Chương 1

Tổng quan về công nghệ phần mềm

Giới thiệu

- Các nước phát triển đều phụ thuộc chủ yếu vào các hệ thống phần mềm.
- Có nhiều hệ thống được kiểm soát bởi phần mềm.
- => Xây dựng và bảo trì hệ thống phần mềm một cách hiệu quả là yêu cầu cần thiết đối với nền kinh tế toàn cầu và của từng quốc gia.

Giới thiệu

- Khái niệm về công nghệ phần mềm được đưa ra lần đầu tiên vào năm 1968 tại hội nghị thảo luận về khủng hoảng phần mềm.
- Công nghệ phần mềm đề cập tới các phương thức và công cụ để xây dựng phần mềm chuyên nghiệp, mang lại lợi nhuận cao.

Giới thiệu (tt1)

Nội dung nghiên cứu của chương 1:

Một số khái niệm cơ bản có liên quan tới phần mềm và công nghệ phần mềm.

Tìm hiểu về những nguyên tắc cơ bản về tính chuyên nghiệp và đúng nguyên tắc đối với kỹ sư phần mềm.

Một số khái niệm cơ bản

Khi tìm hiểu về công nghệ phần mềm, chúng ta thường đặt ra một số câu hỏi sau:

- Phần mềm là gì?
- Công nghệ phần mềm là gì?
- Sự khác biệt giữa công nghệ phần mềm và khoa học máy tính?
- Sự khác biệt giữa công nghệ phần mềm và công nghệ hệ thống?

Một số khái niệm cơ bản

- Quy trình phần mềm là gì?
- Mô hình quy trình phát triển phần mềm là gì?
- Chi phí của công nghệ phần mềm?
- CASE (Computer-Aided Software Engineering) là gì?
- Thế nào là một phần mềm tốt?
- Một số nguyên tắc của kỹ sư phần mềm?

Một số khái niệm cơ bản (tt1)

Phần mềm là gì?

Phần mềm là các chương trình máy tính và những tài liệu liên quan đến nó như: các yêu cầu, mô hình thiết kế, tài liệu hướng dẫn sử dụng…

Các sản phẩm phần mềm được chia thành 2 loại:

- Sản phẩm đại trà (Generic Product)
- Sản phẩm theo đơn đặt hàng (Bespoke Product hoặc Customized Product)

Một số khái niệm cơ bản (tt1)

- Sản phẩm đại trà : được phát triển để bán ra ngoài thị trường, đối tượng người sử dụng tương đối đa dạng và phong phú. Những sản phẩm phần mềm thuộc loại này thường là những phần mềm dành cho máy PC.
- Sản phẩm theo đơn đặt hàng được phát triển cho một khách hàng riêng lẻ theo yêu cầu. Ví dụ: Những hệ thống phần mềm chuyên dụng, hỗ trợ nghiệp vụ cho một doanh nghiệp riêng lẻ …

Một số khái niệm cơ bản (tt2)

- Công nghệ phần mềm là gì?

  - Công nghệ phần mềm là những quy tắc công nghệ (engineering discipline) có liên quan đến tất cả các khía cạnh của quá trình sản xuất phần mềm.

  - Các kỹ sư phần mềm nên tuân theo một phương pháp luận có hệ thống và có tổ chức trong công việc của họ. Đồng thời, họ nên sử dụng các công cụ và kỹ thuật thích hợp với vấn đề cần giải quyết, các ràng buộc và tài nguyên sẵn có.

Một số khái niệm cơ bản (tt3)

- Sự khác biệt giữa công nghệ phần mềm và khoa học máy tính?

  - Khoa học máy tính đề cấp tới lý thuyết và những vấn đề cơ bản; còn công nghệ phần mềm đề cập tới các hoạt động xây dựng và đưa ra một phần mềm hữu ích.

  - Khi sự phát triển của phần mềm trở nên mạnh mẽ thì các lý thuyết của khoa học máy tính vẫn không đủ để đóng vai trò là nền tảng hoàn thiện cho công nghệ phần mềm.

Một số khái niệm cơ bản (tt4)

- Sự khác biệt giữa công nghệ phần mềm và công nghệ hệ thống?

  - Công nghệ hệ thống (hay còn gọi là kỹ nghệ hệ thống) liên quan tới tất cả các khía cạnh của quá trình phát triển hệ thống dựa trên máy tính bao gồm: phần cứng, phần mềm, và công nghệ xử lý.

Một số khái niệm cơ bản (tt4)

- Công nghệ phần mềm chỉ là một phần của công nghệ hệ thống. Kỹ sư hệ thống phải thực hiện việc đặc tả hệ thống, thiết kế kiến trúc hệ thống, tích hợp và triển khai.

Một số khái niệm cơ bản (tt5)

- Quy trình phần mềm là gì?

  - Quy trình phần mềm là một tập hợp các hành động mà mục đích của nó là xây dựng và phát triển phần mềm. Những hành động thường được thực hiện trong các quy trình phần mềm bao gồm:

    Đặc tả: Miêu tả những gì hệ thống phải làm và các ràng buộc trong quá trình xây dựng hệ thống.

    Phát triển: xây dựng hệ thống phần mềm.

Một số khái niệm cơ bản (tt5)

Kiểm thử: kiểm tra xem liệu phần mềm đã thoả mãn yêu cầu của khách hàng.

Mở rộng: điều chỉnh và thay đổi phần mềm tương ứng với sự thay đổi yêu cầu.

Một số khái niệm cơ bản (tt6)

Mô hình quy trình phát triển phần mềm là gì?

Mô hình quy trình phát triển phần mềm là một thể hiện đơn giản của một quy trình phần mềm, và nó được biểu diễn từ một góc độ cụ thể.

Một số ví dụ về mô hình quy trình phát triển phần mềm

- Mô hình luồng công việc (workflow): mô tả một chuỗi các hành động cần phải thực hiện.

Một số khái niệm cơ bản (tt6)

Mô hình luồng dữ liệu (data-flow): mô tả luồng thông tin.

Mô hình Vai trò/Hành động (Role/action): chỉ ra vai trò của những người liên quan trong quy trình phần mềm và nhiệm vụ của từng người.

Ngoài ra, còn có một số mô hình quy trình:

Mô hình thác nước (waterfall)

Mô hình phát triển lặp lại (Iterative development)

Mô hình công nghệ phần mềm dựa thành phần (Component-based software engineering).

Một số khái niệm cơ bản (tt7)

- Các chi phí trong công nghệ phần mềm
  - Để xây dựng một hệ thống phần mềm, chúng ta thường phải đầu tư một khoản ngân sách khá lớn. Theo thống kê cho thấy, chi phí cho việc xây dựng phần mềm chiếm một phần đáng kể ở tất cả các nước phát triển.

Một số khái niệm cơ bản (tt7)

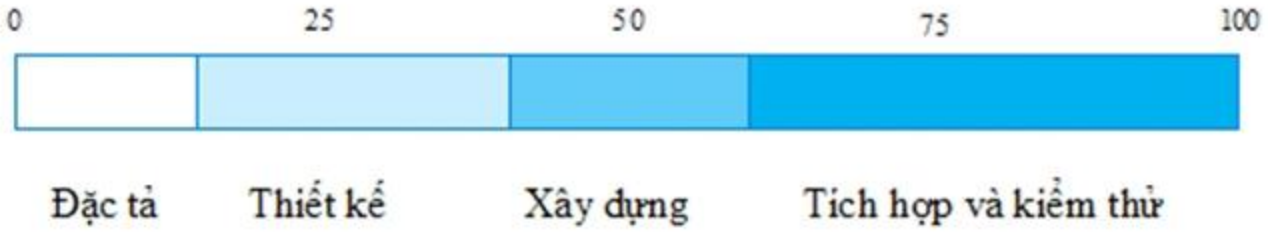
- Chi phí phần mềm thường chiếm phần lớn chi phí của cả hệ thống máy tính.
- Chi phí phần mềm trên máy PC thường lớn hơn chi phí phần cứng.
- Chi phí biến đổi tuỳ thuộc vào từng loại hệ thống được xây dựng và các yêu cầu về đặc điểm của hệ thống như: hiệu năng và độ tin cậy của hệ thống.

Một số khái niệm cơ bản (tt9)

- Việc phân bổ chi phí cũng phụ thuộc vào mô hình phát triển hệ thống được sử dụng. Sau đây là bảng so sánh chi phí của 3 mô hình phổ biến nhất, thường được sử dụng:

  - Mô hình thác nước:

    - Chi phí của các pha đặc tả, thiết kế, cài đặt, tích hợp và kiểm thử được xác định một cách riêng rẽ.



Hình 1.2: Phân bổ chi phí trong mô hình thác nước

Một số khái niệm cơ bản (tt10)

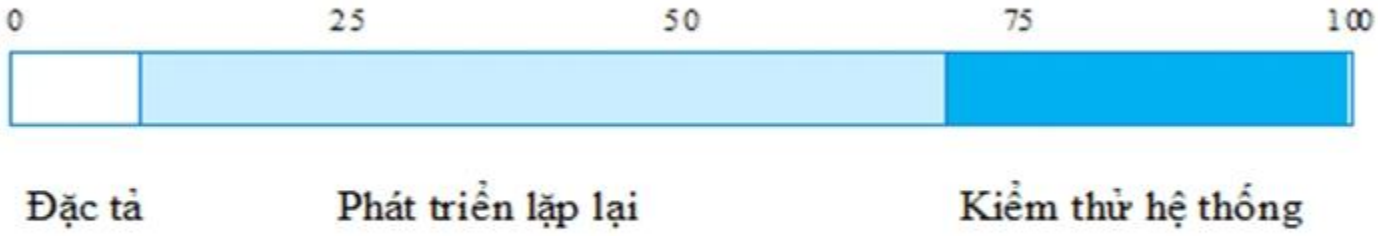
- Mô hình phát triển lặp lại:

-Không thể phân biệt rõ chi phí cho từng pha trong quy trình.

- Chi phí đặc tả giảm.

- Tại mỗi bước lặp, các pha trong quy trình xây dựng hệ thống được thực hiện lại nhằm thực hiện các yêu cầu hệ thống khác nhau ở từng bước lặp.

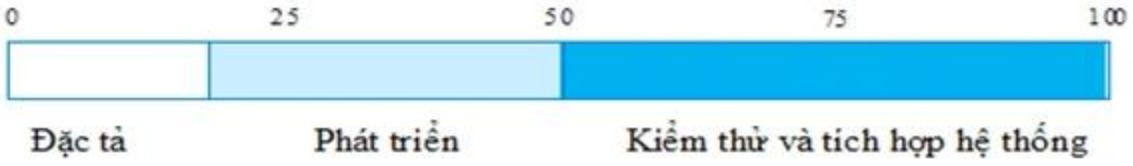
- Sau khi đã thực hiện hết các bước lặp, phải có chi phí kiểm thử toàn bộ hệ thống.



Hình 1.3: Phân bổ chi phí trong mô hình phát triển lặp lại

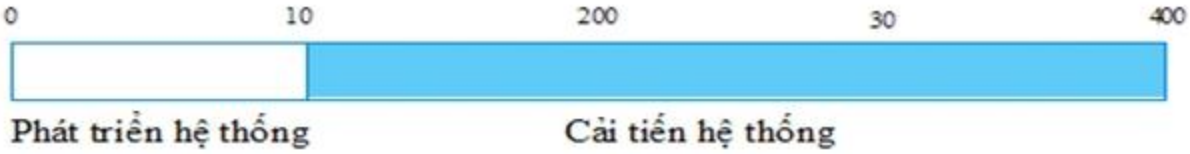
Một số khái niệm cơ bản (tt11)

- Mô hình công nghệ phần mềm hướng thành phần:
  - Chi phí phụ thuộc nhiều vào việc tích hợp và kiểm thử hệ thống.



Hình 1.4: Phân bổ chi phí trong công nghệ phần mềm hướng thành phần

- Ngoài chi phí xây dựng, chúng ta còn phải để một phần lớn chi phí phục vụ cho việc thay đổi phần mềm sau khi nó đã được đưa vào sử dụng. Chi phí cải tiến phần mềm thay đổi phụ thuộc vào từng loại phần mềm.



Hình 1.5: Phân bổ chi phí trên các hệ thống có chu kỳ sống dài

CASE

(Computer-Aided Software Engineering)

- Các hệ thống CASE thường được sử dụng để hỗ trợ các hoạt động trong quy trình xây dựng phần mềm. Có hai loại CASE:
  - **Upper-CASE**: công cụ để hỗ trợ các hoạt động đầu tiên như đặc tả yêu cầu và thiết kế.
  - **Lower-CASE**: công cụ để hỗ trợ các hoạt động sau như lập trình, gỡ lỗi và kiểm thử.

Phần mềm tốt?

- Phần mềm phải đáp ứng các chức năng theo yêu cầu, có hiệu năng tốt, có khả năng bảo trì, đáng tin cậy, và được người sử dụng chấp nhận.

-Khả năng bảo trì: phần mềm phải được điều chỉnh và mở rộng để thoả mãn những yêu cầu thay đổi.

- Mức độ tin cậy: phần mềm phải được tin cậy, bảo mật và chính xác.

Phần mềm tốt?

- Hiệu quả: phần mềm không nên sử dụng lãng phí tài nguyên của hệ thống.

- Khả năng được chấp nhận: người sử dụng phải chấp nhận phần mềm. Điều đó có nghĩa là nó phải dễ hiểu, sử dụng được và tương thích với các hệ thống khác.

Nguyên tắc cần thiết của kỹ sư phần mềm

- Quy trình xây dựng phần mềm được thực hiện trong một môi trường chuyên nghiệp và đòi hỏi tuân thủ các nguyên tắc một cách chính xác. Do đó, những kỹ sư phần mềm phải coi công việc của họ là trách nhiệm to lớn, chứ không đơn thuần chỉ là việc ứng dụng kỹ thuật.

Nguyên tắc cần thiết của kỹ sư phần mềm

- Kỹ sư phần mềm phải ứng xử trung thực và cách làm của họ phải chuyên nghiệp và đúng quy tắc.
- Một số nguyên tắc cần thiết mà một kỹ sư phần mềm phải thực hiện:
  - Sự tin cẩn: kỹ sư phần mềm phải tạo được sự tin cẩn từ phía nhân viên và khách hàng.
  - Năng lực: kỹ sư phần mềm không nên trình bày sai khả năng của mình, không nên nhận những công việc vượt quá khả năng của mình.

Nguyên tắc cần thiết của kỹ sư phần mềm

- Các quyền về tài sản trí tuệ: kỹ sư phần mềm nên quan tâm về các tài sản trí tuệ được bảo hộ như: bằng sáng chế, quyền tác giả … để đảm bảo rằng tất cả tài sản trí tuệ của nhân viên và khách hàng đều được bảo hộ.
- Lạm dụng máy tính: kỹ sư phần mềm không nên sử dụng các kỹ năng của mình để gây ảnh hưởng tới người khác. Lạm dụng máy tính có thể được hiểu là những việc tầm thường (Ví dụ: chơi điện tử trên máy tính của người khác) đến những vấn đề nghiêm trọng (Ví dụ: phát tán virus).

Chương 2

Quy trình xây dựng phần mềm

Nội dung chương 2:

-Một số mô hình phát triển phần mềm thường được ứng dụng và đánh giá ưu và nhược điểm của chúng.

- Xác định chi tiết những công việc phải làm trong quá trình xây dựng một phần mềm và cách thực hiện chúng.

Một số mô hình

Các mô hình phát triển phần mềm phổ biến thường được sử dụng:

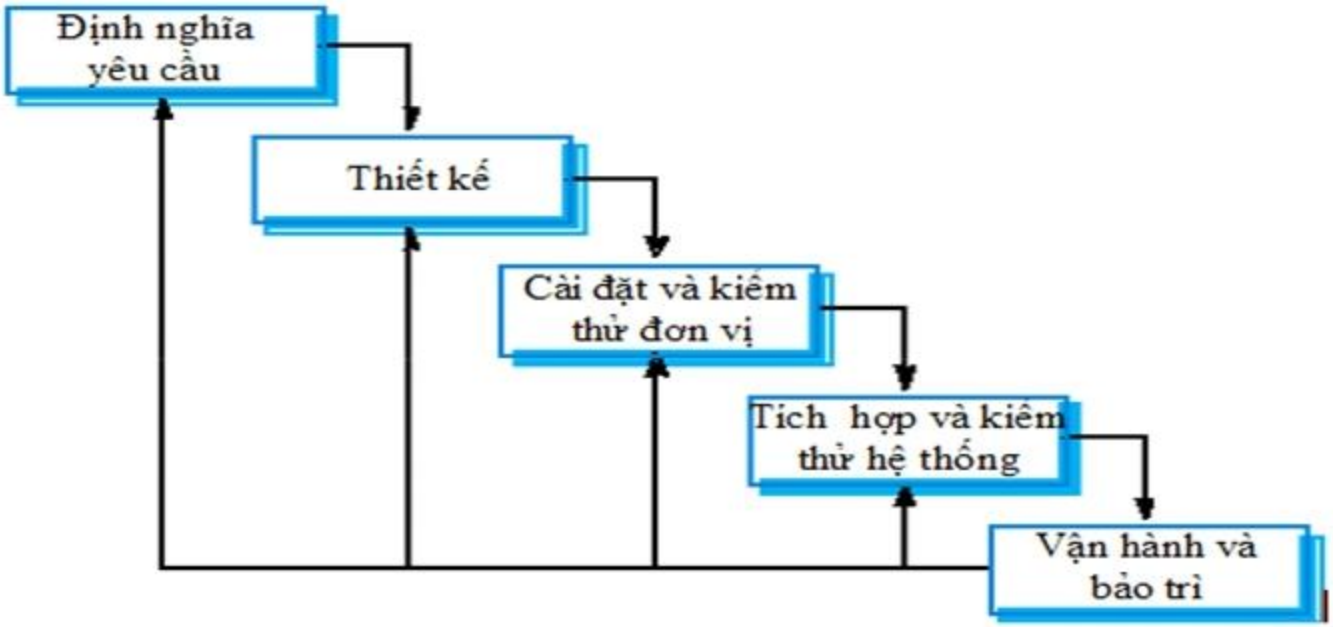
- Mô hình thác nước
- Mô hình xây dựng tiến triển
- Mô hình công nghệ phần mềm dựa thành phần
- Mô hình phát triển lặp lại, tăng thêm
- Mô hình xoắn ốc

Một số mô hình (tt1)

- Mục tiêu:
  - Phải hiểu rõ năm mô hình phát triển phần mềm cơ bản.
  - Phân biệt được sự khác nhau giữa các mô hình; ưu và nhược điểm của từng mô hình.
  - Biết rõ đối với loại hệ thống nào thì nên áp dụng mô hình phát triển nào cho phù hợp.

Một số mô hình (tt2)

- Mô hình thác nước:



Hình 2.1: Mô hình thác nước

Một số mô hình (tt2)

- Trong mô hình thác nước, năm pha trên phải được thực hiện một cách tuần tự; kết thúc pha trước, rồi mới được thực hiện pha tiếp theo.
- Nhược điểm chính của mô hình thác nước là rất khó khăn trong việc thay đổi các pha đã được thực hiện.

Một số mô hình (tt3)

Mô hình này chỉ thích hợp khi các yêu cầu đã được tìm hiểu rõ ràng và những thay đổi sẽ được giới hạn một cách rõ ràng trong suốt quá trình thiết kế.

Một số mô hình (tt4)

Mô hình xây dựng tiến triển:

- Mô hình xây dựng tiến triển dựa trên ý tưởng xây dựng một mẫu thử ban đầu và đưa cho người sử dụng xem xét; sau đó, tinh chỉnh mẫu thử qua nhiều phiên bản cho đến khi thoả mãn yêu cầu của người sử dụng thì dừng lại.

Một số mô hình (tt5)

- Có hai phương pháp để thực hiện mô hình này:
  - Phát triển thăm dò:
    - Mục đích là để làm việc với khách hàng và đưa ra hệ thống cuối cùng từ những đặc tả sơ bộ ban đầu. Phương pháp này thường bắt đầu thực hiện với những yêu cầu được tìm hiểu rõ ràng và sau đó bổ sung những đặc điểm mới được đề xuất bởi khách hàng. Cuối cùng, khi các yêu cầu của người sử dụng được thoả mãn thì cũng là lúc đã xây dựng xong hệ thống.

Một số mô hình (tt5)

Phương pháp loại bỏ mẫu thử:

- Mục đích là để tìm hiểu các yêu cầu của hệ thống. Phương pháp này thường bắt đầu với những yêu cầu không rõ ràng và ít thông tin. Các mẫu thử sẽ được xây dựng và chuyển giao tới cho người sử dụng.

=> Phân loại những yêu cầu nào là thực sự cần thiết và lúc này mẫu thử không còn cần thiết nữa. Như vậy, mẫu thử chỉ có tác dụng để làm sáng tỏ yêu cầu của người sử dụng.

Một số mô hình (tt6)

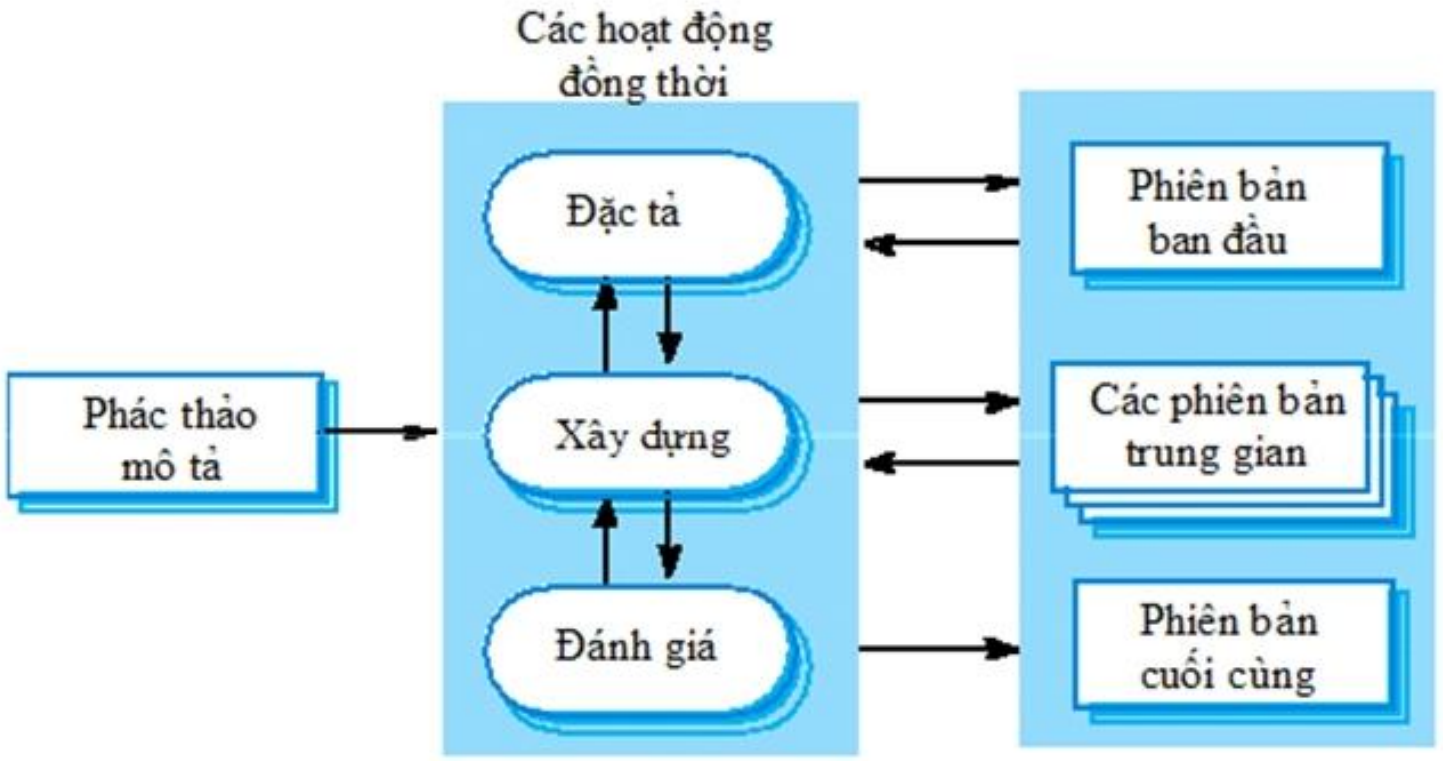
Nhược điểm của mô hình xây dựng tiến triển là:

- Thiếu tầm nhìn của cả quy trình;
- Các hệ thống thường hướng cấu trúc nghèo nàn;

Mô hình xây dựng tiến triển chỉ nên áp dụng với những hệ thống có tương tác ở mức độ nhỏ hoặc vừa; trên một phần của những hệ thống lớn; hoặc những hệ thống có thời gian chu kỳ tồn tại ngắn.

Một số mô hình (tt7)

- Mô hình xây dựng tiến triển (tt3):



Mô hình xây dựng tiến triển

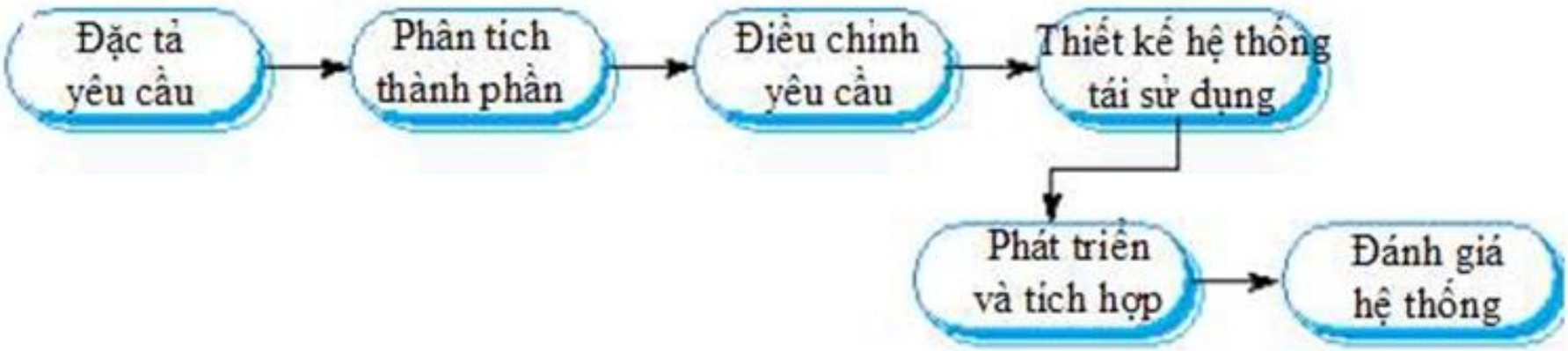
Một số mô hình (tt7)

- **CNPM dựa thành phần:**

  - Mô hình này dựa trên kỹ thuật tái sử dụng một cách có hệ thống; trong đó hệ thống được tích hợp từ nhiều thành phần đang tồn tại hoặc các thành phần thương mại COTS (Commercial-off-the- shelf).

Một số mô hình (tt8)

- CNPM dựa thành phần (tt1):



Công nghệ phần mềm hướng thành phần

Một số mô hình (tt9)

- **Mô hình phát triển lặp lại, tăng thêm:**

  - Mô hình này được đề xuất dựa trên ý tưởng thay vì phải xây dựng và chuyển giao hệ thống một lần thì sẽ được chia thành nhiều vòng, tăng dần. Mỗi vòng là một phần kết quả của một chức năng được yêu cầu.

  - Các yêu cầu của người sử dụng được đánh thứ tự ưu tiên. Yêu cầu nào có thứ tự ưu tiên càng cao thì càng ở trong những vòng phát triển sớm hơn.

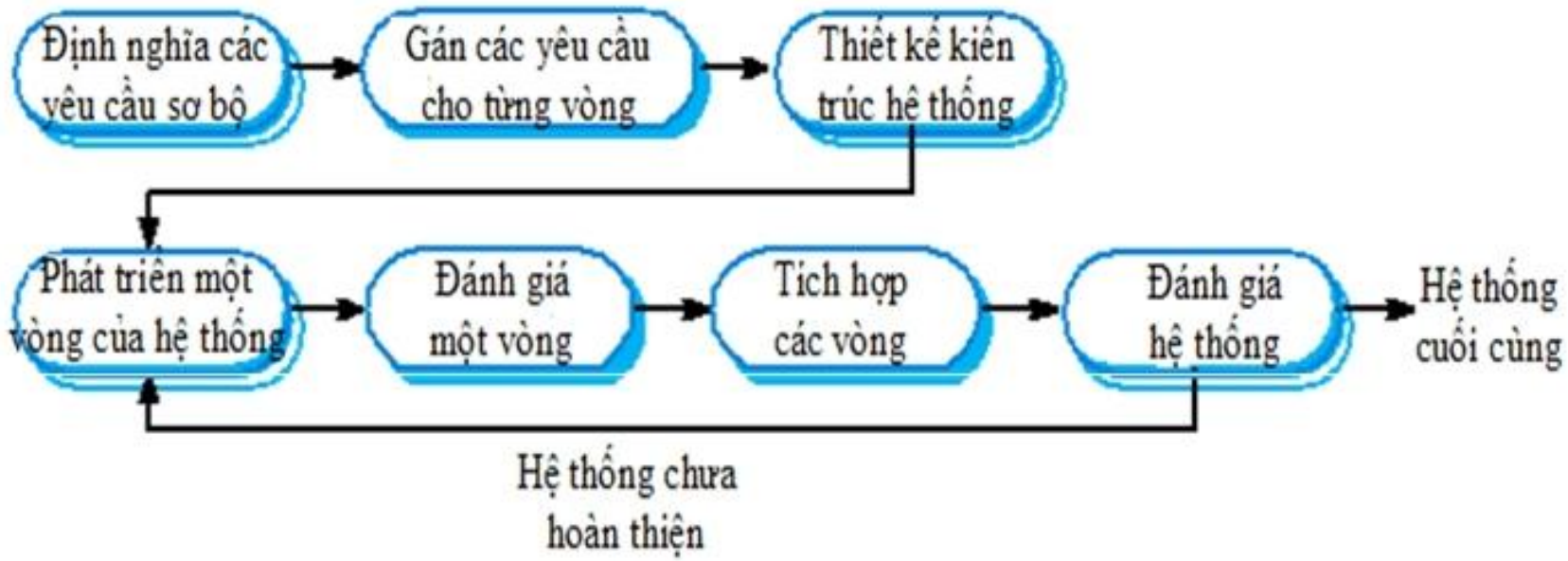
Một số mô hình (tt10)

- Ưu điểm của mô hình phát triển tăng vòng:

- Sau mỗi lần tăng vòng thì có thể chuyển giao kết quả thực hiện được cho khách hành nên các chức năng của hệ thống có thể nhìn thấy sớm hơn.
- Các vòng trước đóng vai trò là mẫu thử để giúp tìm hiểu thêm các yêu cầu ở những vòng tiếp theo.
- Những chức năng của hệ thống có thứ tự ưu tiên càng cao thì sẽ được kiểm thử càng kỹ.

Một số mô hình (tt11)

- Mô hình phát triển lặp lại, tăng thêm (tt2):



Kết quả tăng dần

Một số mô hình (tt12)

Mô hình xoắn ốc:

- Trong mô hình xoắn ốc, quy trình phát triển phần mềm được biểu diễn như một vòng xoắn ốc. Các pha trong quy trình phát triển xoắn ốc bao gồm:

Thiết lập mục tiêu: xác định mục tiêu cho từng pha của dự án.

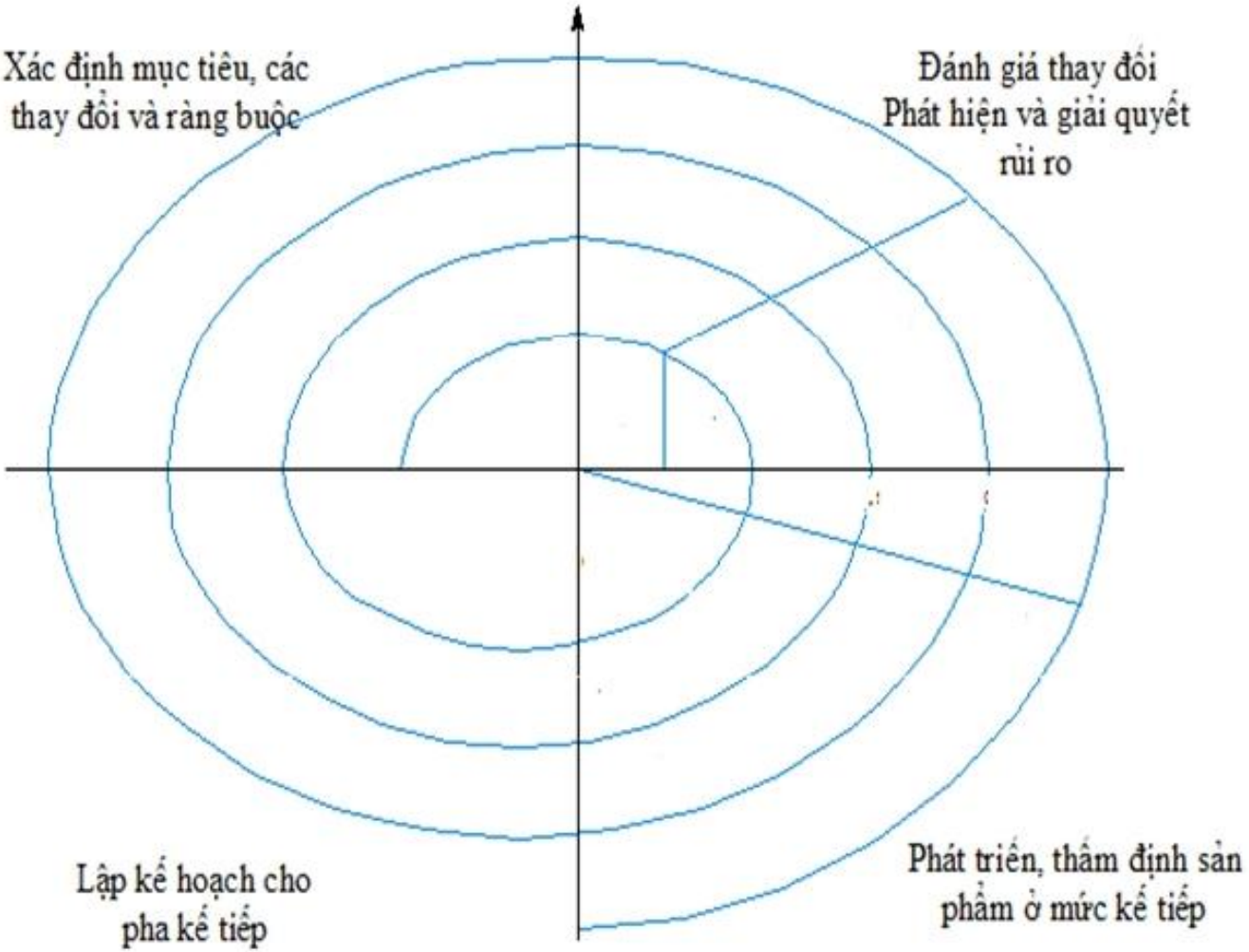
Đánh giá và giảm thiểu rủi ro: rủi ro được đánh giá và thực hiện các hành động để giảm thiểu rủi ro.

Một số mô hình (tt12)

- Phát triển và đánh giá: sau khi đánh giá rủi ro, một mô hình xây dựng hệ thống sẽ được lựa chọn từ những mô hình chung.
- Lập kế hoạch: đánh giá dự án và pha tiếp theo của mô hình xoắn ốc sẽ được lập kế hoạch.

Một số mô hình (tt13)

- Mô hình xoắn ốc (tt1):



Mô hình xoắn ốc

Các hoạt động trong quy trình PM

Trong quy trình phần mềm gồm 4 hoạt động cơ bản sau:

- Đặc tả: các chức năng của hệ thống và những ràng buộc khi vận hành hệ thống cần phải được xác định một cách đầy đủ và chi tiết.
- Thiết kế và cài đặt: phần mềm được xây dựng phải thoả mãn đặc tả của nó.
- Đánh giá: phần mềm phải được đánh giá và thẩm định để đảm bảo rằng nó đã thoả mãn tất cả các yêu cầu.

Các hoạt động trong quy trình PM

- Cải tiến: phần mềm cần phải cải tiến và điều chỉnh để phù hợp với những thay đổi về yêu cầu hệ thống.

Khi xây dựng bất kỳ phần mềm nào, chúng ta đều phải thực hiện bốn công việc trên. Với mỗi mô hình khác nhau thì các hoạt động này cũng được tổ chức theo các cách khác nhau.

Ví dụ, trong mô hình thác nước, chúng được tổ chức một cách tuần tự. Trong mô hình tiến triển, các hoạt động này có thể gối lên nhau.

Các hoạt động (tt)

- Đặc tả phần mềm

  - Đặc tả phần mềm (hay còn gọi là kỹ thuật xác định yêu cầu) là quy trình tìm hiểu và định nghĩa những dịch vụ nào được yêu cầu và các ràng buộc trong quá trình vận hành và xây dựng hệ thống.

Các hoạt động (tt3)

- Quy trình xác định yêu cầu bao gồm bốn pha chính:

Nghiên cứu khả thi: giúp xác định những yêu cầu của người sử dụng có thoả mãn những công nghệ hiện tại hay không.

Về góc độ kinh doanh, nghiên cứu khả thi nhằm xác định hệ thống đưa ra có mang lại lợi nhuận không.

Pha này nên được thực hiện một cách nhanh chóng và không quá tốn kém.

Kết quả của việc nghiên cứu khả thi sẽ xác định có nên tiếp tục xây dựng hệ thống nữa hay không.

Các hoạt động (tt3)

Phân tích và rút ra các yêu cầu: đây là quy trình đưa ra các yêu cầu hệ thống thông qua một số phương pháp như: quan sát hệ thống hiện tại, phỏng vấn và thảo luận với người sử dụng, phân tích nhiệm vụ, phân tích tài liệu hoặc hệ thống cũ … Trong pha này, có thể phải xây dựng một hoặc nhiều mô hình hệ thống và các mẫu thử.

Các hoạt động (tt4)

- *Đặc tả yêu cầu*: Pha này sẽ tư liệu hoá những thông tin thu thập được. Có hai loại yêu cầu cần được xác định:

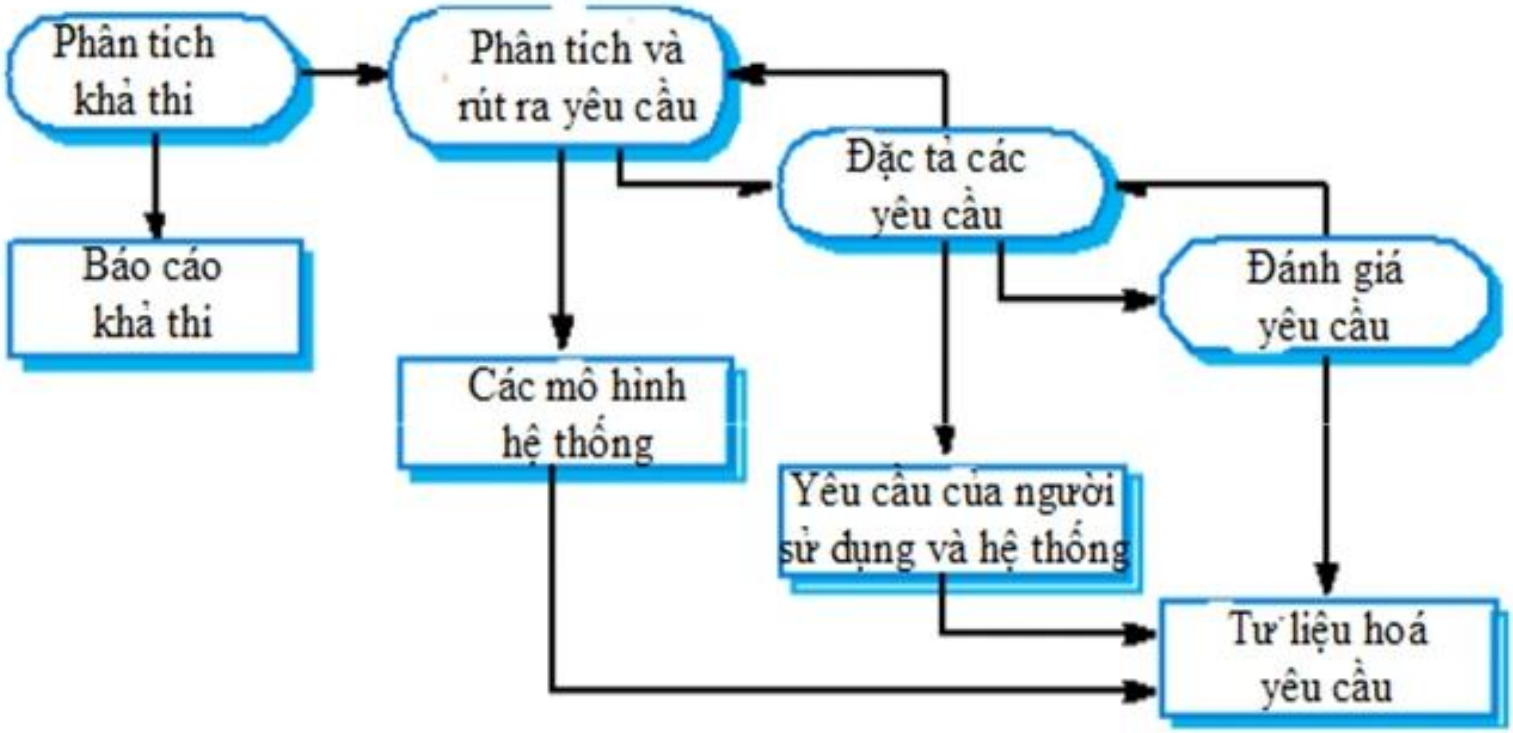
  -Yêu cầu của người sử dụng: là những yêu cầu bằng ngôn ngữ tự nhiên bổ sung thêm cho các biểu đồ của các dịch vụ mà hệ thống cung cấp và các ràng buộc vận hành của nó. Kiểu yêu cầu này được viết bởi người sử dụng.

Các hoạt động (tt4)

-Yêu cầu hệ thống: là những tài liệu có cấu trúc mô tả chi tiết về các chức năng, dịch vụ và các ràng buộc vận hành của hệ thống. Yêu cầu hệ thống sẽ định nghĩa những gì cần phải xây dựng, cho nên nó có thể trở thành bản hợp đồng giữa khách hàng và nhà thầu.

- *Đánh giá yêu cầu:* pha này sẽ kiểm tra lại các yêu cầu xem chúng có đúng thực tế hay không, có thống nhất không, có đầy đủ không. Nếu phát hiện ra lỗi thì phải chỉnh sửa các lỗi này.

Các hoạt động (tt5)



Quy trình xác định yêu cầu

Các hoạt động (tt6)

- Thiết kế phần mềm và cài đặt

-Thiết kế phần mềm là quá trình thiết kế cấu trúc phần mềm dựa trên những tài liệu đặc tả. Hoạt động thiết kế bao gồm những công việc chính sau:

- *Thiết kế kiến trúc:* Các hệ thống con cấu thành nên hệ thống cần xây dựng và mối quan hệ giữa chúng được xác định và tư liệu hoá.

Đặc tả trừu tượng: với mỗi hệ thống con, phải có một bản đặc tả về các dịch vụ của nó và những ràng buộc khi nó vận hành.

Các hoạt động (tt6)

Thiết kế giao diện: với mỗi hệ thống con, các giao diện của nó với những hệ thống con khác phải được thiết kế và tư liệu hoá.

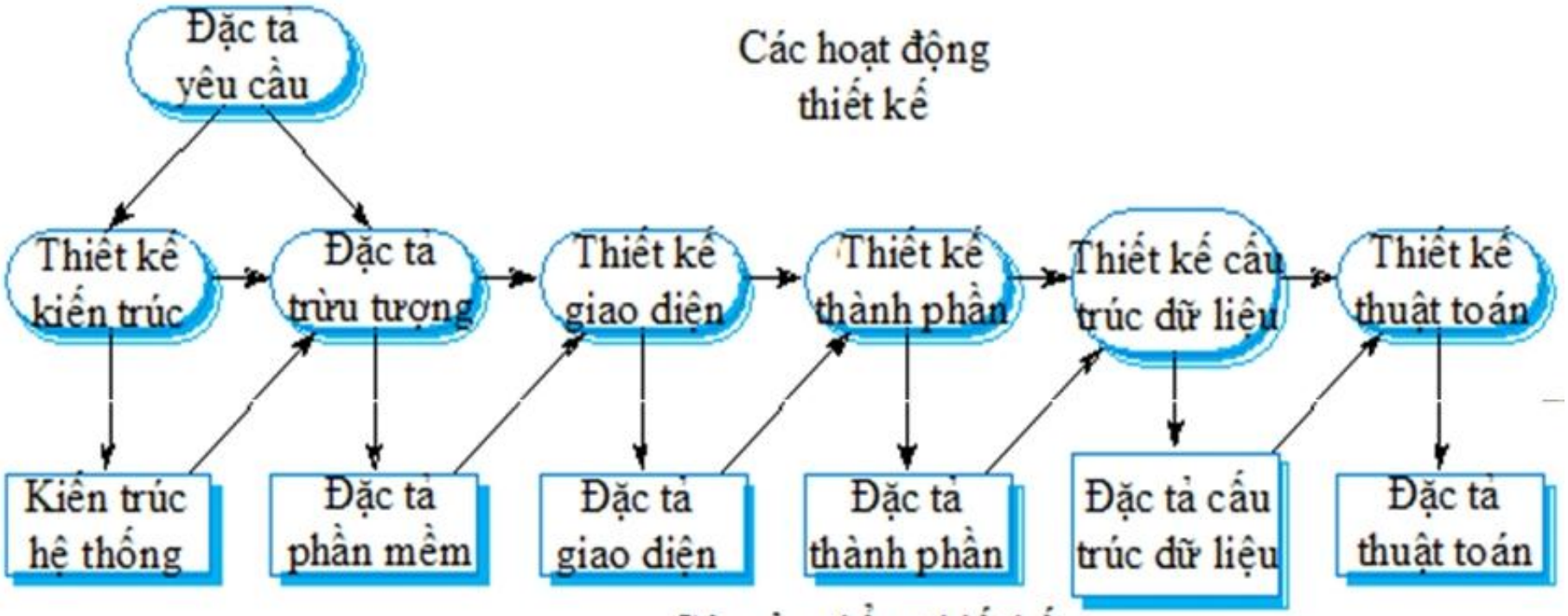
Thiết kế thành phần: các dịch vụ cung cấp cho các thành phần khác và các giao diện tương tác với chúng phải được thiết kế.

Thiết kế cấu trúc dữ liệu: cấu trúc dữ liệu được sử dụng để cài đặt hệ thống phải được thiết kế một cách chi tiết và cụ thể.

Thiết kế thuật toán: Các thuật toán được sử dụng để cung cấp các dịch vụ phải được thiết kế chi tiết và chính xác.

Các hoạt động (tt7)

- Thiết kế phần mềm và cài đặt (tt)



Mô hình chung của quy trình thiết kế

Các hoạt động (tt8)

Cài đặt phần mềm

-Cài đặt là quy trình chuyển đổi từ tài liệu đặc tả hệ thống thành một hệ thống thực, có thể vận hành được và phải loại bỏ các lỗi của chương trình.

- Lập trình là một hành động cá nhân, không có quy trình lập trình chung. Người lập trình phải thực hiện một số kiểm thử để phát hiện ra lỗi trong chương trình và loại bỏ nó trong quy trình gỡ lỗi.

Các hoạt động (tt9)

Đánh giá phần mềm

-Đánh giá phần mềm hay còn gọi là thẩm tra và đánh giá (V&V Verification and validation) được sử dụng để chỉ ra rằng hệ thống đã thực hiện theo đúng các đặc tả và thoả mãn mọi yêu cầu của khách hàng.

-Đánh giá phần mềm bao gồm các công đoạn: kiểm tra, xem xét lại, và kiểm thử hệ thống. Kiểm thử hệ thống tức là cho hệ thống thực hiện trên những trường hợp có dữ liệu thật được lấy từ tài liệu đặc tả hệ thống. Quy trình kiểm thử gồm các pha sau:

Các hoạt động (tt9)

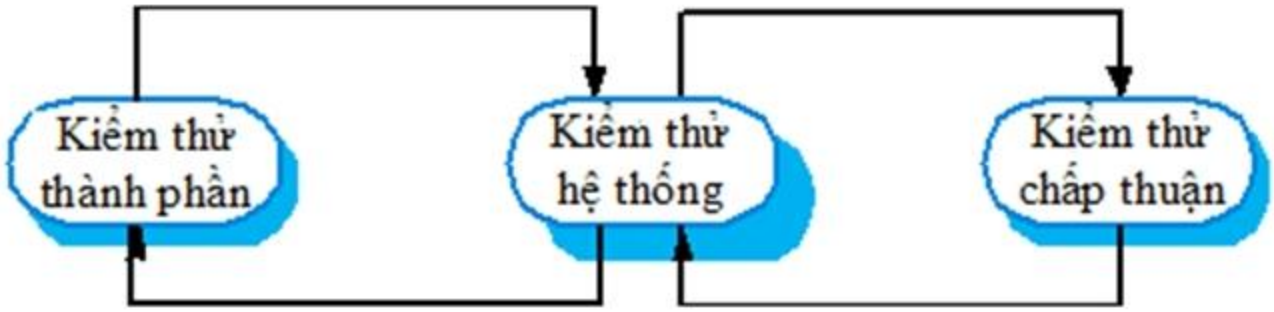
- Kiểm thử thành phần (đơn vị): các thành phần được kiểm thử một cách độc lập, thành phần có thể là một chức năng hoặc một đối tượng hoặc một nhóm các thực thể gắn kết với nhau.

Kiểm thử hệ thống: kiểm thử toàn bộ hệ thống.

Kiểm thử chấp thuận: kiểm thử trên dữ liệu của khách hàng để kiểm tra hệ thống có đáp ứng tất cả các yêu cầu của khách hàng hay không.

Các hoạt động (tt10)

- Khi chuyển giao hệ thống cho khách hàng thì quy trình kiểm thử beta sẽ được thực hiện. Khách hàng sẽ thông báo các lỗi cho đội dự án. Những lỗi này sẽ được chỉnh sửa và tiếp tục kiểm thử beta hoặc chuyển giao thực sự cho khách hàng.

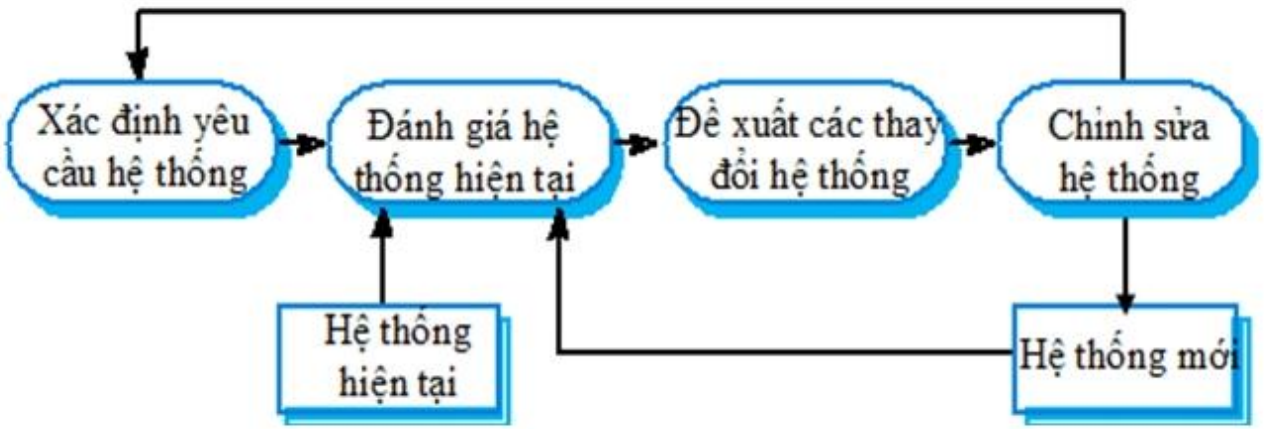


Quy trình kiểm thử

Các hoạt động (tt10)

Cải tiến phần mềm

- Khi các yêu cầu hệ thống thay đổi theo sự thay đổi của các yêu cầu nghiệp vụ thì phần mềm phải cải tiến và thay đổi để hỗ trợ khách hàng. Thông thường chi phí để bảo trì và cải tiến thường đắt hơn nhiều so với chi phí xây dựng phần mềm.



Cải tiến hệ thống

Chương 3

Yêu cầu hệ thống

Yêu cầu hệ thống

- Các yêu cầu của hệ thống phần mềm thường được chia thành ba loại:

  -Yêu cầu chức năng
  - Yêu cầu phi chức năng
  - Yêu cầu miền ứng dụng.

- Tuy nhiên, trong thực tế chúng ta rất khó phân biết ba loại yêu cầu này một cách rõ ràng.

Yêu cầu (tt1)

- Yêu cầu chức năng

  - Yêu cầu chức năng mô tả hệ thống sẽ làm gì. Nó mô tả các chức năng hoặc các dịch vụ của hệ thống một cách chi tiết.

  - Đặc điểm của yêu cầu chức năng:

    - Tính mập mờ, không rõ ràng của các yêu cầu:

      Vấn đề này xảy ra khi các yêu cầu không được xác định một cách cẩn thận. Các yêu cầu mập mờ có thể được người xây dựng và người sử dụng hiểu theo nhiều cách khác nhau.

Yêu cầu (tt1)

- Tính hoàn thiện và nhất quán:

  - Về nguyên tắc, yêu cầu phải chứa tất cả các mô tả chi tiết và không có sự xung đột hoặc đối ngược giữa các yêu cầu. Tuy nhiên, trong thực tế rất khó có thể đạt được điều này.

Yêu cầu (tt3)

- Yêu cầu phi chức năng

  - Yêu cầu phi chức năng không đề cập trực tiếp tới các chức năng cụ thể của hệ thống. Yêu cầu phi chức năng thường định nghĩa các thuộc tính như: độ tin cậy, thời gian đáp ứng, các yêu cầu về lưu trữ …và các ràng buộc của hệ thống như: khả năng của thiết bị vào/ra, giao diện …

Yêu cầu (tt3)

- Một số yêu cầu phi chức năng còn có liên quan đến quy trình xây dựng hệ thống. Ví dụ: các chuẩn được sử dụng, các công cụ CASE Tool, ngôn ngữ lập trình
- Các yêu cầu phi chức năng có thể ít được quan tâm hơn những yêu cầu chức năng. Nhưng nếu nó không được thoả mãn thì hệ thống sẽ không sử dụng được.

Yêu cầu (tt4)

- Yêu cầu phi chức năng (tt1)

  - Các yêu cầu phi chức năng xuất hiện là do yêu cầu của người sử dụng, ràng buộc về ngân sách, các chính sách của tổ chức sử dụng hệ thống, yêu cầu tương thích giữa phần cứng và phần mềm và các tác nhân ngoài khác.

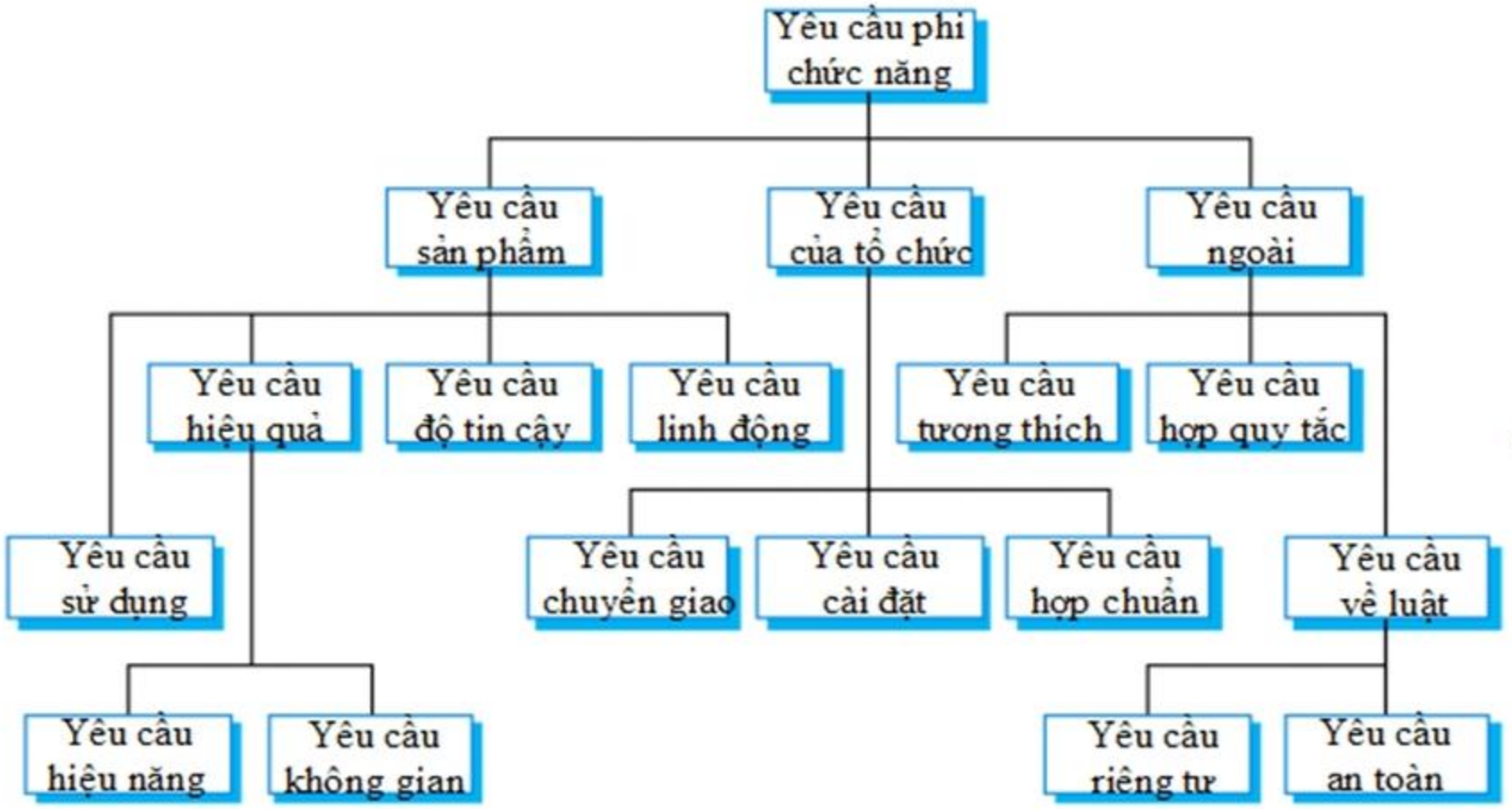
  Phân loại các yêu cầu phi chức năng như sau:

Yêu cầu (tt4)

- Các yêu cầu về sản phẩm: xác định ứng xử của sản phẩm như: hiệu năng, khả năng sử dụng, độ tin cậy … của sản phẩm
- Các yêu cầu về tổ chức: các yêu cầu này được lấy từ những chính sách và quy tắc của khách hàng hoặc tổ chức sử dụng hệ thống.
- Các yêu cầu ngoài: được xác định từ các tác nhân ngoài của hệ thống.

Yêu cầu (tt5)

- Yêu cầu phi chức năng (tt2)



Phân loại các yêu cầu phi chức năng

Yêu cầu (tt6)

- Xác định các yêu cầu phi chức năng của LIBSYS

  Yêu cầu về sản phẩm: LIBSYS phải được cài đặt bằng HTML mà không có frame.

  Yêu cầu về mặt tổ chức: Quy trình xây dựng hệ thống và các tài liệu chuyển giao phải tuân thủ theo qui trình đã thống nhất.

  Yêu cầu ngoài: Hệ thống không được để lộ các thông tin cá nhân của khách hàng.

Yêu cầu (tt7)

- Yêu cầu phi chức năng (tt4)

  - Nói chung, khó xác định chính xác và khó thẩm tra những yêu cầu phi chức năng mập mờ.
  - Do đó, trong tài liệu đặc tả yêu cầu, người ta thường bổ sung các mục tiêu.
  - Với những yêu cầu phi chức năng có thể thẩm định được là những yêu cầu có thể kiểm thử một cách khách quan.

Yêu cầu (tt8)

- Ví dụ các mục tiêu và yêu cầu phi chức năng có thể thẩm định được của hệ thống thư viện
  - Mục tiêu của hệ thống là dễ sử dụng đối với những người sử dụng có kinh nghiệm và được tổ chức để sao cho tối thiểu hoá được lỗi.
  - Các yêu cầu phi chức năng có thể thẩm định được: Những người sử dụng có kinh nghiệm có thể sử dụng được tất cả các chức năng của hệ thống chỉ sau hai tiếng tập huấn. Sau khoá huấn luyện này, số lỗi chương trình gây ra bởi người sử dụng là không quá hai lỗi một ngày.

Yêu cầu (tt9)

- Yêu cầu miền ứng dụng

  - Yêu cầu miền ứng dụng được xác định từ miền ứng dụng của hệ thống và phản ánh các thuộc tính và ràng buộc của miền ứng dụng. Nó có thể là yêu cầu chức năng hoặc phi chức năng.

  - Nếu yêu cầu miền ứng dụng không được thoả mãn thì có thể hệ thống sẽ không làm việc được.

Yêu cầu (tt10)

- Yêu cầu miền ứng dụng (tt1)
  - Khả năng có thể hiểu được: các yêu cầu được biểu diễn dưới ngôn ngữ của lĩnh vực ứng dụng.
  - Các chuyên gia có hiểu biết về lĩnh vực của họ nhưng họ không biết cách xây dựng những yêu cầu miền ứng dụng một cách rõ ràng, mang tính kỹ thuật.

Yêu cầu (tt12)

- Một số kỹ thuật đặc tả yêu cầu hệ thống

  - Nhìn chung, ngôn ngữ tự nhiên thường được sử dụng để viết đặc tả yêu cầu hệ thống cũng như yêu cầu của người sử dụng. Tuy nhiên, yêu cầu hệ thống thường chi tiết hơn yêu cầu của người sử dụng nên đặc tả bằng ngôn ngữ tự nhiên thường gặp một số vấn đề sau:

Yêu cầu (tt12)

Không rõ ràng: Người đọc và người viết yêu cầu phải giải thích các từ theo cùng một nghĩa. Ngôn ngữ tự nhiên có bản chất là mập mờ nên để đạt được yêu cầu trên là rất khó khăn.

Quá mềm dẻo: với cùng một vấn đề nhưng có nhiều cách khác nhau để đặc tả.

Thiếu khả năng mô-đun hoá: cấu trúc của ngôn ngữ tự nhiên không tương xứng với cấu trúc của các yêu cầu hệ thống.

Yêu cầu (tt13)

ặc tả bằng ngôn ngữ tự nhiên thường gây khó hiểu.

ử dụng một số phương pháp được dùng để đặc tả yêu cầu:

- Đặc tả bằng ngôn ngữ hướng cấu trúc

  - Sử dụng ngôn ngữ hướng cấu trúc sẽ yêu cầu người viết đặc tả tuân theo những mẫu được định nghĩa trước. Tất cả các yêu cầu đều được viết theo chuẩn và các thuật ngữ được sử dụng có thể bị hạn chế.

  - Ưu điểm của phương pháp này là đạt được mức độ diễn tả cao nhất của ngôn ngữ tự nhiên nhưng mức độ đồng nhất lại bị lạm dụng trong các đặc tả.

Yêu cầu (tt14)

- Đặc tả dựa biểu mẫu (Form-based)

  - Định nghĩa các chức năng hoặc thực thể, mô tả đầu vào và nơi xuất phát của nó, mô tả đầu ra và nơi nó sẽ đến. Đặc tả dựa biểu mẫu chỉ rõ những thực thể cần thiết, các điều kiện trước và sau (nếu thích hợp), các ảnh hưởng của chức năng.

- Biểu đồ trình tự

  - Biểu đồ trình tự biểu diễn trình tự các sự kiện xảy ra khi người sử dụng tương tác với hệ thống. Nếu ta đọc biểu đồ này từ đầu đến cuối thì ta sẽ thấy được thứ tự của các hành động được thực hiện.

Yêu cầu của người sử dụng

- Yêu cầu của người sử dụng nên mô tả những yêu cầu chức năng và phi chức năng để người sử dụng có thể hiểu được chúng mà không cần phải có những kiến thức về công nghệ một cách chi tiết.
- Yêu cầu của người sử dụng được định nghĩa bằng cách sử dụng ngôn ngữ tự nhiên, bảng hoặc biểu đồ đơn giản.

Yêu cầu của người sử dụng

Tuy nhiên, sẽ gặp phải một số khó khăn khi sử dụng ngôn ngữ tự nhiên:

- Không rõ ràng: Tính chính xác rất khó đạt được nếu tài liệu khó đọc.
- Yêu cầu lộn xộn: các yêu cầu chức năng và phi chức năng không rõ ràng.
- Lẫn lộn giữa các yêu cầu: các yêu cầu khác nhau có thể được diễn tả cùng với nhau.

Yêu cầu của người sử dụng (tt1)

- Để viết yêu cầu của người sử dụng ta nên áp dụng một số quy tắc sau:
  - Đưa ra một định dạng chuẩn và áp dụng nó cho tất cả các yêu cầu.
  - Bắt buộc sử dụng ngôn ngữ một cách thống nhất
  - Đánh dấu những phần quan trọng trong các yêu cầu.
  - Tránh sử dụng những từ ngữ mang tính chuyên môn, kỹ thuật.

Tài liệu đặc tả yêu cầu

- Tài liệu đặc tả yêu cầu là những yêu cầu chính thức về những gì cần phải thực hiện bởi đội phát triển hệ thống.
- Tài liệu đặc tả yêu cầu nên bao gồm cả các định nghĩa về yêu cầu của người sử dụng và đặc tả yêu cầu hệ thống.

Tài liệu đặc tả yêu cầu

- Tài liệu đặc tả yêu cầu không phải là tài liệu thiết kế hệ thống. Nó chỉ thiết lập những gì hệ thống phải làm, chứ không phải mô tả rõ làm như thế nào.

Tài liệu đặc tả yêu cầu (tt1)

- Tài liệu đặc tả yêu cầu dựa theo chuẩn IEEE

1. Giới thiệu
   - 1.1. Mục đích của tài liệu yêu cầu
   - 1.2. Phạm vi của sản phẩm
   - 1.3. Các định nghĩa, từ viết tắt
   - 1.4. Các tham chiếu
   - 1.5. Tổng quan về tài liệu yêu cầu
2. Mô tả chung
   - 2.1. Giới thiệu chung về sản phẩm
   - 2.2. Các chức năng của sản phẩm
   - 2.3. Đặc điểm của người sử dụng
   - 2.4. Các ràng buộc

Tài liệu đặc tả yêu cầu (tt1)

3. Đặc tả yêu cầu: bao gồm các yêu cầu chức năng, phi chức năng, miền ứng dụng và giao diện.

4. Phụ lục

5. Chỉ mục

Chương 4

Quy trình xác định yêu cầu

Giới thiệu

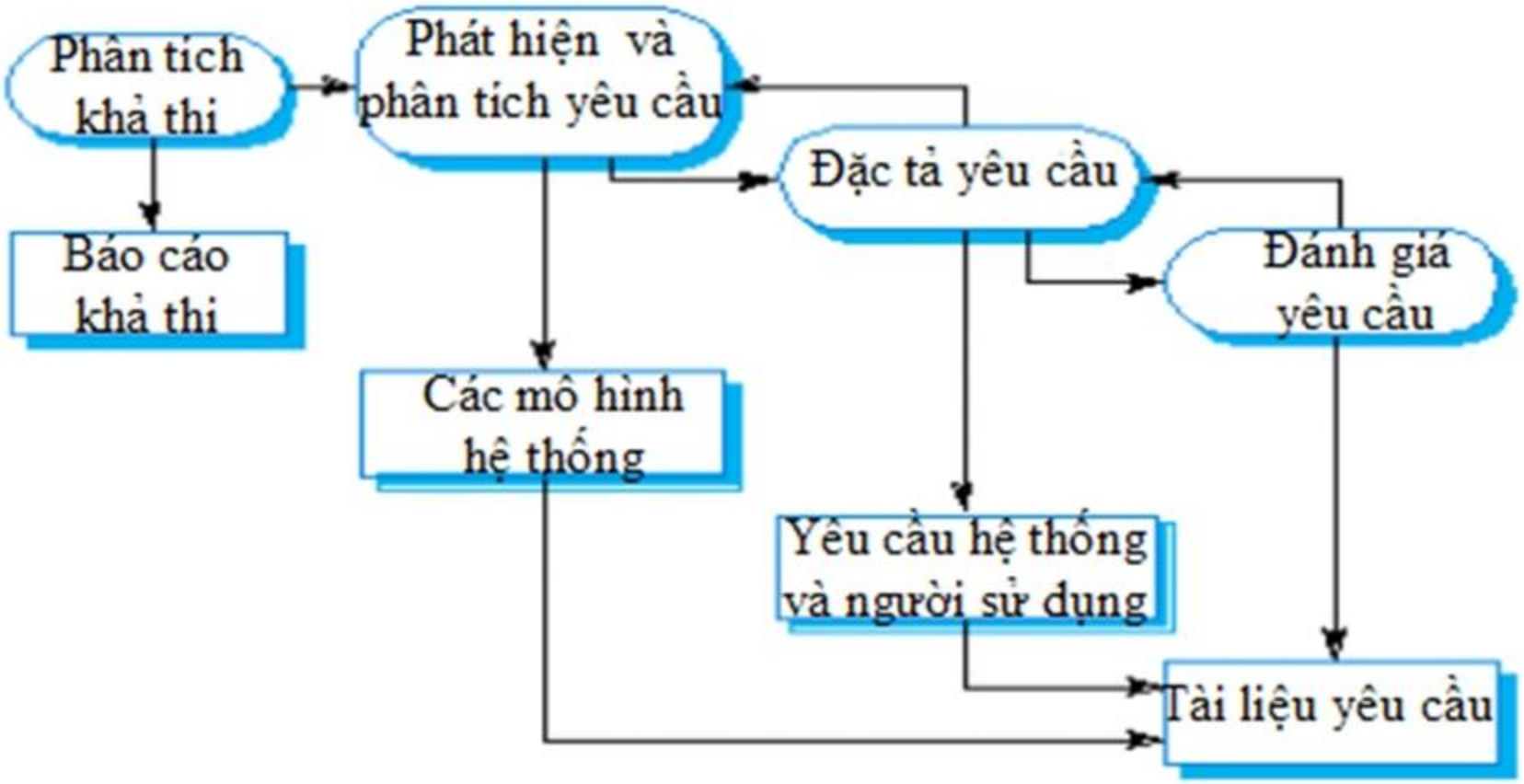
- Mục tiêu của quy trình xác định yêu cầu là đưa ra các tài liệu yêu cầu của hệ thống. Quy trình xác định yêu cầu biến đổi phụ thuộc vào miền ứng dụng, con người và tổ chức xây dựng yêu cầu.

  Tuy nhiên, những quy trình này vẫn có chung một số hoạt động sau: phát hiện yêu cầu, phân tích yêu cầu, đánh giá yêu cầu và quản lý yêu cầu.

Giới thiệu

- Trong thực tế, các yêu cầu luôn luôn thay đổi, thậm chí ngay cả khi đang xây dựng hệ thống. Vì vậy, người ta thường sử dụng mô hình xoắn ốc để xác định các yêu cầu. Mô hình này cho phép việc xác định yêu cầu và cài đặt hệ thống được thực hiện cùng lúc.

Giới thiệu (tt)



Quy trình xác định yêu cầu

Phân tích khả thi

- Đối với tất cả các hệ thống mới, quy trình xác định yêu cầu thường bắt đầu bằng việc phân tích khả thi. Thông tin đầu vào để phân tích khả thi là các yêu cầu nghiệp vụ, mô tả sơ bộ về hệ thống, cách thức hệ thống hỗ trợ các yêu cầu nghiệp vụ. Kết quả của việc phân tích khả thi là một báo cáo để quyết định có nên xây dựng hệ thống đề xuất hay không.

Phân tích khả thi

- Phân tích khả thi thường tập trung vào:

  - Xác định hệ thống có đóng góp vào mục tiêu của tổ chức hay không
  - Kiểm tra xem hệ thống có thể được xây dựng bằng cách sử dụng công nghệ hiện tại và ngân sách cho phép.
  - Kiểm tra xem liệu hệ thống có được tích hợp với các hệ thống khác đang sử dụng hay không.

Phân tích khả thi (tt)

- Thực hiện phân tích khả thi dựa trên việc đánh giá thông tin, lựa chọn thông tin và viết báo cáo.
- Những câu hỏi thường được đặt ra để phân tích khả thi:

  - Nếu hệ thống không được cài đặt thì sao?
  - Vấn đề xử lý hiện tại như thế nào?
  - Hệ thống đề xuất giúp đỡ được gì?
  - Vấn đề về tích hợp là gì?
  - Công nghệ mới cần dùng là gì?
  - Cần có những kỹ năng gì?
  - Những lợi ích mà hệ thống mang lại?

Phát hiện và phân tích yêu cầu

- Trong pha phát hiện và phân tích yêu cầu, nhân viên kỹ thuật và khách hàng cùng hợp tác để xác định miền ứng dụng, các dịch vụ mà hệ thống cung cấp, hiệu năng của hệ thống, các ràng buộc vận hành của hệ thống…

Stakeholder là những người tham dự vào dự án xây dựng hệ thống: người sử dụng cuối, người quản lý, kỹ sư, chuyên gia lĩnh vực, …

Phát hiện và phân tích yêu cầu

- Ví dụ, trong hệ thống ATM gồm các Stakeholder sau: khách hàng của ngân hàng, đại diện của các ngân hàng khác, người quản lý ngân hàng, nhân viên ngân hàng, quản trị CSDL, quản lý bảo mật, phòng marketing, kỹ sư bảo trì phần cứng và phần mềm, người điều hành ngân hàng.

Phát hiện và phân tích (tt)

- Tuy nhiên, việc phát hiện và tìm hiểu yêu cầu của stakeholder, chúng ta thường gặp khó khăn vì những nguyên nhân sau:

  - Stakeholder không biết những gì mà họ thật sự mong muốn.
  - Stakeholder mô tả các yêu cầu theo thuật ngữ của họ.
  - Những stakeholder khác nhau có thể có các yêu cầu xung đột nhau

Phát hiện và phân tích (tt)

- Những yếu tố tổ chức và quyền lực có thể ảnh hưởng tới các yêu cầu hệ thống.

- Các yêu cầu có thể thay đổi trong suốt quá trình phân tích. Những stakeholder mới có thể xuất hiện và môi trường nghiệp vụ có thể thay đổi.

Phát hiện và phân tích (tt)

- Trong quy trình này bao gồm các hoạt động sau:
  - Phát hiện yêu cầu: Phát hiện yêu cầu là quy trình thu thập những thông tin về hệ thống được đề xuất và hệ thống đang tồn tại để xác định các yêu cầu hệ thống và yêu cầu của người sử dụng.
  - Phân loại và sắp xếp yêu cầu: nhóm các yêu cầu có liên quan lẫn nhau và tổ chức chúng thành những nhóm gắn kết với nhau.

Phát hiện và phân tích (tt)

- Sắp thứ tự ưu tiên và điều chỉnh các yêu cầu xung đột: khi có nhiều stakeholder thì các yêu cầu của họ càng có nhiều xung đột. Hoạt động này nhằm đánh thứ tự ưu tiên của các yêu cầu, phát hiện và giải quyết xung đột giữa các yêu cầu.
- Tư liệu hóa yêu cầu: yêu cầu được ghi chép lại để trở thành tài liệu tham khảo cho các bước tiếp theo.

Phát hiện và phân tích (tt)

- Các cách để phát hiện yêu cầu:
  - -Khung nhìn
  - -Phỏng vấn
  - -Kịch bản
  - -Case

Phát hiện và phân tích (tt)

- Khung nhìn (Viewpoint)

  - Khung nhìn là cách xây dựng yêu cầu để trình bày với từng stakeholder khác nhau. Ta có thể phân loại Stakeholder theo nhiều khung nhìn khác nhau.

  - Phân tích dựa trên khung nhìn cho phép phát hiện nhiều khía cạnh khác nhau của một vấn đề và giúp phát hiện ra sự xung đột giữa các yêu cầu.

Phát hiện và phân tích (tt)

- Khung nhìn được chia thành 3 loại chính và mỗi loại sẽ cung cấp các yêu cầu khác nhau.

Khung nhìn tương tác: là những người hoặc hệ thống khác tương tác với hệ thống. Trong hệ thống ATM, khách hàng và CSDL tài khoản là những khung nhìn tương tác

Khung nhìn gián tiếp: là những stakeholder không sử dụng hệ thống trực tiếp nhưng có ảnh hưởng tới hệ thống. Trong hệ thống ATM, nhân viên quản lý và bảo mật là những khung nhìn gián tiếp.

Phát hiện và phân tích (tt)

- Khung nhìn miền ứng dụng: là những đặc điểm và ràng buộc của miền ứng dụng, có ảnh hưởng tới các yêu cầu. Trong hệ thống ATM, các chuẩn để giao tiếp giữa nhiều ngân hàng là một ví dụ.

Phát hiện và phân tích (tt)

- Phỏng vấn

- Phỏng vấn hình thức hoặc phi hình thức là một trong những phần quan trọng nhất của quy trình xác định yêu cầu. Trong quá trình phỏng vấn, những người xác định yêu cầu sẽ đặt ra các câu hỏi cho stakeholder về hệ thống hiện tại họ đang sử dụng và hệ thống sẽ được xây dựng. Và các yêu cầu sẽ được lấy ra từ những câu trả lời của stakeholder.

- Phỏng vấn được chia thành hai loại:

Phát hiện và phân tích (tt)

- Phỏng vấn đóng: tập các câu hỏi đã được định nghĩa trước và có nhiều đáp án để stakeholder lựa chọn trả lời.

  Phỏng vấn mở: tất cả các vấn đề không được xác định trước và stakeholder phải tự giải thích và phát biểu theo quan điểm của mình.

- Trong thực tế, chúng ta thường trộn lẫn phỏng vấn đóng và mở.

Phát hiện và phân tích (tt)

- Một phỏng vấn tốt có nghĩa là sẽ thu thập được tất cả các hiểu biết về công việc phải làm của stakehoder và cách họ tương tác với hệ thống như thế nào.
- Tuy nhiên, khi phỏng vấn những vấn đề có liên quan tới miền ứng dụng hoặc nghiệp vụ của người sử dụng, chúng ta thường gặp khó khăn vì khó hiểu những từ ngữ chuyên ngành

Phát hiện và phân tích (tt)

- Để phỏng vấn thành công, người phỏng vấn nên:

🕐Cởi mở, sẵn sàng lắng nghe stakeholder và không nên có những ý tưởng đã được định hình sẵn về các yêu cầu.

🕐Đưa ra những câu hỏi gợi mở, không nên hỏi những câu như “Anh muốn gì?”

Phát hiện và phân tích (tt)

- Kịch bản
  - Chúng ta thường hiểu một vấn đề thông qua các ví dụ thực tế dễ dàng hơn là thông qua những mô tả trừu tượng về nó.
  - Do đó, chúng ta có thể sử dụng kịch bản để phát hiện ra các yêu cầu hệ thống. Kịch bản là những ví dụ thực tế về cách sử dụng hệ thống. Chúng bao gồm:

Phát hiện và phân tích (tt)

- Mô tả trạng thái khởi động
- Mô tả luồng sự kiện thông thường
- Mô tả những gì có thể đi tới lỗi
- Thông tin về các hành động đồng thời khác
- Mô tả trạng thái khi kịch bản hoàn thành

Phát hiện và phân tích (tt)

- Ca sử dụng
  - Ca sử dụng là kịch bản được xây dựng dựa trên kỹ thuật của UML để xác định các tác nhân trong một tương tác và mô tả chính tương tác đó. Một tập hợp các ca sử dụng sẽ mô tả tất cả các tương tác có thể trong hệ thống.
  - Ngoài ra, chúng ta có thể sử dụng biểu đồ trình tự để bổ sung các thông tin chi tiết cho ca sử dụng bằng cách biểu diễn trình tự các sự kiện được xử lý trong hệ thống.

Đánh giá yêu cầu

- Đánh giá yêu cầu có liên quan đến việc giải thích các yêu cầu đã được định nghĩa trong hệ thống. Vì chi phí cho việc giải quyết các lỗi có liên quan tới yêu cầu sẽ rất cao cho nên việc đánh giá yêu cầu là vô cùng quan trọng.

  Trong quá trình đánh giá yêu cầu, chúng ta phải kiểm tra các yêu cầu ở những khía cạnh sau:

Đánh giá yêu cầu

-Hợp lệ: Hệ thống có cung cấp các chức năng hỗ trợ tốt nhất cho các yêu cầu của người sử dụng hay không?

- Nhất quán: có yêu cầu nào xung đột nhau hay không?

- Hoàn thiện: tất cả các yêu cầu của khách hàng đã được xác định đầy đủ chưa?

- Hiện thực: các yêu cầu có thể được cài đặt với một ngân sách và công nghệ cho trước?

- Xác thực: các yêu cầu có thể được kiểm tra hay không?

Đánh giá yêu cầu (tt1)

- Các kỹ thuật đánh giá yêu cầu sau đây có thể được sử dụng đơn lẻ hoặc hỗn hợp:

  - Xem xét lại các yêu cầu: phân tích các yêu cầu một cách hệ thống.

  - Mẫu thử: Sử dụng các mô hình hệ thống để kiểm tra các yêu cầu

  - Tạo ra các trường hợp kiểm thử

Lập kế hoạch quản lý yêu cầu

- Quản lý yêu cầu là quy trình quản lý sự thay đổi của các yêu cầu trong quá trình phát hiện yêu cầu và phát triển hệ thống.

  Các yêu cầu thường không đầy đủ và không đồng nhất. Đó là do một số nguyên nhân sau:

Lập kế hoạch quản lý yêu cầu

- Những yêu cầu mới xuất hiện trong quy trình khi các yêu cầu nghiệp vụ thay đổi và khi chúng ta có hiểu biết sâu hơn về hệ thống sẽ xây dựng.
- Ở các khung nhìn khác nhau sẽ có các yêu cầu khác nhau và do đó thường xuất hiện các mâu thuẫn.
- Thứ tự ưu tiên từ các khung nhìn khác nhau cũng thay đổi trong suốt quá trình phát triển hệ thống.

Lập kế hoạch quản lý (tt)

- Các yêu cầu lâu dài là những yêu cầu ổn định kế thừa từ những hành động chính của khách hàng. Và nó có thể kế thừa từ nhiều mô hình miền ứng dụng khác.

  Các yêu cầu thay đổi là những yêu cầu dễ bị thay đổi trong quá trình xây dựng hoặc khi hệ thống được đưa vào sử dụng.

Lập kế hoạch quản lý (tt)

- Quy trình lập kế hoạch quản lý yêu cầu:

  - Xác định yêu cầu.

  - Quản lý thay đổi: xác định các hoạt động tiếp theo khi yêu cầu thay đổi.

  - Các chính sách tìm vết: lượng thông tin về mối quan hệ giữa các yêu cầu cần phải được lưu giữ. Thông thường, nó đề cập tới quan hệ giữa tài nguyên và bản thiết kế hệ thống.

Lập kế hoạch quản lý (tt)

- Tìm vết nguồn: là những liên kết từ các yêu cầu tới stakeholder đưa ra những yêu cầu đó.

  Tìm vết yêu cầu: là mối liên hệ giữa các yêu cầu độc lập nhau.

  Tìm vết thiết kế: là những liên kết từ yêu cầu cho tới thiết kế.

Lập kế hoạch quản lý (tt)

- Quy trình lập kế hoạch quản lý yêu cầu (tt1):

- Hỗ trợ CASE tool: sử dụng các công cụ để hỗ trợ quản lý yêu cầu thay đổi. CASE tool thường hỗ trợ những chức năng như:
  - Lưu trữ yêu cầu: các yêu cầu được quản lý một cách bảo mật và được lưu trong kho dữ liệu.
    Quản lý thay đổi: quy trình quản lý thay đổi là quy trình luồng công việc mà các trạng thái có thể được định nghĩa và luồng thông tin giữa các trạng thái là tự động.
  - Quản lý vết tự động tìm kiếm mối liên kết giữa các yêu cầu.

Lập kế hoạch quản lý (tt)

- Nên áp dụng tất cả các khả năng thay đổi có thể cho tất cả các yêu cầu. Các pha chính của hoạt động này bao gồm:

  - Phân tích vấn đề và đặc tả thay đổi: thảo luận về các vấn đề yêu cầu và những thay đổi có thể xảy ra.
  - Phân tích thay đổi và chi phí: đánh giá ảnh hưởng của sự thay đổi trên các yêu cầu khác.
  - Cài đặt thay đổi: Điều chỉnh tài liệu của các yêu cầu và những tài liệu khác để phản ánh sự thay đổi đó.

Chương 5

Các mô hình hệ thống

Giới thiệu

- Các yêu cầu của người sử dụng thường được viết bằng ngôn ngữ tự nhiên để những người không có kiến thức về mặt kỹ thuật có thể hiểu được nó. Tuy nhiên, những yêu cầu hệ thống chi tiết phải được mô hình hoá. Mô hình hoá hệ thống giúp cho người phân tích hiểu rõ các chức năng của hệ thống.
- Ta có thể sử dụng các mô hình khác nhau để biểu diễn hệ thống từ nhiều khía cạnh khác nhau.

Nội dung

- Mô hình ngữ cảnh
- Mô hình ứng xử
- Mô hình dữ liệu
- Mô hình đối tượng

Mô hình ngữ cảnh

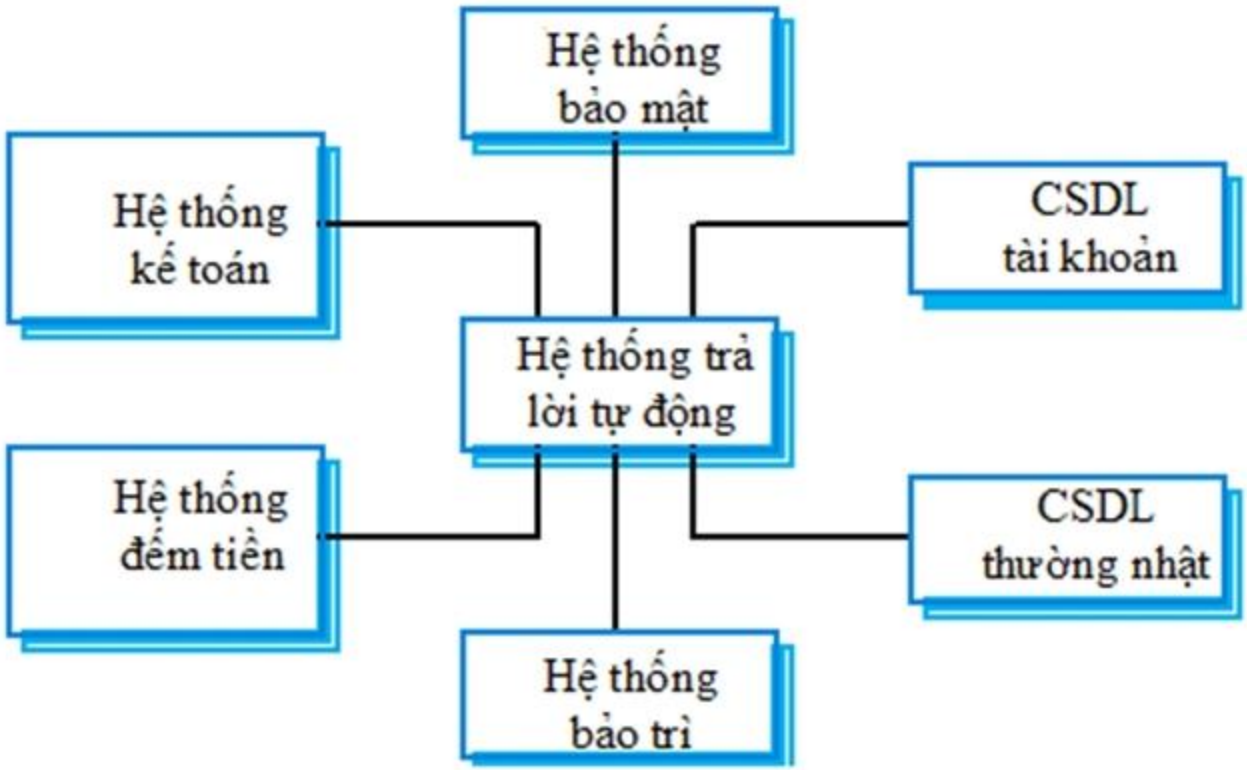
Trong quá trình phát hiện và phân tích yêu cầu, chúng ta nên:

- Xác định phạm vi hệ thống, tức là phân biệt cái gì là hệ thống và cái gì là môi trường của hệ thống.
- Tiếp theo là định nghĩa ngữ cảnh của hệ thống và sự phụ thuộc giữa hệ thống với môi trường của nó.

Thông thường, mô hình kiến trúc đơn giản của hệ thống sẽ được tạo ra trong bước này.

Mô hình ngữ cảnh (tt)

- Ví dụ:
  - mô hình ngữ cảnh của hệ thống ATM



Hình 6.1: Ngữ cảnh của hệ thống ATM

Mô hình ngữ cảnh (tt)

- Mô hình kiến trúc mô tả môi trường của hệ thống, nhưng không chỉ ra quan hệ giữa các hệ thống khác nhau trong một môi trường. Vì vậy, người ta thường sử dụng thêm mô hình tiến trình hoặc mô hình luồng dữ liệu để bổ trợ cho nó.

Mô hình ngữ cảnh (tt)

- Mô hình tiến trình biểu diễn tất cả các tiến trình được hệ thống hỗ trợ.
- Mô hình luồng dữ liệu có thể được sử dụng để biểu diễn các luồng thông tin đi từ tiến trình này tới tiến trình khác.

MÔ HÌNH NGỮ CẢNH

Công dụng

- Xác định phạm vi hệ thống
- Sự phụ thuộc giữa môi trường và hệ thống

Phân loại

- MÔ HÌNH KIẾN TRÚC
- MÔ HÌNH TIẾN TRÌNH/LUỒNG DỮ LIỆU

Mô hình ứng xử

- Mô hình ứng xử được sử dụng để mô tả toàn bộ ứng xử của hệ thống. Có hai kiểu mô hình ứng xử là:
  - Mô hình luồng dữ liệu: biểu diễn cách xử lý dữ liệu trong hệ thống

Mô hình ứng xử

- Mô hình máy trạng thái: biểu diễn cách đáp ứng của hệ thống với các sự kiện xảy ra.

- Hai mô hình này biểu diễn những góc nhìn khác nhau, nhưng cả hai đều cần thiết để mô tả ứng xử của hệ thống.

Mô hình ứng xử (tt1)

- Mô hình luồng dữ liệu

  -Mô hình luồng dữ liệu được sử dụng để mô hình hoá quy trình xử lý dữ liệu của hệ thống. Mô hình này sẽ biểu diễn các bước mà luồng dữ liệu phải trải qua trong hệ thống từ điểm đầu tới điểm cuối.

Mô hình ứng xử (tt1)

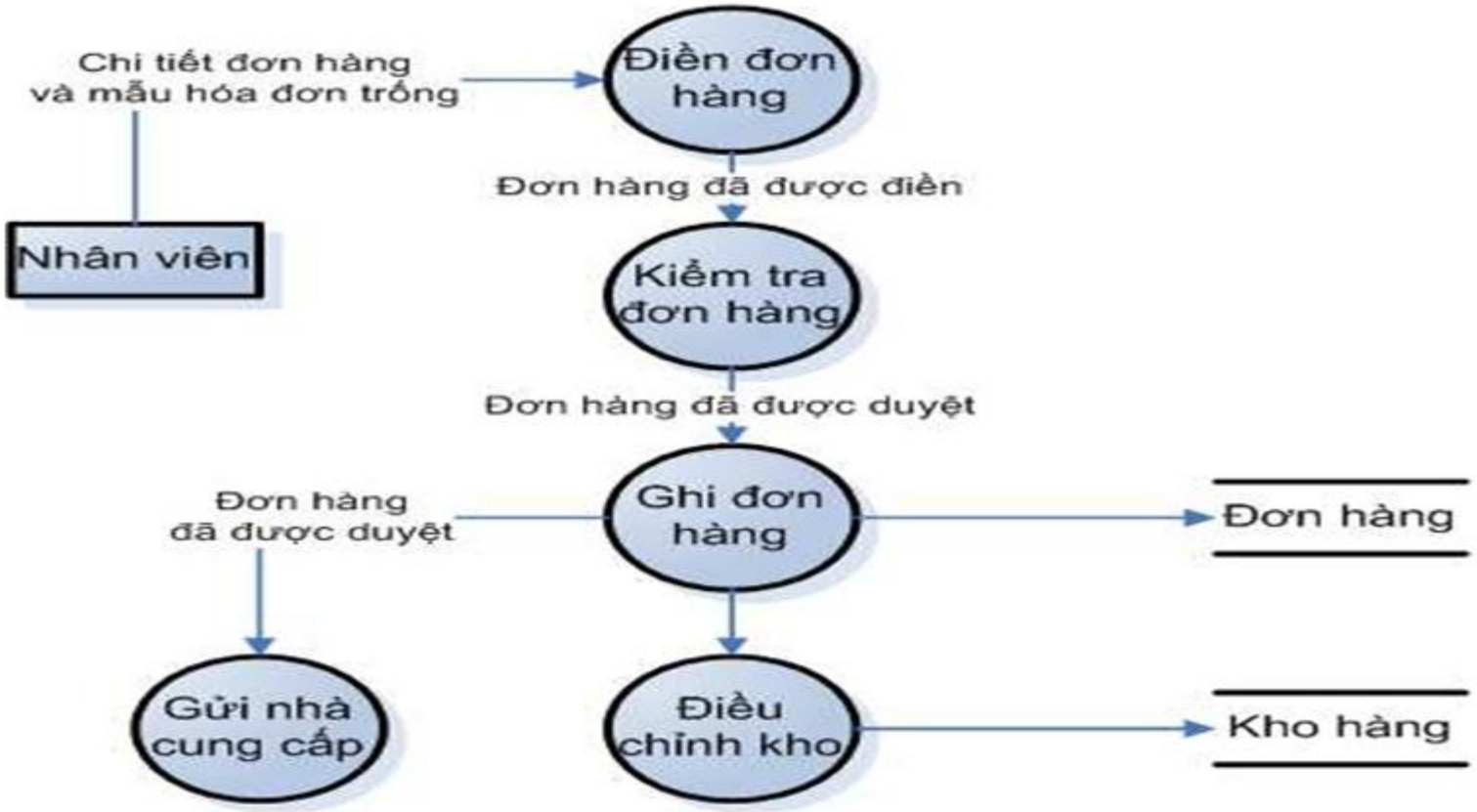
-Mô hình luồng dữ liệu mô hình hoá hệ thống từ góc độ chức năng. Việc tìm vết và tư liệu hoá quan hệ giữa dữ liệu với một quy trình rất có ích đối với việc tìm hiểu toàn bộ hệ thống.

- Mô hình luồng dữ liệu là phần cốt lõi của rất nhiều phương pháp phân tích. Nó chứa các ký pháp rất dễ hiểu đối với khách hàng.

Mô hình ứng xử (tt)

- Mô hình luồng dữ liệu (tt)

  - Ví dụ: Mô hình luồng dữ liệu của chức năng xử lý đơn hàng



Mô hình ứng xử (tt)

- Mô hình máy trạng thái

  - Mô hình máy trạng thái mô tả đáp ứng của hệ thống với các sự kiện bên trong và bên ngoài của nó. Mô hình máy trạng thái biểu diễn các trạng thái của hệ thống và các sự kiện gây ra sự dịch chuyển trạng thái.

  - Mô hình máy trạng thái biểu diễn các trạng thái của hệ thống là các nút và sự kiện là các cung nối giữa các nút đó. Khi có một sự kiện xảy ra, hệ thống sẽ dịch chuyển từ trạng thái này sang trạng thái khác.

Mô hình ứng xử (tt)

- Biểu đồ trạng thái là một biểu đồ trong UML và được sử dụng để biểu diễn mô hình máy trạng thái. Biểu đồ trạng thái cho phép phân tích một mô hình thành nhiều mô hình con và mô tả ngắn gọn về các hành động cần thực hiện tại mỗi trạng thái. Ta có thể vẽ các bảng để mô tả mối quan hệ giữa trạng thái và tác nhân kích hoạt.

MÔ HÌNH ỨNG XỬ

- Công dụng
  - Mô tả toàn bộ ứng xử của hệ thống
- Phân loại
  - MÔ HÌNH LUỒNG DỮ LIỆU
  - MÔ HÌNH MÁY TRẠNG THÁI

Mô hình dữ liệu

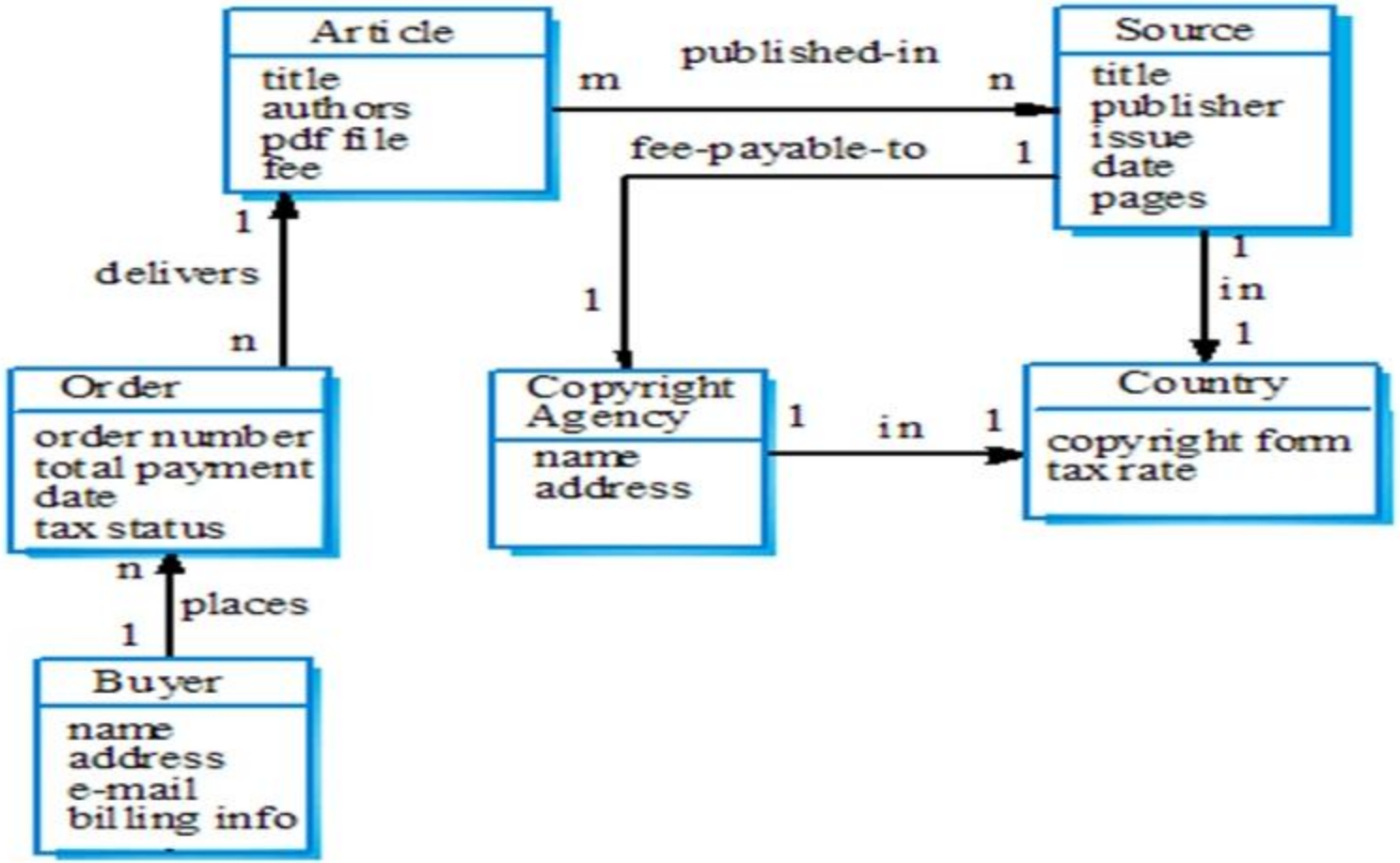
- Giới thiệu:

-Mô hình dữ liệu được sử dụng để mô tả cấu trúc logic của dữ liệu được xử lý bởi hệ thống.

- Thông thường, chúng ta hay sử dụng mô hình thực thể quan hệ thuộc tính (ERA) thiết lập các thực thể của hệ thống, quan hệ giữa các thực thể và thuộc tính của các thực thể. Mô hình này được sử dụng trong thiết kế CSDL và thường được cài đặt trong các CSDL quan hệ.

Mô hình dữ liệu (tt)

- Giới thiệu (tt):
  - Ví dụ mô hình dữ liệu của LIBSYS



Mô hình dữ liệu (tt2)

- Giới thiệu (tt2):

  - Tuy nhiên, mô hình dữ liệu thường không chi tiết. Cho nên, chúng ta có thể sử dụng từ điển dữ liệu làm công cụ bổ trợ. Từ điển dữ liệu là danh sách tất cả các tên gọi được sử dụng trong các mô hình hệ thống. Đó có thể là các thực thể, quan hệ và các thuộc tính …
  - Ưu điểm của từ điển dữ liệu là: hỗ trợ quản lý tên và tránh trùng lặp tên, lưu trữ kiến thức một cách có tổ chức kết nối pha phân tích, thiết kế và cài đặt.

Mô hình dữ liệu (tt3)

- Giới thiệu (tt3):
  - Ví dụ: từ điển dữ liệu của LIBSYS

Tên	Mô tả	Kiểu
Article	Chi tiết về bài báo có trong LIBSYS	Thực thể
Authors	Tên của tác giả viết bài báo	Thuộc tính
Buyer	Tên của người hoặc tổ chức muốn sao chép bài báo	Thực thể
Fee-payable-to	Quan hệ 1:1 giữa Article và Copyright Agency	Quan hệ
Address Buyer	Địa chỉ của người đặt mua được sử dụng để chuyển bài báo tới	Thuộc tính

Mô hình đối tượng

- Giới thiệu:

  - Sử dụng mô hình ứng xử hay mô hình dữ liệu thường rất khó mô tả các vấn đề có liên quan đến thế giới thực. Mô hình đối tượng đã giải quyết được vấn đề này bằng cách kết hợp ứng xử và dữ liệu thành đối tượng.

  - Mô hình đối tượng được sử dụng để biểu diễn cả dữ liệu và quy trình xử lý của hệ thống. Nó mô tả hệ thống dựa theo thuật ngữ các lớp đối tượng và các quan hệ của nó.

Mô hình đối tượng

Một lớp đối tượng là sự trừu tượng hoá trên một tập các đối tượng có thuộc tính và phương thức chung.

- Mô hình đối tượng phản ánh các thực thể trong thế giới thực được vận dụng trong hệ thống. Nếu ta càng có nhiều thực thể trừu tượng thì việc mô hình hoá càng khó khăn.

Mô hình đối tượng (tt)

- Phát hiện các lớp đối tượng là một quy trình rất khó khăn khi tìm hiểu sâu về lĩnh vực của ứng dụng. Các lớp đối tượng thường phản ảnh các thực thể liên quan tới miền ứng dụng của hệ thống.

- Các mô hình đối tượng bao gồm: mô hình thừa kế, mô hình kết hợp và mô hình ứng xử.

Mô hình đối tượng (tt)

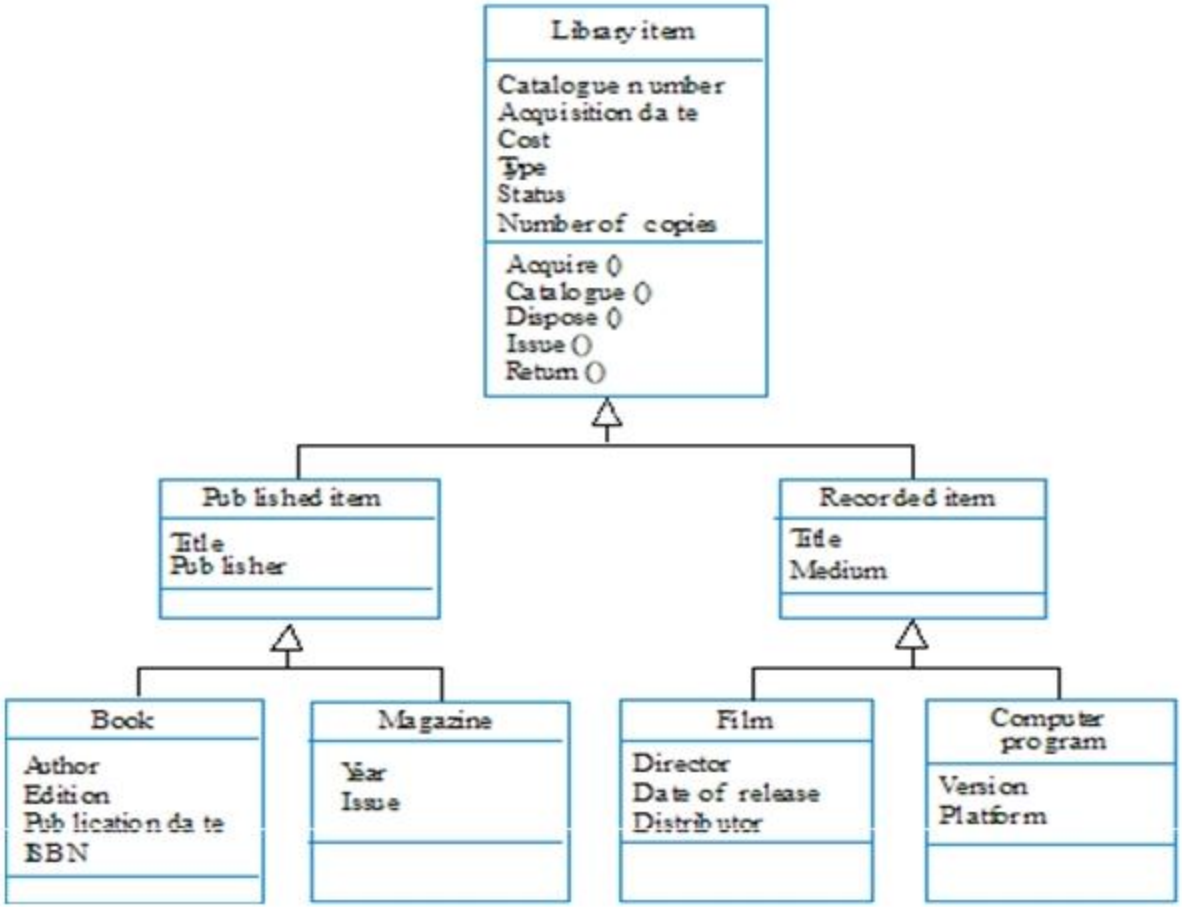
- Mô hình thừa kế

  - Mô hình thừa kế tổ chức các lớp đối tượng theo một cấu trúc phân cấp. Các lớp ở đỉnh của cấu trúc phân cấp phản ánh những đặc trưng chung của tất cả các lớp. Các lớp đối tượng thừa kế những thuộc tính và phương thức của các lớp cha của nó nó có thể bổ sung những đặc điểm của riêng nó.

  - Thiết kế lớp phân cấp là một quy trình khá phức tạp, ta nên loại bỏ sự trùng lặp giữa các nhánh khác nhau.

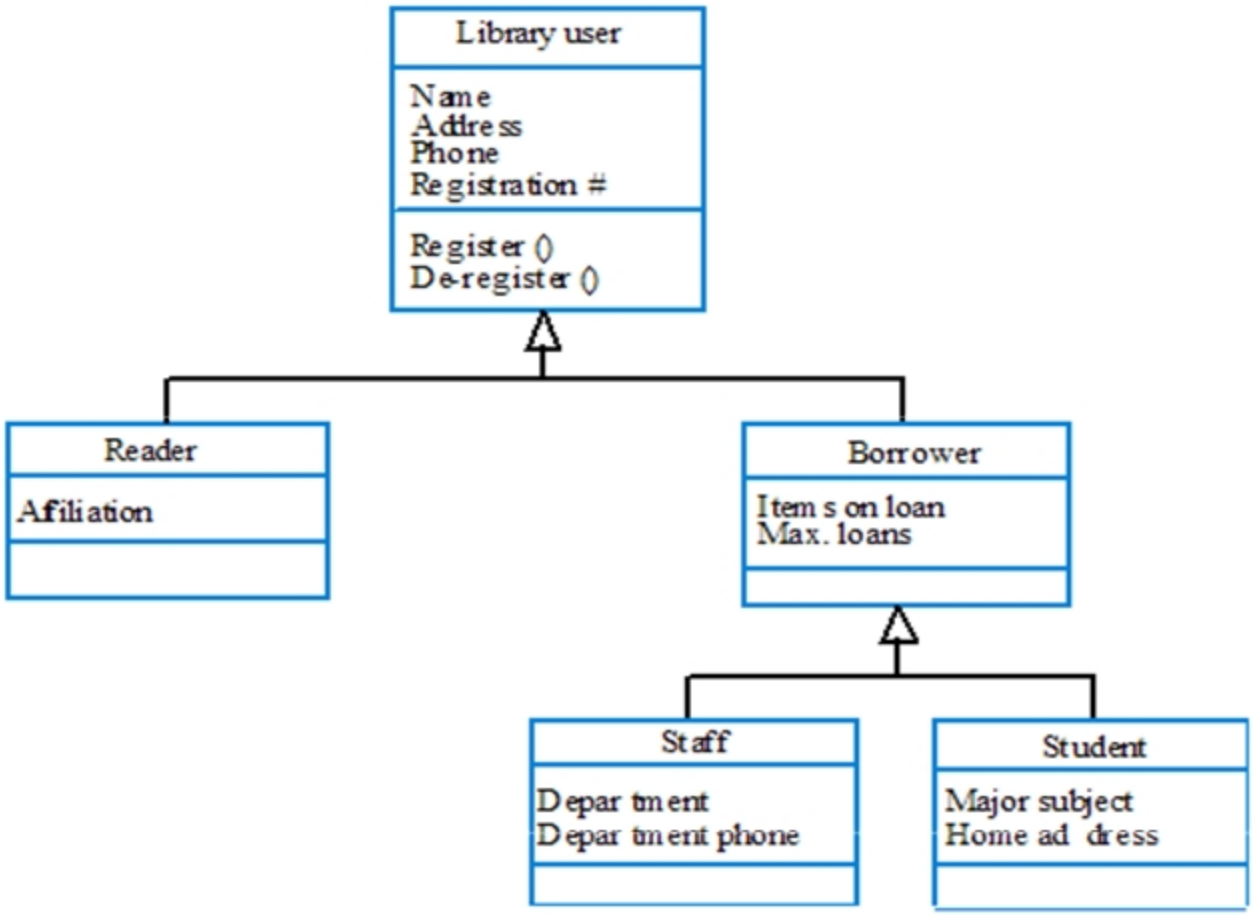
Mô hình đối tượng (tt)

- Mô hình thừa kế (tt1)
  - Ví dụ: cấu trúc phân cấp của lớp Library trong LIBSYS



Mô hình đối tượng (tt)

- Mô hình thừa kế (tt2)
  - Ví dụ: cấu trúc phân cấp của lớp User trong LIBSYS



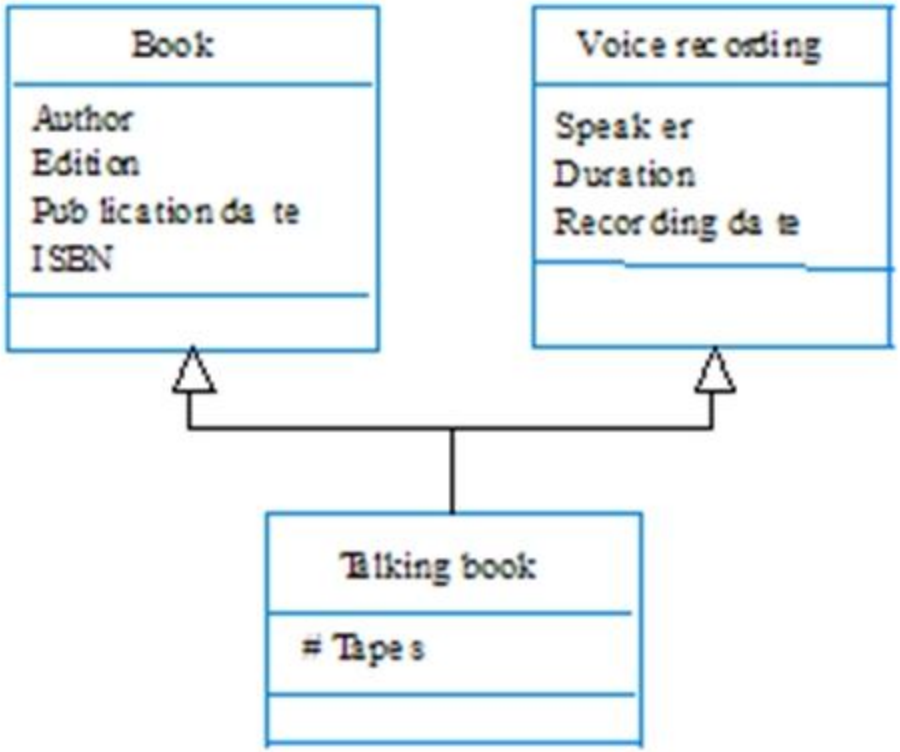
Mô hình đối tượng (tt)

- Mô hình thừa kế (tt3)
  - Cấu trúc đa thừa kế: lớp đối tượng có thể thừa kế từ một hoặc nhiều lớp cha. Tuy nhiên, điều này có thể dẫn tới sự xung đột về ngữ nghĩa khi các thuộc tính/phương thức trùng tên ở các lớp cha khác nhau có ngữ nghĩa khác nhau.

Mô hình đối tượng (tt)

- Mô hình thừa kế (tt4)

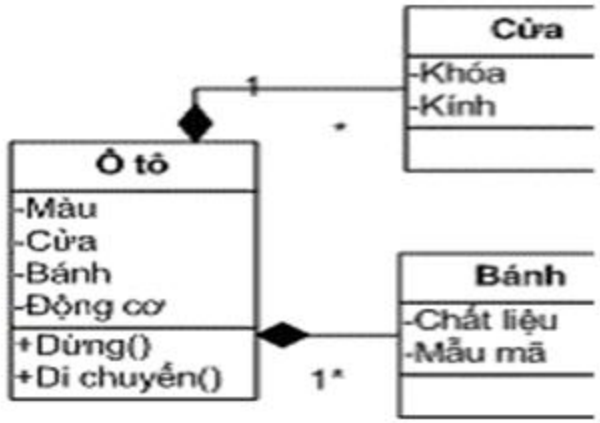
  - Ví dụ: lớp Talking book thừa kế từ hai lớp Book và Voice recording.



Mô hình đối tượng (tt)

- Mô hình kết hợp

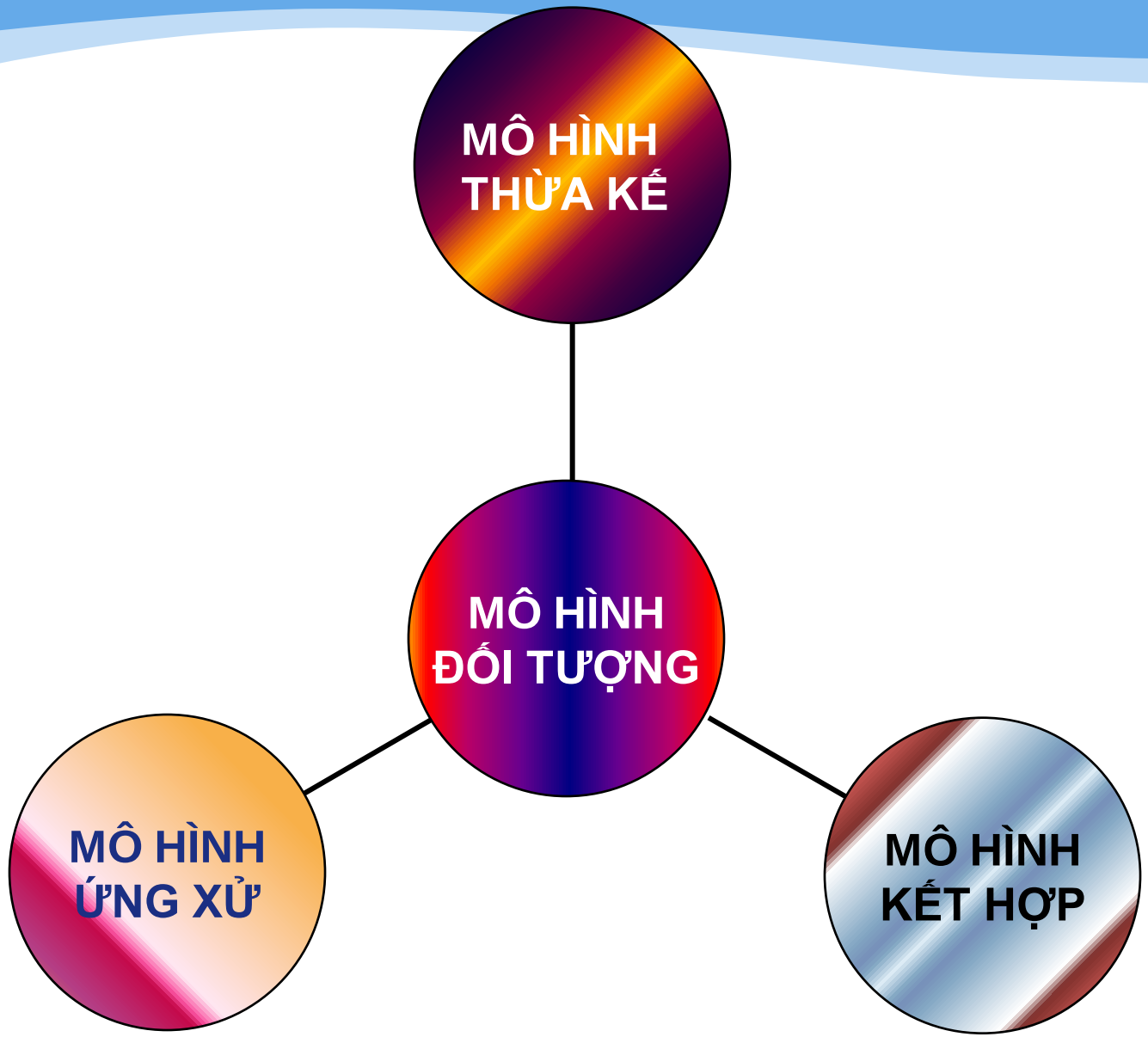
- Mô hình kết hợp biểu diễn cách cấu tạo của một lớp từ các lớp khác. Mô hình kết hợp tương tự như quan hệ hợp thành (part-of).
- Ví dụ: Mô hình kết hợp
- Đối tượng ô tô được tạo thành từ nhiều đối tượng khác như: cửa, bánh xe ...



Mô hình đối tượng (tt)

- Mô hình ứng xử
  - Mô hình ứng xử mô tả tương tác giữa các đối tượng nhằm tạo ra một số ứng xử cụ thể của hệ thống mà đã được xác định như là một ca sử dụng.

- Biểu đồ trình tự hoặc biểu đồ cộng tác trong UML được sử dụng để mô hình hoá tương tác giữa các đối tượng.

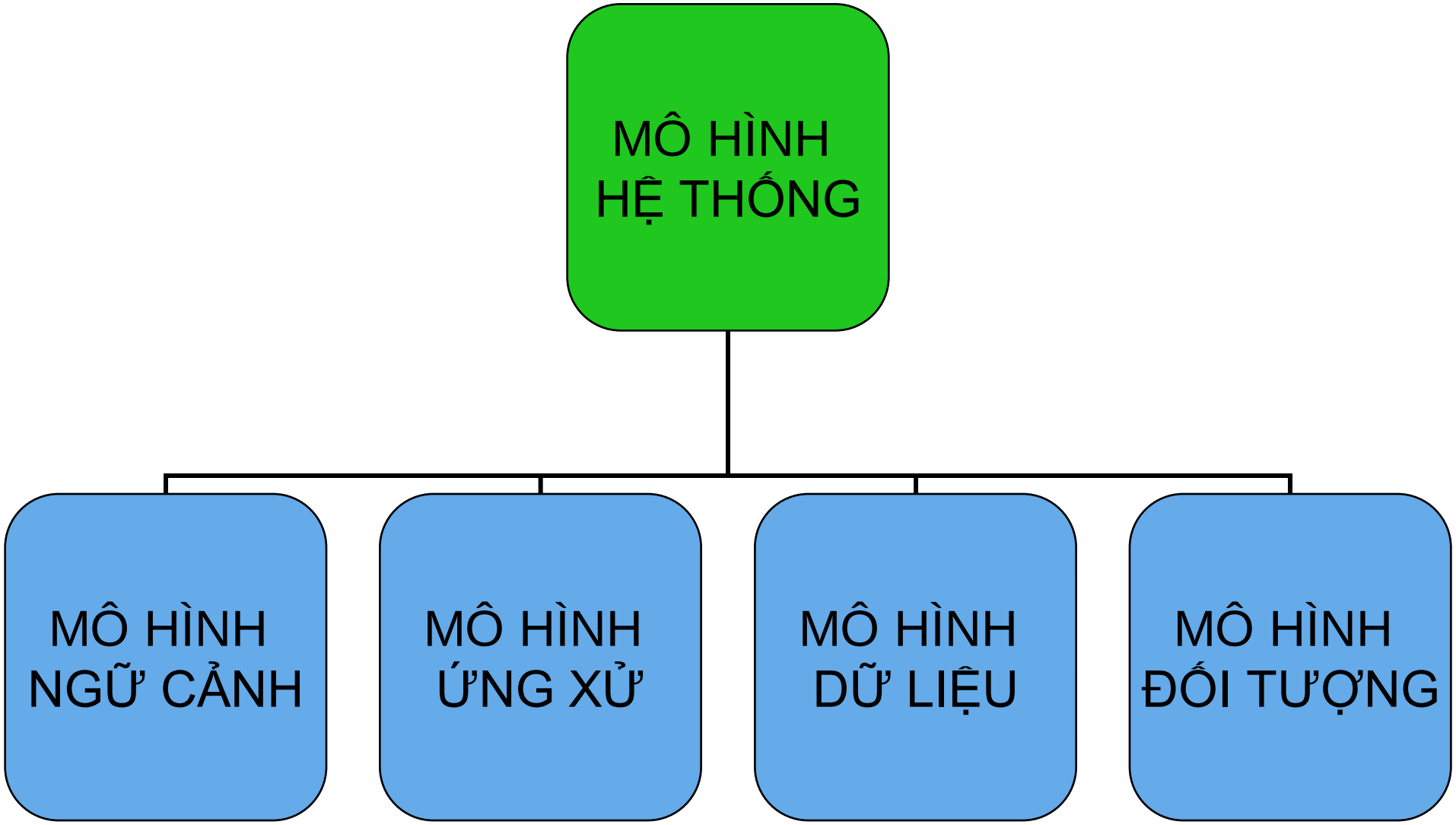

MÔ HÌNH THỪA KẾ
MÔ HÌNH ĐỐI TƯỢNG
MÔ HÌNH ỨNG XỬ
MÔ HÌNH KẾT HỢP

Phương pháp hướng cấu trúc

- Phương pháp hướng cấu trúc rất ít khi được sử dụng do không còn phù hợp với các hệ thống lớn.
- Phương pháp hướng cấu trúc thường có một số nhược điểm sau:
  - Không mô hình hoá được các yêu cầu hệ thống phi chức năng

Phương pháp (tt)

- Không chứa những thông tin để xác định liệu một phương thức có thích hợp với một vấn đề đưa ra hay không.
- Tạo ra quá nhiều tài liệu
- Mô hình hoá hệ thống quá chi tiết và khó hiểu đối với người sử dụng.


MÔ HÌNH HỆ THỐNG
MÔ HÌNH NGỮ CẢNH
MÔ HÌNH ỨNG XỬ
MÔ HÌNH DỮ LIỆU
MÔ HÌNH ĐỐI TƯỢNG

Chương 6

Thiết kế kiến trúc

Giới thiệu

- Pha thiết kế và cài đặt hệ thống được thực hiện sau khi xác định và phân tích yêu cầu hệ thống.
- Thiết kế kiến trúc hệ thống là giai đoạn sớm nhất trong quy trình thiết kế hệ thống.
- Thiết kế kiến trúc cung cấp cho chúng ta bản đặc tả về kiến trúc hệ thống, bao gồm những hệ thống con nào, tương tác với nhau ra sao, framework hỗ trợ điều khiển tương tác giữa các hệ thống con như thế nào …

Thiết kế kiến trúc là gì?

Quy trình thiết kế kiến trúc nhằm xác định các hệ thống con cấu tạo nên hệ thống đề xuất, framework giúp điều khiển các hệ thống con và giao tiếp giữa chúng.

Kết quả của quy trình thiết kế này là bản đặc tả về kiến trúc phần mềm.

Thiết kế kiến trúc là gì? (tt)

- Nếu bản thiết kế kiến trúc rõ ràng sẽ có các ưu điểm trong những hoạt động sau:

  - Giao tiếp giữa các stakeholder: kiến trúc hệ thống thường được sử dụng làm tâm điểm của các buổi thảo luận giữa các stakeholder.

  - Phân tích hệ thống: phân tích để xác định liệu hệ thống có thoả mãn các yêu cầu phi chức năng của nó hay không.

  - Tái sử dụng với quy mô lớn: kiến trúc có thể được tái sử dụng trong nhiều hệ thống.

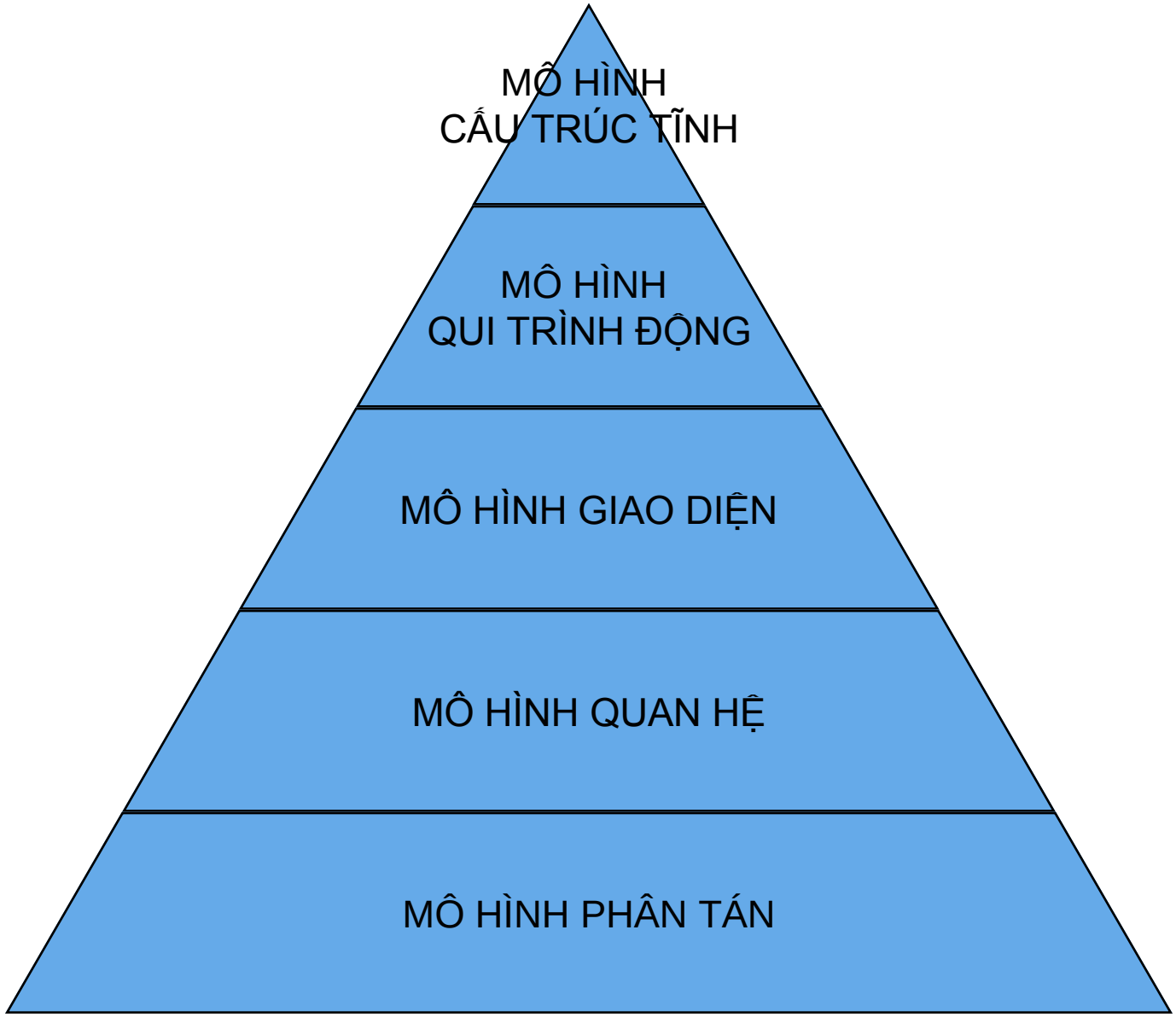
Thiết kế kiến trúc là gì? (tt)

Chú ý rằng các hệ thống có cùng miền ứng dụng có thể có các kiến trúc chung để phản ánh những khái niệm liên quan đến miền ứng dụng đó. Do đó, khả năng tái sử dụng lại kiến trúc hệ thống là rất cao.

Thiết kế kiến trúc là gì? (tt6)

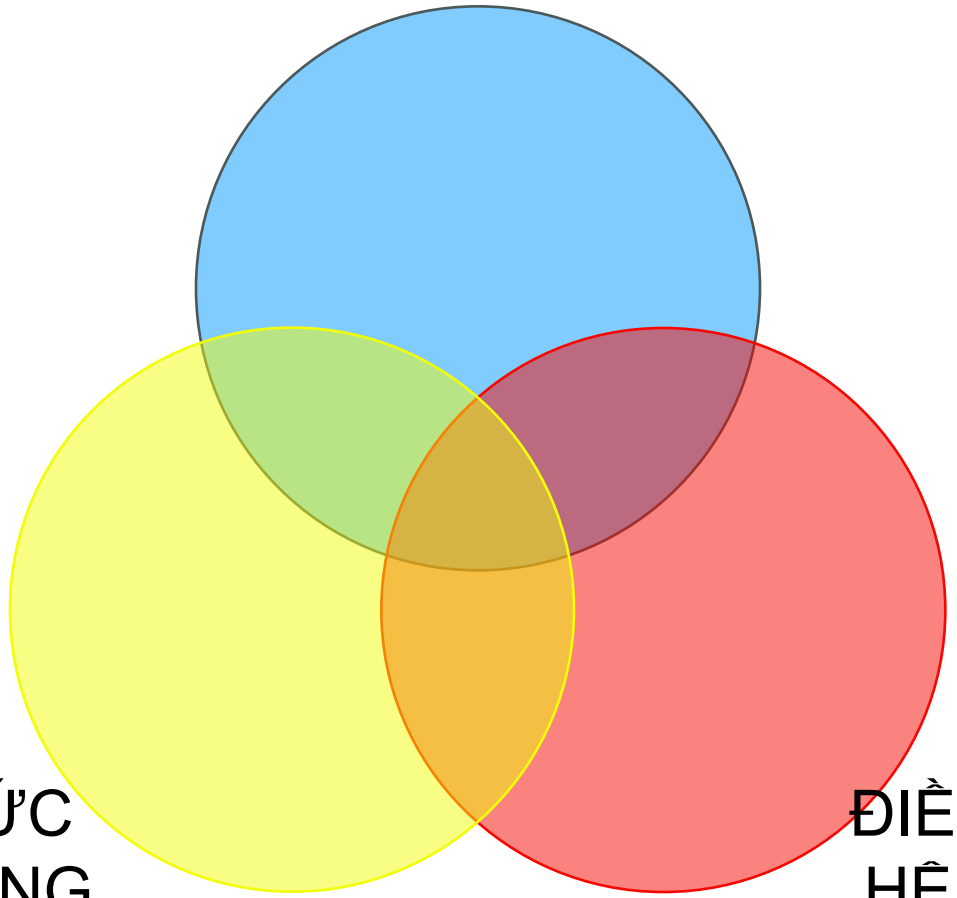
- Các mô hình kiến trúc cơ bản:
  - Mô hình cấu trúc tĩnh: mô tả các thành phần hệ thống chính.
  - Mô hình quy trình động: biểu diễn quy trình cấu trúc hoạt động của hệ thống.
  - Mô hình giao diện: định nghĩa tập hợp các giao diện của hệ thống con
  - Mô hình quan hệ: biểu diễn quan hệ giữa các hệ thống con.
  - Mô hình phân tán: biểu diễn cách cài đặt các hệ thống con trên máy tính.

Thiết kế kiến trúc là gì? (tt6)



Thiết kế kiến trúc là gì? (tt6)

PHÂN RÃ
HỆ THỐNG



TỔ CHỨC
HỆ THỐNG

ĐIỀU KHIỂN
HỆ THỐNG

Tổ chức hệ thống

- Tổ chức hệ thống phản ánh chiến lược cơ bản được sử dụng để cấu trúc hệ thống. Trong quá trình thiết kế kiến trúc hệ thống, hoạt động đầu tiên phải thực hiện là xây dựng mô hình tổ chức hệ thống.
- Có 3 phương pháp tổ chức hệ thống thường được sử dụng:
  - Kho dữ liệu dùng chung
  - Server và các dịch vụ dùng chung (client-server)
  - Phân lớp

Tổ chức hệ thống (tt)

- Kho dữ liệu dùng chung

  - Các hệ thống con phải trao đổi dữ liệu và làm việc với nhau một cách hiệu quả. Việc trao đổi dữ liệu được thực hiện theo hai cách:

    Dữ liệu chia sẻ được lưu ở CSDL trung tâm hoặc kho dữ liệu và được tất cả các hệ thống con truy nhập.
    Mỗi hệ thống con bảo trì CSDL của chính nó và truyền dữ liệu một cách tường minh cho các hệ thống con khác.

Tổ chức hệ thống (tt)

Mô hình kho dữ liệu dùng chung thường được sử dụng nếu số lượng dữ liệu dùng chung lớn. Ưu điểm của mô hình này là:

Phương pháp hiệu quả để chia sẻ số lượng lớn dữ liệu.

- Các hệ thống con không cần quan tâm tới những hoạt động liên quan đến dữ liệu như: sao lưu, bảo mật,… vì đã có bộ quản lý trung tâm thực hiện nhiệm vụ này.

Tổ chức hệ thống (tt)

Tuy nhiên, việc sử dụng kho dữ liệu dùng chung cũng có một số nhược điểm sau:

- Tất cả các hệ thống con phải chấp nhận mô hình kho dữ liệu.
- Việc cải tiến dữ liệu rất phức tạp và tốn kém
- Khó phân tán một cách hiệu quả
- Không có giới hạn cho các chính sách quản lý cụ thể.

Tổ chức hệ thống (tt)

- Mô hình client - server
  - Mô hình kiến trúc client-server là một mô hình hệ thống trong đó hệ thống bao gồm một tập hợp các server cung cấp dịch vụ và các client truy cập và sử dụng các dịch vụ đó. Các thành phần chính của mô hình này bao gồm:
    - Tập hợp các server sẽ cung cấp những dịch vụ cụ thể như: in ấn, quản lý dữ liệu…
    - Tập hợp các client truy cập đến server để yêu cầu cung cấp dịch vụ.
    - Hệ thống mạng cho phép client truy cập tới dịch vụ mà server cung cấp.

Tổ chức hệ thống (tt)

- Client phải biết tên của server và các dịch vụ mà server cung cấp. Nhưng server thì không cần xác định rõ client và hiện tại có bao nhiêu client. Client tạo ra một yêu cầu tới server và chờ server trả lời.

Tổ chức hệ thống (tt4)

- Ưu điểm của mô hình client server là:

  - Phân tán dữ liệu rõ ràng

  - Sử dụng các hệ thống được kết nối mạng một cách hiệu quả và chi phí dành cho phần cứng có thể rẻ hơn.

  - Dễ dàng bổ sung hoặc nâng cấp server

Tổ chức hệ thống (tt4)

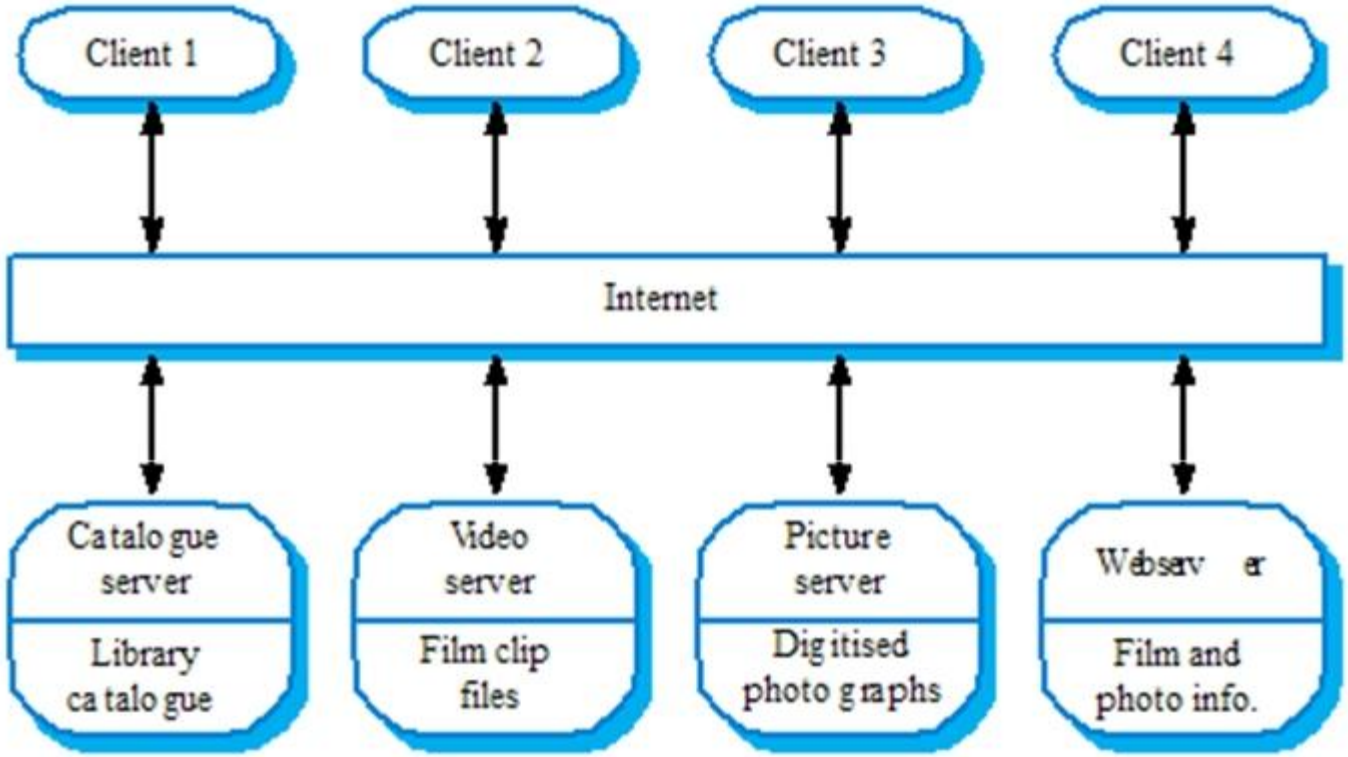
Nhược điểm của mô hình client server là:

-Không phải là mô hình dữ liệu dùng chung nên các hệ thống con có thể sử dụng các tổ chức dữ liệu khác nhau. Do đó, việc trao đổi dữ liệu có thể không hiệu quả.

-Không đăng ký tên và dịch vụ tập trung. Điều này làm cho việc tìm kiếm server hoặc các dịch vụ rất khó khăn.

Tổ chức hệ thống (tt5)

- Mô hình client - server (tt1)



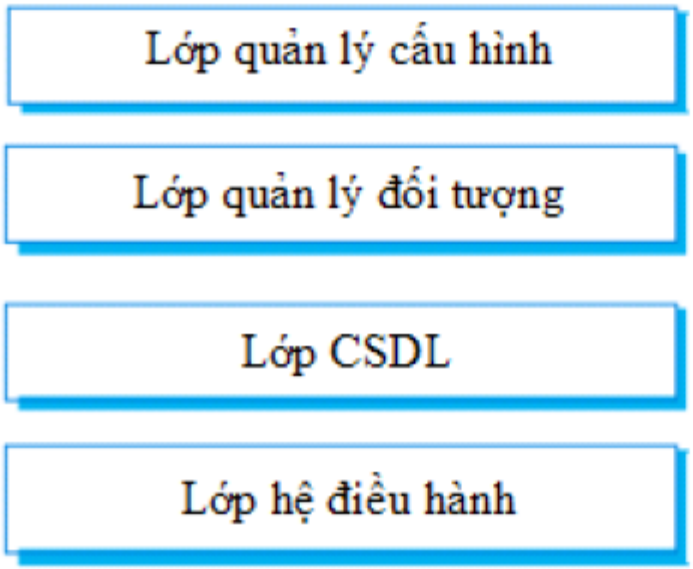
Mô hình client –server của hệ thống thư viện phim và ảnh

Tổ chức hệ thống (tt)

Mô hình phân lớp

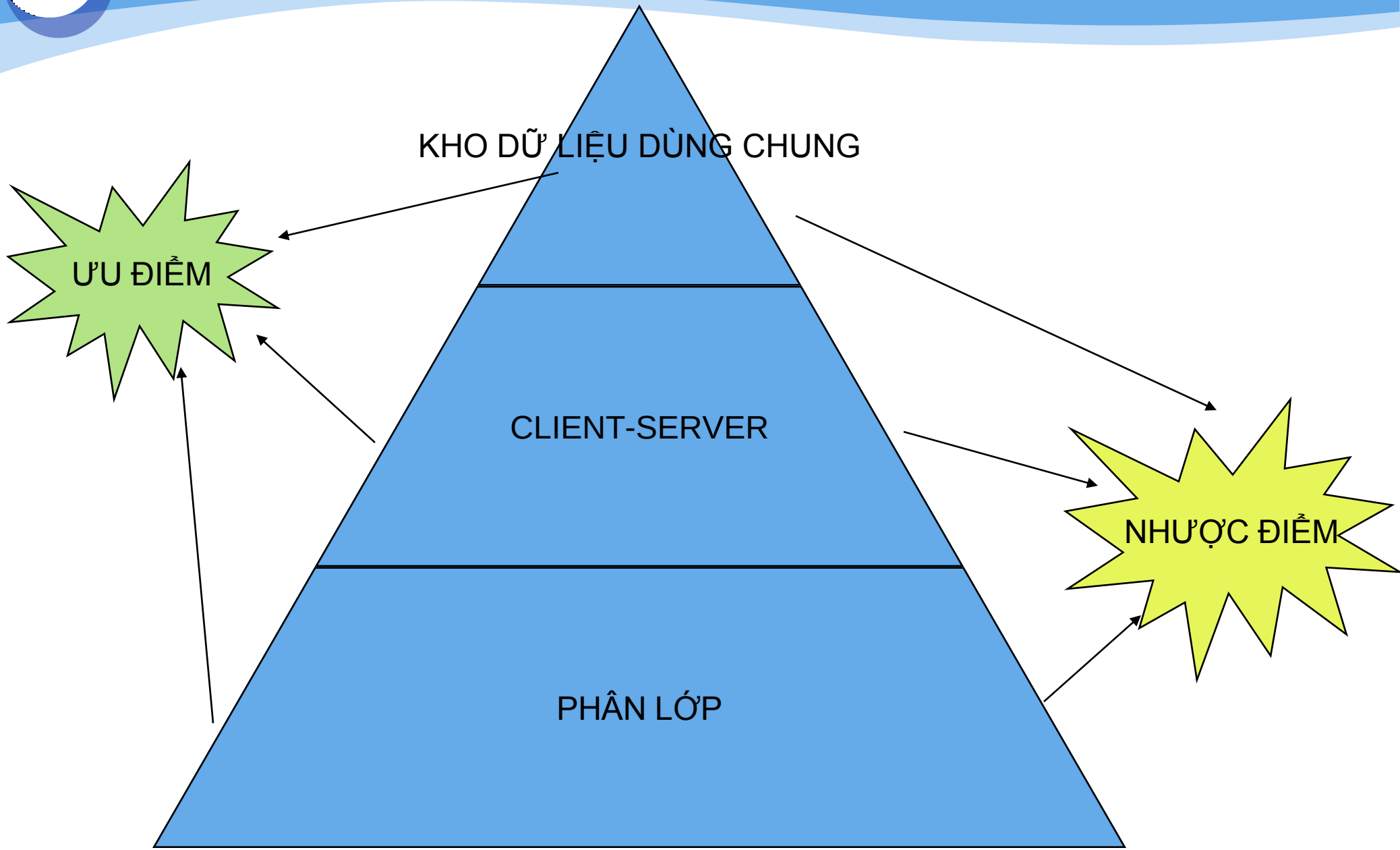
Mô hình phân lớp tổ chức hệ thống thành nhiều lớp và mỗi lớp cung cấp một dịch vụ. Mô hình này thường được sử dụng để mô hình hoá giao diện của hệ thống con

Mô hình phân lớp hỗ trợ phát triển hệ thống con theo kiểu tăng vòng ở nhiều lớp khác nhau. Khi giao diện của một lớp thay đổi thì chỉ những lớp liền kề nó mới bị ảnh hưởng.



Mô hình phân lớp của hệ thống quản lý phiên bản

Tổ chức hệ thống (tt)



Phân rã hệ thống

Sau khi cấu trúc hệ thống đã được lựa chọn, ta cần phải xác định phương pháp phân rã các hệ thống con thành các mô-đun

- Hệ thống con là một hệ thống có thể vận hành một cách độc lập, có thể sử dụng một số dịch vụ được cung cấp bởi các hệ thống con khác hoặc cung cấp dịch vụ cho các hệ thống con khác sử dụng.

Phân rã hệ thống

-Mô-đun là một thành phần hệ thống cung cấp các dịch vụ cho các thành phần khác, nhưng nó thường không được coi như là một hệ thống riêng, độc lập.

-Có hai cách để phân rã các hệ thống con thành các mô-đun:

- Phân rã hướng đối tượng: hệ thống được phân rã thành các đối tượng tương tác với nhau.
- Pipeline hướng chức năng hoặc luồng dữ liệu: hệ thống được phân rã thành các mô-đun chức năng chịu trách nhiệm chuyển đổi thông tin đầu vào thành kết quả đầu ra.

Phân rã hệ thống

- Phân rã hướng đối tượng
  - Mô hình kiến trúc hướng đối tượng cấu trúc hệ thống thành một tập hợp các đối tượng gắn kết dựa trên các giao diện đã được định nghĩa.
  - Phân rã hướng đối tượng liên quan tới việc xác định lớp đối tượng, các thuộc tính và phương thức của nó. Khi cài đặt lớp, các đối tượng sẽ được tạo ra từ các lớp này và có một số mô hình điều khiển được sử dụng để kết hợp các phương thức của đối tượng.

Phân rã hệ thống

- Ưu điểm của mô hình hướng đối tượng:

Đối tượng được gắn kết sao cho khi thay đổi cách cài đặt chúng có thể không ảnh hưởng tới các đối tượng khác.

Đối tượng phản ánh thực thể trong thế giới thực.

Các ngôn ngữ lập trình hướng đối tượng được sử dụng rộng rãi.

Phân rã hệ thống

- Pipeline hướng chức năng

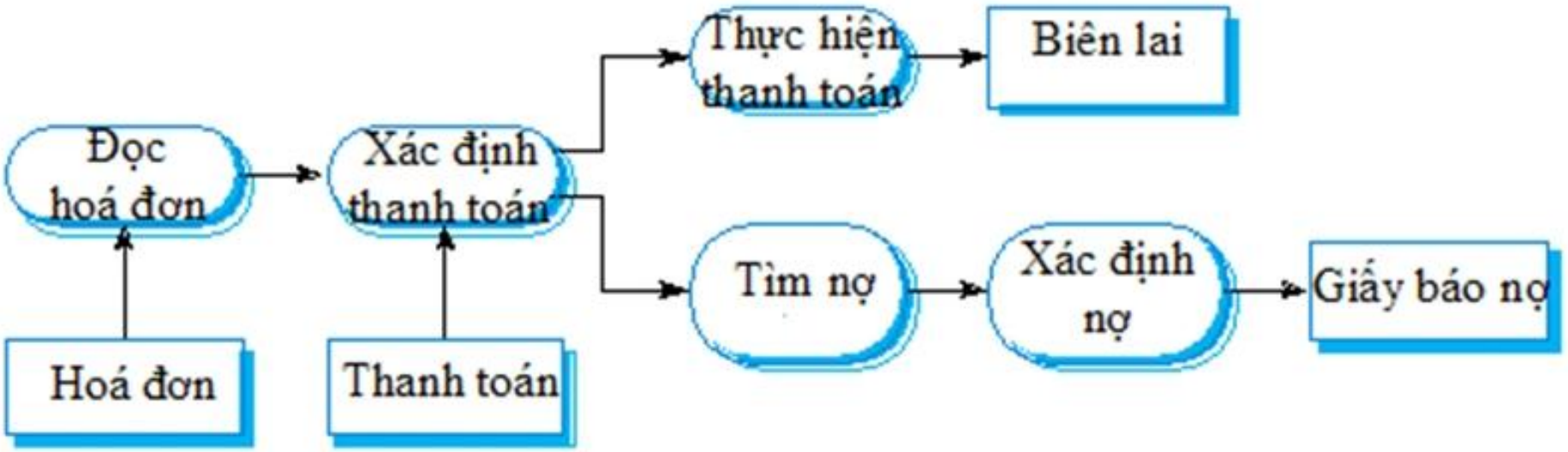
Mô hình pipeline hướng chức năng hoặc mô hình luồng dữ liệu là quy trình chuyển đổi thông tin đầu vào thành kết quả đầu ra. Dữ liệu được xử lý trong quy trình có thể là riêng lẻ hoặc theo nhóm

Phân rã hệ thống

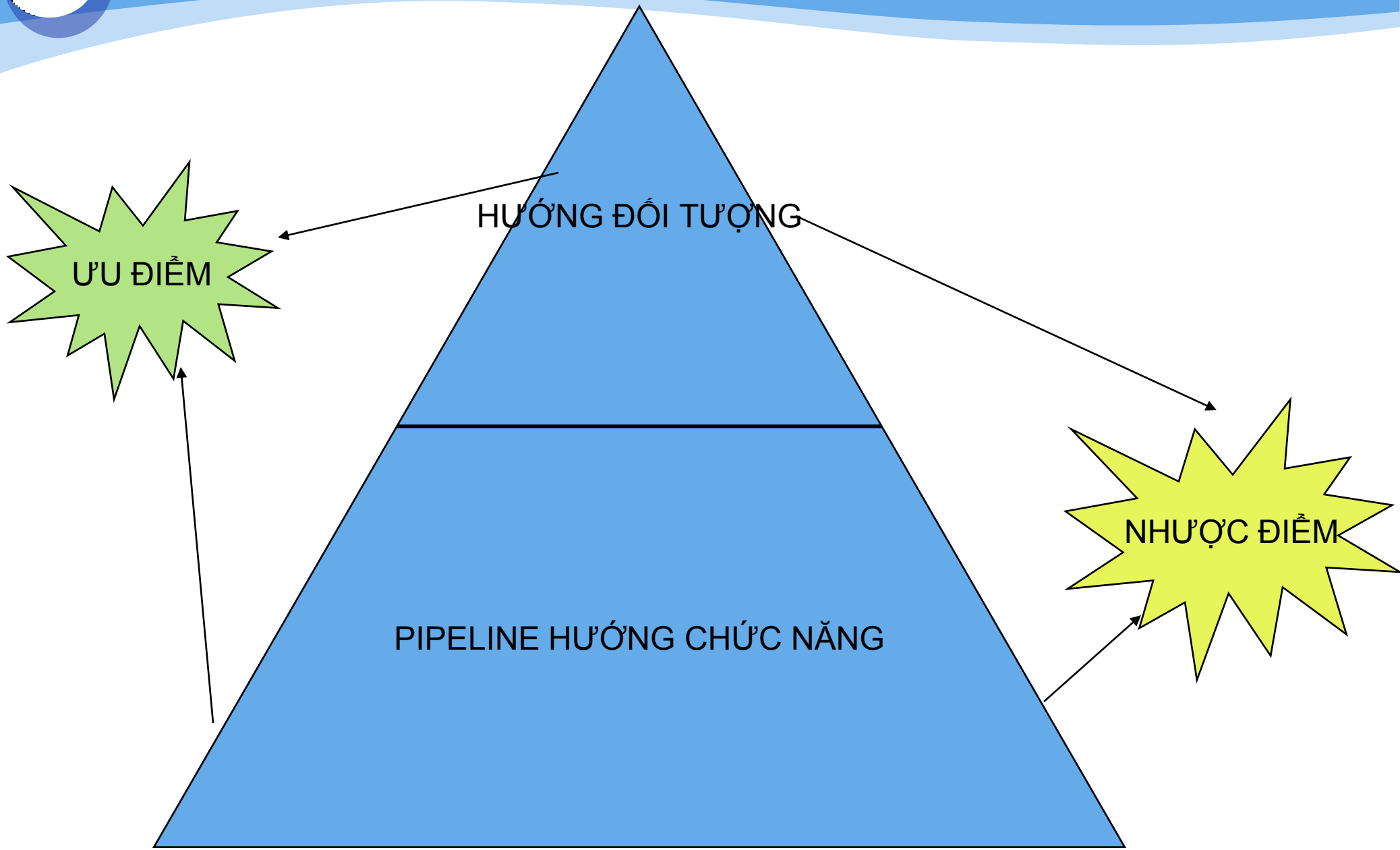
- Ưu điểm của mô hình:
  - Hỗ trợ tái sử dụng quy trình chuyển đổi
  - Cung cấp tài liệu để giao tiếp với stakeholder
  - Dễ dàng bổ sung thêm quy trình chuyển đổi mới.
  - Dễ dàng thực hiện, kể cả với hệ thống tuần tự hoặc song song.
- Tuy nhiên, mô hình này yêu cầu phải có định dạng dữ liệu chung.

Phân rã hệ thống

- Ví dụ: Mô hình luồng dữ liệu của hệ thống xử lý hoá đơn



Phân rã hệ thống



Các chiến lược điều khiển

- Giới thiệu

  - Các mô hình cấu trúc hệ thống có liên quan tới cách phân rã hệ thống thành nhiều hệ thống con. Để hệ thống làm việc tốt, phải điều khiển được các hệ thống con => các dịch vụ của chúng phải được thực hiện đúng chỗ và đúng thời điểm.

  - Có 2 loại chiến lược điều khiển:

Các chiến lược điều khiển

Điều khiển tập trung: một hệ thống con chịu trách nhiệm kiểm soát, khởi tạo hoặc dừng các hệ thống con khác.

Điều khiển hướng sự kiện: mỗi hệ thống đáp ứng với các sự kiện xảy ra từ các hệ thống con khác hoặc từ môi trường của hệ thống.

Các chiến lược điều khiển (tt1)

- Điều khiển tập trung

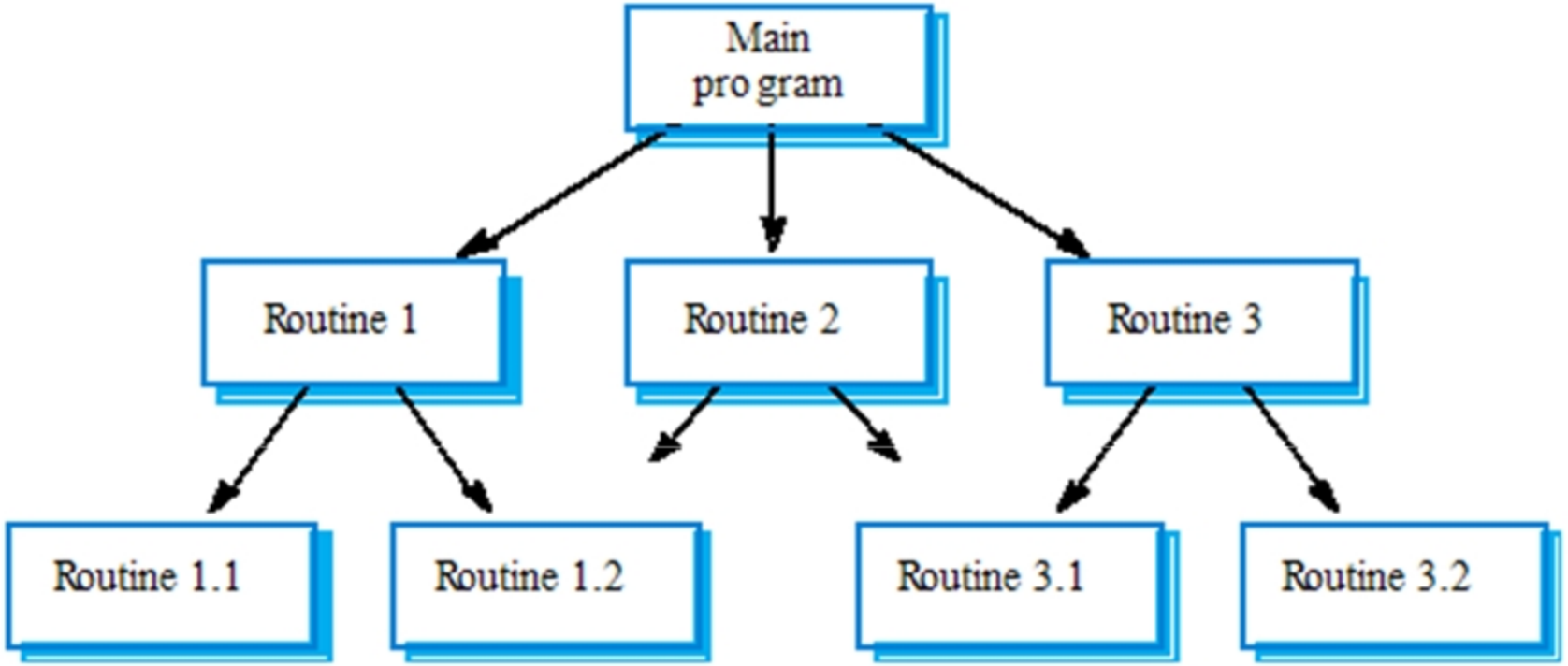
  - Hệ thống con điều khiển chịu trách nhiệm quản lý việc thực hiện của các hệ thống con khác.
  - Chiến lược điều khiển tập trung gồm 2 loại mô hình:

  - Mô hình gọi trả lời (call-return)

    - Gồm các thủ tục con được sắp xếp phân cấp, thủ tục điều khiển nằm ở đỉnh của cấu trúc phân cấp và di chuyển dần xuống dưới.

Các chiến lược điều khiển (tt2)

- Mô hình gọi trả lời (call-return) (tt)



Mô hình gọi - trả lời

Các chiến lược điều khiển (tt3)

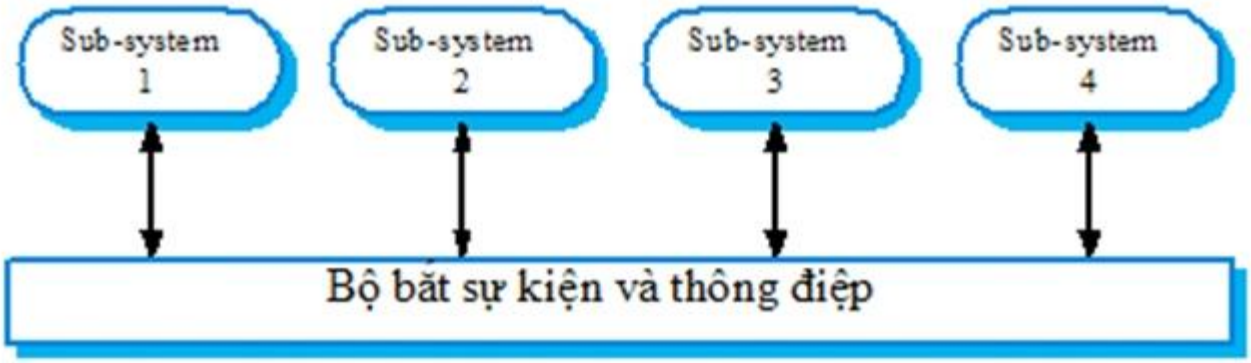
- Điều khiển hướng sự kiện
  - Mô hình lan truyền (Broadcast)
    - Trong mô hình lan truyền, sự kiện được lan truyền tới tất cả các hệ thống con. Bất kỳ hệ thống nào nếu có thể bắt được sự kiện này thì sẽ xử lý nó.
    - Mô hình này có hiệu quả đối với việc tích hợp các hệ thống con trên nhiều máy tính khác nhau trong cùng một mạng.

Các chiến lược điều khiển (tt4)

- Các hệ thống con phải đăng ký những sự kiện mà nó có thể bắt. Khi những sự kiện này xảy ra, điều khiển sẽ được truyền cho hệ thống con có thể bắt được sự kiện đó. Các hệ thống phải quyết định sự kiện nào là sự kiện mà nó đã đăng ký.

Các chiến lược điều khiển (tt5)

- Mô hình lan truyền (Broadcast) (tt)



Mô hình điều khiển lan truyền

Các chiến lược điều khiển (tt6)

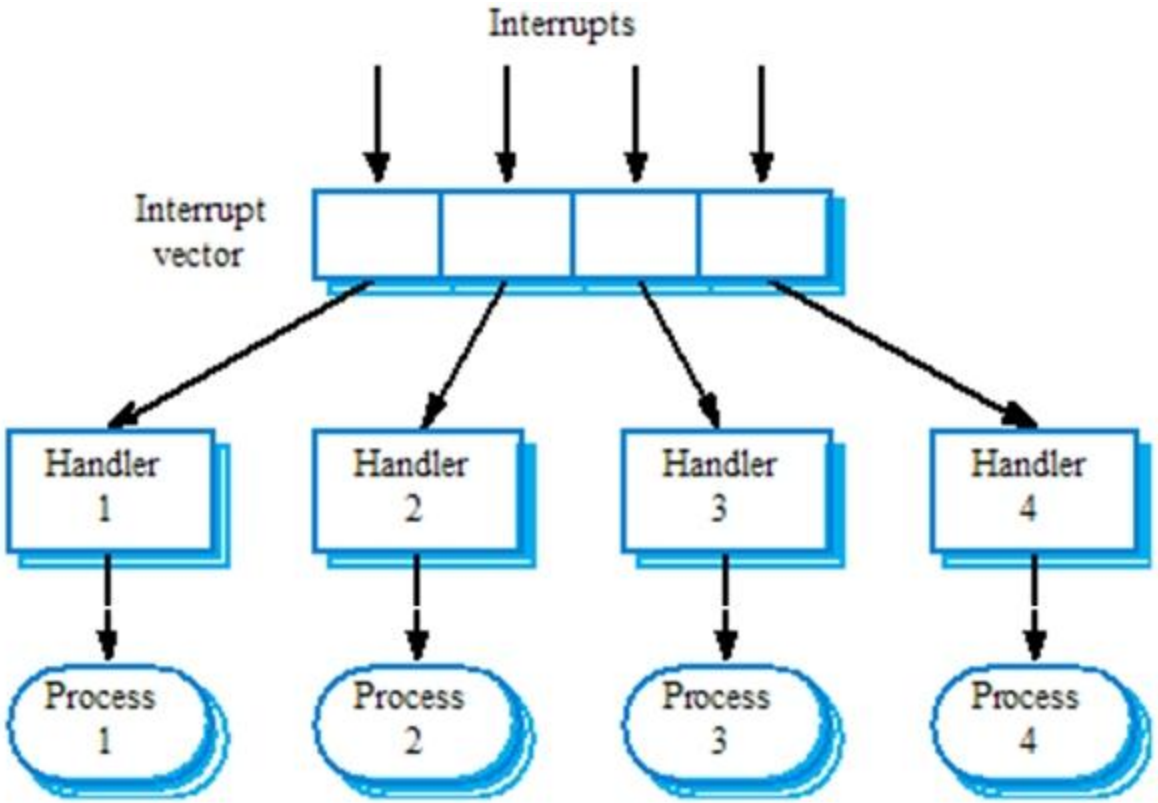
- Điều khiển hướng sự kiện (tt)
  - Mô hình hướng ngắt (Interrupt-driven)
    - Mô hình hướng ngắt được sử dụng trong các hệ thống thời gian thực trong đó các ngắt được phát hiện bởi bộ bắt ngắt (interrupt handler) và được truyền cho một số các thành phần khác để xử lý.

Các chiến lược điều khiển (tt7)

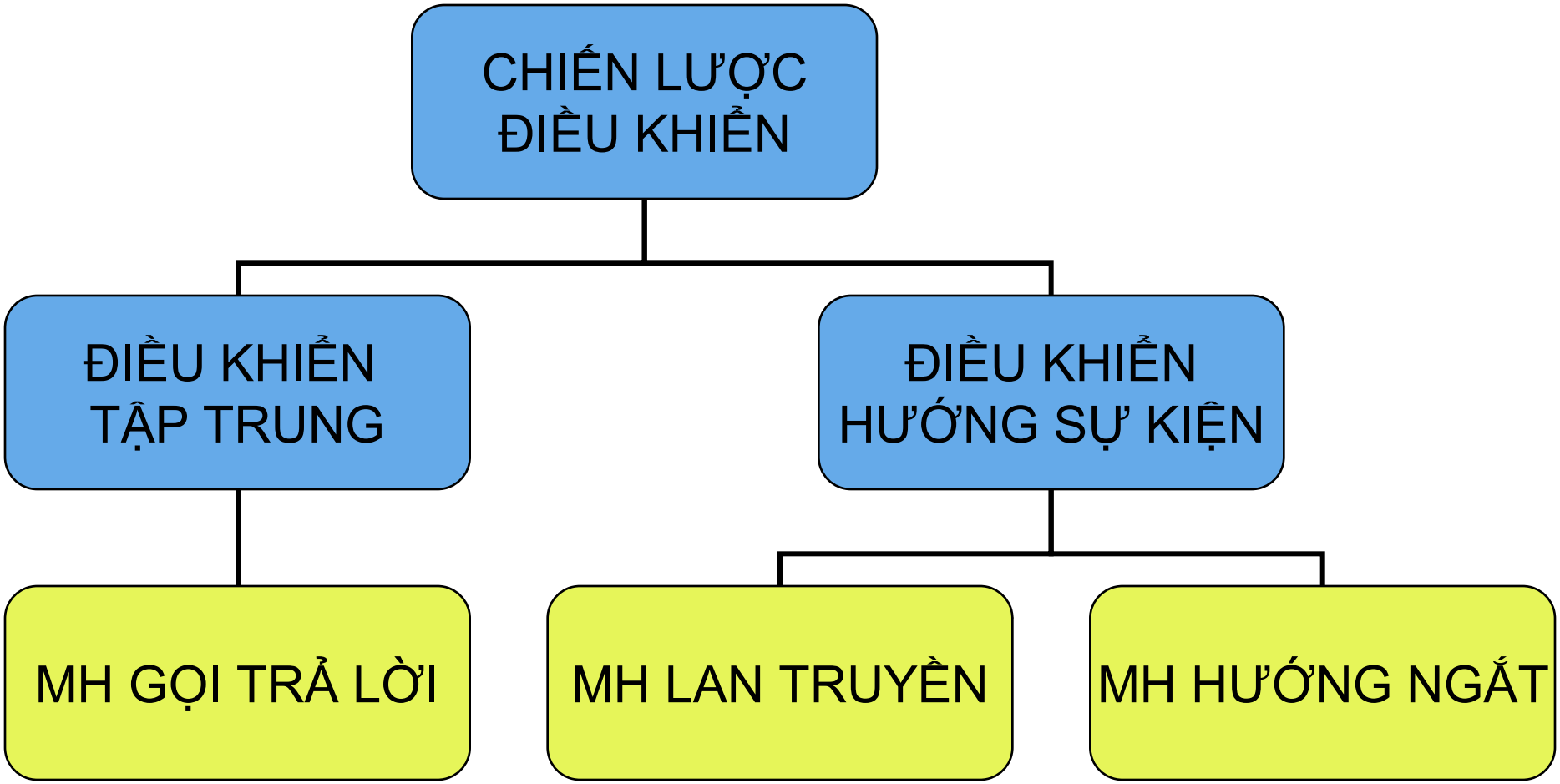
- Các kiểu ngắt và bộ bắt tương ứng được định nghĩa trước. Mỗi kiểu ngắt được gắn với một vị trí nhớ và một bộ chuyển mạch để đưa ngắt tới bộ bắt tương ứng của nó.
- Mô hình này cho phép đáp ứng rất nhanh, nhưng lập trình khá phức tạp và khó đánh giá.

Các chiến lược điều khiển (tt8)

- Mô hình hướng ngắt (Interrupt-driven) (tt)



Các chiến lược điều khiển (tt8)



Chương 7

Thiết kế giao diện người dùng

Giới thiệu

- Nguyên tắc quan trọng khi xây dựng một hệ thống phần mềm
  - Người sử dụng không quan tâm đến cấu trúc bên trong của hệ thống,
  - Người dùng đánh giá hệ thống thông qua giao diện
  - Nếu cảm thấy giao diện không thích hợp, khó sử dụng => không sử dụng cả hệ thống => dự án thất bại.

Giao diện người dùng

- Giao diện người dùng cần phải được thiết kế sao cho phù hợp với kỹ năng, kinh nghiệm và sự trông đợi của người sử dụng nó.

- Mục tiêu:

  - Nắm được sự ảnh hưởng của người sử dụng tới giao diện
  - Một số nguyên tắc khi thiết kế giao diện người dùng
  - Phân loại các khả năng tương tác giữa người và máy để thiết kế giao diện cho phù hợp
  - Biết cách biểu diễn thông tin cho phù hợp với người sử dụng

Giao diện người dùng (tt)

Một số đặc điểm của người sử dụng có liên quan đến giao diện hệ thống:

- Khả năng nhớ tức thời của con người bị hạn chế: con người chỉ có thể nhớ ngay một số loại thông tin. Nếu ta biểu diễn nhiều hơn thì có thể khiến người sử dụng không nhớ hết và gây ra các lỗi.

Giao diện người dùng (tt)

- Người sử dụng có thể gây ra lỗi: khi người sử dụng gây ra lỗi khiến hệ thống sẽ hoạt động sai, những thông báo không thích hợp có thể làm tăng áp lực lên người sử dụng và càng xảy ra nhiều lỗi hơn.

Người sử dụng là khác nhau: con người có những khả năng khác nhau => không nên chỉ thiết kế giao diện phù hợp với những khả năng của chính họ.

Giao diện người dùng (tt)

- Thiết kế giao diện phải phụ thuộc vào yêu cầu, kinh nghiệm và khả năng của người sử dụng hệ thống.

-Người thiết kế cũng nên quan tâm đến những giới hạn vật lý và tinh thần của con người và nên nhận ra rằng con người luôn có thể gây ra lỗi.

- Không phải tất cả các nguyên tắc thiết kế giao diện đều có thể được áp dụng cho tất cả các giao diện.

Giao diện người dùng (tt)

Các nguyên tắc thiết kế giao diện:

- Sự quen thuộc của người sử dụng: sử dụng thuật ngữ và các khái niệm quen thuộc với người sử dụng hơn là những khái niệm liên quan đến máy tính.

  - Ví dụ: hệ thống văn phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy … mà không nên sử dụng những khái niệm như thư mục, danh mục …

Giao diện người dùng (tt)

Thống nhất: hệ thống nên hiển thị ở mức thống nhất thích hợp.

Ví dụ: các câu lệnh và menu nên có cùng định dạng

-Tối thiểu hoá sự bất ngờ: nếu một yêu cầu được xử lý theo cách đã biết trước thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tư.

Giao diện người dùng (tt)

- Khả năng phục hồi: hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi.
- Hướng dẫn người sử dụng: như hệ thống trợ giúp, hướng dẫn trực tuyến
- Tính đa dạng: hỗ trợ nhiều loại tương tác cho nhiều loại người sử dung khác nhau.

Ví dụ: nên hiển thị phông chữ lớn với những người cận thị.

Giao diện người dùng (tt)

- Biểu diễn thông tin

-Biểu diễn thông tin có liên quan tới việc hiển thị các thông tin trong hệ thống tới người sử dụng. Thông tin có thể được biểu diễn một cách trực tiếp hoặc có thể được chuyển thành nhiều dạng hiển thị khác như: dạng đồ hoạ, âm thanh …

Giao diện người dùng (tt)

Thông tin cần biểu diễn được chia thành hai loại:

- Thông tin tĩnh: được khởi tạo ở đầu của mỗi phiên. Nó không thay đổi trong suốt phiên đó và có thể là ở dạng số hoặc dạng văn bản.
- Thông tin động: thay đổi trong cả phiên sử dụng và sự thay đổi này phải được người sử dụng quan sát.

Giao diện người dùng (tt)

- Các nhân tố ảnh hưởng tới việc hiển thị thông tin:

- Người sử dụng thích hiển thị một phần thông tin hay quan hệ dữ liệu?

- Thông tin thay đổi nhanh như thế nào?

- Sự thay đổi đó có cần phải thể hiện ngay lập tức hay không?

- Người sử dụng có phải thực hiện các hành động để đáp ứng với sự thay đổi không?

- Thông tin ở dạng văn bản hay dạng số?

Giao diện người dùng (tt)

- Nếu cần hiển thị số lượng lớn thông tin thì nên trực quan hoá dữ liệu.

  - Ví dụ: thông tin về thời tiết được hiển thị dưới dạng biểu đồ, trạng thái của mạng điện thoại nên được hiển thị bởi các nút có liên kết với nhau.

- Sử dụng màu trong khi thiết kế giao diện -> giúp cho người sử dụng hiểu được những cấu trúc thông tin phức tạp.

Giao diện người dùng (tt)

- Tuy nhiên, khi sử dụng màu để thiết kế giao diện có thể gây phản tác dụng -> một số hướng dẫn:

- Giới hạn số lượng màu được sử dụng và không nên lạm dụng việc sử dụng màu.

- Thay đổi màu khi thay đổi trạng thái của hệ thống.

Giao diện người dùng (tt)

- Sử dụng màu để hỗ trợ cho những nhiệm vụ mà người dùng đang cố gắng thực hiện.
- Sử dụng màu một cách thống nhất và cẩn thận.
- Cẩn thận khi sử dụng các cặp màu.

Giao diện người dùng (tt)

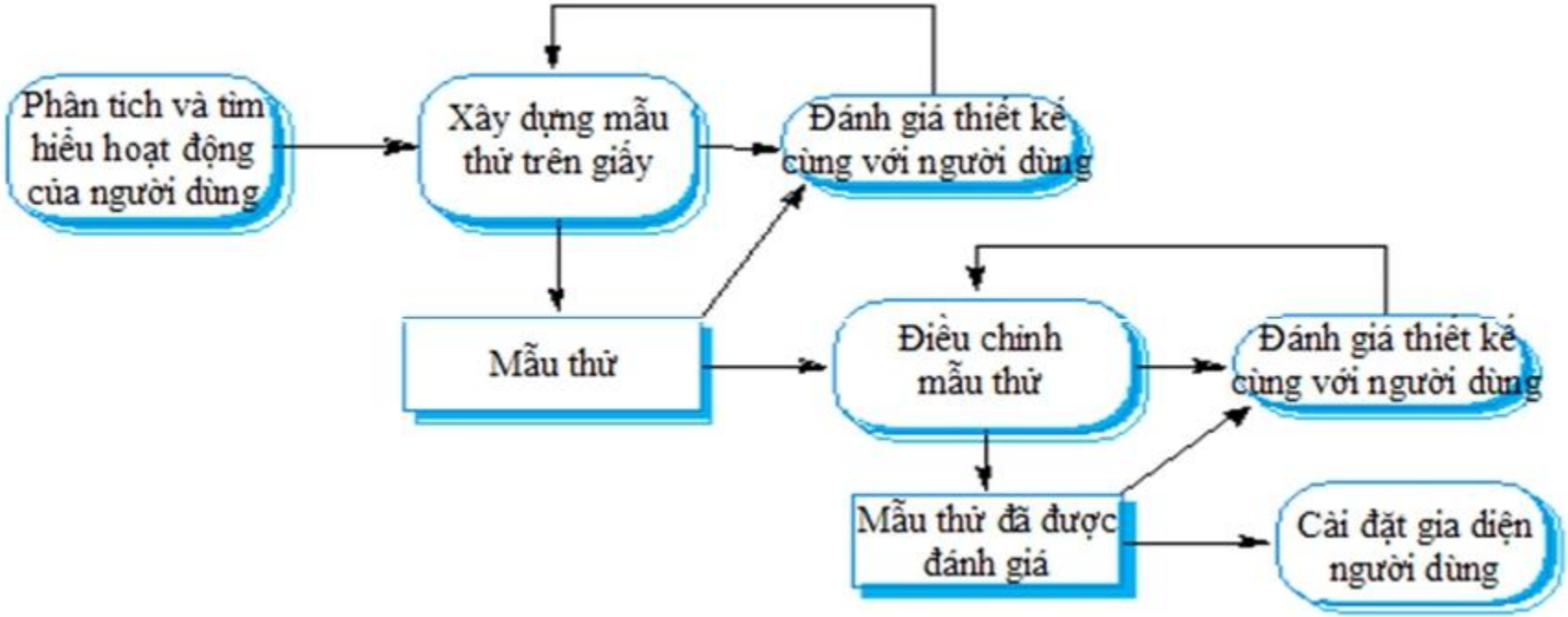
Thông báo lỗi:

- Nếu thông báo lỗi nghèo nàn có thể làm cho người sử dụng từ chối hơn là chấp nhận hệ thống.
- thông báo lỗi nên ngắn gọn, xúc tích, thống nhất và có cấu trúc.
- Việc thiết kế thông báo lỗi nên dựa vào kỹ năng và kinh nghiệm của người sử dụng.

Quy trình thiết kế UI

- Thiết kế giao diện người dùng là một quy trình lặp lại bao gồm sự cộng tác giữa người sử dụng và người thiết kế.
- Trong quy trình này gồm 3 hoạt động cơ bản:
  - Phân tích người sử dụng: tìm hiểu những gì người sử dụng sẽ làm với hệ thống.
  - Lập mẫu thử hệ thống: xây dựng một tập các mẫu thử để thử nghiệm
  - Đánh giá giao diện: thử nghiệm các mẫu thử cùng với người sử dụng.

Quy trình thiết kế UI (tt)



Quy trình thiết kế UI (tt)

- Phân tích người sử dụng
  - Nếu không hiểu rõ những gì người sử dụng muốn làm với hệ thống sẽ không thể thiết kế được một giao diện hiệu quả.
  - Phân tích người sử dụng phải được mô tả theo những thuật ngữ để người sử dụng và những người thiết kế khác có thể hiểu được.

Quy trình thiết kế UI (tt)

-Các kỹ thuật phân tích:

- Phân tích nhiệm vụ: mô hình hoá các bước cần thực hiện để hoàn thành một nhiệm vụ.

- Phân tích nhiệm vụ phân cấp.

- Phỏng vấn và trắc nghiệm: hỏi người sử dụng về những gì mà họ làm. Khi phỏng vấn, nên dựa trên những câu hỏi có kết thúc mở -> người sử dụng cung cấp những thông tin mà họ nghĩ rằng nó là cần thiết; nhưng không phải tất cả các thông tin đó là có thể được sử dụng.

Quy trình thiết kế UI (tt)

Mô tả: quan sát người sử dụng làm việc và hỏi họ về những nghiệp vụ mà chưa được biết tới. Nên nhớ rằng có nhiều nhiệm vụ của người sử dụng thuộc về trực giác và rất khó để mô tả và giải thích chúng. Dựa trên kỹ thuật này ta có thể hiểu thêm về các ảnh hưởng xã hội và tổ chức tác động tới công việc đó.

Quy trình thiết kế UI (tt)

- Lập mẫu thử giao diện người dùng

  - Mẫu thử cho phép người sử dụng có được những kinh nghiệm trực tiếp với giao diện.
  - Lập mẫu thử là một quy trình gồm 2 trạng thái:

    Lập các mẫu thử trên giấy.

    Tinh chỉnh mẫu thử và xây dựng chúng

Quy trình thiết kế UI (tt)

- Đánh giá giao diện người dùng
  - Nên đánh giá bản thiết kế giao diện người dùng để xác định khả năng phù hợp của nó. Tuy nhiên, việc đánh giá trên phạm vi rộng tốn nhiều chi phí và không thể thực hiện được đối với hầu hết các hệ thống.

Quy trình thiết kế UI (tt)

- Các kỹ thuật đánh giá đơn giản:

  - Trắc nghiệm lại các phản hồi của người sử dụng Ghi lại quá trình sử dụng mẫu thử của hệ thống và đánh giá nó.

  - Lựa chọn những thông tin về việc sử dụng dễ dàng và các lỗi của người sử dụng.

  - Cung cấp mã lệnh trong phần mềm để thu thập những phản hồi của người sử dụng một cách trực tuyến.

Chương 8

Cải tiến phần mềm

Giới thiệu

- Thay đổi phần mềm là một điều không thể tránh khỏi vì những lí do sau:

- Những yêu cầu mới sẽ xuất hiện khi sử dụng phần mềm

- Môi trường nghiệp vụ thay đổi

- Các lỗi phần mềm cần phải sửa chữa

- Máy tính và các thiết bị mới được bổ sung vào hệ thống

- Hiệu năng hoặc độ tin cậy của hệ thống phải được cải thiện.

Giới thiệu (tt)

- Vấn đề quan trọng là phải quản lý các thay đổi đối với hệ thống phần mềm và hiểu được tầm quan trọng của việc cải tiến phần mềm:

  - Các tổ chức thường đầu tư một lượng vốn khá lớn vào các hệ thống phần mềm của họ. Cho nên họ có quyền đòi hỏi phải sở hữu một hệ thống hoàn hảo.

  - Để bảo trì giá trị sở hữu của tổ chức, họ phải thay đổi và cải tiến hệ thống.

Bảo trì phần mềm

- Bảo trì phần mềm chính là hoạt động chỉnh sửa chương trình sau khi nó đã được đưa vào sử dụng.
- Bảo trì thường không bao gồm những thay đổi chính liên quan tới kiến trúc của hệ thống.
- Những thay đổi trong hệ thống thường được cài đặt bằng cách điều chỉnh những thành phần đang tồn tại và bổ sung những thành phần mới cho hệ thống.

Bảo trì phần mềm (tt)

- Bảo trì là không thể tránh khỏi vì:
  - Các yêu cầu hệ thống thường thay đổi khi hệ thống đang được xây dựng vì môi trường thay đổi. Vì vậy, hệ thống được chuyển giao có thể không thoả mãn các yêu cầu
  - Các hệ thống phải được bảo trì nếu muốn là những phần hữu ích trong môi trường nghiệp vụ.

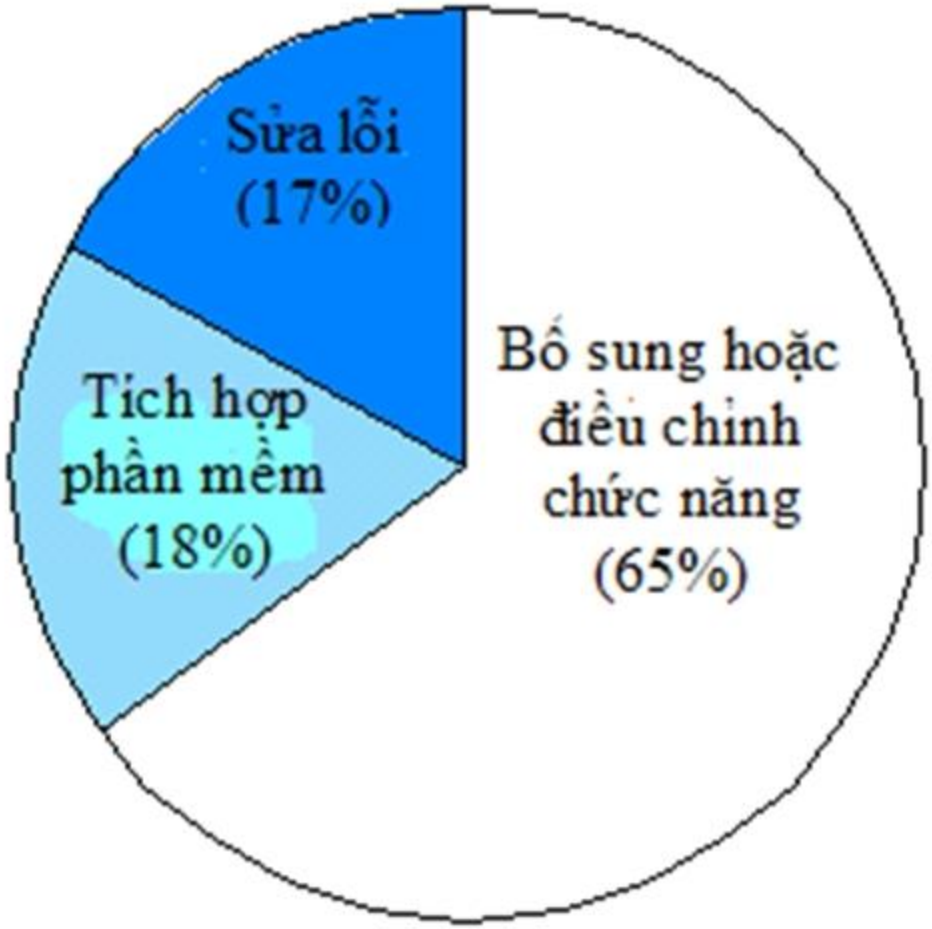
Bảo trì phần mềm (tt)

- Phân loại các kiểu bảo trì:
  - Bảo trì sửa lỗi: thay đổi hệ thống để sửa lại những khiếm khuyết nhằm thoả mãn yêu cầu hệ thống.
  - Bảo trì tích hợp hệ thống vào một môi trường vận hành khác
  - Bảo trì để bổ sung hoặc chỉnh sửa các yêu cầu chức năng của hệ thống: chỉnh sửa hệ thống sao cho thoả mãn các yêu cầu mới.

Bảo trì phần mềm (tt)

- Chi phí bảo trì thường lớn hơn chi phí xây dựng gấp từ 2 đến 100 lần phụ thuộc vào từng ứng dụng. Chi phí bảo trì bị ảnh hưởng bởi cả tác nhân kỹ thuật và phi kỹ thuật.
- Nếu bảo trì càng nhiều, sẽ càng làm thay đổi cấu trúc phần mềm -> bảo trì càng trở lên khó khăn hơn.
- Phần mềm có tuổi thọ càng cao thì chi phí bảo trì càng cao

Bảo trì phần mềm (tt)



Phân bổ chi phí bảo trì

Bảo trì phần mềm (tt)

- Các nhân tố ảnh hưởng đến chi phí bảo trì:

  -Sự ổn định của đội dự án: chi phí bảo trì sẽ giảm nếu nhân viên trong đội dự án không thay đổi.

  - Những trách nhiệm đã cam kết: người xây dựng hệ thống có thể không cam kết trách nhiệm bảo trì cho nên không có gì để bắt buộc họ phải thiết kế lại cho các thay đổi trong tương lai.

Bảo trì phần mềm (tt)

- Kỹ năng của nhân viên: nhân viên bảo trì thường không có kinh nghiệm và hiểu biết về miền ứng dụng của họ bị hạn chế.
- Tuổi thọ và cấu trúc chương trình: khi tuổi thọ và cấu trúc chương trình bị xuống cấp thì chúng càng trở nên khó hiểu và thay đổi nhiều.

Bảo trì phần mềm (tt)

- Dự đoán bảo trì
  - Dự đoán bảo trì có liên quan tới việc đánh giá những phần nào của hệ thống có thể gây ra lỗi và cần bao nhiêu chi phí để bảo trì.
  - Khả năng chịu được sự thay đổi phụ thuộc vào khả năng bảo trì của các thành phần bị ảnh hưởng bởi sự thay đổi đó.

Bảo trì phần mềm (tt)

- Thực hiện các thay đổi có thể làm hỏng hệ thống và giảm khả năng bảo trì của nó.
- Chi phí bảo trì phụ thuộc vào số lượng các thay đổi và chi phí thay đổi phụ thuộc vào khả năng bảo trì.

Bảo trì phần mềm (tt)

- Dự đoán thay đổi
  - Dự đoán số lượng các thay đổi có thể xảy ra và tìm hiểu mối quan hệ giữa hệ thống và môi trường của nó.
  - Sự thay đổi yêu cầu hệ thống có liên quan chặt chẽ tới sự thay đổi của môi trường.

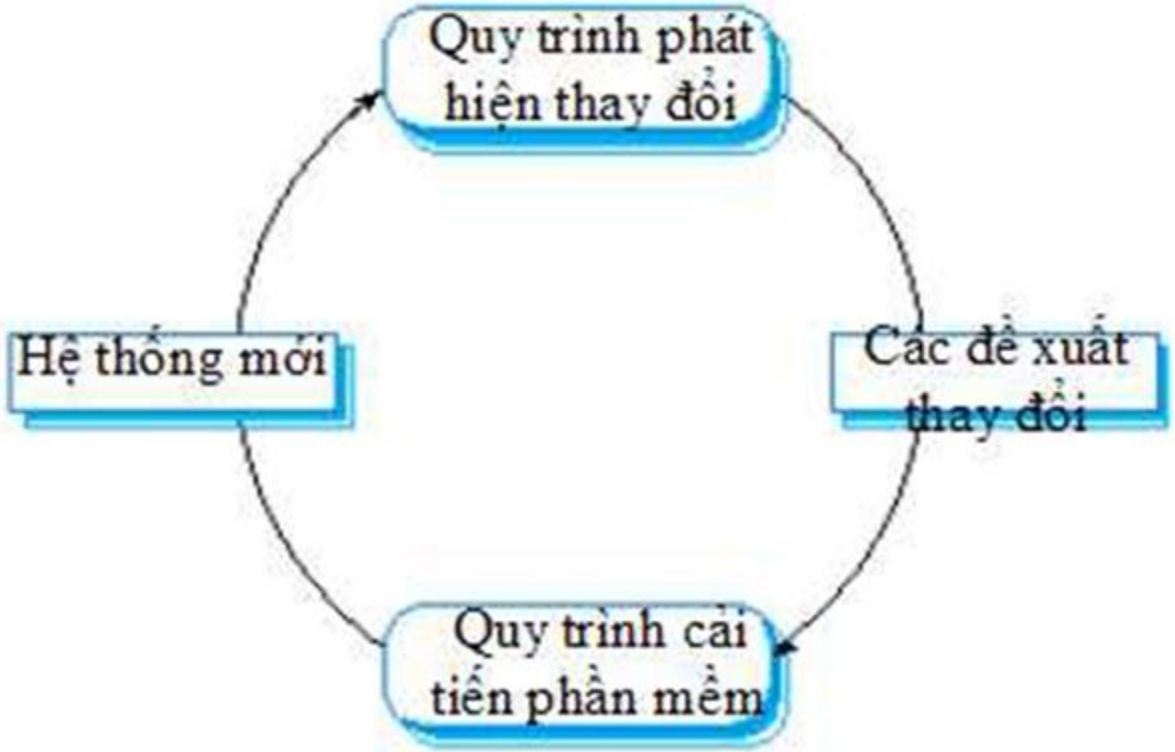
Các quy trình cải tiến phần mềm

- Các quy trình cải tiến phần mềm phụ thuộc vào:

  - Kiểu phần mềm cần bảo trì
  -Quy trình phát triển phần mềm đã được sử dụng
  - Kỹ năng và kinh nghiệm của các stakeholder.
  - Các đề xuất thay đổi là định hướng để cải tiến hệ thống.
  - Phát hiện thay đổi và cải tiến được thực hiện trong vòng đời hệ thống.

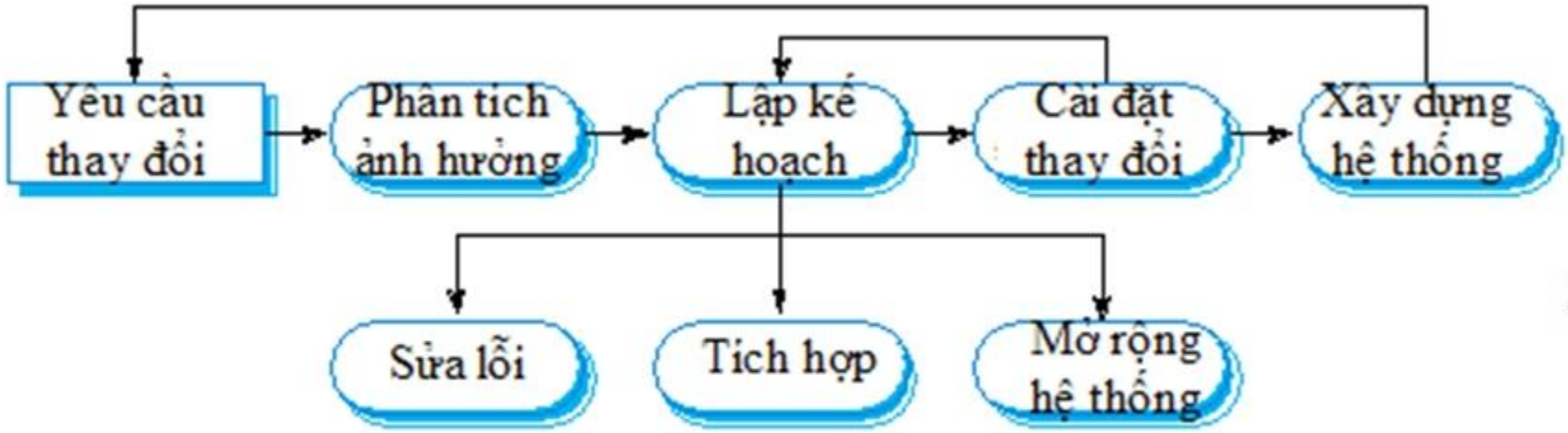
Các quy trình cải tiến phần mềm

Các hình vẽ sau đây thể hiện một cách khái quát các quy trình cải tiến hệ thống.



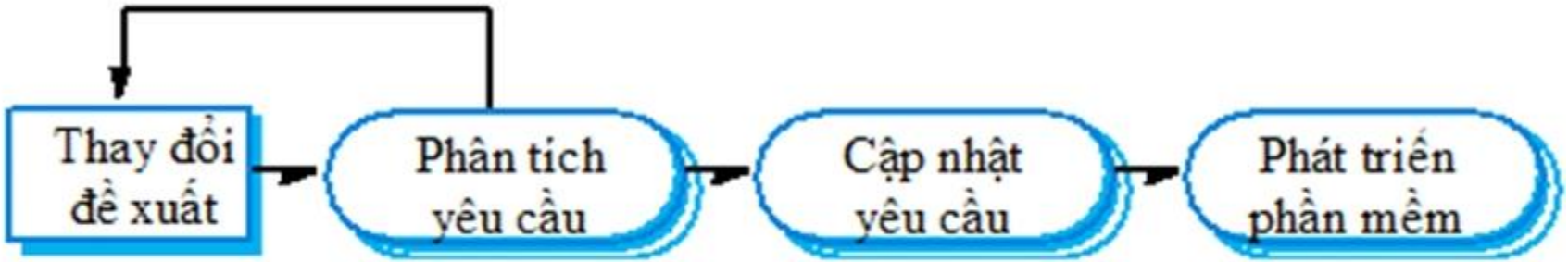
Phát hiện thay đổi và cải tiến

Các quy trình (tt)



Quy trình cải tiến hệ thống

Các quy trình (tt)



Cài đặt thay đổi

Các quy trình (tt)

Với các yêu cầu thay đổi khẩn cấp, ta có thể cài đặt chúng ngay mà không cần phải trải qua tất cả các pha của quy trình công nghệ phần mềm. Những yêu cầu thay đổi khẩn cấp thường xảy ra khi:

- Có một lỗi hệ thống nghiêm trọng xảy ra và cần phải sửa chữa.

Các quy trình (tt)

- Nếu những thay đổi về môi trường của hệ thống gây ra những hiệu ứng không mong đợi.
- Nếu sự thay đổi về mặt nghiệp vụ yêu cầu phải có đáp ứng nhanh

Các quy trình (tt)

- Để cải tiến hệ thống hiện có, bốn chiến lược cơ bản được đề xuất:

  - Tách hệ thống và chỉnh sửa các quy trình nghiệp vụ
  - Tiếp tục bảo trì hệ thống

Các quy trình (tt)

- Biến đổi hệ thống bằng cách tái kỹ nghệ để nâng cấp khả năng bảo trì của nó.
- Thay thế hệ thống bằng một hệ thống mới

Việc lựa chọn chiến lược cải tiến hệ thống phụ thuộc vào chất lượng hệ thống và nghiệp vụ của người dùng.

Các quy trình (tt)

- Đánh giá chất lượng hệ thống thông qua:

  - Quy trình nghiệp vụ: hệ thống xây dựng đã hỗ trợ cho các mục tiêu nghiệp vụ như thế nào?

  - Môi trường hệ thống: môi trường hệ thống có hiệu quả như thế nào và chi phí để bảo trì nó.

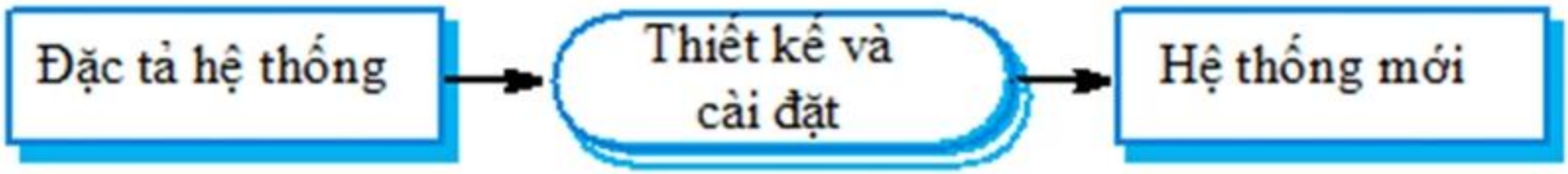
  - Khả năng ứng dụng: chất lượng của ứng dụng như thế nào

Tái kỹ nghệ hệ thống

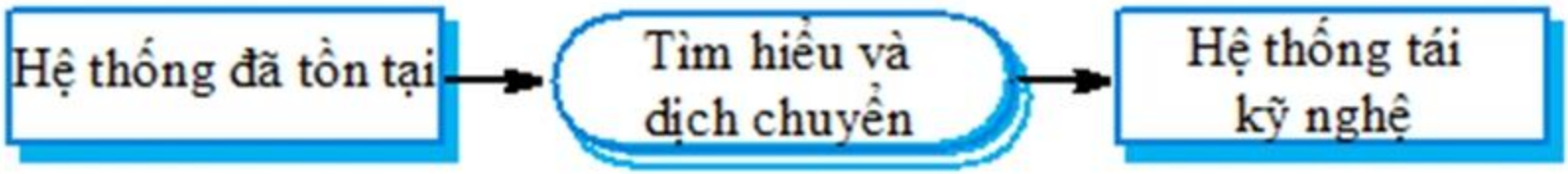
- Tái kỹ nghệ hệ thống là kỹ thuật cấu trúc lại hoặc viết lại một phần hay toàn bộ hệ thống mà không thay đổi chức năng của nó.
- Tái kỹ nghệ giúp giảm rủi ro và giảm chi phí vì trong quá trình xây dựng phần mềm mới thường phát sinh nhiều rủi ro.

Tái kỹ nghệ hệ thống (tt)

Mô hình phân biệt forward-engineering và re-engineering:



Mô hình forward-engineering



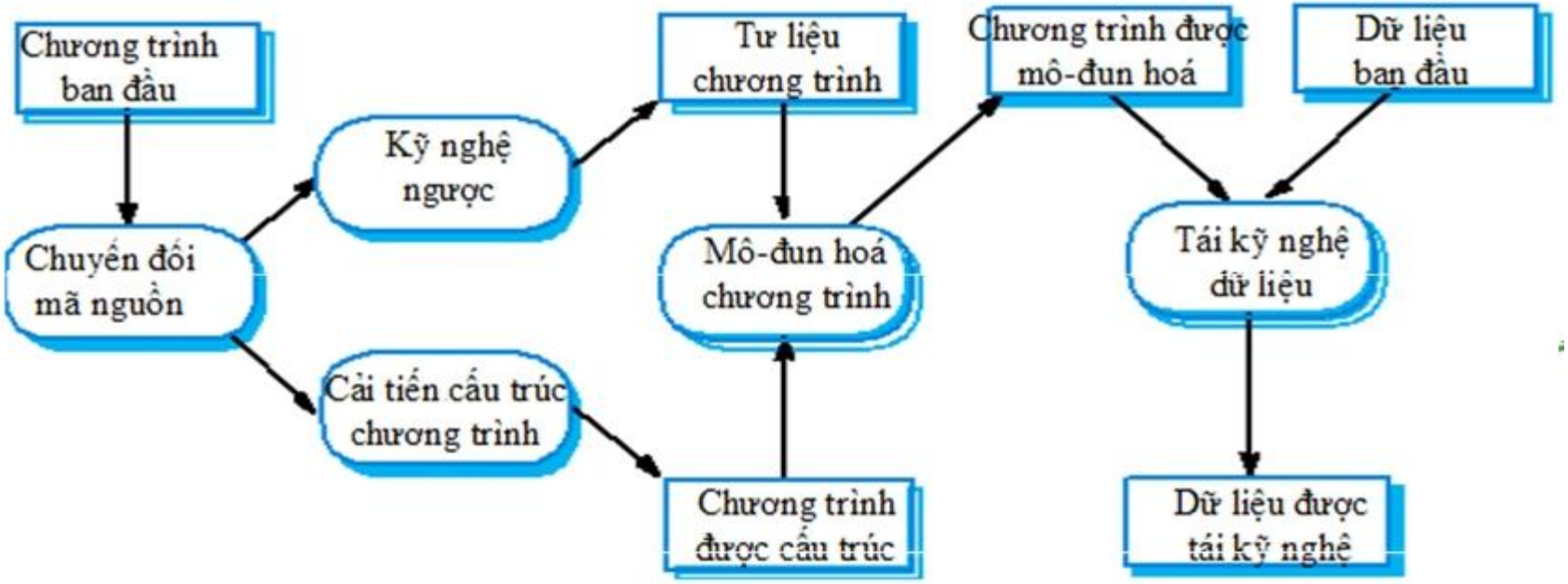
Mô hình re-engineering

Tái kỹ nghệ hệ thống (tt)

- Quy trình tái kỹ nghệ bao gồm các hoạt động sau:

- Dịch mã nguồn: chuyển mã lệnh thành ngôn ngữ mới.
- Kỹ nghệ ngược: phân tích chương trình để tìm hiểu nó.
- Cải thiện cấu trúc chương trình
- Mô-đun hoá chương trình: tổ chức lại cấu trúc chương trình
- Tái kỹ nghệ dữ liệu: thu dọn và cấu trúc lại dữ liệu hệ thống

Tái kỹ nghệ hệ thống (tt)



Tái kỹ nghệ hệ thống (tt)

- Các nhân tố ảnh hưởng tới chi phí tái kỹ nghệ:
  - Chất lượng của hệ thống được tái kỹ nghệ
  - Các công cụ hỗ trợ tái kỹ nghệ
  - Mức mở rộng cần thiết của việc chuyển đổi dữ liệu
  - Những nhân viên có kỹ năng về tái kỹ nghệ hệ thống

Chương 9

Quản lý dự án

Giới thiệu

- Để đảm bảo một dự án xây dựng hệ thống phần mềm thành công, một hoạt động không thể thiếu được đó là quản lý dự án.

Mục tiêu

- Hiểu thế nào là quản lý dự án phần mềm,
- Sự khác biệt so với việc quản lý các loại dự án khác.
- Nhiệm vụ của người quản lý dự án phần mềm là gì?
- Phải biết cách lập kế hoạch và xây dựng lịch biểu cho dự án
- Phải biết được những loại rủi ro nào có thể xảy ra trong quá trình thực hiện dự án và cách khắc phục chúng.

Giới thiêu về quản lý dự án

- Quản lý dự án phần mềm là một phần quan trọng của công nghệ phần mềm. Nếu quản lý tồi thì dự án sẽ thất bại.
- Dự án thất bại khi phần mềm chuyển giao chậm hơn so với kế hoạch, chi phí lớn hơn dự tính, và không thoả mãn các yêu cầu đề ra.

Giới thiệu về quản lý dự án

- Quản lý dự án phần mềm có liên quan tới những hoạt động nhằm đảm bảo chuyển giao phần mềm đúng thời hạn, đúng kế hoạch và phù hợp với các yêu cầu của tổ chức phát triển phần mềm.

Giới thiệu (tt)

- Quản lý dự án phần mềm có một số đặc trưng khác biệt so với các loại dự án khác:

- Sản phẩm là vô hình. Sản phẩm có khả năng thay đổi linh động.
- Quy trình phát triển phần mềm không phải là duy nhất.
- Nhiều dự án phần mềm là những dự án chỉ làm một lần.

Các hoạt động quản lý

- Các hoạt động quản lý dự án bao gồm:

- Viết kế hoạch dự kiến: Mô tả mục tiêu của dự án, phương pháp thực hiện, ước lượng thời gian và chi phí …

- Lập kế hoạch dự án: liên quan đến việc xác định các hành động, các mốc thời gian và các sản phẩm được tạo ra.

Các hoạt động quản lý

- Tính chi phí dự án

- Điều hành và xem xét lại dự án: người quản lý phải giám sát quy trình thực hiện dự án, so sánh quy trình và chi phí thực tế với kế hoạch đã định. Nếu điều hành tốt, người quản lý dự án có thể phát hiện và khắc phục được những rủi ro tiềm tàng.

Các hoạt động quản lý (tt)

- Lựa chọn và đánh giá cá nhân: Khi lựa chọn đội dự án, người quản lý dự án có thể gặp phải một số vấn đề sau:
  - Ngân sách của dự án không đủ để trả cho những nhân viên có mức lương cao
  - Không có được những nhân viên có kinh nghiệm và trình độ thích hợp
  - Tổ chức muốn chỉ định một số nhân viên mới tham gia vào dự án
- Viết báo cáo và trình bày.

Các hoạt động quản lý (tt)

- Lập kế hoạch dự án

- Lập kế hoạch dự án có thể là hoạt động tốn nhiều thời gian nhất trong quá trình quản lý dự án. Nó liệt kê các hành động từ pha khởi tạo cho đến khi đưa ra được hệ thống.

- Kế hoạch phải được theo dõi thường xuyên, nhất là khi có những thông tin hoặc những yêu cầu mới xuất hiện.

Các hoạt động quản lý (tt)

Cấu trúc của bản kế hoạch dự án gồm:

- - Phần giới thiệu: mô tả các mục tiêu của dự án và các ràng buộc gây ảnh hưởng tới việc quản lý dự án.
- - Tổ chức dự án: mô tả cách tổ chức của đội dự án, bao gồm những ai và những nhiệm vụ gì.

- Phân tích rủi ro: mô tả những rủi ro có thể xảy ra, dự báo khi nào chúng xảy ra và đề xuất chiến lược giảm rủi ro.

Các hoạt động quản lý (tt)

- Các yêu cầu về tài nguyên phần cứng và phần mềm: xác định những phần cứng và phần mềm nào cần thiết cho quá trình thực hiện dự án.

- Bảng thống kế công việc: xác định các công việc, từng mốc thời gian và kết quả của từng công việc.

- Lịch biểu của dự án: lịch biểu cho thấy sự phụ thuộc giữa các hành động, thời gian ước tính và phân công công việc cho từng người.

Các hoạt động quản lý (tt)

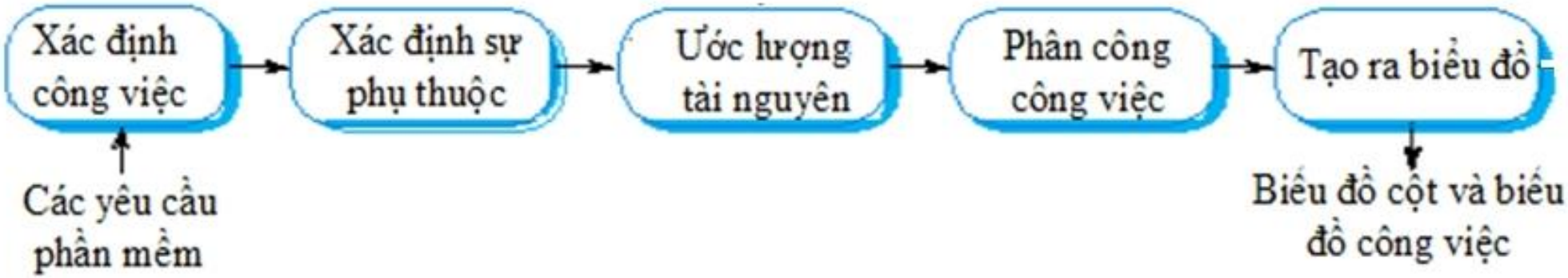
- Lịch biểu của dự án

- Lập lịch biểu dự án: Người quản lý phải chia dự án thành nhiều nhiệm vụ, ước lượng thời gian và tài nguyên cần thiết để hoàn thành từng nhiệm vụ.

- Khi lập lịch biểu, người quản lý nên tổ chức các công việc song song để sử dụng tối ưu lực lượng lao động và tối thiểu hoá sự phụ thuộc lẫn nhau giữa các nhiệm vụ để tránh sự chậm trễ khi một nhiệm vụ phải đợi nhiệm vụ khác hoàn thành.

Các hoạt động quản lý (tt6)

- Lịch biểu của dự án (tt)



Quy trình lập lịch biểu dự án

Các hoạt động quản lý (tt)

- Lịch biểu của dự án (tt)

  - Chất lượng của lịch biểu phụ thuộc vào hiểu biết và kinh nghiệm của người quản lý.

  - Chúng ta sử dụng các ký pháp đồ hoạ để minh hoạ cho lịch biểu của dự án. Sử dụng biểu đồ giúp ta thấy rõ cách chia dự án thành nhiều nhiệm vụ. Các nhiệm vụ không nên quá nhỏ, chúng nên được thực hiện với đơn vị nhỏ nhất là ngày.

Quản lý rủi ro

- Quản lý rủi ro liên quan tới việc xác định rủi ro và lập ra các kế hoạch để tối thiểu hoá ảnh hưởng của chúng tới dự án.
  - Để quản lý rủi ro, chúng ta cần phải thực hiện các hoạt động sau:

    - Phát hiện rủi ro: Phát hiện các loại rủi ro có liên quan đến: công nghệ, con người, tổ chức, các yêu cầu.

    - Phân tích rủi ro: Đánh giá các khả năng xảy ra rủi ro và tính nghiêm trọng của nó nếu nó xảy ra.

Quản lý rủi ro (tt)

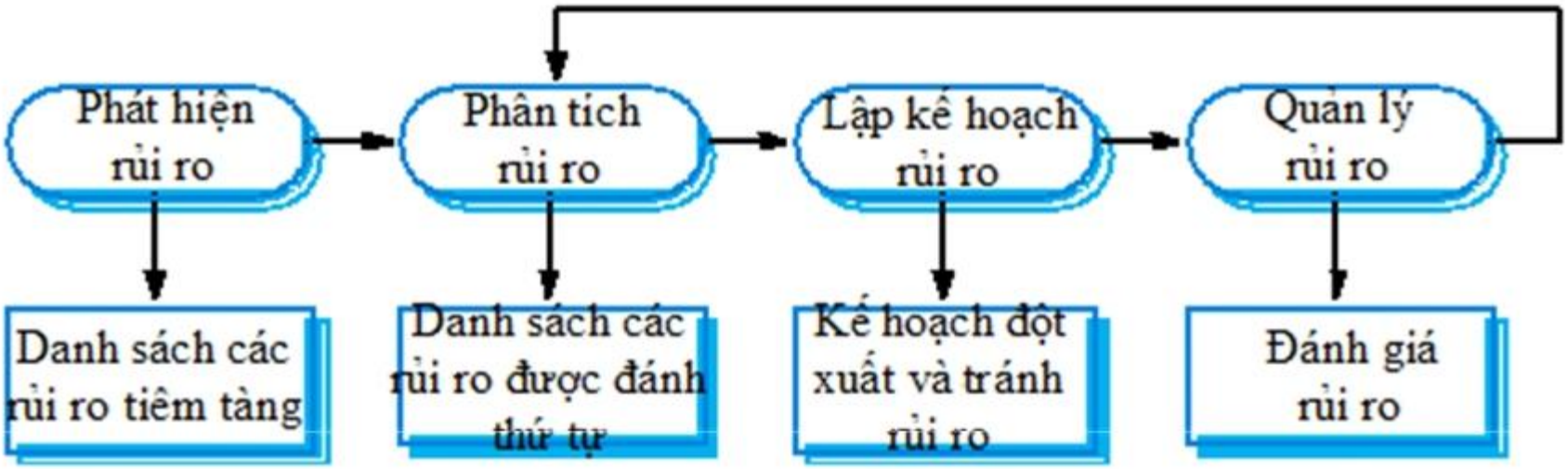
-Lập kế hoạch rủi ro:

Xem xét từng rủi ro và phát triển chiến lược để quản lý nó. Bao gồm các chiến lược như: phòng tránh giảm khả năng xảy ra rủi ro, tối thiểu hoá giảm ảnh hưởng của rủi ro.

-Kiểm soát rủi ro:

Đánh giá từng rủi ro đã được xác định một cách thường xuyên để xác định khả năng nó có thể xảy ra hay không đồng thời đánh giá mức độ ảnh hưởng của nó.

Quản lý rủi ro (tt)



Quy trình quản lý rủi ro

Bài tập

Câu 1: Tại sao một lập trình viên tốt chưa chắc đã là một người quản lý tốt?

Câu 2: Giải thích mục đích của từng công việc trong quá trình lập kế hoạch dự án.

Câu 3: Đội dự án gồm 4 người (A, B, C và D) có nhiệm vụ hoàn thành phần mềm chơi cờ ca-rô trong 10 tuần. Giả sử mỗi người phải làm nhiều hơn 10 tiếng/tuần. Sản phẩm có hai lần chuyển giao. (Lần thứ nhất là sau 4 tuần).

Lập kế hoạch cho dự án trên

Bài tập (tt1)

Lần thứ nhất		Phân công	Thời hạn
	Xác định yêu cầu người dùng	A, B, C, D	Tuần 1
	Viết mã lệnh cài đặt trò chơi	A, B	Tuần 4
	Viết mã lệnh để xây dựng bàn cờ	B, C	Tuần 4
	Viết mã lệnh cho nước cờ	D	Tuần 4
Lần thứ hai	Viết mã lệnh cho giao diện	A, B, C, D	Tuần 10
	Viết bản trợ giúp (help)	D	Tuần 10
	Kiểm thử hệ thống		Tuần 10

Bài tập (tt2)

Lần thứ nhất (số giờ/người)	A	B	C	D
Xác định yêu cầu người dùng	2	2	2	2
Viết đặc tả yêu cầu người dùng				3
Viết mã lệnh cài đặt trò chơi	20	10		
Viết mã lệnh để xây dựng bàn cờ		10	20	
Viết mã lệnh cho nước cờ				25
Kiểm tra lại bản thiết kế	2	2	2	2
Tích hợp	5	10		
Kiểm thử	5	3	3	3
Họp	2	2	2	2

Bài tập (tt3)

Lần thứ hai	Người thực hiện	Tổng số giờ
Xem xét lại pha 1	A, B, C	30
Viết giao diện	A, B, C, D	120
Viết bản trợ giúp (help)	D	20
Kiểm thử hệ thống	B, C, D	20
Họp	A, B, C, D	12

Bài tập (tt4)

Nhiệm vụ	Phụ thuộc	Thời gian	Bắt đầu	Kết thúc
a		10		
b	e	10		
c	d, f	10		
d	a, f, b	20		
e	a, f	8		
f	a	5		

- Cho tập hợp các nhiệm vụ, ràng buộc và thời gian thực hiện như trên. Hãy vẽ mô hình mạng của các nhiệm vụ đó.