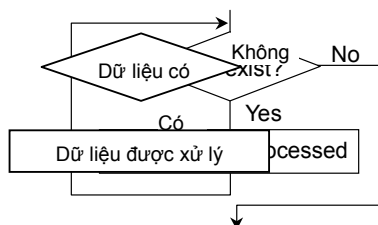
• ③ Cấu trúc lặp (kiểu Do-While)

• Trong cấu trúc lặp (Kiểu Do-While), nếu điều kiện là "đúng" thì cùng chức năng đó được thực hiện lặp lại. Nếu điều kiện là "sai" thì việc xử lý ra khỏi chu trình.

•

• Hình 1-3-4

• Cấu trúc lặp (kiểu Do-While)



• Nếu bất kì thuật toán nào cũng có thể được mô tả bằng ba cấu trúc trên, thì việc dùng câu lệnh GO TO, vẫn là nguyên nhân lớn nhất làm cho chương trình thành phức tạp, sẽ là không cần thiết. Điểm này sẽ được mô tả về sau.

(2) Các cấu trúc phụ

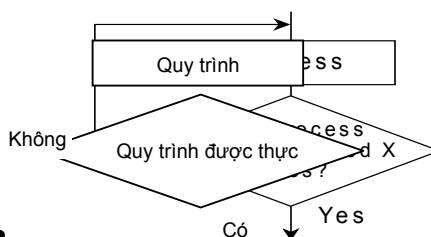
• Bên cạnh ba cấu trúc trên, hai cấu trúc sau đây cũng được dùng để làm cho chương trình dễ hiểu hơn.

• ① Kiểu Repeat-Until

• Trong kiểu Repeat-Until, một điều kiện được kiểm tra sau khi thực hiện một chức năng. Sau đó, nếu điều kiện này là "sai," thì việc thực hiện chức năng đó được lặp lại, còn nếu là "đúng" thì thao tác đi ra ngoài chu trình.

• Hình 1-3-5

• Cấu trúc lặp (kiểu Repeat-Until)



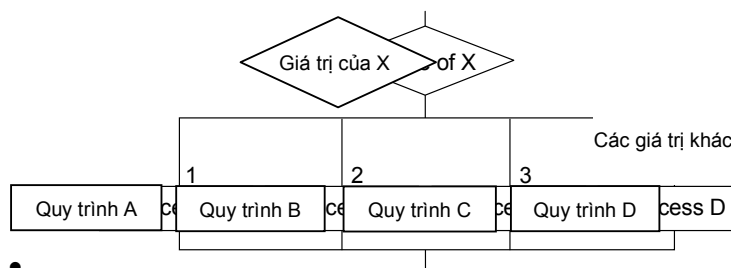
•

- ② Cấu trúc đa nhánh (CASE)

Cấu trúc đa nhánh được dùng khi một hay nhiều chức năng được thực hiện tùy theo kết quả của việc kiểm tra điều kiện.

- Hình 1-3-6

- Cấu trúc đa nhánh (CASE)



(3) Lập trình không GO TO

Việc dùng các câu lệnh GO TO là nhân tố lớn nhất trong việc làm cho chương trình thành phức tạp. Việc dùng câu lệnh GO TO dường như thuận tiện. Cho nên chúng thường được dùng một cách bất cần. Tuy nhiên, nếu các câu lệnh này bị lạm dụng, thì định lý cấu trúc không còn được giữ nữa.

Do đó, việc dùng các câu lệnh GO TO nên được tránh nếu có thể được (điều này không có nghĩa là việc dùng chúng là bị cấm), và những thuật toán nên được mô tả bằng việc tổ hợp các cấu trúc cơ sở.

(4) Thủ tục và hàm

Các đơn vị tạo nên chương trình là thủ tục và hàm. Có những thủ tục và hàm được hệ thống xử lý ngôn ngữ cung cấp, và cũng có những thủ tục và hàm do người dùng tạo ra. Cả hai đều được chương trình gọi để thực hiện. Với một thủ tục, kết quả đã xử lý được trả lại qua đối, trong khi với hàm thì kết quả được trả lại như giá trị của hàm.

1.3.4 Phân tích cú pháp

Các qui tắc cú pháp để mô tả chương trình được ngôn ngữ lập trình cung cấp. Dựa trên các qui tắc này, trình biên dịch phân tích các chương trình nguồn do người dùng tạo ra và thực hiện việc dịch. Việc phân tích cấu trúc của chương trình dựa trên các qui tắc cú pháp của ngôn ngữ được gọi là phân tích cú pháp. Ngôn ngữ chúng ta thường dùng trong trao đổi con người được gọi là ngôn ngữ tự nhiên, trong khi ngôn ngữ lập trình là ngôn ngữ nhân tạo.

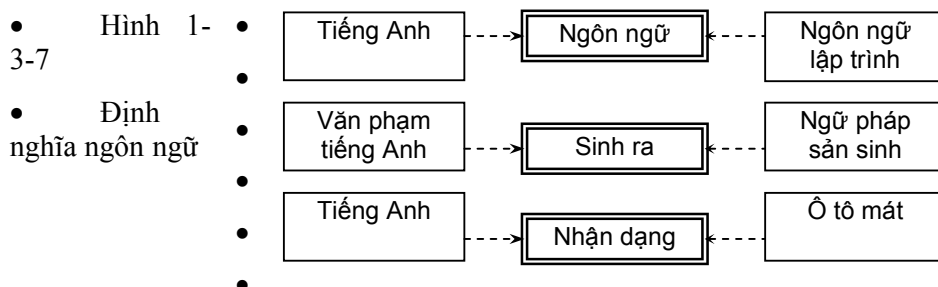
(1) Ngôn ngữ hình thức

Trong ngôn ngữ tự nhiên, từ và câu có một số nghĩa khác nhau. Bên cạnh đó, tồn tại một số mức độ tự do trong cấu trúc câu. Do đó, khó mà phân tích một cách máy móc các câu tự nhiên. Cũng khó cho máy tính làm việc diễn giải do phải xét tới các nhân tố gián tiếp như bầu không khí và ngữ cảnh hội thoại, cảm tính con người, môi trường bao quanh, v.v. Với máy tính ngày nay, vẫn còn khó xác định nghĩa. Ngôn ngữ hình thức được thiết kế để cho không có sự mơ hồ nào còn lại trong diễn giải.

- ① Định nghĩa ngôn ngữ hình thức

Ngôn ngữ hình thức được định nghĩa bằng văn phạm sinh và ô tô mát. Văn phạm sinh là tên

chung cho các văn phạm do Chomsky đề nghị. Nói một cách đơn giản, nó nghĩa là các qui tắc để sinh ra ngôn ngữ. Nói cách khác, ô tô mát có thể đoán nhận chỉ ngôn ngữ hình thức.



- a. Các kiểu và văn phạm của ngôn ngữ hình thức
- Chomsky đã đề nghị bốn ngôn ngữ hình thức sau. Ô tô mát tương ứng với mỗi một trong bốn ngôn ngữ này.

Hình 1-3-8

Kiểu ngôn ngữ	Ô tô mát đoán nhận
---------------	--------------------

Các kiểu ngôn ngữ hình thức

<ul style="list-style-type: none"> Văn phạm cấu trúc câu (ngôn ngữ kiểu 0-) Ngôn ngữ cảm ngữ cảnh (ngôn ngữ kiểu 1) Ngôn ngữ phi ngữ cảnh (ngôn ngữ kiểu 2) Văn phạm chính qui (ngôn ngữ kiểu 3) 	<ul style="list-style-type: none"> BNF Đơn giản nhất về mặt lý thuyết 	<ul style="list-style-type: none"> Máy Turing Ô tô mát hữu hạn tuyến tính Ô tô mát đầy xuống Ô tô mát hữu hạn
--	---	---

- b. Các thao tác trong ngôn ngữ
- Văn phạm cấu trúc câu được mô tả bằng bốn phần tử N, T, P và S. Chẳng hạn, số thực được sinh ra như sau.

N: non-terminal symbol = { <numeral>, <numeral sequence>, <integer>, <exponent>, <real number> }

T: Terminal symbol = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, E, . }

P: Generative rules =

{<numeral> → 0

<numeral> → 1

<numeral> → 2

⑦

<numeral> → 3

⑤

<numeral> → 4

③

<numeral> → 5

<numeral> → 6

- $\langle \text{numeral} \rangle \rightarrow 7$
- $\langle \text{numeral} \rangle \rightarrow 8$
- $\langle \text{numeral} \rangle \rightarrow 9$
- $\langle \text{numeral sequence} \rangle \rightarrow \langle \text{numeral} \rangle$ ④
- $\langle \text{numeral sequence} \rangle \rightarrow \langle \text{numeral sequence} \rangle \langle \text{numeral} \rangle$ ②
- $\langle \text{integer} \rangle \rightarrow \langle \text{numeral sequence} \rangle$ ⑥
- $\langle \text{exponent} \rangle \rightarrow \langle \text{integer} \rangle$
- $\langle \text{real number} \rangle \rightarrow \langle \text{integer} \rangle . \langle \text{numeral sequence} \rangle$ ①
- $\langle \text{real number} \rangle \rightarrow \langle \text{integer} \rangle . \langle \text{numeral sequence} \rangle E \langle \text{exponent} \rangle$
- $\langle \text{real number} \rangle \rightarrow \langle \text{integer} \rangle E \langle \text{exponent} \rangle$
- $\langle \text{numeral} \rangle \rightarrow 0$
- }

• S: start symbol = { $\langle \text{real number} \rangle$ }

- Ví dụ: Sinh ra số thực 123 $\langle \text{real number} \rangle \rightarrow \langle \text{integer} \rangle . \langle \text{numeral sequence} \rangle$; từ ①
- $\rightarrow \langle \text{integer} \rangle . \langle \text{numeral sequence} \rangle \langle \text{numeral} \rangle$; từ ②
- $\rightarrow \langle \text{integer} \rangle . \langle \text{numeral sequence} \rangle 3$; từ ③
- $\rightarrow \langle \text{integer} \rangle . \langle \text{numeral} \rangle 3$; từ ④
- $\rightarrow \langle \text{integer} \rangle . 2 3$; từ ⑤
- $\rightarrow \langle \text{numeral sequence} \rangle . 2 3$; từ ⑥
- $\rightarrow \langle \text{numeral} \rangle . 2 3$; từ ④
- $\rightarrow 1. 2 3$; từ ⑦

• Kí hiệu không kết thúc chỉ xuất hiện trong việc sinh (viết lại), trong khi kí hiệu kết thúc xuất hiện ở cuối.

(2) Biểu thức chính qui

• Biểu thức chính qui là cách mô tả cho các ngôn ngữ chính qui. Khi dùng chỉ dãy kí tự là không đủ, thì biểu thức chính qui có hiệu quả cho việc mô tả hình mẫu này một cách gọn gàng.

- N: non-terminal symbol = { $\langle N2 \rangle, \langle N3 \rangle, \langle N4 \rangle, \langle N5 \rangle, \langle \text{real number} \rangle$ }
- T: Terminal symbol = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, E, . }
- P: Generative rules = { $\langle N5 \rangle \rightarrow \{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \}$ ④
- $\langle N5 \rangle \rightarrow \langle N5 \rangle \{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \}$
- ; " | " chỉ phân cách dữ liệu phía phải
- $\langle N4 \rangle \rightarrow \langle N5 \rangle .$

- $\langle N4 \rangle \rightarrow \langle N4 \rangle \{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \}$
- $\langle N3 \rangle \rightarrow \langle N4 \rangle \{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \}$
- $\langle N2 \rangle \rightarrow \langle N5 \rangle$. ③
- $\langle N2 \rangle \rightarrow \langle N5 \rangle E$
- $\langle N2 \rangle \rightarrow \langle N3 \rangle E$
- $\langle N2 \rangle \rightarrow \langle N2 \rangle \{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \}$ ②
- $\langle \text{real number} \rangle \rightarrow \langle N2 \rangle \{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \}$ ①
- }
-
-
- S: kí hiệu bắt đầu = { <số thực> }
-
- Ví dụ: Sinh ra số thực 1.23 $\langle \text{real number} \rangle \rightarrow \langle N2 \rangle$ 3 ; từ ①
- $\rightarrow \langle N2 \rangle$ 23 ; từ ②
- $\rightarrow \langle N5 \rangle$. 23 ; từ ③
- $\rightarrow 1.23$; từ ④
-

(3) BNF (Dạng chuẩn Backus Naur)

- BNF, cũng còn được gọi là Dạng chuẩn Backus Naur, đưa ra một cách mô tả để xác định các định dạng của văn phạm phi ngữ cảnh. Số thực được mô tả như sau.
- $\langle \text{numeral} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$;
- Tại đây, " | " chỉ ra 'hoặc'.
- $\langle \text{numeral sequence} \rangle ::= \langle \text{numeral} \rangle | \langle \text{numeral sequence} \rangle \langle \text{numeral} \rangle$
- $\langle \text{integer} \rangle ::= \langle \text{numeral} \rangle$
- $\langle \text{exponent} \rangle ::= \langle \text{integer} \rangle$
- $\langle \text{real number} \rangle ::= \langle \text{integer} \rangle . \langle \text{numeral sequence} \rangle |$
- $\langle \text{integer} \rangle . \langle \text{numeral sequence} \rangle E \langle \text{exponent} \rangle |$
- $\langle \text{integer} \rangle E \langle \text{exponent} \rangle$
-

(4) Kí pháp Balan

- Kí pháp Ba lan, cách diễn tả các công thức, đặt toán tử lên trước toán hạng. Kí pháp này cũng còn được gọi là kí pháp viết đầu. Chẳng hạn, "1 + 2" được diễn tả là "+ 12." Mặt khác, kí pháp ("1 + 2") thường được dùng được gọi là kí pháp viết giữa. Trong trình biên dịch có dùng kí pháp viết sau ("12 +"), cũng còn được gọi là kí pháp Balan ngược (Hình 4.42). Với kí pháp này, toán tử được đặt sau toán hạng. Trong trình biên dịch, kí pháp Ba lan ngược được dùng nhiều nhất, bởi vì, với kí pháp này, các toán hạng và toán tử, tạo nên các cấu phần của một công thức, được đặt theo thứ tự của các phép toán, kết quả là làm cho các phép toán được hiệu quả. Do đó, kí pháp viết sau

đôi khi đơn giản được gọi là kí pháp Ba lan.

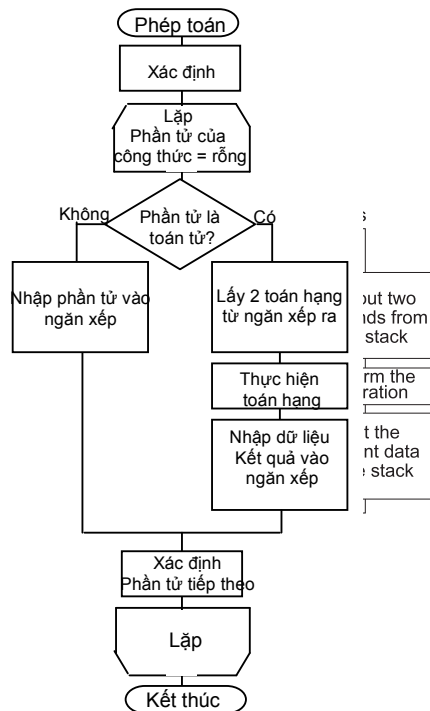
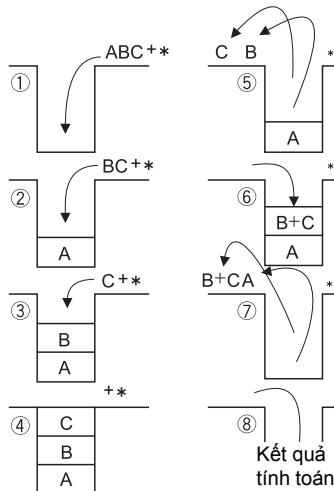
-
-
-
-
-

Hình 1-3-9

Ví dụ về các phép toán dùng kí pháp Ba lan ngược

- [Phương pháp phép toán]
- ① Đẩy nó vào ngăn xếp, nếu nó là toán hạng.
- ② Nếu nó là toán tử, thì tiến hành một thao tác đặc biệt cho hai toán hạng đã được đặt tại vị trí cao nhất của ngăn xếp, và cho lại kết quả vào ngăn xếp.

{ Một công thức ví dụ: $A * (B + C)$
 Với kí pháp Ba lan ngược: $ABC + *$ }



(5) Xử lí ngôn ngữ tự nhiên

Ngôn ngữ tự nhiên bao gồm nhiều cách diễn đạt nhập nhằng. Do đó, ngày nay chỉ một phần của nó mới được xử lí bằng máy tính. Việc xử lí ngôn ngữ tự nhiên phần lớn vẫn ở giai đoạn nghiên cứu. Trong xử lí ngôn ngữ tự nhiên, việc phân tích "nghĩa", "ý định" và "ngữ cảnh" cần làm thêm bên cạnh việc xử lí cú pháp, từ và câu. Cho nên, việc xử lí là phức tạp hơn ngôn ngữ hình thức. Nói riêng, việc dùng cơ sở tri thức trở thành điều bản chất để giải quyết sự nhập nhằng.

- 1. Xử lí hình vị : Xử lí để tìm các từ (hình vị) từ một câu đã cho, và xác định phần tiếng nói của chúng.
- 2. Xử lí cú pháp : Xử lí để áp dụng các qui tắc cú pháp và tìm ra cấu trúc cú pháp.
- 3. Xử lí ngữ nghĩa : Xử lí để tìm ra cấu trúc ngữ nghĩa của câu bằng việc dùng các từ điển và cơ sở tri thức.
- 4. Xử lí ngữ cảnh : Xử lí để diễn giải mối quan hệ giữa câu và phát âm như "nó," "cái đó"

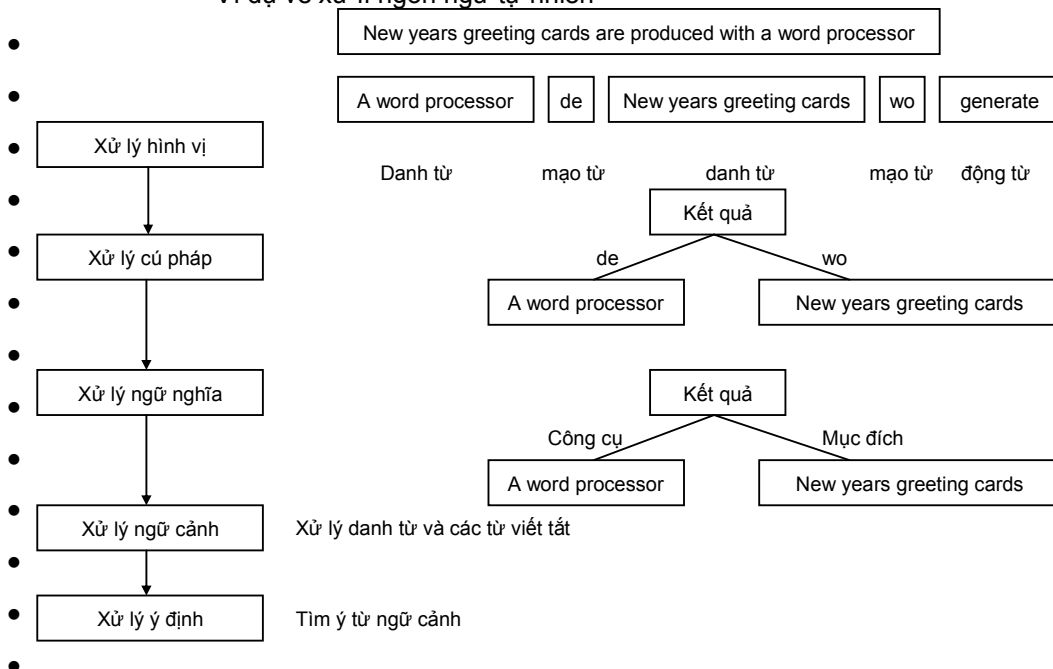
v.v. và phân tích các từ viết tắt và mối quan hệ nguyên nhân hậu quả giữa các câu.

- 5. Xử lý ý định : Xử lý để tìm ra ý định của câu bằng việc tính tới tình huống, thói quen và nền tảng văn hoá.

-
-
-
-

Hình 1-3-10

Ví dụ về xử lý ngôn ngữ tự nhiên

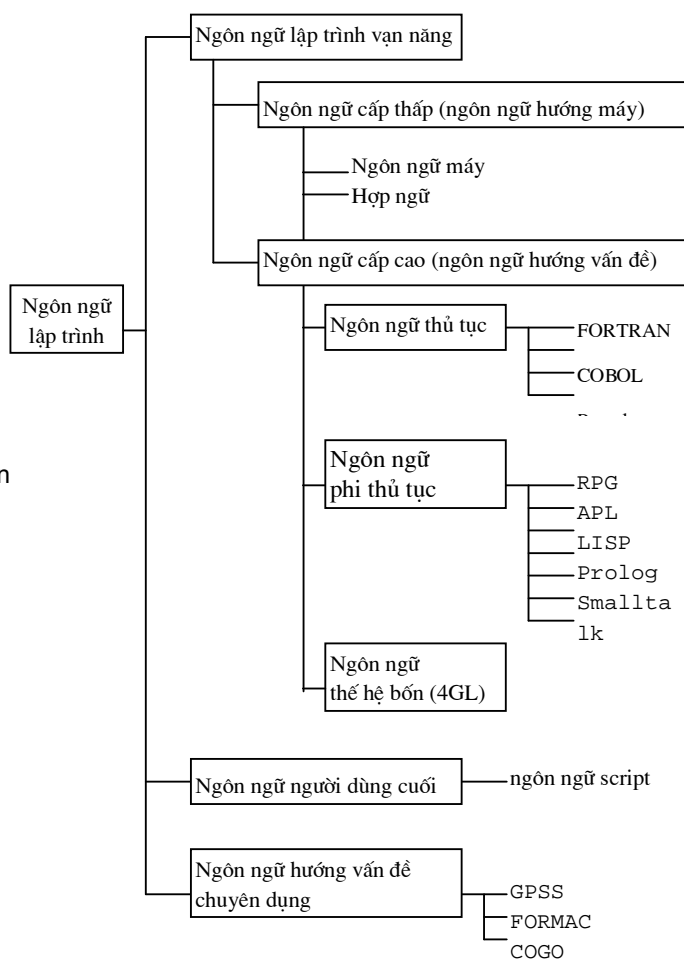


1.3.5 Phân loại về ngôn ngữ lập trình

- Hình 1-3-11 là một phân loại thô về các ngôn ngữ lập trình.

(1) Ngôn ngữ cấp thấp

- Các ngôn ngữ máy, và các ngôn ngữ lập trình có tương ứng một-một giữa các lệnh của chúng và lệnh của ngôn ngữ máy, được gọi là ngôn ngữ cấp thấp. Hợp ngữ thuộc vào ngôn ngữ cấp thấp. Các chương trình được viết trong hợp ngữ có các đặc trưng sau:
 - Cung cấp tốc độ thực hiện chương trình cao.
 - Cung cấp năng suất phần mềm thấp, nếu so với ngôn ngữ cấp cao, bởi vì nó dùng các kí hiệu chữ để mô tả chương trình.
 - Người lập trình phải có tri thức về phần cứng, bởi vì ngôn ngữ này có quan hệ chặt chẽ với các chức năng phần cứng.
 - Các chương trình phải được viết lại khi phần cứng bị thay đổi.
 - Với những đặc trưng như vậy, hợp ngữ chỉ được dùng trong xử lý điều khiển và các lĩnh vực khác nơi cần tới tốc độ xử lý cao. Ngôn ngữ cấp cao được dùng chủ yếu trong phát triển phần mềm.



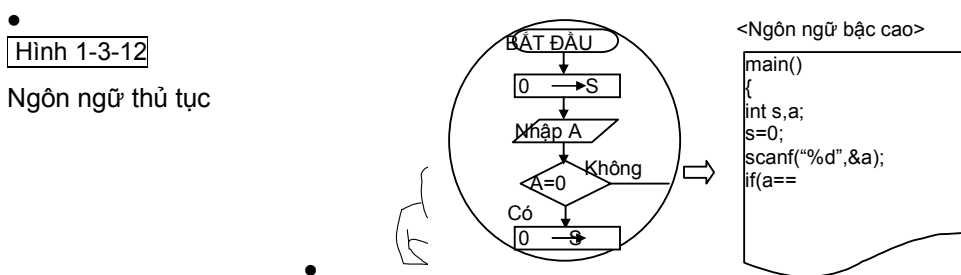
(2) Ngôn ngữ cấp cao

- Ngôn ngữ cấp cao đã được phát triển để giải quyết các vấn đề mà ngôn ngữ cấp thấp đặt ra. Nó cũng còn được gọi là ngôn ngữ hướng vấn đề.
- Ngôn ngữ cấp cao có các đặc trưng sau:
- Nó cung cấp sự dễ dàng trong việc mô tả các thủ tục xử lý và thích hợp cho việc giải quyết vấn đề.
- So với hợp ngữ, nó có cấu trúc gần ngôn ngữ tự nhiên hơn.
- Nó ít phụ thuộc vào phần cứng đặc biệt (thích hợp cho việc dùng trong nhiều ứng dụng).
- Một lệnh có thể chứa nhiều lệnh máy. Cho nên, số các bước chương trình (số các lệnh) có thể ngắn hơn.

- Dùng ngôn ngữ lập trình cấp cao làm giảm thời gian tạo chương trình, làm cho việc sửa chương trình và thêm chức năng dễ dàng hơn, kết quả là dẫn tới làm tăng năng suất phát triển phần mềm. Tuy nhiên, ngôn ngữ cấp cao cần phần mềm để vận hành trực tiếp các chức năng phần cứng, đôi khi đưa tới việc làm giảm hiệu quả xử lý.
- Các ngôn ngữ cấp cao vận năng hiện đang được dùng rộng rãi sẽ được mô tả chi tiết trong mục sau.

(3) Ngôn ngữ thủ tục

- Để mô tả các thuật toán chỉ ra các thủ tục xử lý giải quyết một vấn đề, thì cần phải sử dụng tới các lệnh có dùng ngôn ngữ thủ tục. Nhiều ngôn ngữ cấp cao là ngôn ngữ thủ tục.



(4) Ngôn ngữ phi thủ tục

- Ngôn ngữ phi thủ tục cho phép chương trình được sinh ra mà không phải mô tả thuật toán. Với ngôn ngữ này, chương trình được sinh ra bằng việc cung cấp cái vào, cái ra và điều kiện xử lý để giải quyết vấn đề, và bằng việc chọn các trình xử lý cần thiết do ngôn ngữ này cung cấp trước.
- Ngôn ngữ phi thủ tục có các đặc trưng sau:
 - Không dùng thuật toán nào, ngôn ngữ vẫn cho phép những người với kỹ thuật lập trình yếu tạo ra được chương trình.
 - Nó thường có năng suất cao do việc áp dụng các hình mẫu đã định trước vào thủ tục xử lý, nhưng lại kém linh hoạt trong việc sinh ra chương trình.
 - Ngôn ngữ phi thủ tục được phân loại thành các ngôn ngữ phụ thuộc vào định dạng mô tả và vào các nhân tố khác:
 - **Ngôn ngữ dạng biểu bảng**
 - Việc viết các yêu cầu xử lý theo định dạng đã xác định trước bằng ngôn ngữ dạng biểu bảng sẽ tự động sinh ra chương trình tương ứng.
 - **Ngôn ngữ hàm**
 - Với ngôn ngữ hàm, việc tổ hợp các hàm sinh ra chương trình. LISP cho xử lý danh sách và APL cho tính toán khoa học và kỹ nghệ là các ngôn ngữ hàm.
 - **Ngôn ngữ logic**
 - Với ngôn ngữ logic, chương trình được sinh ra từ các biểu thức logic dựa trên logic mệnh đề. Một ví dụ về kiểu ngôn ngữ này là Prolog, một ngôn ngữ lập trình cho trí tuệ nhân tạo.
 - **Ngôn ngữ hướng đối tượng**
 - Với một ngôn ngữ hướng đối tượng, chương trình được tạo ra bằng việc xem xét mọi thứ

được chương trình xử lý như các đối tượng. Các ngôn ngữ hướng đối tượng điển hình bao gồm Smalltalk, C++, và Java thích hợp cho việc dùng mạng.

-

(5) Ngôn ngữ thể hệ thứ tư (4 GL)

- Người dùng thường phàn nàn, "Chức năng được yêu cầu thì lại không cung cấp", khi việc phát triển phần mềm cho việc tổng hợp các chức năng đến cuối cùng mới hoàn thành.
- Những lí do chính đằng sau tình huống này bao gồm:
- Người phát triển hệ thống không hiểu kĩ những thao tác cần được tổng hợp lại.
- Khách hàng không cung cấp đúng nội dung của các thao tác cần được hệ thống hoá cho việc sáng tạo của người phát triển hệ thống.
- Việc tạo ra chương trình do những người trong phòng ban của người dùng thực hiện, người đã đầu tư vào những thao tác này, là một giải pháp cho vấn đề này. Tuy nhiên, việc giải quyết ngôn ngữ cấp cao để định ra chương trình đòi hỏi một khối lượng đào tạo kĩ thuật đáng kể.
- Để giải quyết tình huống như vậy, người ta đã tạo ra một ngôn ngữ lập trình, cho phép mọi người không có tri thức và kinh nghiệm về chương trình tạo ra được chương trình một cách dễ dàng. Ngôn ngữ này được gọi là ngôn ngữ thể hệ thứ tư.
- Ngôn ngữ thể hệ thứ tư là thể hệ kế tiếp của ngôn ngữ cấp cao (ngôn ngữ thể hệ ba). Với ngôn ngữ cấp cao trước đây, thủ tục để giải quyết vấn đề trong các thao tác ứng dụng được mô tả bằng các lệnh, trong khi đó, với ngôn ngữ thể hệ thứ tư, việc xử lý các thao tác được thực hiện bằng các hướng dẫn có tham biến. Nói cách khác, ngôn ngữ mới này dự tính sinh ra các thủ tục xử lý để giải quyết vấn đề một cách tự động nhiều nhất có thể được, làm giảm số các hướng dẫn sinh ra chương trình.

(6) Ngôn ngữ script

- Với việc sử dụng rộng rãi máy tính, xu hướng chuyển sang "tính toán người dùng cuối" mà trong đó phòng ban người dùng tự mình phát triển hệ thống đã phát triển mạnh. Ngôn ngữ script là ngôn ngữ lập trình do người dùng cuối sử dụng để phát triển những chương trình như vậy.
- Bản chất hướng biến cố là đặc trưng quan trọng nhất của ngôn ngữ script. Chương trình đợi cho dữ liệu được người dùng đưa vào từ bàn phím hay chuột. Khi phát hiện ra cái vào, chương trình bắt đầu xử lý.
- Ngày nay, các môi trường phát triển nâng cao có dùng GUI đã được cung cấp cho ngôn ngữ này. Nó cũng còn được dùng như công cụ để chuyên biệt hoá phần mềm ứng dụng. HTML (Hyper Text Markup Language), ngôn ngữ soạn thảo văn bản để tạo ra các tệp hiển thị được trên trình duyệt WWW (World Wide Web - Mạng toàn cầu) trên Internet, đã được sử dụng rộng rãi.

(7) Ngôn ngữ hướng vấn đề chuyên dụng

- Ngày nay, chúng ta phát triển các chương trình máy tính bằng các ngôn ngữ lập trình, và dùng máy tính để giải quyết vấn đề trong nhiều miền đa dạng. Tuy nhiên, những xử lý trong một số miền lại phức tạp và cao cấp đến mức phải mất vài giờ tới vài chục giờ để hoàn thành những tính toán cần thiết. Ngôn ngữ hướng vấn đề chuyên dụng đã được phát triển nhằm vào việc chỉ xử lý cho các thao tác được dùng trong những miền đặc biệt này.
- Ngôn ngữ hướng vấn đề chuyên dụng có các đặc trưng sau:

- Vì các chương trình được dùng trong miền đặc biệt là có giới hạn, nên nó cho phép những chương trình cần thiết được phát triển nhanh hơn và chính xác hơn trong những miền có giới hạn đó, nếu so với việc dùng các ngôn ngữ vạn năng.
- Thông thường, người làm phần mềm cung cấp các ngôn ngữ như vậy theo kiểu gói phần mềm. Do đó, việc dùng chúng là một ưu thế cho người dùng về độ tin cậy và chi phí phát triển.
- Tri thức đặc biệt trong các miền mà ngôn ngữ này được dùng là cần thiết.
- Ngôn ngữ này không được dùng cho những miền bên ngoài những miền mà nó đã được phát triển.
- Sau đây là các ngôn ngữ chính trong phân loại các ngôn ngữ hướng vấn đề chuyên dụng:
- GPSS (General Purpose Simulation System - Hệ thống mô phỏng vạn năng)
- Ngôn ngữ được dùng cho việc mô phỏng một hiện tượng trên máy tính được gọi là ngôn ngữ mô phỏng. GPSS là ngôn ngữ mô phỏng do IBM phát triển và được sử dụng rộng rãi cho các mục đích mô phỏng rời rạc.
- FORMAC (FORMula MANipulation Compiler - Trình biên dịch thao tác công thức)
- Đây là ngôn ngữ lập trình để thực hiện việc thao tác các công thức phức tạp và cao cấp chủ yếu dùng trong thủy động lực và thiên văn. Trong ngôn ngữ này, các hàm thao tác công thức cấp cao, như giải quyết các phép toán vi phân và biến đổi công thức, được bổ sung thêm vào FORTRAN, một ngôn ngữ lập trình cho tính toán khoa học và kỹ thuật.
- COGO (COordinate GeOmetry - Hình học phối hợp)
- Đây là ngôn ngữ lập trình để phân tích và thiết kế cấu trúc các toà nhà.

1.3.6 Kiểu và đặc trưng của ngôn ngữ lập trình

- Các ngôn ngữ lập trình cấp cao khác nhau đã được phát triển. Số lượng chúng đã vượt quá vài trăm ngôn ngữ. Việc nâng cấp cho người lập trình đối với ngôn ngữ cấp cao còn dễ hơn so với ngôn ngữ cấp thấp. Do đó, ngôn ngữ cấp cao được dùng trong nhiều miền.

(1) FORTRAN (Formula Translator - Bộ biến đổi công thức)

- FORTRAN được John Backus (người Mỹ) và những người khác thiết kế năm 1954. Sau đó nó thực tế đã được IBM phát triển, dựa trên tài liệu thiết kế đó, và những công ti khác cùng IBM đã đóng vai trò chính cho sự phát triển. FORTRAN đã được công bố là trình biên dịch đầu tiên trên thế giới vào năm 1957. Trong những ngày mà FORTRAN được phát triển, máy tính hầu hết được dùng cho các tính toán khoa học và kỹ nghệ, nhằm xử lý các tính toán phức tạp với tốc độ cao. Do đó, FORTRAN đã thiết lập ra vị trí của nó như một ngôn ngữ lập trình cho các tính toán khoa học và kỹ nghệ.
- <Đặc trưng>
- Nó cho phép diễn đạt các bài toán và thuật toán số cũng như các thao tác logic, cả hai đều cần cho tính toán khoa học kỹ nghệ.
- Nhóm các hàm, như hàm lượng giác, hàm lũy thừa và hàm lô ga rit, cũng được cung cấp theo ngôn ngữ.
- Cấu trúc của ngôn ngữ là đơn giản.
- Nó sử dụng cấu trúc tĩnh và không đệ qui.

Hình 1-3-13

- Chương trình ví dụ về
- FORTRAN

```

C CHUONG TRINH
MAU
•   INTEGER I, J, K
•   10 CONTINUE
•   READ(5,100) I, J, K
•   100 FORMAT(3I5)
•   IF (I.EQ.0) GO TO
20
•   L = I * J - K
•   200  FORMAT(1H,
18)
•   GO TO 10
•   20 STOP
•   END

```

(2) COBOL (COmmon Business Oriented Language - Ngôn ngữ hướng nghiệp vụ chung)

- Việc phát triển của COBOL đã được bắt đầu vào năm 1959 như một trình biên dịch cho việc xử lý nghiệp vụ bởi CODASYL (COncference on DAta SYstems Language - Hội thảo về ngôn ngữ hệ thống dữ liệu), một tổ chức các nhà chế tạo máy tính và người dùng máy tính, cùng với Bộ quốc phòng Mỹ làm trung tâm nỗ lực nghiên cứu và phát triển.
- Bản đặc tả đã được công bố vào năm 1960, và đặc tả ANSI (American National Standards Institute - Viện chuẩn quốc gia Mỹ) đã được thiết lập năm 1968. Tại Nhật, JIS COBOL đã được thiết lập năm 1972. Nó đã được sử dụng rộng rãi như trình biên dịch cho việc xử lý nghiệp vụ có dùng các máy tính vạn năng.
- <Đặc trưng>
- Việc dùng cách diễn đạt gần với câu tiếng Anh đưa ra tính dễ đọc tài liệu.
- Nó cho phép các cấu trúc tệp được dùng cho việc đưa vào/đưa ra được định nghĩa chặt chẽ.
- Hỗ trợ cho các phương pháp tổ chức tệp đa dạng tạo khả năng các thao tác vào/ra tệp hiệu quả.
- Nhiều chức năng xử lý nghiệp vụ, như phân lớp và gộp và chức năng sinh báo cáo cũng được cung cấp.

- Hình 1-3-14
- Chương trình ví dụ về COBOL
- IDENTIFICATION DIVISION.
- PROGRAM-ID. SAMPLE.
- ENVIRONMENT DIVISION.
- DATA DIVISION.
- WORKING-STORAGE SECTION.
- 01 INPUT-AREA.
- 03 I PIC 9(3).
- 03 FILLER PIC X(1).
- 03 J PIC 9(3).
- 03 FILLER PIC X(1).
- 03 K PIC 9(3).
- 03 FILLER PIC X(69).
- 01 INSATSU.
- 03 L PIC9.
- PROCEDURE DIVISION
- BATDAU.
- ...

(3) BASIC (Beginner's All purpose Symbolic Instruction Code - Mã lệnh kí hiệu vạn năng cho người mới bắt đầu)

- BASIC được phát triển năm 1965 như một ngôn ngữ lập trình để dạy học sinh tại trường đại học Dartmouth College của Mỹ. Khi máy tính cá nhân được dùng rộng rãi thì BASIC đã được chấp nhận là ngôn ngữ cấp cao cho phép dễ sử dụng.

• <Đặc trưng>

Nó sử dụng phương pháp thông dịch theo đó việc dịch chương trình và thực hiện chương trình được tiến hành đồng thời.

Việc dùng phương pháp thông dịch làm cho tốc độ thực hiện chậm hơn nếu so với việc dùng ngôn ngữ biên dịch.

Số dòng được đặt vào trước từng dòng lệnh.

Việc dùng các mô tả gần với từ tiếng Anh làm cho việc học ngôn ngữ được dễ dàng.

Các giá trị mặc định được tự động thêm vào, ngay cả khi các mô tả cho khai báo kiểu biến hay định dạng cái ra bị bỏ sót.

- Hình 1-3-15
- Chương trình ví dụ về
- việc dùng BASIC

```

• 10 INPUT I, J, K
• 20 IF I = 0
  THEN END
• 30 L = I * J - K
• 40 LPRINT L
• 50 GO TO 10

```

(4) Ngôn ngữ lập trình C

• Ngôn ngữ lập trình C là ngôn ngữ biên dịch cho mô tả hệ thống được W. Kernighan và D. Ritchie tại Bell Laboratories của AT&T phát triển năm 1969 như ngôn ngữ cho việc phát triển hệ điều hành có tên UNIX.

• Ban đầu, ngôn ngữ này đã được dùng riêng cho UNIX trên máy PDP-11, máy tính mini của DEC. Tuy nhiên bởi tính ích lợi của nó nên ngôn ngữ C ngày nay được dùng rộng rãi trên nhiều máy tính kể cả trạm làm việc và máy tính cá nhân. Ngôn ngữ C dựa trên tài liệu, "Ngôn ngữ lập trình C," do Kernighan và Ritchie viết, được gọi là ngôn ngữ C K&R. Mặt khác, còn có ANSI, đã thiết lập năm 1989 một đặc tả chuẩn của ngôn ngữ này, bao hàm những chức năng cải tiến và mở rộng. Ngôn ngữ C tuân thủ với đặc tả đó được gọi là "ANSI-C."

- <Đặc trưng>
- Nó cho phép các biểu thức đơn giản.
- Nhiều toán tử, cấu trúc dữ liệu và cấu trúc điều khiển được cung cấp.
- Nó cho phép các thao tác bit gần với các thao tác bit của ngôn ngữ cấp thấp.
- Vài ràng buộc bị áp đặt lên định dạng chương trình hay phương pháp phi định dạng được sử dụng.
- Cung cấp tính khả chuyển cao.
- Ngôn ngữ C++, con cháu của C có chứa các chức năng hướng đối tượng, đang được sử dụng rộng rãi.

- Hình 1-3-16
- Chương trình thí dụ về
- ngôn ngữ C

```

• #include <stdio.h>
• main()
• {
•   int x, y;
•   printf("Số đưa vào = ");
•   scanf("%d", &x);
•   y = x % 2;
•   if ( y == 0) printf("Số chẵn\n");
•   else       printf("Số lẻ\n");
• }

```

(5) Các ngôn ngữ cấp cao khác

Bên cạnh FORTRAN, COBOL, BASIC và ngôn ngữ C, các ngôn ngữ lập trình khác cũng đã được phát triển với nhiều mục đích. Các đặc trưng của những ngôn ngữ lập trình chính được mô tả dưới đây:

- ① **ALGOL** (ALGO^rithmic language - ngôn ngữ thuật toán)
 - ALGOL là ngôn ngữ lập trình được phát triển cho các tính toán khoa học và kỹ nghệ.
 - <Đặc trưng>
 - Chương trình được mô tả theo các qui tắc cú pháp được gọi là kí pháp Backus.
 - Cho phép gọi đệ qui.
 - Ngôn ngữ này được thiết kế dựa trên khái niệm về lập trình có cấu trúc.
 - Mặc dầu ngôn ngữ này có những đặc trưng cao cấp như mô tả ở trên, ALGOL lại khó được dùng bởi vì FORTRAN cũng nhằm vào cùng mục đích đó.
- ② **Pascal**
 - Pascal là ngôn ngữ lập trình cho giảng dạy lập trình và nó bao gồm các đặc trưng của ALGOL.
 - <Đặc trưng>
 - Nó cung cấp các cấu trúc điều khiển dựa trên lập trình có cấu trúc.
 - Bản thân chương trình là dễ đọc, bởi vì ngôn ngữ này đã được phát triển cho việc dạy lập trình.
 - Cung cấp nhiều cấu trúc dữ liệu.
- ③ **Ada**
 - Ada, ngôn ngữ lập trình do Bộ quốc phòng Mỹ phát triển, có các đặc trưng sau:
 - <Đặc trưng>
 - Ngôn ngữ này dựa trên Pascal với nhiều cải tiến được thêm vào nó.
 - Độ tin cậy cao cung cấp cho việc phát triển các phần mềm cỡ lớn.
 - Cung cấp khả năng dễ bảo trì và thực hiện có hiệu quả.
- ④ **LISP** (LISt Processor)
 - LISP, ngôn ngữ lập trình có các đặc trưng sau:
 - <Đặc trưng>
 - Nó là ngôn ngữ lập trình phi thủ tục.
 - Cả dữ liệu và chương trình đều có cấu trúc danh sách.
 - Chương trình được mô tả bằng việc tổ hợp các hàm chuẩn do hệ thống cung cấp và những hàm do người dùng định nghĩa.
- ⑤ **Prolog** (Program^ming in logic - lập trình theo logic)
 - Prolog là ngôn ngữ cho nghiên cứu và phát triển về trí tuệ nhân tạo.

- <Đặc trưng>
- Nó là ngôn ngữ logic mệnh đề.
- Cung cấp các hàm suy diễn.
- ⑥ APL (A Programming Language)
- APL, do IBM phát triển, có các đặc trưng sau:
- <Đặc trưng>
- Nó cung cấp kí pháp dựa trên kí pháp toán học thông thường.
- Các phép toán số học và logic có thể được mở rộng cho các vectơ, ma trận và cấu trúc cây.
- Chương trình có thể được mô tả bằng việc tổ hợp các kí tự đặc biệt và các kí hiệu.
- ⑦ PL/I (Programming Language/One)
- PL/I, được IBM và tổ chức người dùng của nó cùng phát triển, dành cho cả tính toán khoa học và kĩ nghệ lẫn xử lí nghiệp vụ.
- <Đặc trưng>
- Nó dùng cấu trúc lồng nhau với các khối được dùng làm đơn vị chương trình.
- Nó cho phép các phép toán bit và các phép toán danh sách cho dữ liệu.

(6) Ngôn ngữ hướng đối tượng và các ngôn ngữ lập trình khác

- ① Visual Basic
- Visual Basic, một ngôn ngữ trực quan điển hình, là BASIC nhưng có thể chạy trong môi trường Windows. Giao diện người dùng là quan trọng cho chương trình tương tác. Visual Basic cung cấp một môi trường để phát triển các ứng dụng cung cấp cách tương tác thích hợp cho môi trường đồ họa Windows.
- <Đặc trưng>
- Visual Basic, một sản phẩm của Microsoft, có liên hệ chặt chẽ với Windows.
- Các chương trình được sinh ra cùng Visual Basic có thể được chạy trực tiếp trong Windows.
- ② C++
- C++, do B. Straustrap và những người khác tại Bell Laboratories phát triển, đã được tạo ra bằng việc bổ sung thêm các khái niệm hướng đối tượng vào ngôn ngữ C.
- <Đặc trưng>
- Nó cung cấp sự tương hợp hoàn toàn với ngôn ngữ C.
- Nó bao gồm các khái niệm về "lớp," "kế thừa," và "hàm ảo," được dùng trong lập trình hướng đối tượng.
- ③ Java
- Java là ngôn ngữ hướng đối tượng kiểu thông dịch dựa trên ngôn ngữ C và C++. Chạy Java trên trình duyệt WWW trên Internet tạo khả năng cho các thao tác tương tác.
- <Đặc trưng>

- Ngôn ngữ này có thể chạy trên bất kì nền nào.
- Độ an toàn được nâng cao và có cung cấp các chức năng kết mạng.
- Cung cấp các chức năng động.
- ④ Perl (Practical Extraction and Report Language - Ngôn ngữ báo cáo và trích rút thực hành)
 - Perl, ngôn ngữ thông dịch do Larry Wall phát triển cho xử lí văn bản, được dùng trên các hệ điều hành họ UNIX và trên Windows. Nó chịu ảnh hưởng lớn của ngôn ngữ C và script vỏ.
 - <Đặc trưng>
 - Nó được dùng như ngôn ngữ chuẩn cho các chương trình CGI (Common Gateway Interface) chạy trên các nguồn phục vụ WWW.
 - Nó là ngôn ngữ script tương tác.
- ⑤ SGML (Standard Generalized Markup Language - Ngôn ngữ đánh dấu tổng quát chuẩn)
 - SGML là ngôn ngữ để mô tả cấu trúc logic và cấu trúc ngữ nghĩa của tài liệu bằng các thẻ đơn giản (các dấu hiệu).
 - <Đặc trưng>
 - Nó đã được phát triển như một ngôn ngữ để cho phép các tài liệu điện tử được xử lí trên máy tính. Tuy nhiên, việc đảm bảo sự phức tạp của nó đã ngăn cản không cho phép nó được sử dụng rộng rãi.
- ⑥ HTML (Hyper Text Markup Language - Ngôn ngữ đánh dấu siêu văn bản)
 - HTML là ngôn ngữ để mô tả siêu văn bản được dùng trên WWW và các ứng dụng khác.
 - <Đặc trưng>
 - - Nó là phiên bản mở rộng của SGML.
 - - Việc dùng các thẻ tạo khả năng đặc tả dễ dàng cho các tệp ảnh hoặc các liên kết.
- ⑦ XML (eXtensible Markup Language - Ngôn ngữ đánh dấu mở rộng)
 - XML, được đặt vào vị trí là ngôn ngữ kế tục HTML, là ngôn ngữ mô tả trang trong tiến trình được chuẩn hoá. XML là ngôn ngữ tổ hợp những điểm mạnh của cả HTML và SGML trong đó các chức năng mấu chốt của HTML được mở rộng và SGML được tối ưu cho sử dụng Internet. Công trình chuẩn hoá của liên đoàn WWW về XML đã được hoàn tất vào tháng 12/1997.
 - <Đặc trưng>
 - - So với HTML, nó cung cấp mức độ tự do và linh hoạt cao hơn cho người dùng.
 - - Các ứng dụng được trông đợi để bao gồm cả trao đổi dữ liệu trong thương mại điện tử, lưu giữ tài liệu xí nghiệp, và truyền phát BS số thức.
- ⑧ PostScript
 - PostScript là ngôn ngữ mô tả trang do Adobe Systems phát triển. PostScript được dùng như chuẩn mặc định trong xuất bản tại bàn (chế bản điện tử).
 - <Đặc trưng>

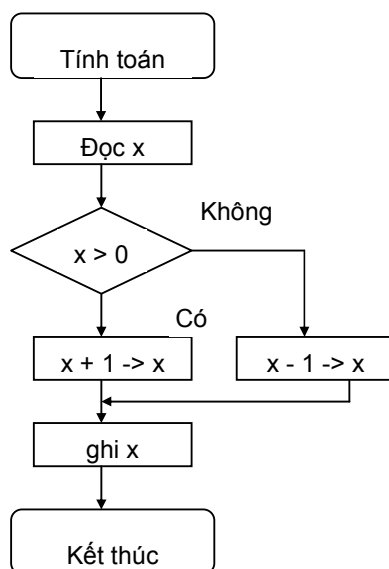
- Các mô tả là độc lập với hiệu năng của thiết bị đưa ra.
- Việc nhúng một bộ thông dịch PostScript vào máy in tạo khả năng in ra nhiều kiểu fonts với tốc độ cao.
- Khi phần mềm thông dịch PostScript được dùng, thì máy in thường có thể tạo ra cùng chất lượng như máy in có nhúng PostScript.

1.4 Kỹ thuật lập trình

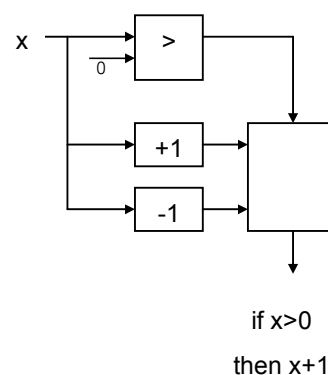
1.4.1 Lập trình thủ tục

- Kỹ thuật lập trình dùng "thủ tục" để mô tả giải pháp được gọi là lập trình thủ tục. Lập trình thủ tục là ngôn ngữ lập trình đầu tiên được dùng trong các ngôn ngữ cấp cao. Nó là kỹ thuật lập trình qui ước được sử dụng rộng rãi nhất ngày nay.
- Các ngôn ngữ hỗ trợ cho lập trình thủ tục bao gồm FORTRAN, COBOL, PL/1, Pascal, ALGOL, BASIC và ngôn ngữ C. Trong những ngôn ngữ này, cú pháp được cung cấp cho từng lệnh để làm cho máy tính thực hiện. Nói cách khác, một chương trình thủ tục được coi như kết tập của các câu lệnh. Cũng có thể nói rằng, "chương trình bao gồm các cấu trúc dữ liệu và các thuật toán." Điều này là ích lợi nhất cho phương pháp chương trình được lưu trữ, một đặc trưng của máy tính kiểu von Neumann.
- Tính hiệu quả và tính bảo trì của lập trình thủ tục được nâng cao thêm với việc đưa vào lập trình có cấu trúc và sơ đồ có cấu trúc.
- Khái niệm về lập trình có cấu trúc là diễn đạt mọi thuật toán bằng việc dùng ba cấu trúc điều khiển cơ bản (tuần tự, tuyển chọn và lặp). Bằng việc dùng sơ đồ lập trình này, chúng ta có thể tối thiểu hoá việc dùng câu lệnh goto thường gây ra vấn đề trong công việc bảo trì.

Hình 1-4-1 Lập trình thủ tục và lập trình hàm



Phương pháp thủ tục
(luồng điều khiển)



Phương pháp hàm
(luồng dữ liệu)

1.4.2 Lập trình hàm

- Kỹ thuật lập trình dùng các hàm để mô tả cho giải pháp được gọi là lập trình hàm. LISP, được phát triển để làm cho việc giải quyết các ký tự và ký hiệu dễ dàng hơn, là ngôn ngữ điển hình hỗ trợ cho lập trình hàm. LISP ban đầu được phát triển để dùng trong nghiên cứu về trí tuệ nhân tạo.

Đặc trưng của ngôn ngữ này là việc dùng cấu trúc danh sách. Bởi vì các đối tượng (dữ liệu) được xử lý là được giải quyết như các cấu trúc danh sách nên nhiều hàm được cung cấp để xử lý danh sách. Để mô tả và định nghĩa các hàm, một hệ thống ngôn ngữ trừu tượng cao, gọi là tính toán lambda, được dùng tới.

- Với lập trình thủ tục, các giá trị trong cấu trúc dữ liệu trong bộ nhớ chính bị thay đổi khi chương trình được thực hiện, chỉ ra rằng hiệu quả phụ động xảy ra. Tuy nhiên, với lập trình hàm, các hàm xử lý dữ liệu vào và đưa ra các giá trị hàm. Cho nên nó không chứa hiệu quả phụ. Việc viết lại các giá trị trong cấu trúc dữ liệu là không được phép. Việc này cũng còn được gọi là lập trình khai báo, vì chương trình được mô tả theo kiểu khai báo. Luồng điều khiển được mô tả bằng lập trình thủ tục, trong khi luồng dữ liệu lại được mô tả bằng lập trình hàm, và trong trường hợp đó có sự khác biệt giữa hai cách lập trình này (xem Hình 1-4-1).

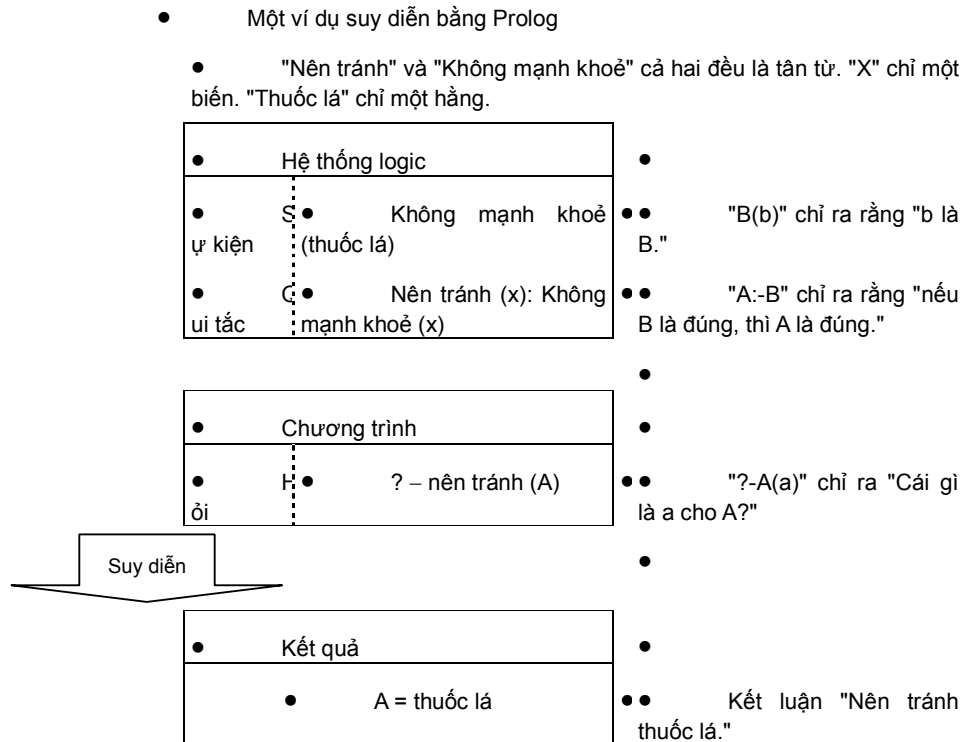
•

1.4.3 Lập trình logic

- Phương pháp lập trình, mô tả giải pháp bằng một "kết tập các khai báo" được gọi là lập trình logic. Lập trình logic bao gồm các biểu thức logic và cơ chế lập luận thuộc vào hệ thống logic nền tảng, và không có hiệu quả phụ. Prolog là ngôn ngữ lập trình tiêu biểu hỗ trợ cho lập trình logic. Đặc trưng của ngôn ngữ này là qui tắc giải dựa trên tam đoạn luận. Trong trường hợp của Prolog, ba cú pháp được dùng: qui tắc, sự kiện và hỏi. Việc sánh mẫu (thống nhất) tìm tự động (lần ngược) v.v.. Được dùng như cơ chế điều khiển cơ bản.

- Hình 1-4-2 Ví dụ về lập trình logic

•



- Với lập trình logic, các quan hệ cố hữu trong vấn đề và các ràng buộc được mô tả theo kiểu khai báo. Các quan hệ vào-ra được mô tả bằng các biểu thức logic. Kết quả của tính toán (suy diễn)

là hoặc "thành công" hoặc "không thành công." Việc đưa vào một biểu thức logic và chứng minh tiếp đó rằng biểu thức này trùng với kết luận của biểu thức logic đã cho xem như chương trình là tương đương với việc thực hiện (tính toán) của chương trình.

- Như được lấy làm điển hình trong Hình 1-4-2, suy diễn này được tiến hành bằng qui tắc tam đoạn luận.
- "Hút thuốc là không mạnh khỏe." → "Nên tránh việc dùng thứ không mạnh khỏe." → "Nên tránh hút thuốc lá."
-

1.4.4 Lập trình hướng đối tượng

Phương pháp lập trình mô tả cho giải pháp bằng trạng thái của đối tượng và hành vi của chúng được gọi là lập trình hướng đối tượng. Smalltalk-80 là ngôn ngữ điển hình hỗ trợ cho lập trình hướng đối tượng. Tuy nhiên, các ngôn ngữ lập trình sau đây, tất cả đều là những phiên bản mở rộng của các ngôn ngữ đã có, cũng đang được sử dụng rộng rãi.

- - Phiên bản mở rộng của C → C++ và Objective-C
- - Phiên bản mở rộng của Prolog → ESP
- - Phiên bản mở rộng của LISP → Flavor và CLOS
- - Phiên bản mở rộng của Pascal → Object Pascal
- Như được mô tả trong 1.2.3, với cách lập trình hướng đối tượng, việc phát triển chương trình được tiến hành có hiệu quả bằng việc dùng các tiện nghi sau:

•

- | | | |
|-----------------------|--------------------|---|
| • Hướn
g đối tượng | • Thử nghiệm | • • Tạo ra đối tượng từ lớp |
| | • Bao bọc | • • Tổ hợp dữ liệu và phương pháp |
| | • Truyền thông báo | • • Gửi/nhận thông báo |
| | • Kế thừa | • • Làm cho con kế thừa các đặc trưng của bố mẹ |

1.5 Phương pháp kiểm thử và kiểm điểm

- Kiểm thử và kiểm điểm là cốt yếu để nâng cao chất lượng phần mềm

1.5.1 Phương pháp kiểm thử

- Phương pháp kiểm thử bao gồm kiểm thử hộp trắng và kiểm thử hộp đen.

(1) Kiểm thử hộp trắng

- Với kiểm thử hộp trắng, các trường hợp kiểm thử được thiết kế bằng việc chú ý đặc biệt tới cấu trúc bên trong của mô đun, cùng logic và luồng điều khiển. Trong kiểm thử, người ta mong muốn kiểm điểm tất cả các lệnh (câu lệnh) có trong chương trình chi tiết. Tuy nhiên, điều đó là khó bởi vì khối lượng công việc cần làm. Do đó, thiết kế phải tính tới sự cân xứng giữa "mức độ bao phủ" và "năng suất."

- Mức độ bao phủ được dùng trong thiết kế các trường hợp kiểm thử được mô tả như sau:

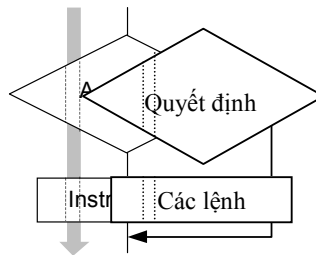
- ① Bao phủ lệnh

- Bao phủ lệnh (câu lệnh) nói tới việc thiết kế các trường hợp kiểm thử tạo khả năng cho mọi lệnh (câu lệnh) trong chương trình đều được thực hiện ít nhất một lần. Bao phủ lệnh không tính tới hiệu quả của quyết định (đường đi).

-

Hình 1-5-1

Bao phủ lệnh



-

- Trong ví dụ nêu ở Hình 1-5-1, tất cả các lệnh được thực hiện đều được bao trong một đường đi. Việc kiểm thử đường đi bao gồm tất cả các lệnh này. Do đó, việc cung cấp chỉ một phép kiểm tra là đủ cho kiểm thử này.

-

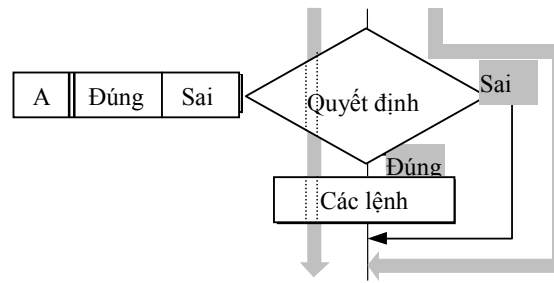
- ② Bao phủ quyết định

- Bao phủ quyết định nói tới thiết kế các trường hợp kiểm thử có bao hàm việc thực hiện điều kiện "đúng" và việc thực hiện điều kiện "sai" ít nhất một lần cho mỗi quyết định (xem Hình 1-5-2).

-

Hình 1-5-2

Bao phủ quyết định



- Trong ví dụ được vẽ trong Hình 1-5-2, quyết định (nhánh) được bao phủ nếu hai đường nhánh đều được kiểm thử. Tuy nhiên quyết định được thực hiện theo những điều kiện hợp thành, có thể có điều kiện sẽ không được thử.

•

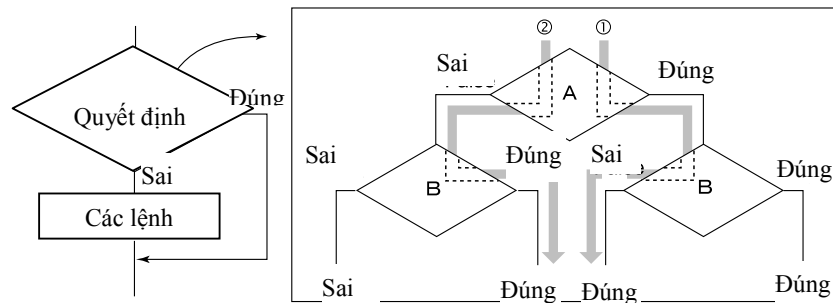
③ Bao phủ điều kiện

- Bao phủ điều kiện nói tới thiết kế các trường hợp kiểm thử có khả năng kiểm thử tổ hợp các điều kiện trong đó các trạng thái "đúng" và "sai" của từng điều kiện đều được thực hiện ít nhất một lần.

•

Hình 1-5-3

Bao phủ điều kiện



•

- Chẳng hạn, nếu điều kiện quyết định trong Hình 1-5-3 là "Nếu A = đúng Hoặc B = đúng," thì tất cả các đường đi được lấy khi hoặc A hoặc B đúng phải được bao phủ bởi các trường hợp kiểm thử. Cho nên, dữ liệu gây ra việc thực hiện của hai đường được vẽ trong Hình 1-5-4 thiết lập nên các trường hợp kiểm thử.

•

Hình 1-5-4

Trường hợp kiểm thử của bao phủ điều kiện

	①	②
A	Đúng	Sai
B	Sai	Đúng

•

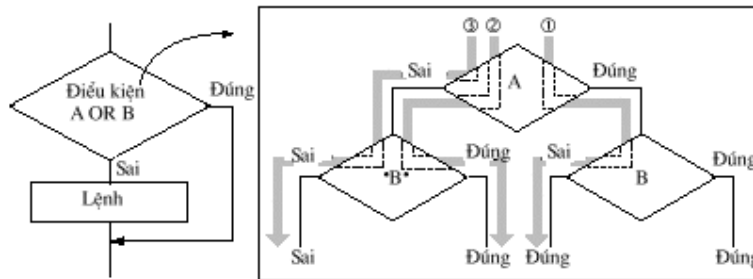
④ Bao phủ quyết định/điều kiện

- Trong trường hợp bao phủ điều kiện, các trạng thái "đúng" và "sai" của từng điều kiện A và B của đa điều kiện đều được kiểm thử. Tuy nhiên, đường đi cho "A = "đúng" và B = "đúng"" và đường đi cho "A = "sai" và B = "sai" không được kiểm thử. Mặt khác, bao phủ quyết định/điều kiện được áp dụng cho bao phủ điều kiện được tổ hợp thêm (xem Hình 1-5-5).

•

Hình 1-5-5

Ví dụ về bao phủ quyết định/điều kiện



Trong ví dụ được vẽ trong Hình 1-5-5, đường đi được thực hiện khi điều kiện là "sai" được bao phủ bởi trường hợp kiểm thử. Do đó, ba đường đi được vẽ trong Hình 1-5-6 được trường hợp kiểm thử này bao quát.

Hình 1-5-6

Các trường hợp kiểm thử cho bao quát quyết định/điều kiện

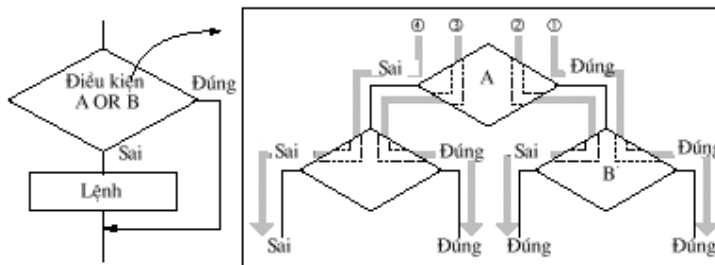
	①	②	③
A	Đúng	Sai	Sai
B	Sai	Đúng	Sai

⑤ Bao phủ đa điều kiện

Bao phủ đa điều kiện sử dụng các trường hợp kiểm thử bao quát tất cả các đường đi được thực hiện tùy thuộc vào tất cả các tổ hợp điều kiện "đúng" và "sai" và làm cho mọi lệnh được bao hàm đều được thực hiện ít nhất là một lần.

Hình 1-5-7

Bao phủ đa điều kiện



Trong ví dụ được vẽ trong Hình 1-5-7 nơi điều kiện quyết định là "Nếu A = đúng hoặc B = sai, " các đường đi được chọn giữa tất cả các giá trị của A và B đều được xét để chuẩn bị cho các trường hợp kiểm thử. Kết quả là bốn đường đi sau đây trong Hình 1-5-8 tạo nên các tổ hợp được thực hiện bởi các trường hợp kiểm thử.

Hình 1-5-8

Trường hợp kiểm thử cho bao phủ đa điều kiện

	①	②	③	④
A	Đúng	Đúng	Sai	Sai
B	Đúng	Sai	Đúng	Sai

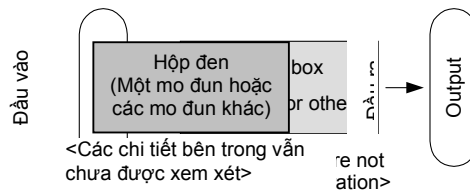
(2) Kiểm thử hộp đen

Trong kiểm thử hộp đen, không xem xét tới cấu trúc bên trong và cấu trúc logic của mô đun. Nói cách khác, mô đun được coi như hộp đen. Sau đó, dữ liệu kiểm thử được thiết kế bằng cách chỉ chú ý tới giao diện (cái vào và cái ra) của từng mô đun.

•

Hình 1-5-9

Kiểm thử hộp đen



Các trường hợp kiểm thử được thiết kế theo cách sau:

① Phân hoạch tương đương

Trong phân hoạch tương đương, dữ liệu được phân hoạch thành nhiều nhóm khác nhau (các lớp tương đương), mỗi phần tử có cùng tính chất. Sau đó một mảnh dữ liệu trong mỗi nhóm được dùng như giá trị đại diện cho từng nhóm. Trong phân hoạch tương đương, dữ liệu được phân hoạch thành một trong hai lớp sau:

•

• Lớp tương đương • • • Lớp dữ liệu nằm trong miền hợp lệ.
hợp lệ

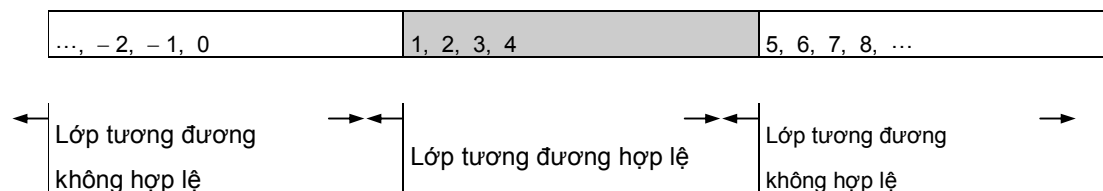
• • • Lớp tương đương • • • Lớp dữ liệu nằm trong miền không hợp lệ (lỗi).
không hợp lệ

Sau đó dữ liệu điển hình cho từng lớp được lựa ra cho trường hợp kiểm thử.

•

Hình 1-5-10 Phân hoạch tương đương

•



Dữ liệu kiểm thử: -1, 3, 8

•

Trong Hình 1-5-10, nếu lớp tương đương hợp lệ bao gồm 1 tới 4, thì tập dữ liệu biểu diễn cho từng lớp, chẳng hạn, "-1, 3, 8" có thể được dùng làm dữ liệu kiểm thử.

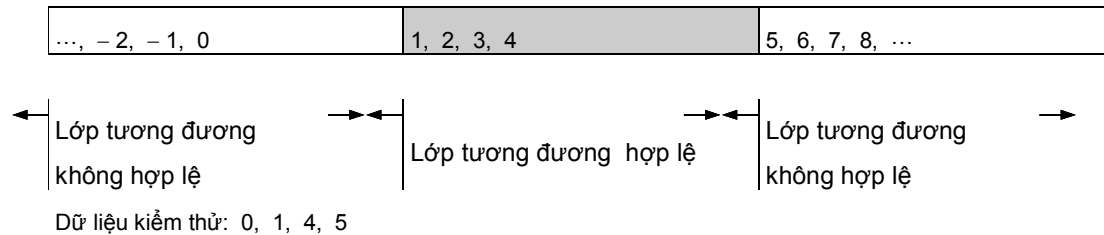
•

• ② Phân tích giá trị biên

Phân tích giá trị biên là phương pháp phân tích dùng các giá trị biên của lớp tương đương hợp lệ làm dữ liệu kiểm thử.

Hình 1-5-11 Phân tích giá trị biên

•



Trong Hình 1-5-11, nếu lớp tương đương hợp lệ bao gồm dữ liệu của 1 tới 4, thì tập các giá trị biên của từng lớp, chẳng hạn, "0, 1, 4, 5" có thể được dùng làm dữ liệu kiểm thử.

• ③ Đồ thị nhân-quả

Trong phân hoạch tương đương và phân tích giá trị biên, dữ liệu được phân loại để phân tích. Tuy nhiên, đồ thị nhân-quả cung cấp một phương pháp để thiết kế các trường hợp kiểm thử có hiệu quả cho các mô đun mà việc phân hoạch thành lớp có khó khăn.

<Cách tạo ra đồ thị nhân quả>

- 1. Tất cả các nguyên nhân (cái vào) và kết quả (cái ra) đều được lấy dựa trên đặc tả.
- 2. Mỗi quan hệ giữa nguyên nhân (cái vào) và kết quả (cái ra) được diễn đạt để làm cho mỗi quan hệ logic được rõ ràng.
- 3. Bảng quyết định được chuẩn bị dựa trên các đồ thị. Dữ liệu kiểm thử được tạo ra dựa trên các bảng này.
- Ví dụ
- (Mô tả về đặc tả)
- - Số các chủ thể kiểm tra là 5.
- - "Pass" là cái ra, nếu kết quả của bốn hay nhiều chủ thể là thành công.
- - "Temporary pass" là cái ra, nếu kết quả trong hai hay ba chủ thể là không thành công.
- - "Failure" là cái ra, nếu kết quả trong chỉ một chủ thể là thành công.
- (Thủ tục 1) Lấy ra tất cả các quan hệ giữa nguyên nhân (cái vào) và kết quả (cái ra) dựa trên đặc tả.

Hình 1-5-12

Quan hệ giữa nguyên nhân
(cái vào) và kết quả (cái ra)

Nguyên nhân (cái vào)

Kết quả (cái ra)

① Kết quả trong bốn hay nhiều chủ thể là thành công

② Kết quả trong hai hay nhiều chủ thể là thành công

③ Pass

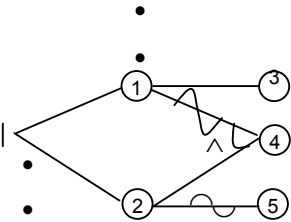
④ Temporary pass

⑤ Failure

- (Thủ tục 2) Diễn đạt mối quan hệ giữa nguyên nhân (cái vào) và hậu quả (cái ra) trong đồ thị để làm cho mối quan hệ logic được rõ ràng.

Hình 1-5-13

Đồ thị nhân-quả



Các số ① tới ⑤ tương ứng với những số trong Hình Figure 1-5-12 (Thủ tục 1).

- Ý nghĩa của kí hiệu được dùng trong đồ thị nhân-quả là như sau:

Kí hiệu	Kí hiệu	N	Mô tả
1)	T	ương đương	Nếu ① đúng, ② đúng.
2)	A	ND (tích)	Nếu ① đúng và ② đúng, ③ đúng.
3)	O	R (tổng)	Nếu ① đúng hay ② đúng, ③ đúng.
4)	N	OT (Phủ định)	Nếu ① sai, ② đúng.
5)	B	ao hàm	① bao hàm ②.
6)	L	oại trừ	Nếu ① đúng, ② sai, hay nếu ① sai, ② đúng.
7)	Y	êu cầu	① yêu cầu ②.

- (Thủ tục 3) Bảng quyết định được chuẩn bị dựa trên các đồ thị.

Hình 1-5-14

Bảng quyết định

Trường hợp kiểm thử		1	2	3
Nguyên nhân và kết quả				
Nguyên nhân	① Qua bốn hay nhiều chủ thể	Y	N	N
	② Qua 2 hay nhiều chủ thể	—	Y	N
	③			
Kết quả	④ “Pass” được đưa ra	X	—	—
	⑤ “Temporary” được đưa ra	—	X	—
	⑥ “Failure” được đưa ra	—	—	X

Quy tắc được dùng trong bảng quyết định được vẽ như sau:				
Tên bảng	Tiêu đề quy tắc			
Điều kiện 1				
Điều kiện 2				
Điều kiện 3				
⋮				
Điều kiện n				
Điều kiện 1				

• Điều kiện 2	•	•	•	•
• Điều kiện 3	•	•	•	•
• ⋮	•	•	•	•
• Điều kiện n	•	•	•	•

- Tên bảng: chỉ ra tên logic
- • Tiêu đề qui tắc: chỉ ra số hiệu để phân biệt
- các qui tắc quyết định logic.
- • Các hàng cho điều kiện: Mỗi hàng bao gồm
- một điều kiện để ra quyết định trong chương trình.
- • Y: "true"
- • N: "false"
- —: Không ra quyết định.

④ Phương pháp thiết kế thực nghiệm

Lí tưởng là thiết kế mọi trường hợp kiểm thử có thể quan niệm được và tiến hành các kiểm thử bằng việc dùng chúng. Tuy nhiên, điều này yêu cầu khối lượng lớn công việc kiểm thử. Do đó, phương pháp thiết kế thực nghiệm, một dạng của phân tích thống kê, được dùng cho các kiểm thử có yêu cầu số lượng lớn dữ liệu. Việc phân tích dữ liệu kiểm thử bằng phương pháp này và dùng các kết quả như các trường hợp kiểm thử tạo khả năng tiến hành kiểm thử hiệu quả.

Ví dụ Kiểm tra dữ liệu về sinh viên -- việc kiểm tra hợp lệ về số hiệu đăng kí sinh viên và mã các trường phổ thông họ tốt nghiệp ra, và kiểm tra về bậc -- được tiến hành.

- (Thủ tục 1) Tám trường hợp kiểm thử được hình dung ra, nếu tất cả các trường hợp kiểm thử đều được giả thiết cho tất cả các tổ hợp của ba khoản mục này: kiểm tra tính hợp lệ của số đăng kí sinh viên và kiểm tra tính hợp lệ của mã trường phổ thông họ tốt nghiệp và kiểm tra về bậc tốt nghiệp.

Hình 1-5-15

Dữ liệu sinh viên

Khoản mục	Số hiệu kiểm thử	Số đăng kí sinh viên	Mã trường phổ thông	Bậc
• 1	• 1	• Hợp lệ	• Hợp lệ	• Số được đưa vào
• 2	• 2	• Hợp lệ	• Hợp lệ	• Số không được đưa vào
• 3	• 3	• Hợp lệ	• Không hợp lệ	• Số được đưa vào
• 4	• 4	• Hợp lệ	• Không hợp lệ	• Số không được đưa vào
• 5	• 5	• Không hợp lệ	• Hợp lệ	• Số được đưa vào
• 6	• 6	• Không hợp lệ	• Hợp lệ	• Số không được đưa vào

• 7	• Không hợp lệ	• Không hợp lệ	• Số được đưa vào
• 8	• Không hợp lệ	• Không hợp lệ	• Số không được đưa vào

- (Thủ tục 2) Như với ba khoản mục này, bảng sau (ma trận) chỉ ra mối quan hệ giữa chúng được tạo ra:

Hình 1-5-16 Bảng chỉ ra mối quan hệ

• Số hiệu đăng kí sinh viên	• Hợp lệ	• Không hợp lệ
• Mã trường phổ thông được tốt nghiệp từ	• Số được đưa vào	• Số không được đưa vào
• Hợp lệ	• Số không được đưa vào	• Số được đưa vào
• Không hợp lệ	• Số không được đưa vào	• Số được đưa vào

- Được tổ hợp
- <Trường hợp kiểm thử>

• Số hiệu kiểm thử.	• Mã của trường phổ thông được tốt nghiệp từ	• Số đăng kí sinh viên	• Bạc
• 1	• Hợp lệ	• Hợp lệ	• Số được đưa vào
• 4	• Hợp lệ	• Không hợp lệ	• Số không được đưa vào
• 6	• Không hợp lệ	• Hợp lệ	• Số không được đưa vào
• 7	• Không hợp lệ	• Không hợp lệ	• Số được đưa vào

- (Thủ tục 3) Khi được so sánh với dữ liệu về sinh viên trong Hình 1-5-15, người ta thấy rằng bốn trong chúng được lấy ra. Có thể kết luận rằng, nếu tất cả bốn phép kiểm thử này đều cho kết quả thành công, thì bốn phép kiểm thử kia cũng phải cho kết quả thành công.

1.5.2 Phương pháp kiểm điểm

- Để thực hiện phát triển hệ thống theo cách từ trên xuống, cần phải phát hiện lỗi sớm nhất có thể được. Điều này là vì các lỗi bị ẩn kín sẽ ảnh hưởng rất lớn tới công việc ở những giai đoạn sau. Các lỗi được sinh ra trong thiết kế hệ thống đại thể ảnh hưởng còn lớn hơn, như trong pha lập kế hoạch cơ sở và thiết kế ngoài gây ảnh hưởng rất lớn, làm tốn kém rất nhiều để sửa lỗi. Công việc

sửa lỗi trong bảo trì nghe nói còn gấp vài chục lần việc sửa lỗi trong viết mã.

- Cho dù kiểm thử có được tiến hành bằng việc bao quát về mặt lí thuyết mọi trường hợp kiểm thử có thể quan niệm được, thì việc khử bỏ đi mọi lỗi vẫn cứ là khó khăn theo quan điểm cả những ràng buộc về thời gian cần thiết và các nhân tố khác. Tuy nhiên, việc làm giảm lỗi nhiều nhất có thể được vẫn có tác động lớn lên chất lượng và chí phí sửa lỗi.

- Do đó, phải tiến hành hết mức những nỗ lực để hợp kiểm điểm đưa ra các biện pháp có hiệu quả tìm ra lỗi.

- Hợp kiểm điểm nêu ra thảo luận và đánh giá điều được thực hiện, trong từng pha từ việc lập kế hoạch cơ sở cho tới lập trình, để rút ra những mơ hồ hay điểm có vấn đề nằm trong cái ra (các tài liệu thiết kế khác nhau và mã nguồn) của từng pha. Nói riêng, cuộc hợp kiểm điểm được tiến hành trong pha thiết kế được gọi là "kiểm điểm thiết kế," trong khi hợp kiểm điểm cho pha phát triển mã chương trình được gọi là "kiểm điểm mã."

- Các tác động sau đây được trông đợi từ việc tiến hành các cuộc hợp kiểm điểm.

- <Các tác động của hợp kiểm điểm>

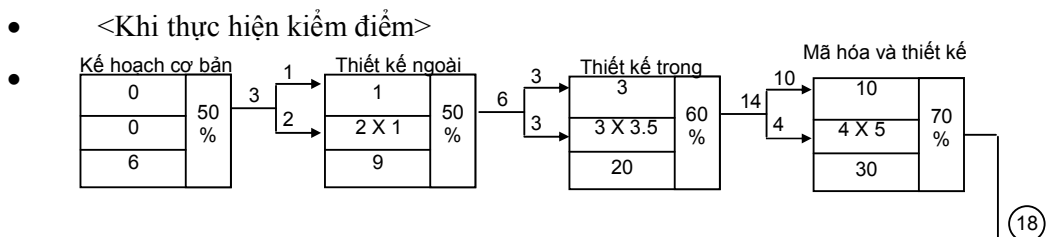
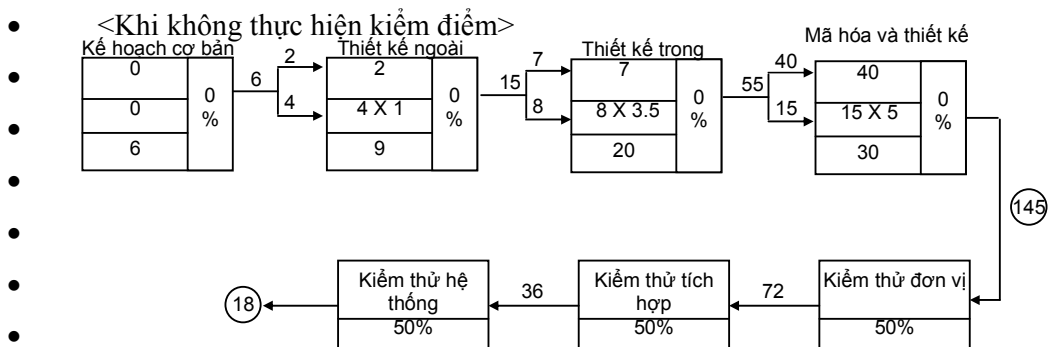
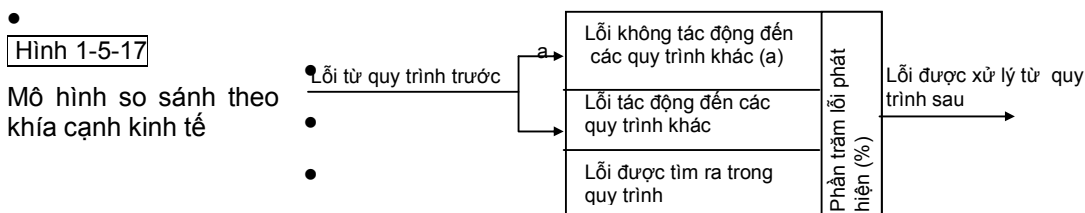
- Cơ hội để xem xét lại các điểm còn mơ hồ đã nêu, cho phép những người tham dự chia sẻ cùng hiểu biết.

- Vấn đề (lỗi), mà người chịu trách nhiệm phát triển thực tế không thể tìm ra nổi, lại có thể được người khác phát hiện ra.

- Các chức năng, hiệu năng và chất lượng (kể cả độ tin cậy, tính vận hành và tính bảo trì) đều có thể được cải thiện.

- Trong những khoản mục được mô tả trên đây, hợp kiểm điểm đặc biệt mạnh trong việc phát hiện ra vấn đề (lỗi).

- Ta hãy so sánh, với ví dụ được cho trong Hình 1-15-17 xem như mô hình, các khía cạnh kinh tế cho trường hợp kiểm điểm được tiến hành và trường hợp khác mà kiểm điểm không được thực hiện.



-
-
-
-
-
-
- <Khi kiểm điểm không được thực hiện>
- (Lập kế hoạch cơ sở)
 - Sáu lỗi được len vào và được truyền qua thiết kế ngoài tiếp theo.
- (Thiết kế ngoài)
 - Chín lỗi mới được len vào, và 15 lỗi tổng cộng được truyền qua thiết kế trong tiếp theo.
- (Thiết kế trong)
 - Trong số 15 lỗi, 7 lỗi được truyền qua thiết kế trong. Ta giả thiết ở đây là 8 lỗi còn lại ảnh hưởng tới các lỗi khác, làm len thêm trung bình 3.5 lỗi cho mỗi lỗi nhận được. Bên cạnh đó, 20 lỗi mới được len vào trong pha này, đem tổng số lỗi bị truyền sang pha thiết kế chương trình và viết mã là 55.
- (Thiết kế chương trình và viết mã)
 - Trong số 55 lỗi tổng cộng, 40 lỗi truyền qua pha thiết kế chương trình và viết mã. Tuy nhiên, ta lại giả thiết ở đây rằng 15 lỗi còn lại ảnh hưởng tới các lỗi khác, làm len thêm trung bình năm lỗi cho mỗi lỗi nhận được. Bên cạnh đó, 30 lỗi mới được tạo ra, đem số lỗi bị truyền vào pha tiếp là 145.
- (Kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử hệ thống)
 - Ta giả sử rằng tỉ lệ phát hiện lỗi là 50% cho từng kiểm thử.
- (Kết quả)
 - Hệ thống hoàn chỉnh chứa 18 lỗi.
-
- <Khi kiểm điểm được thực hiện>
- (Lập kế hoạch cơ sở)
 - Sáu lỗi được len vào. Giả thiết rằng hợp kiểm điểm ở đây tạo khả năng lỗi được phát hiện quãng 50%, số lỗi bị truyền qua pha thiết kế ngoài trở thành ba.
- (Thiết kế ngoài)
 - Chín lỗi mới được len vào. Giả sử rằng hợp kiểm điểm ở đây tạo khả năng lỗi được phát hiện quãng 50%, số lỗi bị truyền qua pha thiết kế ngoài trở thành sáu.
- (Thiết kế trong)
 - Trong số sáu lỗi, ba lỗi được truyền qua thiết kế trong. Ta giả thiết ở đây là ba lỗi còn lại ảnh hưởng tới các lỗi khác, làm len thêm trung bình 3.5 lỗi cho mỗi lỗi nhận được. Bên cạnh đó, giả sử rằng 20 lỗi mới được tạo ra và rằng hợp kiểm điểm tạo khả năng lỗi được phát hiện với tỉ lệ 60%, số lỗi bị truyền sang pha thiết kế chương trình và viết mã trở thành 14.

- (Thiết kế chương trình và viết mã)
 - Trong số 14 lỗi tổng cộng, 10 lỗi truyền qua pha thiết kế chương trình và viết mã. Ta lại giả thiết ở đây rằng bốn lỗi còn lại ảnh hưởng tới các lỗi khác, làm len trung bình năm lỗi cho mỗi lỗi nhận được. Bên cạnh đó, giả sử rằng 30 lỗi mới được tạo ra và rằng hợp kiểm điểm tạo khả năng lỗi được phát hiện với tỉ lệ 70%, số lỗi bị truyền vào pha kiểm thử trở thành 18.
- (Kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử hệ thống)
 - Ta giả sử rằng tỉ lệ phát hiện lỗi là 50% cho từng kiểm thử.
- (Kết quả)
 - Hệ thống đã phát triển chứa 2 lỗi.
- Các trường hợp khác nhau được xem xét. Tuy nhiên, như được mô tả ở trên, tổng số lỗi còn lại trong trường hợp kiểm điểm được thực hiện là hoàn toàn khác. Do đó, kiểm điểm hiệu quả nên được tiến hành.
- Các phương pháp hợp kiểm điểm điển hình bao gồm giải trình (walkthroughs) và giám định (inspection).

(1) Giải trình

- Giải trình được tiến hành bởi người tạo ra vật phẩm, và những người có liên quan (ngang quyền) nhằm phát hiện lỗi trong thiết kế và mã hoá sớm nhất có thể được để cải thiện chất lượng.

(2) Giám định

- Giám định, một phương pháp chính thức hoá việc giải trình, nhằm phát hiện lỗi trong thiết kế và mã hoá sớm nhất có thể được để làm tăng chất lượng. Cuộc họp giám định, do một người chủ toạ, còn gọi là "người dẫn chương trình", chịu trách nhiệm tổ chức họp, sẽ kiểm điểm lại các tiến trình kể cả việc sửa lỗi và kiểm tra kết quả sửa chữa, và vai trò của từng người tham dự cũng được xác định chi tiết.

(3) Những điểm cần được xem xét trong khi tiến hành hợp kiểm điểm

- Khi tiến hành hợp kiểm điểm, thì những điểm sau nên được tính tới:
- <Những điểm cần xét tới>
- Tài liệu sử dụng trong cuộc họp nên được phân phát hai hay ba ngày trước đó, và mọi người tham dự cần phải đọc chúng trước khi vào thảo luận với những người tham dự khác.
- Tập trung vào việc phát hiện lỗi và làm cuộc họp ngắn nhất có thể được. Tổ chức các cuộc họp như vậy thường xuyên sẽ làm giảm khối lượng công việc phải làm lại.
- Nhiều người, nhưng chỉ người phát triển mới nên tham gia vào cuộc họp này.
- Người quản lý nên tránh tham dự vào cuộc họp này. Cuộc họp này được tổ chức để phát hiện lỗi. Nếu người quản lý tham dự vào cuộc họp, thì những người tham dự sẽ ngần ngại đưa ra ý kiến riêng của mình một cách tự do, làm giảm tiềm năng phát hiện lỗi mà cuộc họp đáng có.

1.5.3 Thiết kế kiểm thử và phương pháp quản lý

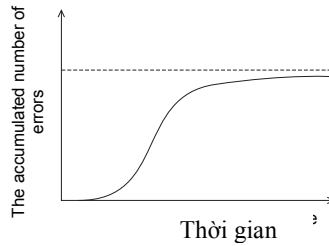
(1) Đường cong lỗi

- Đường cong lỗi đưa ra một mô hình dùng một số các lỗi để ước lượng về mặt định lượng độ tin cậy của chương trình. Đường cong này, cũng còn được gọi là "đường cong tăng trưởng độ tin cậy" và "mô hình cận toán học," được chuẩn bị bằng việc đặt số lỗi tích lũy vào trục đứng và thời gian vào trục ngang. Một điểm uốn được định vị ở giữa đường cong này và độ hội tụ cuối cùng được đạt tới. Đường cong hậu cần và đường cong Gompertz thường được dùng cho mục đích này.

• Hình 1-5-18

• Ví dụ về đường cong lỗi ARC (đường cong tăng trưởng độ tin cậy)

Số lỗi tích lũy



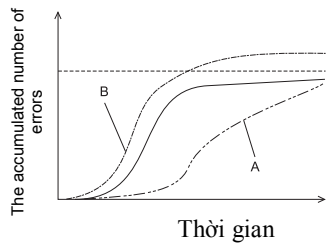
(2) Biểu đồ quản lí lỗi

- Biểu đồ quản lí lỗi quản lí các lỗi bằng việc dùng đường cong lỗi (xem Hình 1-5-19).

• Hình 1-5-19

• Ví dụ về biểu đồ quản lí lỗi

Số lỗi tích lũy



- Dữ liệu thực tế được so sánh với đường cong lỗi, và có thể tiến hành các đánh giá sau.

- | | |
|--|---|
| • - Việc xây dựng bị chậm (A) | • - Dùng các trường hợp kiểm thử phát hiện và môi trường kiểm thử không thỏa đáng cần được xem xét. |
| • - Sự hội tụ không đạt tới (A) | • - Chương trình phải được xem xét lại. |
| • - Số các lỗi tích lũy được phát hiện vẫn còn thấp hơn đường cong lỗi (A) | • - Chất lượng có thể cao hay các trường hợp kiểm thử được dùng có thể không tốt. |
| • - Số các lỗi tích lũy được phát hiện vượt hơn đường cong lỗi (B) | • - Các trường hợp kiểm thử được dùng có thể tốt hay phẩm chất của chương trình có thể không tốt. |

- Tính tới những dữ liệu thu được thực tế và mức độ kĩ năng của người phát triển, có thể đưa ra đánh giá về trạng thái của các tiến trình kiểm thử và chất lượng của chương trình.

(3) Bao phủ

- "Bao phủ" cũng còn được gọi là "bao phủ kiểm thử." Việc bao phủ một chương trình nghĩa

là số phần trăm các con đường được bao phủ tính theo tất cả các đường đi có trong chương trình, và cụm từ này chỉ ra cách diễn đạt định lượng cho phương pháp bao phủ đã được mô tả về kiểm thử hộp trắng.

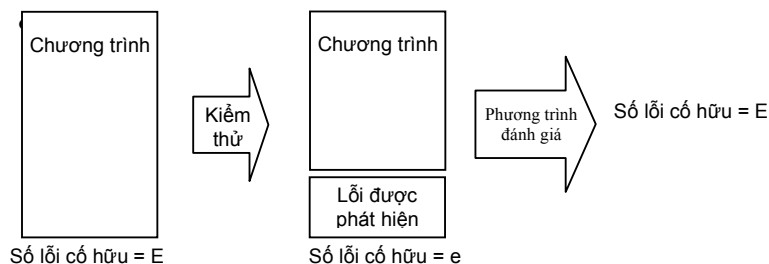
(4) Thiết kế kiểm thử

- Người ta đã biết rằng phương pháp kiểm thử bao gồm kiểm thử hộp trắng và kiểm thử hộp đen và có vài phương pháp để thiết kế các trường hợp kiểm thử. Việc dùng các trường hợp thiết kế được sinh ra chỉ với một trong hai phương pháp này là không đủ. Nói chung, tiến hành kiểm thử dựa trên kiểm thử hộp đen là hiệu quả cho kiểm thử chức năng, và bổ sung vào chúng với kiểm thử hộp trắng.
- Các điều kiện biên cho cái vào và cái ra và các điều kiện chuyển đổi được kiểm tra dựa trên tài liệu thiết kế ngoài cho chương trình. Sau đó các trường hợp kiểm thử được sinh ra. Các trường hợp kiểm thử bao giờ cũng nên được giới hạn vào những trường hợp kiểm tra liệu chức năng cần thiết (cái gì) có được đạt tới không.
- Dựa trên danh sách nguồn chương trình và các đặc tả chi tiết (liên quan tới các thuật toán được mô tả trên cơ sở câu lệnh, như đặc tả chương trình và đặc tả mô đun), các hình mẫu phân nhánh của chương trình được kiểm tra. Sau đó các trường hợp kiểm thử tạo khả năng cho các điều kiện "đúng" và "sai" của mỗi chỗ có kiểm tra điều kiện được thực hiện và tạo ra.
- Kiểm tra rằng các trường hợp kiểm thử ① và ② được mô tả ở trên có thể thực hiện mọi lệnh được bao hàm. Với các chu trình, các trường hợp kiểm thử điều kiện ra khỏi chu trình 0, 1, và số lần tối đa sẽ được tạo ra.
- Sau khi tạo ra các trường hợp kiểm thử cho ①, ② và ③, kết quả kiểm thử cho từng trường hợp kiểm thử về mặt lý thuyết được giả thiết làm sinh ra tài liệu điều kiện kiểm thử.

(5) Len lỏi

- Không thể nào khử bỏ hết mọi lỗi trong phần mềm. Điều này là vì không thể nào kiểm chứng được rằng không có lỗi còn lại. Dijkstra, người đề nghị lập trình có cấu trúc, đã nói:
- "Kiểm thử chỉ ra rằng lỗi tồn tại, nhưng không thể chứng minh được rằng không còn lỗi nữa."
- Myers, người đề nghị thiết kế có cấu trúc, phải nói điều này về kiểm thử:
- "Kiểm thử là tiến trình kiểm tra rằng chương trình là đúng, hay nói cách khác, là không chỉ ra rằng chương trình không có lỗi."
- "Kiểm thử là tiến trình thực hiện chương trình để phát hiện ra lỗi."
- Rút cục, không ai có thể nói rằng, "chương trình là không có lỗi" ngay cả sau khi đã tiến hành các phép kiểm thử. Tuy nhiên, thời hạn chuyển giao được xác định cho việc phát triển hệ thống. Do đó, cần phải hoàn thành việc phát triển hệ thống và bắt đầu vận hành thực tế. Hậu quả là, sẽ đến lúc người thiết kế bị buộc phải tuyên bố rằng hoạt động kiểm thử đã được hoàn tất.
- Việc len lỏi là một phương pháp để đánh giá ngày hoàn thành hoạt động kiểm thử về mặt con số; các lỗi đã biết được len vào một cách có chủ ý, và số lỗi cố hữu vẫn còn sẽ được ước lượng dựa trên tỉ số của lỗi cố hữu đã được phát hiện với số lỗi đã biết được phát hiện.

Hình 1-5-20
Len lỏi



•

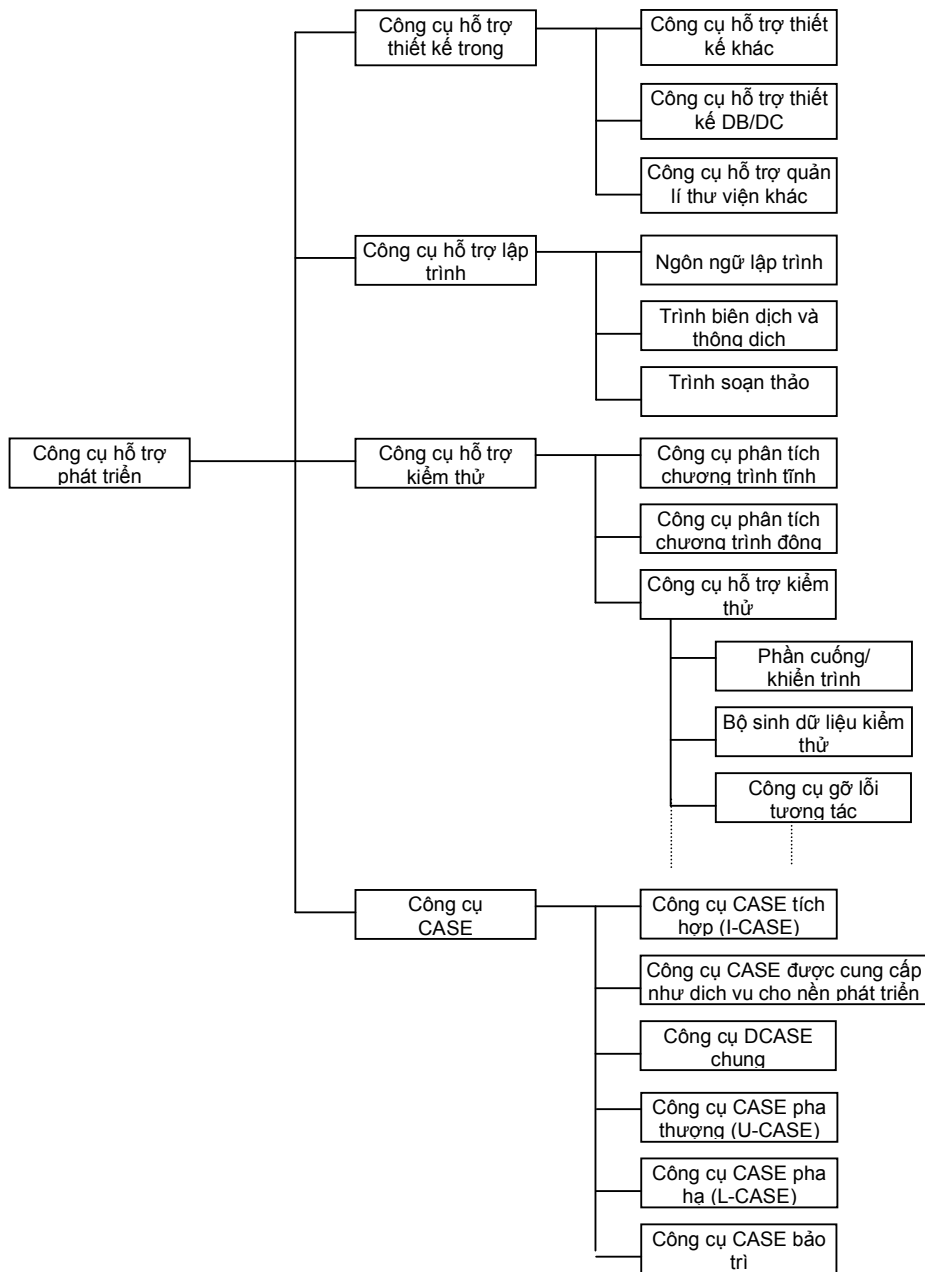
1.6 Môi trường phát triển

- Nhu cầu của các xí nghiệp đã ngày một đa dạng và nâng cao. Tình huống thị trường thay đổi nhanh chóng. Các công ti sẽ bị đe dọa nếu họ không đáp ứng nhanh chóng cũng như nhất quán với những thay đổi này. Do đó, hệ thông tin, hỗ trợ cho mọi khía cạnh hoạt động kinh doanh, phải được phát triển trong thời kì ngắn, và phải cao về chất lượng. Nếu việc phát triển hệ thống cần nhiều năm, thì tình huống bao quanh hệ thống sẽ trở nên hoàn toàn khác khi nó được hoàn tất, và nhiều đầu tư vào việc phát triển bị phí hoài. Do đó, những thời kì phát triển dài như vậy là không được phép đối với các hệ thống qui mô lớn, có tầm quan trọng lớn.
- Trong mục này chúng ta sẽ thảo luận về các môi trường phát triển, phần lớn nhằm vào "làm tăng năng suất phát triển". Nếu công việc được tự động hoá nhiều nhất có thể được và các cấu phần được dùng đến mức tối đa có thể được, như trong sản xuất các sản phẩm công nghiệp, thì các sản phẩm chất lượng cao và đều nhau sẽ được sinh ra.

1.6.1 Công cụ hỗ trợ phát triển

- Các kiểu công cụ hỗ trợ phát triển hệ thống hiện có sẵn được nêu trong Hình 1-6-1.
- Các công cụ phát triển bao gồm mọi thứ từ trình soạn thảo văn bản và ngôn ngữ lập trình cả hai đều đã được dùng trong các công cụ CASE được mô tả sau đây. Có nhiều tùy chọn, và chúng có thể làm tăng thực sự về năng suất.
- Dưới đây, các kiểu, chức năng và đặc trưng của công cụ hỗ trợ phát triển sẽ được mô tả.

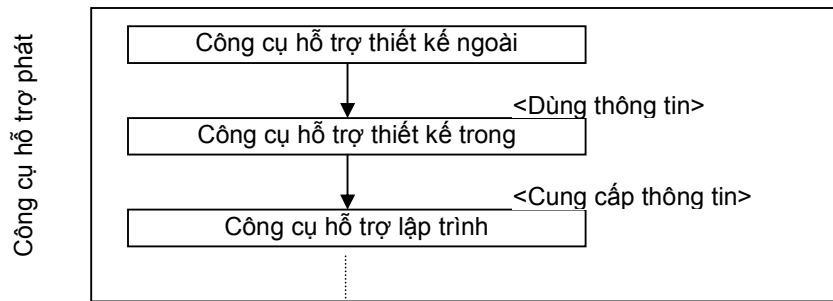
Hình 1-6-1 Phân loại các công cụ hỗ trợ phát triển



(1) Công cụ hỗ trợ thiết kế trong

Các công cụ hỗ trợ thiết kế trong, được cung cấp chủ yếu như một phần của hệ thống hỗ trợ phát triển tích hợp. Vì việc phát triển bằng cách dùng mô hình thác đổ được tiến hành theo phương pháp trên xuống, nên công việc tiến trình về sau được thực hiện với việc dùng thông tin được đăng kí và sinh ra theo công cụ hỗ trợ thiết kế trong.

Hình 1-6-2 Vị trí của công cụ hỗ trợ thiết kế trong



Công cụ hỗ trợ thiết kế trong được dùng chủ yếu để hỗ trợ cho công việc sau.

① Hỗ trợ thiết kế các loại

- Phân hoạch chức năng, và thiết kế có cấu trúc
- Thiết kế màn hình
- Thiết kế (tệp) dữ liệu vật lý
- Thiết kế các tài liệu đa dạng
- Thiết kế thông báo

② Hỗ trợ thiết kế DB/DC (DataBase/Data communication - Cơ sở dữ liệu/truyền thông dữ liệu)

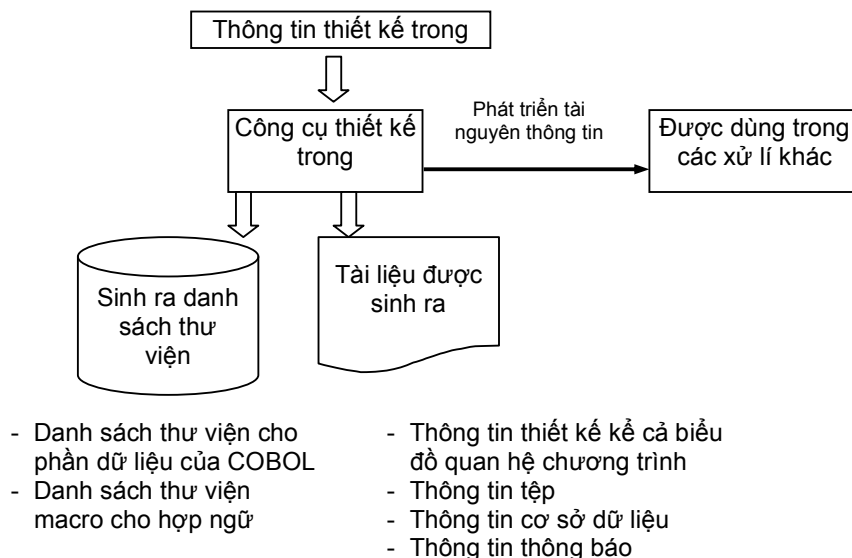
- Thiết kế DB
- Thiết kế DC (kể cả các thông báo giữa các thiết bị cuối và máy chủ trong hệ thống trực tuyến)

③ Quản lý các thư viện

- Quản lý thư viện các phần dữ liệu của COBOL
- Quản lý các thư viện macro của hợp ngữ

Hình 1-6-3 chỉ ra mối quan hệ giữa cái vào của một công cụ hỗ trợ thiết kế, việc thực hiện của công cụ, và cái ra từ công cụ này.

Hình 1-6-3 Cái vào của công cụ hỗ trợ thiết kế trong và cái ra từ công cụ này



(2) Công cụ hỗ trợ lập trình

- Công cụ hỗ trợ lập trình bao gồm phần mềm như các chương trình dịch vụ được dùng trong lập trình. Nó cũng bao gồm:

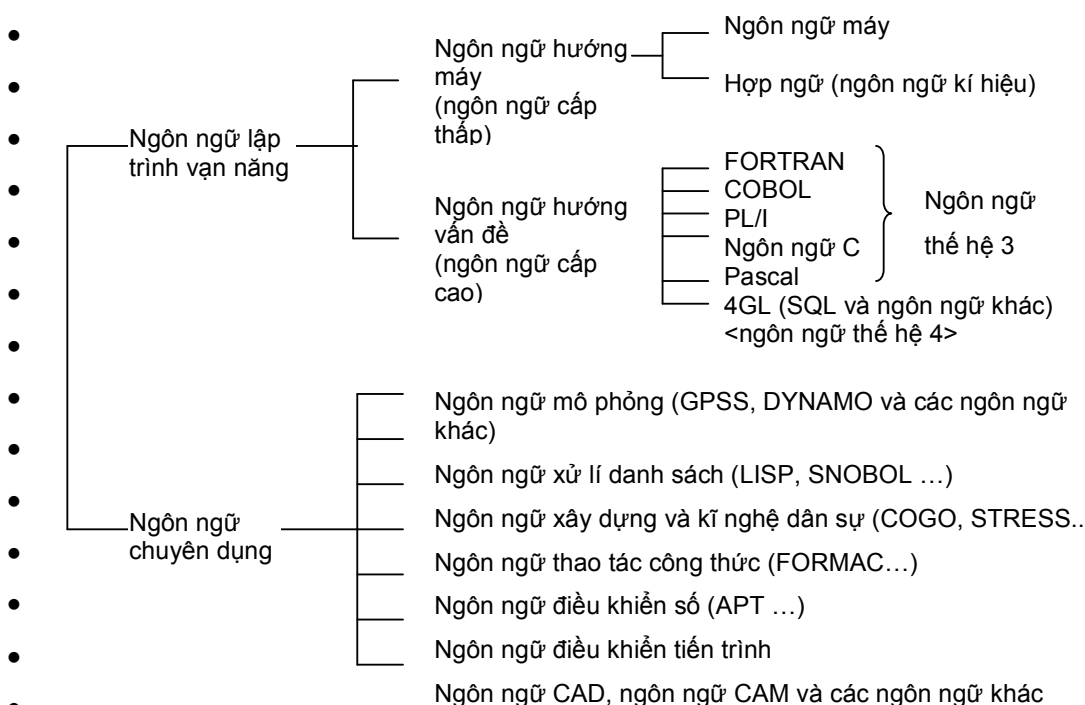
- - Công cụ lập trình
- - Các bộ xử lý ngôn ngữ
- - Trình soạn thảo
- Dưới đây sẽ cung cấp các chi tiết về từng khoản mục này.

① Ngôn ngữ lập trình

- Ngôn ngữ lập trình có thể được phân loại như trong Hình 1-6-4.

- Gần đây, các ngôn ngữ phi thủ tục, điển hình là ngôn ngữ thể hệ bốn (4GL) và kể cả SQL, đã được dùng thường xuyên. Điều này là vì 4GL cung cấp năng suất cao hơn 3GL (ngôn ngữ thể hệ ba).

Hình 1-6-4 Các kiểu của ngôn ngữ lập trình



② Bộ xử lý ngôn ngữ

a. Trình biên dịch - Compiler

- Trình biên dịch, bộ xử lý ngôn ngữ được dùng trong các ngôn ngữ biên dịch như COBOL, dịch toàn bộ chương trình gốc, xem như một toàn thể, thành các đối tượng (các chương trình ngôn ngữ máy).

b. Trình thông dịch - Interpreter

- Trình thông dịch, bộ xử lý ngôn ngữ được dùng trong các ngôn ngữ thông dịch như BASIC, thực hiện các lệnh nguồn theo từng dòng một. Nó cho phép dễ dàng gỡ lỗi, bởi vì ngay cả mã của chương trình không đầy đủ cũng có thể được thực hiện.

-

- ③ Trình soạn thảo

- Trình soạn thảo là phần mềm để sinh ra và soạn thảo các chương trình. Trình soạn thảo bao gồm trình soạn thảo văn bản và trình soạn thảo cấu trúc.

- a. Trình soạn thảo văn bản

- Trình soạn thảo văn bản được dùng để sinh ra và sửa đổi chương trình.

- b. Trình soạn thảo cấu trúc

- Trình soạn thảo cấu trúc, một trình soạn thảo làm cho chương trình được sửa đổi, với cấu trúc của chúng được làm rõ ràng, cho phép nhiều đặc tả khác nhau bên cạnh chương trình được soạn thảo.

(3) Công cụ hỗ trợ kiểm thử

- Mục đích chính của kiểm thử là loại bỏ lỗi, và nỗ lực tốt nhất phải được thực hiện để đạt tới mục đích. Tuy nhiên, công việc sinh ra dữ liệu kiểm thử lại đòi rất nhiều lao động. Tốn nhiều thời gian vào những việc như vậy là không khôn ngoan.

- Để loại bỏ lỗi một cách có hiệu quả cũng như để làm cho từng phép kiểm thử cân bằng và không thiên vị, cần dùng công cụ nhiều nhất có thể được.

- Có các kiểu công cụ hỗ trợ kiểm thử khác nhau. Chúng đại thể được phân loại thành hai kiểu sau:

- - Công cụ hỗ trợ kiểm thử chương trình

- - Công cụ hỗ trợ thực hiện kiểm thử

- ① Công cụ hỗ trợ kiểm thử chương trình

- Công cụ hỗ trợ kiểm thử chương trình bao gồm công cụ phân tích chương trình tĩnh được dùng để phân tích chương trình mà không thực hiện chúng và công cụ phân tích chương trình động được dùng để phân tích chương trình trong khi thực hiện chúng.

- ② Công cụ hỗ trợ thực hiện kiểm thử

- Công cụ hỗ trợ thực hiện kiểm thử bao gồm các công cụ cung cấp môi trường cho các mô đun cần cần tới trong kiểm thử mớic nói, và các bộ sinh dữ liệu kiểm thử mà với tham biến được cho, sẽ sinh ra dữ liệu một cách tự động.

Hình 1-6-5 Các công cụ hỗ trợ kiểm thử khác nhau

Công cụ			Chức năng	
Công cụ hỗ trợ kiểm thử chương trình	Công cụ phân tích chương trình tĩnh	Công cụ phân tích mã		Được dùng để tăng tính dễ đọc của chương trình nguồn.
		Công cụ phân tích cấu trúc		Được dùng để sinh ra trường hợp kiểm thử.
		Công cụ kiểm giao diện mô đun		Được dùng để kiểm giao diện giữa các mô đun.
	Công cụ phân tích chương trình động	Công cụ điều phối		Các luồng điều khiển được theo dõi. Sau đó, dữ liệu trên số thời gian mỗi đường dẫn được sử dụng và thời gian thực hiện được cung cấp tới các tệp.
		Bộ mô phỏng trực tuyến		
		Công cụ ghi nhật kí hệ thống		Tỉ lệ dùng tài nguyên được đưa ra tệp.
		Công cụ sinh và thực hiện dữ liệu kiểm thử tự động		Được dùng để sinh ra dữ liệu kiểm thử và thực hiện chương trình.
	Công cụ hỗ trợ thực hiện kiểm thử	Công cụ cuống/driver		Được dùng để thiết lập môi trường nơi các mô đun và chương trình được cho chạy.
Bộ sinh dữ liệu kiểm thử		Được dùng để sinh ra dữ liệu kiểm thử chương trình một cách tự động.		
Công cụ gỡ lỗi tương tác		Chương trình nguồn được hiển thị trên màn hình, việc thực hiện được tiến hành theo kiểu tương tác.		
Chương trình tiện ích (dịch vụ)		Xổ	Dữ liệu trong đơn vị bộ nhớ chính và thanh ghi được đưa ra thiết bị ra và thiết bị bộ nhớ.	
		Bộ theo dõi vết	Trạng thái thực hiện của chương trình được theo dõi liên tiếp để thu dữ liệu cần thiết.	

(4) Công cụ CASE

CASE là viết tắt của Computer Aided Software Engineering - kỹ nghệ phần mềm có máy tính hỗ trợ.

Công cụ CASE chỉ ra các công cụ hỗ trợ phát triển nhằm tới công việc phát triển phần mềm tự động hay để làm tăng hiệu quả của công việc, và dùng các kỹ thuật phát triển khác nhau được tổ hợp với thiết kế chương trình có cấu trúc, công nghệ kỹ nghệ phần mềm, làm lỗi.

① Các đặc trưng của công cụ CASE

Công cụ CASE bao gồm những đặc trưng sau:

Công cụ CASE được phát triển với mục đích tập trung quản lý việc phát triển hệ thống và tự động hoá các công việc đa dạng. Do đó, trong công cụ CASE, các dữ liệu khác nhau về việc phát triển hệ thống được lưu giữ trong cơ sở dữ liệu thông tin phát triển hệ thống gọi là "kho chứa." Nói cách khác, dữ liệu được lưu giữ trong kho chứa được dùng vào bất kỳ lúc nào để tập trung việc phát

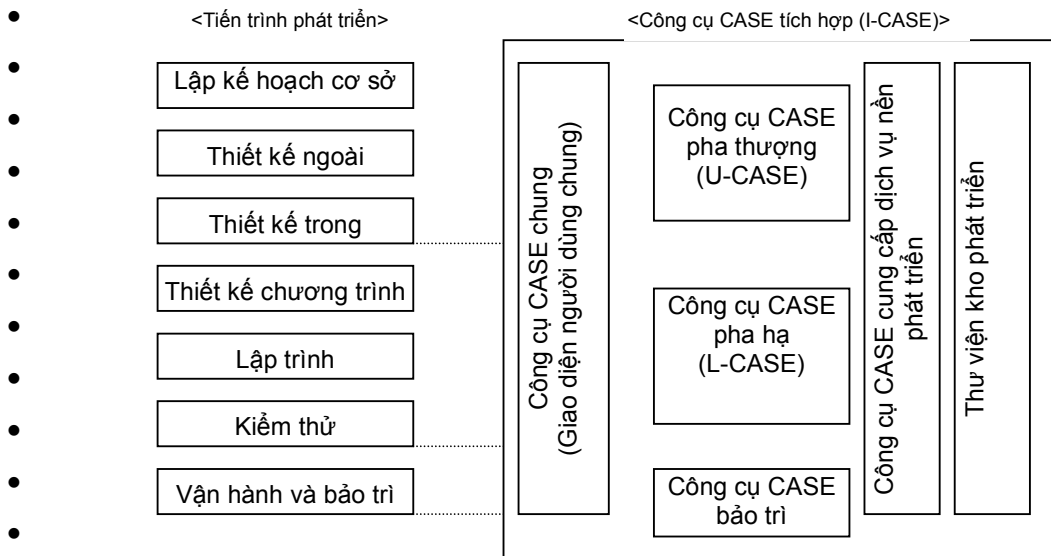
triển hệ thống và tự động hoá nhiều công việc.

- Các công cụ CASE khác nhau được dùng cùng nhau để bao quát toàn bộ các tiến trình của việc phát triển hệ thống.
- Giao diện người dùng được thống nhất để làm công việc dễ dàng hơn.
- Công cụ này cung cấp dữ liệu, không chỉ cho công việc phát triển như phân tích và thiết kế, mà tất cả các dữ liệu về phát triển hệ thống, kể cả dữ liệu về quản lý chất lượng và quản lý dự án.

② Các kiểu công cụ CASE

- Công cụ CASE bao gồm các công cụ được nêu trong Hình 1-6-6.

Hình 1-6-6 Các kiểu công cụ CASE



a. Công cụ CASE cho pha thượng (U-CASE)

- Công cụ CASE cho pha thượng bao gồm các pha lập kế hoạch cho tới pha thiết kế trong. Với phương pháp có cấu trúc như DFD, công cụ này hỗ trợ cho các loại công việc thiết kế bao gồm các công việc cho mô hình hoá, xác định yêu cầu, cơ sở dữ liệu và mạng.

b. Công cụ CASE cho pha hạ (L-CASE)

- Công cụ CASE cho pha hạ bao quát pha thiết kế chương trình cho tới pha thiết kế kiểm thử. Nó hỗ trợ việc lập trình tự động và tự động hoá các phép kiểm thử.

c. Công cụ CASE bảo trì

- Công cụ CASE bảo trì hỗ trợ cho việc sửa đổi và sửa chữa hệ thống, và công việc bảo trì.

d. Công cụ CASE chung (Giao diện người dùng chung)

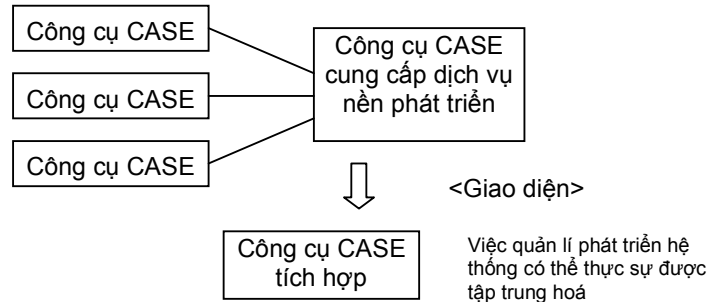
- Công cụ CASE chung hỗ trợ cho công việc quản lý bao quát toàn bộ tiến trình phát triển hệ thống, bao gồm quản lý tài liệu và quản lý dự án.

e. Công cụ CASE cung cấp các dịch vụ nền phát triển

- Công cụ CASE cung cấp các dịch vụ nền phát triển đưa ra giao diện giữa công cụ CASE và việc hỗ trợ cho việc tích hợp của các công cụ.

- f. Công cụ CASE tích hợp (I-CASE)
- Công cụ CASE tích hợp, trích ra những phần đáng chú ý nhất trong số các công cụ CASE, tích hợp các chức năng do công cụ CASE cung cấp trong từng pha từ pha thượng tới pha bảo trì, cung cấp sự hỗ trợ cho mọi pha của vòng đời hệ thống một cách nhất quán.
- RAD (Rapid Application Development - Phát triển ứng dụng nhanh) là phương pháp tạo khả năng cho việc phát triển hệ thống có hiệu quả bằng cách buộc sự tham dự của người dùng ngay từ các pha đầu tiên trong việc phát triển hệ thống và bằng việc tự động hoá công việc phát triển bằng việc dùng đầy đủ công cụ CASE tích hợp.

Hình 1-6-7 Công cụ CASE tích hợp



1.6.2 EUC và EUD

- Một cách hình thức, việc phát triển hệ thống là công việc do bộ phận hệ thống tin thực hiện. Tuy nhiên, khi khối lượng bảo trì hệ thống hiện có làm tăng lượng việc tồn đọng lại thì bộ phận này không thể đảm đương việc sẵn sàng giải quyết các dự án phát triển mới. Thêm vào đó, người dùng cuối bây giờ muốn dùng dữ liệu một cách tự do. Trong những tình huống như vậy, khái niệm về EUC (End User Computing - Tính toán người dùng cuối) và EUD (End User Development - Phát triển người dùng cuối), chỉ ra rằng bản thân người dùng cuối phát triển hệ thống riêng của mình, đã thu được sự ủng hộ. Đằng sau những lí do làm cho EUC và EUD thành có thể, tồn tại sự kiện là máy tính cá nhân đã được dùng rộng rãi và các chức năng của chúng cũng đã nâng cao, và rằng các gói phần mềm thân thiện người dùng, như trang tính và phần mềm cơ sở dữ liệu đơn giản, đã trở nên sẵn có.

(1) Dùng phần mềm trang tính

- Phần mềm trang tính đã được phát triển để dùng trong việc kết tập dữ liệu và sinh ra các bảng. Phần mềm này cũng tạo khả năng cho dữ liệu được kết tập được biểu diễn dưới dạng thấy được như đồ hoạ. Với những chức năng này được dùng như công cụ nghiệp vụ, có thể phát sinh ra các hệ thống nghiệp vụ đơn giản. Thêm vào đó, phần mềm trang tính được cung cấp cùng các chức năng để lấy dữ liệu từ hệ thống xương sống. Các chức năng này cho phép người dùng lấy các dữ liệu cần thiết từ hệ thống xương sống và thao tác nó một cách tự do.
- Gần đây, GUI đã được hỗ trợ trong các giao diện người dùng, cung cấp môi trường phát triển tốt hơn và các khả năng thao tác tốt hơn.

(2) Dùng phần mềm cơ sở dữ liệu

- Phần mềm cơ sở dữ liệu lớn như phần mềm được dùng trong các hệ xương sống là không cần cho EUC. Tuy nhiên, phần mềm cơ sở dữ liệu có thể lấy dữ liệu từ hệ thống xương sống. Do đó,

phần mềm này có khả năng lấy chỉ những dữ liệu cần thiết từ hệ thống xương sống, rồi thao tác và phân tích nó. Nói chung, dữ liệu được lấy ra và thao tác trong hệ thống phòng ban thì không được ghi lại vào hệ thống xương sống. Không cần phải nói, có thể sinh ra cơ sở dữ liệu bằng việc đưa vào dữ liệu riêng của phòng ban và dùng nó.

- ① Chuẩn hoá cơ sở dữ liệu

- Thiết kế cơ sở dữ liệu là quan trọng trong việc tạo ra cơ sở dữ liệu. Việc chuẩn hoá cơ sở dữ liệu là công việc nặng nề, nhưng một số phần mềm cơ sở dữ liệu được cung cấp với các chức năng chuẩn hoá cơ sở dữ liệu tự động khi dữ liệu cần thiết được đưa vào.

- ② Thao tác dữ liệu

- Dữ liệu được thao tác một cách tương tác bằng việc dùng định dạng chỉ lệnh hay định dạng menu, còn các câu lệnh SQL được tự động sinh ra. Do đó, việc làm cách sử dụng tương tự mô phỏng, như phân tích, là hoàn toàn dễ dàng, không đòi hỏi tri thức đặc biệt. Khi kĩ năng được cải tiến, việc sử dụng được nâng cao hơn, như mã hoá và thực hiện các câu lệnh SQL, cũng trở thành có thể.

1.7 Quản lý phát triển

1.7.1 Lập kế hoạch dự án

• Để hoàn thành thành công một dự án phát triển, thì việc lập kế hoạch trước, và quản lý dựa trên các kế hoạch đó là quan trọng. Việc chuẩn bị các kế hoạch giúp làm cái nhìn toàn cảnh về dự án phát triển được rõ ràng, tạo khả năng kiểm tra và nghiên cứu trước về các vấn đề và rủi ro của dự án, bên cạnh việc xem xét về mục đích, chức năng và bao quát về hệ thống, cũng như khối lượng công việc và số nhân công cần cho dự án.

(1) Lập kế hoạch dự án

- Các kế hoạch dự án được tạo ra cho các khoản mục sau:
 - - Cái ra, công việc, lịch biểu, chất lượng, rủi ro và những cái khác
 - - Người phát triển
 - • Tổ chức, bồi dưỡng nhân lực, phương tiện trao đổi và những thứ khác
 - - Mua sắm bên ngoài
 - • Nơi mua sắm, cách mua, thời gian giao hàng, chất lượng (hay kỹ năng) và những thứ khác
 - - Môi trường phát triển
 - • Phần cứng, phần mềm và các thứ khác
 - - Chi phí phát triển
 - • Chi phí nhân sự, chi phí trang thiết bị, chi tiêu và các chi khác
- Các kế hoạch chi tiết được tạo ra cho những khoản mục này. Những kế hoạch này cung cấp cơ sở để đánh giá xem liệu dự án có nên được triển khai hay không. Bên cạnh đó, chúng cũng được dùng làm đích cho việc quản lý sau khi dự án được chấp thuận.

(2) Tính sinh lợi của dự án

- Vì việc phát triển hệ thống được thực hiện như một phần của hoạt động kinh doanh, nên một cách tự nhiên tính sinh lợi cần được thăm dò sau khi phát triển. Điều này có nghĩa là việc đánh giá chi phí-hiệu quả là cần thiết. Nếu việc phát triển hệ thống được biết là không sinh lợi từ pha lập kế hoạch, thì dự án này sẽ không được chấp thuận mà không có lý do thỏa đáng, chẳng hạn như, hệ thống là cần thiết để đáp ứng các yêu cầu của luật pháp hay qui chế.
- Cho dù tính sinh lợi có được trông đợi tại pha lập kế hoạch, thì điều thường xảy ra là vào lúc hoàn thành người ta thấy dự án này hoàn toàn chẳng lợi lộc gì. Chẳng hạn, ta có thể xét kịch bản sau đây: Khách hàng thay đổi yêu cầu vào pha thiết kế, làm tăng khối lượng công việc cần thiết. Các chức năng được giả thiết trước đây cho môi trường phát triển thực tế không được hỗ trợ thỏa đáng. Những thứ mua từ bên ngoài lại kém về chất lượng, làm phát sinh nhiều việc phải làm lại và ảnh hưởng tới thời hạn giao hàng. Một số kịch bản chỉ nêu ra khi nào các hành động được tiến hành. Có thể nói rằng các dự án đó rất có thể là không được xử lý như đã được lập kế hoạch.
- Do đó, việc xem xét lại các kế hoạch trong từng pha trở thành cần thiết, để duy trì tính sinh lợi. Nói riêng, những sửa đổi về các đặc tả phải được phản ánh đúng. Cho nên, để an toàn chất lượng, điều quan trọng là tiến hành việc quản lý thay đổi.

•

1.7.2 Lập kế hoạch, quản lí và đánh giá chất lượng

(1) QFD (Quality Function Deployment - Triển khai chức năng chất lượng)

- Việc kiểm thử phần mềm chỉ để loại bỏ các khiếm khuyết và để duy trì chất lượng đã được thiết kế. Mặt khác, QFD là một công nghệ để sinh ra phần mềm chất lượng cao hơn. QFD được phát minh ra để làm tăng chất lượng của thiết kế phần cứng. Đại cương về QFD là như sau: với QFD, chất lượng của phần mềm bao gồm các hệ con và các mô đun được biểu diễn như đặc trưng chất lượng cho phép đánh giá rõ ràng và cùng phương pháp đánh giá đó được dùng một cách hệ thống cho chất lượng của các hệ con và các mô đun nữa.

(2) Chất lượng phần mềm

- Như chúng ta đều đã biết rõ, máy tính không mềm dẻo. Nói rằng "Máy tính khôn lắm" là không đúng. Đúng hơn phải nói là, "Phần mềm (chương trình) này khôn lắm." Tuy nhiên, chất lượng của phần mềm không bao giờ được cải tiến lên mà không tạo ra phần mềm tốt hơn. Trong máy tính vạn năng, việc đưa dữ liệu kí tự vào thay vì các khoản mục số làm cho máy tính bị kết thúc bất thường. Để tránh điều này, với một chương trình có thể kiểm tra dữ liệu được đưa vào và yêu cầu rằng dữ liệu đó phải được đưa vào lại nếu dữ liệu đầu không phải là số. Nói cách khác, việc đưa ra một chức năng kiểm tra lỗi làm cho chương trình khôn hơn, làm tăng chất lượng phần mềm.

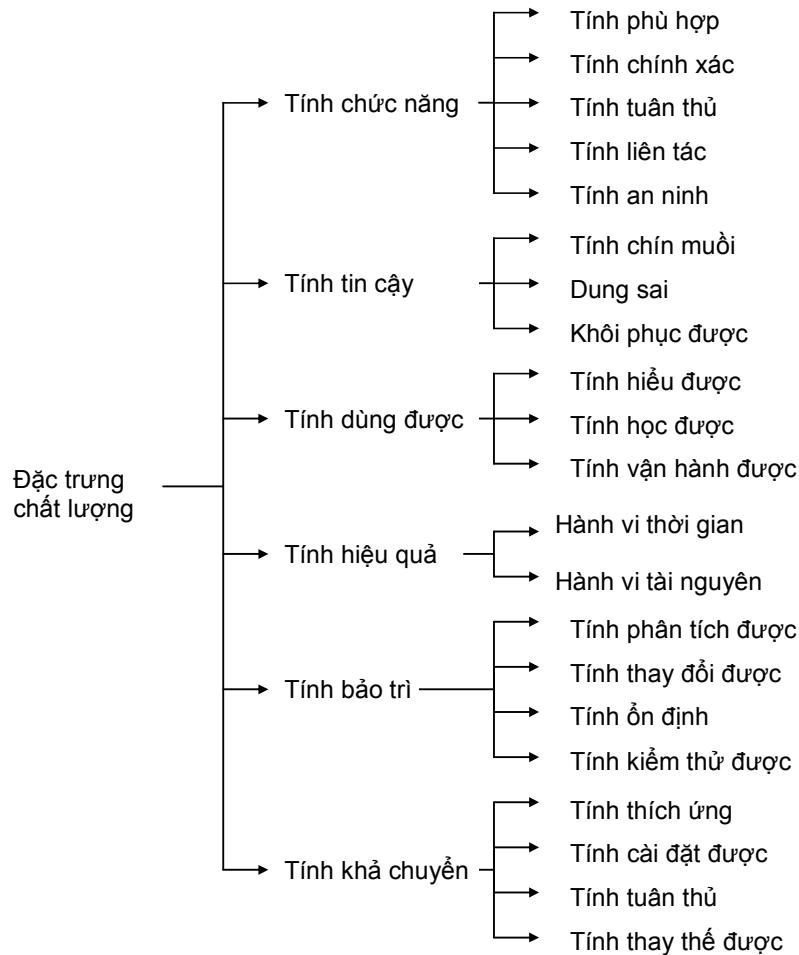
- Một chức năng như vậy không nên phụ thuộc vào khả năng của người lập trình. Nó phải được tính tới mà không bị bỏ quên mất trong pha thiết kế nơi các yêu cầu người dùng được cụ thể hoá chi tiết. Nếu sự cần thiết của chức năng kiểm tra được xác nhận như một nhu cầu tiềm tàng của người dùng, nhưng lại không được mô tả trong đặc tả trong pha thiết kế, thì chức năng này sẽ không bao giờ được đạt tới. Những khiếm khuyết bắt nguồn từ các định nghĩa yêu cầu trong pha sớm nhất có thể chỉ được phát hiện ra trong kiểm thử vận hành, pha muộn nhất. Hậu quả là việc sửa chữa rất có thể là điều không thể nào thực hiện được, ngay cả sau khi đã tìm ra khiếm khuyết.

- Nói cách khác, nếu được cấp đủ thời gian trong các giai đoạn đầu cho việc kiểm chứng và sửa chữa, thì khối lượng sửa đổi được cần tới trong các pha cuối sẽ được rút bớt lại. Thực tế, người ta nói, "Một giờ được dùng để ngăn cản các khiếm khuyết trong những pha đầu xoá bỏ đi ba tới mười giờ làm việc sửa chữa trong pha cuối." Để làm tăng chất lượng phần mềm, việc thực hiện cái gọi là chu trình Kế hoạch-Thực hiện-Kiểm tra-Hành động Plan-Do-Check-Action (PDCA) -- lập kế hoạch, thực hiện kế hoạch, đánh giá cái ra và chọn hành động dựa trên các đánh giá - là được cần tới.

(3) Đặc trưng chất lượng phần mềm

- Có nhiều phương pháp đánh giá phần mềm. Ở đây, chúng ta mô tả sáu đặc trưng chất lượng, được liệt kê trong ISO/IEC 9126 được nêu trong Hình 1-7-1.

Hình 1-7-1 Đặc trưng chất lượng của ISO/IEC 9126



① Chức năng (đặc trưng chức năng)

- Các chức năng cần cho hệ thống được thực hiện (tính thích hợp)
- Độ chính xác chức năng được cung cấp (tính chính xác)
- Các chức năng đáp ứng cho đặc tả (tính tuân thủ)
- Cung cấp sự dễ dàng nối với các hệ thống khác (tính liên tác)
- Cung cấp tính an ninh bản chất (an ninh)

② Tính tin cậy (đặc trưng tin cậy)

Phần mềm không có lỗi: chín muồi.

Một mức độ hệ thống nào đó được duy trì ngay cả khi xuất hiện trục trặc: dung sai.

Hoạt động bình thường được khôi phục sẵn sàng khi lỗi xuất hiện: tính khôi phục được.

③ Tính dùng được (đặc trưng dễ dùng)

- Vận hành dễ dàng: tính hiểu được.
- Dễ nhớ: khả năng học.

- Cho phép quản lý thao tác dễ dàng: tính vận hành.
- ④ Tính hiệu quả (đặc trưng hiệu năng)
 - Cung cấp những đáp ứng tốt và hiệu năng cao: hành vi thời gian.
 - Cho phép dùng hiệu quả các tài nguyên hệ thống: hành vi tài nguyên.
- ⑤ Tính bảo trì được (đặc trưng bảo trì)
 - Cho phép phân tích dễ dàng các tài liệu thiết kế và chương trình khi tìm ra lỗi: khả năng phân tích.
 - Cho phép mở rộng và sửa đổi dễ dàng cho hệ thống: tính thay đổi được.
 - Việc sửa đổi hệ thống không ảnh hưởng tới các hệ thống khác: tính ổn định.
 - Không đòi hỏi kiểm thử mất công sức sau khi tiến hành sửa đổi: tính kiểm thử được.
- ⑥ Tính khả chuyển (đặc trưng của việc chuyển chương trình sang các máy tính khác)
 - Có tính thích ứng: tính thích ứng.
 - Cung cấp công việc thiết đặt dễ dàng: khả năng thiết đặt.
 - Tuân thủ các đặc tả chuyển: tính tuân thủ.
 - Cho phép dễ dàng thay thế bằng phần mềm khác: khả năng thay thế.

1.7.3 Quản lý tiến trình

- Quản lý tiến trình được chia thành lập kế hoạch tiến độ và quản lý tiến trình. Ở đây, các đặc trưng của từng việc, và phương pháp được dùng cho chúng sẽ được giải thích.

(1) Đại cương về việc lập kế hoạch tiến trình và quản lý tiến độ

- ① Lập kế hoạch tiến trình
 - Việc phát triển hệ thống được hoàn thành qua nhiều tiến trình khác nhau. Một số công việc lớn phải mất nhiều năm để hoàn thành. Do đó, việc lập kế hoạch tiến độ chính xác là một khoản mục quản lý công việc quan trọng.
 - Nói riêng, các dự án phát triển mới bao gồm nhiều nhân tố không chắc chắn mà không thể nào được xác định dứt khoát trong pha lập kế hoạch tiến độ. Do đó, khi tiến độ công việc tiến lên, việc giải quyết linh hoạt cho các tình huống, như làm cho ngày chuyển giao sớm hơn (tùy theo tình huống) hay tối thiểu hoá chậm trễ, trở thành cần thiết.
- ② Quản lý tiến trình
 - Quản lý tiến trình là việc quản lý sự diễn biến của công việc. Nó cần kiểm lại tiến độ công việc và tiến hành hành động nào đó đối với công việc có tiến độ bị chậm so với lịch biểu. Hiệu năng công việc phát triển hệ thống bị chia sẻ cho nhiều người. Do đó, sự chậm trễ của người này trong công việc dẫn tới sự chậm trễ trong tiến độ của dự án xem như một tổng thể. Hậu quả là, việc quản lý tiến trình thấu đáo là cần để làm tối thiểu tác động của chậm trễ cũng như để phát hiện ra vấn đề dễ nhất có thể được.

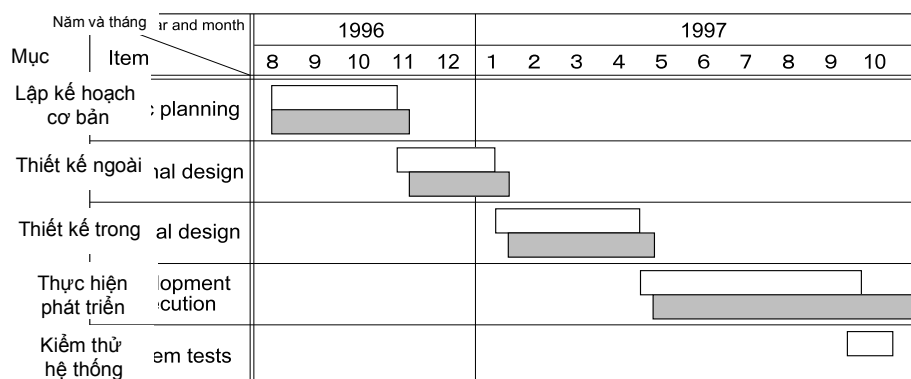
(2) Lập kế hoạch tiến trình

- Sơ đồ Gantt và PERT được dùng điển hình như các phương pháp lập kế hoạch tiến độ.

① Sơ đồ Gantt

- Sơ đồ Gantt cũng còn được gọi là "sơ đồ thanh."

Hình 1-7-2 Một ví dụ về sơ đồ Gantt



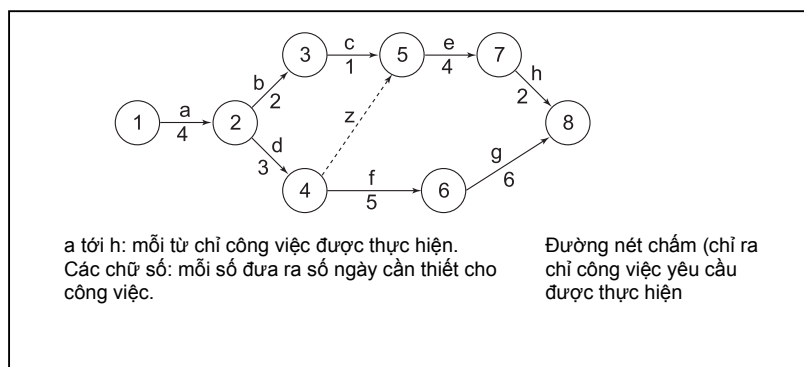
Kế hoạch Planned Thực hiện

- <Đặc trưng>
- Nó cung cấp các biểu đồ dễ hiểu theo đó lịch biểu của từng việc được chỉ ra bằng đường ngang (thanh).
- Trong sơ đồ này, thời gian bắt đầu và kết thúc theo lịch của từng phần việc và trạng thái hiện tại của chúng được biểu thị rõ ràng.
- Ưu tiên của các phần việc không được vẽ ra.
- Các mức độ theo đó sự chậm trễ trong từng phần việc ảnh hưởng tới các công việc khác không được vẽ ra.
- ② PERT (Program Evaluation and Review Technique - Kỹ thuật kiểm điểm và đánh giá chương trình)
- PERT cung cấp một kỹ thuật để sinh ra lịch biểu cho phần công việc (tiến độ) của một dự án, và thể rồi để quản lý chúng.

Hình 1-7-3 Ví dụ về biểu đồ PERT (biểu đồ thứ nhất của hai biểu đồ)

Hình 1-7-3

Ví dụ về biểu đồ PERT
(Hình thứ nhất của biểu đồ)



- <Đặc trưng>
- PERT có thể giải quyết việc phát triển các hệ thống qui mô lớn và phức tạp.
- Nó tạo khả năng cho việc tính toán tổng số ngày cần thiết (thời kì tối thiểu cần thiết).
- Thứ tự công việc cần được thực hiện được làm rõ ràng, tạo khả năng làm sáng tỏ các điểm quản lí quan trọng.
- Số ngày được bao gồm trong từng mục như lễ co dẫn được dễ dàng tính ra.
- Nó có thể được áp dụng cho các tính toán làm giảm chi phí phát triển và làm giảm số ngày cần cho công việc (dựa trên phương pháp đường găng CPM (Critical Path Method) hay các phương pháp khác).
- <Thủ tục>
 1. Xác định số ngày cần cho từng phần việc. Rồi kết quả được gắn lại với nhau trong bảng như trong Hình 1-7-4.

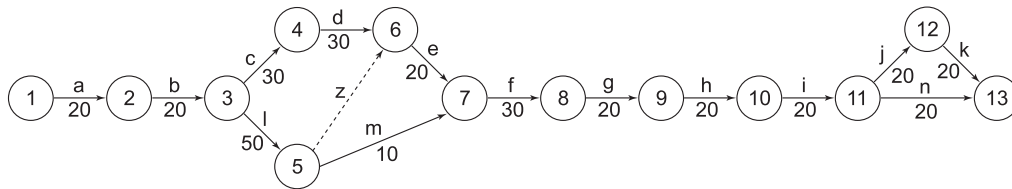
Hình 1-7-4

Ví dụ về
bảng ước
lượng công
việc

• í hiệu	• Mục việc	• S ố ngày cần thiết	• Các công việc phải được làm trước
•	• Phân tích và xác định yêu cầu hệ thống	• 20	• –
•	• Thiết kế hệ thống	• 20	• a
•	• Thiết kế vận hành chi tiết	• 30	• b
•	• Phân tích và xác định yêu cầu phần mềm	• 30	• c
•	• Thiết kế hệ thống phần mềm	• 20	• d, l
•	• Thiết kế phần mềm chi tiết	• 30	• e, m
•	• Lập trình	• 20	• f
•	• Kiểm thử móc nối phần mềm	• 20	• g
•	• Kiểm thử toàn bộ phần mềm	• 20	• h
•	• Kiểm thử móc nối hệ thống	• 20	• i
•	• Kiểm thử hệ thống	• 20	• j
•	• Thiết đặt phần cứng	• 50	• b
•	• Tạo dựng môi trường phát triển	• 10	• l
•	• Đào tạo và huấn luyện người liên quan	• 20	• i

2. Dựa trên bảng ước lượng công việc, biểu đồ PERT được vẽ trong Hình 1-7-5 (tính tới thứ tự công việc).

Hình 1-7-5 Ví dụ về biểu đồ PERT (hình thứ hai của hai biểu đồ)



-
-
- 3. Số ngày sau đây được quyết định tại từng nút.
 - Thời gian móc nối sớm nhất có thể
 - Chỉ ra thời gian sớm nhất có thể, trước đó công việc không thể được bắt đầu.
 - Thời gian móc nối muộn nhất có thể
 - Chỉ ra thời gian muộn nhất mà công việc phải được hoàn thành.
- 4. Đường đi được sinh ra bằng cách nối các nút, với từng nút mà thời gian sớm nhất có thể và thời gian muộn nhất có thể là như nhau (chỉ ra không được phép có lẽ có dân nào) được gọi là "đường găng". Công việc trên đường này là quan trọng nhất cho việc quản lý.

(3) Quản lý tiến trình

- Việc quản lý tiến trình được tiến hành theo hai quan điểm sau:
- Định thời gian bắt đầu và kết thúc của từng phần việc
- Trạng thái tiến độ của công việc cá nhân của từng người
- ① Quản lý việc định thời gian bắt đầu và kết thúc của từng việc
 - Việc quản lý được tiến hành sao cho từng phần việc được bắt đầu như đã được xác định trong bản kế hoạch tiến độ và được kết thúc như được xác định bằng mọi phương pháp, bằng cách kiểm tra trạng thái tiến độ tức khắc của từng phần việc và bằng cách lấy những cách đo thích hợp dựa trên trạng thái đó.
 - Sau đây ta xét các lý do cho việc chậm trễ trong công việc:
 - Kỹ năng của kỹ sư liên quan không đủ.
 - Việc lập kế hoạch và đặt mục tiêu không được xem xét thích hợp.
 - Các vấn đề liên quan tới nhân sự (kể cả việc bố trí lại thành viên phát triển, việc chuyển một số thành viên sang vị trí khác, và một số người rời bỏ công ti)
 - Vấn đề ngân sách (kể cả công cụ hỗ trợ phát triển cần mua thực tế có thể không được mua).
 - Trục trặc phần cứng và/hoặc phần mềm
 - Khi chậm trễ trong công việc được phát hiện ra, người quản lý, như người lãnh đạo dự án, phải điều tra các biện pháp để giải quyết tình huống này và lấy những biện pháp cụ thể, như thay đổi tiến độ, sớm nhất có thể được.
 - Để tạo khả năng cho các biện pháp như vậy, mọi thành viên lực lượng lao động phải báo cáo trạng thái tiến độ của công việc của mình một cách đều đặn thông qua nhật ký công việc hay báo cáo công việc tuần. Nói riêng, khi một tình huống bất ngờ xuất hiện, người đó phải báo cáo sớm nhất có thể được.
- ② Lập lịch cho từng thành viên lực lượng lao động
 - Việc lập lịch được dùng để phân bổ công việc của từng tiến trình cho từng thành viên lực lượng lao động, để quyết định thứ tự của từng phần việc được tiến hành, và để quản lý trạng thái tiến

độ công việc trên cơ sở hàng ngày. Việc lập lịch cũng có hiệu quả để làm cho thời gian chuyển giao sớm hơn và tối thiểu việc chậm trễ.

-
- **Ví dụ** Thiết kế ngoài, cho việc thiết kế đại cương một hệ thống, bao gồm nhiều phần việc, trong đó một số lớn các kỹ sư hệ thống (SEs) có tham dự vào. Trong ví dụ này, việc lập lịch trở thành như sau khi được xét theo quan điểm người lãnh đạo và quan điểm của thành viên lực lượng lao động, tương ứng:

- <Người lãnh đạo dự án>
 - Công việc thiết kế hệ thống được chia thành một số việc nhỏ.
 - Mỗi việc được phân bổ cho từng thành viên tùy theo mức độ kỹ năng của người đó.

Việc lập lịch cho từng thành viên (từng công việc) được thực hiện.

Hướng dẫn phân bổ công việc được trao cho từng thành viên.

Việc hoàn thành của từng công việc được kiểm tra bằng cách đọc nhật ký công việc hay báo cáo tuần của từng thành viên.

Tiến hành những biện pháp linh hoạt, như thay đổi kế hoạch nếu cần, khi phát hiện ra chậm trễ trong công việc.

- <Thành viên lực lượng lao động>
 1. Người đó tự quản lý mình bằng cách so sánh trạng thái tiến độ công việc với lịch công việc đã được phân bổ và cố gắng giữ thời gian kết thúc như kế hoạch.
 2. Người đó báo cáo về trạng thái tiến độ công việc thông qua nhật ký công việc hay báo cáo tuần, bên cạnh việc báo cáo khi công việc hoàn thành xong.

-
- Như đã mô tả ở trên, quản lý tiến trình được tiến hành bằng việc tổ hợp ① kế hoạch tiến trình thô (quyết định ngày tháng bắt đầu công việc và kết thúc công việc) và ② lập lịch chi tiết.

- Nói riêng, việc báo cáo cho người quản lý là rất quan trọng cho cả ① và ②.

- Hình 1-7-6 chỉ ra mối quan hệ giữa kế hoạch và báo cáo.

Hình 1-7-6 Quan hệ giữa kế hoạch và báo cáo		Mức trung	Mức trung	Mức thấp
	Kế hoạch	Báo cáo bao quát dự án như một toàn thể được soạn ra	Tạo ra kế hoạch tiến độ cho từng hệ con	Lịch công việc cho từng thành viên lực lượng lao động được tạo ra.
	Báo cáo	Báo cáo hoàn thành công việc	Báo cáo hoàn thành tiến trình	Báo cáo công việc Báo cáo công việc theo tuần

1.7.4 Năng suất phần mềm

- Để đánh giá năng suất phần mềm, phải đánh giá được qui mô phần mềm. Việc phát triển phần mềm bao gồm nhiều cái ra khác nhau tùy theo từng tiến trình, như các đề án, đặc tả yêu cầu, đặc tả thiết kế, đặc tả chương trình, và chương trình nguồn. Tuy nhiên, phần nhiều trong chúng

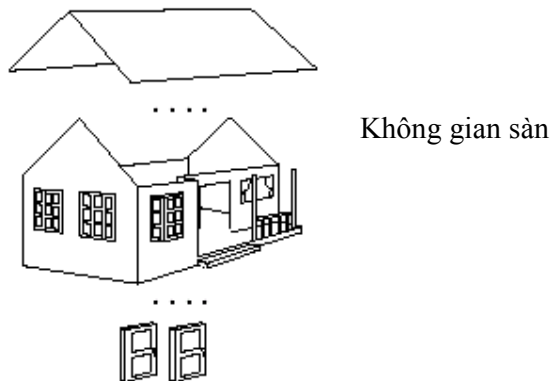
được tạo ra bằng công việc kiểu thủ công, tùy thuộc phần lớn vào kinh nghiệm và cảm giác con người. Do đó, việc ước lượng chi phí cũng phụ thuộc chủ yếu vào kinh nghiệm và cảm giác. Để cải thiện tình hình này, người ta đề nghị dùng kỹ nghệ phần mềm. Sau đó nhiều phương pháp ước lượng chi phí đã được đề nghị, và một số trong chúng bây giờ đang được dùng.

(1) Đại cương về ước lượng

- Chẳng hạn, Hình 1-7-7 chỉ ra ví dụ về ước lượng khi xây nhà.

Hình 1-7-7

Ví dụ ước lượng chi phí làm nhà



-
- Ước lượng chi phí nhà được so sánh dễ dàng
-
- Mái, cửa và cửa sổ tất cả đều thấy được. Do đó, nếu bản đại cương được thiết kế, thì chi phí có thể được ước lượng (dựa trên không gian sàn nhà và các nhân tố khác).
- Tuy nhiên, tốp, cơ sở dữ liệu và chương trình tất cả lại không thấy được. Do đó, các ước lượng trong pha thiết kế chi tiết trở thành khác biệt lớn với các ước lượng trong pha lập kế hoạch cơ sở. Hậu quả là, ước lượng về phát triển hệ thống nên được tiến hành trong nhiều pha được mô tả như sau:
 1. Trong pha lập kế hoạch cơ sở (khi việc hệ thống hoá được lập kế hoạch)
 2. Trong pha thiết kế ngoài (khi việc phân hoạch thành các hệ con được tiến hành)
 3. Trong pha thiết kế trong (khi chương trình được thiết kế).

(2) Cách làm ước lượng

- Có các phương pháp sau đây để ước lượng tỉ lệ phát triển:
 - Ước lượng dựa theo dữ liệu quá khứ
 - Ước lượng dựa theo số dòng mã
 - Phương pháp nhiệm vụ chuẩn
 - Phương pháp FP (điểm chức năng)
 - Các mô hình ước lượng đa dạng (mô hình COCOMO, v.v.)
- Trong phần sau đây, từng phương pháp được nhắc tới trên sẽ được mô tả ngắn gọn:
- ① Ước lượng dựa trên dữ liệu quá khứ
 - Trong phương pháp này, các ước lượng về hệ thống được phát triển và suy ra dựa trên dữ liệu thực của hệ thống tương tự đã xây dựng trong quá khứ. Có hai cách để làm ước lượng.
 - Toàn bộ tiến trình phát triển hệ thống được phân hoạch thành một số bước, và các ước lượng

được suy dẫn ra dựa trên dữ liệu thực cho công việc tương tự.

- Hệ thống được phân hoạch thành một số mô đun chương trình, và các ước lượng được suy ra dựa trên dữ liệu thực tế cho các mô đun chương trình tương tự.
- <Đặc trưng>
- Với hệ thống tương tự trong quá khứ, các lỗi cơ sở khó mà bị bao hàm vào.
- Nhiệm vụ ước lượng là tương đối đơn giản.
- Số lỗi trong các ước lượng trở nên lớn hơn nếu hệ thống quá khứ thích hợp không được chọn cho việc ước lượng.
- Việc áp dụng phương pháp này là không thể được nếu không có hệ thống tương tự trong quá khứ.

• ② Phương pháp dựa trên LOC

- Phương pháp dựa trên LOC là hay được dùng nhất làm phương pháp cho việc ước lượng kích cỡ phát triển. Với phương pháp này, kích cỡ phát triển được ước lượng bằng số dòng mã (chẳng hạn, LOC, XXXX kilo tương đương COBOL), và dựa trên dữ liệu này, khối lượng tài nguyên cần thiết được ước lượng ra.

- <Thủ tục>
 - 1. Hệ thống được diễn tả như một tập các mô đun chương trình
 - Các chức năng hệ thống được phân hoạch thành các mô đun chương trình, với mối quan hệ giữa chúng được chỉ ra bằng biểu đồ khối cấu trúc hay các phương tiện khác.
 - 2. Tính toán kích cỡ của từng chương trình
 - Số các LOC trong từng mô đun chương trình trong biểu đồ được ước lượng. Rồi tổng số các LOC được tính toán.
 - 3. Ước lượng nhân lực cho tất cả các chương trình cần làm.
 - Tổng số các LOC được chuyển thành tổng nhân lực, như dữ liệu và người-tháng (số người cần thiết nhân với số tháng cần thiết). Chẳng hạn, nếu việc phát triển hệ thống cần nỗ lực làm việc 2 năm của 20 người, thì nhân lực là 20 người x 24 tháng = 480 người-tháng.
 - 4. Ước lượng trên cơ sở tiến trình
 - Khối lượng nhân lực được phân bổ cho từng tiến trình, như lập kế hoạch cơ sở và các thiết kế khác nhau, với số phần trăm phân bổ được quyết định dựa trên dữ liệu quá khứ.
 - 5. Ước lượng về nhân lực gián tiếp
 - Trọng số cho nhân lực đối với các công việc KNPM, như phân tích và thiết kế hệ thống, và trọng số cho nhân lực đối với công việc hành chính, sẽ được quyết định.
 - 6. Tổng nhân lực được ước lượng
 - Tổng nhân lực được tính bằng việc kết tập dữ liệu nhân lực cho từng tiến trình.
- <Đặc trưng>
- Nó cung cấp phương pháp tiêu biểu nhất.
- Nếu có các chuẩn rõ ràng để ước lượng chương trình LOC và để chuyển chúng thành khối lượng nhân lực, thì tính toán được bao hàm là khá đơn giản.
- Điều tiên quyết là đại cương về các chức năng của chương trình cần phát triển phải được hiểu thấu.

- ③ Phương pháp dựa trên nhiệm vụ chuẩn

- Với phương pháp dựa trên nhiệm vụ chuẩn, công việc được chia ra trên cơ sở cái ra hay trên cơ sở xử lý bằng WBS (Work Breakdown Structure - Cấu trúc phân việc). Sau đó, ước lượng chi tiết được thực hiện cho từng đơn vị, và ước lượng kết quả được tích lũy theo cách từ dưới lên. Tham khảo 1.1.2 về WBS.

-
-

- <Thủ tục>

- 1. Kiểm tra đầu ra và công việc được yêu cầu

- Hệ thống được chia ra thành một cấu trúc phân cấp dựa trên WBS, và tất cả các đầu ra cần được phát triển trong dự án được liệt kê ra. Sau đó, tất cả công việc cần làm để sinh ra những cái ra này được chọn lấy.

- 2. Kích cỡ của từng công việc được chuyển thành khối lượng nhân lực.

- Khối lượng nhân lực cần cho từng đơn vị công việc được chọn được đưa ra ước lượng theo những chuẩn nào đó, như dữ liệu thực tế cho chuẩn đã có tác dụng trong quá khứ.

- 3. Kết tập toàn bộ nhân lực

- Khối lượng nhân lực được ước lượng cho từng công việc được tính tổng lại.

- <Đặc trưng>

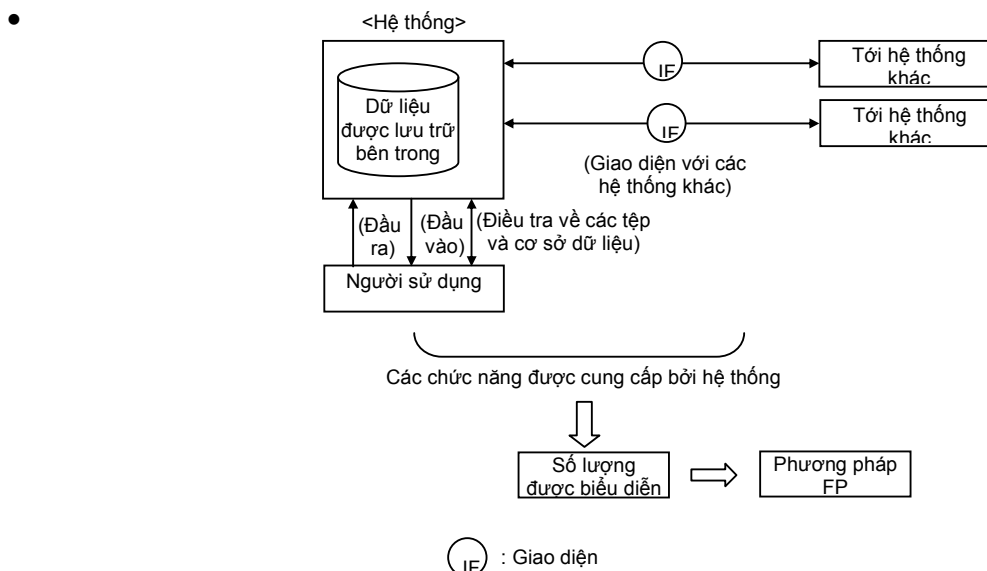
- Với phương pháp này, việc ước lượng được thực hiện sau khi công việc được chia thành mức cái ra chi tiết hay mức xử lý, rồi các ước lượng được tích lũy theo cách từ dưới lên. Do đó, nền cho các ước lượng được làm rõ ràng.

- Nếu có phát sinh sai biệt thì việc nhận diện nguyên nhân là dễ dàng.

- Dữ liệu thực tế cho công việc chuẩn là cần có. Thêm vào đó, công việc ước lượng đòi hỏi nhiều nỗ lực.

- ④ Phương pháp FP (Function Point - điểm chức năng)

- Với phương pháp (điểm chức năng), từng chức năng được đưa vào trong hệ thống sẽ được diễn đạt định lượng bằng một phương pháp nào đó, và do vậy dữ liệu được biểu diễn theo định lượng được dùng như cách đo ước lượng (xem Hình 1-7-8).



Hình 1-7-8

Ví dụ về phương pháp FP

-
-
-
-
-
-
-
-
-
-

• Phương pháp này khác cơ bản với ba phương pháp ① tới ③ đã mô tả trên trong việc dùng từng chức năng được cung cấp cho khách hàng xem như đơn vị đo (hoặc phương pháp này được gọi là phương pháp ước lượng hướng khách hàng). Chẳng hạn, các chức năng sau đây được người dùng sử dụng được lựa làm đơn vị để diễn đạt về định lượng.

-
-
- <Đơn vị được dùng như chuẩn>
 - Cái vào
 - Cái ra
 - Tập và cơ sở dữ liệu (dữ liệu được lưu giữ nội bộ)
 - Yêu cầu về tệp và cơ sở dữ liệu
 - Giao diện với hệ thống khác
- <Thủ tục>
 - 1. Kiểm các chức năng ("đơn vị được dùng làm chuẩn" đã được mô tả ở trên) được hệ thống cung cấp
 - 2. Các chức năng được lựa trong Khoản mục 1 trên, được phân lớp thành các loại "đơn giản," "trung bình" hay "phức tạp." Sau đó, một trọng số được gán cho từng loại dựa trên những chuẩn nào đó.
 -
 - 3. Các giá trị được cho trong Khoản mục 2 trên được kết tập (vậy dữ liệu suy dẫn thiết lập nên FP trước khi việc điều chỉnh được thực hiện).
 - 4. Các hệ số chuyên hệ thống được suy ra tùy theo đặc trưng của hệ thống đích.
 - 5. FP cuối cùng được tính toán bằng việc nhân dữ liệu từ Khoản mục 3 ở trên, với dữ liệu từ Khoản mục 4 ở trên.
 - 6. Giá trị FP được chuyển thành khối lượng nhân lực dự án.
- <Đặc trưng >
- Dữ liệu dễ hiểu với người dùng, bởi vì việc ước lượng được thực hiện cho các khoản mục thấy được với người dùng.

- Việc điều chỉnh được thực hiện dựa trên dữ liệu thực tế được tích lũy trong quá khứ. Do đó, việc tích lũy dữ liệu là cần thiết.
- Cần có tiêu chuẩn đánh giá chuẩn hoá trong việc áp dụng phương pháp ước lượng này.
- ⑤ **Mô hình COCOMO (CONstruction COSt MOdel)**
 - Mô hình COCOMO, một phương pháp ước lượng do Boehm đề xuất, là phù hợp cho việc ước lượng các hệ thống cỡ trung tới cỡ lớn.
 - Với mô hình COCOMO, hệ thống được phân lớp dựa trên ba mức sau. Sau đó với từng mức, nhân lực phát triển tổng cộng và thời kì phát triển được tính toán từ số các câu lệnh được dự kiến vào lúc hệ thống được trao cho người dùng.
 - <Ba mức>
 - Một tổ chức (phát triển hệ thống cỡ nhỏ)
 - Một nửa nhúng (phát triển hệ thống cho vận hành bình thường)
 - Hệ thống nhúng (phát triển các hệ thống lớn và có ràng buộc dư thừa)
 - <Đặc trưng>
 - Bên cạnh việc được chính nó dùng, phương pháp này cũng còn được dùng để kiểm các ước lượng theo các phương pháp khác.
 - Cần tích lũy và phân tích dữ liệu thực tế bao gồm dữ liệu nhân lực phát triển cho các hệ thống đa dạng.
-

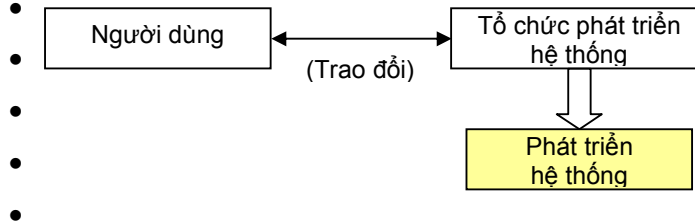
1.7.5 Tổ chức phát triển

- Có nhiều điểm đóng góp cho sự thành công của việc phát triển hệ thống, kể cả các khoản mục liên quan tới quản lý công việc đa dạng như sau:
 - Sự tham dự tích cực của người dùng (thiết lập tổ chức phát triển)
 - Quản lý tiến trình kỹ lưỡng và quản lý tiến trình của công việc từng người
 - Quản lý chất lượng hệ thống kỹ lưỡng
- Việc phát triển hệ thống qui mô lớn cần thời gian phát triển lâu (đôi khi đến nhiều năm), và đòi hỏi số tiền và tài nguyên nhân lực rất lớn.
- Tuy nhiên, không phải bị quan rằng việc phát triển hệ thống thất bại trong pha lập kế hoạch cơ sở hay pha thiết kế hay, mặc dầu việc phát triển được hoàn tất, chất lượng lại quá nghèo nàn không dùng được trong vận hành thực tế.
- Một cách tự nhiên, công việc phát triển được quản lý bởi một người quản lý, chẳng hạn, người quản lý dự án. Tuy nhiên, việc quản lý đúng công việc của riêng từng thành viên phát triển và tạo ra cái ra chất lượng cao là nhân tố quan trọng dẫn việc phát triển hệ thống tới thành công.
- Điều quan trọng thứ nhất để đưa việc phát triển hệ thống tới thành công là thiết lập một tổ chức phát triển vững chắc. Làm tiến độ công việc đúng theo lịch phát triển hệ thống (lịch mức cao nhất) đã tạo ra trong pha lập kế hoạch cơ sở là điều tiên quyết lớn nhất cho sự thành công. Hơn nữa, liệu công việc có tiến hành trôi chảy hay không cũng có thể được nói là tùy thuộc phần lớn vào công việc hợp tác của các thành viên lực lượng lao động.

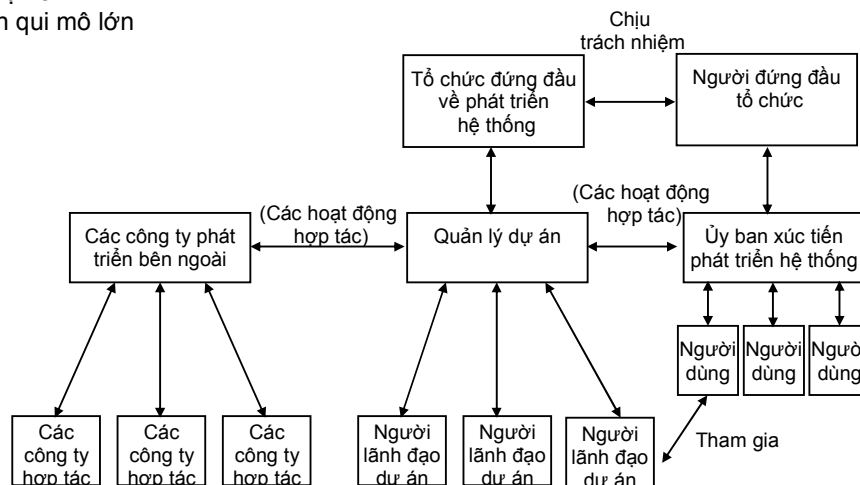
(1) Các phong cách tổ chức

- Kiểu phong cách tổ chức phát triển nào nên được dùng còn tùy theo qui mô phát triển hay những người tạo nên lối cho việc phát triển. Tuy nhiên, sự tham dự của người dùng là không thể thiếu được trong bất kì tổ chức nào. Trong nhiều việc phát triển hệ thống thành công, tổ chức của người dùng tham dự vào việc lập kế hoạch cơ sở và thiết kế ngoài.
- Hơn nữa, việc phát triển hệ thống thường được tiến hành trong sự hợp tác với các công ti phát triển bên ngoài. Chẳng hạn, việc khởi thảo cho tới pha thiết kế được thực hiện nội bộ, còn việc lập trình thì thuê khoán ngoài với các công ti khác.
- Xem như một ví dụ về tổ chức phát triển, Hình 1-7-9 chỉ ra một tổ chức cho dự án qui mô nhỏ, trong khi Hình 1-7-10 nêu ra một tổ chức cho dự án qui mô lớn.

Hình 1-7-9 Ví dụ về tổ chức phát triển qui mô nhỏ



Hình 1-7-10 Ví dụ về tổ chức phát triển qui mô lớn



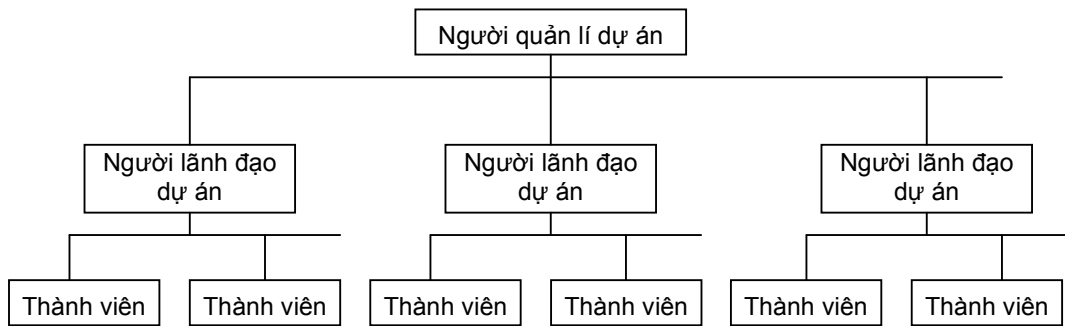
(2) Tổ chức phát triển

- Tổ chức phát triển nhận các yêu cầu hệ thống hoá từ người dùng, và tiến hành công việc về lập kế hoạch cơ sở, các kiểu thiết kế, lập trình và các kiểu kiểm thử. Gần đây, trong nhiều trường hợp, các tổ dự án nội bộ thực hiện công việc về lập kế hoạch cơ sở và thiết kế, còn lập trình và kiểm thử được uỷ quyền cho các công ti phát triển phần mềm bên ngoài. Tuy nhiên, tổ chức phát triển vẫn tiến hành công việc kiểm nhận sau khi kiểm thử đã hoàn tất.

① Các kiểu tổ chức phát triển

- Nhiều việc phát triển hệ thống được tiến hành như các dự án. Định nghĩa của NASA (National Aeronautics and Space Administration) về dự án là "Các nhiệm vụ được tiến hành trong nhiều tổ chức kéo dài từ một tới năm năm và có liên quan lẫn nhau." Nói cách khác, dự án chỉ ra một tổ chức với các mục đích xác định kéo dài trong một khoảng thời gian giới hạn.

Hình 1-7-11 Ví dụ về tổ chức phát triển (tổ cấp bậc)



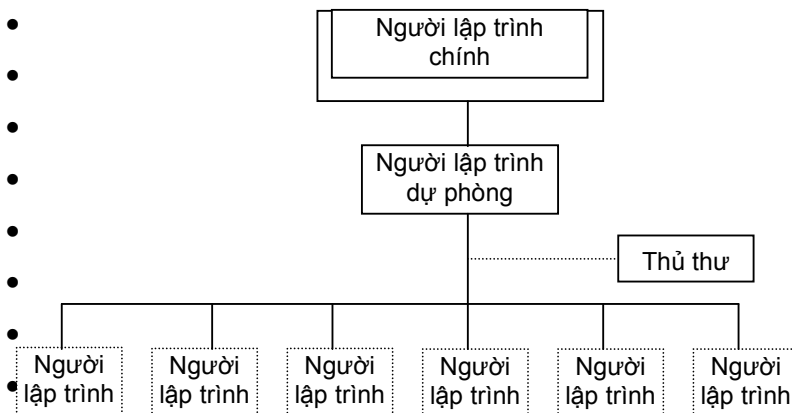
Có ba kiểu tổ dự án điển hình.

- Tổ người lập trình chính
- Tổ chuyên gia
- Tổ phân cấp

a. Tổ người lập trình chính

Tổ người lập trình chính là một tổ dự án bao gồm một số tương đối nhỏ tối đa mười thành viên, với người lập trình chính có hoàn toàn trách nhiệm thực hiện quyền lãnh đạo trong việc phân bổ công việc cho từng thành viên một cách rõ ràng và làm tăng năng suất và chất lượng.

Hình 1-7-12 Ví dụ về tổ người lập trình chính



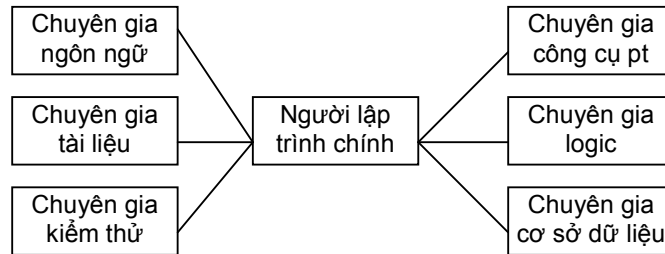
< Đặc trưng >

- Dự án qui mô tương đối nhỏ có thể chấp nhận kiểu tổ chức tổ này.
- Đặc trưng có ý nghĩa nhất của tổ chức này là sự tồn tại của người lập trình dự phòng và thủ thư.
- Nó phù hợp cho việc rèn luyện người lãnh đạo (người lập trình chính phải chịu gánh nặng trách nhiệm).
- Nó có khuynh hướng gây ra sự suy giảm tinh thần của người lập trình.
- b. Tổ chuyên gia
- Tổ chuyên gia là một kiểu sửa đổi của kiểu tổ người lập trình chính, và bao gồm một người

lập trình chính cùng nhiều chuyên gia kỹ thuật.

Hình 1-7-13

Ví dụ về tổ chuyên gia



<Đặc trưng>

Người lập trình chính tạo ra tất cả các chương trình.

Các chuyên gia kỹ thuật chịu trách nhiệm các lĩnh vực đặc biệt (như công cụ phát triển, kiểm thử, tài liệu, cơ sở dữ liệu, v.v.) giúp cho công việc của người lập trình chính, mở rộng khả năng của người lập trình tới mức tối đa có thể.

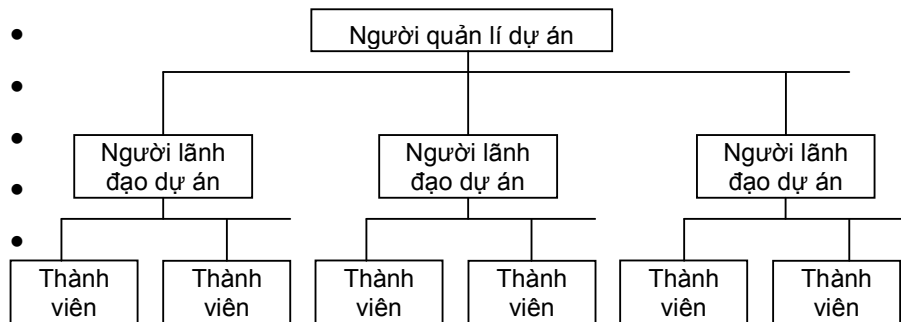
Điều bản chất là các thành viên có kỹ năng mức cao.

c. Tổ phân cấp

Tổ phân cấp bao gồm một người quản lý dự án, nhiều người quản lý dự án và các thành viên lực lượng lao động.

Hình 1-7-14

Ví dụ về tổ phân cấp



<Đặc trưng>

- Kiểu tổ chức tổ này được sử dụng rộng rãi nhất ở Nhật.

- Nó được chấp nhận trong việc phát triển phần mềm qui mô tương đối lớn.

- Trao đổi trở nên kém thích hợp hơn, nếu so với tổ người lập trình chính.

② Vai trò của các thành viên tổ

a. Người chịu trách nhiệm tổ chức phát triển (người quản lý dự án)

Trong nhiều trường hợp, người chịu trách nhiệm của tổ chức phát triển trở thành người quản lý dự án. Người quản lý, tại một vị trí quan trọng, hoàn toàn chịu trách nhiệm cho dự án phát triển.

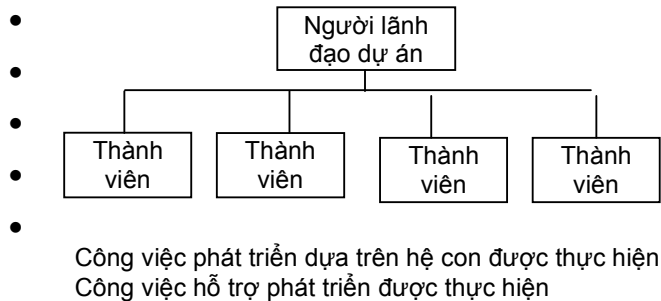
Người quản lý dự án phải không chỉ có kỹ năng công nghệ thông tin mức cao, mà còn phải có

khả năng quản lý dự án và khả năng lập kế hoạch. Thêm vào đó, việc duy trì trao đổi đúng đắn bên trong công ti và với các bên ở ngoài công ti là một vai trò quan trọng của người quản lý.

- <Vai trò>
- Lập kế hoạch, soạn thảo kế hoạch, thực hiện và ước lượng dự án.
- Trao đổi với người dùng và các tổ chức có liên quan (kể cả các bên ở ngoài công ti)
- Truyền sinh lực cho công việc dự án (kể cả việc triển khai nhân sự và truyền quyền lực)
- Công việc quản lý khác
- b. Người lãnh đạo dự án
- Người lãnh đạo dự án đóng vai trò phân bổ người quản lý dự án, đặt hoạt động tổ vào trật tự, hay hành động như người ở giữa các thành viên lực lượng lao động và người quản lý dự án.
- Một tổ chức được người lãnh đạo dự án điều khiển được gọi là "tổ dự án con", thực hiện công việc phát triển thực tại hay công việc hỗ trợ phát triển (kỹ thuật, kiểm thử, chuẩn hoá hay các công việc khác).

Hình 1-7-15

Ví dụ về tổ dự án con



- c. Thành viên
- Được chỉ dẫn bởi người lãnh đạo dự án, các thành viên thực hiện công việc phát triển thực tế (thiết kế, lập trình c.c..) hay công việc hỗ trợ phát triển.

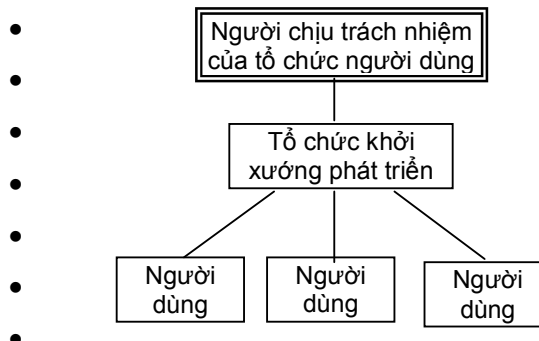
(3) Tổ chức người dùng

- Việc phát triển hệ thống được tiến hành theo yêu cầu của tổ chức người dùng, và tổ chức người dùng dùng hệ thống đã được phát triển. Do đó, mặc dầu tổ chức phát triển thực hiện việc phát triển hệ thống, việc phát triển hệ thống vẫn không thể thành công được nếu không có sự hợp tác của tổ chức người sử dụng.

- Hình 1-7-16 chỉ ra một cấu trúc tổ chức đơn giản của tổ chức người dùng.

Hình 1-7-16

Cấu trúc tổ chức đơn giản hoá của tổ chức người dùng



- ① **Người chịu trách nhiệm của tổ chức người dùng**
 - Người chịu trách nhiệm của tổ chức người dùng có quyền lớn nhất trong mọi pha từ khía cạnh ngân sách tới việc thúc đẩy dự án phát triển hệ thống hiện tại. Với người chịu trách nhiệm của tổ chức người dùng, người đó cũng được yêu cầu rằng người đó phải làm nỗ lực tối đa để làm tăng tỉ lệ hiệu quả-đầu tư bằng việc thực hiện nhiều kế hoạch khác (như tổ chức các khoá huấn luyện) như người lãnh đạo.
- ② **Tổ chức khởi xướng phát triển**
 - Tổ chức khởi xướng phát triển được tổ chức với những người có trách nhiệm (những người ở vị trí quản lí) trong tổ chức người dùng làm cốt lõi, và đưa ra sự chấp thuận các cái ra từ việc phát triển hệ thống. Tuy nhiên điều đó không liên quan tới chi tiết hệ thống
- ③ **Người dùng**
 - Người dùng thực tế sử dụng hệ thống. Vậy, người dùng nên tham gia vào việc phát triển hệ thống cho lời khuyên về nhiều hoạt động đa dạng. Do đó, người dùng tham gia nên quen các tiến trình nghiệp vụ.

1.8 Gói phần mềm

- Gói phần mềm là sản phẩm phần mềm được nhiều người dùng sử dụng. Ngày nay, kinh doanh về gói phần mềm rất phát đạt, và rất nhiều sản phẩm gói đã có sẵn, cho nên chúng nên được dùng một cách có hiệu quả cũng như thường xuyên. Sau đây sẽ mô tả toàn bộ các khía cạnh của gói phần mềm.

1.8.1 Đại cương về gói phần mềm

- Vai trò của gói phần mềm và tính ích lợi của chúng sẽ được mô tả sau đây.
-

(1) Ý nghĩa và vai trò của gói phần mềm

- Có những khác biệt sau đây giữa gói phần mềm và các mảnh phần mềm riêng lẻ:
- - Gói phần mềm đặt nặng vào những phần công việc thông thường chung.
- Các mảnh phần mềm riêng lẻ bao gồm cả những khác biệt tồn tại trong công việc.
- Nói cách khác, phần mềm được tạo ra cho việc sử dụng chung được gọi là gói phần mềm.
- Gần đây, người ta nói rằng khối lượng việc tồn đọng (công việc phát triển mà không thể được bắt đầu) đã tăng lên. Về mặt thống kê, nghe nói có số trung bình việc tồn đọng lên đến quãng hai ba năm. Các nguyên nhân sau đây được xem như là nguyên nhân chính cho việc tồn đọng:
- Tổ chức phát triển hệ thống không thể đáp ứng kịp việc tăng về nhu cầu phát triển hệ thống. Năng suất phát triển hệ thống là thấp (các kỹ thuật phát triển và công cụ hỗ trợ phát triển là không đủ).

Việc cung cấp các kỹ sư phát triển hệ thống quá thấp và kỹ năng của họ không đủ (do tình hình kinh tế trì trệ kéo dài).

- Gói phần mềm cung cấp một phương tiện hiệu quả để giải quyết những vấn đề này.
- Có thời việc đưa gói phần mềm vào đã bị phản ứng lại. Đằng sau cảm giác này, có xu hướng là hệ thống nên được đặt hàng và phát triển một cách cá nhân. Tuy nhiên, gói phần mềm với tỉ lệ hiệu năng-chi phí cao gần đây đã trở nên sẵn có. Một số sản phẩm có thể đáp ứng nhu cầu vận hành ngay cả ở mức độ cao đáng kể.
- Với xu hướng như vậy, nhiều nhà chế tạo phần mềm đã chế tạo và bán gói phần mềm, như các nhà chế tạo sản phẩm công nghiệp chế tạo và bán sản phẩm của họ. Bây giờ, việc kinh doanh gói phần mềm là lĩnh vực kinh doanh chính trong ngành công nghiệp phần mềm.
-

(2) Tính hữu dụng của gói phần mềm

- Gói phần mềm được bán như một bộ đĩa CD-ROM và một số tài liệu sử dụng. Các gói này bao quát những lĩnh vực rộng lớn, kể cả những gói cho kỹ sư phát triển hệ thống và những gói dành cho người dùng thông thường.
- Thêm vào đó, chúng lại đưa ra hiệu năng rất cao. Có rất nhiều gói phần mềm sẵn có ở Nhật

và ở nước ngoài. Ích lợi do các gói phần mềm này đem tới cho hệ thống tin là không lường được.

- <Ích lợi sử dụng>
Thời gian phát triển được rút ngắn lại.
Kích cỡ phát triển được rút lại (chỉ những phần mà gói phần mềm không bao quát mới cần được phát triển).
Chất lượng được tăng lên (các gói phần mềm ít bị lỗi đáng kể, bởi vì chúng được phát triển dưới sự quản lý chất lượng nghiêm ngặt).
Dịch vụ nâng cấp được kèm theo (thông thường, các gói phiên bản mới là có sẵn và không đắt, trong khi các chức năng của gói phần mềm lại được mở rộng).
- Tuy nhiên, việc dùng gói phần mềm trong phát triển hệ thống có thể gây ra những bất tiện trong giao diện người dùng. Trong trường hợp như vậy, các gói có liên quan được chuyên biệt hoá (sửa đổi logic bên trong hay những phần khác), nhưng việc xử trí cẩn thận là cần thiết bởi vì, khi phiên bản của một gói được cập nhật, thì các thao tác không được đảm bảo cho phiên bản đã chuyên biệt hoá của gói này.

1.8.2 Phân loại các gói phần mềm

- Gói phần mềm theo nhiều kích cỡ và kiểu là có sẵn. Một số là các gói kích cỡ nhỏ, như phần mềm truyền thông được cung cấp cùng chức năng thư điện tử, và một số là những gói kích cỡ lớn như phần mềm lập kế hoạch tài nguyên công ti ERP (Enterprise Resource Planning).
- Các hoạt động trong công ti được xem xét theo cấp bậc như trong Hình 1-8-1. Gói phần mềm được cung cấp để hỗ trợ cho từng hoạt động.

(1) Gói phần mềm trên cơ sở công nghiệp

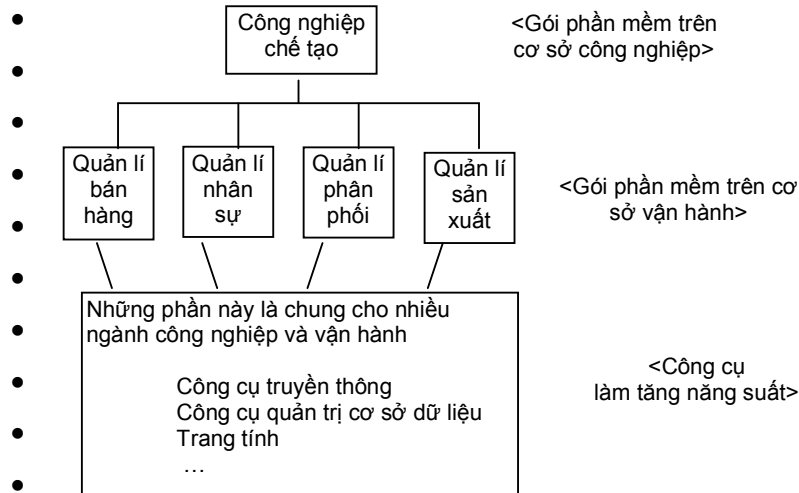
- Gói phần mềm trên cơ sở công nghiệp là sản phẩm phần mềm nhìn theo các đặc trưng chung cho từng công ti trong một ngành công nghiệp, và các sản phẩm cho toàn bộ hoạt động của một công ti hay cho từng đơn vị công việc (công nghiệp chế tạo, công nghiệp phân phối, công nghiệp tài chính, v.v.) của một tổ chức, được bán trên thị trường.

(2) Gói phần mềm trên cơ sở vận hành

- Gói phần mềm trên cơ sở vận hành là sản phẩm xét theo đơn vị vận hành được phân công cho từng tổ chức của công ti. Tuy nhiên, gần đây, các sản phẩm được dùng chung cho từng tổ chức cũng đã có sẵn.
-

Hình 1-8-1

Phân loại gói phần mềm



- ① ERP (Enterprise Resource Planning - Lập kế hoạch tài nguyên xí nghiệp)
 - ERP (Enterprise Resource Planning - Lập kế hoạch tài nguyên xí nghiệp) nêu ra một khái niệm hay phương pháp để làm cho các hoạt động nghiệp vụ có hiệu quả bằng việc quản lý các tài nguyên nghiệp vụ được tích hợp theo quan điểm dùng chung một cách có năng suất.
 - Gói phần mềm hỗ trợ cho các hoạt động để đạt tới khái niệm được gọi là ERP (gói dành cho các hoạt động tích hợp). Các sản phẩm của SAP hay Oracle Application là khá nổi tiếng.
- ② CRM (Customer Relationship Management - Quản lý quan hệ khách hàng)
 - CRM là một phiên bản mở rộng của ERP. ERP được phát triển bằng việc đặt ưu tiên vào hệ thống xử lý đơn hàng, như bán hàng, mua sắm, sản xuất, quản lý kho, dịch vụ và kế toán, xem như kết cấu nền thông tin của công ti. CRM tổ hợp các hoạt động của cơ sở dữ liệu khách hàng được quản lý tại trung tâm với ERP, tạo khả năng cho các hoạt động tiếp thị, chăm sóc khách hàng, làm tăng mức độ thoả mãn của khách hàng, cuối cùng làm vững chắc sự ủng hộ từ các khách hàng tốt.
- ③ SFA (Sales Force Automation - Tự động hoá hiệu lực bán hàng)
 - SFA là khái niệm có nguồn gốc từ BPA (Business Process Automation - Tự động hoá tiến trình nghiệp vụ). BPA là khái niệm về việc tăng mức độ thoả mãn của khách hàng, trong khi vẫn đạt tới việc giảm chi phí, bằng cách tăng năng suất do tự động hoá tiến trình nghiệp vụ. Với SFA, người ta dự định rằng năng suất trí tuệ sẽ được tăng lên bằng cách tự động tất cả các công việc văn phòng đơn giản và bằng việc dùng thời gian sinh ra dành cho dịch vụ khách hàng và/hoặc cung cấp đề xuất. Với hoạt động như vậy, việc dùng máy tính cá nhân kiểu sổ tay hay PDA (Trợ lý số tự cá nhân) là bản chất.
- ④ SCM (Supply Chain Management - Quản lý dây chuyền cung cấp)
 - SCM, cũng là một phiên bản mở rộng của ERP, là một gói hỗ trợ cho việc làm ra các kế hoạch trong toàn tổ chức bằng việc dùng kết cấu nền thông tin (tài nguyên) để đưa vào trật tự. Với SCM, người ta dự định rằng việc ra quyết định được hỗ trợ để làm tăng tốc độ làm kế hoạch.
- ⑤ CTI (Computer Telephone Integration - Tích hợp điện thoại máy tính)
 - CTI (Computer Telephone Integration - Tích hợp điện thoại máy tính) là gói phần mềm hỗ trợ cho việc tích hợp máy tính và điện thoại. Mục đích là để mở rộng cách thức theo dõi hệ thống máy tính được sử dụng. Có nhiều ứng dụng, kể cả hoạt động nhận đơn, khách hàng hỏi về các hoạt động, giúp đỡ tại bàn và chiến dịch tiếp thị, tất cả đều dùng đơn vị đáp ứng tiếng nói.

(3) Công cụ làm tăng năng suất

- Các công cụ làm tăng năng suất là các sản phẩm phần mềm để làm việc, như thư tín, quản lý cơ sở dữ liệu, trang tính và việc sinh ra tài liệu, đó là đối với các đơn vị nhỏ hơn thao tác và là thông thường cho công nghiệp và vận hành. Trong một số lĩnh vực, công cụ làm tăng năng suất được phân loại là OA (Office Automation - tự động hoá văn phòng) và công cụ thiết kế để hỗ trợ cho công việc thiết kế.

- <Kiểu>

Công cụ truyền thông (truyền tệp, thư tín điện tử, phần mềm nhóm (groupware), phần mềm truyền thông, v.v...).

Công cụ quản lý cơ sở dữ liệu (hệ quản trị cơ sở dữ liệu quan hệ (RDBMS), và các tiện ích khác)

Trang tính/công cụ sinh tài liệu

Công cụ quản lý dự án

Công cụ hỗ trợ phát triển hệ thống (công cụ CASE, công cụ tạo GUI, các công cụ kiểm thử đa dạng, v.v.).

-

1.8.3 Công cụ quản lý sản xuất

(1) CAD (Computer Aided Design - Thiết kế có máy tính hỗ trợ)

- CAD là một hệ thống mà người thiết kế làm ra bản thiết kế bằng cách nhận sự hỗ trợ của máy tính qua hiển thị đồ họa. CAD, dựa trên xử lý ảnh, xử lý cái nhìn 3 chiều từ cái nhìn 2 chiều đối với các bản vẽ.

(2) CAM (Computer Aided Manufacturing - Chế tạo có máy tính hỗ trợ)

- Theo nghĩa rộng hơn, CAMs là các hệ thống hỗ trợ cho tiến trình chế tạo, như quản lý tiến trình, chuẩn bị cho sản xuất, kiểm thử xử lý và lắp ráp. Tuy nhiên, thông thường chính hệ thống hỗ trợ cho việc sinh ra dữ liệu để hướng dẫn các đơn vị kiểm soát số. Các công cụ CAM ngày nay sinh ra dữ liệu lệnh từ cái ra lấy từ hệ thống CAD.

(3) CAE (Computer Aided Engineering - Kỹ nghệ có máy tính hỗ trợ)

- CAE là hệ thống để hỗ trợ cho một loạt công việc được bao hàm trong thiết kế sản phẩm, kiểm thử hiệu năng và chế tạo. CAE được dùng để làm cho công việc hiệu quả bằng cách phân tích hiệu năng cần cho sản phẩm, thiết kế dựa trên kết quả phân tích và mô phỏng các sản phẩm thực nghiệm trên máy tính.

- Tên "hệ thống CAD/CAM/CAE" đôi khi được dùng để bao quát cả hai hệ thống thiết kế và chế tạo.

-

-

1.8.4 Ví dụ về việc dùng hiệu quả gói phần mềm

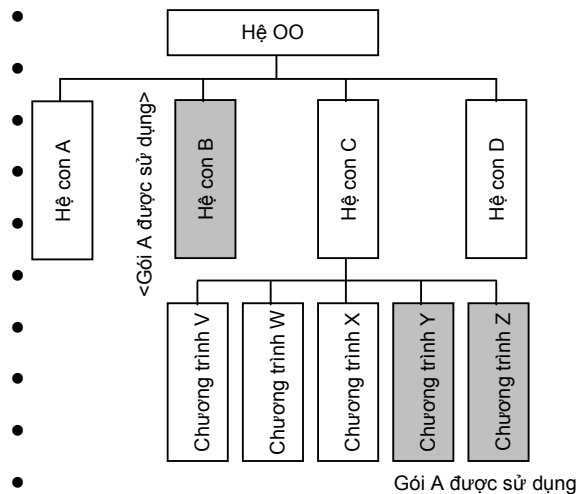
- Như mô tả ở trên, việc dùng các gói phần mềm đem lại ích lợi lớn cho nghiệp vụ. Sau đây sẽ mô tả các ví dụ điển hình về việc dùng các gói phần mềm có hiệu quả.

(1) Dùng gói phần mềm như một phần của hệ thống

- Gói phần mềm không chỉ bao gồm ứng dụng mà còn cả gói chương trình. Do đó, việc dùng gói phần mềm tại mức hệ thống con và mức chương trình cũng là có thể (xem Hình 1-8-2). Việc thiết lập thích hợp hay hơi sửa đổi các tham biến cho phép nhiều gói phần mềm được dùng cho những mục đích này. Do đó, gói phần mềm nên được dùng nhiều nhất có thể được.

- Gần đây, các thư viện lớp trong lập trình hướng đối tượng đã được tổ hợp vào trong các gói phần mềm. Do đó, đã xuất hiện các trường hợp chỉ riêng phần cần thiết mới được tạo ra và thư viện được dùng cho phần còn lại.

Hình 1-8-2
Ví dụ về nhúng hệ thống



(2) Tăng năng suất phát triển hệ thống

- Nhiều công cụ hỗ trợ cho việc phát triển hệ thống đã có sẵn như các gói phần mềm. Do đó, việc dùng chúng tạo khả năng cho việc phát triển hệ thống hiệu quả.

- <Ví dụ sản phẩm của gói phần mềm hỗ trợ phát triển>
 - Công cụ CASE: U-CASE, L-CASE, I-CASE, v.v...
 - Công cụ sinh GUI/tài liệu
 - Công cụ quản lý dự án
 - Công cụ kiểm thử/gỡ lỗi

(3) Thúc đẩy tự động hoá văn phòng (OA)

- Các nghiệp vụ đã đưa vào gói phần mềm để thúc đẩy tự động hoá văn phòng với ý định bao hàm những điều sau:

- Làm cho mọi dữ liệu được quản lý như các cơ sở dữ liệu.
- Quản lý luồng thông tin liên quan tới vận hành, như thư điện tử và luồng công việc

- - Quản lý lịch biểu
- RDBMS được dùng chủ yếu như cơ sở dữ liệu, trong khi phần mềm nhóm groupware (phần mềm hỗ trợ công việc được thực hiện như một nhóm) được dùng rộng rãi để thúc đẩy tự động hoá văn phòng.

(4) Dùng như nghiệp vụ

- Trong công nghiệp phần mềm, nghiệp vụ của một số lớn công ti là để phát triển và bán gói phần mềm. Nếu một công ti có thể phát triển và bán sản phẩm đáp ứng nhu cầu của khách hàng, thì người ta trông đợi sự phát triển mạnh mẽ cho công ti này. Tuy nhiên, việc phát triển hệ thống cho các gói phần mềm có hơi khác với việc phát triển phần mềm lẻ. Hình 1-8-3 chỉ ra sự khác biệt giữa việc phát triển gói phần mềm và việc phát triển phần mềm lẻ.

Hình 1-8-3 So sánh giữa việc phát triển gói phần mềm và việc phát triển phần mềm lẻ

	Phát triển gói phần mềm	Phát triển phần mềm lẻ
Lập kế hoạch cơ sở	<ul style="list-style-type: none"> • Dựa trên tiếp thị, thực hiện công việc phát triển phần mềm cần thiết trong xã hội. • Trong một số trường hợp, gói phần mềm mới được sinh ra bằng cách tích hợp gói phần mềm đã có. 	<ul style="list-style-type: none"> • Qua gặp gỡ với người dùng, việc phân tích và xác định được tiến hành để hệ thống hoá điều người dùng cần.
Kiểm thử/chuyển giao	<ul style="list-style-type: none"> • Phần mềm được cài đặt trong nhiều kiểu máy tính cho nên cần tiến hành quản lý chất lượng kỹ lưỡng để loại bỏ lỗi. • Hơn nữa, việc chuyên biệt hoá được thực hiện cho khách hàng, sau đó nội dung được ghi lại trên CD-ROM hay phương tiện khác, để chuyển giao cho khách hàng. 	<ul style="list-style-type: none"> • Kiểm thử bình thường, kể cả kiểm thử đơn vị cho tới kiểm thử vận hành, được tiến hành, sau đó hệ thống được trao cho người dùng.
Bảo trì	<ul style="list-style-type: none"> • Phiên bản của một sản phẩm được cập nhật, có xem xét tới xu hướng thị trường và mối quan hệ với phần mềm khác cũng làm việc với sản phẩm này. 	<ul style="list-style-type: none"> • Công việc bảo trì được tiến hành dựa trên yêu cầu của người dùng hay sự thay đổi của tình huống.

- Từ nay trở đi, người ta trông đợi rằng, bên cạnh việc phát triển các gói phần mềm mới, nhiều gói phần mềm tích hợp tích hợp cả các gói phần mềm hiện có sẽ được tạo ra.

1.8.5 Phần mềm nhóm - Groupware

- Để hỗ trợ cho công việc con người, giao diện với con người của máy tính đã được cải tiến và nâng cao. Tuy nhiên, công việc con người gần như được tiến hành theo nhóm. Việc phát triển hệ thống là một ví dụ điển hình.
- Với tiến bộ về công nghệ mạng trong những năm gần đây, nhiều máy tính được nối lại qua mạng và việc trao đổi thông tin giữa chúng đã trở thành có thể. Phần mềm nhóm là một hệ thống để hỗ trợ cho việc gắn công việc qua mạng mà qua đó đã được phát triển.

-

(1) Truyền thông

- Các chức năng truyền thông là những chức năng quan trọng nhất của phần mềm nhóm. Truyền thông bên trong một nhóm có thể được duy trì qua thư điện tử và bản tin điện tử mà không cần triệu tập họp ở đâu đó.

(2) Dùng chung dữ liệu

- Dữ liệu dùng chung tạo khả năng coi máy tính như chỗ làm việc chung. Việc soạn tài liệu trong sự hợp tác với một số người đưa ra một ví dụ điển hình. Những người viết ở các chỗ khác nhau có thể dùng chung một tệp tài liệu được đặt ở một máy phục vụ định sẵn.
- Thêm vào đó, vì phần mềm nhóm cho phép việc quản lý dữ liệu tập trung, nên không cần những phần khác nhau giữ cùng một dữ liệu.

(3) Quản lý lịch biểu

- Chức năng quản lý tập trung lịch biểu của các thành viên nhóm, được cung cấp cùng phần mềm nhóm, tạo khả năng lập lịch cho tất cả các thành viên nhóm, chẳng hạn cho 10 thành viên, được kiểm tra ngay lập tức mà không cần việc kiểm tra tách biệt. Chức năng này khử bỏ công việc cần thiết phải làm hiệu chỉnh về thời gian họp.
- Việc đạt tới những chức năng này đòi hỏi các chức năng xử lý tài liệu như bộ xử lý văn bản, và các chức năng cơ sở dữ liệu bên cạnh chức năng mạng. Trong một số trường hợp, những chức năng này được thực hiện bằng việc cung cấp giao diện với các sản phẩm hiện có. Từ nay trở đi, các hệ thống hỗ trợ nâng cao hơn đưa vào khái niệm về PDM (Product Data Management - Quản lý dữ liệu sản phẩm), quản lý các dữ liệu tập trung, sẽ đi vào sử dụng.

-

1.8.6 Công cụ OA

- Máy tính cá nhân đã được sử dụng nhanh chóng và rộng khắp trong các doanh nghiệp và ở gia đình bởi vì phần mềm cho phép ngay cả người không có kinh nghiệm cũng dễ dùng chúng, đã được cài đặt trong chúng. Trước đây, máy tính chỉ do các kỹ sư xử lý thông tin được đào tạo về CNTT vận hành. Tuy nhiên, với tiến bộ trong công nghệ thông tin, các chức năng xử lý đã được làm giàu thêm, và máy tính bây giờ đã trở thành sản phẩm quen thuộc như đồ điện tử gia dụng.
- Nói riêng, ba công cụ sau đóng vai trò chính trong tự động hoá văn phòng (OA).
 - Xử lý văn bản
 - Phần mềm trang tính
 - Phần mềm trình bày

(1) Xử lý văn bản

- Nhân tố lớn nhất làm tăng việc dùng máy tính cá nhân là việc đưa vào bộ xử lý văn bản. Việc đưa vào các chức năng xử lý văn bản đã biến máy tính từ "máy xử lý dữ liệu chỉ các chuyên gia mới có thể dùng được" thành "máy xử lý thông tin quen thuộc."

-

- ① Định dạng

- Mục đích quan trọng nhất của việc dùng bộ xử lý văn bản là để sinh ra tài liệu, qua việc đưa vào dữ liệu, trong đó các chức năng định dạng được dùng để quyết định cách bố trí tài liệu sẽ in ra.

- Việc định dạng phần lớn là xác định cỡ giấy được dùng và dáng vẽ của văn bản in.

- Đặt kiểu giấy được dùng

- Kích cỡ giấy được dùng phải được xác định.

- Đặt hướng in

- In theo chiều dọc giấy hay chiều ngang giấy cũng phải được xác định.

- Đặt số kí tự được in ra trên một dòng

- Số kí tự được in ra trên một dòng được xác định để theo đó xác định ra khoảng cách giữa các kí tự được in kề nhau.

- Đặt số dòng được in ra trên một trang

- Số dòng được in ra trên từng trang được xác định để theo đó xác định khoảng cách giữa các dòng kề nhau.

- Với bộ xử lý văn bản, có thể xác định một tập các định dạng và in ra với định dạng đó để xem cách các dáng vẽ và hình ảnh trông như thế nào, sau đó mới làm ra các đặc tả lại.

- ② Chức năng soạn thảo

- Khi các câu được trực tiếp viết ra giấy đã đúng rồi, phần cần được sửa sẽ bị xoá đi bằng bộ xoá, rồi từ mới hay câu mới được ghi đè lên. Tuy nhiên, các câu được sinh ra bằng bộ xử lý văn bản tồn tại trong bộ nhớ chính của máy tính. Do đó, việc sửa chúng là cực dễ.

- Bên cạnh chức năng sửa chữa, bộ xử lý đưa ra nhiều chức năng soạn thảo và xử lý khác.

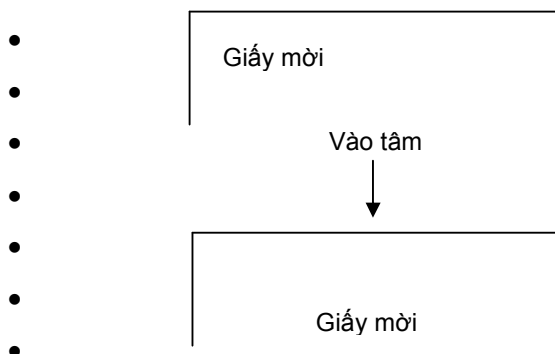
- a. Định tâm

- Bộ xử lý văn bản cung cấp khả năng đặt tiêu đề hay các câu hay dãy các kí tự khác theo chiều ngang hay chiều dọc vào vị trí trung tâm dựa trên số các kí tự trên một dòng được đặt trong định dạng. Chức năng này được gọi là "định tâm".

-

Hình 1-8-4

Ví dụ về định tâm



-

- b. Chuyển xâu kí tự sang bên phải nhất hay trái nhất

- Bộ xử lý văn bản cung cấp khả năng di chuyển xâu kí tự sang vị trí bên phải nhất hay trái nhất.

- Không có chức năng này, cần phải đưa vào "dấu cách" (bằng phím dấu cách) để đẩy xâu sang bên phải nhất hay dùng phím xoá để di chuyển xâu kí tự sang vị trí bên trái nhất của trang.

- c. Trang trí kí tự

- Bộ xử lý văn bản cho phép các dấu ký tự được gạch dưới, hay dùng phông Gothic cho chúng, hay phông cho tiêu đề được thay đổi, hay dùng phông nghiêng, khi được in ra. Những chức năng như vậy là có sẵn bởi vì các hình mẫu phông được ghi nhớ trong phần mềm được dùng.

Hình 1-8-5 Mẫu trang trí ký tự

- | |
|--|
| <ul style="list-style-type: none"> Hãy để trái đất xanh cho trẻ em Giấy tái sinh tiết kiệm gỗ. Hãy bắt đầu sống có ý thức tới trái đất để giữ cây xanh trên trái đất cho trẻ em thế hệ sau. <u>Một tấn giấy đã sử dụng tương đương với 20 cây sống.</u> |
|--|

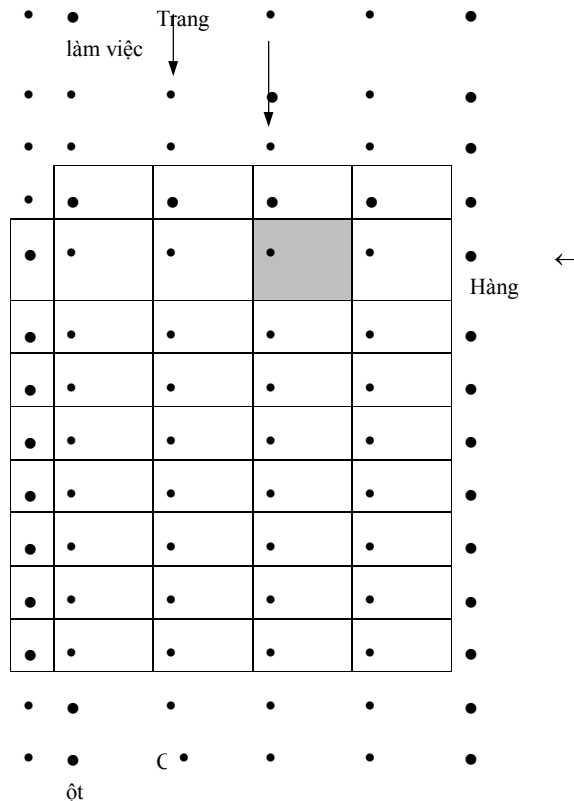
- d. Sao và di chuyển
- Các câu được sinh ra bằng bộ xử lý văn bản được cất giữ trong bộ nhớ chính. Do đó, các câu hay dấu ký tự trong những câu này có thể dễ dàng được sao chép hay di chuyển tới các vị trí khác.

(2) Chương trình trang tính

- Chương trình trang tính là để sinh ra các bảng (hai chiều) hay để kết tập dữ liệu. Nó cũng có thể bố trí lại dữ liệu hay diễn đạt chúng như đồ thị.

Hình 1-8-6

Trang làm việc và ô



① Trang công việc và ô

- Bảng được hiển thị trên màn hình được gọi là trang công việc, trong khi từng đơn vị không gian chữ nhật được gọi là ô. Trong trang công việc, một tập các ô được bố trí theo chiều ngang theo cùng mức được gọi là một hàng, trong khi tập các ô được bố trí theo chiều dọc trên cùng dòng được gọi là cột. Hàng được nhận diện bằng số, như 1,2,3,---, còn cột được nhận diện bằng các ký tự chữ

•	•	•	•	•
•	•	•	•	•

-
- Chương trình trang tính nhận diện bằng các kí tự đầu, liệu một chuỗi đưa vào là dữ liệu hay công thức. Nếu kí tự đầu là một kí hiệu để xác định công thức, như '+', '@', '=', v.v. (mỗi chương trình dùng các kí hiệu khác nhau cho mục đích này), thì các kí tự đi sau được coi như tạo nên công thức.
- Bên cạnh đó, nó cho phép dùng các hàm vào chỗ công thức. Chẳng hạn, trong Hình 1-8-7, "Total (D2 - D4)" chỉ ra hàm để kết tập dữ liệu từ ô D2 tới ô D4. Trong một số chương trình, biểu thức "Total (D2, D4)" được dùng thay vì "Total (D2 - D4)."
- ② Chức năng tính lại
 - Nếu dữ liệu trên trang làm việc bị thay đổi, thì kết quả của công thức hay hàm dùng dữ liệu này sẽ thay đổi. Kết quả cũng sẽ thay đổi khi cột và/hoặc hàng được thêm vào hay bị xóa đi. Chức năng này được gọi là "chức năng tính lại", và là một đặc trưng quan trọng của chương trình trang tính.
 - Với chức năng này, có thể cho chạy việc mô phỏng bằng cách thay đổi dữ liệu một cách đa dạng.
- ③ Chức năng soạn thảo
 - a. Sao chép hay di chuyển dữ liệu, và chèn thêm hay xóa dữ liệu
 - Việc sao chép dữ liệu được tiến hành theo cùng cách như trong bộ xử lý văn bản. Trước hết, vùng cần sao được xác định. Sau đó, vùng dữ liệu dự định được sao tới sẽ được xác định. Sau đó, việc sao chép được thực hiện. Việc chuyển dữ liệu có thể được thực hiện theo cùng cách. Việc chuyển khác với việc sao ở chỗ, khi chuyển, dữ liệu gốc bị mất khỏi vùng nguồn.
 - Cả sao chép lẫn di chuyển đều có thể được thực hiện cho dữ liệu trong một ô hay cho một miền xác định.
 - Mặt khác, việc chèn thêm và xóa đi có thể được thực hiện cho cả hàng hay cột.
 - b. Tự động điều chỉnh số ô
 - Tính cao cấp của chương trình trang tính là ở chỗ nó cho phép dữ liệu của ô bị ảnh hưởng bởi chức năng hiệu chỉnh được tự động điều chỉnh. Điều này được gọi là chức năng điều chỉnh tự động số các ô.
 - Chẳng hạn, giả thiết rằng có nhiều ô trong đó cùng phép tính cần được thực hiện. Với việc hiệu chỉnh này, có thể là công thức được đưa vào chỉ một trong những ô này, với công thức đó được sao cho các ô còn lại. Dữ liệu được tính toán bằng cùng công thức được đưa vào trong từng ô trong các ô này.
 - c. Bố trí lại dữ liệu
 - Dữ liệu được đưa vào không xét tới bất kì thứ tự nào có thể được bố trí lại trên cơ sở hàng, chẳng hạn, theo thứ tự dữ liệu giá trị lớn hơn được đặt trước giá trị nhỏ hơn; theo thứ tự dữ liệu giá trị nhỏ hơn được đặt trước dữ liệu giá trị lớn hơn, theo thứ tự lí tự "kana" của Nhật, thứ tự chữ cái; hay theo thứ tự tăng hoặc giảm của JIS.
- ④ Công thức và hàm

- Chương trình trang tính cho phép nhiều tính toán hơn là bốn phép toán số học cơ bản được sử dụng. Bên cạnh đó, các hàm cũng được nhúng vào để hỗ trợ cho các tính toán phức tạp.
- Các hàm khác nhau tùy thuộc vào các ứng dụng được cung cấp, kể cả các ứng dụng điển hình được liệt kê sau đây:
 - Các hàm số học: tính làm tròn, giá trị tuyệt đối và các hàm khác
 - Các hàm logic: AND, OR và các hàm khác
 - Các hàm thống kê: tính tổng, giá trị trung bình, tối đa và tối thiểu, và các hàm khác
 - Hàm ngày tháng: hiển thị dữ liệu ngày tháng, tính thời gian và các hàm khác
 - Các hàm tài chính: tính tỉ lệ lợi nhuận, đầu tư, chi phí khấu hao, và các hàm khác

(3) Phần mềm trình bày

Một cách chính thức, các công cụ trình bày chính là máy chiếu từ chiếu hay máy chiếu qua đầu. Tuy nhiên, gần đây chúng đã bị thay thế bằng các gói phần mềm có tên là phần mềm trình bày, cung cấp các chức năng còn nhiều hơn nhiều so với máy chiếu. Phần mềm trình bày cung cấp cả chức năng hoạt hình và chức năng âm thanh, và tạo khả năng trình bày hiệu quả bằng việc đem kết quả của chương trình trang tính hay cơ sở dữ liệu lên màn hình. Phần mềm trình bày tiêu biểu bao gồm PowerPoint của Microsoft, Freelance của Lotus và Appleworks của Macintosh.

Bài tập

- Q1 Điều nào trong những điều sau là thích hợp nhất cho đặc trưng của việc dùng công nghệ thông tin trong doanh nghiệp ngày nay?
 - A) Theo quan điểm sinh lợi, doanh nghiệp làm tăng mức độ chế tạo nội bộ mà không khoán ngoài các hệ thống hay vận hành cho các công ti ngoài.
 - B) Cái gọi là "tính toán người dùng cuối", trong đó tự bản thân người dùng xây dựng hệ thống và có quyền truy nhập hay xử lý thông tin cho ứng dụng riêng của mình, đang thu được sự ủng hộ.
 - C) Để cắt giảm chi phí hay để rút ngắn thời gian phát triển phần mềm, các ứng dụng đã được phát triển dựa trên đơn hàng nhận được thay vì dùng gói phần mềm.
 - D) Việc dùng mạng ngày càng rộng làm giảm phạm vi bị ảnh hưởng bởi hỏng hóc hệ thống, làm cho việc quản lí an ninh được dễ dàng hơn.
- Q2 Điều nào trong những điều sau là thích hợp nhất xem như việc giải thích cho mô hình thác đổ, một phương pháp luận phát triển hệ thống?
 - A) Ứng dụng được chia ra thành các đơn vị nhỏ. Sau đó từng đơn vị trong chúng được thiết kế và chế tạo tuần tự rồi cái nọ tiếp cái kia.
 - B) Việc phát triển hệ thống được thực hiện theo trật tự các tiến trình, không quay trở lại công việc ở tiến trình mức cao hơn.
 - C) Một sản phẩm thực nghiệm làm việc được, được sinh ra, và việc kiểm tra các đặc tả yêu cầu và ước lượng được thực hiện trong giai đoạn sớm.
 - D) Thời gian phát triển được rút ngắn bởi sự tham gia của người dùng, bằng việc phát triển với ít kĩ sư hơn và bằng việc dùng có hiệu quả các công cụ phát triển.
- Q3 Điều nào trong những điều sau là thích hợp nhất xem như việc giải thích cho làm bản mẫu, một phương pháp luận phát triển hệ thống?
 - A) Công việc được thực hiện theo thứ tự lập kế hoạch cơ sở, thiết kế ngoài, thiết kế trong, thiết kế chương trình, lập trình và kiểm thử. Do đó, với kĩ thuật này, người ta thu được viễn cảnh của công việc như một tổng thể, làm cho việc xác định lịch biểu và phân bổ tài nguyên được dễ dàng hơn.
 - B) Một sản phẩm thử nghiệm được tạo ra trong giai đoạn đầu của việc phát triển hệ thống, tạo khả năng loại bỏ sự mơ hồ và khác biệt trong hiểu biết của người dùng và tổ chức phát triển.
 - C) Phần mềm được phân chia thành phần mềm có đặc tả được ấn định và không yêu cầu sửa đổi, và phần mềm có đặc tả cần sửa đổi. Sau đó, với phần mềm có đặc tả yêu cầu thay đổi, tiến trình phát triển, xem xét lại và sửa đổi được lặp lại.
 - D) Một ứng dụng qui mô lớn được chia thành các đơn vị con, từng đơn vị này độc lập cao. Sau đó với từng đơn vị con, tiến trình thiết kế, lập trình và kiểm thử được lặp lại, mở rộng dần miền phát triển.
 -
- Q4 Từng câu sau đây mô tả công việc cho một trong các tiến trình phát triển hệ thống. Câu nào cho thứ tự đúng của tiến trình phát triển?

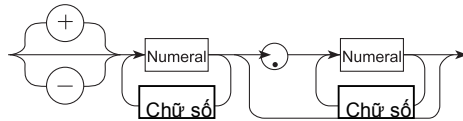
- A) Vấn đề hiện tại được khảo sát và phân tích, sau đó các yêu cầu về hệ thống đích được xác định.

- B) Các chức năng cần cho việc xây dựng hệ thống được phân hoạch thành các chương trình để làm cho luồng tiến trình được rõ ràng hơn.
- C) Các thủ tục xử lý được thiết kế, mã hoá và chỉnh sửa.
- D) Các kiểm thử được tiến hành.
- E) Thiết kế có cấu trúc của từng chương trình được thực hiện dựa trên tài liệu thiết kế trong.
- F) Dựa trên yêu cầu về hệ thống, các chức năng cần thiết cho hệ thống được xác định.
-
- A) a-f-b-c-e-d B) a-f-b-e-c-d
- C) a-f-e-b-c-d D) a-f-e-c-b-d
- Q5 Phát biểu nào trong những phát biểu sau là thích hợp nhất cho việc giải thích về kỹ nghệ phần mềm đảo?
- A) Các đặc tả thiết kế được tạo ra từ phần mềm đã cài đặt. Rồi phần mềm được phát triển dựa trên các đặc tả đã được tạo ra đó.
- B) Phần mềm được thiết kế để đưa ra, xử lý và đưa vào.
- C) Các chức năng đã được cài đặt bằng phần mềm được đạt tới bằng phần cứng.
- D) Một ngôn ngữ phát triển và công cụ phát triển được tuyển chọn tùy theo các đặc trưng xử lý của phần mềm.
- Q6 Với phương pháp phân tích có cấu trúc, các luồng dữ liệu và chức năng được diễn tả bằng các kí hiệu tương ứng chỉ ra luồng dữ liệu, xử lý (chức năng), lưu trữ dữ liệu và bên ngoài (nguồn dữ liệu và nơi nhận dữ liệu). Cái nào trong những cái sau là phương pháp ấy?
- A) DFD B) ERD C) Sơ đồ NS
- D) Biểu đồ chuyển trạng E) Biểu đồ Warnier
- Q7 Biểu đồ nào trong những biểu đồ sau là biểu đồ được dùng trong lập trình có cấu trúc và diễn tả toàn bộ cấu trúc của chương trình dưới dạng cấu trúc phân cấp?
- A) Sơ đồ NS B) Biểu đồ PERT
- C) Biểu đồ chuyển trạng thái D) Sơ đồ bọt
- Q8 Người ta gọi việc làm chi tiết cài đặt đối tượng vô hình bằng việc gắn dữ liệu và động pháp với nhau trong lập trình hướng đối tượng là gì?
- A) Thể hiện B) Bao bọc
- C) Kết cụm D) Trừu tượng hoá
-
- Q9 Phát biểu nào trong phát biểu sau là phù hợp nhất để lấy làm mô tả cho lập trình hướng đối tượng?
- A) Dữ liệu trao đổi giữa các đối tượng được thực hiện qua các thể hiện.
- B) Đối tượng chỉ ra các mô tả về đặc trưng lớp.

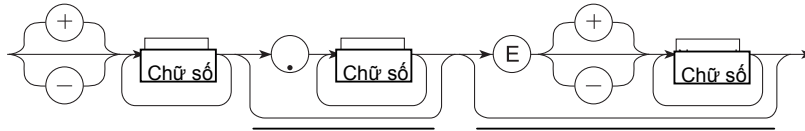
- C) Bao bọc chỉ ra việc gắn các lớp lại thành thư viện.
- D) Lớp có thể kế thừa các động pháp từ lớp bố mẹ.
- Q10 Cái nào là khoản mục không thích hợp được xem xét trong thiết kế màn hình của thiết kế ngoài và thiết kế trong?
 - A) Trong việc chuyển màn hình, một động pháp lựa trực tiếp được dự định dành cho người dùng có kinh nghiệm nên được dùng thay cho việc lựa từng bước bằng việc dùng menus.
 - B) Khoản mục đưa vào trên màn hình nên được bao bằng hay [] để làm rõ ràng rằng khoảng cách đó là dành cho trường đưa vào.
 - C) Bố trí màn hình nên được thiết kế sao cho các khoản mục được tham khảo có thể được bố trí từ trái sang phải hay từ trên xuống dưới.
 - D) Để hoàn thành một thao tác xử lý, thiết kế nên được làm sao cho việc dùng đưa dữ liệu vào và trở lại màn hình trước không thể được phép.
 - E) Bố trí màn hình được chuẩn hoá, chẳng hạn, các vị trí thống nhất cho hiển thị tiêu đề và thông báo, nên được dùng.
- Q11 Phát biểu nào trong các phát biểu sau là thích hợp nhất xem như mô tả về thiết kế mã và quản lý mã?
 - A) Mã không tránh khỏi thay đổi, cho nên điều quan trọng là đặt sách mã theo thứ tự và quản lý chúng.
 - B) Điều mong muốn là tự chúng có thể hiểu được. Do đó việc dùng mã dài là tốt hơn.
 - C) Số chủ yếu nên được dùng làm mã, và chữ Trung Quốc không nên dùng.
 - D) Mã nên được gán để làm cho việc phân loại dữ liệu dễ dàng hơn, nhưng việc bổ sung và mở rộng mã không nên được tính tới.
- Q12 Giả sử rằng mã bốn chữ số $N_1N_2N_3C$. Chữ C bên phải nhất chỉ ra chữ số kiểm tra, được tính bằng công thức sau đây.
 - $C = \text{mod} ((N_1 \times 3 + N_2 \times 2 + N_3 \times 1), 10)$
 - Tại đây, $\text{mod} (a, b)$ chỉ ra phần dư của a/b . Vậy, số cho \square là gì trong mã bốn số sau "81□6?"
 - A) 0 B) 2 C) 4 D) 6 E) 8
- Q13 Phát biểu nào trong các phát biểu sau là thích hợp nhất xem như lời giải thích về chương trình đồng dụng?
 - A) Chương trình cho kết quả đúng ngay cả nếu nó được thực hiện lại mà không phải nạp lại sau một lần đã thực hiện.
 - B) Được đặt vào bất kì địa chỉ nào trong bộ nhớ thật, chương trình đều có thể được thực hiện.
 - C) Chương trình được phân hoạch thành nhiều đoạn, và có thể được nạp vào và thực hiện trên cơ sở từng đoạn.
 - D) Cho dù nhiều nhiệm vụ thực hiện chương trình song song, ta vẫn thu được kết quả đúng.
- Q14 Phát biểu nào trong các phát biểu sau là thích hợp nhất xem như lời giải thích về lập

trình có cấu trúc, một nhân tố quan trọng trong thiết kế mô đun trong việc sinh chương trình?

- A) Lập trình có cấu trúc có nghĩa là đưa ra qui tắc viết tụt dòng cho việc viết mã để làm cho bản in chương trình gốc được dễ đọc.
- B) Lập trình có cấu trúc có nghĩa là dùng ghi chú một cách có hiệu quả để làm cho dễ hiểu phương pháp tiến trình chỉ bằng việc đọc chúng.
- C) Lập trình có cấu trúc được mô tả bằng ba cấu trúc cơ sở "tuần tự", "tuyển chọn" và "lặp"
- D) Lập trình có cấu trúc là đặt kích cỡ chuẩn cho mô đun phải là từ 50 đến 150 bước.
- Q15 Có cú pháp mô tả theo cấu trúc cú pháp sau đây. Các biểu thức như -100, 5.3, và +13.07 đáp ứng cú pháp này.



Khi phương pháp mô tả này được dùng, biểu thức nào trong những biểu thức số sau đáp ứng cho cú pháp được xác định bởi biểu đồ cú pháp sau?

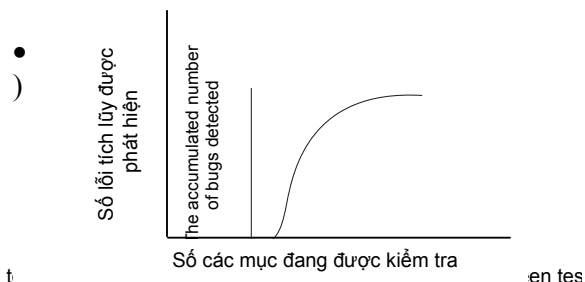
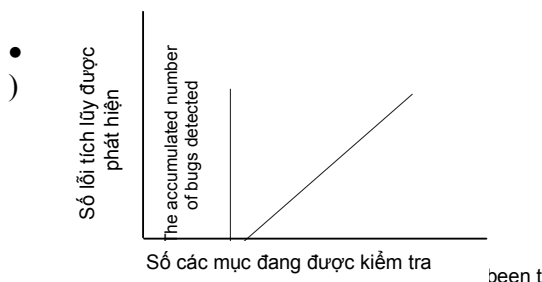
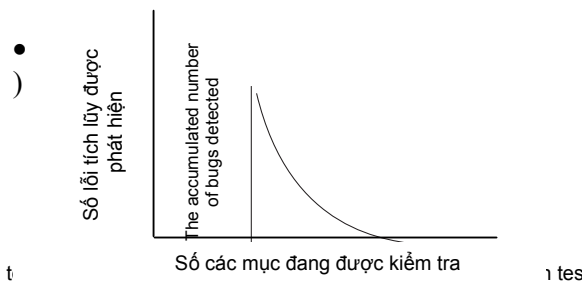
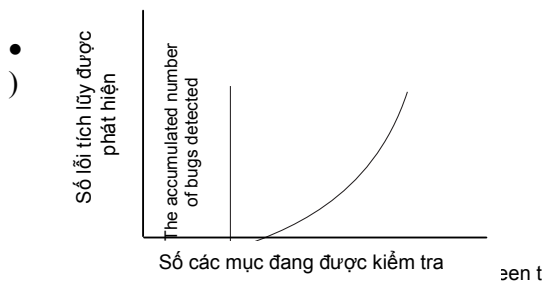


- A) 5.2E - 07 B) + 1.E4 C) - .9 D) 9.89E
- Q16 Điều nào trong những điều sau đây cho câu trả lời đúng cho công thức được mô tả dưới đây theo kí pháp Ba lan ngược. Ở đây, xy- chỉ ra rằng y là bị trừ đi khỏi x, trong khi xy÷ chỉ ra thương của phép toán xy÷.
- Công thức: 4 3 5 - ÷

• -2 • -0. • 0.2 • 0.5 • 5
)) 2))))

- Q17 Phát biểu nào trong các phát biểu sau là đúng khi giải thích về Java?
- A) Nó là giao thức truyền thông được dùng trên Internet.
- B) Nó là trình duyệt cho Internet.
- C) Nó là ngôn ngữ lập trình hướng đối tượng.
- D) Nó là kĩ thuật viết mã cho dữ liệu ảnh màu tĩnh.
- Q18 Phát biểu nào trong các phát biểu sau là phương pháp sinh trường hợp kiểm thử được dùng trong kiểm thử hộp trắng?
- A) Đồ thị nhân quả
- B) Phương pháp thiết kế thực nghiệm

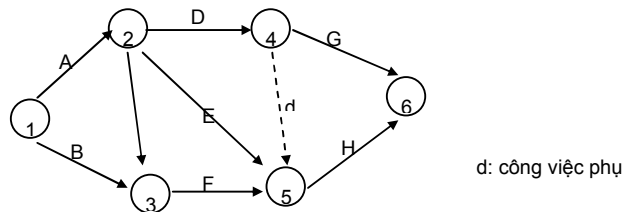
- C) Bao phủ điều kiện
- D) Phân hoạch tương đương
- Q19 Kiểm thử nào trong những kiểm thử sau là thích hợp nhất để được dùng trong phân tích giá trị biên?
 - A) Các giá trị cực đại và cực tiểu
 - B) Các giá trị cực đại và cực tiểu, và những giá trị cộng 1 vào các giá trị này
 - C) Giá trị cực tiểu và giá trị đó cộng với 1
 - D) Giá trị cực đại và giá trị đó cộng với 1
- Q20 Để duy trì chất lượng của tài liệu thiết kế, các cuộc họp kiểm điểm được tiến hành trong từng pha phát triển. Phát biểu nào trong những phát biểu sau là thích hợp nhất xem như giải thích cho việc giám định, một kỹ thuật kiểm điểm?
 - A) Kiểm điểm xem như một toàn thể được tiến hành với từng thành viên lần lượt giữ vai trò trách nhiệm.
 - B) Để kiểm điểm, một phần của phần mềm đích được tạo ra bằng thực nghiệm rồi được thực hiện thực tế.
 - C) Các khoản mục được kiểm điểm được lựa chọn trước. Sau đó tài liệu được kiểm điểm nhanh chóng bằng việc kiểm từng khoản mục mỗi lúc.
 - D) Người có thẩm quyền về tài liệu thiết kế được kiểm điểm làm chủ tịch cuộc họp kiểm điểm.
- Q21 Mối quan hệ giữa số khoản mục đã được kiểm thử và số lỗi tích lũy được dùng như khoản mục quản lý để kiểm tra tình trạng chất lượng của tiến trình kiểm thử. Đồ thị nào trong những đồ thị sau chỉ ra rằng chất lượng đang trở nên ổn định?



- Q22 Phát biểu nào trong phát biểu sau được phân loại là chức năng thượng lưu của công

cụ CASE?

- A) Chức năng phân tích chương trình nguồn
 - B) Chức năng phân tích và định nghĩa hệ thống
 - C) Chức năng hỗ trợ kiểm thử
 - D) Chức năng sinh chương trình tự động
 - E) Chức năng quản lý dự án
- Q23 Các đặc trưng chất lượng phần mềm bao gồm độ tin cậy, tính sử dụng, tính bảo trì và tính khả chuyển. Vậy, phát biểu nào trong các phát biểu sau giải thích cho độ tin cậy?
- A) Nó chỉ ra các thao tác có thể được làm chủ dễ dàng thế nào.
 - B) Nó chỉ ra liệu các chức năng được yêu cầu cho phần mềm bao giờ cũng có thể được bảo trì thông thường theo những điều kiện đã được chỉ định không.
 - C) Nó chỉ ra mức độ của sửa đổi trở nên cần thiết khi phần mềm được dùng trong một môi trường máy tính khác.
 - D) Nó chỉ ra mức độ dễ dàng theo đó các yêu cầu sửa đổi từ người dùng có thể được giải quyết.
- Q24 Có một dự án được mô tả bằng biểu đồ PERT sau. Kí hiệu trên từng mũi tên chỉ ra tên công việc, trong khi con số chỉ ra số ngày cần cho công việc đó. Ngày bắt đầu sớm nhất cho công việc H là gì? Giả sử rằng ngày bắt đầu của dự án là ngày 0.



- A)4 B)5 C)6 D)7 E)8
- Q25 Phát biểu nào trong những phát biểu sau là thích hợp nhất để lấy làm mô tả cho các đặc trưng của hệ thống tổ người lập trình chính so với hệ thống phân cấp?
- A) Khối lượng trọng trách của người lãnh đạo trong hệ phân cấp là nặng hơn trong hệ thống tổ người lập trình chính.
 - B) Hệ thống phân cấp ít thích hợp cho các hệ thống lớn hơn là hệ thống tổ người lập trình chính.
 - C) Trao đổi bên trong tổ hệ thống phân cấp là dễ dàng hơn trong hệ thống tổ người lập trình chính.
 - D) Hệ thống tổ người lập trình chính không cần người lập trình dự phòng.
 - E) Hệ thống tổ người lập trình chính cần một thủ thư.

2 Vận hành và bảo trì hệ thống

Mục đích của chương

Người ta gọi "vận hành" thực tế là cho chạy hệ thống đã phát triển, còn công việc giải quyết các vấn đề trong vận hành được gọi là "bảo trì". Đây là những việc cần tới liên tục cho tới khi hệ thống bị loại bỏ, áp đặt gánh nặng cho kỹ sư xử lý thông tin.

Chương này nêu ra đại cương về vận hành và bảo trì để xét hệ thống nên được phát triển thế nào, và để làm giảm gánh nặng vận hành và bảo trì nhiều nhất có thể được.

- ① Đại cương về vận hành và nội dung của từng khoản mục quản lý.
- ② Đại cương về bảo trì, và các kiểu và nội dung của công việc bảo trì.

Giới thiệu

- Vận hành và bảo trì hệ thống tạo nên pha cuối cùng của vòng đời phần mềm. Người ta quan tâm tới pha này lúc hệ thống được phát triển thực tế đang chạy, chiếm phần lớn nhất của vòng đời phần mềm. Do đó, ta sẽ thấy sự khác biệt lớn giữa hệ thống đưa ra việc vận hành và bảo trì hiệu quả với hệ thống cung cấp việc vận hành và bảo trì kém hiệu quả.
- Để đạt được vận hành và bảo trì hiệu quả, làm kế hoạch cho chúng sau khi hệ thống đã được phát triển là quá trễ. Các biện pháp cho vận hành và bảo trì nên được đưa vào trong nỗ lực phát triển hệ thống.
- Trong chương này, nội dung của vận hành và bảo trì được dạy nhằm mục đích chuẩn bị cho vận hành hữu hiệu và các kế hoạch bảo trì.

2.1 Vận hành hệ thống

- Vận hành hệ thống được thực hiện dựa trên các chuẩn dịch vụ vận hành. Các khoản mục quản lý cần cho vận hành hệ thống bao gồm:
 - Quản lý tài nguyên
 - Quản lý vấn đề
 - Quản lý tiện nghi
 - Quản lý an ninh
 - Quản lý hiệu năng
 - Quản lý chi phí
-

2.1.1 Quản lý tài nguyên

- Quản lý tài nguyên chiếm vị trí quan trọng trong các khoản mục quản lý vận hành. Để dùng tài nguyên hữu hiệu, cần phải có tri thức đúng về tài nguyên cần cho vận hành.
- Tài nguyên hệ thống bao gồm:
 - Tài nguyên phần cứng
 - Tài nguyên phần mềm
 - Tài nguyên dữ liệu
 - Tài nguyên mạng

(1) Quản lý tài nguyên phần cứng

- Quản lý tài nguyên phần cứng chỉ ra việc quản lý máy tính và các thiết bị ngoại vi của chúng. Tài nguyên phần cứng phải được bảo trì đúng.
- Thực tại, những nhà quản lý vận hành nên xác nhận rằng tài nguyên được dùng có hiệu quả

bằng việc kiểm tra các trang thiết bị phần cứng có được sử dụng hay không. Bên cạnh đó, nếu tìm thấy có sự thất thường, thì phải xem xét cách bố trí lại tài nguyên phần cứng để phân phối tải lượng tốt hơn. Khái niệm cơ sở là dùng tài nguyên có hiệu quả để làm tăng tỉ lệ vận hành của từng thiết bị phần cứng ngang nhau.

- Cuộc đời của tài nguyên phần cứng cũng nên được tính tới. Nói chung, trang thiết bị dùng quá một thời kì nào đó thường rất có thể gây ra vấn đề thường xuyên hơn. Xem xét việc thay thế các thiết bị bằng cách kiểm tra tỉ lệ phát sinh vấn đề là một khoản mục quan trọng trong quản lí tài nguyên phần cứng.
- Trong quản lí tài nguyên phần cứng, dữ liệu sau được thu thập và kết quả của chúng nên được ước lượng và phân tích đều đặn:
 - - Hiệu năng đáp ứng
 - - Khả năng xử lí (số khoản mục được xử lí trên mỗi giờ)

(2) Tài nguyên phần mềm

- Quản lí tài nguyên phần mềm chỉ ra việc quản lí chương trình đang chạy trong hệ thống. Ngược với tài nguyên phần cứng, nhiều bộ phận của tài nguyên phần mềm là không thấy được. Do đó, giữ các chuẩn xác định trước là quan trọng cho việc quản lí.
- ① Quản lí thư viện
 - Các khoản mục được đưa vào quản lí thư viện bao gồm:
 - Nơi thư viện (kể cả thư viện dự phòng) được lưu giữ về mặt vật lí phải được làm rõ ràng.
 - Dữ liệu phiên bản trong thư viện phải được quản lí (Phải tránh cùng tồn tại các phiên bản mới và cũ của cùng phần mềm).
 - Thư viện nên được bảo vệ (về an ninh và chống vi rút).
- ② Ngăn ngừa việc dùng trái phép
 - Các biện pháp sau đây nên được tính tới để ngăn cản việc sử dụng trái phép tài nguyên phần mềm:
 - Liệu việc sao chép bất hợp pháp có được thực hiện không hay không nên được quản lí.
 - Cách thức tài nguyên phần mềm được sử dụng nên được quản lí.

(3) Quản lí tài nguyên dữ liệu

- Quản lí tài nguyên dữ liệu chỉ ra việc quản lí và điều chỉnh dữ liệu trong hệ thống từ mọi quan điểm của tổ chức. Bản thân người dùng quản lí nhiều dữ liệu. Tuy nhiên, mục đích của quản lí tài nguyên dữ liệu là để quản lí dữ liệu này một cách hệ thống, và để lựa ra dữ liệu quan trọng cho việc quản lí đặc biệt nhằm hoàn thiện an ninh.
- Trong nhiều hệ thống ngày nay đều có sử dụng cơ sở dữ liệu. Do đó, việc quản lí tài nguyên cơ sở dữ liệu trên cơ sở vận hành nên được tính tới.
- Những điều sau đây nên được thực hiện trong quản lí tài nguyên dữ liệu:
 - - Hoàn thiện an ninh
 - - Đảm bảo an ninh (ngăn ngừa sử dụng trái phép)
 - - Quản lí có hệ thống tài nguyên dữ liệu

- Thêm vào đó, kiểm toán dữ liệu, được tiến hành để khảo sát và phân tích các phương pháp thực thi quản lý tài nguyên dữ liệu và để thực hiện việc quản lý ở mức độ hoàn hảo cao hơn, là một khoản mục quan trọng.

(4) Quản lý tài nguyên mạng

- Không phải nói quá rằng hệ thống máy tính ngày nay bao gồm cả việc được nối với mạng lưới. Trong quản lý tài nguyên mạng, các trang bị tạo nên mạng, như CCU (Communication Control Unit - đơn vị kiểm soát truyền thông, DCE (Data Circuit Terminating Equipment - thiết bị cuối mạch dữ liệu), v.v. được quản lý. Việc quản lý tài nguyên mạng về cơ bản được thực hiện dưới quản lý phân cứng. Tuy nhiên, đối với các mạch truyền thông, việc giải quyết các mạch chọn đường, bên cạnh mạch xương sống, cũng còn được kể tới. Do đó, tổ chức quản lý bao gồm các công ti viễn thông cũng phải được thiết lập.

•

2.1.2 Quản lý vấn đề

- Người ta mong muốn rằng không vấn đề nào xuất hiện trong vận hành hệ thống. Tuy nhiên, trong thực tế không có hệ thống nào mà không có vấn đề xuất hiện. Do đó, một khía cạnh quan trọng là hệ thống có thể được khôi phục nhanh chóng thế nào sau khi vấn đề xuất hiện. Việc quản lý vấn đề là về các biện pháp cần được thực hiện khi có vấn đề xuất hiện trong hệ thống.
- Các thủ tục chuẩn cần được tính tới khi vấn đề xuất hiện là như sau:
 - Tìm và báo cáo trực trực
 - Tạo ra báo cáo trực trực
 - Phân tích trực trực
 - Làm việc khôi phục từ một vấn đề
 - Công việc phục hồi hệ thống
- Sau khi việc phục hồi hệ thống được hoàn thành, cần đánh giá xem các biện pháp đã tiến hành có thích hợp hay không và suy nghĩ về kết quả của các biện pháp cần được lấy về sau.

(1) Tìm và báo cáo trực trực

- Vấn đề càng được tìm ra sớm, thì tác động của nó lên hệ thống xem như một tổng thể sẽ càng nhỏ và càng dễ lấy biện pháp đáp ứng. Do đó, để tìm ra vấn đề sớm nhất có thể được, điều quan trọng là luôn luôn chăm nom về dữ liệu được thu thập trong quản lý tài nguyên và hiểu thấu tình huống vận hành thông thường.
- Thêm vào đó, việc thiết lập một tổ chức cho phép trực trực được tìm ra được báo cáo ngay lập tức với người quản lý chịu trách nhiệm về vấn đề này là quan trọng.

(2) Tạo ra báo cáo trực trực

- Báo cáo trực trực phải được tạo ra ngay khi nhận được một thông báo rằng trực trực đã xuất hiện. Báo cáo trực trực có hai công dụng. Một là để phân tích vấn đề và lấy biện pháp đúng đắn và ngay lập tức; và hai là xem như dữ liệu thống kê được sử dụng để ngăn cản vấn đề từ trước.
- Thêm vào đó, vào lúc này, vùng bị ảnh hưởng bởi vấn đề cũng được nhận diện và lưu ý được tạo ra cho những bộ phận có liên quan. Nói riêng, những vấn đề được coi như có ảnh hưởng lớn tới sự vận hành của hệ thống thì cần sự hỗ trợ cho công tác khôi phục lại từ vấn đề này. Do đó,

việc lưu ý ngay lập tức là bản chất.

-

(3) Phân tích trực trực

- Để điều tra các nguyên nhân của vấn đề, việc phân tích được tiến hành trong những tình huống có xuất hiện vấn đề. Để điều tra nguyên nhân, cần dùng tới dữ liệu nhật ký vào lúc vấn đề xuất hiện và bản in xõ ra về trạng thái lúc đó. Nhiều nguyên nhân của vấn đề phần cứng được tìm thấy bằng những phương tiện này. Tuy nhiên, với các vấn đề liên quan tới phần mềm, việc tìm ra nguyên nhân đôi khi tốn nhiều thời gian. Trong trường hợp như vậy, các biện pháp tạm thời có thể được dùng, còn việc tìm kiếm nguyên nhân thực sẽ được bỏ lại về sau

- Các phương pháp sau đây được dùng như các biện pháp, nếu như dữ liệu nhật ký và dữ liệu in ra không đủ để tìm ra nguyên nhân:

- - Tình huống khi vấn đề xuất hiện được sinh lại một cách nhân tạo.
- - Một biện pháp được tiến hành, nếu những trực trực tương tự lại xuất hiện, cho phép thu được các dữ liệu chi tiết.
- Làm rõ nguyên nhân của vấn đề tạo khả năng ngăn ngừa sự xuất hiện của vấn đề tương tự lần nữa.

(4) Làm việc phục hồi từ một vấn đề

- Dựa trên nguyên nhân của vấn đề, các phương pháp phục hồi hệ thống được xác định và các thao tác phục hồi được thực hiện. Các phương pháp này tùy thuộc nhiều vào nguyên nhân vấn đề.

- ① **Trực trực phần cứng**
- Thiết bị dự phòng được đưa vào sử dụng.
- Các thiết bị có vấn đề được cô lập ra. Sau đó, chúng được sửa chữa (chẳng hạn, bằng việc chế tạo các thiết bị này).

- ② **Trực trực phần mềm**
- - Phần mềm được kích hoạt lại.
- - Phần mềm phiên bản cũ được khôi phục thể vào chỗ phần mềm phiên bản hiện thời.
- - Việc sửa đổi được thực hiện cho phần mềm hiện tại.

- ③ **Trực trực dữ liệu**
- - Dữ liệu gây ra vấn đề được loại bỏ hay sửa đổi.
- - Các thao tác Roll-back hay roll-forward được thực hiện.

- Thêm vào đó, việc gìn giữ các báo cáo về công tác khôi phục đã được thực hiện như thế nào sẽ tạo khả năng dùng lại chúng như những tư liệu để xem xét các biện pháp cho các vấn đề tương tự sẽ xảy ra hiện về sau.

-

-

(5) Công việc khôi phục hệ thống

- Hệ thống có các thao tác bị dừng lại được khôi phục. Nó được kiểm tra để xem liệu công việc khôi phục từ một vấn đề có làm cho hệ thống vận hành bình thường hay không. Sau đó hệ thống được khôi phục bằng các dịch vụ thông thường được cung cấp lại.
- Tùy theo cách được dùng để khôi phục từ một vấn đề, các nguyên nhân sau cũng nên được tính tới:
 - ① Trục trặc phần cứng
 - Điều sau đây nên được xét tới, nếu việc khôi phục được tiến hành bằng cách dùng phần cứng dự phòng:
 - - Hiệu năng so với phần cứng chính
 - - Công việc khôi phục khi việc sửa chữa phần cứng chính được hoàn tất
 - ② Trục trặc phần mềm
 - Điều sau đây nên được tính tới, nếu việc khôi phục được tiến hành bằng việc dùng một phiên bản phần mềm cũ hơn:
 - - Loại bỏ trong mức chức năng sẵn có (như các dịch vụ sẵn có)
 - - Giới hạn việc dùng khi tính tới khả năng đáp ứng.
 - ③ Trục trặc dữ liệu
 - Mục sau nên được xét tới, nếu dữ liệu đúng:
 - - Dữ liệu đúng là nhất quán với dữ liệu không đúng.
 - Công việc khôi phục hệ thống tiếp tục cho tới khi tất cả các chức năng của hệ thống được khôi phục đầy đủ.
 -

2.1.3 Quản lí tiện nghi

- Để vận hành một hệ thống máy tính, tiện nghi và trang bị của trung tâm máy tính phải được bảo trì trên một mức độ chất lượng nào đó. Xem như chuẩn thiết lập về các tiện nghi và trang thiết bị, Bộ Công nghiệp, thương mại và Kinh tế Nhật Bản (trước đây là bộ Công nghiệp và thương mại quốc tế) đã đặt ra bản hướng dẫn mang tiêu đề "Các chuẩn cho việc đo an toàn hệ thống tin."
- Trong các hoạt động điều tra (thiết kế) về trang thiết bị, ba khía cạnh về độ tin cậy, tính mở rộng được và chi phí phải được tính tới. Liên quan tới tính mở rộng được, thiết kế phải bao gồm các lễ trong việc xét cả tới những tiến bộ mới đây về công nghệ và các biến thiên của các nhân tố ngoài.
- Các tiện nghi cần xem xét trong vận hành hệ thống bao gồm những điều sau:
 - - Các tiện nghi liên quan tới nguồn điện
 - Các tiện nghi điều hoà nhiệt độ
 - Các tiện nghi ngăn ngừa thảm hoạ
 - Các tiện nghi ngăn ngừa tội phạm
 - Các tiện nghi cất giữ

(1) Quản lí liên quan tới nguồn điện

- Hệ thống máy tính không thể vận hành được nếu thiếu nguồn điện cung cấp. Do đó, phải cung cấp các tiện nghi liên tục cung cấp nguồn ổn định.

- ① **Nguồn điện chính**

- Thông thường, một hệ thống cấp nguồn thương mại được dùng làm bộ cấp nguồn chính. Tuy nhiên, một cơ chế cho việc cấp nguồn ổn định an toàn vẫn cần được yêu cầu. Thực tế, để giải quyết việc suy giảm chất lượng trong điện thế cấp từ hệ thống cấp nguồn thương mại, các biện pháp để duy trì điện thế cung cấp ở mức hằng số, kể cả việc dùng các tiện nghi của AVR (Bộ điều chỉnh điện áp tự động), được dùng tới.

- ② **Các tiện nghi điện không tiện ích**

- Tiện nghi điện không tiện ích được dùng làm dự phòng khi nguồn cấp điện chính có vấn đề (chẳng hạn, do điện áp ra). Tiện nghi này có thể được sử dụng làm nguồn cấp điện chính nếu hệ thống cấp nguồn thương mại không sẵn có. Tuy nhiên, thường nó được sử dụng chỉ khi vấn đề xuất hiện. Do đó, việc giám định để kiểm tra dầu máy phát và hỏng hóc của tiện nghi này phải được tiến hành đều đặn.

- ③ **UPS (Uninterrupted Power Supply - Nguồn không ngắt)**

- UPS (nguồn không ngắt) tạm thời được dùng vào lúc việc cấp điện từ nguồn điện chính bị dừng lại cho tới lúc nguồn điện chính được bắt đầu lại. UPS cũng đóng vai trò bù cho những khoảng ngắt ngắn trong nguồn cấp điện chính.

- ④ **Các tiện nghi khác**

- **Pin**

Thông thường, pin được nạp cho từng thiết bị trong khi có nguồn điện cung cấp. Trong một số trường hợp, pin được nạp bằng bộ nạp pin hay các phương tiện khác được cung cấp tách riêng theo thiết bị.

- **Tiện nghi phân phối nguồn**

Điều cũng quan trọng là giám định đều kì các bảng phân phối, các cầu chì, bảo vệ quá tải và những thiết bị khác có liên quan.

(2) Tiện nghi điều hoà nhiệt độ

- Phần lớn các thiết bị được dùng trong hệ thống đều sinh ra nhiệt khi vận hành. Cho dù nhiệt từ từng thiết bị là nhỏ, tổng lượng nhiệt được sinh ra trong phòng máy tính, nơi nhiều thiết bị vận hành, lại trở thành lớn đáng kể. Thông thường, nhiệt độ và độ ẩm tạo điều kiện vận hành ổn định được xác định cho từng phần thiết bị hệ thống. Do đó, việc cung cấp tiện nghi điều hoà nhiệt độ là quan trọng để duy trì điều kiện làm việc đúng.

- Có hai kiểu tiện nghi điều hoà nhiệt độ: kiểu tập trung và kiểu phân tán. Với kiểu tập trung, một máy điều hoà lớn được lắp đặt trong trung tâm máy tính, kiểm soát nhiệt độ của toàn bộ phòng. Kiểu tiện nghi điều hoà này đòi hỏi không gian lắp đặt nhỏ. Tuy nhiên, một khi tiện nghi điều hoà này hỏng, thì sự vận hành của cả trung tâm như một toàn thể có thể bị dừng lại. Tuy nhiên với kiểu phân tán, máy điều hoà được trang bị theo từng thiết bị. Do đó, kiểu tiện nghi này cung cấp tính mở rộng cao và cũng vẫn có giá trị khi việc hỏng nguồn được phân tán, cục bộ. Tuy nhiên, kiểu này có hơi kém hơn về tính hiệu quả của đầu tư so với kiểu tập trung.

- Thiết bị làm lạnh bằng nước có thể được dùng cho một tiện nghi sinh ra nhiều nhiệt, như máy tính chủ lớn. Ngày nay, nhiều thiết bị sinh ít nhiệt và có thể được dùng trong môi trường văn phòng bình thường. Kết quả là việc dùng các tiện nghi điều hoà sẽ có thể bị giới hạn cho các tiện nghi trung tâm.

(3) Tiện nghi phòng ngừa thảm họa

- Việc hoạt động ổn định của hệ thống đòi hỏi dùng các tiện nghi phòng ngừa thảm họa trong trường hợp có thảm họa, như cháy, động đất, v.v... Các tiện nghi ngăn ngừa thảm họa bao gồm những điều sau:

- ① Tiện nghi phòng cháy

- Tiện nghi phòng cháy bao gồm các báo động cháy (kiểm cảm biến nhiệt và kiểu cảm biến khói) và thiết bị dập cháy. Hệ thống bình phun không thể được dùng như thiết bị dập lửa trong các tiện nghi trung tâm. Do đó, thiết bị dập lửa dùng halogen hay khí CO₂ được dùng thay thế. Kiểu bộ dập lửa cho các đám cháy do điện gây ra là khác với đám cháy thông thường. Do đó, việc luyện tập dập lửa nên được tiến hành đều đặn.

- ② Tiện nghi chống động đất

- Xem như biện pháp trực tiếp chống động đất, nên dùng cách gắn cố định các thiết bị vào các then giữ chặt, hay thiết bị chống lung lay, và dùng thiết bị hấp thụ rung động. Tuy nhiên, điều đáng sợ khi động đất là đám cháy phụ. Để phòng ngừa đám cháy phụ, cần dùng thiết bị cắt nguồn bằng cảm biến rung động.

- ③ Thiết bị báo khẩn cấp

- Khi thảm họa xuất hiện, vai trò của thiết bị báo khẩn cấp công bố thông tin đúng là rất quan trọng.

- Tiện nghi ngăn ngừa thảm họa không được dùng trong những lúc thông thường. Do đó, việc tiến hành kiểm tra các tiện nghi này làm việc trong trường hợp khẩn cấp mà không có hỏng hóc là quan trọng. Đồng thời, điều mong muốn là cung cấp ra một cơ chế (như việc rèn luyện về thảm họa) để kiểm tra các biện pháp cần tiến hành trong trường hợp khẩn cấp cũng như các biện pháp truyền thông và các đường tiếp xúc.

(4) Tiện nghi ngăn ngừa tội phạm

- Tiện nghi ngăn ngừa tội phạm là để bảo vệ hệ thống khỏi sự đe dọa do con người gây ra như đánh cắp thông tin và các hoạt động phá hoại. Vì sự đe dọa do con người gây ra được đem tới từ người ngoài, nên các tiện nghi này, trong nhiều trường hợp, là để ngăn cản việc đi vào của những người như vậy.

- ① Thiết bị kiểm soát vào/ra

- Thiết bị kiểm soát vào/ra cho phép người quản lý đi vào và ra từ một tiện nghi trung tâm. Để nhận diện liệu một người có quyền vào hay ra khỏi tiện nghi trung tâm, việc xác thực về người này được thực hiện bằng mật hiệu hay nhiều mật hiệu, bằng thẻ vào/ra, như thẻ từ hay thẻ IC, dấu tay, tiếng nói hay mẫu võng mạc.

- Điều cũng quan trọng nữa là quản lý nhật ký vào/ra (tên của người vào hay ra khỏi trung tâm và thời gian vào/ra của họ. Biện pháp này là hiệu quả để ngăn ngừa những đe dọa do con người gây ra, những người có quyền vào trung tâm).

-

- ② Bộ giám sát

- Bộ giám sát được trang bị để quan sát một tiện nghi trung tâm để tìm ra người đáng ngờ và để ngăn ngừa những người này vào tiện nghi đó. Bộ giám sát cũng được lắp đặt bên trong một tiện

ngiht để kiểm tra liệu có người với hành vi đáng ngờ hay không. Trước đây nhiều bộ giám sát phải lắp đặt, bởi vì các bộ giám sát tĩnh đã được dùng. Tuy nhiên, nhiều bộ giám sát dùng ống kính rộng và có nhiều chức năng, kể cả chuyển động theo hướng và chức năng phóng to nhỏ, gần đây đã trở nên có sẵn. Do đó, có thể trông đợi một hiệu quả đủ với một số nhỏ các bộ giám sát được lắp đặt.

(5) Tiệm nghi lưu giữ

- Cho dù chức năng an ninh mức cao nhất được dùng để ngăn ngừa dữ liệu khỏi bị đánh cắp, chúng vẫn là vô nghĩa nếu bản sao lưu hay dữ liệu nào đó khác bị lấy ra dễ dàng. Phải dồn sự chú ý đầy đủ tới các tiệm nghi cất giữ bản sao lưu và các dữ liệu đưa ra khác của hệ thống.
- Tiệm nghi lưu giữ để dành cho việc lưu giữ các dữ liệu quan trọng. Do đó, tiệm nghi này phải được trang bị với các chức năng phòng ngừa thảm hoạ, như các chức năng chống cháy hay chống nước, bên cạnh chức năng an ninh được cung cấp để ngăn ngừa việc ăn trộm.
- Về cơ bản, tiệm nghi lưu giữ phải được xây dựng tại chỗ xa với tiệm nghi trung tâm. Thêm vào đó, kiểm soát vào/ra phải được tiến hành theo cùng cách như với tiệm nghi trung tâm.

2.1.4 Quản lí an ninh

- Mục đích của quản lí an ninh là để ngăn ngừa việc sử dụng hệ thống trái phép và việc dò rỉ thông tin trong vận hành hệ thống.
- Quản lí an ninh bao gồm những điều sau.:
 - Quản lí người dùng
 - Quản lí truy nhập
 - Quản lí việc dùng
- Công nghệ điển hình được dùng bao gồm:
- - Mật mã hoá

(1) Quản lí người dùng

- Việc quản lí người dùng của hệ thống thiết lập nên cơ sở cho tất cả việc quản lí an ninh. Để dùng hệ thống, cần phải có tên người dùng user ID do người quản trị hệ thống cấp. user ID là thông tin được trao cho từng người dùng với mục đích tạo khả năng kiểm tra tính xác thực của quyền người dùng trong việc dùng hệ thống và việc hiểu rõ trạng thái sử dụng của người dùng.
- Để quản lí người dùng user ID, cần xét tới các việc sau:
- Chỉ số hiệu ID của người dùng cần thiết mới được cấp. User IDs đã trở nên không cần thiết nên được xoá đi sớm nhất có thể được để ngăn cản việc dùng trái phép các ID đó.
- Việc dùng chung user ID phải bị cấm. ID khác nhau phải được cấp, ngay cả cho từng thành viên của tổ phát triển.
- Số lượng bộ có thẩm quyền đặt user ID nên nhỏ nhất có thể được.
- Mật hiệu để xác thực người dùng cũng phải được quản lí. Mật hiệu được sử dụng rộng rãi nhất như phương tiện xác thực người dùng.
- Để quản lí mật hiệu, những điều sau cần được xét tới:
- Nên khuyến khích dùng mật hiệu không thể dễ dàng đoán ra. Người quản lí phải kiểm tra mật hiệu đều đặn, và yêu cầu thay đổi nếu phát hiện mật hiệu cho phép dễ dàng đoán ra.

- Nên cung cấp cơ chế thay đổi mật hiệu đều đặn, chẳng hạn, bằng việc giới hạn thời hạn hợp lệ cho chúng.
- Mức độ an toàn của tệp tương ứng với mật hiệu phải được tăng lên bằng việc dùng mật mã hoá hay các phương tiện khác. Bên cạnh đó, phải cấm tham chiếu về các tệp này từ những người dùng chung.

(2) Quản lí truy nhập

- Trong quản lí truy nhập, những quyền truy nhập khác nhau có thể được thiết lập cho từng người dùng. Bằng việc làm như vậy, ngay cả với những người dùng của cùng một hệ thống, thông tin và/hoặc dịch vụ sẵn có có thể được đặt một cách khác nhau, dựa trên vị trí của người dùng.

(3) Quản lí việc dùng

- Trong quản lí việc dùng, việc thu thập dữ liệu sử dụng về hệ thống làm tăng mức độ an ninh của hệ thống như một tổng thể.
- Dữ liệu được thu thập bao gồm những điều sau:
 - Tên người dùng (user ID)
 - Ngày dùng
 - Thời gian dùng (thời gian đăng nhập/đăng xuất)
 - Thiết bị cuối được dùng
 - Hệ thống được dùng
 - Tài nguyên được dùng

(4) Mật mã hoá

- Bí mật của công ti được hệ thống giải quyết nên được mật mã hoá để ngăn cản chúng khỏi bị thay đổi. Mong muốn nhất là các dữ liệu thường trú trong hệ thống đều được mật mã hoá.

•

2.1.5 Quản lí hiệu năng

- Một mục đích trong việc quản lí hiệu năng của sự vận hành hệ thống là để kiểm tra xem liệu các dịch vụ được cung cấp cho người dùng có đáp ứng với các chuẩn được yêu cầu hay không. Đồng thời, việc tìm ra sự suy giảm hiệu năng cục bộ dẫn tới việc ngăn ngừa hỏng hóc hệ thống.
- Các khoản mục cần được quản lí bao gồm:
 - Thời gian đáp ứng và thời gian quay vòng
 - Hiệu suất
 - Thời gian có sẵn (thời gian bắt đầu và thời gian kết thúc)
 - Số tối đa các thiết bị cuối vận hành
 - Chất lượng của dữ liệu ra
 - SLA (Service Level Agreement thoả thuận mức độ dịch vụ) của mạng
- Việc thu thập và phân tích những dữ liệu này tạo khả năng cho người ta xác định xem liệu hiệu năng trông đợi đối với hệ thống có được duy trì hay không.

- Những phàn nàn của người dùng cũng nên chú ý, bởi vì về tiềm năng nó có thể chứa sự suy giảm hiệu năng, điều không thể được nhận diện bằng việc đo đơn giản.
- Thêm vào đó, những thay đổi trong các nhân tố ngoài (như việc tăng số các giao tác) có thể làm cho không thể duy trì được hiệu năng hệ thống trừ phi tiến hành biện pháp nào đó. Việc quản lý hiệu năng cũng nên được dùng một cách thường lệ để tạo khả năng dự đoán sự xuất hiện của những tình huống như vậy, và để cho phép đưa ra những đề nghị về thiết bị mới và để thay đổi phiên bản phần mềm.

-

2.1.6 Quản lý chi phí

- Việc vận hành hệ thống tự nhiên ngụ ý tới chi phí. Quản lý chi phí là rất quan trọng cho xí nghiệp trong việc tìm kiếm lợi nhuận, và do đó phải xem xét tới nó. Do đó, việc thu thập dữ liệu chi phí, và việc cắt giảm những chi phí không cần thiết nhiều nhất có thể được là quan trọng trong việc cho hệ thống chạy.
- TCO (Total Cost of Ownership - Tổng chi phí của quyền làm chủ) được dùng cho việc quản lý chi phí được phân loại thành hai khoản sau.
 - - Chi phí khởi đầu
 - - Chi phí vận hành

(1) Chi phí khởi đầu

- Chi phí khởi đầu là chi phí một lần trong pha thiết đặt hệ thống, và không được sinh ra sau khi việc vận hành hệ thống được bắt đầu.
- Các kiểu chi phí khởi đầu bao gồm:
 - - Chi phí mua thiết bị (thiết bị hệ thống, thiết bị mạng, thiết bị cuối, v.v.)
 - - Chi phí mua phần mềm (phần mềm cơ sở, gói phần mềm, v.v.)
 - - Chi phí phát triển phần mềm
- Những chi phí này không phải bao giờ cũng được sinh ra. Chẳng hạn, chi phí mua thiết bị là không cần đối với thiết bị thuê. Do đó, nên có xem xét cẩn thận về những chi phí này cho việc thiết lập hệ thống.

(2) Chi phí vận hành

- Chi phí vận hành trở thành cần thiết một khi hệ thống đi vào vận hành. Nó là chi phí thường xuyên được sinh ra đều kì và cố định.
- Các kiểu của chi phí vận hành bao gồm:
 - - Chi phí thuê thiết bị (thiết bị hệ thống, thiết bị mạng, thiết bị cuối, v.v.)
 - - Phí mua bằng sử dụng phần mềm (phần mềm cơ sở, gói phần mềm, v.v..)
 - - Chi phí bảo trì (về bảo trì phần cứng và phần mềm)
 - - Chi phí bảo trì thiết bị (chi phí truyền thông sóng, chi phí sưởi và ánh sáng)
 - - Chi phí có thể tiêu được
 - - Chi phí nhân sự

- Chi phí vận hành được phân loại thành chi phí cố định, được sinh ra thường xuyên, và chi phí lưu động và chi tiêu phụ thuộc vào hoàn cảnh. Chi phí lưu động và chi tiêu nên được quản lý trên cơ sở hàng tháng, để tìm ra sự khác biệt giữa dữ liệu thực và ngân sách. Sau đó, các biện pháp đúng được thực hiện, nếu cần.
- Phương pháp tính cước có thể được đưa vào trong đó người dùng (tổ chức người dùng) trả chi phí vận hành. Phương pháp tính cước bao gồm phương pháp cấp phát tỉ lệ, trong đó chi phí được cấp tùy thuộc vào khối lượng sử dụng, và phương pháp cấp phát cơ sở mà theo đó khối lượng được cấp là được xác định tùy theo tỉ lệ lợi nhuận thu được bởi việc dùng hệ thống.
-

2.1.5 Việc quản lý vận hành khác

(1) Vận hành hệ thống

- Theo quan điểm sử dụng, hệ thống phải được vận hành với việc xét tới những điều sau:
- Cung cấp tài liệu vận hành, mô tả phương pháp vận hành hệ thống và các thủ tục vận hành. Cung cấp phát biểu về kiểm soát công việc (lập lịch công việc) để tạo khả năng xử lý tự động công việc.
- Cung cấp việc đo tính toàn vẹn đối với việc đưa dữ liệu vào bên cạnh việc quản lý đưa vào và đưa ra dữ liệu.

(2) Công cụ vận hành hệ thống

- Các công cụ vận hành hệ thống đa dạng được dùng để làm cho vận hành hệ thống được trôi chảy và dễ dàng nhất có thể được.
- Các công cụ vận hành hệ thống điển hình bao gồm:
 - Công cụ cho phép toán tự động
 - Công cụ giám sát
 - Công cụ chuẩn đoán.

(3) Chuyển dịch hệ thống

- Việc chuyển dịch nghĩa là công việc chuyển vận hành của hệ thống từ một phiên bản cũ sang phiên bản mới. Việc chuyển dịch hệ thống được tiến hành với những thủ tục sau:
 - Chuẩn bị kế hoạch chuyển dịch
 - Chuẩn bị tài liệu thủ tục chuyển dịch
 - Thực hiện việc chuyển dịch
 - Kiểm thử vận hành
 - Chuyển sang pha vận hành (công việc được kế tục)
- Trong công tác chuyển dịch, chú ý phải được dồn vào việc quản lý dữ liệu phiên bản. Trong các hệ thống xử lý phân bố, công việc chuyển dịch có thể mất nhiều ngày. Trong những trường hợp như vậy, phải kiểm tra xem liệu có thể vận hành mà không gây trục trặc gì trong khi các phiên bản cũ và mới cùng được sử dụng hay không.

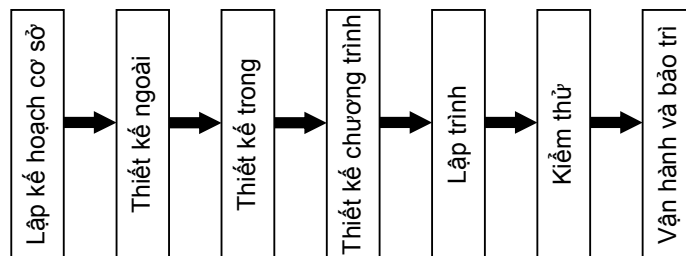
2.2 Bảo trì hệ thống

- "Vấn đề Y2K" đã hấp dẫn nhiều sự chú ý của ngành công nghiệp phần mềm. Điều này là vì các chương trình, vẫn vận hành bình thường cho tới cuối năm 1999, có thể sinh ra lỗi khi tới năm 2000. Không cần phải nói, phần mềm này phải sửa lại trước khi lỗi bật ra, nhưng lại không dễ phát hiện ra chương trình cần việc sửa chữa từ một khối lượng khổng lồ phần mềm. Tuy nhiên, đây là vấn đề phải được giải quyết. Do đó, phần mềm giúp giải quyết vấn đề này đã được phát triển và đưa ra thị trường. Vấn đề Y2K này đã có một tác động lên các kỹ sư xử lý thông tin như lời cảnh báo về khó khăn của công việc bảo trì.
- Khối lượng công việc bảo trì đã tăng lên năm nọ tiếp năm kia; với chi phí bảo trì bây giờ chiếm tới 60 đến 70 phần trăm của chi phí cho toàn bộ cuộc đời của hệ thống.
- Bước thứ nhất để đối phó với những tình huống này là làm đơn giản hoá công việc bảo trì. Việc đơn giản hoá công việc bảo trì là một trong những biện pháp trực tiếp đưa tới việc giảm đáng kể chi phí phát triển. Để đạt tới điều đó, việc trao đổi chặt chẽ và tiếp xúc thường xuyên với người dùng và việc xác định các biện pháp cần tiến hành khi trục xuất hiện, và tạo ra các tài liệu sử dụng mô tả các chi tiết để ngăn ngừa trục trặc là điều quan trọng.
- Điều lí tưởng là phát triển hệ thống để bảo trì chứa rất ít khiếm khuyết, và tạo khả năng giải quyết các vấn đề khác nhau được trông đợi trong tương lai.

2.2.1 Bảo trì là gì?

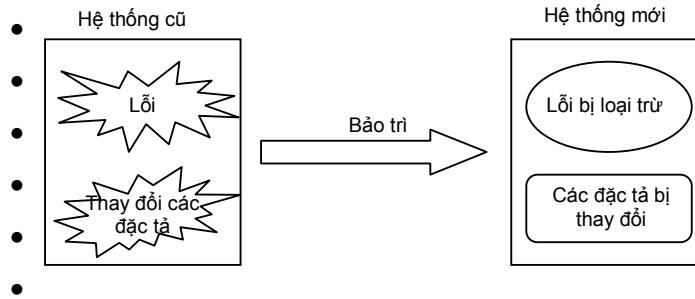
- Một hệ thống qui mô lớn phần lớn được phát triển theo mô hình thác đổ sau đây như được vẽ trong Hình 2-2-1:

Hình 2-2-1 Mô hình thác đổ



- Vận hành thực tế của hệ thống đã phát triển được bắt đầu sau khi các kiểm thử vận hành thấu đáo đã hoàn tất. Tuy nhiên, hệ thống (chương trình) phải được sửa lại nếu phát hiện ra lỗi còn ẩn trong hệ thống, hay khi người dùng yêu cầu thay đổi đặc tả hệ thống. Công việc liên quan tới những sửa chữa và thay đổi này được gọi là bảo trì (xem Hình 2-2-2).

Hình 2-2-2 Bảo trì

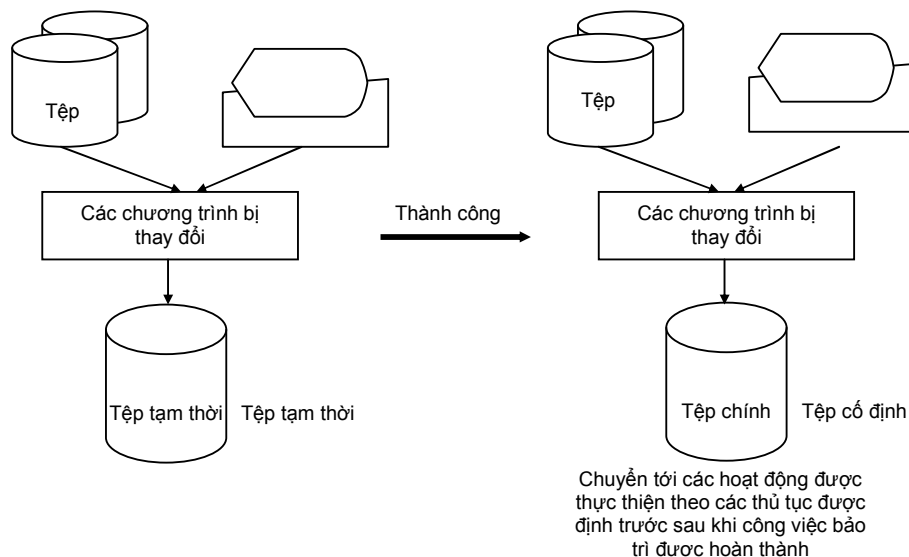


2.2.2 Tầm quan trọng của công việc bảo trì

Bảo trì là công việc quan trọng bao gồm việc khôi phục hệ thống về trạng thái bình thường bằng việc sửa chữa lỗi, giải quyết linh hoạt các yêu cầu thay đổi của người dùng. Công việc bảo trì nào đó được cần tới ngay tức khắc sau khi hoàn thành việc phát triển. Tuy nhiên, một số việc vẫn còn được đòi hỏi vài năm sau đó. Thêm vào đó, những sửa đổi phải được tiến hành cho hệ thống hiện có. Do đó, công việc này bao gồm nhiều rủi ro, đòi hỏi nhiều sự quan tâm (xem Hình 2-2-3). Bên cạnh đó, như đã mô tả ở trên, chi phí cho việc bảo trì và khối lượng công việc bảo trì đã được giữ lại cứ tăng lên qua từng năm. Trước khi bắt đầu công việc bảo trì, việc điều tra kĩ càng, xem xét các điểm sau, cần phải được tiến hành:

- < Các vấn đề cần được xem xét cho công việc bảo trì >
 - Các yêu cầu sửa đổi của người dùng có hợp lí không?
 - Người dùng khác có bị ảnh hưởng không?
 - Cấu hình hệ thống có cần phải thay đổi không?
- Nói riêng, phép kiểm thử để kiểm tra xem liệu việc sửa đổi do công việc bảo trì có ảnh hưởng tới các hệ thống khác không được gọi là "kiểm thử rà lại - regression test."

Hình 2-2-3 Sửa đổi hệ thống hiện có



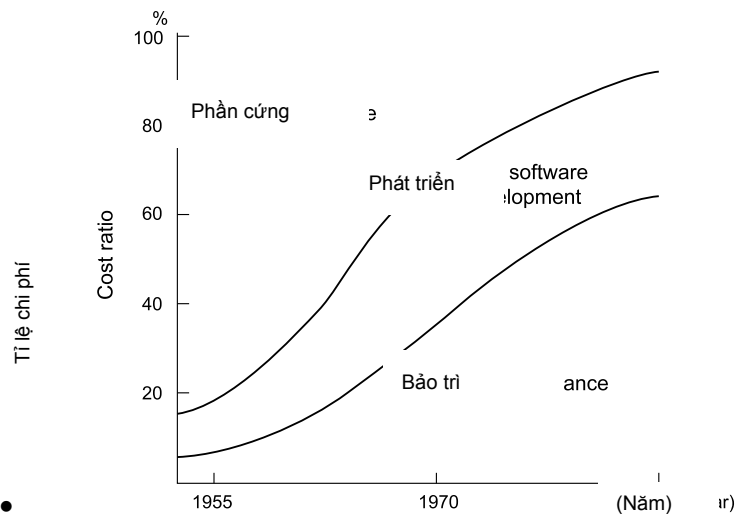
- Trong ví dụ của Hình 2-2-3, cái ra từ việc xử lý dùng các chương trình đã sửa được lưu giữ trong tệp tạm thời, thay vì vào tệp chính. Sau khi được xác nhận rằng tệp tạm thời vận hành bình thường, cái ra này được truyền cho tệp chính được dùng trong vận hành thực tế.
- Hình 2-2-3 chỉ ra một ví dụ về công việc bảo trì nên được tiến hành cẩn thận.
-

2.2.3 Chi phí bảo trì

- Hình 2-2-4 chỉ ra đường cong dịch chuyển của Boehm. Đường cong này chỉ ra tỉ lệ theo năm của chi phí bảo trì chiếm trong toàn bộ chi phí phát triển.
-

Hình 2-2-4

Đường cong dịch chuyển của Boehm



- Trước đây chi phí phần cứng là trội. Tuy nhiên ngày nay chi phí bảo trì đã trở nên trội khi chi phí phần cứng đã trở nên giảm thấp hơn.
- Công việc bảo trì gần như trở thành cần thiết không tránh khỏi. Do đó, chi phí bảo trì phải được rút lại nhiều nhất có thể được. Để đạt tới việc rút gọn này, các vận hành thông thường phải được tiến hành dựa trên tài liệu hệ thống của người dùng. Trong trường hợp trực trực xuất hiện, phải cung cấp tài liệu mô tả cho những tình huống đó, nguyên nhân và các biện pháp đối phó.
-

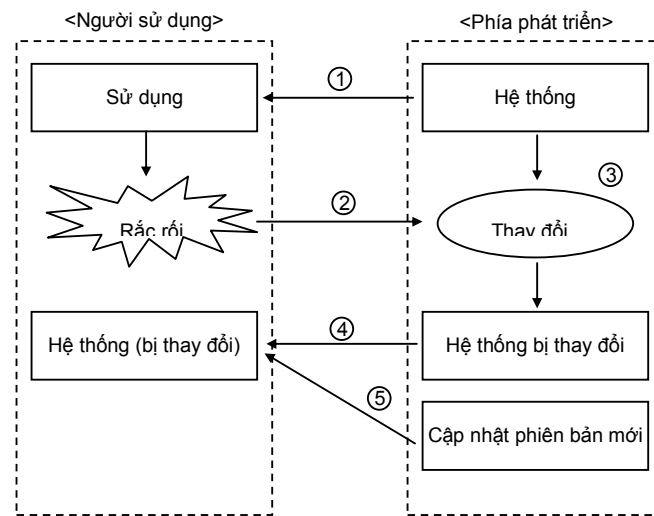
2.2.4 Nhiệm vụ bảo trì

- Sau đây, công việc bảo trì xem như một tổng thể sẽ được mô tả định tâm vào mối quan hệ giữa phía người dùng và phía phát triển.

(1) Trao đổi giữa phía người dùng và phía phát triển

- Hình 2-2-5 chỉ ra một loạt những trao đổi giữa phía người dùng và phía phát triển trong việc thiết lập hệ thống mới và việc bảo trì của nó.
-

Hình 2-2-5 Trao đổi giữa phía người dùng và phía phát triển



- Về cơ bản, những trao đổi sau đây được thực hiện giữa phía người dùng và phía phát triển.
- Phía phát triển trao hệ thống (kể cả tài liệu sử dụng) cho phía người dùng.
- Người dùng sử dụng hệ thống, và lưu ý với phía phát triển mọi lúc trục trặc xuất hiện, và yêu cầu sửa đổi về đặc tả.
- Phía phát triển điều tra những yêu cầu này, và tiến hành công việc sửa đổi cho những trục trặc đã xuất hiện, và trả lời cho yêu cầu của người dùng.
- Phía phát triển gửi tới phía người dùng dữ liệu đã sửa và/hay chương trình đã sửa.
- Phía phát triển đều đặn gửi các chương trình phiên bản mới (theo phiên bản) cho người dùng.

(2) Các biện pháp cần được tiến hành từ phía phát triển và phía người dùng

- Những biện pháp nào mà phía người dùng và phía phát triển cần tiến hành tương ứng, trong trường hợp chi phí bảo trì tăng lên? Sau đây các biện pháp được mô tả từng bước một.
- ① Phải tiến hành nỗ lực giảm công việc bảo trì
 - a. Phía phát triển: Phải hiểu rõ mức độ bất tiện và khó khăn người dùng phải chịu từ những trục trặc, và hiệu năng kém của hệ thống
 - Xem như một ví dụ, vào tháng 8/1997, vận hành của hệ thống trao đổi chứng khoán ở thị trường chứng khoán Tokyo đã dừng lại trong hai giờ, dừng việc bán và mua hơn 1,700 cổ phần trong nửa ngày (xem Hình 2-2-6).

- Hình 2-2-6 Một loạt các biến cố giữa một lần xuất hiện trục trặc ở thị trường chứng khoán Tokyo và việc kết thúc công việc bảo trì

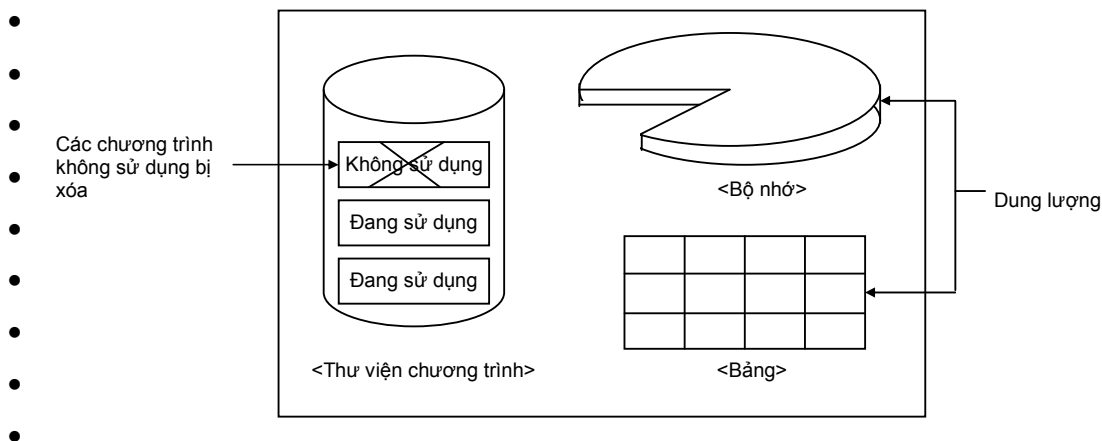
•	1	•	•
•	/8		
•	6	•	Hệ thống chứng khoán đi vào vận hành
•	:50	•	
•		•	
•	8	•	Việc nhả chứng khoán được bắt đầu
•	:20	•	
•		•	
•	8	•	Vận hành của TRS (một qui định liên quan tới hệ thống) đột ngột bị dừng, và hệ thống được chuyển sang hệ thống dự phòng.
•	:21	•	
•		•	
•	8	•	Vận hành của hệ thống dự phòng cũng bị dừng, dẫn tới việc vận hành của TRS dừng hoàn toàn.
•	:22	•	Chương trình được nạp lại, và hệ thống dự phòng được kích hoạt lại bằng "IPL nhanh."
•		•	
•	8	•	Hệ thống dự phòng được kích hoạt lại bằng "IPL nhanh " lại bị dừng.
•	:31	•	"IPL nhanh" được lặp lại hai lần. Tuy nhiên, vận hành hệ thống đều không được cho mỗi lần.
•		•	
•	8	•	Vận hành của hệ thống nhận đơn bị đưa tới chỗ dừng.
•	:50	•	
•		•	
•	9	•	Các giao dịch cho phiên buổi sáng đã được bắt đầu.
•	:00	•	
•		•	
•	9	•	Vận hành để đưa hệ thống trao đổi chứng khoán vào sử dụng đã được tiến hành lại một lần nữa (sau khi xoá các tệp), sau đó hệ thống được kích hoạt lại.
•	:20	•	
•		•	
•	1	•	Người ta xác nhận rằng vận hành bình thường của hệ thống đã được khôi phục.
•	0:30	•	
•		•	
•	1	•	Giao tác cho phiên buổi sáng được đóng.
•	1:00	•	
•		•	
•	1	•	Vận hành nhận đơn cho buổi chiều được bắt đầu 45 phút sớm hơn bình thường.
•	1:30	•	
•		•	
•	1	•	Các giao tác cho tất cả các chứng khoán được bắt đầu cho phiên buổi chiều.
•	2:30	•	
•		•	

• G iữa đêm	• Vào giữa đêm việc điều tra nguyên nhân được hoàn tất. Chương trình được sửa đổi để ngăn ngừa kích hoạt "tính toán trung gian các giá trị chứng khoán ngoại," làm lấy cò trục trặc.
• 2 /8	• •
• 1 2:30	• Kiểm thử hệ thống đã sửa đổi trên máy kiểm thử được hoàn tất.
• 1 8:00	• Hệ thống được sửa đổi được vận hành trong môi trường kiểm thử (hệ điều hành) của máy thực.
• 3 /8	• •
• 1 5:30	• Xác nhận cuối cùng về vận hành máy thực được tiến hành trong môi trường thực.

• • * IPL: Tải chương trình ban đầu

- Thị trường chứng khoán Tokyo là một trong những thị trường chứng khoán nổi bật trên thế giới, cũng như ở Nhật. Việc dừng hệ thống như vậy là một vấn đề quốc tế, ảnh hưởng đáng kể tới thế giới.
- Nguyên nhân được báo cáo lại là do việc dừng chạy của các chương trình do xuất hiện một biến cố ngẫu nhiên.
- Với vận hành bình thường, phía phát triển phải tiến hành các kiểm thử dồn dập bao quát tất cả các trường hợp có thể quan niệm được. Tuy nhiên, việc xoá các chương trình không cần thiết và kiểm tra khả năng bộ nhớ và băng, cũng là những công việc quan trọng cần được thực hiện đều đặn.

Hình 2-2-7 Xoá chương trình và khả năng kiểm tra



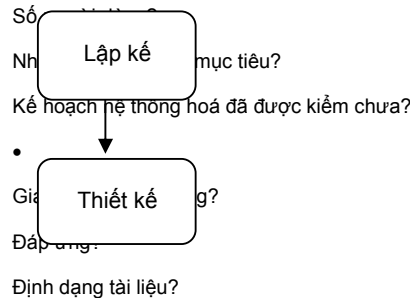
- b. Phía người dùng: Yêu cầu sửa đổi các chức năng và khả năng vận hành sau khi các thao tác thực được bắt đầu, và việc cải tiến hiệu năng do tăng khối lượng sử dụng.
- Trong khi dùng hệ thống, các yêu cầu để cải tiến chức năng và khả năng vận hành, hay các yêu cầu cải tiến sự đáp ứng vốn bị giảm đi do việc tăng số người dùng và khối lượng lưu thông, sẽ được thực hiện. Việc sửa đổi ở mức độ nào đó sẽ là không tránh được do sự thay đổi xã hội hay các

nhân tố khác. Tuy nhiên, sẽ có một số thay đổi có khả năng tránh được nếu trao đổi kỹ lưỡng với phía người dùng được tiến hành dựa trên các pha lập kế hoạch cơ sở và thiết kế ngoài của việc phát triển hệ thống. Việc có những cuộc họp đủ với phía người dùng trong xem xét về tương lai là cần thiết trong pha thiết kế hệ thống.

-

Hình 2-2-8

Trao đổi trong pha thiết kế



-

- Điều tra thấu đáo là cần với sự tham gia của người dùng

- Thêm vào đó, nếu việc sửa đổi là không tránh khỏi, thì điều quan trọng là kiểm tra đầy đủ tính hợp lệ của việc bảo trì.

- ② Nỗ lực làm cho việc bảo trì được trôi chảy

- a. Làm tài liệu về trực trặc trong khi hình thành

- Người dùng phải làm tài liệu chi tiết cho vấn đề cho dù nó có thể được coi như tầm thường. Mô tả trực trặc, chẳng hạn, theo định dạng được thống nhất trong toàn công ti là tốt hơn chỉ báo cáo miệng. Điều này là vì các mô tả này sẽ trở nên rất có ích cho việc lấy biện pháp phòng ngừa chống lại những trực trặc tương lai.

- b. Quản lý toàn diện việc ghi sổ trực trặc

- Cần phải cung cấp một bộ ghi trực trặc, tạo khả năng cho việc quản lý thống nhất các trực trặc trong hệ thống. Bộ ghi trực trặc có thể được cung cấp và quản lý cho từng người dùng hay từng hệ thống. Tuy nhiên, theo quan điểm hiệu quả, người ta mong ước cung cấp một phương pháp quản lý thống nhất trong toàn công ti.

- c. Phân tích trực trặc, phân tích hiệu năng và cải tiến sửa đổi, và quản lý tài liệu thiết kế và chương trình nguồn

- Việc bảo trì hệ thống đòi hỏi nhân lực nhiều hơn các kiểu công việc khác. Khối lượng nhân lực so với trực trặc thông thường là 10 người-ngày. Không cần phải nói, công việc đối với một trực trặc khôi phục điều được coi là khó thì tốn nhiều thời gian hơn. Trong phần lớn việc khôi phục, cần dùng tới nhiều tài liệu thiết kế và chương trình nguồn. Do đó, các tài liệu thiết kế dễ hiểu phải được cung cấp trong pha thiết kế khởi đầu có xem xét tới việc dùng về sau. Các chương trình nguồn cũng phải được viết dưới dạng cho phép dễ dàng gỡ lỗi, chẳng hạn, bằng việc dùng phương pháp lập trình có cấu trúc.

- d. Duyệt lại tài liệu làm việc

- Hệ thống do con người vận hành. Do đó, cho dù công việc được thực hiện đúng đắn theo những hướng dẫn tài liệu, lỗi vẫn có thể có trong dữ liệu đưa vào hay do việc vận hành máy tính không đúng. Để tránh việc tạo ra những lỗi như vậy, tài liệu phải được duyệt lại để cho việc xuất hiện lỗi có thể được phòng ngừa nhiều nhất có thể được. Thêm vào đó, tự nhiên cũng phải chăm nom tới những người thiếu kinh nghiệm vận hành, như nhân viên mới, và việc đào tạo bên trong công ti phải được tiến hành tốt.

- e. Duyệt dự án để bổ sung chức năng và cải tiến hiệu năng
- Hệ thống, khi đã hỗ trợ đầy đủ nhu cầu của người dùng lúc khởi đầu, sẽ trở thành kém tiên tiến qua thời gian. Do đó, việc cập nhật phần mềm là bản chất. Dự án phải được tổ chức để điều tra cách hệ thống được dùng, và hệ thống phải được cập nhật để nâng cao tính hữu ích của hệ thống.

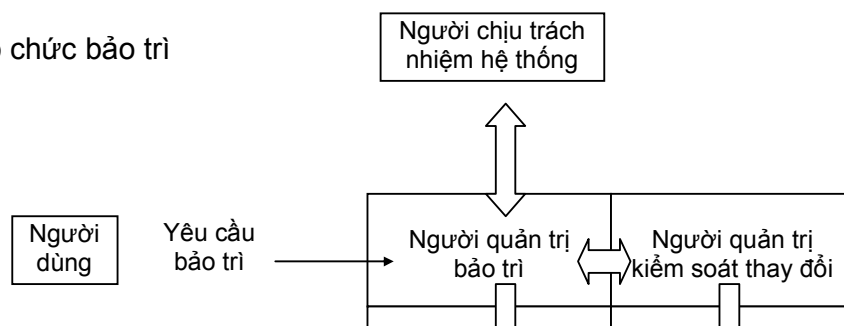
(3) Nhiệm vụ bảo trì

- Các nhiệm vụ bảo trì thực tế được phân loại thành ba nhiệm vụ sau.
 - Nhiệm vụ sửa chữa**
 - Lỗi trong chương trình và hệ thống được loại bỏ. Nói riêng, một hành động khẩn cấp là cần cho OLTP (OnLine Transaction Processing - xử lý giao tác trực tuyến).
 - Nhiệm vụ sửa đổi**
 - Dữ liệu như tên thời đại, tỉ lệ lãi, tỉ lệ thuế và khoản mục dữ liệu được thay đổi tùy theo những thay đổi xã hội và khi nhu cầu nảy sinh.
 - Nhiệm vụ cải tiến**
 - Hệ thống được cải tiến, chẳng hạn, qua việc thay đổi đặc tả do yêu cầu của người dùng.
- Trong số những nhiệm vụ này, khối lượng nhiệm vụ cải tiến gần đây đã tăng lên. Đó là vì các yêu cầu của người dùng, như việc tăng tốc độ xử lý, và với thay đổi hay sửa đổi giao diện làm cho họ dễ dùng hơn, đã trở nên đa dạng.

2.2.5 Tổ chức bảo trì

- Tổ chức bảo trì nghĩa là tổ chức chịu trách nhiệm thực hiện công việc bảo trì khi việc bảo trì được yêu cầu cho một hệ thống.
- Để thiết lập tổ chức bảo trì, những xem xét đặc biệt phải được tính tới như sau:
- Tiến hành công việc bảo trì cũng như làm nỗ lực phát triển đồng thời là không thể được. Tổ chức phát triển và tổ chức bảo trì phải được thiết lập tách biệt nhau.
- Điều mong muốn là người chịu trách nhiệm bảo trì có thể được đưa vào trong việc phát triển hệ thống.
- Cái ra từ việc phát triển hệ thống bao gồm tài liệu như tài liệu thiết kế, bên cạnh bản thân hệ thống. Do đó, một hay nhiều người chịu trách nhiệm quản lý tài liệu phải được bổ nhiệm để tạo ra việc quản lý thống nhất cho những tài liệu này.
- Một người (người quản trị bảo hành) làm nhiệm vụ nối liên hệ giữa người dùng và tổ chức phát triển phải được bổ nhiệm. Số những liên hệ đó phải được giới hạn là một.
- Nếu người phát triển thực hiện việc bảo trì theo ý riêng của mình, thì vấn đề nghiêm trọng có thể xảy ra sau đó. Do đó, một nhóm phải được tổ chức để thực hiện công việc bảo trì.
- Sau đây, tổ chức bảo trì thực tế và vai trò mỗi thành viên nắm giữ sẽ được mô tả.
- Hình 2-2-9 chỉ ra ví dụ về tổ chức bảo trì.

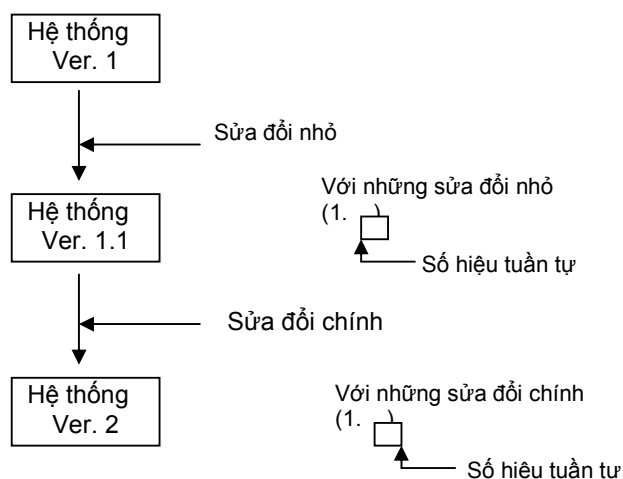
Hình 2-2-9 Tổ chức bảo trì



đổi.

-
-
-
-
-
-
-
-
-
-

Hình 2-2-10
Phiên bản cập nhật

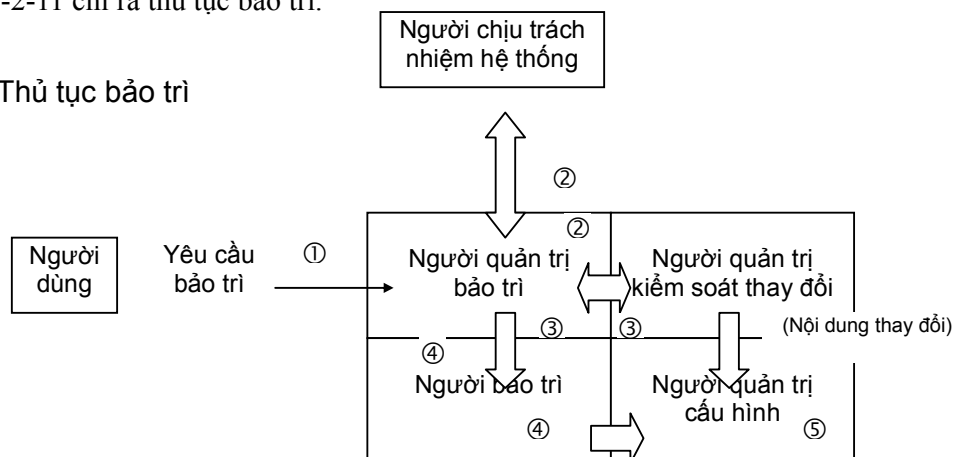


(2) Thủ tục bảo trì

- Hình 2-2-11 chỉ ra thủ tục bảo trì.

Hình 2-2-11 Thủ tục bảo trì

-
-
-
-
-
-
-
-
-
-



- Công việc thực tế là như sau:

Người dùng yêu cầu bảo trì với người quản trị bảo trì.

Sau khi nhận được yêu cầu này, người quản trị bảo trì kiểm tra, thảo luận về tính hợp lệ, và ra quyết định cuối cùng cùng với người quản trị kiểm soát thay đổi và người chịu trách nhiệm về hệ thống.

Một khi việc thực hiện công việc bảo trì được quyết định, thì người quản trị bảo trì nêu hướng dẫn cho những người bảo trì tiến hành công việc. Bên cạnh đó, từng việc sửa đổi hệ thống đều được làm tư liệu và quản lý.

Công nhân bảo trì báo cáo trạng thái tiến độ công việc của họ cho người quản trị bảo hành vào những thời gian thích hợp và việc hoàn thành công việc của họ cho người quản trị bảo trì và người quản trị cấu hình.

Người quản trị cấu hình cập nhật phiên bản của hệ thống đích.

-

2.2.6 Các kiểu bảo trì

- Các kiểu bảo trì bao gồm bảo trì phòng ngừa, cho phép các kế hoạch được lập ra trước, và bảo trì hậu kỳ, được thực hiện khi hỏng hóc xuất hiện.

(1) Bảo trì phòng ngừa

- Bảo trì phòng ngừa được thực hiện để ngăn cản hỏng hóc, còn kế hoạch bảo trì thì được lập ra từ trước. Bảo trì phòng ngừa, cho phép kiểm nhân sự bảo trì theo kế hoạch, và tạo khả năng cho bảo trì hiệu quả.
- Bảo trì phòng ngừa bao gồm:
 - - Bảo trì hàng ngày
 - - Bảo trì theo lịch
- ① Bảo trì hàng ngày
 - Bảo trì hàng ngày được tiến hành mọi ngày để kiểm tra trạng thái và hiệu năng của thiết bị tạo nên hệ thống. Thông thường, các biện pháp cho phép các kiểm tra được tiến hành mà không làm dừng vận hành hệ thống được sử dụng. Việc bảo trì từ xa được dùng rộng rãi trong hệ thống có các vị trí xa nhau.
- ② Bảo trì theo lịch
 - Việc bảo trì được tiến hành vào những khoảng thời gian xác định trước.
 - Việc bảo trì này là công việc qui mô tương đối lớn mà trong đó vận hành của hệ thống hoặc là bị dừng lại hoặc được thực hiện bằng thiết bị thay thế. Vì việc vận hành bị dừng lại tạm thời nên công việc phải được tiến hành khẩn trương.

(2) Bảo trì về sau

- Bảo trì về sau được tiến hành khi một hỏng hóc xuất hiện, hay có thể xuất hiện.
- Bảo trì không phải là công việc được lập lịch từ trước. Do đó, việc lấy người bảo trì cho công việc có thể khó. Trong trường hợp một tình huống như vậy xuất hiện, việc có nhân viên bảo trì khẩn cấp là điều mong muốn.
- Bảo trì sau bao gồm:
 - - Bảo trì thăm dò
 - - Bảo trì khẩn cấp
- ① Bảo trì thăm dò
 - Bảo trì thăm dò được tiến hành chỉ khi một tình huống bất thường được tìm thấy trong vận hành hệ thống, như sự suy giảm bất thần về hiệu năng hay âm thanh bất thường được phát hiện tới từ thiết bị đang vận hành.
 - Mục đích của bảo trì này là lấy các biện pháp trước khi hỏng hóc thực tế xảy ra.
- ② Bảo trì khẩn cấp
 - Bảo trì khẩn cấp được tiến hành để làm việc phục hồi từ hỏng hóc khi nó xuất hiện. Công việc này nảy sinh khi dừng tạm thời hệ thống mà không báo trước.

- Sự cần thiết của việc bảo trì khẩn cấp như vậy nên được tránh nhiều nhất có thể được. Do đó, nếu số công việc bảo trì khẩn cấp có khuynh hướng tăng lên, thì các kế hoạch bảo trì, kể cả việc xem xét lại hệ thống như một toàn thể, phải được đặt ra và thực hiện.
-
- Việc tiến hành bảo trì tự nhiên đòi hỏi phải có nhân sự bảo trì. Tuy nhiên, việc duy trì nhân sự này có thể đặt ra vấn đề với chi phí. Do đó, công việc bảo trì có thể được uỷ thác cho các công ti ngoài.
- Lợi ích thu được bằng việc uỷ thác công việc cho các công ti ngoài qua hợp đồng bảo trì bao gồm:
 - Chi phí có thể được rút bớt.
 - Vấn đề nhân sự có thể được giải quyết, bởi vì việc xem xét nhân sự trở thành không cần thiết. Nó xoá bỏ nhu cầu dịch vụ 24 giờ đối với các nhân viên.
 - Có thể trông đợi vào việc bảo trì chi tiết bằng các chuyên gia.
- Tuy nhiên, nhược điểm lại bao gồm:
 - - Vấn đề có thể phát sinh ra từ khía cạnh an ninh.
 - Nhân viên bảo trì có thể bị thay đổi tuỳ ý công ti đã được kí hợp đồng.
 - Không phải tất cả nhân viên bảo trì được bổ nhiệm đều có đủ tài.
 - Gây ra vấn đề về việc vận hành không được thực hiện.
- Để kết luận về hợp đồng bảo trì, nhiều khía cạnh, kể cả các công ti được uỷ thác làm việc này, và chế độ bảo trì (thường trú hay theo lời gọi) cũng phải được xét tới.
-

2.2.7 Bảo trì phần cứng và bảo trì phần mềm

- Mặc dầu cùng từ, "bảo trì," được dùng, việc bảo trì phần cứng là khác hoàn toàn với việc bảo trì phần mềm trong những vấn đề phải chú ý tới, và phương pháp được dùng.

(1) Bảo trì phần cứng

- Việc bảo trì phần cứng được tiến hành bằng việc đặc biệt chú ý tới những điều sau:
 - - Độ tin cậy
 - Cân bằng hiệu năng
 - Trạng thái kho
- ① Độ tin cậy
 - Các khoản mục bảo trì xét theo độ tin cậy bao gồm những điều sau:
 - **Giám sát các hiện tượng quan trọng**
Các biến cố quan trọng trong hệ thống được giám sát.
 - **Giám sát xu hướng hỏng hóc**
Trong việc giám sát này, dữ liệu được lấy mẫu trong một thời kì dài để phân tích khuynh hướng hỏng hóc.
 - **Giám sát đặc biệt**
Việc giám sát này là dành cho những vấn đề quan trọng trong khi xét tới đặc trưng độ tin cậy

của thiết bị tạo nên hệ thống.

- ② **Cân bằng hiệu năng**

- Nếu một hệ thống đang dùng trong một thời gian dài, thì việc thay thế các thiết bị cũ bằng thiết bị mới có thể gây ra sự mất cân bằng giữa chúng. Nếu sự cân bằng bị làm tồi đi, thì những thiết bị không đủ hiệu năng có thể ảnh hưởng tới hệ thống như một toàn thể. Do đó, sự cân bằng hiệu năng giữa các thiết bị phải được kiểm tra.

- ③ **Tình trạng kho**

- Những chỉ tiêu cần thiết cho hệ thống, và kho các bộ phận cần thay thế hay sửa chữa phải được kiểm tra. Điều quan trọng là những khoản mục này được kiểm tra và giám định đều đặn để ngăn cản việc bị buộc phải dừng vận hành chừng nào các bộ phận còn chưa tới.

(2) Bảo trì phần mềm

- Bảo trì phần mềm được tiến hành bằng việc chú ý đặc biệt tới điều sau:

- - Thêm và cải tiến chức năng
- Khôi phục
- Phòng ngừa

- ① **Thêm và cải tiến chức năng**

- Để đáp ứng những thay đổi theo yêu cầu của hệ thống đang vận hành, việc bảo trì được tiến hành cho các yêu cầu thay đổi về đặc tả phần mềm và/hoặc cải tiến hiệu năng.

- ② **Khôi phục**

- Bảo trì liên quan tới việc khôi phục phần mềm trong vận hành bao gồm những điều sau:

- **Bảo trì sửa chữa**
Bảo trì này được thực hiện nếu các yêu cầu hiệu năng hay đặc tả yêu cầu không được thỏa mãn.
- **Bảo trì thích nghi**
Việc bảo trì này dành cho sự cần thiết thay đổi phần mềm do những thay đổi trong các nhân tố bên ngoài.
- **Bảo trì hoàn thiện**
Bảo trì này được thực hiện để làm tăng mức độ hoàn hảo phần mềm.

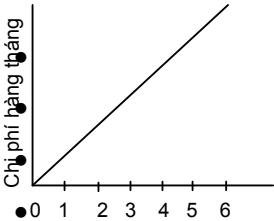
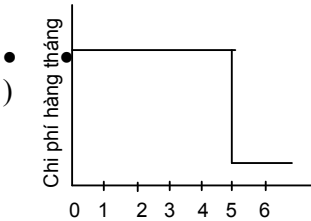
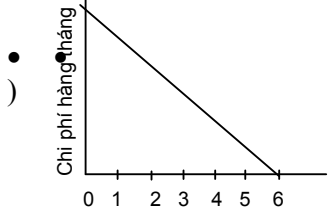
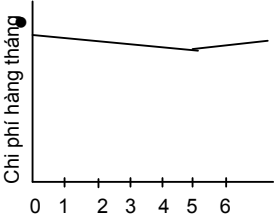
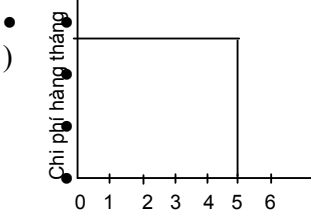
- ③ **Phòng ngừa**

- Việc bảo trì này được thực hiện trên phía phần mềm để phòng ngừa hỏng hóc hệ thống, hay để hỗ trợ cho việc giải quyết vấn đề vận hành, và bao gồm những điều sau:

- **Hỗ trợ kĩ thuật**
Đây là việc bảo trì để giải quyết vấn đề vận hành.
- **Thay đổi phòng ngừa**
Đây là việc bảo trì để phòng ngừa trước về sự xuất hiện hỏng hóc.

Bài tập

- Q1 Thành phần nào trong những thành phần sau không được đưa vào tài nguyên cần quản lý trong quản lý vận hành?
 - Máy tính)
 - Cơ sở dữ liệu)
 - Người lập trình)
 - Chương trình)
- Q2 Thủ tục nào trong những thủ tục sau được xem là thủ tục chuẩn cần thực hiện trong việc hỏng hệ thống?
 - Tiến hành biện pháp tạm thời .
 - Việc dùng biện pháp tạm thời được kết thúc và chúng bị loại bỏ.
 - Phần hỏng được nhận diện và sự hỏng được định vị.
 - Phần hỏng được tách riêng .
- - Xuất hiện hỏng → tìm chỗ hỏng → a → c → d → tiến hành biện pháp thường xuyên) → b
 - Xuất hiện hỏng → tìm chỗ hỏng → b → c → a → tiến hành biện pháp thường xuyên) → d
 - Xuất hiện hỏng → tìm chỗ hỏng → c → a → b → tiến hành biện pháp thường xuyên) → d
 - Xuất hiện hỏng → tìm chỗ hỏng → c → d → a → tiến hành biện pháp thường xuyên) → b
- Q3 Phát biểu nào trong những phát biểu sau là thích hợp nhất cho việc quản lý số hiệu người dùng user ID?
 - Mọi người dùng ID tham gia vào cùng dự án đều dùng cùng ID.)
 - Người dùng có nhiều hơn một ID đặt cùng mật hiệu cho mọi IDs.)
 - Nếu quyền được cấp cho một user ID, nó phải bị giới hạn ở số tối thiểu nhất có thể được.)
 - Việc xoá một user ID phải được tiến hành thời gian dài sau khi việc bỏ ID đó đã được lưu ý.)
- Q4 Phát biểu nào trong những phát biểu sau là không thích hợp nhất cho việc giải quyết mật hiệu và tệp mật hiệu trong tổ chức quản lý hệ thống?
 - Liệu mật hiệu có thể được đoán ra dễ dàng hay không được kiểm tra đều đặn

-) không, và việc dùng các mật hiệu khác nhau có bị thôi thúc cho một vấn đề không.
- • Để làm giảm mức độ các mật hiệu được tham chiếu tới, cần khuyến cáo rằng người dùng ghi lại mật hiệu của mình vào sổ tay hay đâu đó khác.
 - • Nếu thời gian hiệu lực có thể được đặt cho mật hiệu, thì chức năng này phải được dùng.
 - • Việc tham chiếu của người thường tới tệp mật hiệu phải bị cấm, cho dù các mật hiệu, đã được mật mã hoá.
 -
 -
 - Q5 Việc nào trong những việc sau là không liên quan tới chuẩn dịch vụ cho thao tác hệ trực tuyến?
 - • Thời gian đáp ứng
 - • Thời gian bắt đầu vận hành
 - • Thời gian khôi phục hỏng
 - • Thời gian gỡ lỗi
 - Q6 Đồ thị nào trong những đồ thị sau, với số năm trôi qua trên trục hoành và số tiền thanh toán hàng tháng theo trục tung, là thích hợp nhất nếu máy tính được đưa vào trong những hoàn cảnh sau?
 - ① Chi phí máy tính
 - • Trong năm năm đầu được trả, số tiền trả hàng tháng cố định (của tiền thuê) được tính dựa trên giá mua máy tính và tỉ lệ thuê của chúng.
 - • Từ tháng thứ sáu và những năm sau, số tiền trả hàng tháng được tính dựa trên một phần mười giá mua máy tính và tỉ lệ thuê của chúng được trả.
 - ② Chi phí bảo trì
 - • Số tiền hàng tháng cố định được trả như tiền bảo trì cho công ti bảo trì.
 - - -) 
 -) 
 -) 
 - •
 - -) 
 -) 
 - •
 - Q7 Phương pháp nào trong những phương pháp kiểm thử sau được dùng để kiểm tra xem liệu những thay đổi đã được tiến hành cho việc bảo trì phần mềm có ảnh hưởng tới các

phần khác hay không?

- | | | | | | |
|---|---|-------------------|---|---|------------------|
| • | • | Kiểm thử vận hành | • | • | Kiểm thử móc nối |
|) | | |) | | |
| • | • | Kiểm thử hệ thống | • | • | Kiểm thử rà lại |
|) | | |) | | |
| • | | | | | |

- Q8 Mô tả nào trong những mô tả sau là không thích hợp cho mô tả công việc bảo trì cho phần mềm ứng dụng đã được phát triển tại nhà?
 - • Người quản lý vận hành bắt đầu dùng phần mềm mới sau khi những sửa đổi đã được chấp thuận, và loại bỏ phần mềm cũ dựa trên kế hoạch đã định trước.
 - • Người lập trình đã phát triển chương trình gốc thực hiện việc sửa đổi chương trình được liên kết với những đặc tả đã được thực hiện sau khi việc phát triển đã được hoàn tất. Sau đó, phần mềm mới được nhanh chóng đưa vào sử dụng trong môi trường thực.
 - • Người tiến hành kiểm thử phải phân tích những lĩnh vực bị ảnh hưởng bởi sự thay đổi, tiến hành việc kiểm thử cho những bộ phận có liên hệ tới chương trình bị thay đổi, và làm đánh giá.
 - • Trong việc thực hiện bảo trì, các chuẩn, phương pháp luận và thủ tục liên quan tới quản trị tài liệu, phương pháp bảo trì và thủ tục sửa đổi chương trình phải được cung cấp trước.

-
-
-

3 Cập nhật xử lý thông tin và an ninh

Mục đích chương

Hiểu khái niệm về kinh doanh điện tử e-business, học điện tử e-learning và chính phủ điện tử e-government cũng như an ninh cần thiết để bảo vệ hệ thống:

Hiểu kinh doanh điện tử e-business bao gồm những gì, các chuẩn được bao hàm và các kiểu mô hình kinh doanh điện tử

Hiểu kết cấu nền của học điện tử e-learning, các chuẩn được sử dụng và các giao diện được cung cấp

Hiểu chính phủ điện tử e-government bao gồm những gì và các bước cần có để đạt tới nó

Hiểu kiểu đe dọa cho an ninh và các biện pháp đối phó để khắc phục chúng

Giới thiệu

- Năng lực tính toán tăng lên không ngừng và việc giảm giá thành đã dẫn tới việc đưa vào những miền ứng dụng mới. Việc sử dụng Internet ngày càng rộng rãi đã làm cho người ta phải chú ý nghiêm chỉnh tới yếu tố an ninh. Có nhu cầu xem xét lại các kiểu đe dọa và các biện pháp đối phó với những đe dọa đó.
- Trong chương này, chúng ta sẽ bao quát các lĩnh vực về kinh doanh điện tử e-business, học điện tử e-learning và chính phủ điện tử e-government. Các kiểu đe dọa và các biện pháp đối phó cũng sẽ được thảo luận.

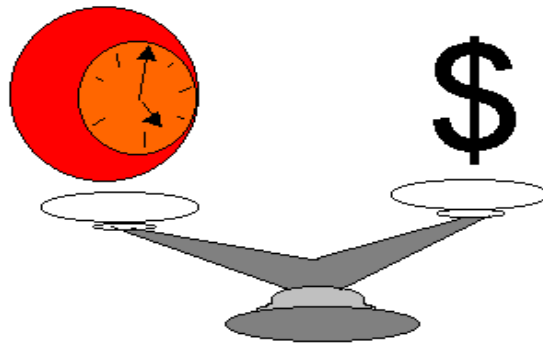
3.1 Lĩnh vực ứng dụng của hệ thống máy tính

3.1.1 Tổng quan về kinh doanh điện tử (e-business)

(1) Các kiểu kinh doanh điện tử (e-business)

- Việc dùng kinh doanh điện tử cho phép tiết kiệm thời gian và tiền bạc.

Các kiểu e-business có thể được phân loại như sau.



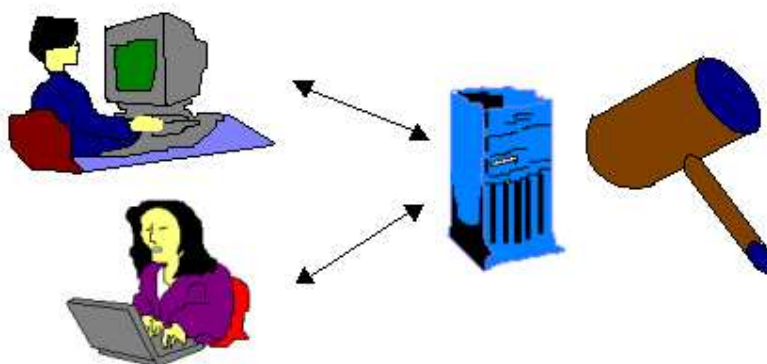
① B2B (B to B - doanh nghiệp với doanh nghiệp)

Kinh doanh này chủ yếu dành cho các công ti. Một số mô hình như các site bán đấu giá cũng có thể được công chúng sử dụng. Các kiểu của mô hình B2B được giải thích sau.



② Bán đấu giá

Các sites này cho phép người dùng đấu thầu cho sản phẩm hay dịch vụ họ cần.



③ B to C (Doanh nghiệp với người tiêu thụ)

Khách hàng trong mô hình B2B là các công ti tạo ra khối lượng mua bán. Trong mô hình B2C, khách hàng được nhắm tới là công chúng. Nó dự định đóng vai trò môi trường bán lẻ từ môi trường "gạch và vữa" và tạo nên sự hiện diện trực tuyến. Việc tổ hợp cả hai môi trường vẫn được dùng còn được biết dưới tên là mô hình "băm và vữa" nơi cả kho vật lí và kho trực tuyến đều tồn tại.

④ Doanh nghiệp với chính phủ

Điều này biểu thị cho các giao tác như mở thầu hay đệ trình các tài liệu có liên quan.



(2) Tiến trình kinh doanh

Các bước sau đây biểu thị cho cách các công ti tiến hành kinh doanh

- ① Phát hiện ra sản phẩm và dịch vụ nào được cung cấp.
- ② Xác định tiến trình chung nào và việc trao đổi thông tin để dùng cho việc thu được các sản phẩm và dịch vụ đó.
- ③ Xác định điểm liên hệ và thiết lập dạng trao đổi các tài liệu kinh doanh.
- ④ Thoả thuận về các khoản mục hợp đồng, điểm liên hệ và dạng trao đổi.

(3) Chuẩn trong kinh doanh điện tử (e-business)

ebXML được xây dựng trên các nguyên tắc sau

- ① Kết cấu nền truyền thông vững chắc, tin cậy và an toàn.
- ② Cơ chế làm mô hình hoá thông tin và tiến trình kinh doanh.
- ③ Thị trường cho phép tất cả các công ti tìm lẫn nhau, thoả thuận trở thành đối tác thương mại và tiến hành kinh doanh với nhau.

ebXML đang trở thành chuẩn được sử dụng trong giao tác kinh doanh điện tử. Chuẩn này do UN/CEFACT (United Nations Center For Trade Facilitation And Electronic Business) và OASIS (Organization for the Advancement of Structural Information Standards) đưa ra. Nó cho phép một khuôn khổ được định nghĩa cho việc kinh doanh điện tử toàn cầu. Các chức năng chính của ebXML là:

- a. Cho phép các doanh nghiệp tìm lẫn nhau
- b. Cho phép tiến hành kinh doanh
- c. Dựa trên XML

ebXML được thiết kế cho tương tác doanh nghiệp với doanh nghiệp. Nó sử dụng ký pháp UML (Unified Modeling Language - Ngôn ngữ mô hình hoá thống nhất).

(4) Cơ sở hạ tầng truyền thông

Cơ sở hạ tầng truyền thông được cung cấp thông qua:

- ① Cơ cấu vận chuyển thông báo theo chuẩn, giao diện được xác định rõ, các nguyên tắc đóng gói chuẩn, và giao hàng dự kiến trước và an ninh.
- ② Một “giao diện dịch vụ kinh doanh”, tức là phần mềm đặt tại mỗi doanh nghiệp để xử lý các thông báo đi và đến.
- ③ Tích hợp các ứng dụng hậu văn phòng hiện có. (Điều này hiện thời nằm ngoài phạm vi của ebXML, nhưng có thể biểu diễn cho cả thách thức lớn nhất và cơ hội lớn nhất cho các nhà cung cấp phần mềm)

(5) Mô hình kinh doanh

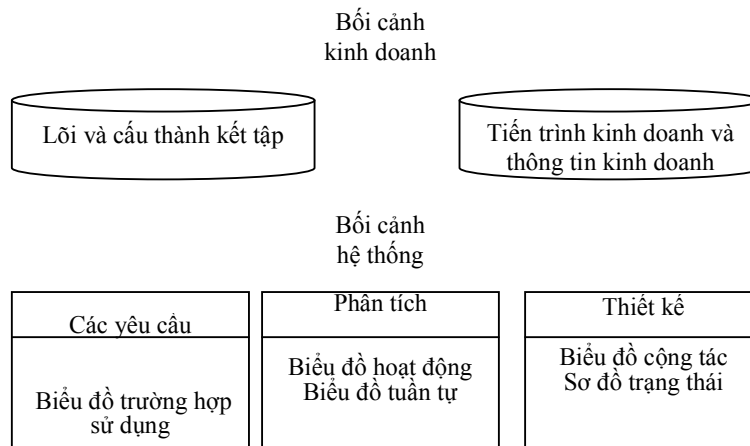
Mô hình kinh doanh được cung cấp qua các yếu tố của chuẩn ebXML sau:

- ① Một đặc tả để xác định các tiến trình kinh doanh và mô hình thông tin.
- ② Một tập các đối tượng kinh doanh tái dụng dựa trên các cấu phần lõi.
- ③ Tiến trình xác định các cấu trúc thông báo.

(6) Thị trường

Thị trường cho phép các doanh nghiệp tìm ra nhau và đồng ý trở thành đối tác thương mại, được cung cấp bởi các yếu tố sau của chuẩn ebXML:

- ① Kho chứa chung (nơi đăng ký) nơi các doanh nghiệp có thể đăng ký và tìm nhau.
- ② Tiến trình để xác định và đồng ý với Thỏa thuận đối tác thương mại hình thức (Trading Partner Agreement: TPA).



3.1.2 Tổng quan về e-learning

(1) Lí do căn bản của e-learning

- ① Cá nhân hóa

Lĩnh vực nghiên cứu có thể được chuyên biệt hoá cho một công ty, một phòng ban hoặc thậm chí một người học cá nhân.

- ② Tương tác

Sự mô phỏng là có thể và có khả năng sử dụng các tính năng hỗ trợ như diễn đàn và các cuộc nói chuyện trực tuyến, với người dạy và bạn học, tạo cho học viên tình huống tương tác rộng hơn.

- ③ Kịp thời

Học các chủ đề liên quan khi các sinh viên thực sự cần nó, so sánh với việc học truyền thống nơi họ mãi về sau mới có thể áp dụng kiến thức.

④ Hiện thời

Để dàng thực hiện cập nhật nội dung và do đó các chương trình và tài liệu mới có thể được cung cấp dễ dàng.

⑤ Đặt trọng tâm vào người dùng

Nó tập trung vào nhu cầu của sinh viên, thay vì khả năng của người dạy.

(2) Các định nghĩa cơ bản về e-learning

① Khóa học

Đây là đơn vị học chính thức. Nó bao gồm nội dung được biểu diễn bởi các đối tượng học, một cơ cấu phân phối và sự đánh giá.

② Đối tượng học

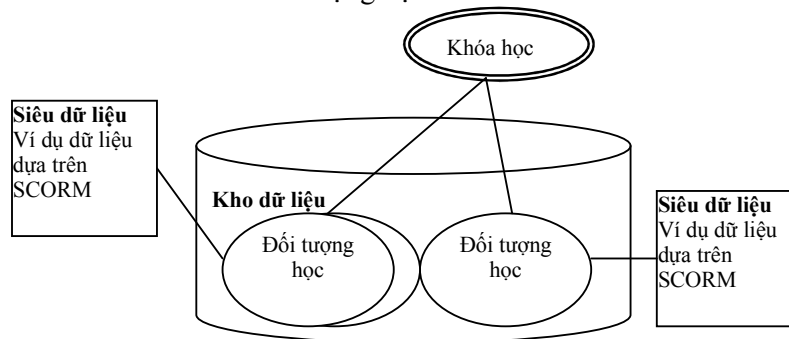
Một mẫu độc lập của việc học. Nó thể được trộn lẫn với các đối tượng học khác để tạo nên khóa học.

③ Kho chứa

Cơ sở dữ liệu các đối tượng học

④ Siêu dữ liệu

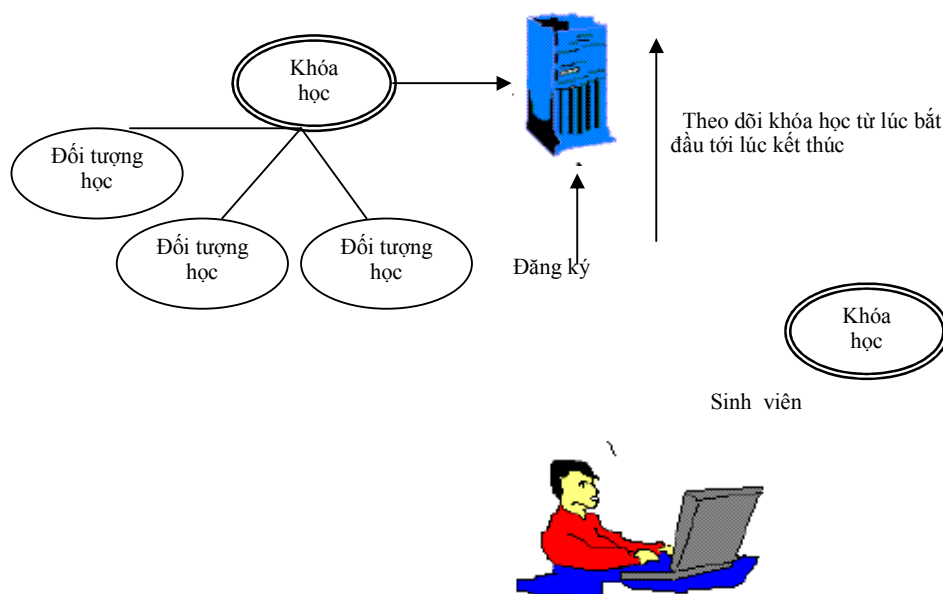
Nó bao gồm dữ liệu mô tả cấu trúc của các đối tượng học.



(3) Các kiểu hệ thống quản lý trong e-learning

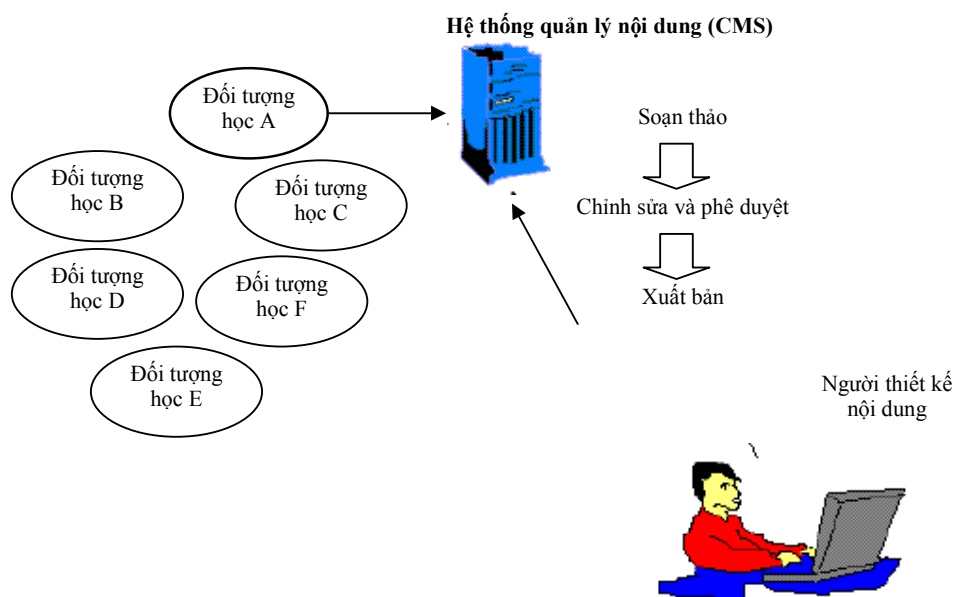
① Hệ thống quản lý việc học (Learning Management System:LMS)

Các khóa học bao gồm các đối tượng học được định nghĩa trước. Sinh viên bị giới hạn vào điều được cung cấp. Không có khả năng chuyên biệt hoá.



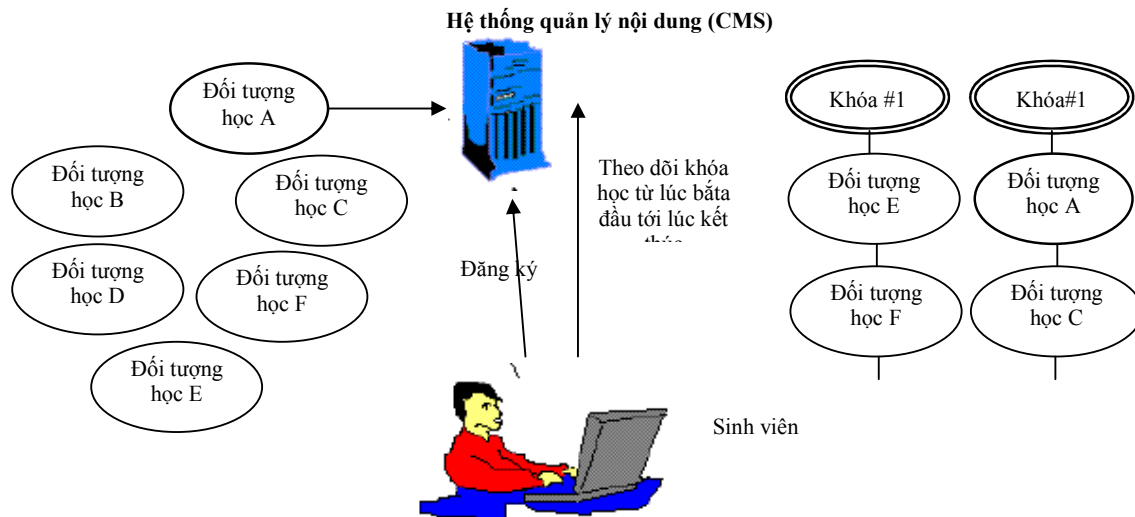
② Hệ thống quản lý nội dung (Content Management System: CMS)

Hệ thống này cho phép tạo ra các đối tượng học. Một đối tượng học có thể đơn giản là đồ họa hoặc video. Người thiết kế đồ họa và những người phát triển nội dung sẽ là tác giả và tạo các đối tượng học và lưu giữ nó trong CMS.



③ Hệ thống quản lý nội dung học (Learning Content Management System: LCMS)

LCMS cho phép lắp ráp các đối tượng học thành các khóa học cá nhân hoá cho sinh viên



(4) Các chuẩn trong e-learning

Sau đây trình bày các tổ chức tham gia vào việc chuẩn hóa định nghĩa các tiêu chuẩn học.

- ① Advanced Distributed Learning (ADL) Initiative - Sáng kiến học phân bố nâng cao
 - a. Mô hình tham chiếu đối tượng phần mềm khoá học dùng chung được - Shareable Courseware Object Reference Model (SCORM):

Đây là một tổ chức chính phủ Mỹ đưa ra SCORM. Đây là tập các chuẩn được chấp nhận gần đây nhất.

- b. Liên đoàn học tập toàn cầu - Global Learning Consortium IMS (Instructional Management System - Hệ quản lí dạy)

Liên đoàn học tập toàn cầu IMS có tổng hành dinh tại Burlington Massachusetts. Nó phát triển và thúc đẩy các đặc tả mở để tạo điều kiện thuận lợi cho các hoạt động học phân bố trực tuyến như định vị và sử dụng nội dung giáo dục, theo dõi tiến bộ của người học, báo cáo học tập của người học và trao đổi các hồ sơ của sinh viên giữa các hệ thống quản trị. Có hai mục tiêu chính:

Định nghĩa các chuẩn kỹ thuật cho tính liên tác của các dịch vụ và ứng dụng trong việc học phân bố.

Hỗ trợ tổ hợp đặc tả IMS vào trong các sản phẩm và dịch vụ toàn cầu.

IMS thúc đẩy việc chấp nhận rộng rãi các đặc tả cho phép môi trường học phân bố và nội dung từ nhiều tác giả làm việc với nhau.

IMS là liên đoàn toàn cầu với các thành viên từ các tổ chức chính phủ, thương mại và giáo dục.

Quỹ của hiệp hội do các thành viên đóng góp.

- ② AICC: Ủy ban đào tạo dựa trên máy tính của công nghiệp hàng không (The Aviation Industry Computer-Based Training Committee)

Ủy ban CBT đào tạo dựa trên máy tính của công nghiệp hàng không (AICC) là hiệp hội quốc tế các chuyên gia đào tạo dựa trên công nghệ. AICC đã phát triển bản hướng dẫn cho ngành công nghiệp hàng không về việc phát triển, chuyển giao và đánh giá CBT và các công nghệ đào tạo liên quan.

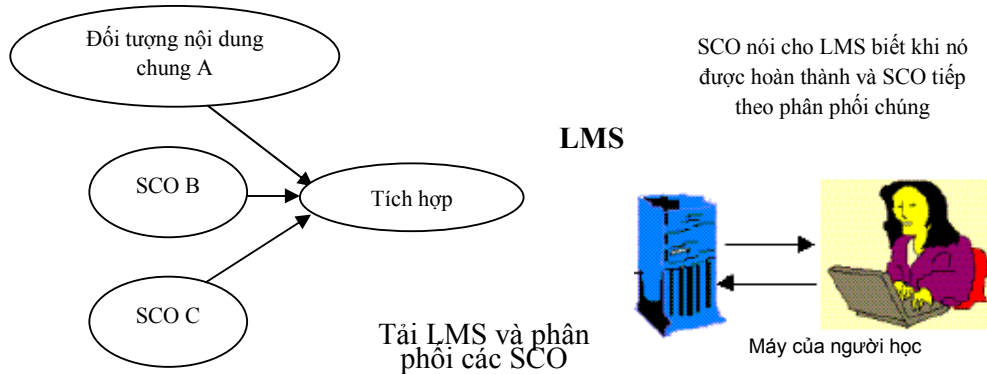
Mục đích chính của AICC là:

- a. Phát triển bản hướng dẫn thúc đẩy việc thực hiện đào tạo dựa trên máy tính cho các thao tác viên máy bay một cách hiệu quả và kinh tế.
- b. Phát triển bản hướng dẫn để có thể hoạt động liên tác.
- c. Cung cấp diễn đàn mở để thảo luận về CBT và các công nghệ đào tạo khác.

Mặc dù AICC chú tâm chính tới công nghiệp hàng không, nhưng 12 năm qua vẫn tập trung vào các đặc tả được yêu cầu để thỏa mãn các nhu cầu của ngành công nghiệp dẫn đầu này. Mặc dù ban đầu AICC nhắm tới ngành công nghiệp hàng không, nhưng các đặc tả đã phát triển cũng được sử dụng cho việc dạy học được quản lý bằng máy tính. Kết quả là rất nhiều liên đoàn học tập và nhóm chuẩn được công nhận chính thức đang trong tiến trình chấp nhận và thích ứng với bản hướng dẫn của AICC với các ngành công nghiệp của riêng họ. Đây là một trong những chuẩn được phát triển sớm nhất.

③ Chuẩn SCORM

- a. Tái dụng các đối tượng học



Nội dung được chia thành các chùm. Yêu cầu với mỗi chùm là

1. Mỗi chùm phải có khả năng trao đổi với hệ thống học sử dụng phương pháp được chuẩn hóa mà không phụ thuộc vào hệ thống.
2. Điều xảy ra trong chùm là công việc của chùm.
3. Cách người học di chuyển *giữa* các chùm được kiểm soát bởi hệ thống học.
4. Mỗi chùm phải có mô tả để làm cho người thiết kế có khả năng tìm kiếm và tìm đúng chùm cho đúng việc. Các chùm như vậy được gọi là các **đối tượng học**. Không có chuẩn về kích cỡ (hoặc *lỗi*) của đối tượng học. Các đối tượng học lớn hơn về cơ bản khó sử dụng lại hơn và các đối tượng của người học nhỏ hơn tiết kiệm ít công việc hơn cho những người sử dụng lại chúng.

b. Kết tập nội dung

SCO là đơn vị tự chứa đựng kiến thức. Chúng có thể được sử dụng như các khối xây dựng để tạo các gói SCO, nhưng chúng không thể được chia thành các đơn vị nhỏ hơn nữa. Ba điều cần được thực hiện để tạo nên đơn vị học lớn hơn từ SCO:

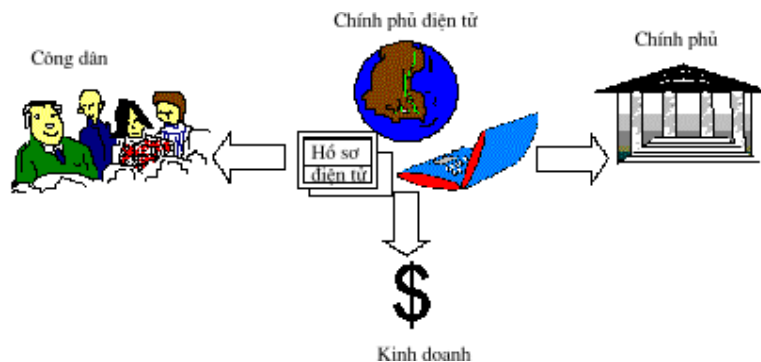
1. SCOs phải được tìm ra và được tổ chức trong một cấu trúc.
2. Các lệnh cần được viết ra để báo cho LMS những SCO nào sẽ đi sau.
3. SCO và các lệnh phải được bó lại thành gói khả chuyển.

Tiến trình này được gọi là kết tập nội dung. Chú ý là kết tập nội dung bao gồm các lệnh để di chuyển giữa các SCO nhưng không dành cho chuyển động bên trong các SCO riêng lẻ. SCORM đã chấp nhận dạng thức đóng gói nội dung từ liên đoàn học toàn cầu IMS. Một gói SCORM bao gồm một tệp kê khai, công bố nội dung của gói và được thiết lập để mô tả trình tự theo đó SCO được chuyển giao. Nó cũng nói cho LMS nơi tìm thấy chính các SCO. Tài nguyên vật lý được diễn tả bởi SCO có thể bao gồm gói vật lý hoặc chúng có thể được tham khảo bên ngoài bởi gói.

3.1.3 Tổng quan về chính phủ điện tử (e-government)

Chính phủ điện tử là ứng dụng công nghệ thông tin vào các quy trình của chính phủ. Điều này tạo ra kết quả làm trong suốt hơn tương tác với các công dân và các doanh nghiệp qua việc cho phép truy nhập vào thông tin.

Phân loại chính phủ điện tử



① G to C (Chính phủ với công dân)

Loại này cho phép các công dân tương tác từ nhà của họ. Bên cạnh việc hoạt động như một nguồn thông tin các giao tác cũng có thể được tiến hành, việc truy vấn và tìm kiếm có thể được thực hiện. Việc huấn luyện có thể được cung cấp trực tuyến, ví dụ huấn luyện để lấy bằng lái xe. Có đa dạng dịch vụ được cung cấp trực tuyến. Một lĩnh vực đang được quan tâm lớn đó là bầu cử trực tuyến.

② G to B (Chính phủ với doanh nghiệp)

Loại này tự động làm tương tác giữa chính phủ và doanh nghiệp. Hai lĩnh vực nơi các dịch vụ được cung cấp là mua sắm điện tử và bán hàng dư thừa của chính phủ qua mạng.

③ Mua sắm điện tử (e-procurement)

Có thể sử dụng hiệu quả ngược với khái niệm bán đấu giá là mua sắm điện tử (e-procurement). Chính phủ có thể nêu ra nhu cầu của mình và việc đấu thầu có thể được thực hiện từ các nhà cung cấp. Điều này cho phép cái nhìn trong suốt hơn vào quy trình đấu thầu.

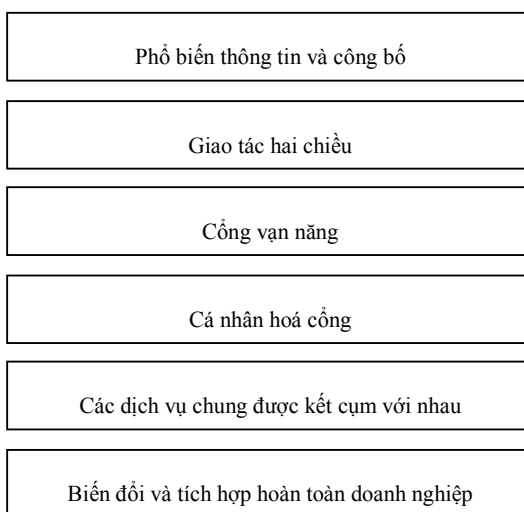
④ Đấu giá điện tử (e-auction)

Loại này cho phép các chính phủ có thể tận dụng việc bán đấu giá bên thứ ba hoặc thiết lập các site đấu giá của riêng họ để loại bỏ hàng hóa hoặc vật liệu dư thừa.

⑤ G to G (Chính phủ với chính phủ)

Các hoạt động liên chính phủ giữa các cơ quan đưa ra các tương tác giữa các cơ quan.

(2) Các giai đoạn thực hiện chính phủ điện tử



① Phổ biến thông tin và công bố

Các cơ quan tư nhân thiết lập các trang web của họ cung cấp thông tin và phạm vi dịch vụ và liên lạc. Bảng từ điện tử có thể được xây dựng để giảm số cuộc gọi khi công chúng tương tác với người có quyền liên quan.

② Giao tác hai chiều

Công chúng có thể tiến hành các giao tác an toàn và tiền tệ bằng việc sử dụng tổ hợp chữ kí số thức và site an toàn. Công chúng phải tin tưởng vào quyền riêng tư trong trao đổi.

③ Các cổng vạn năng

Một cổng trở thành một điểm truy nhập có móc nối tới các cơ quan và bộ phận có liên quan.

④ Cá nhân hoá cổng

Điều này cho phép người sử dụng cá nhân hóa cách nhìn của họ về site này. Họ có thể chọn kiểu dịch vụ và cơ quan sẽ xuất hiện khi họ kết nối với cổng. Điều này trở thành tương đương với CRM (Customer Relationship Management - hệ quản lý quan hệ khách hàng) từ chính phủ.

⑤ Các dịch vụ chung được kết cụm

Công chúng sẽ được trình bày qua một cách nhìn về các dịch vụ mà không cần đưa ra sự khác biệt giữa các bộ phận. Các quy trình của cơ quan được sắp xếp hợp lý lẫn nhau cho phép các luồng tiến trình thực hiện trôi chảy.

⑥ Biến đổi và tích hợp hoàn toàn doanh nghiệp

Điều này trở thành một trung tâm dịch vụ đầy đủ với việc loại bỏ đi sự dư thừa và gộp các phòng ban. Các phòng ban mới có thể phát sinh.

(3) Lợi ích của chính phủ điện tử

Chính phủ điện tử giảm nhu cầu về dạng giấy tờ và nhập dữ liệu bằng tay. Nó làm giảm số cuộc gọi điện thoại và các yêu cầu gặp mặt trực tiếp. Nó khử bỏ các tiến trình thủ công tốn thời gian. Công nghệ mới nhất được sử dụng để sắp xếp hợp lý hoá các hoạt động. Nó cho phép các cơ quan đưa ra mức dịch vụ cao hơn trong những thời gian ngân sách bị giảm.

(4) Các vấn đề riêng tư

Cuộc cách mạng truyền thông cũng đem tới vô số cơ hội chia sẻ thông tin. Điều này cũng làm tăng thêm mối băn khoăn về tính riêng tư. Vấn đề riêng tư là nhạy cảm khi có liên quan tới các bản ghi được chính phủ soạn thảo và duy trì. Khi cố gắng phục vụ các công dân tốt hơn, các cơ quan phải có ý thức về trách nhiệm của họ là những người giữ thông tin nhạy cảm.

(5) Chính phủ giữ vai trò gì?

Chính phủ cung cấp cơ sở hạ tầng. Số lượng cực lớn thông tin được chính phủ lưu giữ. Tính hiệu quả phụ thuộc vào việc thu thập và sử dụng. Người sử dụng sẽ tự nguyện tương tác với chính phủ thông qua trang web chỉ khi họ tin tưởng vào tính toàn vẹn an ninh. Điều sống còn là chính phủ thực hiện các thủ tục và giao thức an ninh đúng đắn. Việc lập kế hoạch cẩn thận và thận trọng là mấu chốt.

(6) Các vấn đề an ninh

Thực hành an ninh được chấp nhận chung là tổ chức thông tin ở bên thứ ba đáng tin cậy. Thông tin và website được bảo vệ an toàn bởi công nghệ tường lửa và site này được cập nhật bởi các cơ quan từ xa.

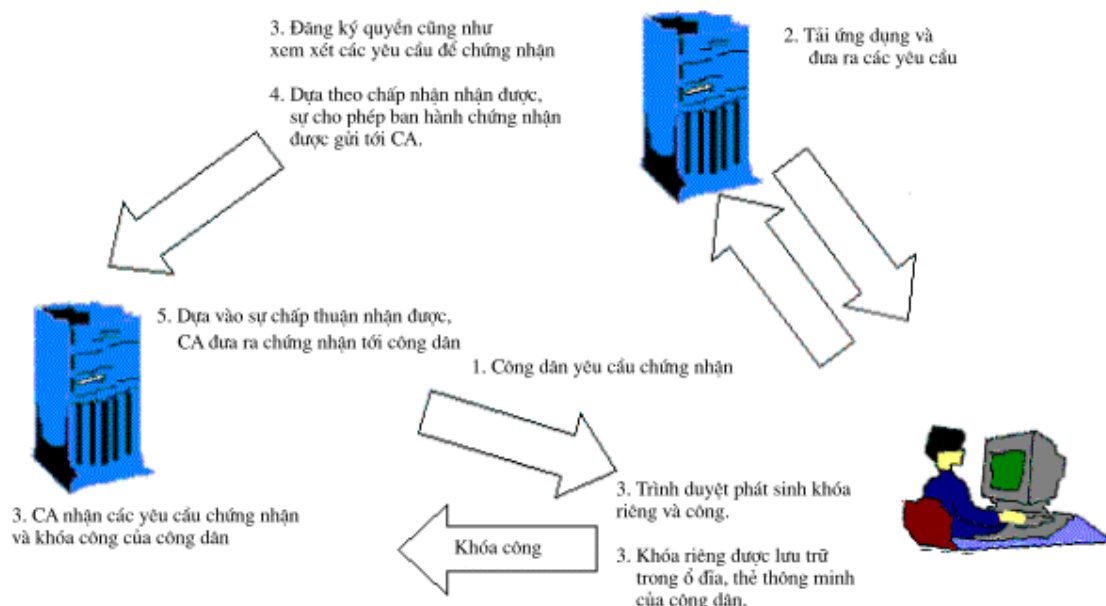
Các ứng dụng tiên tiến hơn cho phép truy lục thông tin từ cơ sở dữ liệu của cơ quan. Đôi khi điều mong muốn là đặt thông tin nhạy cảm trực tuyến nhưng hạn chế truy nhập.

① Chứng chỉ số thức

Thủ tục an ninh nghiêm ngặt hơn và cao cấp hơn cần tới chứng chỉ số thức và chữ kí số thức. Chứng chỉ số thức là dạng tương đương điện tử với bằng lái xe hay hộ chiếu. Chứng chỉ số thức ràng buộc mỗi danh xưng với cặp khóa điện tử. Các khóa có thể được sử dụng để mật mã hóa và ký vào thông tin số thức.

Khi được cài đặt, các cá nhân được nhắc về chứng chỉ số thức của họ khi truy nhập vào các vùng bị hạn chế.

Khi được sử dụng thích hợp, các chứng chỉ số thức có thể cung cấp sự tin cậy cơ bản trong việc cho phép truy nhập thông tin nhạy cảm bị hạn chế của cơ quan.



(7) Truy nhập thông tin của hã

Công chúng muốn chức năng được tăng lên. Bằng cách cung cấp truy nhập qua Internet có thể làm giảm cuộc gọi điện thoại và lưu thông quầy thanh toán. Ứng dụng thông thường sẽ cung cấp truy nhập tới cơ sở dữ liệu của cơ quan. Các giao diện Web được xây dựng để cho phép mọi người tiến hành việc truy vấn của họ.

Rồi thông tin được yêu cầu được hiển thị trong trình duyệt của người yêu cầu. Điều này đem lại kết quả là tiết kiệm lớn. Việc cập nhật thông tin của cơ quan yêu cầu một tổng phí cao vì nó đòi hỏi nhiều công việc giấy tờ. Việc giữ thông tin được hiện thời và chính xác là vấn đề mà mọi cơ quan đều phải đối diện.

Ngày nay, các cá nhân có thể trình thông tin và thực hiện thanh toán điện tử thông qua web. Việc gõ và nhập dữ liệu dư thừa của cơ quan được giảm bớt hay khử bỏ.

(8) Sự phát triển của các chuẩn

Mặc dù EDI đã tồn tại một thời gian dài, xu hướng gần đây hướng theo các chuẩn cho việc phục vụ chức năng bên ngoài việc trao đổi dữ liệu. Đây là các chuẩn dựa trên XML như ebXML. Khi các giao tác được thực hiện, một chuẩn như vậy bắt nguồn từ kinh doanh điện tử là ebXML. Ưu điểm của chuẩn này là bên cạnh việc hỗ trợ các giao tác nó cũng cho phép mô tả các tiến trình và các kiểu cộng tác giữa chúng. Nó có thể được sử dụng như một chuẩn để mô tả cơ sở hạ tầng của các quy trình của chính phủ.

Chuẩn ebXML bao gồm:

Các mức	Mô tả
Mô hình tham chiếu kinh doanh	xác định khung tham chiếu cho toàn bộ môi trường kinh doanh trong đó các quy trình kinh doanh xuất hiện
Phát hiện và nhận	cung cấp kho các quy trình kinh doanh, giống như các trường hợp sử dụng

diện tiến trình kinh doanh	mức cao để ghi lại các quy trình và các bên
Phác thảo tiến trình kinh doanh	nhận diện các tác nhân riêng lẻ cũng như các điều kiện trước và sau quy trình.
Xác định cộng tác kinh doanh	xác định các biến cố kinh tế, nêu ra đại cương về các giới hạn của hệ thống và các giao thức chi phối luồng thông tin trong quy trình
Xác định giao tác kinh doanh	bao quát các hoạt động riêng lẻ xảy ra trong quy trình được gây ra bởi các bên có thẩm quyền trong các tổ chức; trong khi việc biết ai thực hiện các hành động và vị trí của họ trong tổ chức là rất quan trọng, thì hầu hết các định nghĩa quy trình bao quát các tương tác giữa các tổ chức (chứ không ở bên trong các tổ chức).
Xác định thông tin kinh doanh	nhận diện nội dung của các thông báo, bao gồm các chi tiết như kích cỡ các trường được yêu cầu, các kiểu dữ liệu, các mô tả, các yêu cầu kiểm định và theo dõi dấu vết, và bất kỳ hoàn cảnh kinh doanh bổ sung nào cho các câu phần về ngữ nghĩa tái sử dụng trong ebXML

(9) Tính sẵn sàng cho chính phủ điện tử

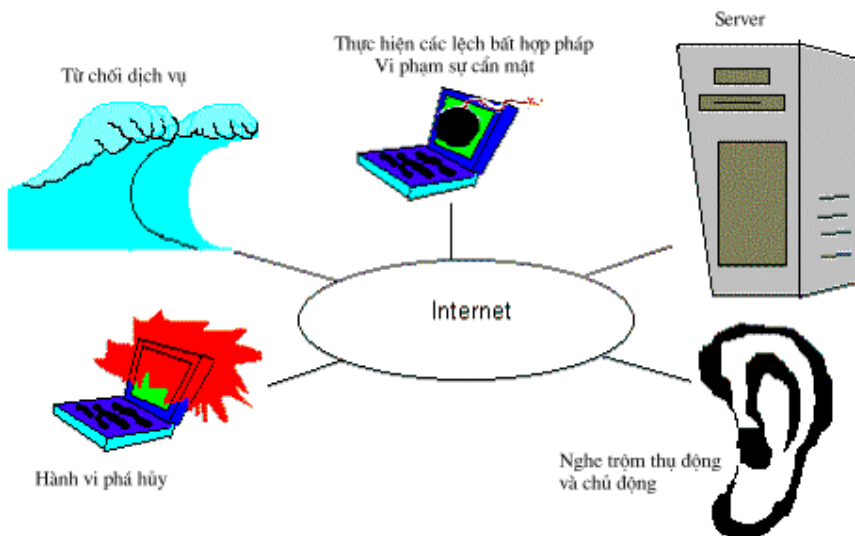
Tiêu chí	Mô tả
Tính nổi	Các mạng có đủ khả năng và dễ dàng truy nhập và sử dụng? Sự sẵn sàng của các dịch vụ truyền thông mạng vô tuyến và hữu tuyến, trung tâm truy nhập cộng đồng (miễn phí và có phí) và các máy tính nối mạng trong doanh nghiệp, trường học, và hộ gia đình. Tính đảm đương được và độ tin cậy của việc truy nhập mạng, kể cả chi phí dịch vụ, thời gian chết của máy móc, và sự phổ biến của việc truy nhập chung giữa các cá nhân.
Quyền lãnh đạo điện tử (E-Leadership)	Tính sẵn sàng điện tử có là ưu tiên quốc gia không? Mức ưu tiên do chính phủ mang lại để thúc đẩy sự phát triển của xã hội điện tử ở mức quốc gia. Mức tiến bộ được biểu lộ về chính phủ điện tử, bao gồm các nỗ lực tự động hóa các quy trình chính phủ. Chất lượng của các quan hệ đối tác giữa các nhà lãnh đạo công nghiệp và chính phủ để cải thiện tính sẵn sàng điện tử. Mức nỗ lực để thúc đẩy truy nhập cho tất cả các công dân. An ninh thông tin - Việc lưu giữ và xử lý thông tin kết mạng có thể tin cậy được không? Sức mạnh của việc bảo vệ theo luật pháp và tiến bộ trong bảo hộ quyền sở hữu trí tuệ, đặc biệt là phần mềm. Qui mô nỗ lực bảo vệ quyền riêng tư Hiệu quả và sức mạnh của khuôn khổ pháp lý để đề cập và truy tố tội phạm máy tính, cấp quyền chữ ký số thức, và tạo khả năng cơ sở hạ tầng khóa công
Công nhân tri thức	Mọi người có quyền hỗ trợ kinh doanh điện tử và xây dựng xã hội dựa trên tri thức không? Chất lượng và mức độ tham gia trong hệ thống đào tạo, với việc nhấn mạnh vào các nỗ lực để tạo ra và hỗ trợ cho xã hội dựa trên tri thức.

	Văn hóa của sự sáng tạo địa phương và chia sẻ thông tin bên trong xã hội. Các kỹ năng và hiệu quả của lực lượng lao động.
Môi trường kinh doanh điện tử	Ngày nay, thực hiện kinh doanh điện tử dễ dàng như thế nào? Tồn tại sự cạnh tranh có hiệu quả giữa các nhà cung cấp dịch vụ thông tin và truyền thông. Tính trong suốt và có thể dự báo trước của việc thực hiện các qui chế, tính công khai của chính phủ, nguyên tắc luật pháp, và rủi ro kinh doanh nói chung (sự ổn định chính trị, sự lành mạnh về mặt tài chính). Tính công khai đối với sự tham gia về tài chính và cá nhân bởi các nhà đầu tư nước ngoài trong các doanh nghiệp ICT. Khả năng của hệ thống tài chính hỗ trợ các giao tác điện tử.

3.2 An ninh

3.2.1 An ninh Internet

Các kiểu đe dọa



① Nghe trộm thụ động

Đây là việc nghe lưu thông trên mạng. Tuy nhiên không có lưu thông mới được đưa vào.

② Nghe trộm chủ động

Lưu thông bị đọc và lưu thông mới được đưa vào trong mạng. Sự lừa bịp nghĩa là giả vờ là nhà cung cấp dịch vụ hợp pháp để đánh lừa khách hàng trong việc để lộ thông tin truy nhập. Mật mã hóa được sử dụng để bảo vệ chống lại các loại nghe trộm thụ động và chủ động.

③ Các cuộc tấn công vào địa chỉ IP

a. Giả mạo địa chỉ IP

Đây là nơi một máy chủ công bố có địa chỉ IP của máy khác. Vì nhiều hệ thống (như danh sách kiểm soát truy nhập của bộ định đường) định nghĩa các gói có thể hoặc không thể đi qua dựa trên địa chỉ IP phía gửi, đây là kỹ thuật hữu ích đối với kẻ tấn công: hắn có thể gửi các gói tin tới một máy chủ, có thể gây cho nó chiếm được một loại hành động nào đó.

b. Cướp phiên địa chỉ IP

Cướp phiên địa chỉ IP là kiểu tấn công mà qua đó phiên làm việc của người sử dụng bị xâm chiếm, bị kiểm soát bởi kẻ tấn công. Nếu người sử dụng đang ở giữa chừng một thư điện tử, kẻ tấn công sẽ xem thư và sau đó có thể thực hiện bất kỳ lệnh mà anh ta muốn giống như người sử dụng bị tấn công có thể làm. Người sử dụng bị tấn công chỉ thấy phiên làm việc của mình bị dẹp bỏ, và có thể đơn giản đăng nhập lại, có thể thậm chí không để ý rằng kẻ tấn công vẫn đang đăng nhập và làm mọi thứ.

④ Tấn công vào việc Từ chối dịch vụ (Denial Of Service (DOS) attacks)

Điều này có nghĩa là máy chủ bị tràn ngập quá nhiều lưu thông đến mức bị không làm việc được. Các hành động có thể được thực hiện để ngăn chặn các cuộc tấn công từ chối dịch vụ là

- a. Không chạy các máy phục vụ mà thế giới thấy được ở mức quá gần với khả năng của nó.
- b. Sử dụng bộ lọc gói để ngăn chặn các gói giả mạo rõ ràng không cho nhập vào trong không gian địa chỉ mạng của bạn.
- c. Luôn cập nhật các miếng vá liên quan tới an ninh cho hệ điều hành của máy chủ của bạn.

• (2) Truy nhập trái phép

Việc truy nhập này do những người không có quyền gây ra.

① Thực hiện các chỉ lệnh bất hợp pháp

Chúng ta phải ngăn cản các chỉ lệnh không cho thực hiện bất hợp pháp trên hệ thống. Có hai kiểu truy nhập có thể:

- a. Truy nhập của người sử dụng bình thường
- b. Truy nhập hành chính

Những người sử dụng thông thường không nên được cho quyền truy nhập ngoài điều được cần tới. Những người sử dụng thông thường về cơ bản có yêu cầu đọc thông tin. Quyền người quản trị cho phép kẻ tấn công thay đổi cấu hình của hệ thống.

② Vi phạm tính mật

Điều này nói tới việc ăn cắp thông tin từ máy phục vụ. Trong trường hợp này, truy nhập đủ để đọc thông tin có thể đủ cho kẻ tấn công thu được việc truy nhập vào dữ liệu

③ Hành vi phá hoại

Có hai kiểu hành vi phá hoại chính

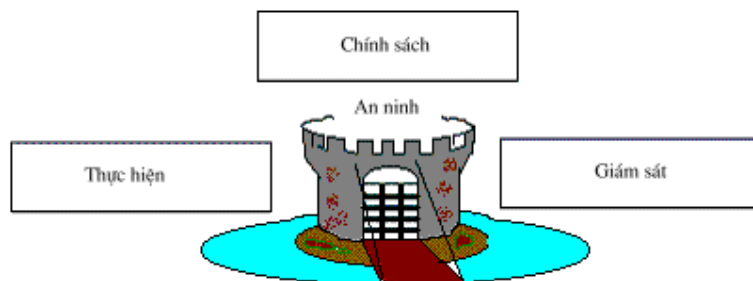
a. Thay đổi dữ liệu

Việc này có thể làm thay đổi trong dữ liệu. Những thay đổi này có thể mãi về sau mới nổi ra bề ngoài.

b. Phá hủy dữ liệu

Việc này nhằm phá hỏng và xóa dữ liệu.

(3) Chính sách an ninh



- ① Phát triển chính sách an ninh mạnh
- ② Giữ an toàn mạng lưới.
- ③ Giám sát mạng và đáp ứng với các cuộc tấn công.
- ④ Kiểm tra các bộ phận an toàn an ninh hiện có.
- ⑤ Quản lý và nâng cao an ninh công ti.

Các câu hỏi sau đây được đưa ra khi thiết lập chính sách an ninh

- a. Những tài sản nào cần bảo vệ?
- b. Có rủi ro nào cho những tài sản đó?
- c. Tác động của cuộc tấn công thành công là gì? (dưới dạng danh tiếng, doanh thu, lợi nhuận, nghiên cứu)?
- d. Có bao nhiêu thông tin nhạy cảm đang trực tuyến? Tác động như thế nào khi thông tin này bị phá hủy hoặc bị đánh cắp?
- e. Người sử dụng nào có truy nhập vào những tài sản đó?
- f. Người sử dụng (và điều này bao gồm cả các đối tác kinh doanh/ hoặc các khách hàng) mong đợi gì về các cơ cấu và thủ tục kiểm soát an ninh?
- g. Người sử dụng của bạn có thường truy nhập tới các tài sản từ xa hoặc nội bộ hoặc kết hợp cả hai phương thức này không?
- h. Bạn có cần các mức an ninh khác nhau cho các bộ phận khác nhau trong tổ chức không?
- i. Kiểu lưu thông nào hiện có trên mạng của bạn?
- j. Nhu cầu về an ninh có nhất quán với nhu cầu hoạt động/kinh doanh của tổ chức không?
- k. Có cam kết mạnh mẽ từ cấp quản lý để cung cấp đủ tài nguyên cho việc thực hiện

các chính sách và công nghệ an ninh không?

I. Có cam kết mạnh mẽ cho việc huấn luyện nhận biết về an ninh?

⑥ Mặc định nên là không truy nhập

Trong nhiều hệ thống, có một thực hành là cho phép bất kỳ ai truy nhập tới bất cứ cái gì theo mặc định, nhưng cho phép người quản trị hệ thống tắt hoặc giới hạn truy nhập tới một số điều họ tin là cần thiết. Đây là cách tiếp cận sai, trong hệ thống như vậy rất dễ quên mất giới hạn việc truy nhập và thay vì vậy mặc định nên là không người nào có bất cứ quyền truy nhập nào chừng nào họ chưa được trao tường minh quyền đó.

⑦ Trao quyền tối thiểu có thể

Khi ai đó được trao quyền truy nhập vào tài nguyên nào đó, họ nên được trao cho quyền thấp nhất mà họ thực sự cần để thực hiện công việc họ làm. Chú ý rằng quyền nên được ấn định dựa trên nhu cầu theo nhiệm vụ chứ không dựa trên điều gì đó như thâm niên công tác.

Nguyên tắc cơ bản là ở chỗ mọi người sẽ phàn nàn nếu họ có quá ít quyền truy nhập tới một cái gì đó nhưng họ không phàn nàn nếu họ quá nhiều quyền truy nhập.

⑧ Kiểm tra quyền hiện tại

Đối với mọi truy nhập tới tài nguyên được bảo vệ, bạn nên kiểm tra xem liệu một cá nhân có những quyền thích hợp không. Chỉ dựa vào sự kiện họ đôi khi có truy nhập trong quá khứ là không đủ, vì quyền của họ có thể đã bị hủy bỏ.

(4) Các cơ chế bảo vệ

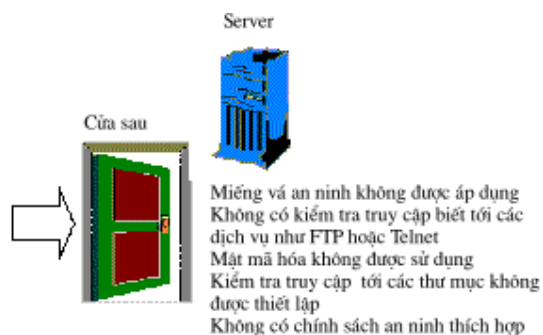
Các điều sau có thể được đưa vào để nâng cao an ninh

- ① Sử dụng tường lửa
- ② Mật mã hóa
- ③ Chữ ký số thức
- ④ An ninh ứng dụng

Kiểm định sổ kí sự

Lỗi hồng pháp lí

Điều này liên quan tới các đặc trưng của việc thiết lập. Một số dịch vụ nên bị giới hạn trong máy phục vụ được đề lộ ra

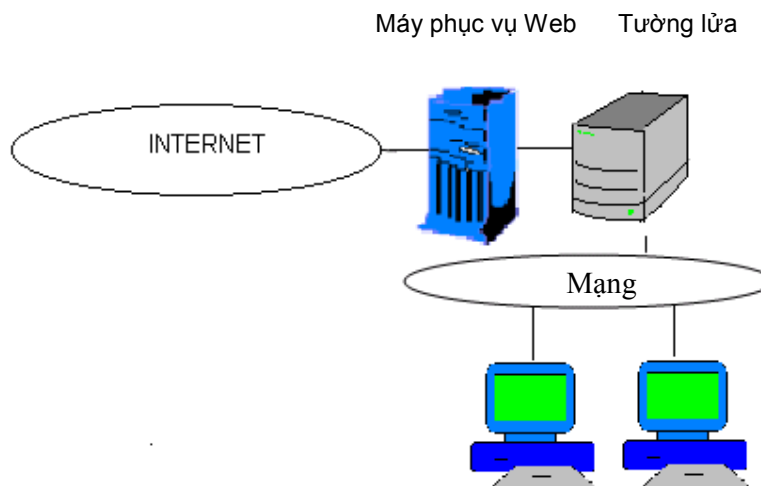


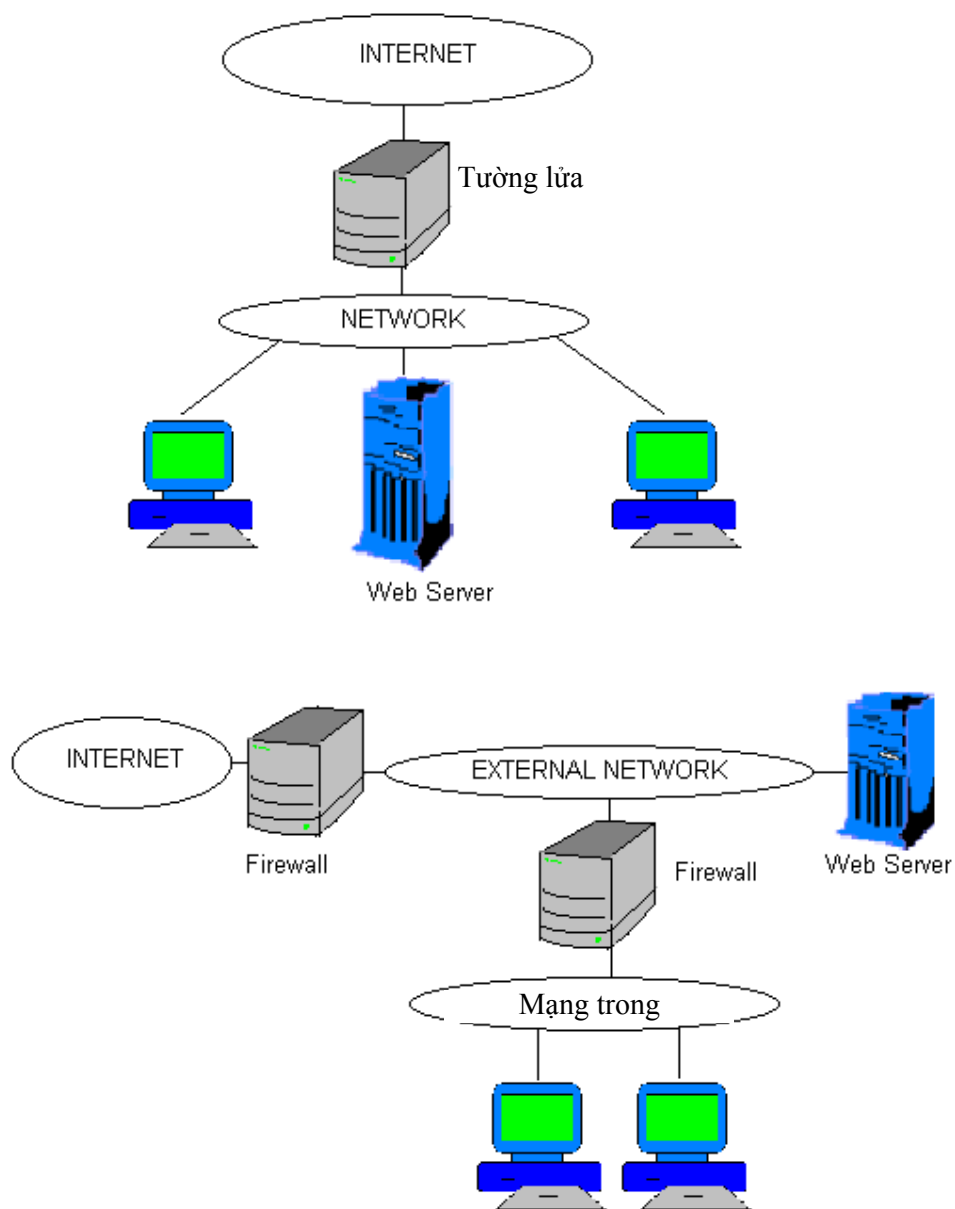
Các dịch vụ	Mô tả
Netstat	Dịch vụ này có thể để lộ mẫu sử dụng và cấu hình hệ thống
FTP	FTP nặc danh nên được thực hiện trên một máy khác
Telnet	Telnet nên bị giới hạn chỉ cho những người quản trị
Thiết lập ODBC hoặc vị trí cơ sở dữ liệu	Vị trí máy phục vụ cơ sở dữ liệu không nên được cài đặt trong máy phục vụ hoạt động.

Các miếng vá dịch vụ nên được áp dụng và giữ luôn hiện hành.

b. Tường lửa

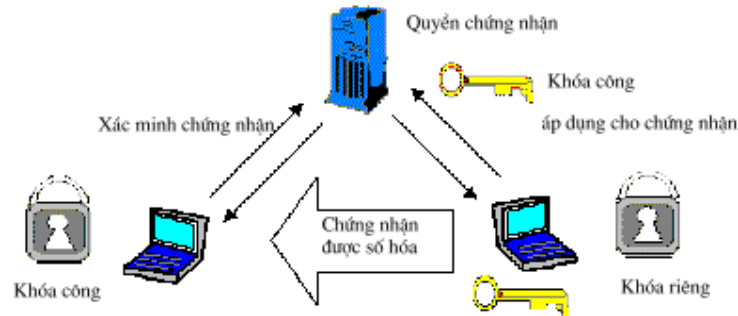
Máy phục vụ web có thể được đảm bảo bằng cách sử dụng tường lửa. Tường lửa hình thành nên rào chắn giữa internet và mạng.





c. Mật mã hóa

Các thuật toán khóa công



Một tập tách biệt các khóa được sử dụng để mật mã hóa và giải mật mã. Khóa mật mã được biết tới là **khóa công**. Khóa giải mật mã được biết tới là **khóa tư** hoặc khóa bí mật. Điều này nghĩa là khóa công có thể được công bố tự do. Sử dụng khóa công này, một thông báo có thể được gửi một cách an toàn tới bên khác. Chỉ bên giữ khóa bí mật mới có thể giải mật mã thông báo này.

Các thuật toán khóa công cũng được sử dụng để tạo chữ ký số thức trên dữ liệu. Khóa bí mật được sử dụng để tạo chữ ký số thức và khóa công được sử dụng để kiểm chứng nó.

Giám sát

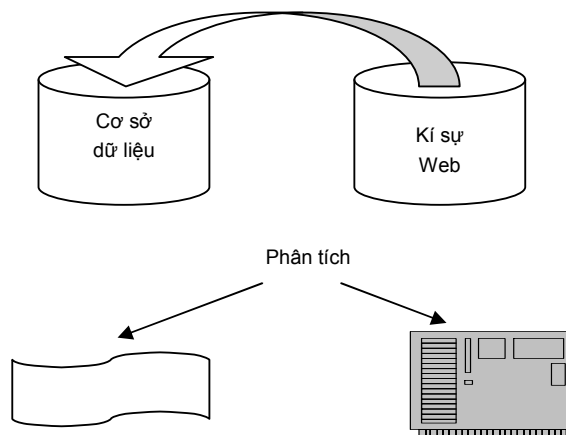
Kiểu giám sát bắt nguồn từ việc phân tích về thông tin kí sự được máy phục vụ thu thập. Bên cạnh đó, phần mềm cũng có thể được đưa vào để thực hiện một số kiểu giám sát.

Sau đây là một số hoạt động giám sát được thực hiện

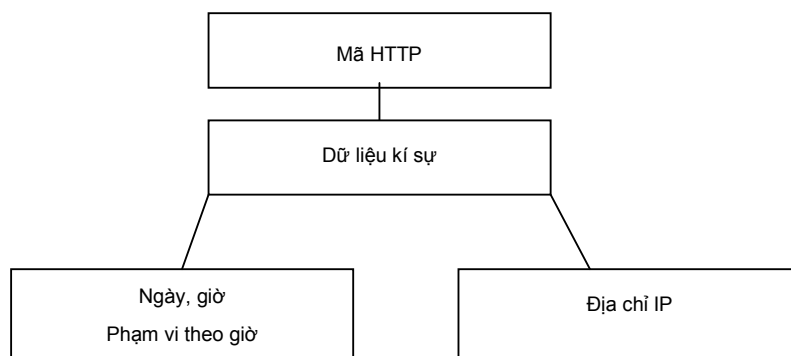
- Phân tích kí sự trên Web (Web log analysis)
- Hệ thống phát hiện xâm nhập

d. Phân tích kí sự trên Web (Web log analysis)

Đa dạng thông tin kí sự được thu thập cho máy phục vụ. Các báo cáo sẽ được sinh ra và việc phân tích được thực hiện để làm lộ ra những âm mưu xâm nhập. Các bản kí sự có thể được lưu giữ trong cơ sở dữ liệu và việc phân tích có thể được thực hiện.



Ví dụ về tiêu chí được sử dụng để phân tích bản ghi web



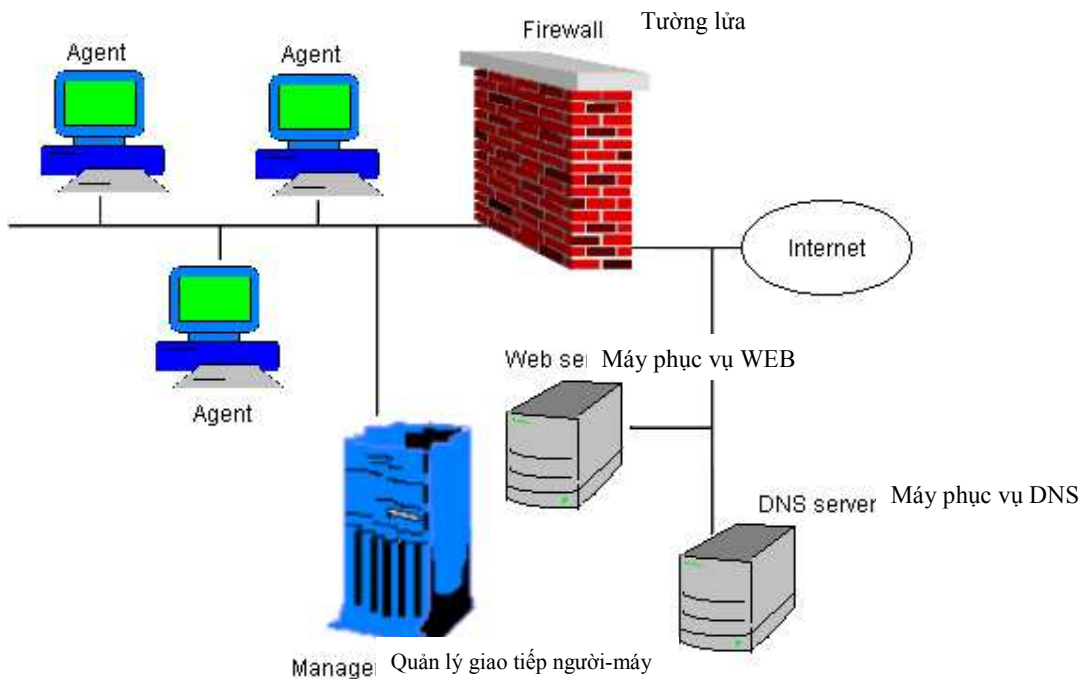
Việc kiểm soát có thể được sử dụng theo các mã HTTP

Mã HTTP	Mô tả
Unauthorized 401	Tham số của thông báo này cho đặc tả về các lược đồ có thẩm quyền chấp nhận được. Máy khách nên thử yêu cầu lại với tiêu đề có thẩm quyền phù hợp.
Forbidden 403	Yêu cầu về cái gì đó bị cấm. Việc cấp quyền sẽ không có tác dụng.
Proxy Authentication Required 407	Mã này tương tự 401 (không có thẩm quyền), nhưng chỉ ra rằng máy khách trước hết phải xác thực với máy uỷ quyền proxy.
Method Not Allowed 405	Phương pháp được xác định trong dòng yêu cầu không được cho phép với tài nguyên được nhận diện bởi yêu cầu của URI.
Service temporarily overloaded 502	Máy phục vụ không thể xử lý yêu cầu do tải cao (dù là đang phục vụ HTTP hoặc các yêu cầu khác). Ngụ ý đây là điều kiện tạm thời mà có thể được giảm bớt tại các thời điểm khác.

e. Hệ thống phát hiện xâm nhập (IDS)

Hệ thống dựa trên máy chủ được sử dụng để bảo vệ các hệ thống chứa đựng thông tin quan trọng. Các tác tử được tải tại mỗi máy chủ. Các tác tử này sẽ phân tích các vật kiểm định. Chúng có thể được xác định để dò ra theo mức độ chi tiết mong muốn, ví dụ chính xác người dùng nào truy nhập vào tệp nào. Vì các tác tử phải được cài đặt tại mỗi máy chủ có yêu cầu theo dõi, nên không thể có qui mô lớn. Các tác tử này sẽ so sánh hoạt động theo tập các nguyên tắc và lấy cò báo động nếu bất kì quy tắc nào bị vi phạm.

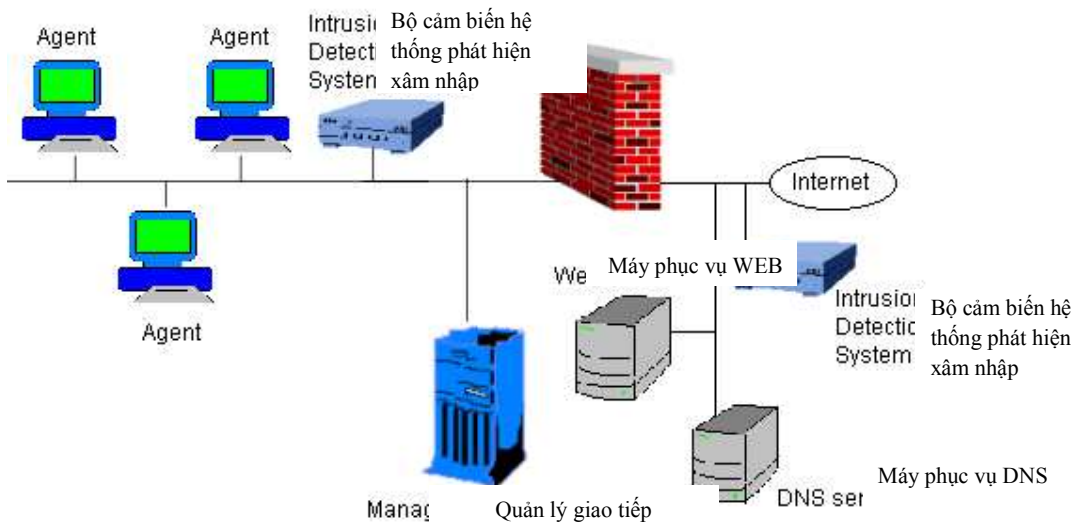
Máy chủ dựa trên IDS



Mạng dựa trên IDS

Mạng dựa trên IDS giám sát một bộ phận mạng xác định. Thông thường có các cấu phần chuyên dụng bao gồm bộ cảm biến và hệ thống quản lý.

Hệ thống quản lý được sử dụng để lập cấu hình việc phát hiện và cũng để hiển thị thông tin báo động. Các bộ cảm biến này thực hiện việc phân tích dựa trên nguyên tắc hoặc phân tích theo hệ chuyên gia về lưu thông bằng việc dùng các tham biến đã được lập cấu hình. Các bộ cảm biến này phân tích các tiêu đề gói để xác định các địa chỉ nguồn và đích, kiểu dữ liệu được truyền, và phân tích tải gói để phát hiện thông tin trong dữ liệu đã được truyền. Một khi bộ cảm biến phát hiện việc sử dụng sai, nó có thể thực hiện các hoạt động khác liên quan tới an ninh: ghi lại biến cố, gửi báo động tới bàn điều khiển quản lý, đặt lại kết nối dữ liệu, v.v...



Các qui tắc

Bản lược sử hoạt động của người sử dụng được sinh ra và việc phát hiện xuất hiện khi bất cứ hoạt động nào bắt đầu ra ngoài phạm vi đó. Bản lược sử hoạt động của người sử dụng có thể chứa các thuộc tính như các tệp các máy phục vụ thường được truy nhập, thời gian dành cho việc vào mạng, vị trí truy nhập mạng, v.v...

Tuy nhiên, việc này có thể khó thực hiện khi người sử dụng có thể thay đổi lược sử của họ như việc bổ nhiệm của họ đòi hỏi họ truy nhập vào các tài nguyên khác. Điều này được biết tới như sự phát hiện dựa trên lược sử.

Phát hiện dựa trên chữ ký có thể được so sánh với các chương trình kiểm tra virus. Các chữ ký này được cung cấp bởi các nhà cung cấp. Nhà cung cấp cũng sẽ cung cấp việc cập nhật cho các chữ ký này.

Trả lời bài tập

Trả lời cho Quyển 2 Chương 1 (Phát triển phần mềm)

Danh sách đáp án

Đáp án

• Q1. b	• Q2. b	• Q3. b	• Q4. b	• Q5. a
• Q6. a	• Q7. a	• Q8. b	• Q9. d	• Q10. d
• Q11. c	• Q12. a	• Q13. d	• Q14. c	• Q15. a
• Q16. a	• Q17. c	• Q18. c	• Q19. b	• Q20. c
• Q21. d	• Q22. b	• Q23. b	• Q24. d	• Q25. e

•

Trả lời và mô tả

• Q1

• Trả lời

- Cái gọi là "tính toán người dùng cuối", trong đó tự bản thân người dùng xây dựng hệ thống và có quyền truy nhập hay xử lý thông tin cho ứng dụng riêng của mình, đang thu được sự ủng hộ.

• Mô tả

Trong câu hỏi này cần xác định đặc trưng thích hợp nhất của việc dùng công nghệ thông tin trong doanh nghiệp ngày nay.

- Theo quan điểm sinh lợi, doanh nghiệp làm tăng mức độ chế tạo nội bộ mà không khoán ngoài các hệ thống hay vận hành cho các công ti ngoài.
- Để cắt giảm chi phí hay để rút ngắn thời gian phát triển phần mềm, các ứng dụng đã được phát triển dựa trên đơn hàng nhận được thay vì dùng gói phần mềm.
- Việc dùng mạng ngày càng rộng làm giảm phạm vi bị ảnh hưởng bởi hỏng hóc hệ thống, làm cho việc quản lí an ninh được dễ dàng hơn.

a. Khoán ngoài tăng vì chi phí thấp phát triển trong nhà. → sai

b. EUC (Máy tính người dùng cuối) → Đúng, đây là câu trả lời

c. Câu này cho mục đích (thời gian phát triển ngắn và cắt giảm chi phí) sử dụng phát triển gói thì tốt hơn. → Sai

d. Trong các hệ thống mạng, vùng ảnh hưởng bởi lỗi hệ thống trở nên rộng hơn, và đe dọa từ mạng gia tăng. → Sai

• Q2

• Trả lời

- • Việc phát triển hệ thống được thực hiện theo trật tự các tiến trình, không quay trở lại công việc ở tiến trình mức cao hơn.

• Mô tả

Trong câu hỏi này cần xác định giải thích thích hợp nhất về mô hình thác đổ.

Trong mô hình thác đổ, mỗi giai đoạn phải hoàn thành trước khi giai đoạn tiếp theo bắt đầu. Nó không cung cấp phản hồi giữa các giai đoạn hoặc cập nhật/định nghĩa lại các giai đoạn sớm hơn. → Câu trả lời là b.

- • Ứng dụng được chia ra thành các đơn vị nhỏ. Sau đó từng đơn vị trong chúng được thiết kế và chế tạo tuần tự cái nọ tiếp cái kia.
- • Một sản phẩm thử nghiệm làm việc được, được sinh ra, và việc kiểm tra các đặc tả yêu cầu và ước lượng được thực hiện trong giai đoạn sớm.
- • Thời gian phát triển được rút ngắn bởi sự tham gia của người dùng, bằng việc phát triển với ít kỹ sư hơn và bằng việc dùng có hiệu quả các công cụ phát triển.

• Q3

• Trả lời

- • Một sản phẩm thử nghiệm được tạo ra trong giai đoạn đầu của việc phát triển hệ thống, tạo khả năng loại bỏ sự mơ hồ và khác biệt trong hiểu biết của người dùng và tổ chức phát triển.

• Mô tả

Trong câu hỏi này cần xác định mô tả thích hợp nhất về bản mẫu.

Mô hình bản mẫu là một mô hình quy trình mà các bản mẫu được tạo ra trong các giai đoạn sớm của phát triển phần mềm. Nó được sử dụng ở cả phía nhà phát triển và khách hàng để kiểm tra các đặc tả yêu cầu cho hệ thống. Trong trường hợp này, khách hàng và nhà phát triển có thể xác định những khác nhau trong quan điểm của họ về những gì họ đang cố gắng tạo ra.

- • Công việc được thực hiện theo thứ tự lập kế hoạch cơ sở, thiết kế ngoài, thiết kế trong, thiết kế chương trình, lập trình và kiểm thử. Do đó, với kỹ thuật này, người ta

thu được viễn cảnh của công việc như một tổng thể, làm cho việc xác định lịch biểu và phân bổ tài nguyên được dễ dàng hơn.

Câu này mô tả mô hình thác nước.

- Một sản phẩm thử nghiệm được tạo ra trong giai đoạn đầu của việc phát triển hệ thống, tạo khả năng loại bỏ sự mơ hồ và khác biệt trong hiểu biết của người dùng và tổ chức phát triển.

Câu này mô tả về bản mẫu. → Đây là câu trả lời

- Một ứng dụng qui mô lớn được chia thành các đơn vị con, từng đơn vị này độc lập cao. Sau đó với từng đơn vị con, tiến trình thiết kế, lập trình và kiểm thử được lặp lại, mở rộng dần miền phát triển.

Câu này mô tả mô hình xoắn ốc, bao gồm các quy trình như thiết kế, lập trình và kiểm thử được lặp đi lặp lại cho mỗi đơn vị con của hệ thống, sự phát triển tạo ra sự lặp đi lặp lại và gấp lên nhiều lần

• Q4

• Trả lời

- a-f-b-e-c-d

• Mô tả

Trong câu hỏi này cần xác định lựa chọn đưa ra thứ tự đúng về quy trình phát triển.

- Vấn đề hiện tại được khảo sát và phân tích, sau đó các yêu cầu về hệ thống đích được xác định.
- Các chức năng cần cho việc xây dựng hệ thống được phân hoạch thành các chương trình để làm cho luồng tiến trình được rõ ràng hơn.
- Các thủ tục xử lý được thiết kế, mã hoá và chỉnh sửa.
- Các kiểm thử được tiến hành.
- Thiết kế có cấu trúc của từng chương trình được thực hiện dựa trên tài liệu thiết kế trong.
- Dựa trên yêu cầu về hệ thống, các chức năng cần thiết cho hệ thống được xác định.

•

- a-f-b-c-e-d

- a-f-b-e-c-d

- a-f-e-b-c-d

- a-f-e-c-b-d

a. Phân tích yêu cầu, b. Thiết kế trong, c. Thực hiện, d. Kiểm thử, e. Thiết kế chương trình f. Thiết kế ngoài

Vì vậy thứ tự là a, f, b, e, c và d. → Câu trả lời là b)

-
-
-

• Q5

• Trả lời

- • Các đặc tả thiết kế được tạo ra từ phần mềm đã cài đặt. Rồi phần mềm được phát triển dựa trên các đặc tả đã được tạo ra đó.

• Mô tả

Trong câu hỏi này cần xác định giải thích thích hợp nhất về kỹ nghệ phần mềm đảo.

“Kỹ nghệ đảo” nghĩa là quy trình đảo ngược các đặc tả phần mềm có sẵn. Phần mềm có sẵn được phân tích để các cấu thành của hệ thống và mối quan hệ giữa chúng được xác định và tạo ra diễn tả hệ thống → Câu trả lời là a.

- • Phần mềm được thiết kế để đưa ra, xử lý và đưa vào.
- • Các chức năng đã được cài đặt bằng phần mềm được đạt tới bằng phần cứng.
- • Một ngôn ngữ phát triển và công cụ phát triển được tuyển chọn tùy theo các đặc trưng xử lý của phần mềm.

• Q6

• Trả lời

- • DFD

• Mô tả

Trong câu hỏi này cần xác định lưu đồ được sử dụng trong phân tích theo cấu trúc nơi luồng dữ liệu và chức năng được diễn tả bằng các ký hiệu tương ứng chỉ ra luồng dữ liệu, xử lý (chức năng), lưu trữ dữ liệu, và bên ngoài (nguồn dữ liệu và nơi nhận dữ liệu).

- • DFD • • ERD • • Sơ đồ NS
- • Biểu đồ chuyển trạng thái • • Biểu đồ Warnier

Trong số các lựa chọn ở trên, DFD (Biểu đồ luồng dữ liệu) mô tả hệ thống tập trung vào luồng dữ liệu của hệ thống.

Mũi tên diễn tả luồng dữ liệu, hộp hình chữ nhật là quy trình, và hai đường thẳng diễn tả lưu trữ dữ liệu. Hình chữ nhật diễn tả nguồn dữ liệu hoặc nơi nhận dữ liệu.

- **Q7**

- **Trả lời**

- • Sơ đồ NS

- **Mô tả**

- Trong câu hỏi này, cần tìm biểu đồ sử dụng trong chương trình cấu trúc và diễn tả cấu trúc thực thể của chương trình dưới dạng cấu trúc phân cấp.

- • Sơ đồ • • Biểu đồ • • Biểu đồ chuyển • • Sơ đồ
• NS • PERT • trạng thái • Bubble

Sơ đồ NS do Nassi và Schneiderman phát triển.

Trong biểu đồ này, toàn bộ cấu trúc của chương trình được diễn tả theo cấp bậc. → Câu trả lời là a.

Không giống như phương pháp sơ đồ luồng, Sơ đồ NS đưa ra luồng logic mà không sử dụng các kết nối.

Biểu đồ này còn được gọi là “biểu đồ cấu trúc ”

- **Q8**

- **Trả lời**

- • Bao bọc

- **Mô tả**

Trong câu hỏi này cần xác định kỹ thuật lập trình hướng đối tượng tạo ra các chi tiết của sự thực hiện đối tượng vô hình bằng cách đặt các phương pháp và dữ liệu với nhau.

- • Thể hiện • • Bao bọc

- • Kết cụm
- • Trừu tượng hóa

Trong lập trình hướng đối tượng, các đối tượng bao bọc các phương pháp và dữ liệu. Người sử dụng đối tượng xem đối tượng như hộp đen cung cấp các dịch vụ. Các chi tiết thực hiện đối tượng không ẩn đối với người sử dụng. Điều này gọi là bao bọc. → Câu trả lời là b.

• Q9

• Trả lời

- • Lớp có thể kế thừa các động pháp từ lớp bố mẹ.

• Mô tả

Trong câu hỏi này cần xác định mô tả thích hợp nhất về lập trình hướng đối tượng.

- • Dữ liệu trao đổi giữa các sự vật được thực hiện qua các thể hiện.
- • Sự vật chỉ ra các mô tả về đặc trưng lớp.
- • Bao bọc chỉ ra việc gắn các lớp lại thành thư viện.
- • Lớp có thể kế thừa các động pháp từ lớp bố mẹ.

b. Đối tượng là thể hiện của một lớp.

c. Bao bọc nghĩa là đặc các phương pháp và dữ liệu của đối tượng lại với nhau.

d. Đúng. → Câu trả lời là d.

• Q10

• Trả lời

- • Để hoàn thành một thao tác xử lý, thiết kế nên được làm sao cho việc dùng đưa dữ liệu vào và trở lại màn hình trước không thể được phép.

• Mô tả

Trong câu hỏi này, cần xác định các mục không thích hợp được xem xét trong thiết kế màn hình ngoài và thiết kế trong.

- • Trong việc chuyển màn hình, một động pháp lựa trực tiếp được dự định dành cho người dùng có kinh nghiệm nên được dùng thay cho việc lựa từng bước bằng việc dùng menus.
- • Khoản mục đưa vào trên màn hình nên được bao bằng hay [] để làm rõ ràng rằng khoảng cách đó là dành cho trường đưa vào.
- • Bố trí màn hình nên được thiết kế sao cho các khoản mục được tham khảo có thể được bố trí từ trái sang phải hay từ trên xuống dưới.
- • Để hoàn thành một thao tác xử lí, thiết kế nên được làm sao cho việc dùng đưa dữ liệu vào và trở lại màn hình trước không thể được phép.
- • Bố trí màn hình được chuẩn hoá, chẳng hạn, các vị trí thống nhất cho hiển thị tiêu đề và thông báo, nên được dùng.

Trong thiết kế màn hình, giao diện con người được đưa ra xem xét.

(ví dụ, chuyển trạng thái màn hình, cách bố trí các mục vào và ra trong mỗi màn hình)

c. Không cho phép treo đầu vào và trở lại màn hình trước làm cho hoạt động của người sử dụng bị giới hạn và kém linh hoạt → Không thích hợp → Đây là câu trả lời

Tính kiên định trong thiết kế màn hình cũng được chú ý.

• Q11

• Trả lời

- • Số chủ yếu nên được dùng làm mã, và chữ Trung Quốc không nên dùng.

• Mô tả

Trong câu hỏi này cần xác định mô tả thích hợp nhất về quản lý mã và thiết kế mã.

- • Mã không tránh khỏi thay đổi, cho nên điều quan trọng là đặt sách mã theo thứ tự và quản lí chúng.
- • Điều mong muốn là tự chúng có thể hiểu được. Do đó việc dùng mã dài là tốt hơn.
- • Số chủ yếu nên được dùng làm mã, và chữ Trung Quốc không nên dùng.
- • Mã nên được gán để làm cho việc phân loại dữ liệu dễ dàng hơn, nhưng việc bổ sung và mở rộng mã không nên được tính tới.

Thiết kế mã, như xác định một hệ thống mã, được tiến hành như một trong các hoạt động thiết kế ngoài.

Trong hệ thống mã, nên tránh sử dụng các ký tự trung quốc. → Câu trả lời là c.

d. Hệ thống mã tạo sự phân loại dữ liệu dễ hơn. (Câu trên là đúng). Nhưng thêm hoặc mở rộng mã

không nên được xem xét. (Sai. Điều này nên được xem xét.)

- **Q12**

- **Trả lời**

- • 0

.

- **Mô tả**

Trong câu hỏi này cần tìm số cho trong ô trống theo mã 4 số "81□6".

Số kiểm tra là

$$C = \text{mod}((N_1 \times 3 + N_2 \times 2 + N_3 \times 1), 10)$$

Ở đây, $\text{mod}(a, b)$ nghĩa là phần dư của phép chia a / b .

81□6

Giả sử p là giá trị trong ô trống.

Thì đẳng thức sau là đúng.

$$\text{mod}((8 \times 3 + 1 \times 2 + p \times 1), 10) = 6$$

$$\text{mod}((26 + p \times 1), 10) = 6$$

Nếu $p=0$ thì $\text{mod}(26, 10)=6$

Nếu $p=2$ thì $\text{mod}(28, 10)=8$

Nếu $p=4$ thì $\text{mod}(30, 10)=0$

Nếu $p=6$ thì $\text{mod}(32, 10)=2$

Nếu $p=8$ thì $\text{mod}(34, 10)=4$

Vì vậy, câu trả lời là a).

- • 0 • • 2 • • 4 • • 6 • • 8

.

- **Q13**

- **Trả lời**

- • Cho dù nhiều nhiệm vụ thực hiện chương trình song song, ta vẫn thu được kết quả đúng.

.

- **Mô tả**

Trong câu hỏi này cần tìm giải thích thích hợp nhất về chương trình đồng dụng.

- • Chương trình cho kết quả đúng ngay cả nếu nó được thực hiện lại mà không phải nạp lại sau một lần đã thực hiện.
- • Được đặt vào bất kì địa chỉ nào trong bộ nhớ thật, chương trình đều có thể được thực hiện.
- • Chương trình được phân hoạch thành nhiều đoạn, và có thể được nạp vào và thực hiện trên cơ sở từng đoạn.
- • Cho dù nhiều nhiệm vụ thực hiện chương trình song song, ta vẫn thu được kết quả đúng.

a mô tả chương trình “khả năng tái sử dụng”.

b mô tả chương trình “khả năng tái định vị”.

c mô tả chương trình “chèn lấp”.

d mô tả chương trình “đồng dụng”. → Đây là câu trả lời

• Q14

• Trả lời

- • Lập trình có cấu trúc được mô tả bằng ba cấu trúc cơ sở "tuần tự", "tuyển chọn" và "lặp"

• Mô tả

Trong câu hỏi này cần xác định giải thích thích hợp nhất về lập trình theo cấu trúc.

- • Lập trình có cấu trúc được mô tả bằng ba cấu trúc cơ sở "tuần tự", "tuyển chọn" và "lặp"

c mô tả lập trình theo cấu trúc. → Đây là câu trả lời

- • Lập trình có cấu trúc có nghĩa là đưa ra qui tắc viết tụt dòng cho việc viết mã để làm cho bản in chương trình gốc được dễ đọc.
- • Lập trình có cấu trúc có nghĩa là dùng ghi chú một cách có hiệu quả để làm cho dễ hiểu phương pháp tiến trình chỉ bằng việc đọc chúng.
- • Lập trình có cấu trúc là đặt kích cỡ chuẩn cho mô đun phải là từ 50 đến 150 bước..

a, b, d là mã hóa các quy ước để đảm bảo khả năng đọc của chương trình, nhưng không mô tả lập trình theo cấu trúc.

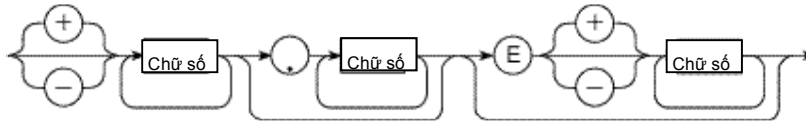
• Q15

• Trả lời

• • 5.2E -
• 07

• **Mô tả**

Trong câu hỏi này cần tìm biểu thức đáp ứng cú pháp được xác định theo biểu đồ cú pháp sau.



• • 5.2E - • • + 1.E4 • • - .9 • • 9.89E
• 07

Phần này diễn tả dấu cộng hoặc trừ có thể lựa chọn ở bên trái nhất	Sau đó, một hoặc nhiều số lựa chọn theo phần phân số hoặc điểm thập phân (cũng có thể một hoặc nhiều số)	Cuối cùng, "E" và lựa chọn dấu trừ hoặc dấu cộng và một hoặc nhiều chữ số
	5.2	E-07

a. 5.2E-07 phù hợp với cú pháp ở trên. Vì vậy câu trả lời là a.

• **Q16**

• **Trả lời**

• • -2
•

• **Mô tả**

Trong câu hỏi này, cần tìm câu trả lời đúng cho công thức được mô tả dưới đây theo ký pháp ba lan ngược "4 3 5 - ÷"

• • -2 • • -0. • • 0.2 • • 0.5 • • 5
• 2

Ký pháp ba lan ngược "4 3 5 - ÷" nghĩa là

$$4 \div (3-5) = 4 \div (-2) = -2$$

Vì vậy câu trả lời là a.

- **Q17**

- **Trả lời**

- • Nó là ngôn ngữ lập trình hướng sự vật.

- **Mô tả**

- Trong câu hỏi này cần tìm giải thích đúng về Java.

- • Nó là giao thức truyền thông được dùng trên Internet.

- • Nó là trình duyệt cho Internet.

- • Nó là kĩ thuật viết mã cho dữ liệu ảnh màu tĩnh.

Java là ngôn ngữ lập trình hướng đối tượng được phát triển bởi Sun Microsystems. Nó có vùng ứng dụng rộng, từ các ứng dụng nhỏ (gọi là applets) chạy trên trình duyệt web, các ứng dụng đứng một mình tới các ứng dụng phía máy phục vụ. Các ứng dụng Java có thể được phân phối dễ dàng qua Internet, hoặc bất kỳ mạng nào mà không cần hệ thống hoạt động hoặc các vấn đề nền phần cứng có khả năng so sánh.

- **Q18**

- **Trả lời**

- • Bao phủ điều kiện

- **Mô tả**

- Trong câu hỏi này cần tìm phương pháp thiết kế trường hợp kiểm thử sử dụng trong kiểm thử hộp trắng.

- • Đồ thị nhân quả

- • Phương pháp thiết kế thực nghiệm

- • Bao phủ điều kiện

- • Phân hoạch tương đương

Kiểm thử hộp trắng là kiểm thử đặc tả trong của chương trình và chủ yếu sử dụng các đặc tả của các thuật toán chi tiết và nguồn chương trình để kiểm tra. Trường hợp kiểm thử hộp trắng như các trường hợp sử dụng bao phủ điều kiện, bao phủ điều kiện quyết định và bao phủ điều kiện.

Vì vậy, câu trả lời là c (Bao phủ điều kiện).

a,b,d là các phương pháp thiết kế trường hợp kiểm thử cho kiểm thử hộp đen.

- **Q19**

- **Trả lời**

- • Các giá trị cực đại và cực tiểu, và những giá trị cộng 1 vào các giá trị này

- **Mô tả**

Trong câu hỏi này, cần xác định kiểm thử thích hợp nhất được sử dụng trong phân tích giá trị biên.

- • Các giá trị cực đại và cực tiểu
- • Các giá trị cực đại và cực tiểu, và những giá trị cộng 1 vào các giá trị này
- • Giá trị cực tiểu và giá trị đó cộng với 1
- • Giá trị cực đại và giá trị đó cộng với 1

Trong kiểm thử hộp đen, kỹ thuật kiểm thử như phân hoạch tương đương, phân tích giá trị biên, đồ thị nhân quả và đánh giá lỗi được sử dụng.

Trong phân tích giá trị biên, các giá trị dữ liệu kiểm thử được chọn để chấp nhận các thái cực dữ liệu. Giá trị biên bao gồm giá trị nhỏ nhất, lớn nhất, biên trong/biên ngoài, giá trị điển hình, và giá trị lỗi. Hy vọng rằng, nếu hệ thống làm việc đúng với các giá trị cụ thể thì nó sẽ làm việc đúng với tất cả các giá trị ở giữa. Vì vậy câu trả lời là b.

- **Q20**

- **Trả lời**

- • Các khoản mục được kiểm điểm được lựa chọn trước. Sau đó tài liệu được kiểm điểm nhanh chóng bằng việc kiểm từng khoản mục mỗi lúc.

- **Mô tả**

Trong câu hỏi này cần xác định mô tả thích hợp nhất về giám định.

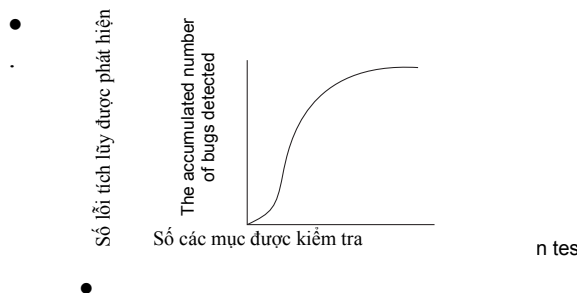
- • Kiểm điểm xem như một toàn thể được tiến hành với từng thành viên lần lượt giữ vai trò trách nhiệm.
- • Để kiểm điểm, một phần của phần mềm đích được tạo ra bằng thực nghiệm rồi được thực hiện thực tế.

- Các khoản mục được kiểm điểm được lựa chọn trước. Sau đó tài liệu được kiểm điểm nhanh chóng bằng việc kiểm từng khoản mục mỗi lúc.
- Người có thẩm quyền về tài liệu thiết kế được kiểm điểm làm chủ tịch cuộc họp kiểm điểm.

- Trong giám định, “người điều tiết” là người đảm nhiệm, chú không phải lần lượt các thành viên.
- mô tả bản mẫu. Trong giám định, các tài liệu được phân phối và xem xét chứ không phải các bản mẫu.
- Đúng --> đây là câu trả lời
- Ai tổ chức giám định là “người điều tiết”.

• Q21

• Trả lời



• Mô tả

Trong câu hỏi này cần tìm đồ thị minh họa chất lượng được ổn định.
(chất lượng phần mềm được quan sát trong đồ thị)

Câu trả lời là d.

Đây gọi là “đường cong tăng trưởng độ tin cậy của phần mềm”

• Q22

- **Trả lời**

- Chức năng phân tích và định nghĩa hệ thống

- **Mô tả**

Trong câu hỏi này cần tìm chức năng xác định như **chức năng thượng lưu của các công cụ CASE**

- | | |
|--|--|
| • Chức năng phân tích chương trình nguồn | • Chức năng phân tích và định nghĩa hệ thống |
| • Chức năng hỗ trợ kiểm thử | • Chức năng sinh chương trình tự động |
| • Chức năng quản lý dự án | • |

Các chức năng thượng lưu của CASE

- câu này cho kỹ nghệ đảo ngược và CASE hạ lưu
- đúng (đây là câu trả lời)
- hỗ trợ kiểm thử là CASE hạ lưu
- sinh mã là CASE hạ lưu
- quản lý dự án có liên quan tới toàn bộ quy trình, hoặc cả hai giai đoạn lên và xuống.

- **Q23**

- **Trả lời**

- Nó chỉ ra liệu các chức năng được yêu cầu cho phần mềm bao giờ cũng có thể được bảo trì thông thường theo những điều kiện đã được chỉ định không.

- **Mô tả**

Trong câu hỏi này cần xác định giải thích của từ “ độ tin cậy”.

- Nó chỉ ra các thao tác có thể được làm chủ dễ dàng thế nào.
- Nó chỉ ra liệu các chức năng được yêu cầu cho phần mềm bao giờ cũng có thể được bảo trì thông thường theo những điều kiện đã được chỉ định không.
- Nó chỉ ra mức độ của sửa đổi trở nên cần thiết khi phần mềm được dùng trong một môi trường máy tính khác.
- Nó chỉ ra mức độ dễ dàng theo đó các yêu cầu sửa đổi từ người dùng có thể được giải quyết.

- mô tả khả năng sử dụng
- mô tả độ tin cậy --> đây là câu trả lời
- mô tả khả năng di chuyển
- mô tả khả năng duy trì

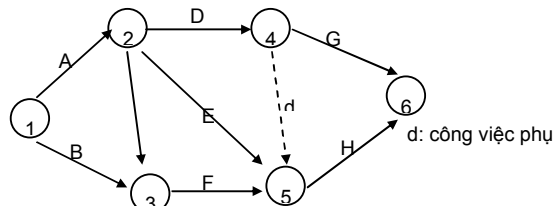
• **Q24**

• **Trả lời**

• • 7

• **Mô tả**

Trong câu hỏi này cần tìm ngày bắt đầu sớm nhất cho công việc H được tính theo biểu đồ PERT sau.



Số nút	Thời gian bắt đầu sớm nhất tại nút	Nhiệm vụ làm trước	Đánh dấu
1	0	-	Ngày bắt đầu dự án =0
2	3	A	A(3 ngày)
3	4	B,C	A(3 ngày)+C(1ngày)
4	7	D	A(3 ngày)+D(4 ngày)
5	7	D,E,F	A(3 ngày)+D(4 ngày)
6	8	G,H	A(3 ngày)+D(4 ngày)+G(1 ngày)

Thời gian bắt đầu sớm nhất của hoạt động H là thời gian bắt đầu sớm nhất của nút 5.

Vì vậy câu trả lời là 7 ngày. → đáp án là d.

• • 4 • • 5 • • 6 • • 7 • • 8

• **Q25**

• **Trả lời**

• • Hệ thống tổ người lập trình chính cần một thủ thư.

• **Mô tả**

Trong câu hỏi này cần tìm mô tả thích hợp nhất của các đặc trưng của hệ thống tổ người lập trình chính được so sánh với hệ thống phân cấp.

- • Khối lượng trọng trách của người lãnh đạo trong hệ phân cấp là nặng hơn trong hệ thống tổ người lập trình chính.
- • Hệ thống phân cấp ít thích hợp cho các hệ thống lớn hơn là hệ thống tổ người lập trình chính.
- • Trao đổi bên trong tổ hệ thống phân cấp là dễ dàng hơn trong hệ thống tổ

- . người lập trình chính.
- • Hệ thống tổ người lập trình chính không cần người lập trình dự phòng.
- .
- • Hệ thống tổ người lập trình chính cần một thủ thư.
- .

Trong hệ thống tổ người lập trình chính, thủ thư chương trình trả lời để lưu trữ tất cả các phiên bản của tất cả các chương trình và dữ liệu được tạo về dự án → Câu trả lời là e.

Trả lời cho Quyển 2 Chương 2 (Bảo trì và vận hành hệ thống)

Danh sách đáp án

Đáp án _____

- | | | | | |
|---------|---------|---------|---------|---------|
| • Q1. c | • Q2. d | • Q3. c | • Q4. b | • Q5. d |
| • Q6. b | • Q7. d | • Q8. b | • | • |

Trả lời và mô tả

- Q1
- Trả lời
- • Người lập trình

- **Mô tả**

Trong câu hỏi này cần xác định thành phần nào không được đưa vào tài nguyên cần quản lý trong quản lý vận hành.

- • Máy tính
- • Cơ sở dữ liệu
- • Người lập trình
- • Chương trình

- a. máy tính là tài nguyên được quản lý trong giai đoạn bảo dưỡng và vận hành.
- b. cơ sở dữ liệu cũng tương tự như máy tính.
- c. người lập trình không bị quản lý --> đây là câu trả lời
- d. chương trình cũng được quản lý.

- **Q2**

- **Trả lời**

- • Xuất hiện hỏng → tìm chỗ hỏng → c → d → a → tiến hành biện pháp thường xuyên → b

- **Mô tả**

Trong câu hỏi này cần xác định thủ tục chuẩn đúng được thực hiện trong việc hỏng hệ thống.

- • Xuất hiện hỏng → tìm chỗ hỏng → a → c → d → tiến hành biện pháp thường xuyên → b
- • Xuất hiện hỏng → tìm chỗ hỏng → b → c → a → tiến hành biện pháp thường xuyên → d
- • Xuất hiện hỏng → tìm chỗ hỏng → c → a → b → tiến hành biện pháp thường xuyên → d
- Xuất hiện hỏng → tìm chỗ hỏng → c → d → a → tiến hành biện pháp thường xuyên → b

Trả lời là:

d hỏng --> tìm

--> xác định --> cách ly --> biện pháp tạm thời

--> biện pháp tạm thời --> chuyển biện pháp tạm thời

Trước tiên, phần hỏng phải được xác định và cách ly, sau đó tiến hành các biện pháp tạm thời.

- **Q3**

- **Trả lời**

- • Nếu quyền được cấp cho một user ID, nó phải bị giới hạn ở số tối thiểu nhất có thể được.

- **Mô tả**

Trong câu hỏi này cần tìm phát biểu thích hợp nhất cho quản lý ID của người dùng.

- • Mọi người dùng ID tham gia vào cùng dự án đều dùng cùng ID.
- • Người dùng có nhiều hơn một ID đặt cùng mật hiệu cho mọi ID.
- • Nếu quyền được cấp cho một user ID, nó phải bị giới hạn ở số tối thiểu nhất có thể được.
- • Việc xóa một user ID phải được tiến hành thời gian dài sau khi việc bỏ ID đó đã được lưu ý.

- Mỗi người dùng chỉ nên có một ID người dùng.
- Người sử dụng nên tránh đặt mật khẩu giống với ID với họ.
- Người dùng có quyền tối thiểu có thể có. → đúng → Đây là câu trả lời
- Khi ID người dùng không dùng nữa, ID nên bị xóa ngay lập tức.

- **Q4**

- **Trả lời**

- • Để làm giảm mức độ các mật hiệu được tham chiếu tới, cần khuyến cáo rằng người dùng ghi lại mật hiệu của mình vào sổ tay hay đâu đó khác.

- **Mô tả**

Trong câu hỏi này cần xác định giải thích không thích hợp nhất về xử lý mật hiệu và tệp mật hiệu trong tổ chức quản lý hệ thống.

- • Liệu mật hiệu có thể được đoán ra dễ dàng hay không được kiểm tra đều đặn không, và việc dùng các mật hiệu khác nhau có bị thôi thúc cho một vấn đề không.
- • Để làm giảm mức độ các mật hiệu được tham chiếu tới, cần khuyến cáo rằng người dùng ghi lại mật hiệu của mình vào sổ tay hay đâu đó khác.
- • Nếu thời gian hiệu lực có thể được đặt cho mật hiệu, thì chức năng này phải được dùng.
- • Việc tham chiếu của người thường tới tệp mật hiệu phải bị cấm, cho dù các mật hiệu, đã được mật mã hoá.

a, c và d là thích hợp.

b không thích hợp vì nếu mật hiệu bị viết ra thì khả năng bị đánh cắp cao hơn.

- **Q5**

• **Trả lời**

- Thời gian gỡ lỗi

• **Mô tả**

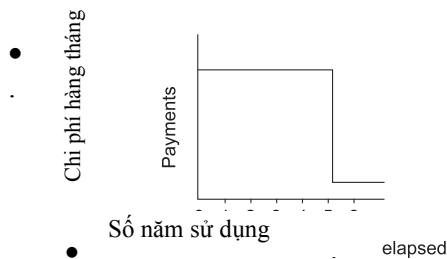
Trong câu hỏi này cần tìm việc không liên quan tới chuẩn cho các thao tác hệ trực tuyến.

- Thời gian đáp ứng
- Thời gian bắt đầu vận hành
- Thời gian khôi phục hỏng
- Thời gian gỡ lỗi

a, b và c liên quan nhưng d không liên quan tới chuẩn các thao tác hệ trực tuyến (nó là vấn đề trong phát triển chứ không phải trong thao tác thực tế)

• **Q6**

• **ả lời**



• **Mô tả**

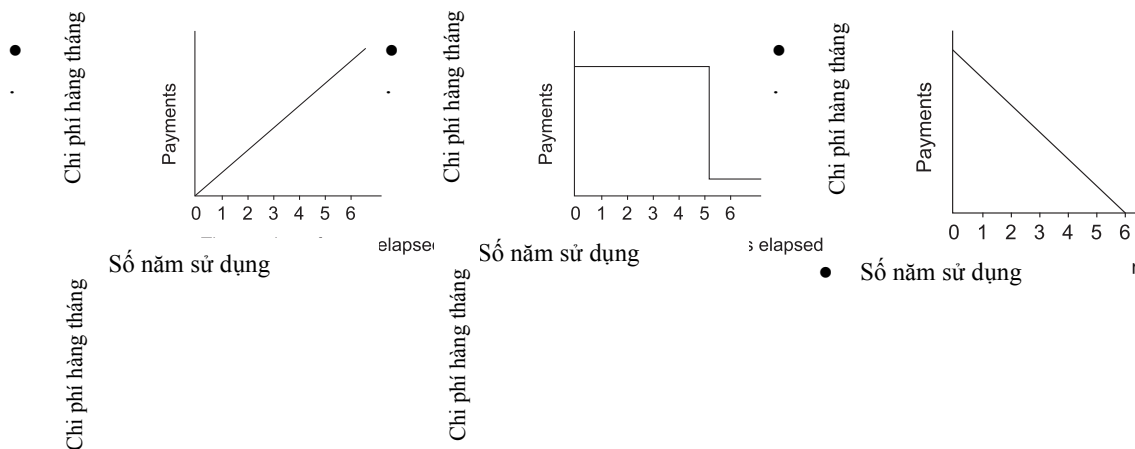
Trong câu hỏi này cần tìm đồ thị thích hợp nhất mô tả số năm, số tiền thanh toán nếu máy tính được giới thiệu theo các điều kiện dưới đây.

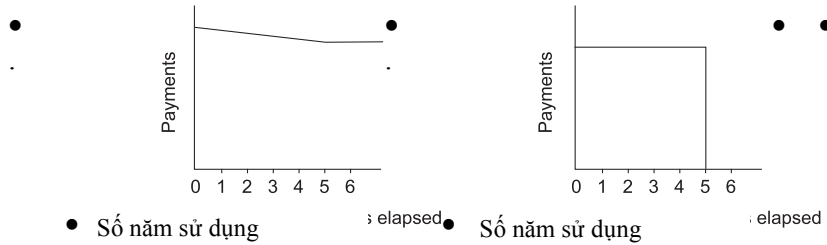
① Chi phí máy tính

- Trong năm đầu được trả, số tiền trả hàng tháng cố định (của tiền thuê) được tính dựa trên giá mua máy tính và tỉ lệ thuê của chúng.
- Từ tháng thứ sáu và những năm sau, số tiền trả hàng tháng được tính dựa trên một phần mười giá mua máy tính và tỉ lệ thuê của chúng được trả.

② Chi phí bảo trì

- Số tiền hàng tháng cố định được trả như tiền bảo trì cho công ti bảo trì.





• Q7

• Trả lời

- Kiểm thử rà lại

• Mô tả

Trong câu hỏi này cần xác định phương pháp kiểm thử phần mềm được sử dụng để kiểm tra xem liệu những thay đổi được tiến hành cho bảo trì phần mềm có ảnh hưởng tới các phần khác hay không.

- Kiểm thử vận hành
- Kiểm thử tích hợp
- Kiểm thử hệ thống
- Kiểm thử mốc nối
- Kiểm thử rà lại

a kiểm thử vận hành

Trong kiểm thử vận hành, nhóm vận hành từ bộ phận người dùng tiến hành kiểm thử theo điều kiện và môi trường tương tự với các hoạt động thực tế.

b Kiểm thử tích hợp

Kiểm thử tích hợp được thiết kế để kiểm thử các cấu phần phần mềm tích hợp để xác định xem chúng hoạt động như một chương trình thực thụ hay không. Hoạt động của chương trình và giao diện giữa các module được kiểm tra.

c kiểm thử hệ thống

Kiểm thử hệ thống đảm bảo toàn bộ hệ thống phần mềm tích hợp phù hợp với các yêu cầu.

Hoạt động của hệ thống như toàn bộ được kiểm tra tổng thể từ quan điểm của hoạt động và đối tượng được yêu cầu. Sau đó, sự bắt đầu của các hoạt động thực tế được quyết định dựa trên các kết quả.

d Kiểm thử rà lại

Kiểm thử rà lại là một loạt các kiểm thử để đảm bảo không có sự bất lợi nào thay đổi được giới thiệu tới ứng dụng trong khi thay đổi bảo trì, nâng cấp hoặc những thay đổi khác. → Đây là câu trả lời

• Q8

• Trả lời

- Người lập trình đã phát triển chương trình gốc thực hiện việc sửa đổi chương trình được liên kết với những đặc tả đã được thực hiện sau khi việc phát triển đã được hoàn tất. Sau đó, phần mềm mới được nhanh chóng đưa vào sử dụng trong môi trường

thực.

- **Mô tả**

Trong câu hỏi này cần xác định mô tả không thích hợp về công việc bảo trì cho phần mềm ứng dụng được phát triển trong nhà.

- • Người quản lý vận hành bắt đầu dùng phần mềm mới sau khi những sửa đổi đã được chấp thuận, và loại bỏ phần mềm cũ dựa trên kế hoạch đã định trước.
- • Người lập trình đã phát triển chương trình gốc thực hiện việc sửa đổi chương trình được liên kết với những đặc tả đã được thực hiện sau khi việc phát triển đã được hoàn tất. Sau đó, phần mềm mới được nhanh chóng đưa vào sử dụng trong môi trường thực.
- • Người tiến hành kiểm thử phải phân tích những lĩnh vực bị ảnh hưởng bởi sự thay đổi, tiến hành việc kiểm thử cho những bộ phận có liên hệ tới chương trình bị thay đổi, và làm đánh giá.
- • Trong việc thực hiện bảo trì, các chuẩn, phương pháp luận và thủ tục liên quan tới quản trị tài liệu, phương pháp bảo trì và thủ tục sửa đổi chương trình phải được cung cấp trước.
-

a đúng (sự thay đổi có thể dưới sự kiểm soát. Nó có thể được lập kế hoạch và chấp nhận trước hoạt động thực tế)

b sự thay đổi cần được thực hiện bởi người lập trình, người mã hóa chương trình trong câu hỏi)

c đúng (kiểm thử rà lại nên được thực hiện)

d đúng (các nguyên tắc duy trì và vận hành nên được xác định trước)

Bảng đổi chiếu thuật ngữ Anh - Việt

3GL	3GL (ngôn ngữ thế hệ 3)
4GL	4GL (ngôn ngữ thế hệ 4)
A	
abstract data type	kiểu dữ liệu trừu tượng
access rights	quyền truy nhập
Ada	Ada
adaptive maintenance	bảo trì thích nghi
aggregation and decomposition relationship	mối quan hệ kết tập và phân tách
air conditioning facilities	Tiện nghi điều hoà nhiệt độ
ALGOL	ALGOL
APL	APL
array type	kiểu mảng
attribute	thuộc tính
automatic adjustment function	chức năng điều chỉnh tự động
automaton	ô tô mát
AVR	AVR (Bộ điều chỉnh điện áp tự động)
B	
backlog	việc tồn đọng lại
backup programmer	người lập trình dự phòng
base allocation	cấp phát cơ sở
BASIC	BASIC
basic data type	kiểu dữ liệu cơ sở
basic planning	Lập kế hoạch cơ sở
battery	Pin
black box test	kiểm thử hộp đen
BNF	BNF (Dạng chuẩn Backus Naur)
bottom-up approach	cách tiếp cận dưới lên
boundary value analysis	Phân tích giá trị biên
bug	lỗi
bug curve	Đường cong lỗi
bug management diagram	Biểu đồ quản lý lỗi
business operation model	mô hình vận hành nghiệp vụ
C	
C language	ngôn ngữ C
C++	C++
CAD	CAD (Thiết kế có máy tính hỗ trợ)
CAE	CAE (Kỹ nghệ có máy tính hỗ trợ)
CAM	CAM (Chế tạo có máy tính hỗ trợ)
CASE	CASE (kỹ nghệ phần mềm có máy tính hỗ trợ)
CASE tool	công cụ CASE
CASE tool supplying development platform services	Công cụ CASE cung cấp các dịch vụ nền phát triển
cause-effect graph	Đồ thị nhân-quả
cause-result graph	Đồ thị nhân-quả
cell	ô

change control administrator	Người quản trị kiểm soát thay đổi
character type	kiểu kí tự
charging	tính cước
check digit method	Phương pháp chữ số kiểm tra
chief programmer	người lập trình chính
chief programmer team	tổ (của) người lập trình chính
Chomsky	Chomsky
class	lớp
class library	thư viện lớp
COBOL	COBOL
COCOMO (COstruction COst MOdel)	COCOMO (COstruction COst MOdel)
model	
CODASYL	CODASYL (Hội thảo về ngôn ngữ hệ thống dữ liệu)
code	bộ mã
code book	Sách bộ mã
code design	thiết kế (bộ) mã
code review	kiểm điểm mã
coding	Mã hoá
common CASE tool	công cụ CASE chung
communications function	chức năng truyền thông
compiler	Trình biên dịch
component libraries	Thư viện cấu phần
components-creating tool	công cụ tạo cấu phần các công cụ tạo cấu phần
components-searching tool	công cụ tìm cấu phần
componentware	componentware
compound object	đối tượng hợp thành
condition coverage	Bao phủ điều kiện
configuration administrator	Người quản trị cấu hình
control structure	Cấu trúc điều khiển
CORBA	CORBA (Common Object Request Broker Architecture)
correction task	Nhiệm vụ sửa chữa
corrective maintenance	bảo trì sửa chữa
cost management	quản lí chi phí
coverage	Bao phủ
CPM	CMP (phương pháp đường găng)
crime prevention facilities	Tiện nghi ngăn ngừa tội phạm
critical path	đường găng
CTI	CTI (Tích hợp điện thoại máy tính)
customize	chuyên biệt hoá
D	
daily maintenance	Bảo trì hàng ngày
data	dữ liệu
data audit	kiểm toán dữ liệu
Data Flow Diagram	Sơ đồ luồng dữ liệu
data resource management	Quản lí tài nguyên dữ liệu
data trouble	Trục trặc dữ liệu
data type	kiểu dữ liệu

data-oriented approach
 data-oriented design
 decision coverage
 decision table
 decision/condition coverage
 delegation
 design review
 designing document
 designing files
 designing physical data
 designing programs
 designing screen
 detailed input-output design
 development organization
 development plan
 development support tools
 development-promoting organization
 DFD
 difference programming
 disaster prevention facilities
 DOA
 Do-While type
 dynamic program analysis tool
 dynamically re-locatable program
 E
 earliest possible linkage time
 editing and processing functions
 editor
 emergency maintenance
 encapsulation
 encryption
 end user computing
 Enterprise Resource Planning
 entity
 entry/exit log
 equivalent class
 equivalent partitioning
 ERD
 ERP
 estimation based on data in the past
 EUC
 EUD
 event-driven
 experimental design method
 external design
 external design documents
 extranet
 F
 facility management
 failure tendency monitoring

cách tiếp cận hướng dữ liệu
 thiết kế hướng dữ liệu
 Bao phủ quyết định
 bảng quyết định
 bao phủ quyết định/điều kiện
 uỷ quyền
 kiểm điểm thiết kế
 thiết kế tài liệu
 thiết kế tệp
 thiết kế dữ liệu vật lí
 thiết kế chương trình
 thiết kế màn hình
 thiết kế vào ra chi tiết
 Tổ chức phát triển
 kế hoạch phát triển
 công cụ hỗ trợ phát triển
 Tổ chức thúc đẩy phát triển
 DFD (sơ đồ luồng dữ liệu)
 lập trình khác biệt
 Tiện nghi phòng ngừa thảm hoạ
 tiếp cận hướng dữ liệu (DOA)
 kiểu Do-While
 công cụ phân tích chương trình động
 chương trình khả tái định vị động

 Thời gian móc nối sớm nhất có thể
 chức năng soạn thảo và xử lí
 Trình soạn thảo
 Bảo trì khẩn cấp
 bao bọc
 mật mã hoá
 tính toán (của) người dùng cuối
 lập kế hoạch tài nguyên xí nghiệp (ERP)
 thực thể
 nhật kí vào/ra
 lớp tương đương
 phân hoạch tương đương
 ERD (Biểu đồ thực thể quan hệ)
 ERP (lập kế hoạch tài nguyên xí nghiệp)
 Ước lượng dựa trên dữ liệu quá khứ
 EUC (tính toán (của) người dùng cuối))
 EUD (phát triển người dùng cuối)
 điều khiển theo biến cố
 Phương pháp thiết kế thực nghiệm
 thiết kế ngoài
 Tài liệu thiết kế ngoài
 mạng ngoại bộ (extranets)

 Quản lí tiện nghi
 Giám sát xu hướng hỏng hóc

fault injection	Len lỗi
fixed cost	chi phí cố định
flowcharts	Lưu đồ
formal language	ngôn ngữ hình thức
formatting function	chức năng định dạng
FORTRAN	FORTRAN
forward engineering	Kỹ nghệ tiến
fourth generation language	Ngôn ngữ thế hệ thứ tư
FP (Function Point) method	phương pháp điểm chức năng (FP)
functional language	ngôn ngữ hàm
functional partitioning	Phân hoạch chức năng
functional programming	lập trình chức năng
G	
Gantt chart	Sơ đồ Gantt
generalization	tổng quát hoá
generalization and specialization	các quan hệ tổng quát hoá và khu biệt hoá
relationships	
generative grammar	văn phạm sinh
groupware	Phần mềm nhóm
GUI	GUI (giao diện người dùng (bằng) đồ hoạ)
H	
hardware maintenance	Bảo trì phần cứng
hardware resource management	quản lý tài nguyên phần cứng
hardware trouble	Trục trặc phần cứng
HCP	HCP (Sơ đồ mô tả gọn và có thứ bậc)
hierarchical team	tổ phân cấp
highest-level schedule	Lịch biểu mức cao nhất
high-level language	ngôn ngữ cấp cao
HIPO	Cấp bậc cộng với Vào Xử lý Ra (HIPO)
HTML	HTML (ngôn ngữ đánh dấu siêu văn bản)
human interface	giao diện con người
I	
If-then-else type	kiểu If-then-else
important phenomena monitoring	Giám sát các hiện tượng quan trọng
improvement task	Nhiệm vụ cải tiến
infix notation	kí pháp viết giữa
information hiding	che giấu thông tin
inheritance	Kế thừa
initial cost	chi phí khởi đầu
inspection	Giám định
instance	thể hiện, thể nghiệm
instruction coverage	Bao phủ lệnh
integer type	kiểu số nguyên
integrated CASE tool	Công cụ CASE tích hợp (I-CASE)
integrated software package	gói phần mềm tích hợp
integration tests	Kiểm thử tích hợp
integrity measure	việc đo tính toàn vẹn
interfaces between modules	giao diện giữa các mô đun
internal design	thiết kế trong
internal design documents	tài liệu thiết kế trong

internal design support tools	công cụ hỗ trợ thiết kế trong
interpreter	Trình thông dịch
intranet	mạng nội bộ intranets
invalid equivalence class	Lớp tương đương không hợp lệ
inventory status	Tình trạng kho
is_a relation	quan hệ là_một
ISO/IEC 9126	ISO/IEC 9126
J	
Japanese word processor	bộ soạn thảo tiếng Nhật
Java	Java
job scheduling	lập lịch công việc
L	
LAN	LAN (mạng cục bộ)
language processor	bộ xử lý ngôn ngữ
latest possible linkage time	Thời gian móc nối muộn nhất có thể
librarian	thủ thư
library management	quản lý thư viện
LISP	LIPS
LOC-based method	Phương pháp dựa trên LOC
logic data design	thiết kế dữ liệu logic
logic programming	lập trình logic
logical language	ngôn ngữ logic
logical type	kiểu logic
lower phase CASE tool	Công cụ CASE cho pha hạ (L-CASE)
lowest-level schedule	Lịch biểu mức thấp nhất
low-level language	ngôn ngữ cấp thấp
M	
maintainers	Người bảo trì
maintenance	Bảo trì
maintenance administrator	Người quản trị bảo trì
maintenance CASE tool	Công cụ CASE bảo trì
maintenance contract	hợp đồng bảo trì
maintenance for perfection	bảo trì hoàn thiện
member	Thành viên
message	Thông báo
method	phương pháp
methods to create components	Phương pháp tạo ra các cấu phần
middle-level schedule	Lịch biểu mức trung
moderator	người dẫn chương trình
modification task	Nhiệm vụ sửa đổi
module	mô đun
module cohesion	Tính cố kết mô đun
module coupling	Tính gắn nối mô đun
module design	Thiết kế mô đun
module design document	Tài liệu thiết kế mô đun
module partitioning	phân hoạch mô đun
multi-branch structure	Cấu trúc đa nhánh (CASE)
multiple condition coverage	Bao phủ đa điều kiện
N	
natural language	ngôn ngữ tự nhiên

network resource management
 new version
 non-procedural language
 NS chart
 O
 object
 object-oriented design
 object-oriented development
 object-oriented language
 object-oriented programming
 operation manual
 operation tests
 operational processing system
 overlay
 P
 PAD
 part_of relation
 Pascal
 password
 PDCA

 PDM
 performance balance
 performance management
 Perl
 PERT

 phrase structure grammar
 PL/I
 pointer
 pointing device
 Polish notation
 polymorphism
 post maintenance
 postfix notation
 PostScript
 power distribution facilities
 prefix notation
 presentation software
 prevention of illicit use
 preventive maintenance
 preventive modification
 problem management
 problem-oriented language
 procedural language
 procedural programming
 process
 process chart
 process management
 process model

Quản lý tài nguyên mạng
 phiên bản mới
 ngôn ngữ phi thủ tục
 Sơ đồ NS (sơ đồ Nassi-Shneiderman)

 đối tượng
 thiết kế hướng đối tượng
 phát triển hướng đối tượng
 ngôn ngữ hướng đối tượng
 lập trình hướng đối tượng
 tài liệu vận hành
 Kiểm thử vận hành
 Hệ xử lý tác nghiệp
 Chèn lấp

 PAD (Biểu đồ phân tích vấn đề)
 quan hệ bộ phận-của
 Pascal
 Mật hiệu
 Kế hoạch-Thực hiện-Kiểm tra-Hành động
 (PDCA)
 PDM (Quản lý dữ liệu sản phẩm)
 Cân bằng hiệu năng
 quản lý hiệu năng
 Perl
 PERT (kỹ thuật kiểm điểm và đánh giá
 chương trình)
 văn phạm cấu trúc câu
 PL/I
 con trỏ
 thiết bị trỏ
 kí pháp Ba lan
 đa hình thái
 bảo trì đột xuất, bảo trì về sau
 kí pháp viết sau
 PostScript
 Tiện nghi phân phối nguồn điện
 kí pháp viết đầu
 phần mềm trình bày
 Ngăn ngừa việc dùng trái phép
 Bảo trì phòng ngừa
 thay đổi phòng ngừa
 quản lý vấn đề
 ngôn ngữ hướng vấn đề
 Ngôn ngữ thủ tục
 lập trình thủ tục
 tiên trình
 sơ đồ tiến trình
 quản lý tiến trình
 mô hình tiến trình

process planning	lập kế hoạch tiến trình
process stage	giai đoạn tiến trình
process-oriented approach	Tiếp cận hướng tiến trình
process-oriented design	Thiết kế hướng tiến trình
program design documents	Tài liệu thiết kế chương trình
program flowchart	Lưu đồ chương trình
program test support tool	Công cụ hỗ trợ kiểm thử
programmer	lập trình viên
programming	lập trình
programming support tool	Công cụ hỗ trợ lập trình
progress management	Quản lý tiến độ
project implementation plan	Kế hoạch thực hiện dự án
project leader	Người lãnh đạo dự án
project manager	người quản lý dự án
Prolog	Prolog
proportional allocation	cấp phát (theo) tỉ lệ
prototype model	Mô hình bản mẫu
Q	
QFD	QFD (Triển khai chức năng chất lượng)
quality characteristics	Đặc trưng chất lượng
R	
RAD	RAD (Phát triển ứng dụng nhanh)
Rapid Application Development	Phát triển ứng dụng nhanh
RDBMS	RDBMS (hệ quản trị dữ liệu quan hệ)
recalculation function	Chức năng tính lại
record type	kiểu bản ghi
recursive	đệ qui
recursive program	chương trình đệ qui
reengineering	Tái kỹ nghệ
reentrant	đồng dụng
reentrant programs	chương trình đồng dụng
regression test	kiểm thử rà lại (hồi qui)
regular expression	biểu thức chính qui
relationship	quan hệ
reliability	Độ tin cậy
reliability growth curve	đường cong tăng trưởng độ tin cậy
reload	nạp lại
re-locatable	Khả năng tái định vị
re-locatable program	chương trình khả tái định vị
remote maintenance	bảo trì từ xa
Repeat-Until type	kiểu Repeat-Until
repository	kho chứa
requirement definition	Xác định yêu cầu
requirement specification	đặc tả yêu cầu
resource management	quản lý tài nguyên
reusable	Dùng lại được
reusable programs	chương trình dùng lại được
reverse engineering	kỹ nghệ đảo
reverse Polish notation	pháp Ba lan ngược
review	xét duyệt, kiểm điểm

running cost	chi phí vận hành (chạy)
S	
SA	phân tích có cấu trúc (SA)
schedule	lịch
scheduled maintenance	Bảo trì theo lịch
scheduling	lập lịch
SCM	SCM (Quản lý dây chuyền cung cấp)
screen design	thiết kế màn hình
screen flow	luồng màn hình
script language	Ngôn ngữ script
SD	thiết kế có cấu trúc (SD)
SE	SE (kỹ sư hệ thống)
search systems	hệ thống tìm kiếm
security management	Quản lý an ninh
segment	đoạn
sequential type	kiểu tuần tự
SFA	SFA (Tự động hoá hiệu lực bán hàng)
SGML	SGML (Ngôn ngữ đánh dấu tổng quát chuẩn)
SLC	vòng đời phần mềm (SLC)
software life cycle	vòng đời phần mềm
software maintenance	Bảo trì phần mềm
software package	gói phần mềm
software resource management	Quản lý tài nguyên phần mềm
software trouble	Trục trặc phần mềm
SPD	SPD (Biểu đồ lập trình có cấu trúc)
specialist team	Tổ chuyên gia
specialization	khu biệt hoá
specific problem oriented language	Ngôn ngữ hướng vấn đề chuyên dụng
specified monitoring	Giám sát đặc biệt
spiral model	mô hình xoắn ốc
spreadsheet	bảng tính
spreadsheet program	Chương trình trang tính
standard task-based method	phương pháp dựa trên nhiệm vụ chuẩn
standardization	Chuẩn hoá
Standards for Information System Safety Measures	Các chuẩn cho việc đo an toàn hệ thống tin
state transition diagrams	Biểu đồ chuyển trạng thái
static program analysis tool	công cụ phân tích chương trình tĩnh
step-by-step integration method	phương pháp tích hợp từng bước
step-by-step refinement method	phương pháp làm mịn từng bước
storage facilities	Tiện nghi lưu giữ
strategic information system	Hệ thống tin chiến lược
structure editor	trình soạn thảo cấu trúc
structure theorem	định lý cấu trúc
structure type	kiểu cấu trúc
structured analysis	phân tích cấu trúc
structured charts	Sơ đồ có cấu trúc
structured design	thiết kế có cấu trúc
structured design of programs	Thiết kế có cấu trúc cho chương trình

structured methods	phương pháp có cấu trúc
structuring	(lập) cấu trúc
sub-class	Lớp con
sub-project team	tổ dự án con
sub-system	hệ con
super class	siêu lớp
syntactic analysis	phân tích cú pháp
system engineer	kỹ sư hệ thống
system flowchart	Lưu đồ hệ thống
system operation tools	công cụ vận hành hệ thống
system tests	Kiểm thử hệ thống
systematization plan	Kế hoạch hệ thống hoá
T	
tabular form language	Ngôn ngữ dạng biểu bảng
TCO	TCO (Tổng chi phí của quyền làm chủ)
technical specialist	chuyên gia kỹ thuật
technical support	hỗ trợ kỹ thuật
tentative maintenance	Bảo trì thăm dò
test	kiểm thử
test coverage	Thiết kế kiểm thử
test execution support tool	Công cụ hỗ trợ thực hiện kiểm thử
test support tool	Công cụ hỗ trợ kiểm thử
text editor	Trình soạn thảo văn bản
third generation language	ngôn ngữ thế hệ 3
top-down approach	tiếp cận trên xuống
Total Cost of Ownership	Tổng chi phí của quyền làm chủ
transition	chuyển dịch (hệ thống)
U	
UML	UML (ngôn ngữ mô hình hoá thống nhất)
unit tests	kiểm thử đơn vị
upper phase CASE tool	Công cụ CASE cho pha thượng (U-CASE)
UPS	UPS (Nguồn không ngắt)
user	người dùng
user ID	ID người dùng
V	
valid equivalence class	Lớp tương đương hợp lệ
variable cost and expenses	chi phí lưu động và chi tiêu
version data	dữ liệu phiên bản
version-up	nâng phiên bản
Visual Basic	Visual Basic
W	
walkthrough	Giải trình
WAN	WAN (mạng diện rộng)
waterfall model	mô hình thác đổ
WBS	WBS (Cấu trúc phân việc).
white box test	kiểm thử hộp trắng
Work Breakdown Structure	Cấu trúc phân việc
worksheet	trang công việc
X	
XML	XML (Ngôn ngữ đánh dấu mở rộng)

Y

YAC II (Yet Another Control chart II

Sơ đồ điều khiển khác II)

Bảng đối chiếu thuật ngữ Việt - Anh

(lập) cấu trúc	structuring
3GL (ngôn ngữ thế hệ 3)	3GL
4GL (ngôn ngữ thế hệ 4)	4GL
A	
Ada	Ada
ALGOL	ALGOL
APL	APL
AVR (Bộ điều chỉnh điện áp tự động)	AVR
B	
bảng quyết định	decision table
bảng tính	spreadsheet
bao bọc	encapsulation
Bao phủ	coverage
Bao phủ đa điều kiện	multiple condition coverage
Bao phủ điều kiện	condition coverage
Bao phủ lệnh	instruction coverage
Bao phủ quyết định	decision coverage
bao phủ quyết định/điều kiện	decision/condition coverage
Bảo trì	maintenance
bảo trì đột xuất, bảo trì về sau	post maintenance
Bảo trì hàng ngày	daily maintenance
bảo trì hoàn thiện	maintenance for perfection
Bảo trì khẩn cấp	emergency maintenance
Bảo trì phần cứng	hardware maintenance
Bảo trì phần mềm	software maintenance
Bảo trì phòng ngừa	preventive maintenance
bảo trì sửa chữa	corrective maintenance
Bảo trì thăm dò	tentative maintenance
Bảo trì theo lịch	scheduled maintenance
bảo trì thích nghi	adaptive maintenance
bảo trì từ xa	remote maintenance
BASIC	BASIC
Biểu đồ chuyển trạng thái	state transition diagrams
Biểu đồ quản lý lỗi	bug management diagram
biểu thức chính qui	regular expression
BNF (Dạng chuẩn Backus Naur)	BNF
bộ mã	code
bộ soạn thảo tiếng Nhật	Japanese word processor
bộ xử lý ngôn ngữ	language processor
C	
C++	C++
Các chuẩn cho việc đo an toàn hệ thống tin	Standards for Information System Safety Measures
các quan hệ tổng quát hoá và khu biệt hoá	generalization and specialization relationships
cách tiếp cận dưới lên	bottom-up approach

cách tiếp cận hướng dữ liệu	data-oriented approach
CAD (Thiết kế có máy tính hỗ trợ)	CAD
CAE (Kỹ nghệ có máy tính hỗ trợ)	CAE
CAM (Chế tạo có máy tính hỗ trợ)	CAM
Cân bằng hiệu năng	performance balance
Cấp bậc cộng với Vào Xử lý Ra (HIPO)	HIPO
cấp phát (theo) tỉ lệ	proportional allocation
cấp phát cơ sở	base allocation
CASE (kỹ nghệ phần mềm có máy tính hỗ trợ)	CASE
Cấu trúc đa nhánh (CASE)	multi-branch structure
Cấu trúc điều khiển	control structure
Cấu trúc phân việc	Work Breakdown Structure
che giấu thông tin	information hiding
Chèn lấp	overlay
chi phí cố định	fixed cost
chi phí khởi đầu	initial cost
chi phí lưu động và chi tiêu	variable cost and expenses
chi phí vận hành (chạy)	running cost
Chomsky	Chomsky
Chuẩn hoá	standardization
chức năng điều chỉnh tự động	automatic adjustment function
chức năng định dạng	formatting function
chức năng soạn thảo và xử lý	editing and processing functions
Chức năng tính lại	recalculation function
chức năng truyền thông	communications function
chương trình đệ qui	recursive program
chương trình đồng dụng	reentrant programs
chương trình dùng lại được	reusable programs
chương trình khả tái định vị	re-locatable program
chương trình khả tái định vị động	dynamically re-locatable program
Chương trình trang tính	spreadsheet program
chuyên biệt hoá	customize
chuyển dịch (hệ thống)	transition
chuyên gia kỹ thuật	technical specialist
CMP (phương pháp đường gân)	CPM
COBOL	COBOL
COCOMO (CONstruction COst MOdel)	COCOMO (CONstruction COst MOdel)
	model
CODASYL (Hội thảo về ngôn ngữ hệ thống dữ liệu)	CODASYL
componentware	componentware
con trỏ	pointer
công cụ CASE	CASE tool
Công cụ CASE bảo trì	maintenance CASE tool
Công cụ CASE cho pha hạ (L-CASE)	lower phase CASE tool
Công cụ CASE cho pha thượng (U-CASE)	upper phase CASE tool
công cụ CASE chung	common CASE tool
Công cụ CASE cung cấp các dịch vụ nền phát triển	CASE tool supplying development platform services

Công cụ CASE tích hợp (I-CASE)	integrated CASE tool
Công cụ hỗ trợ kiểm thử	program test support tool
Công cụ hỗ trợ kiểm thử	test support tool
Công cụ hỗ trợ lập trình	programming support tool
công cụ hỗ trợ phát triển	development support tools
công cụ hỗ trợ thiết kế trong	internal design support tools
Công cụ hỗ trợ thực hiện kiểm thử	test execution support tool
công cụ phân tích chương trình động	dynamic program analysis tool
công cụ phân tích chương trình tĩnh	static program analysis tool
công cụ tạo cấu phần các công cụ tạo cấu phần	components-creating tool
công cụ tìm cấu phần	components-searching tool
công cụ vận hành hệ thống	system operation tools
CORBA (Common Object Request Broker Architecture)	CORBA
CTI (Tích hợp điện thoại máy tính)	CTI
D	
đa hình thái	polymorphism
đặc tả yêu cầu	requirement specification
Đặc trưng chất lượng	quality characteristics
đệ qui	recursive
DFD (sơ đồ luồng dữ liệu)	DFD
điều khiển theo biến cố	event-driven
định lý cấu trúc	structure theorem
Đồ thị nhân-quả	cause-effect graph
Đồ thị nhân-quả	cause-result graph
Độ tin cậy	reliability
đoạn	segment
đối tượng	object
đối tượng hợp thành	compound object
đồng dụng	reentrant
dữ liệu	data
dữ liệu phiên bản	version data
Dùng lại được	reusable
Đường cong lỗi	bug curve
đường cong tăng trưởng độ tin cậy	reliability growth curve
đường găng	critical path
E	
ERD (Biểu đồ thực thể quan hệ)	ERD
ERP (lập kế hoạch tài nguyên xí nghiệp)	ERP
EUC (tính toán (của) người dùng cuối))	EUC
EUD (phát triển người dùng cuối)	EUD
F	
FORTTRAN	FORTTRAN
G	
giai đoạn tiến trình	process stage
Giải trình	walkthrough
Giám định	inspection
Giám sát các hiện tượng quan trọng	important phenomena monitoring
Giám sát đặc biệt	specified monitoring

Giám sát xu hướng hỏng hóc	failure tendency monitoring
giao diện con người	human interface
giao diện giữa các mô đun	interfaces between modules
gói phần mềm	software package
gói phần mềm tích hợp	integrated software package
GUI (giao diện người dùng (bằng) đồ hoạ)	GUI
H	
HCP (Sơ đồ mô tả gọn và có thứ bậc)	HCP
hệ con	sub-system
hệ thống tìm kiếm	search systems
Hệ thống tin chiến lược	strategic information system
Hệ xử lý tác nghiệp	operational processing system
hỗ trợ kỹ thuật	technical support
hợp đồng bảo trì	maintenance contract
HTML (ngôn ngữ đánh dấu siêu văn bản)	HTML
I	
ID người dùng	user ID
ISO/IEC 9126	ISO/IEC 9126
J	
Java	Java
K	
Kế hoạch hệ thống hoá	systematization plan
kế hoạch phát triển	development plan
Kế hoạch thực hiện dự án	project implementation plan
Kế hoạch-Thực hiện-Kiểm tra-Hành động (PDCA)	PDCA
Kế thừa	inheritance
Khả năng tái định vị	re-locatable
kho chứa	repository
khu biệt hoá	specialization
kỹ nghệ đảo	reverse engineering
Kỹ nghệ tiến	forward engineering
kí pháp Ba lan	Polish notation
kí pháp viết đầu	prefix notation
kí pháp viết giữa	infix notation
kí pháp viết sau	postfix notation
kiểm điểm mã	code review
kiểm điểm thiết kế	design review
kiểm thử	test
kiểm thử đơn vị	unit tests
Kiểm thử hệ thống	system tests
kiểm thử hộp đen	black box test
kiểm thử hộp trắng	white box test
kiểm thử rà lại (hồi qui)	regression test
Kiểm thử tích hợp	integration tests
Kiểm thử vận hành	operation tests
kiểm toán dữ liệu	data audit
kiểu bản ghi	record type
kiểu cấu trúc	structure type
kiểu Do-While	Do-While type

kiểu dữ liệu
kiểu dữ liệu cơ sở
kiểu dữ liệu trừu tượng
kiểu If-then-else
kiểu kí tự
kiểu logic
kiểu mảng
kiểu Repeat-Until
kiểu số nguyên
kiểu tuần tự
kỹ sư hệ thống

L

LAN (mạng cục bộ)
Lập kế hoạch cơ sở
lập kế hoạch tài nguyên xí nghiệp (ERP)
lập kế hoạch tiến trình
lập lịch
lập lịch công việc
lập trình
lập trình chức năng
lập trình hướng đối tượng
lập trình khác biệt
lập trình logic
lập trình thủ tục
lập trình viên
Len lỗi
lịch
Lịch biểu mức cao nhất
Lịch biểu mức thấp nhất
Lịch biểu mức trung
LIPS
lỗi
lớp
Lớp con
lớp tương đương
Lớp tương đương hợp lệ
Lớp tương đương không hợp lệ
luồng màn hình
Lưu đồ
Lưu đồ chương trình
Lưu đồ hệ thống

M

Mã hoá
mạng ngoại bộ (extranets)
mạng nội bộ intranets
Mật hiệu
mật mã hoá
mô đun
Mô hình bản mẫu
mô hình thác đổ

data type
basic data type
abstract data type
If-then-else type
character type
logical type
array type
Repeat-Until type
integer type
sequential type
system engineer

LAN
basic planning
Enterprise Resource Planning
process planning
scheduling
job scheduling
programming
functional programming
object-oriented programming
difference programming
logic programming
procedural programming
programmer
fault injection
schedule
highest-level schedule
lowest-level schedule
middle-level schedule
LISP
bug
class
sub-class
equivalent class
valid equivalence class
invalid equivalence class
screen flow
flowcharts
program flowchart
system flowchart

coding
extranet
intranet
password
encryption
module
prototype model
waterfall model

mô hình tiến trình
mô hình vận hành nghiệp vụ
mô hình xoắn ốc
mối quan hệ kết tập và phân tách

N

nâng phiên bản
 nạp lại
 Ngăn ngừa việc dùng trái phép
 ngôn ngữ C
 ngôn ngữ cấp cao
 ngôn ngữ cấp thấp
 Ngôn ngữ dạng biểu bảng
 ngôn ngữ hàm
 ngôn ngữ hình thức
 ngôn ngữ hướng đối tượng
 ngôn ngữ hướng vấn đề
 Ngôn ngữ hướng vấn đề chuyên dụng
 ngôn ngữ logic
 ngôn ngữ phi thủ tục
 Ngôn ngữ script
 ngôn ngữ thế hệ 3
 Ngôn ngữ thế hệ thứ tư
 Ngôn ngữ thủ tục
 ngôn ngữ tự nhiên
 Người bảo trì
 người dẫn chương trình
 người dùng
 Người lãnh đạo dự án
 người lập trình chính
 người lập trình dự phòng
 người quản lí dự án
 Người quản trị bảo trì
 Người quản trị cấu hình
 Người quản trị kiểm soát thay đổi
 nhật kí vào/ra
 Nhiệm vụ cải tiến
 Nhiệm vụ sửa chữa
 Nhiệm vụ sửa đổi

O

ô
ô tô mát

P

PAD (Biểu đồ phân tích vấn đề)
Pascal
PDM (Quản lí dữ liệu sản phẩm)
Perl
PERT (kĩ thuật kiểm điểm và đánh giá
chương trình)
Phân hoạch chức năng
phân hoạch mô đun

process model
business operation model
spiral model
aggregation and decomposition relationship

version-up
reload
prevention of illicit use
C language
high-level language
low-level language
tabular form language
functional language
formal language
object-oriented language
problem-oriented language
specific problem oriented language
logical language
non-procedural language
script language
third generation language
fourth generation language
procedural language
natural language
maintainers
moderator
user
project leader
chief programmer
backup programmer
project manager
maintenance administrator
configuration administrator
change control administrator
entry/exit log
improvement task
correction task
modification task

cell
automaton

PAD
Pascal
PDM
Perl
PERT

functional partitioning
module partitioning

phân hoạch tương đương	equivalent partitioning
Phần mềm nhóm	groupware
phần mềm trình bày	presentation software
phân tích cấu trúc	structured analysis
phân tích có cấu trúc (SA)	SA
phân tích cú pháp	syntactic analysis
Phân tích giá trị biên	boundary value analysis
pháp Ba lan ngược	reverse Polish notation
phát triển hướng đối tượng	object-oriented development
Phát triển ứng dụng nhanh	Rapid Application Development
phiên bản mới	new version
phương pháp	method
Phương pháp chữ số kiểm tra	check digit method
phương pháp có cấu trúc	structured methods
phương pháp điểm chức năng (FP)	FP (Function Point) method
Phương pháp dựa trên LOC	LOC-based method
phương pháp dựa trên nhiệm vụ chuẩn	standard task-based method
phương pháp làm mịn từng bước	step-by-step refinement method
Phương pháp tạo ra các cấu phần	methods to create components
Phương pháp thiết kế thực nghiệm	experimental design method
phương pháp tích hợp từng bước	step-by-step integration method
Pin	battery
PL/I	PL/I
PostScript	PostScript
Prolog	Prolog
Q	
QFD (Triển khai chức năng chất lượng)	QFD
quan hệ	relationship
quan hệ bộ phận-của	part_of relation
quan hệ là_một	is_a relation
Quản lí an ninh	security management
quản lí chi phí	cost management
quản lí hiệu năng	performance management
quản lí tài nguyên	resource management
Quản lí tài nguyên dữ liệu	data resource management
Quản lí tài nguyên mạng	network resource management
quản lí tài nguyên phần cứng	hardware resource management
Quản lí tài nguyên phần mềm	software resource management
Quản lí tiến độ	progress management
Quản lí tiện nghi	facility management
quản lí tiến trình	process management
quản lí vấn đề	problem management
quản lý thư viện	library management
quyền truy nhập	access rights
R	
RAD (Phát triển ứng dụng nhanh)	RAD
RDMBS (hệ quản trị dữ liệu quan hệ)	RDBMS
S	
Sách bộ mã	code book
SCM (Quản lí dây chuyền cung cấp)	SCM

SE (kỹ sư hệ thống)	SE
SFA (Tự động hoá hiệu lực bán hàng)	SFA
SGML (Ngôn ngữ đánh dấu tổng quát chuẩn)	SGML
siêu lớp	super class
Sơ đồ có cấu trúc	structured charts
Sơ đồ điều khiển khác II)	YAC II (Yet Another Control chart II
Sơ đồ Gantt	Gantt chart
Sơ đồ luồng dữ liệu	Data Flow Diagram
Sơ đồ NS (sơ đồ Nassi-Shneiderman)	NS chart
sơ đồ tiến trình	process chart
SPD (Biểu đồ lập trình có cấu trúc)	SPD
T	
Tái kỹ nghệ	reengineering
Tài liệu thiết kế chương trình	program design documents
Tài liệu thiết kế mô đun	module design document
Tài liệu thiết kế ngoài	external design documents
tài liệu thiết kế trong	internal design documents
tài liệu vận hành	operation manual
TCO (Tổng chi phí của quyền làm chủ)	TCO
Thành viên	member
thay đổi phòng ngừa	preventive modification
thể hiện, thể nghiệm	instance
thiết bị trợ	pointing device
thiết kế (bộ) mã	code design
thiết kế chương trình	designing programs
thiết kế có cấu trúc	structured design
thiết kế có cấu trúc (SD)	SD
Thiết kế có cấu trúc cho chương trình	structured design of programs
thiết kế dữ liệu logic	logic data design
thiết kế dữ liệu vật lý	designing physical data
thiết kế hướng đối tượng	object-oriented design
thiết kế hướng dữ liệu	data-oriented design
Thiết kế hướng tiến trình	process-oriented design
Thiết kế kiểm thử	test coverage
thiết kế màn hình	designing screen
thiết kế màn hình	screen design
Thiết kế mô đun	module design
thiết kế ngoài	external design
thiết kế tài liệu	designing document
thiết kế tệp	designing files
thiết kế trong	internal design
thiết kế vào ra chi tiết	detailed input-output design
Thời gian móc nối muộn nhất có thể	latest possible linkage time
Thời gian móc nối sớm nhất có thể	earliest possible linkage time
Thông báo	message
thủ thư	librarian
Thư viện cấu phần	component libraries
thư viện lớp	class library
thực thể	entity

thuộc tính	attribute
Tiện nghi điều hoà nhiệt độ	air conditioning facilities
Tiện nghi lưu giữ	storage facilities
Tiện nghi ngăn ngừa tội phạm	crime prevention facilities
Tiện nghi phân phối nguồn điện	power distribution facilities
Tiện nghi phòng ngừa thảm hoạ	disaster prevention facilities
tiến trình	process
tiếp cận hướng dữ liệu (DOA)	DOA
Tiếp cận hướng tiến trình	process-oriented approach
tiếp cận trên xuống	top-down approach
Tính cố kết mô đun	module cohesion
tính cước	charging
Tính gắn nối mô đun	module coupling
tính toán (của) người dùng cuối	end user computing
Tình trạng kho	inventory status
tổ (của) người lập trình chính	chief programmer team
Tổ chức phát triển	development organization
Tổ chức thúc đẩy phát triển	development-promoting organization
Tổ chuyên gia	specialist team
tổ dự án con	sub-project team
tổ phân cấp	hierarchical team
Tổng chi phí của quyền làm chủ	Total Cost of Ownership
tổng quát hoá	generalization
trang công việc	worksheet
Trình biên dịch	compiler
Trình soạn thảo	editor
trình soạn thảo cấu trúc	structure editor
Trình soạn thảo văn bản	text editor
Trình thông dịch	interpreter
Trục trặc dữ liệu	data trouble
Trục trặc phần cứng	hardware trouble
Trục trặc phần mềm	software trouble
U	
UML (ngôn ngữ mô hình hoá thống nhất)	UML
Ước lượng dựa trên dữ liệu quá khứ	estimation based on data in the past
UPS (Nguồn không ngắt)	UPS
ủy quyền	delegation
V	
văn phạm cấu trúc câu	phrase structure grammar
văn phạm sinh	generative grammar
việc đo tính toàn vẹn	integrity measure
việc tồn đọng lại	backlog
Visual Basic	Visual Basic
vòng đời phần mềm	software life cycle
vòng đời phần mềm (SLC)	SLC
W	
WAN (mạng diện rộng)	WAN
WBS (Cấu trúc phân việc).	WBS
X	
Xác định yêu cầu	requirement definition

-
-
-
-

Tra cứu thuật ngữ

3GL 91		Các kiểu đe dọa 176	
4GL 91		các phương pháp có cấu trúc 13	
Ada 68		các qui tắc 185	
ALGOL 68		cách tiếp cận dưới lên 11	
An ninh 176		cách tiếp cận hướng dữ liệu 4	
APL 69		cách tiếp cận hướng thiết kế 31	
AVR 138		CAD 117	
B to B 162		CAE 117	
B to C 163		CAM 117	
Bản đặc tả yêu cầu 13		Cân bằng hiệu năng 156	
Bảng quyết định 30		cấp phát cơ sở 143	
bảng tính 4		cấp phát tỉ lệ 143	
bao bọc 32		CASE 4, 21, 28, 56, 88, 93, 94, 95, 117, 118, 130	
Bao phủ 86		Cấu trúc đa nhánh (CASE) 56	
Bao phủ đa điều kiện 77		Cấu trúc điều khiển 54	
Bao phủ điều kiện 76		Cấu trúc phân việc 9	
bao phủ kiểm thử 86		che giấu thông tin 32	
Bao phủ lệnh 75		Chèn lấp 52	
Bao phủ quyết định 75		chi phí cố định 143	
Bao phủ quyết định/điều kiện 76		Chi phí khởi đầu 142	
Bảo trì 145		chi phí lưu động và chi tiêu 143	
Bảo trì hàng ngày 154		Chi phí vận hành 142	
Bảo trì hoàn thiện 157		Chính phủ điện tử 170	
Bảo trì khẩn cấp 155		Chính sách an ninh 178	
Bảo trì phần cứng 155		chuẩn hoá 19	
Bảo trì phần mềm 156		chuẩn trong e-learning 168	
Bảo trì phòng ngừa 154		Chuẩn trong kinh doanh điện tử 164	
Bảo trì sửa chữa 156		chức năng điều chỉnh tự động 124	
Bảo trì thăm dò 155		chức năng định dạng 121	
Bảo trì theo lịch 154		chức năng soạn thảo và xử lý 121	
Bảo trì thích nghi 157		Chức năng tính lại 123	
bảo trì từ xa 154		chức năng truyền thông 120	
Bảo trì về sau 154		Chứng chỉ số thức 173	
BASIC 66		chương trình đệ qui 52	
Biểu đồ chuyển trạng thái 24		chương trình đồng dụng 52	
Biểu đồ luồng dữ liệu 22		chương trình dừng lại được 52	
Biểu đồ quản lý lỗi 85		chương trình khả tái định vị 52	
Biểu thức chính qui 58		Chương trình trang tính 122	
BNF 59		chuyên biệt hoá 115	
bộ mã 45		chuyển dịch 143	
bộ phận-của quan hệ 36		Chuyển dịch hệ thống 143	
Bộ xử lý ngôn ngữ 91		chuyên gia kỹ thuật 111	
C++ 67		Class 32	
Các chuẩn cho việc đo an toàn hệ thống tin 137		CMS 167	

cơ chế bảo vệ	179	ebXML	164, 174
COBOL	66	e-learning	165
COCOMO	108	ERD	4, 13, 23
CODASYL	66	ERP	116
componentware	21	EUC	5
Con trỏ	53	EUD	95
Công cụ CASE	93	FORTTRAN	65
Công cụ CASE bảo trì	94	G to B	171
Công cụ CASE cho pha hạ (L-CASE)	94	G to C	170
Công cụ CASE cho pha thượng (U-CASE)	94	G to G	171
Công cụ CASE chung	94	Giả mạo địa chỉ IP	176
Công cụ CASE cung cấp các dịch vụ nền phát triển	94	giai đoạn tiến trình	10
Công cụ CASE tích hợp	94	Giải trình	84
Công cụ hỗ trợ kiểm thử	92	Giám định	84
Công cụ hỗ trợ phát triển	88	Giám sát	182
Công cụ hỗ trợ thiết kế trong	89	Giám sát các hiện tượng quan trọng	156
Công cụ hỗ trợ thực hiện kiểm thử	92	Giám sát đặc biệt	156
công cụ phân tích chương trình động	92	Giám sát xu hướng hỏng hóc	156
công cụ phân tích chương trình tĩnh	92	giao diện con người	50
công cụ tạo cấu phần	20	giao diện giữa các mô đun	18
công cụ tìm cấu phần	20	gói phần mềm 4, 114	
Công cụ vận hành hệ thống	143	gói phần mềm tích hợp	119
CORBA	37	GUI	39
CPM	102	HCP	29
CTI	116	hệ con	14
Cướp phiên địa chỉ IP	177	Hệ thống phát hiện xâm nhập	183
Đa hình thái	35	Hệ thống quản lý nội dung học	167
Đặc trưng chất lượng	98	Hệ thống quản lý việc học	166
Đệ qui	52	hệ thống tìm kiếm	20
DFD	13	Hệ thống tin chiến lược	3
điều khiển theo biến cố	37	Hệ xử lý tác nghiệp	3
định lý cấu trúc	54	HIPO	4, 14
Đồ thị nhân-quả	79	Hỗ trợ kỹ thuật	157
Độ tin cậy	156	hợp đồng bảo trì	155
đoạn	52	HTML	70
Doanh nghiệp với chính phủ	163	ISO/IEC 9126	98
doanh nghiệp với doanh nghiệp	162	Java	69
Doanh nghiệp với người tiêu thụ	163	Kế hoạch hệ thống hoá	12
Đối tượng	32	kế hoạch phát triển	13
Đối tượng hợp thành	36	Kế hoạch thực hiện dự án	12
Đồng dụng	52	Kế hoạch-Thực hiện-Kiểm tra-Hành động	98
dữ liệu	32	Kế thừa	34
dữ liệu phiên bản	144	Kết tập nội dung	170
Dùng lại được	52	Khả năng tái định vị	52
Đường cong lỗi	85	kho chứa	93
đường cong tăng trưởng độ tin cậy	85	khu biệt hoá	35
đường găng	103	kỹ nghệ đảo	21
		kỹ nghệ tiến	21
		Kí pháp Balan	59

kí pháp Balan ngược	59		Lỗ hổng pháp lí	179
kí pháp viết đầu	59		lỗi	145
kí pháp viết giữa	59		Lợi ích của chính phủ điện tử	172
kí pháp viết sau	59		Lớp tương đương hợp lệ	78
kĩ sư hệ thống	3, 103		Lớp tương đương không hợp lệ	78
kiểm điểm	82		luồng màn hình	40
kiểm điểm mã	82		Lưu đồ	25
kiểm điểm thiết kế	82		Lưu đồ chương trình	27
Kiểm thử	18		Lưu đồ hệ thống	26
Kiểm thử đơn vị	18		Mã hoá	17
Kiểm thử hệ thống	18		mạng ngoại bộ	4
Kiểm thử hộp đen	78		mạng nội bộ	4
Kiểm thử hộp trắng	75		Mật hiệu	140
kiểm thử rà lại	146		mật mã hoá	141
Kiểm thử tích hợp	18		Mật mã hóa	182
Kiểm thử vận hành	18		mô đun	17
kiểm toán dữ liệu	135		Mô hình bản mẫu	7
Kiểu bản ghi	53		Mô hình thác đổ	6
Kiểu cấu trúc	53		mô hình tiến trình	9
kiểu Do-While	55		mô hình vận hành nghiệp vụ	23
Kiểu dữ liệu	53		Mô hình xoắn ốc	8
Kiểu dữ liệu cơ sở	53		Ngăn ngừa việc dùng trái phép	134
Kiểu dữ liệu trừu tượng	54		ngôn ngữ C	67
kiểu If-then-else	55		Ngôn ngữ cấp cao	62
Kiểu kí tự	53		Ngôn ngữ cấp thấp	61
Kiểu logic	53		<i>Ngôn ngữ dạng biểu bảng</i>	63
Kiểu mảng	53		<i>Ngôn ngữ hàm</i>	63
Kiểu Repeat-Until	55		Ngôn ngữ hình thức	56
Kiểu số nguyên	53		<i>Ngôn ngữ hướng đối tượng</i>	63
kiểu trình tự	54		ngôn ngữ hướng vấn đề	62
kinh doanh điện tử	164		Ngôn ngữ hướng vấn đề chuyên dụng	64
LAN	4		<i>Ngôn ngữ logic</i>	63
Lập kế hoạch cơ sở	12		Ngôn ngữ phi thủ tục	63
Lập kế hoạch tài nguyên xí nghiệp	116		Ngôn ngữ script	64
Lập kế hoạch tiến trình	101		Ngôn ngữ thể hệ thứ tư	63
lập lịch	103		Ngôn ngữ thủ tục	62
lập lịch công việc	143		ngôn ngữ tự nhiên	56
Lập trình hàm	72		Người bảo trì	152
Lập trình hướng đối tượng	74		người dẫn chương trình	84
lập trình khác biệt	35		Người dùng	113
Lập trình logic	73		Người lãnh đạo dự án	112
Lập trình thủ tục	72		người lập trình dự phòng	111
LCMS	167		người quản lí dự án	112
Len lỗi	86		Người quản trị bảo trì	152
lịch	13		Người quản trị cấu hình	153
Lịch biểu mức cao nhất	13		Người quản trị kiểm soát thay đổi	153
Lịch biểu mức thấp nhất	13		nhật kí vào/ra	139
Lịch biểu mức trung	13		Nhiệm vụ cải tiến	151
LISP	68		Nhiệm vụ sửa chữa	151
LMS	166			

Nhiệm vụ sửa đổi 151
 ô 122
 ô tô mát 56
 PAD 28
 Pascal 68
 PDM 120
 Perl 70
 PERT 101
 Phân hoạch 15
 Phân hoạch mô đun 16
 Phân hoạch tương đương 78
 Phân loại chính phủ điện tử 170
 phần mềm nhóm 119
 Phần mềm trình bày 124
 phân tích có cấu trúc 31
 Phân tích cú pháp 56
 Phân tích giá trị biên 79
 Phân tích kí sự trên Web 182
 Phát triển hướng đối tượng 8
 phiên bản mới 148
 phương pháp 32
 Phương pháp chữ số kiểm tra 50
 Phương pháp để tạo ra các cấu phần 19
 Phương pháp dựa trên LOC 106
 Phương pháp dựa trên nhiệm vụ chuẩn 106
 Phương pháp FP 107
 phương pháp làm mịn từng bước 11
 Phương pháp thiết kế thực nghiệm 81
 phương pháp tích hợp từng bước 11
 Pin 138
 PL/I 69
 PostScript 70
 Prolog 68
 QFD 98
 Quan hệ 23
 quan hệ kết tập và phân tách 36
 quan hệ là_một 35
 quan hệ tổng quát hoá và khu biệt hoá 35
 Quản lí an ninh 140
 Quản lí chi phí 142
 Quản lí hiệu năng 141
 Quản lí tài nguyên 133
 Quản lí tài nguyên dữ liệu 134
 Quản lí tài nguyên mạng 135
 Quản lí tài nguyên phần cứng 133
 Quản lí tài nguyên phần mềm 134
 Quản lí thư viện 134
 quản lí tiến độ 100

Quản lí tiện nghi 137
 Quản lí tiến trình 100
 Quản lí vấn đề 135
 quyền truy nhập 141
 RAD 95
 Rapid Application Development 95
 RDBMS 4
 Sách bộ mã 48
 SCM 116
 SCORM 168, 169
 SFA 116
 SGML 70
 siêu lớp 35
 SLC 10
 Sơ đồ có cấu trúc 27
 Sơ đồ Gantt 101
 Sơ đồ NS 28
 sơ đồ tiến trình 26
 SPD 29
 tái kĩ nghệ 21
 tài liệu thiết kế chương trình 17
 Tài liệu thiết kế mô đun 17
 Tài liệu thiết kế ngoài 15
 Tài liệu thiết kế trong 16
 tài liệu vận hành 143
 TCO 142
 Thành viên 113
 Thay đổi phòng ngừa 157
 thể hiện 32
 thiết bị trợ 41
 Thiết kế chương trình 16
 thiết kế có cấu trúc 4, 31
 Thiết kế có cấu trúc cho chương trình 16
 Thiết kế dữ liệu logic 14
 Thiết kế dữ liệu vật lí 15
 Thiết kế hướng đối tượng 32
 Thiết kế hướng dữ liệu 31
 Thiết kế hướng tiến trình 31
 Thiết kế mã 14
 thiết kế màn hình 14
 Thiết kế màn hình 38
 Thiết kế mô đun 37
 Thiết kế ngoài 14
 Thiết kế tài liệu 14
 thiết kế tập 15
 Thiết kế trong 15
 Thiết kế vào-ra chi tiết 16
 Thời gian móc nối muộn nhất có thể 103
 Thời gian móc nối sớm nhất có thể 103

Thông báo	34	
Thư viện câu phần	20	
Thư viện lớp	36	
Thực thể	23	
Thuộc tính	23	
Tiện nghi điều hoà nhiệt độ	138	
Tiện nghi lưu giữ	140	
Tiện nghi ngăn ngừa tội phạm	139	
Tiện nghi phân phối nguồn	138	
Tiện nghi phòng ngừa thảm hoạ	139	
Tiến trình	9	
Tiến trình kinh doanh	163	
tiếp cận hướng dữ liệu	4, 31	
tiếp cận trên xuống	11	
Tính cổ kết mô đun	38	
tính cước	143	
Tính gắn nối mô đun	38	
Tính sẵn sàng cho chính phủ điện tử	175	
tính toán người dùng cuối	5	
Tình trạng kho	156	
Tổ chức khởi xướng phát triển	113	
Tổ chức phát triển	110	
Tổ chuyên gia	111	
tổ dự án con	112	
Tổ người lập trình chính	110	
Tổ phân cấp	111	
tổng quát hoá	35	
Total Cost of Ownership	142	

•

trang công việc	122	
Trình biên dịch	91	
Trình soạn thảo	92	
Trình soạn thảo cấu trúc	92	
Trình soạn thảo văn bản	92	
Trình thông dịch	91	
Trục trặc dữ liệu	136	
Trục trặc phần cứng	136	
Trục trặc phần mềm	137	
Truy nhập trái phép	177	
Từ chối dịch vụ	177	
UML	24	
Ước lượng dựa trên dữ liệu quá khứ	105	
UPS	138	
user ID	140	
Uỷ quyền	36	
Văn phạm cấu trúc câu	57	
văn phạm sinh	56	
việc đo tính toàn vẹn	143	
việc tồn đọng lại	4, 95	
Visual Basic	69	
Vòng đời phần mềm	10	
WAN	4	
WBS	9	
Xác định yêu cầu	13	
XML	70	
Xử lý văn bản	120	
YAC II	30	