



# Chương 1: Tổng quan về CNPM



- ❖ Một số khái niệm cơ bản
- ❖ Công nghệ phần mềm
- ❖ Quy trình công nghệ phần mềm:
  - Quy trình thác nước
  - Quy trình thác nước cải tiến
  - Quy trình Prototype
  - Quy trình xoắn ốc



# Khái niệm về phần mềm

- ❖ Ví dụ: xét một số phần mềm sau:
  - Phần mềm quản lý học sinh cấp 3.
  - Phần mềm quản lý thư viện.
  - Phần mềm quản lý nhà sách.
  - Phần mềm quản lý khách sạn.
  - Phần mềm quản lý phòng mạch tư.
  - Phần mềm quản lý sổ tiết kiệm.
  - Phần mềm quản lý bán vé chuyến bay.
  - Phần mềm xếp thời khóa biểu
  - ...



# Khái niệm về phần mềm

## Phần mềm là gì?



- ❖ Được xem xét ở hai góc độ:
  - Góc nhìn **người dùng**
  - Góc nhìn **chuyên viên tin học**



# Khái niệm về phần mềm

- ❖ Phần mềm dưới **góc nhìn của người sử dụng**:
  - **Chương trình** thực thi được trên **máy tính** hoặc **các thiết bị chuyên dụng khác**
  - Nhằm **hỗ trợ** cho các nhà chuyên môn trong từng lĩnh vực chuyên ngành thực hiện **tốt hơn** các thao tác nghiệp vụ của mình

# Khái niệm về phần mềm

- ❖ Môi trường triển khai phần mềm:
  - **Máy tính:** Desktop, Laptop, Tablet PC...
  - **Thiết bị chuyên dụng:**
    - Thiết bị di động: PDA, Pocket PC, ĐTDĐ
    - Các thiết bị chuyên dụng khác: set-top box, router,...
- ❖ Hỗ trợ làm tốt hơn các thao tác nghiệp vụ:
  - Tin học hóa nghiệp vụ hiện **đang làm thủ công**
  - Cải tiến chức năng nghiệp vụ hiện **đang được thực hiện trên máy tính**
  - Đề ra, xây dựng và triển khai chức năng nghiệp vụ mới

# Khái niệm về phần mềm

- ❖ Phần mềm dưới góc nhìn của **chuyên viên Tin học**:
  - Đây là một hệ thống bao gồm **3 thành phần** cơ bản:
    - Thành phần giao tiếp
    - Thành phần xử lý
    - Thành phần lưu trữ
  - Cần được xây dựng để thực hiện theo yêu cầu của người sử dụng

# Khái niệm về phần mềm

## ❖ Thành phần giao tiếp (Giao diện):

- Cho phép **tiếp nhận** các yêu cầu về việc sử dụng phần mềm từ người sử dụng, từ các thiết bị thu thập dữ liệu, hoặc từ các phần mềm khác.
- Cho phép **trình bày** các kết quả của việc thực hiện các yêu cầu cho người dùng (kết quả của công việc khi thực hiện trên máy tính) hoặc điều khiển hoạt động các thiết bị điều khiển (đóng/mở cửa, dừng hay cho chuyển động...)



# Khái niệm về phần mềm

## ❖ Thành phần giao tiếp (Giao diện):

- Một cách tổng quát, thành phần giao tiếp cho phép **nhập/xuất thông tin** cùng với hình thức trình bày/giao tiếp tương ứng.
- Mục tiêu chính của thành phần này là đưa thông tin từ thế giới thực bên ngoài phần mềm (người sử dụng, các thiết bị, phần mềm khác...) vào bên trong, hoặc ngược lại.

# Khái niệm về phần mềm

## ❖ Thành phần xử lý:

- **Kiểm tra tính hợp lệ** của thông tin nguồn được cung cấp từ người dùng theo các quy định ràng buộc trong thế giới thực
  - Ví dụ: chỉ cho mượn tối đa 3 quyển sách, mỗi lớp học không quá 50 học sinh...
- **Tiến hành xử lý** cho ra kết quả mong đợi theo **quy định** tính toán có sẵn **trong thế giới thực**
- Hoặc **tiến hành xử lý** theo **thuật giải tự đề xuất**

# Khái niệm về phần mềm

## ❖ Thành phần xử lý (tt):

- Việc xử lý dựa trên thông tin nguồn từ **người sử dụng cung cấp**
  - Ví dụ: tính nghiệm phương trình bậc 2 dựa trên các hệ số nhập vào hoặc **dữ liệu lưu trữ có sẵn**
  - Ví dụ: tính tiền phạt dựa trên ngày trả sách được nhập vào và thông tin về loại sách đã được lưu trữ
- Việc xử lý cho ra **kết quả** có thể dùng để **xuất cho người dùng xem qua thành phần giao diện**, hay **lưu trữ lại qua thành phần lưu trữ**, hoặc **cả hai**



# Khái niệm về phần mềm

- ❖ Một cách tổng quát, thành phần xử lý là hệ thống chuyên xử lý tính toán, biến đổi dữ liệu.
  - Dùng thông tin nguồn từ thành phần giao diện (**chức năng nhập**) hay thành phần dữ liệu (**chức năng đọc**);
  - Kiểm tra tính hợp lệ (**chức năng kiểm tra**) và sau đó tiến hành xử lý (**chức năng xử lý**) – nếu cần thiết
  - Để cho ra kết quả sẽ được trình bày thông qua thành phần giao diện (**chức năng xuất**) hoặc lưu trữ lại trong thành phần dữ liệu (**chức năng ghi**)

# Khái niệm về phần mềm

## ❖ Thành phần lưu trữ (thành phần dữ liệu)

- Cho phép lưu trữ lại (**chức năng ghi**) các kết quả đã xử lý
  - Ví dụ: Việc mượn sách đã được kiểm tra hợp lệ, bảng lương tháng đã được tính trên bộ nhớ phụ với tổ chức lưu trữ được xác định trước
  - Ví dụ: tập tin có cấu trúc, tập tin nhị phân, cơ sở dữ liệu

# Khái niệm về phần mềm

## ❖ Thành phần lưu trữ (thành phần dữ liệu)

- Cho phép truy xuất lại (**chức năng đọc**) các dữ liệu đã lưu trữ phục vụ cho các hàm xử lý tương ứng
- Một cách tổng quát thành phần dữ liệu là hệ thống chuyên **đọc ghi** dữ liệu cùng với mô hình tổ chức lưu trữ dữ liệu tương ứng.
- Mục tiêu chính của thành phần này là **chuyển đổi dữ liệu** giữa bộ nhớ chính và bộ nhớ phụ

# Khái niệm lớp phần mềm

## ❖ Phân lớp theo mục đích sử dụng:

- Lớp phần mềm quản lý học sinh
- Lớp phần mềm đánh cờ
- Lớp phần mềm giải bài tập
- ...

## ❖ Phân lớp theo mục tiêu tạo lập

- Lớp phần mềm theo hợp đồng
- Lớp phần mềm khung
- Lớp phần mềm đóng gói

# Khái niệm lớp phần mềm

## ❖ Lớp phần mềm:

- Lớp phần mềm là hệ thống các phần mềm trên cùng một lĩnh vực hoạt động nào đó
- Do cùng lĩnh vực hoạt động nên các phần mềm cùng lớp thường có cấu trúc và chức năng tương tự nhau

## ❖ Mục tiêu của ngành Công nghệ Phần mềm

- Xây dựng được phần mềm có chất lượng
- Dễ dàng xây dựng phần mềm mới từ các phần mềm có sẵn cùng lớp

# Chất lượng phần mềm

Tính đúng đắn:

- Đầy đủ
- Chính xác

**Tính đúng đắn**



**Phần mềm**



**Phần mềm khác**



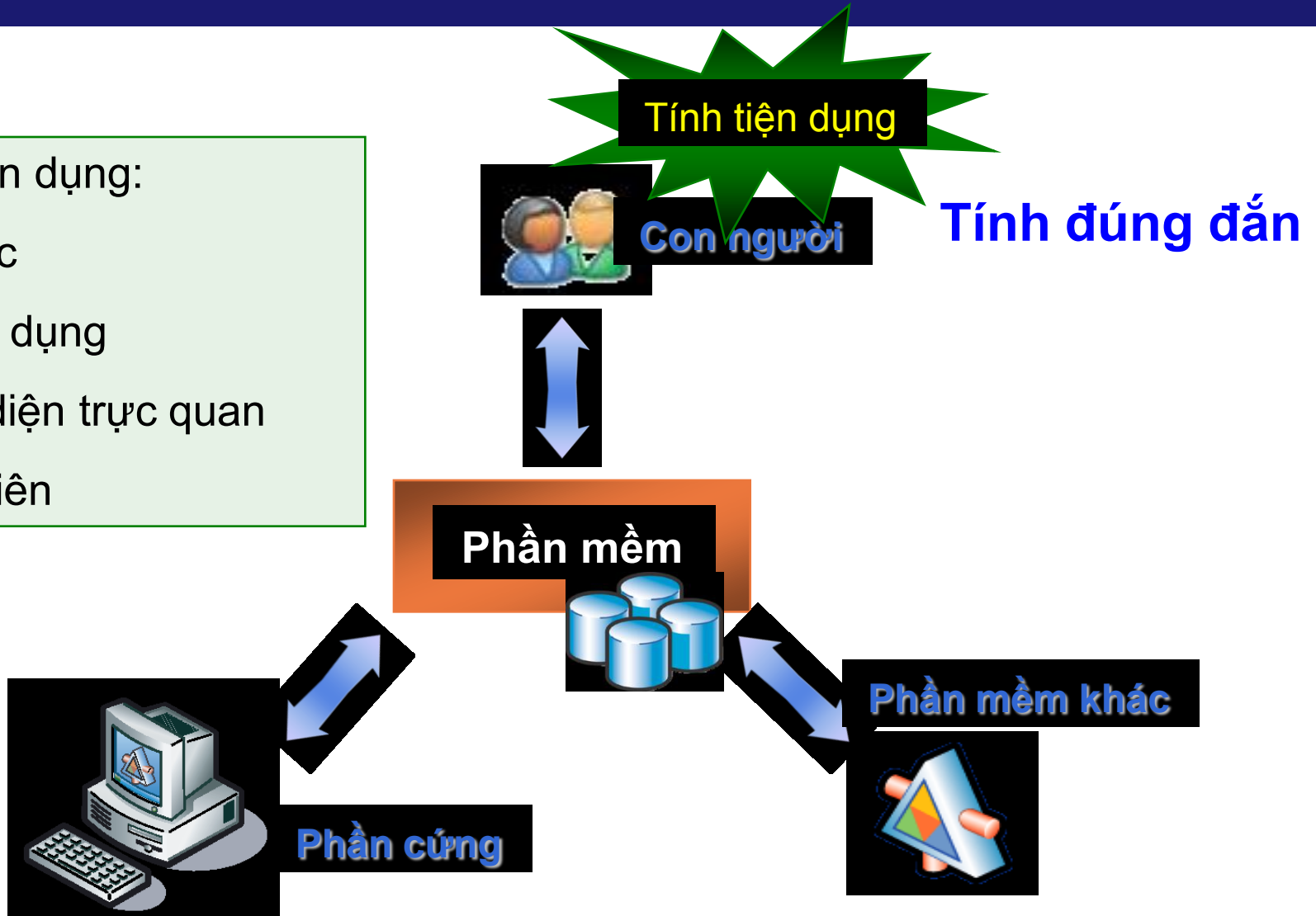
**Phần cứng**

Dưới góc nhìn của Người sử dụng

# Chất lượng phần mềm

Tính tiện dụng:

- Dễ học
- Dễ sử dụng
- Giao diện trực quan
- Tự nhiên



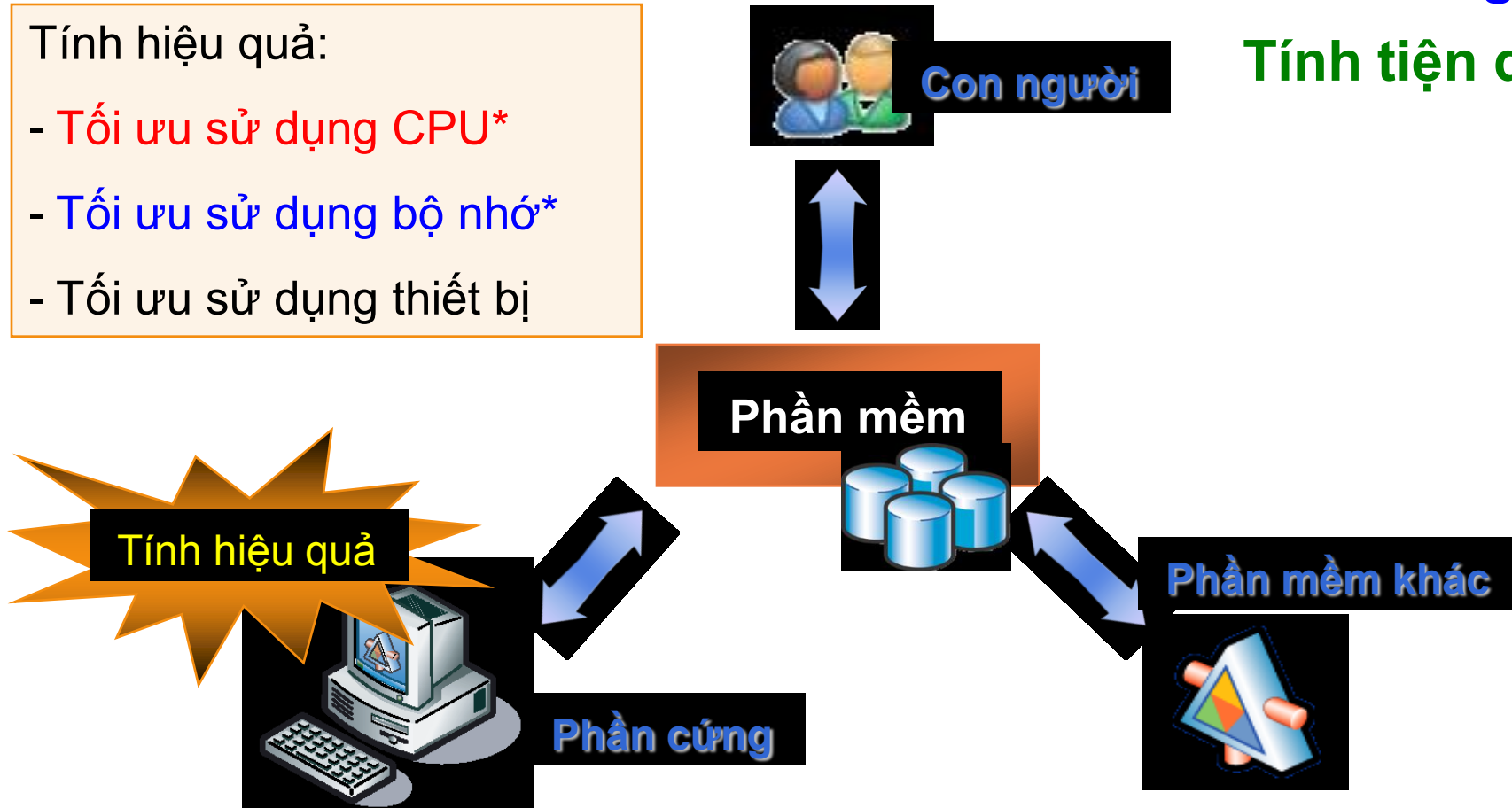
Dưới góc nhìn của Người sử dụng

# Chất lượng phần mềm

Tính hiệu quả:

- Tối ưu sử dụng CPU\*
- Tối ưu sử dụng bộ nhớ\*
- Tối ưu sử dụng thiết bị

Tính đúng đắn  
Tính tiện dụng



Dưới góc nhìn của Người sử dụng

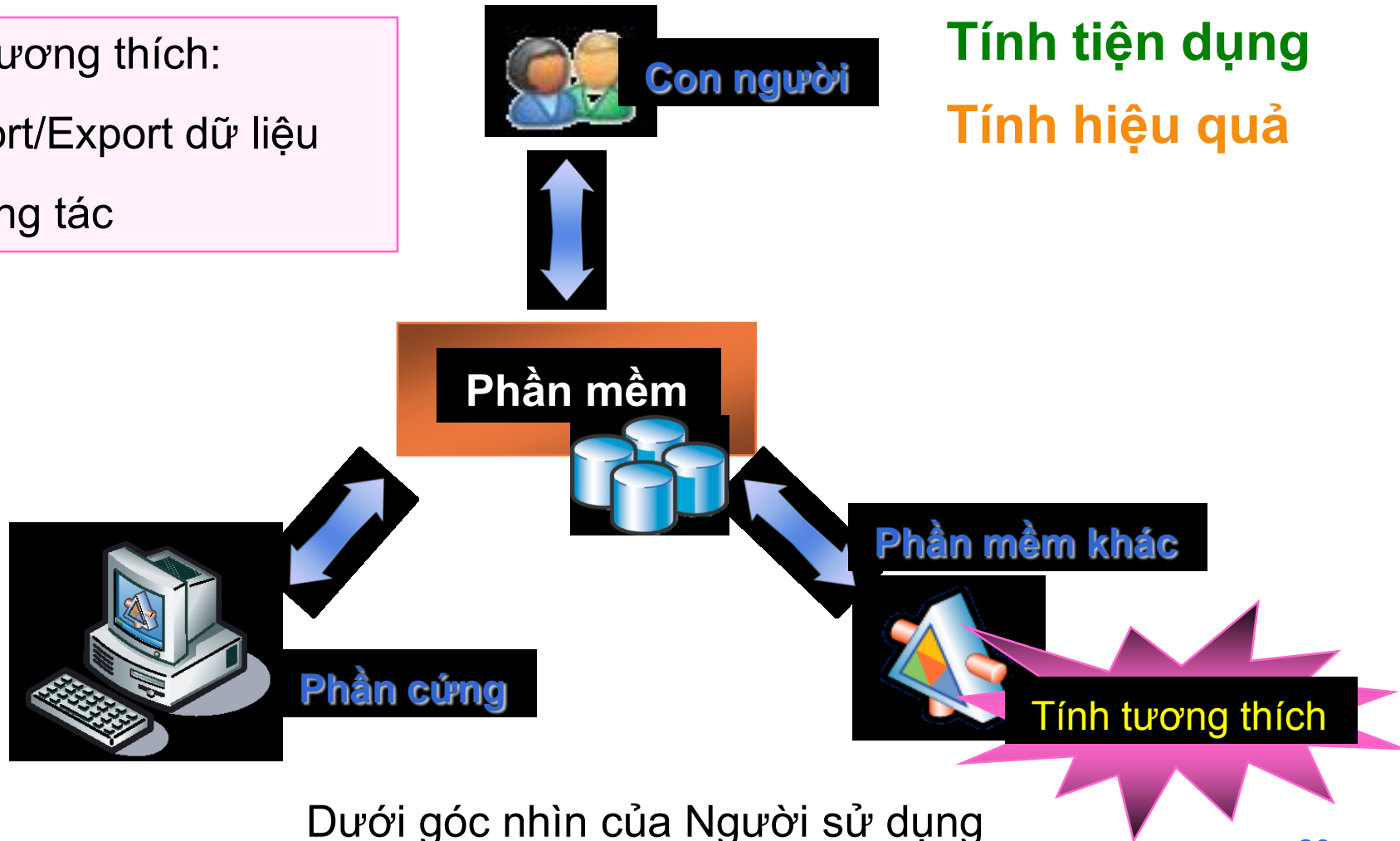


# Chất lượng phần mềm

Tính tương thích:

- Import/Export dữ liệu
- Tương tác

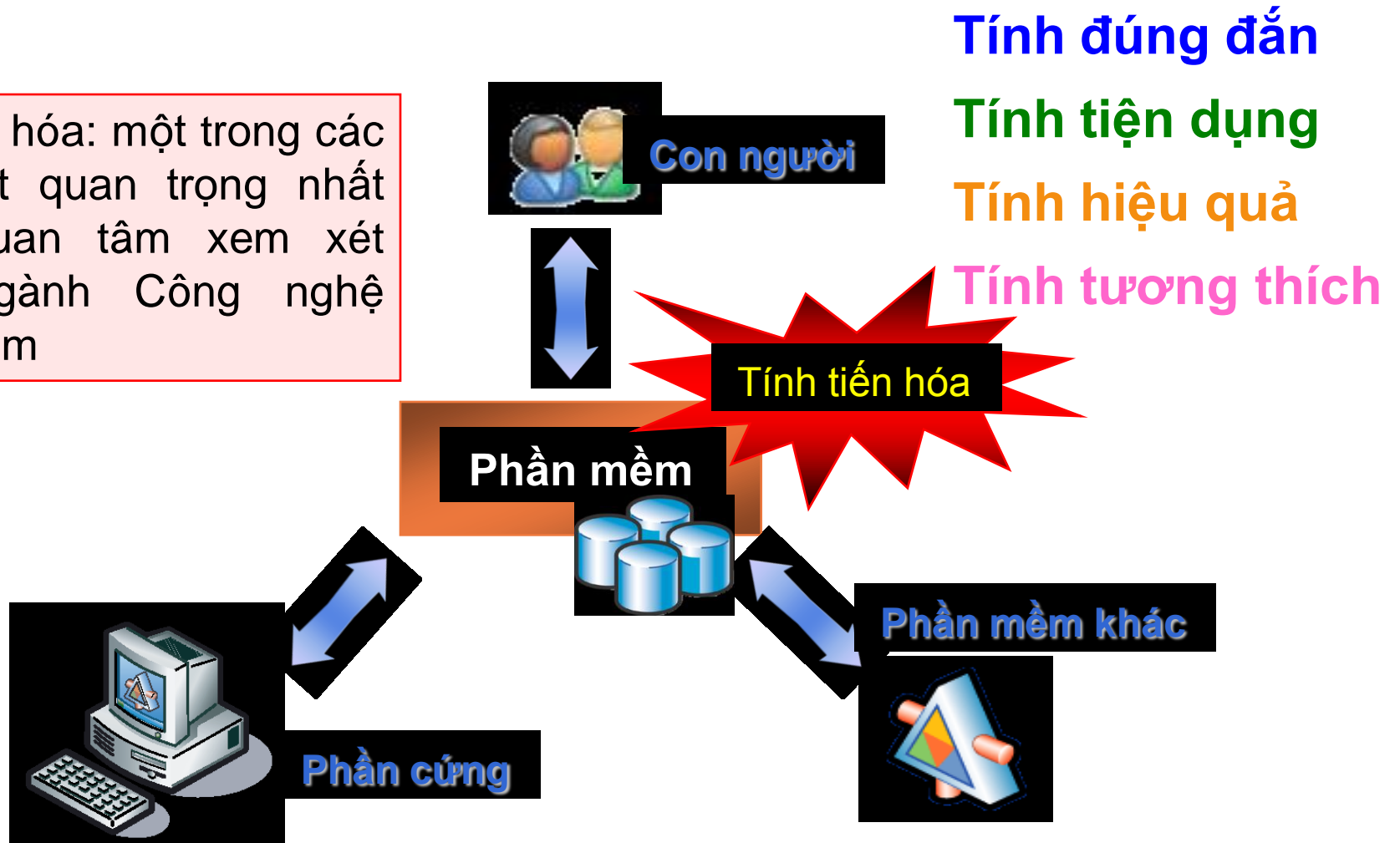
Tính đúng đắn  
Tính tiện dụng  
Tính hiệu quả



Dưới góc nhìn của Người sử dụng

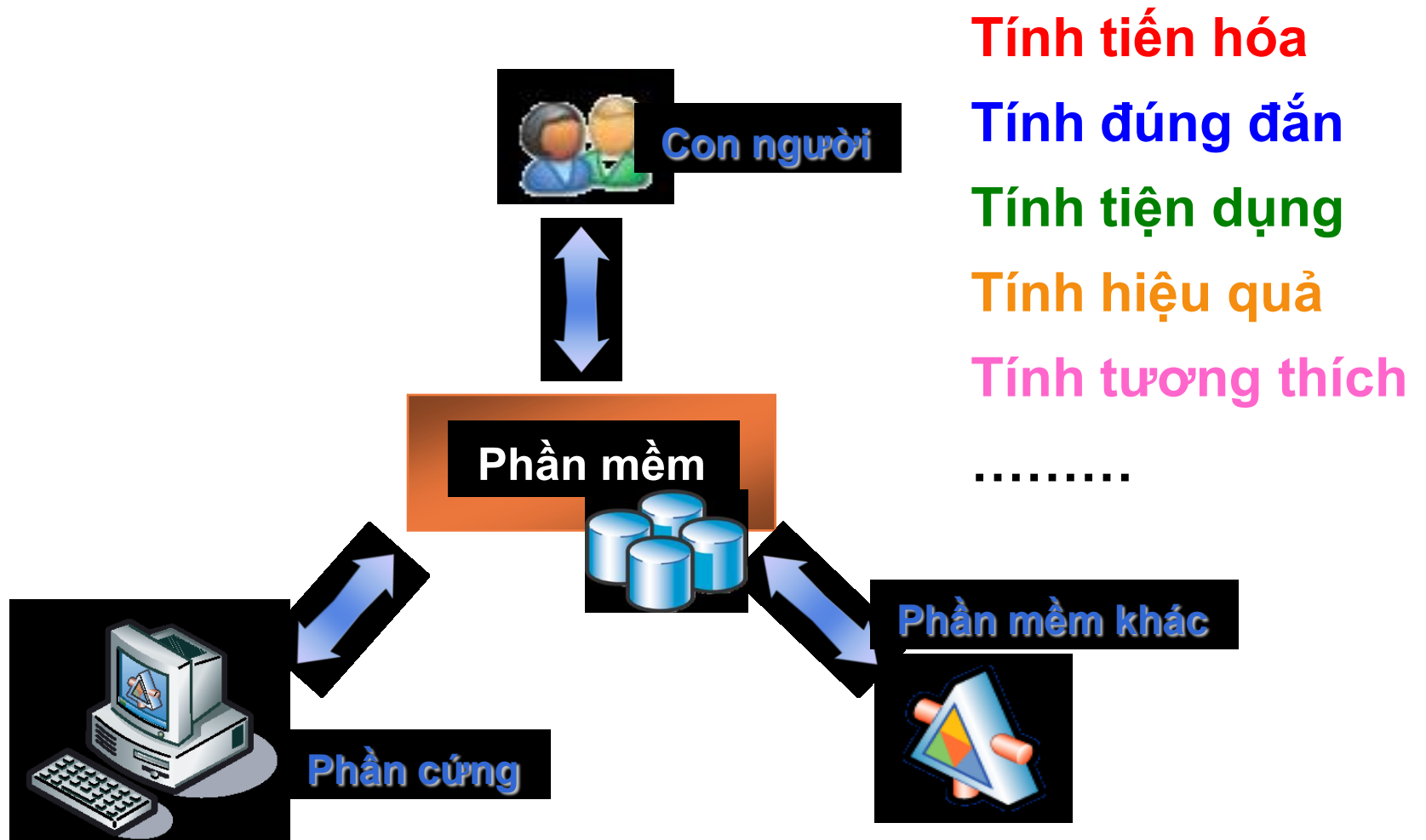
# Chất lượng phần mềm

Tính tiến hóa: một trong các tính chất quan trọng nhất được quan tâm xem xét trong ngành Công nghệ Phần mềm



Dưới góc nhìn của Người sử dụng

# Chất lượng phần mềm



# Chất lượng phần mềm

❖ Tính dễ kiểm tra: việc kiểm tra các thành phần phù hợp với yêu cầu phần mềm là dễ dàng nhất có thể được

❖ Tính dễ sửa lỗi: khi có sự không phù hợp (so với yêu cầu) trong quá trình kiểm tra một thành phần, việc phát hiện chính xác “vị trí lỗi” và sửa lỗi là nhanh nhất có thể được.

❖ Tính dễ bảo trì: khi cần nâng cấp, cải tiến một thành phần (theo yêu cầu mới), việc cập nhật phần mềm là nhanh, chính xác nhất có thể được và đặc biệt là cố gắng hạn chế ảnh hưởng đến các thành phần khác

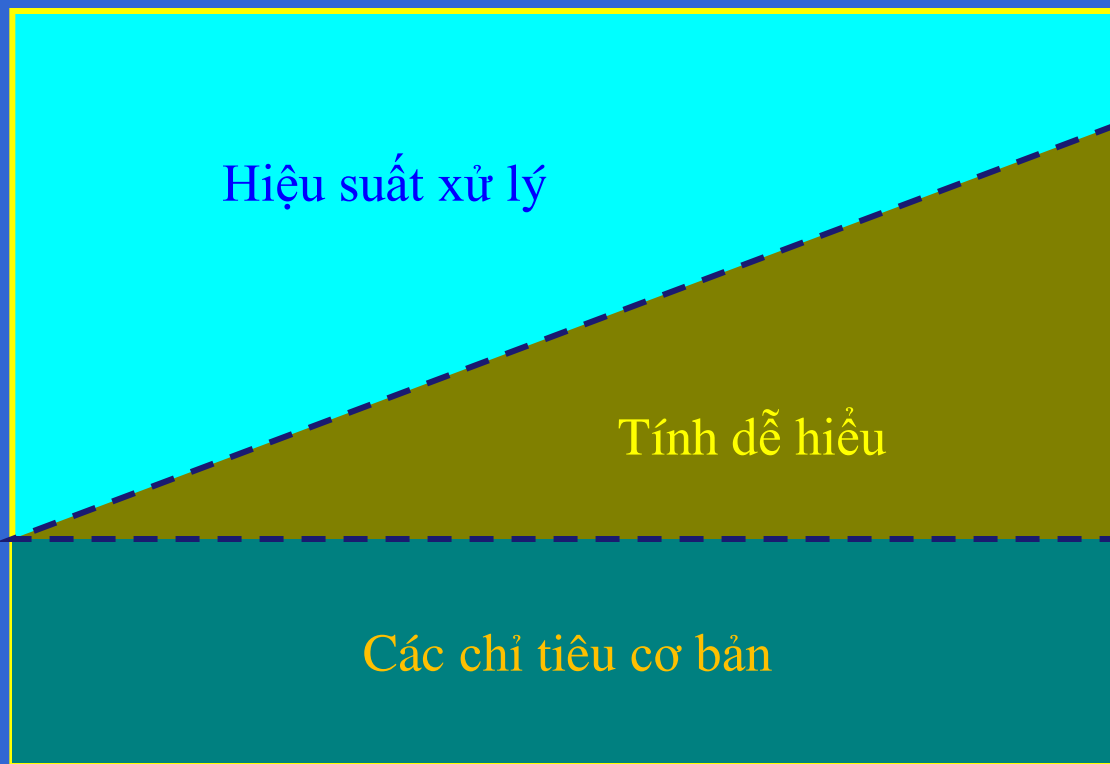
❖ Tính tái sử dụng: các thành phần đã thực hiện có thể dùng lại trong các phần mềm cùng lớp (hoặc cùng lĩnh vực) với thời gian và công sức ít nhất có thể được

.....

Dưới góc nhìn của chuyên viên Tin học



Yếu  
tố  
khái  
niệm  
phần  
mềm  
tốt



Đặc  
trung  
gần  
đây

Thời gian  
(Phần cứng phát triển)



# Phát triển của Công nghệ phần mềm

- ❖ Giai đoạn 1 (1950 – giữa 1960)
  - Xử lý theo lô, xử lý tập trung, ít xử lý phân tán, ít sửa đổi phần mềm
- ❖ Giai đoạn 2 (từ giữa 1960 đến giữa 1970)
  - Hệ thống đa chương trình và đa người dùng
  - Bắt đầu cuộc “khủng hoảng” phần mềm
- ❖ Giai đoạn 3 (từ giữa 1970 đến giữa 1980)
  - Sự phát triển và sử dụng rộng rãi máy tính cá nhân
  - Sự phát triển của các công ty phần mềm
- ❖ Giai đoạn 4 (từ giữa 1980 đến nay)
  - Phần cứng ngày càng phát triển
  - Hệ thống phần mềm ngày càng đa dạng, phong phú, xử lý ngày càng phức tạp, công nghệ ngày càng phát triển...



# Cuộc khủng hoảng phần mềm

- ❖ Số lượng các phần mềm tăng vọt (do sự phát triển của phần cứng: tăng khả năng, giá thành hạ)
- ❖ Có quá nhiều **khuyết điểm** trong các phần mềm được dùng trong xã hội:
  - Thực hiện **không đúng yêu cầu** (tính toán sai, không ổn định...)
  - Thời gian **bảo trì** nâng cấp quá lâu, chi phí cao, hiệu quả thấp
  - Khó sử dụng, Thực hiện chậm
  - Không chuyển đổi dữ liệu giữa các phần mềm
  - ...

# Cuộc khủng hoảng phần mềm

## ❖ Một số kết luận:

- Việc tăng vọt số lượng phần mềm là điều hợp lý và sẽ còn tiếp diễn
- Các khuyết điểm của phần mềm có nguồn gốc chính từ **phương pháp, cách thức và quy trình** tiến hành xây dựng phần mềm:
  - **Cảm tính**: mỗi người theo một phương pháp riêng
  - **Thô sơ, đơn giản**: chỉ tập trung vào việc lập trình mà ít quan tâm đến các công việc cần làm khác (khảo sát hiện trạng, phân tích yêu cầu, thiết kế...)
  - **Thủ công**: còn thiếu các công cụ hỗ trợ quy trình phát triển



# Công nghệ phần mềm

## ❖ Khái niệm:

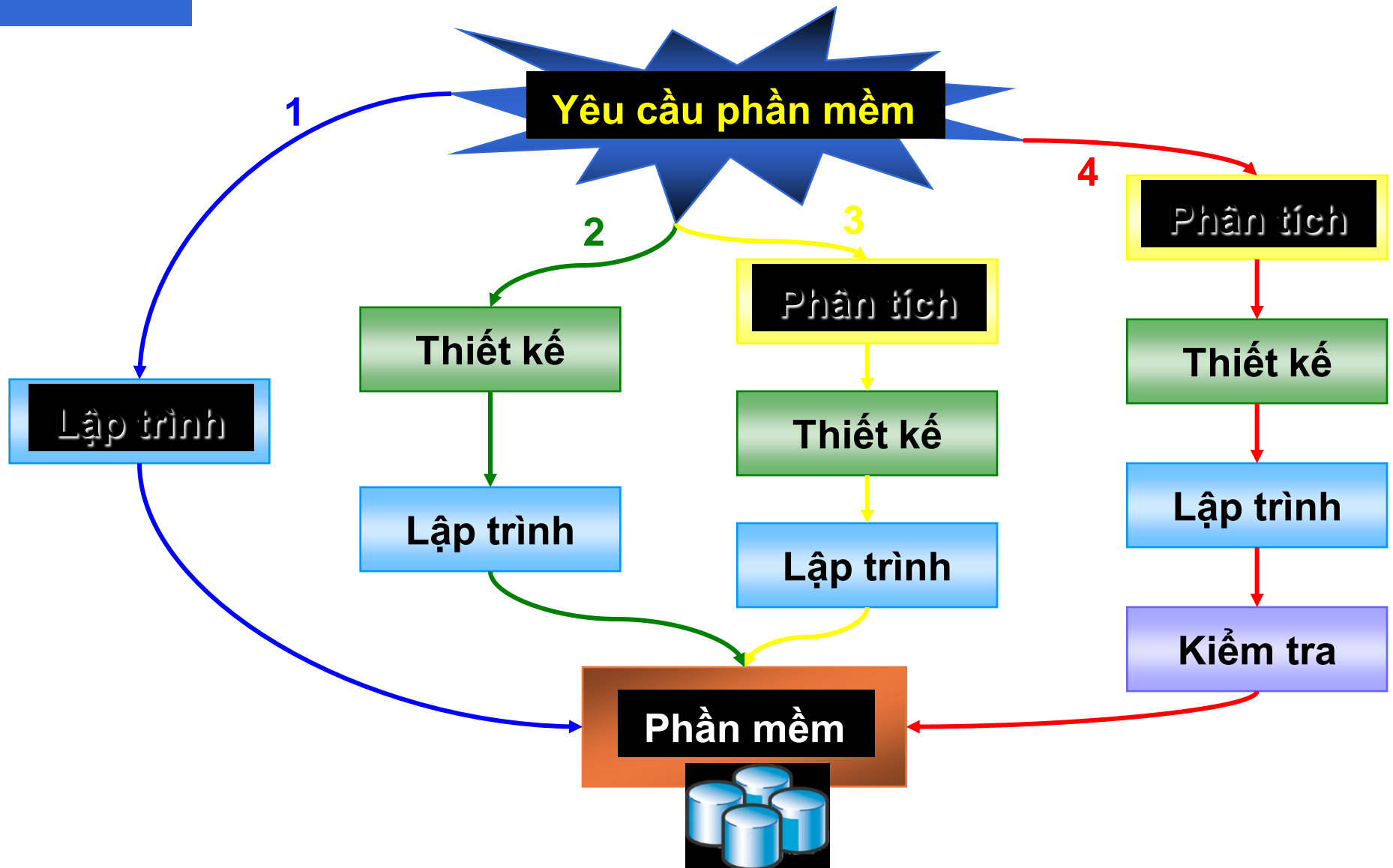
- Công nghệ phần mềm là ngành khoa học nghiên cứu về việc xây dựng các phần mềm có chất lượng cao trong thời gian và chi phí thực hiện hợp lý



- ❖ Các đối tượng nghiên cứu của Công nghệ phần mềm:
  - Quy trình công nghệ phần mềm:
    - Hệ thống các giai đoạn mà quá trình phát triển phần mềm phải trải qua, với mỗi giai đoạn cần xác định rõ:
      - Mục tiêu, kết quả nhận từ giai đoạn trước đó,
      - Kết quả chuyển giao cho giai đoạn kế tiếp
  - Phương pháp phát triển phần mềm:
    - Hệ thống các hướng dẫn cho phép từng bước thực hiện một giai đoạn nào đó trong quy trình phần mềm

- ❖ Các đối tượng nghiên cứu của Công nghệ phần mềm:
  - Công cụ và Môi trường phát triển phần mềm:
    - Hệ thống các phần mềm trợ giúp trong lĩnh vực xây dựng phần mềm
    - Hỗ trợ các chuyên viên tin học trong các bước xây dựng phần mềm theo một phương pháp nào đó với một quy trình được chọn trước

# Quy trình Công nghệ phần mềm

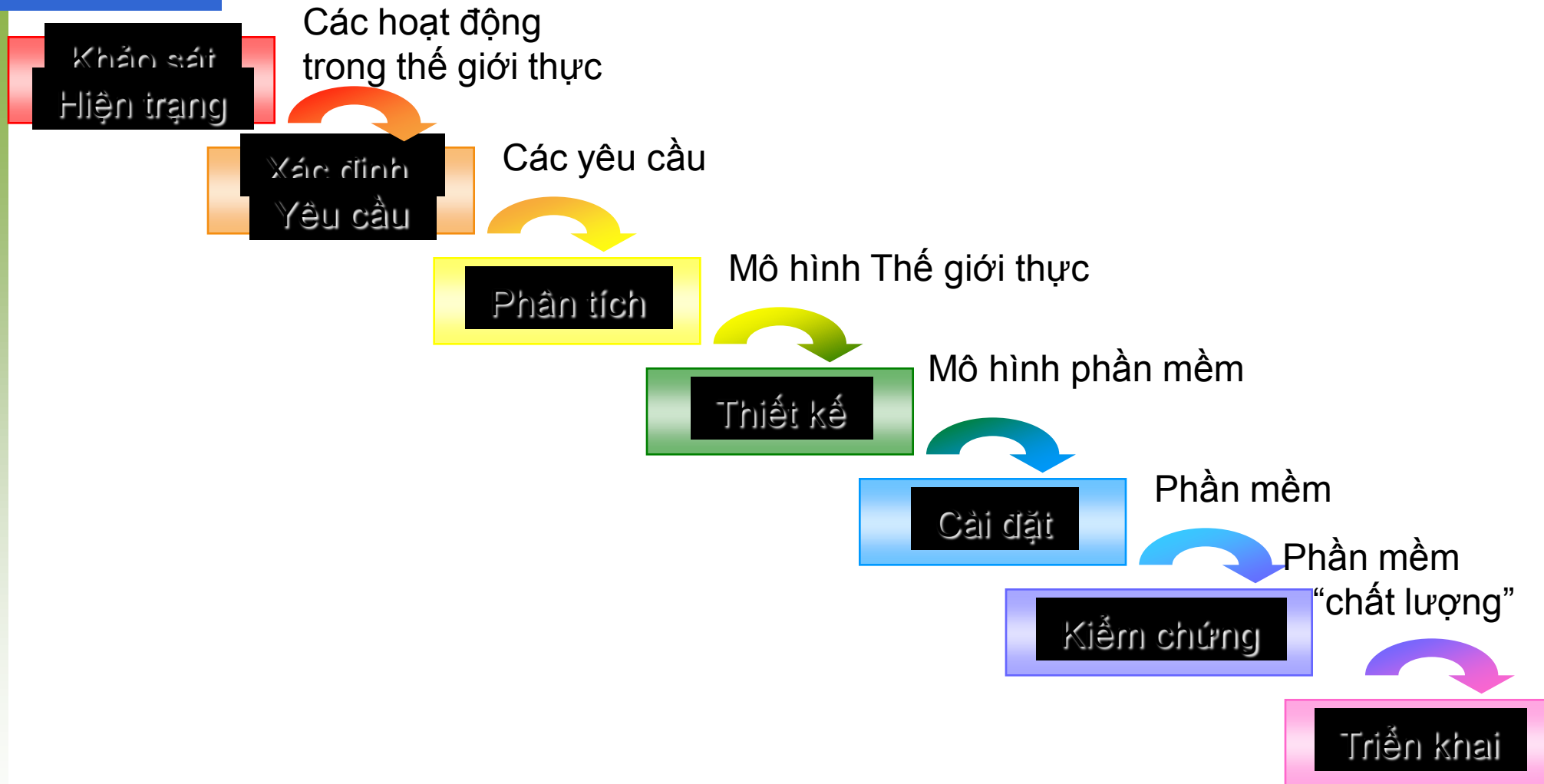




# Quy trình Công nghệ phần mềm

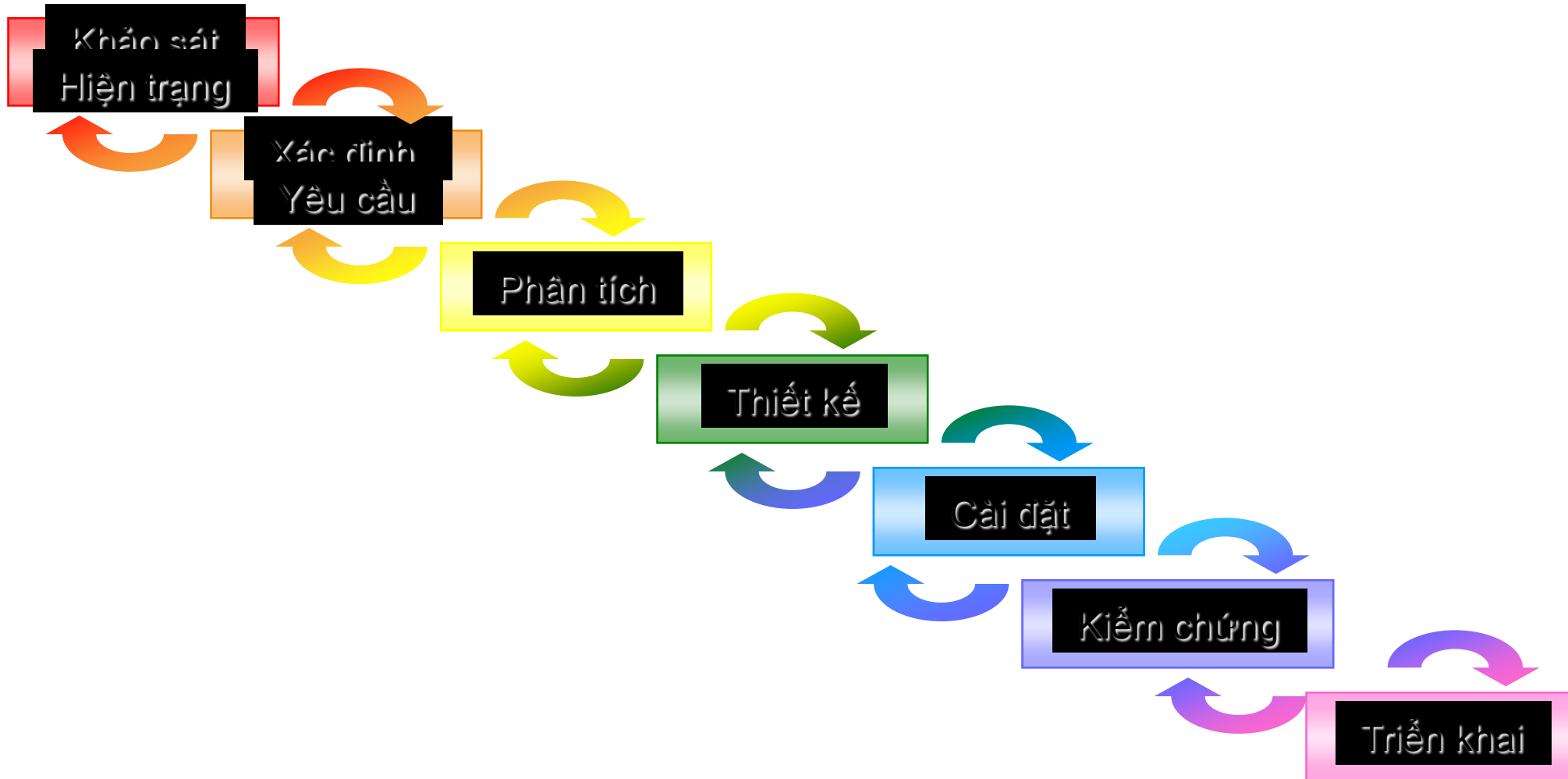
- ❖ **Phân tích:** Mô tả mức phát thảo các thành phần của phần mềm (đã có yêu cầu)
- ❖ **Thiết kế:** Mô tả mức chi tiết các thành phần của phần mềm (đã có yêu cầu)
- ❖ **Lập trình:** Thực hiện các thành phần của phần mềm (đã thiết kế)
- ❖ **Kiểm tra:** kiểm chứng các thành phần của phần mềm (đã thực hiện)

# Quy trình thác nước

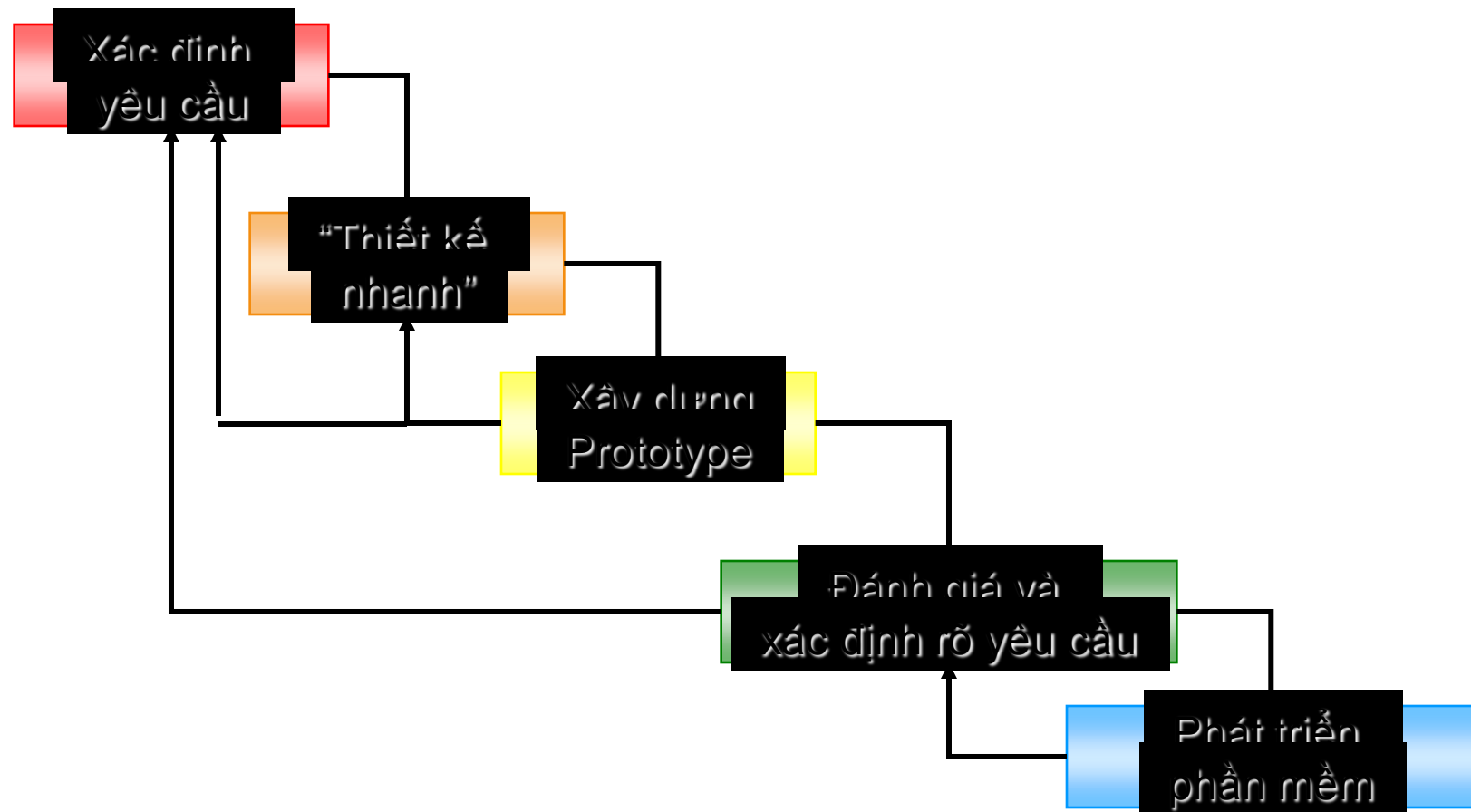


Waterfall

# Quy trình thác nước cải tiến

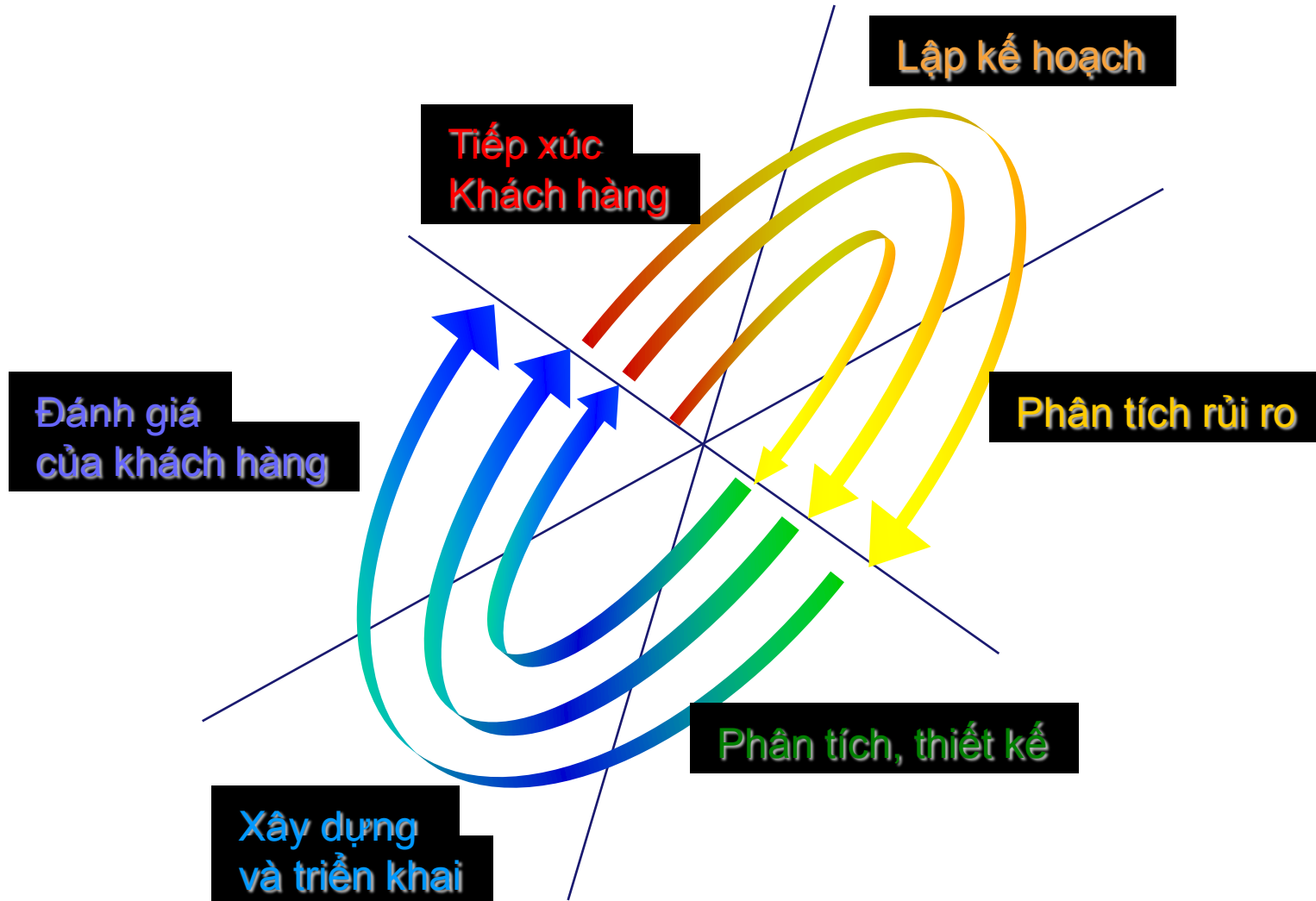


# Quy trình Prototype





# Quy trình xoắn ốc



# Bài tập

1. Xét phần mềm giải bài tập về đa thức. Giả sử chỉ xét chức năng tính đạo hàm một đơn thức  $P(x) = a.x^n$  (với  $a$  là số thực,  $n$  là số nguyên không âm), theo qui tắc tính đạo hàm  $Q(x) = P'(x) = n.a.x^{n-1}$
2. Viết chương trình giải phương trình bậc 2
3. Xét phần mềm quang hình học, giả sử chỉ xem xét chức năng xác định ảnh của một vật qua một thấu kính. Cho biết công thức xác định ảnh:  $1/d + 1/d' = 1/f$
4. Chức năng xác định hình chiếu điểm  $M$  trên đường thẳng  $d$



# Chương 2: Xác định và phân tích yêu cầu



- ❖ Giai đoạn khảo sát hiện trạng
  - Hiện trạng tổ chức
  - Hiện trạng nghiệp vụ
  - Hiện trạng Tin học (phần cứng, phần mềm, con người)
- ❖ Xác định và thu thập yêu cầu:
  - Phân loại yêu cầu:
    - Yêu cầu **chức năng**: Lưu trữ, tra cứu, tính toán, kết xuất,...
    - Yêu cầu **phi chức năng**: Yêu cầu bảo mật, Mã hóa dữ liệu, Sao lưu định kỳ, Phục hồi khi có sự cố,...

- ❖ Xác định và thu thập yêu cầu(tt):
  - Kỹ thuật thu thập yêu cầu:
    - Phỏng vấn
    - Bảng câu hỏi
    - Nghiên cứu các tài liệu
    - Quan sát thực tế
    - Phân tích thiết kế nhóm (JAD - Joint Application Design)
- ❖ Phân tích yêu cầu (Mô hình hóa yêu cầu)

## ❖ Hiện trạng tổ chức

- Đối nội: → Cơ cấu tổ chức nội bộ
  - Bản thân tổ chức là một hệ thống, có cơ cấu tổ chức
  - Sơ đồ cơ cấu tổ chức nội bộ
    - Cách nhìn tổng thể về 1 tổ chức
- Đối ngoại:
  - Tổ chức ⇔ Môi trường của tổ chức

## ❖ Hiện trạng nghiệp vụ

- Hiểu được quy trình nghiệp vụ: mục tiêu quan trọng nhất của khảo sát hiện trạng
- Có bao nhiêu nghiệp vụ, bao nhiêu quy trình?
  - Dưới góc nhìn của người làm quản lý, không phải của chuyên viên Tin học
- Nghiệp vụ được thực hiện như thế nào?
  - Các công đoạn, bộ phận liên quan...
- Tần suất? Thời điểm thực hiện



# Khảo sát hiện trạng

## ❖ Hiện trạng nghiệp vụ (tt)

- Khối lượng tác vụ/quyết định?
- Đánh giá nghiệp vụ hiện tại
  - Cần có những nhận xét của những người chuyên môn trong guồng máy công tác hiện tại
- Có vấn đề/khó khăn gì hiện tại hay không? Nguyên nhân?
  - Vấn đề/khó khăn độc lập với công nghệ, chỉ liên quan đến chuyên môn nghiệp vụ thì cần giải quyết ngay

## ❖ Hiện trạng Tin học

- Phần cứng:
  - Các thiết bị hiện tại,
  - Số lượng,
  - Cấu hình,
  - Vị trí (vật lý),
  - Tình hình kết nối mạng,
  - Loại kết nối...

## ❖ Hiện trạng Tin học (tt)

- Phần mềm:
  - Hệ điều hành
  - Hệ quản trị CSDL
  - Các phần mềm tiện ích khác
  - ...
- Con người:
  - Trình độ chuyên môn Tin học

# Xác định và thu thập yêu cầu

## ❖ Phân loại yêu cầu:

- Yêu cầu chức năng:
  - Lưu trữ
  - Tra cứu
  - Tính toán
  - Kết xuất
- Yêu cầu phi chức năng:
  - ?

# Xác định và thu thập yêu cầu

- ❖ Kỹ thuật thu thập yêu cầu:
  - Phỏng vấn
  - Bảng câu hỏi
  - Nghiên cứu các tài liệu
  - Quan sát thực tế
  - Phân tích thiết kế nhóm (JAD)
  - ...

- ❖ Phỏng vấn cá nhân/phỏng vấn nhóm?
  - Phỏng vấn cá nhân
  - Phỏng vấn nhóm
- ❖ Phỏng vấn tự do/phỏng vấn có định hướng?
  - Phỏng vấn tự do
    - Người được hỏi có cảm giác thoải mái, cung cấp nhiều thông tin sâu sắc
    - Nguy cơ: không có được những thông tin cần thiết, thông tin khó hệ thống được

## ❖ Phỏng vấn tự do/phỏng vấn có định hướng?

### ○ Phỏng vấn có định hướng

- Người được hỏi có thể cảm thấy không thoải mái, ít có khả năng ghi nhận được nhận xét, ý kiến, suy nghĩ riêng của người được phỏng vấn, ít cảm nhận được thái độ của họ đối với hiện trạng.
- Có thể định hướng nội dung cần tìm hiểu, có thể hệ thống hóa các vấn đề ghi nhận được



# Kỹ thuật Phỏng vấn

- ❖ Làm việc với cấp lãnh đạo để **nắm mục tiêu của hệ thống phần mềm** cần xây dựng, những **đối tượng cần phỏng vấn**
- ❖ Yêu cầu cấp lãnh đạo thông báo xuống các phòng ban, đơn vị để hợp tác
- ❖ Phân tích để xác định **đúng và đủ** những đối tượng cần phỏng vấn
- ❖ Hẹn lịch làm việc, Xác định trước **vị trí/trách nhiệm** của người sắp phỏng vấn



## ❖ Khi tìm hiểu, cần ghi nhận các thông tin:

- Nội dung: cái gì?
- Bao giờ có: thời gian
- Bằng cách nào có nội dung thông tin đó
- Nội dung đó ở dạng gì?
- Đánh giá của người được phỏng vấn về tình hình hiện tại thực hiện nghiệp vụ

## ❖ Không nên:

- Đưa nhận xét cá nhân của người phỏng vấn
- Dùng thuật ngữ/ngôn ngữ Tin học



# Kỹ thuật dùng bảng câu hỏi

## ❖ Phải trình bày rõ:

- Mục đích của bảng câu hỏi,
- Mục đích sử dụng những thông tin trong bảng câu hỏi,
- Tính bảo mật thông tin trả lời (không tiết lộ ai là người cung cấp thông tin, không để lộ ra ngoài tổ chức...)

## ❖ Hướng dẫn cách điền: rất cần thiết, cần lưu ý để tránh hiểu nhầm

## ❖ Thời hạn trả về:

- Cần nhắc khi gần đến thời hạn



# Kỹ thuật dùng bảng câu hỏi

- ❖ Câu hỏi trình bày rõ ràng
- ❖ Hình thức bảng câu hỏi phải dễ dàng để xử lý tự động
- ❖ Cần để dành chỗ để ghi câu trả lời.
  - Thêm chỗ cho lời bình
  - Không phải chỉ ở cuối trang, hay cuối bảng câu hỏi,
  - Nên dự kiến những câu hỏi nào sẽ có ý kiến thêm thì nên có sẵn chỗ để ghi lời bình ngay dưới câu hỏi đó)



# Kỹ thuật Nghiên cứu tài liệu

- ❖ Các tài liệu (có thể tìm hiểu những văn bản chung)
- ❖ Những quy định nội bộ, Các báo cáo liên quan
- ❖ Những quy định về quy trình nghiệp vụ
  - Rất khó có đầy đủ văn bản quy định về quy trình nghiệp vụ
  - Đơn vị đạt chuẩn ISO?
- ❖ Những quy định “bất thành văn” !!!
- ❖ Thường dễ hơn kỹ thuật phỏng vấn hay bảng câu hỏi
- ❖ Thường được tiến hành trước làm cơ sở chuẩn bị cho việc phỏng vấn hay dùng bảng câu hỏi



# Kỹ thuật Quan sát thực tế

- ❖ Tiến hành sau cùng (nếu cần thiết)
- ❖ Kiểm tra lại:
  - Đã hiểu đúng nghiệp vụ hiện tại?
  - Có những ngoại lệ?
  - Phát hiện những khó khăn, lỗ hổng trong quy trình nghiệp vụ
- ❖ Nhược điểm: ?



# Phân tích thiết kế nhóm - JAD

- ❖ JAD – Joint Application Design
- ❖ Kỹ thuật áp dụng cho các giai đoạn **phân tích yêu cầu và đặc tả**
- ❖ Các nhà phát triển và khách hàng làm việc như một nhóm chung và **có trách nhiệm chung** đối với kết quả đầu ra.
- ❖ Nhóm làm việc sẽ **thảo luận** các yêu cầu cần có, thiết kế các màn hình và báo cáo, **xây dựng mô hình định khung nhanh, rút ra các đặc tả**
- ❖ Chủ yếu dựa trên sự **đồng thuận** (consensus)

# Mô hình hóa yêu cầu

- ❖ Tại sao phải mô hình hóa yêu cầu?
- ❖ Mô hình hóa yêu cầu:
  - Mô hình hóa là một trong các cách thức **mô tả trực quan** một vấn đề dưới dạng các sơ đồ
  - Mô hình hóa sử dụng hệ thống các **ký hiệu** tương ứng với các **thành phần** của vấn đề cần mô tả
- ❖ **Mô hình hóa phần mềm** (Hệ thống dựa trên phần mềm):
  - Mô tả trực quan các thành phần của phần mềm dưới dạng các sơ đồ

# Mô hình hóa yêu cầu

## ❖ Có hai mức mô hình hóa:

- **Mức quan niệm** (giai đoạn phân tích): Mô tả **phát thảo** các thành phần của phần mềm.
- **Mức logic** (giai đoạn thiết kế): Mô tả **chi tiết** các thành phần của phần mềm

## ❖ Các loại mô hình:

- Mô hình **chức năng**: Mô tả thành phần xử lý
- Mô hình **dữ liệu**: Mô tả thành phần dữ liệu
- Mô hình **đối tượng**: Mô tả đồng thời dữ liệu và xử lý

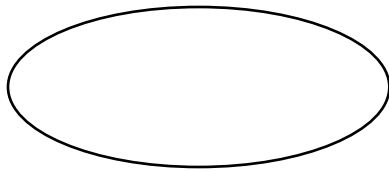


- ❖ Mô hình hóa **hướng chức năng** với sơ đồ luồng dữ liệu – **DFD (Data Flow Diagram)**
  - Mục tiêu: Mô tả mức **quan niệm (phát thảo)** các thành phần của phần mềm với sự **chú trọng** trên thành phần **xử lý**

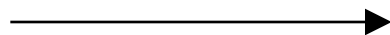
## ❖ Các ký hiệu



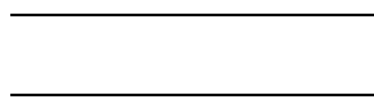
Tác nhân/thiết bị (Người sử dụng, thiết bị phát sinh hay tiếp nhận dữ liệu)



Khối xử lý

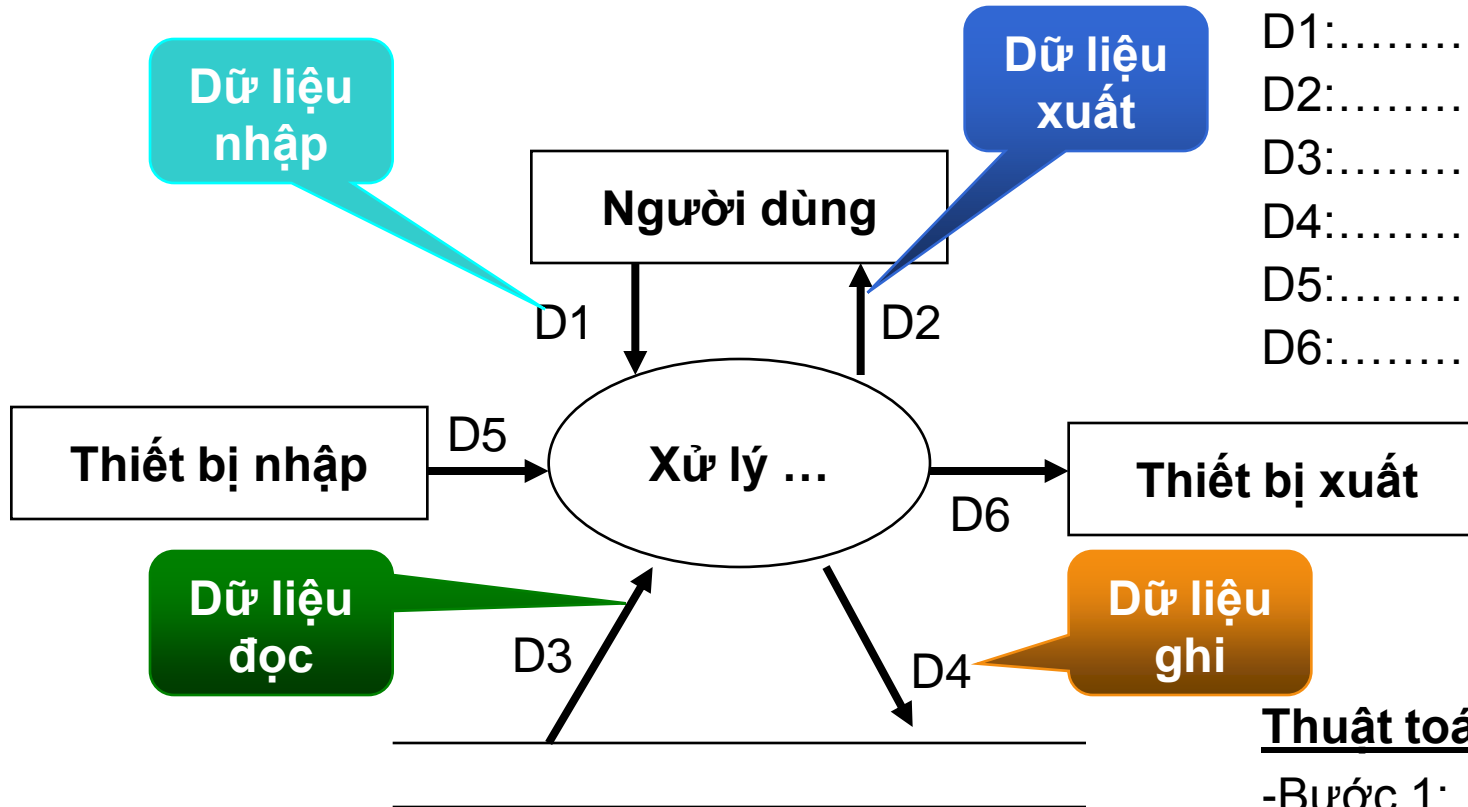


Luồng dữ liệu (thông tin)



Bộ nhớ phụ (Hồ sơ, Sổ sách, tập tin, csdl...)

# Sơ đồ tổng quát



## Ý nghĩa từng dòng dữ liệu

- D1:.....
- D2:.....
- D3:.....
- D4:.....
- D5:.....
- D6:.....

## Thuật toán xử lý:

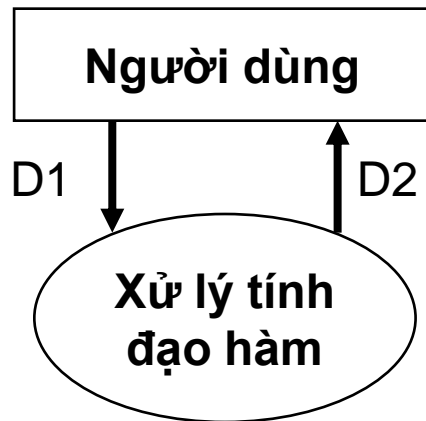
- Bước 1:.....
- Bước 2:.....
- Bước 3:.....
- .....

# Ví dụ 1

❖ Xét chức năng tính đạo hàm của một đơn thức

## 1. Phân tích

Sơ đồ luồng dữ liệu



❖ D1: Đơn thức cần tính đạo hàm  $P$

❖ D2: Đơn thức kết quả  $Q$

❖ Thuật toán xử lý

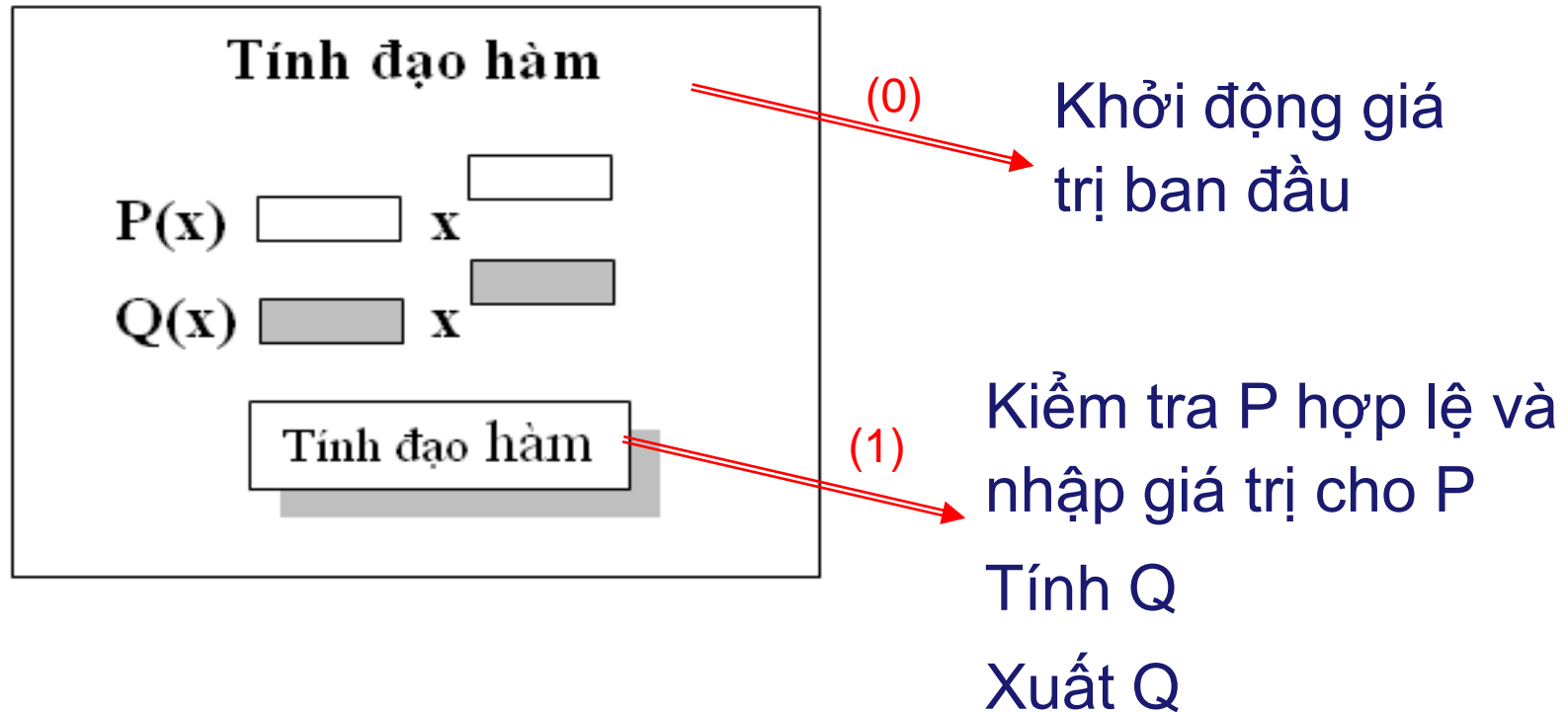
- Nhập và kiểm tra D1

- Tính và xuất D2

# Ví dụ 1

## 2. Thiết kế

- Mô tả chi tiết cách thức giao diện



## 2. Thiết kế

- Mô tả chi tiết kiểu dữ liệu: Sử dụng kiểu cấu trúc DON\_THUC với hai thành phần:
  - Hệ số có kiểu số thực
  - Số mũ có kiểu số nguyên (không âm)
- Mô tả chi tiết các hàm xử lý
  - Hàm xử lý biến cố 1
  - Hàm kiểm tra hệ số, Hàm kiểm tra số mũ
  - Hàm nhập đơn thức
  - Hàm tính đạo hàm, Hàm xuất đơn thức

## 3. Lập trình

'Khai báo biến

```
Private Type DON_THUC
```

```
    Heso as Single
```

```
    Somu as Integer
```

```
End type
```

'Khai báo biến

```
Dim P as DON_THUC
```

```
Dim Q as DON_THUC
```

# Ví dụ 1

## 3. Lập trình

'Hàm xử lý biến cố tính đạo hàm trên màn hình

```
Private sub cmdDaoham_Click()
```

```
    if Kiem_Tra_He_So() and Kiem_Tra_So_Mu() then
```

```
        Nhap()
```

```
        DaoHam()
```

```
        Xuat()
```

```
    End if
```

```
End sub
```

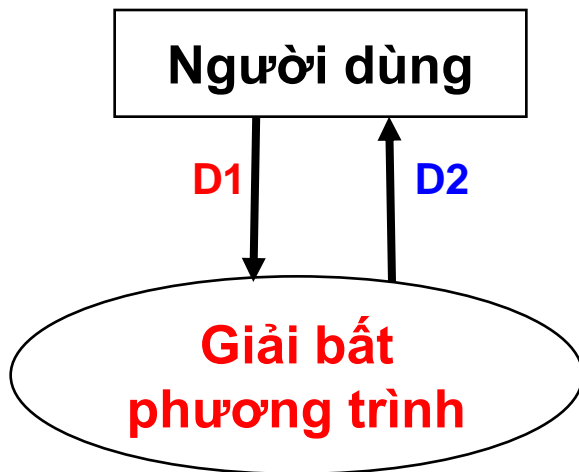


## Ví dụ 2

- ❖ Xét chức năng giải bất phương trình bậc nhất có dạng  $ax + b \geq 0$ , với  $a \neq 0$
- ❖ Hãy phân tích, thiết kế và lập trình

## Ví dụ 2

### ❖ Sơ đồ luồng dữ liệu



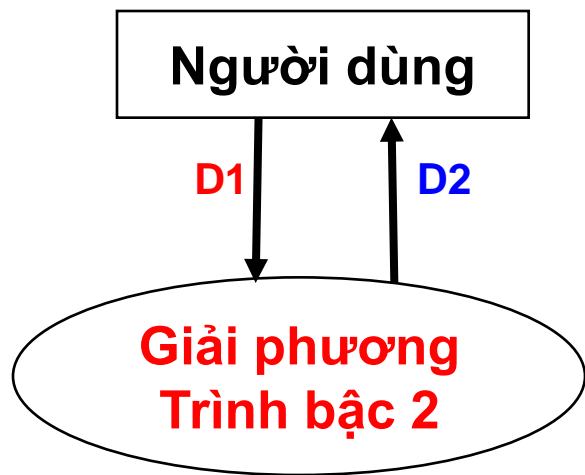
- ❖ **D1**: Các hệ a, b của bất phương trình
- ❖ **D2**: Nghiệm của bất phương trình thuộc một trong hai dạng sau:
  - Dạng 1:  $(-\infty, x_0]$ , Dạng 2:  $[x_0, +\infty)$
- ❖ **Xử lý**:
  - Nhập và kiểm tra D1,  $a \neq 0$
  - Tính D2 theo qui tắc:
    - $a > 0$ : nghiệm thuộc dạng 2
    - $a < 0$ : nghiệm thuộc dạng 1
    - Với  $x_0 = -b/a$
  - Xuất D2

## Ví dụ 3

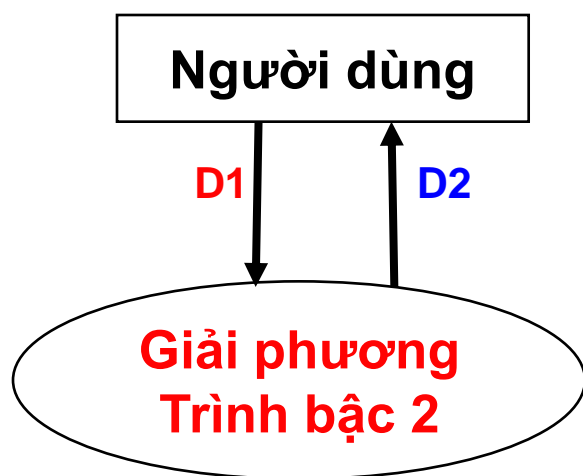
- ❖ Xét chức năng giải phương trình bậc hai:  
 $ax^2 + bx + c = 0$ , với  $a \neq 0$
- ❖ Hãy phân tích, thiết kế và lập trình

## Ví dụ 3

### ❖ Sơ đồ luồng dữ liệu



- ❖ **D1**: Các hệ  $a, b, c$  của tam thức bậc 2
  - $P(x) = ax^2 + bx + c$
- ❖ **D2**: Nghiệm của phương trình  $P(x) = 0$  thuộc một trong ba loại sau:
  - Vô nghiệm
  - Nghiệm kép  $x_1 = x_2$
  - Hai nghiệm phân biệt  $x_1$  và  $x_2$



### ❖ Xử lý:

- Nhập và kiểm tra D1 (  $a \neq 0$  )
- Tính D2 theo các bước:
  - $\Delta = b^2 - 4ac$
  - Nếu  $\Delta < 0$  : nghiệm loại 1
  - Nếu  $\Delta = 0$  : nghiệm loại 2, với  $x_1 = x_2 = -b/2a$
  - Nếu  $\Delta > 0$  : nghiệm loại 3, với  $x_1 =$

$$x_2 = \frac{-b \pm \sqrt{\Delta}}{2a}$$

- Xuất D2

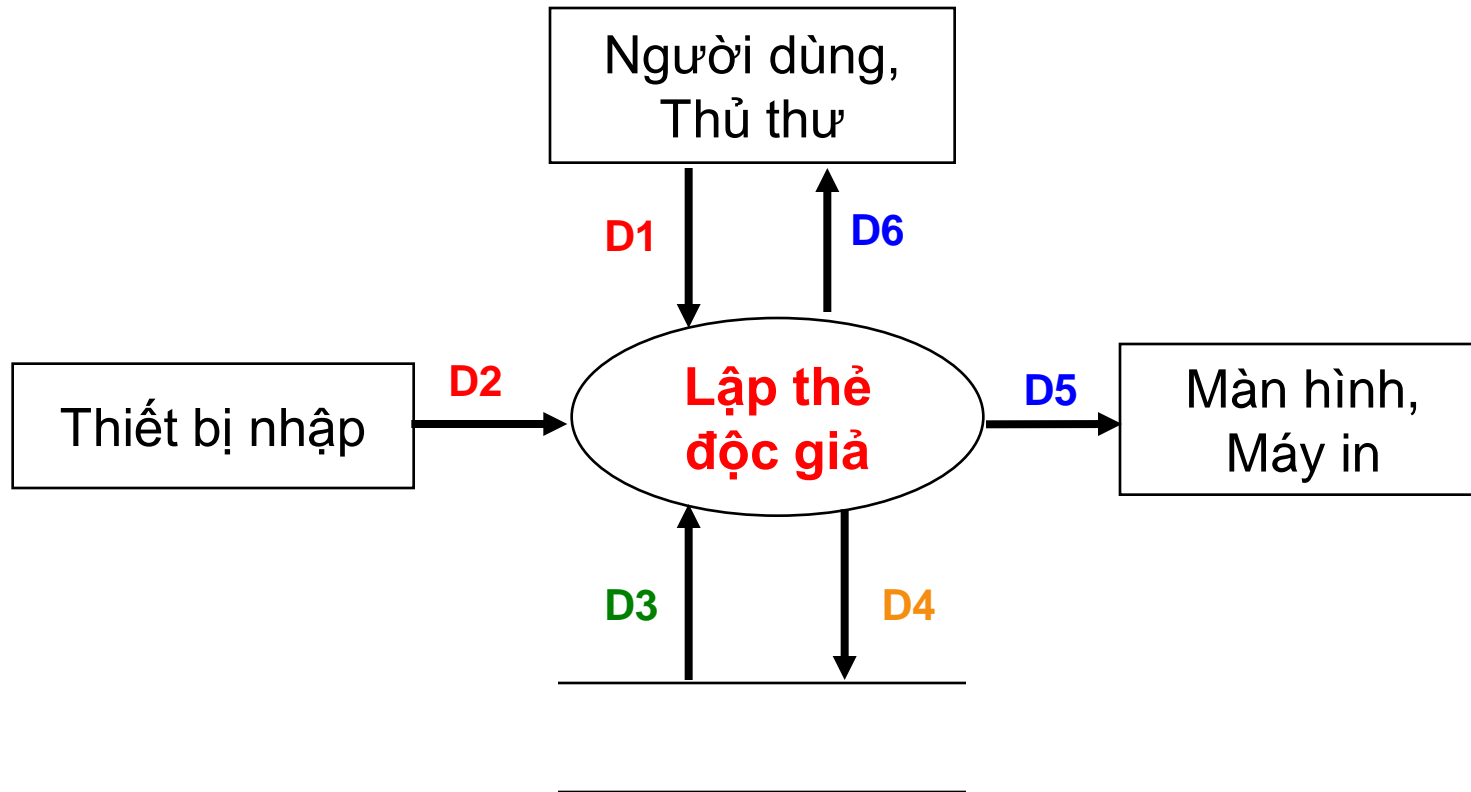
## Ví dụ 4

- ❖ Xét phần mềm quản lý thư viện, hãy lập sơ đồ luồng dữ liệu cho yêu cầu **Lập thẻ đọc giả**

BM2:	Thẻ Đọc Giả	
Họ và tên: .....	Loại đọc giả:.....	Ngày sinh: .....
Địa chỉ: .....	Email: .....	Ngày lập thẻ: .....
Người lập: .....		

**QĐ2: Có 2 loại đọc giả (X,Y). Tuổi đọc giả từ 18 đến 55. Thẻ có giá trị 6 tháng.**

# Ví dụ 4



## Ví dụ 4

### ❖ Giải thích:

- D1: Thông tin về thẻ độc giả: Họ tên, Loại độc giả, Ngày sinh, Địa chỉ, E-Mail, Ngày Lập Thẻ.
- D2: Không có
- D3: Danh sách các loại độc giả, Tuổi tối thiểu, Tuổi tối đa, Thời hạn sử dụng.
- D4: D1
- D5: D4
- D6: Danh mục loại độc giả



## Ví dụ 4

### ❖ Thuật toán:

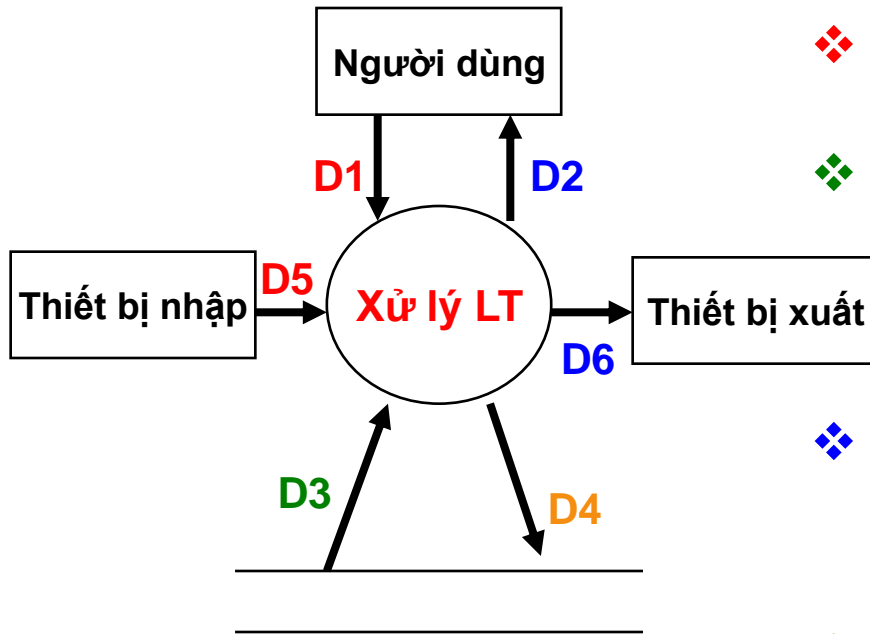
- Bước 01: Kết nối dữ liệu
- Bước 02: Đọc D3 từ bộ nhớ phụ
- Bước 03: Nhận D1 từ người dùng
- Bước 04: Kiểm tra “Loại độc giả” có thuộc “danh sách các loại độc giả” hay không?
- Bước 05: Tính tuổi độc giả.
- Bước 06: Kiểm tra qui định “Tuổi tối thiểu”
- Bước 07: Kiểm tra qui định “Tuổi tối đa”

## Ví dụ 4

### ❖ Thuật toán:

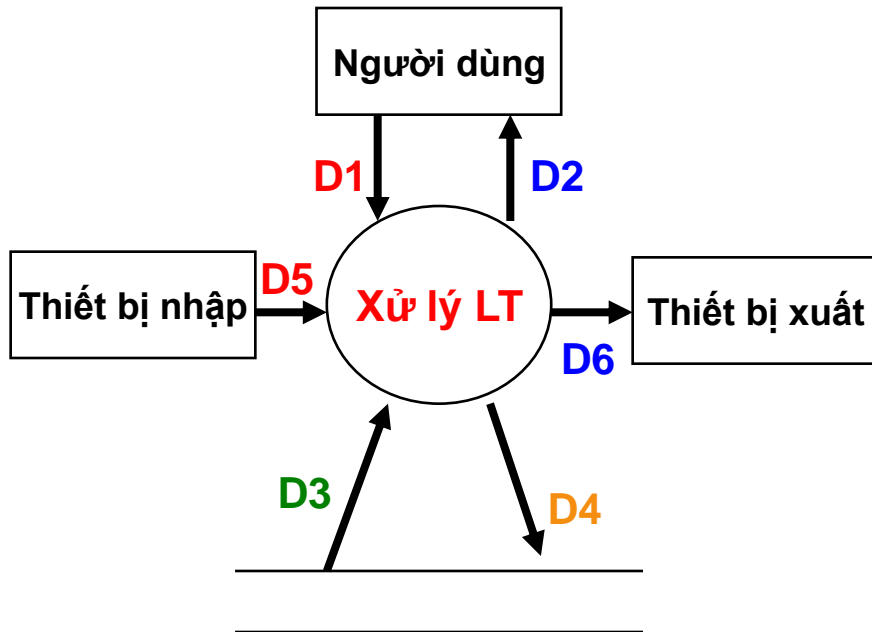
- Bước 08: Nếu không thỏa tất cả các qui định trên thì tới bước 12
- Bước 09: Tính ngày hết hạn của thẻ.
- Bước 10: Lưu D4 xuống bộ nhớ phụ
- Bước 11: Xuất D5 ra máy in
- Bước 12: Đóng kết nối cơ sở dữ liệu
- Bước 13: Kết thúc.

# Sơ đồ tổng quát cho Yêu cầu lưu trữ



- ❖ **D1**: Thông tin cần lưu trữ (dựa vào biểu mẫu liên quan)
- ❖ **D5**: Thông tin cần lưu trữ (chỉ có trong một số yêu cầu đặc biệt)
- ❖ **D3**:
  - Các danh mục để chọn lựa
  - Dữ liệu cần thiết cho việc kiểm tra tính hợp lệ (dựa vào quy định)
- ❖ **D2**:
  - Các danh mục để chọn lựa
  - Kết quả thành công/thất bại
- ❖ **D4**: Dữ liệu được lưu trữ (dựa vào biểu mẫu).
  - Ghi chú: Thông thường  
 $D4 = D1 (+ D5) (+ ID \text{ tự phát sinh})$
- ❖ **D6**: Dữ liệu kết xuất (chỉ có trong một số yêu cầu đặc biệt)

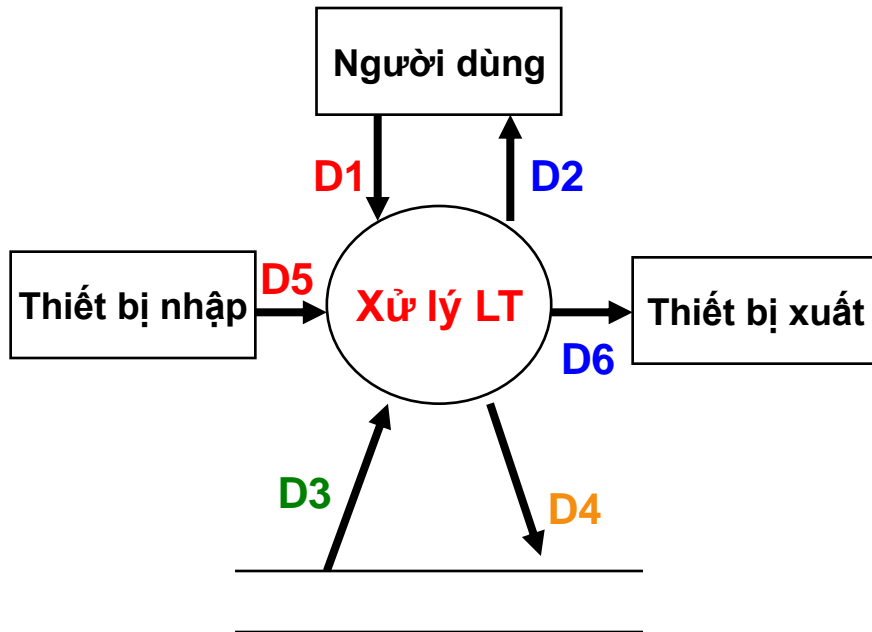
# Sơ đồ tổng quát cho Yêu cầu lưu trữ



## ❖ Xử lý lưu trữ

- **Đọc D3** để lấy các tham số, quy định và danh mục
- **Hiển thị D2** (các danh mục)
- **Nhận thông tin D1, D5** (nếu cần)
- Kiểm tra các thông tin D1, D5 có thỏa quy định liên quan hay không (dựa vào D3 nếu cần thiết)
- Nếu thỏa quy định, **ghi D4**, **thông báo kết quả D2** (nếu cần) và **xuất D6** (nếu cần thiết)

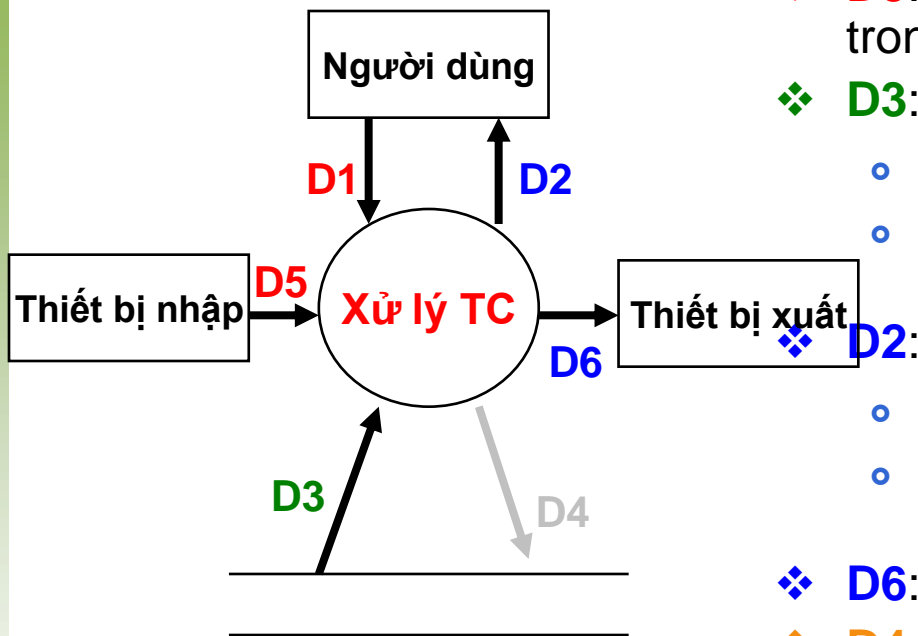
# Sơ đồ tổng quát cho Yêu cầu lưu trữ



## ❖ Ghi chú:

- **D1** không nhất thiết chứa toàn bộ thông tin trong biểu mẫu liên quan
- Tùy theo quy định có thể có hay không có **D5**
- **D4** hoặc **D6** không nhất thiết phải trùng với **D1** hoặc **D5**
- **D2** không nhất thiết phải trùng với **D3**

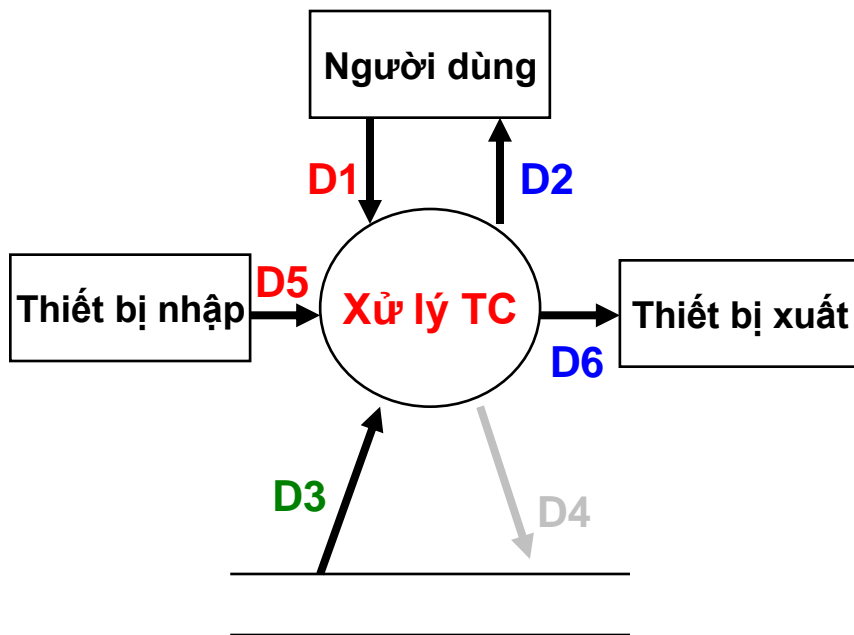
# Sơ đồ tổng quát cho Yêu cầu tra cứu



- ❖ **D1**: Thông tin về đối tượng muốn tìm kiếm (dựa vào **biểu mẫu liên quan đến đối tượng cần tìm kiếm**)
- ❖ **D5**: Thông tin về đối tượng muốn tìm kiếm (chỉ có trong một số yêu cầu đặc biệt)
- ❖ **D3**:
  - Các danh mục để chọn lựa
  - Dữ liệu về đối tượng khi tìm thấy (dựa vào **biểu mẫu liên quan đến đối tượng cần tìm kiếm**)
- ❖ **D2**:
  - Các danh mục để chọn lựa
  - Dữ liệu về đối tượng khi tìm thấy (dựa vào **biểu mẫu liên quan đến đối tượng cần tìm kiếm**)
- ❖ **D6**: Dữ liệu kết xuất (thông thường là cần thiết)
- ❖ **D4**: Dữ liệu cần lưu trữ lại
  - Thông thường không cần thiết
  - Cần thiết khi nào???

# Sơ đồ tổng quát cho Yêu cầu tra cứu

## ❖ Xử lý tra cứu

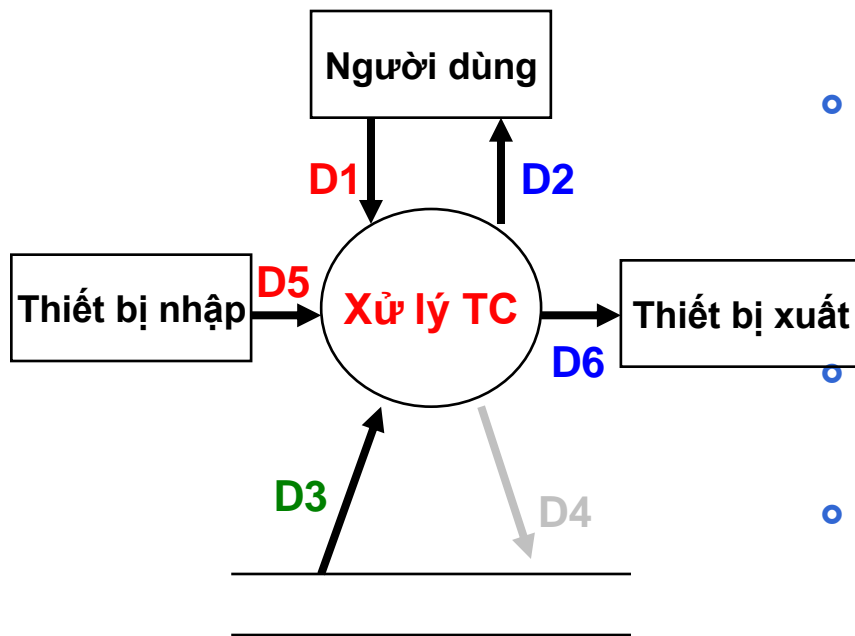


- Đọc để lấy các danh mục (D3)
- Hiển thị D2 (các danh mục)
- Nhận thông tin về tiêu chí tìm kiếm D1, D5 (nếu cần)
- Tìm kiếm theo các tiêu chí D1, D5, nhận được danh sách các đối tượng tìm được (D3)
- Hiển thị thông tin kết quả (D2) và kết xuất D6 (nếu cần)

# Sơ đồ tổng quát cho Yêu cầu tra cứu

## ❖ Ghi chú:

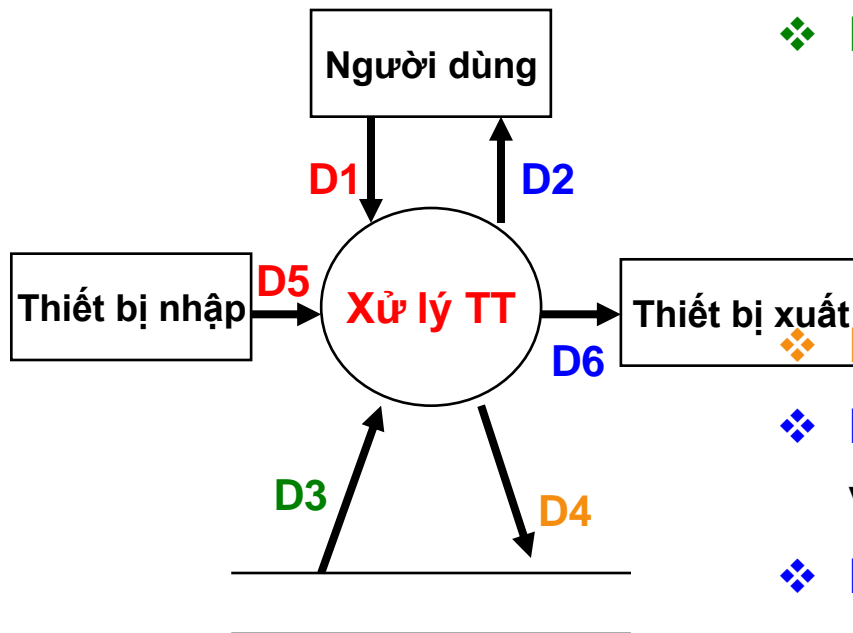
- Có rất nhiều mức độ khác nhau từ rất đơn giản đến rất phức tạp để xác định **D1**
- **D1** chứa nhiều thông tin thì việc tìm kiếm sẽ dễ dàng cho người dùng và ngược lại sẽ khó khăn cho phần thiết kế và cài đặt chức năng này
- **D3** thông thường là danh sách các đối tượng tìm thấy cùng với thông tin liên quan.
- **D3** cũng có rất nhiều mức độ khác nhau để xác định các thông tin của đối tượng tìm thấy
- **D2** và **D6** thường trùng với **D3** (nhưng không nhất thiết)



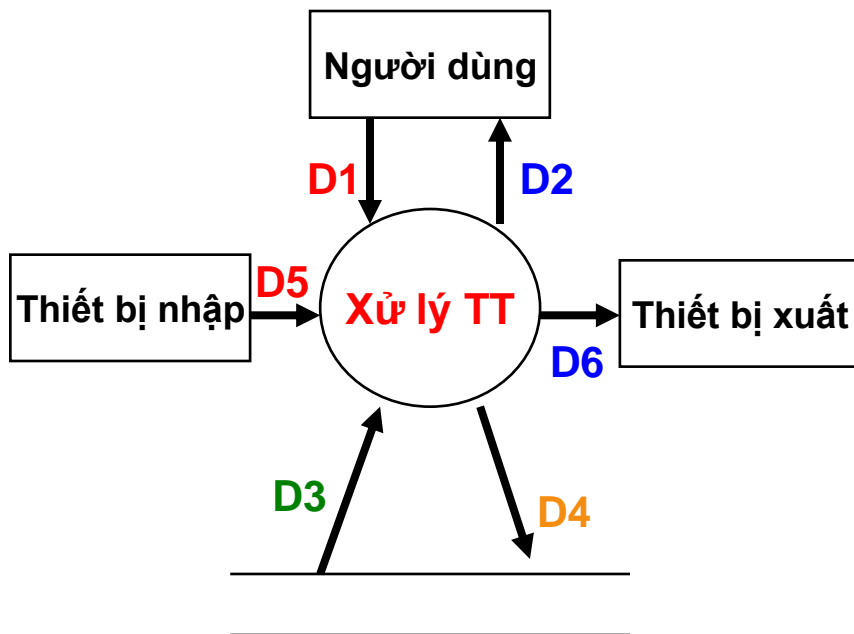


# Sơ đồ tổng quát cho Yêu cầu tính toán

- ❖ **D1**: Thông tin về đối tượng cần thực hiện việc xử lý tính toán (dựa vào **các biểu mẫu liên quan**)
- ❖ **D5**: Thông tin về đối tượng cần thực hiện việc xử lý tính toán (chỉ có trong một số yêu cầu đặc biệt)
- ❖ **D3**:
  - Dữ liệu cần thiết cho việc xử lý tính toán (dựa vào biểu mẫu và quy định liên quan)
  - Các tham số tính toán
- ❖ **D4**: Kết quả của xử lý tính toán
- ❖ **D2**: Kết quả của xử lý tính toán (thường gồm cả **D3** và **D4**)
- ❖ **D6**: Dữ liệu kết xuất (thường gồm cả **D3** và **D4**)



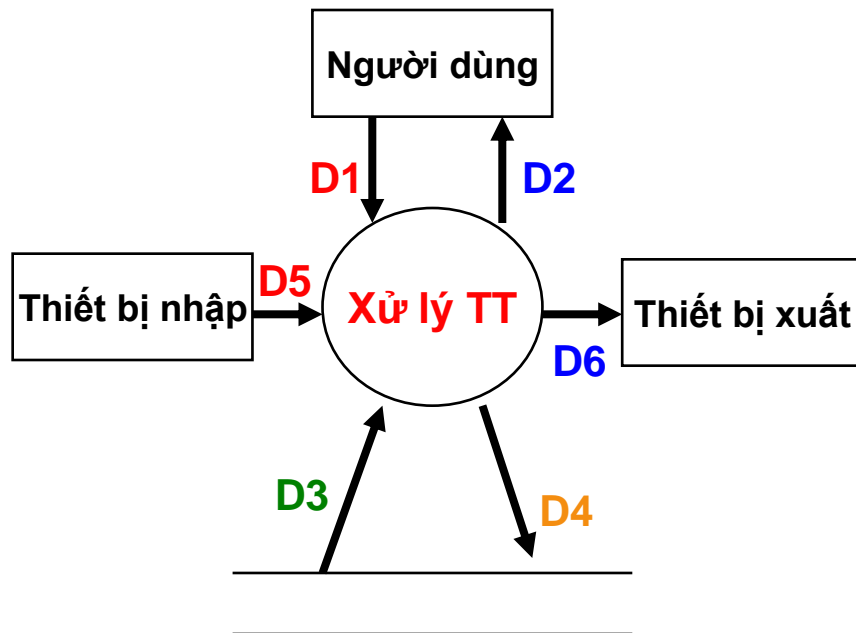
# Sơ đồ tổng quát cho Yêu cầu tính toán



## ❖ Xử lý tính toán

- **Nhận thông tin D1, D5** (nếu cần)
- **Đọc D3** để lấy các dữ liệu cần thiết cho việc tính toán (kể cả các tham số)
- Sử dụng **D1, D3, D5** và quy định liên quan để tính kết quả **D4**
- **Ghi kết quả D4**
- **Hiển thị** thông tin kết quả **D2** và kết xuất **D6**

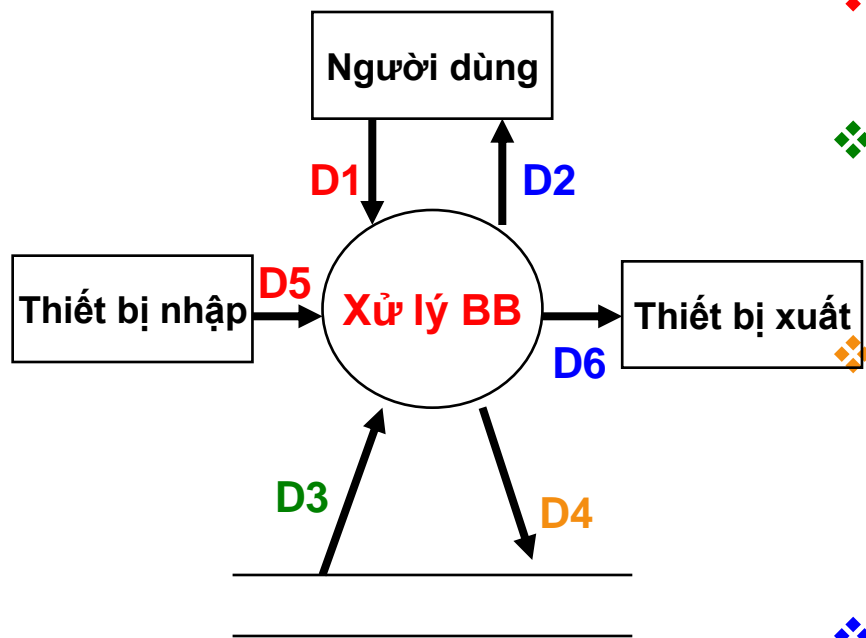
# Sơ đồ tổng quát cho Yêu cầu tính toán



## ❖ Ghi chú:

- **D1** thường có chứa yếu tố thời gian thực hiện xử lý tính toán
- Có nhiều mức độ khác nhau xác định **D1** trong xử lý tính toán (để tăng tính tiện dụng)
- **D1** có thể rỗng (tính toán cho mọi đối tượng trong tất cả cột mốc thời gian liên quan)
- **D4** có thể có hay không có  
=> Khi nào cần **D4**?
- Thông thường **D2** và **D6** bao gồm **D3** và **D4**

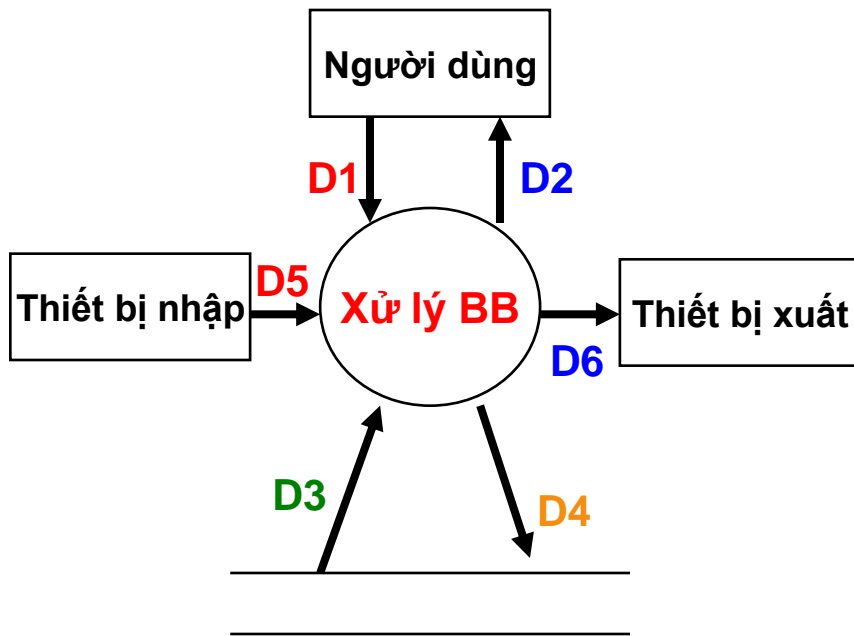
# Sơ đồ tổng quát cho Yêu cầu báo biểu



- ❖ **D1**: Thông tin về báo biểu muốn thực hiện (dựa vào *biểu mẫu liên quan*)
- ❖ **D5**: Thông tin về báo biểu muốn thực hiện (chỉ có trong một số yêu cầu đặc biệt)
- ❖ **D3**: Dữ liệu cần thiết cho việc thực hiện báo biểu (dựa vào biểu mẫu và quy định liên quan)
- ❖ **D4**: Thông tin có trong báo biểu liên quan (cần thiết phải lưu lại) nhưng chưa được xử lý và ghi nhận lại (yêu cầu xử lý tính toán)
- ❖ **D2**: Thông tin về báo biểu được lập (biểu mẫu liên quan)
- ❖ **D6**: Dữ liệu kết xuất (thường giống **D2**)

# Sơ đồ tổng quát cho Yêu cầu báo biểu

## ❖ Xử lý báo biểu

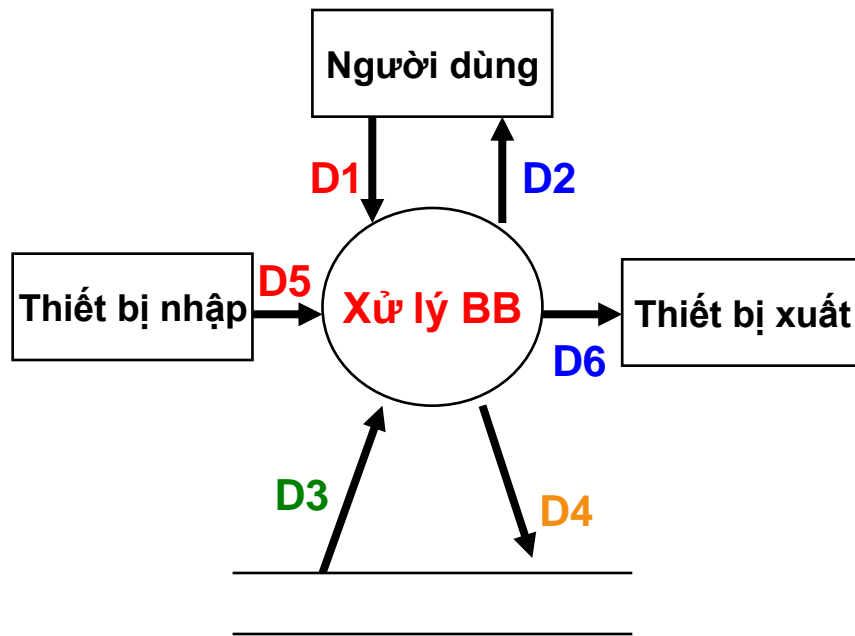


- Nhận thông tin D1, D5 (nếu cần)
- Đọc D3 để lấy các dữ liệu cần thiết cho việc lập báo biểu
- Nếu có D4 thì tính toán theo quy định và Ghi kết quả D4
- Hiển thị thông tin báo biểu D2 và kết xuất D6

# Sơ đồ tổng quát cho Yêu cầu báo biểu

## ❖ Ghi chú:

- **D1** thường có chứa yếu tố thời gian của báo biểu
- Có nhiều mức độ khác nhau xác định **D1** trong xử lý tính toán (để tăng tính tiện dụng)
- **D4** có thể có hay không có  
=> Khi nào cần **D4**?
- Thông thường **D2** và **D6** bao gồm **D3** và **D4**



# Bài tập

1. Xét phần mềm quản lý học sinh với nghiệp vụ tiếp nhận hồ sơ học sinh

**BM1:**

**HỒ SƠ HỌC SINH**

Họ và tên:.....

Giới tính:.....

Ngày sinh:.....

Địa chỉ:.....

Email:.....

**Qui định:** Tuổi học sinh từ 15 đến 20.

Hãy lập sơ đồ luồng dữ liệu

# Bài tập

2. Xét phần mềm quản lý bán hàng với nghiệp vụ lập phiếu thu tiền của khách hàng

Biểu mẫu	Phiếu thu tiền
Họ tên khách hàng:.....	CMND:.....
Địa chỉ:.....	Ngày thu:.....
Số tiền:.....	Lý do thu:.....

Qui định: Tiền thu tối thiểu là 100.000đ, Có 3 lý do thu là A, B, C

Hãy lập sơ đồ luồng dữ liệu



## 3. Xét phần mềm quản lý các đại lý với nghiệp vụ tiếp nhận hồ sơ đại lý

**BM1:** HỒ SƠ ĐẠI LÝ

Tên:..... Loại đại lý:.....

Điện thoại:..... Địa chỉ:..... Quận:.....

Ngày tiếp nhận:..... Email:.....

**Qui định:** Có 2 loại đại lý (1,2). Có 20 quận. Trong mỗi quận có tối đa 4 đại lý

Hãy lập sơ đồ luồng dữ liệu

## 4. Xét phần mềm quản lý giải vô địch bóng đá quốc gia

**BM1:**

**HỒ SƠ ĐỘI BÓNG**

Tên đội:

Sân nhà:

Stt	Cầu thủ	Ngày sinh	Loại cầu thủ	Ghi chú

**QĐ1:** Tuổi cầu thủ từ 16 đến 40. Có 2 loại cầu thủ (trong nước và nước ngoài). Số cầu thủ từ 15 đến 22. Tối đa 3 cầu thủ nước ngoài.

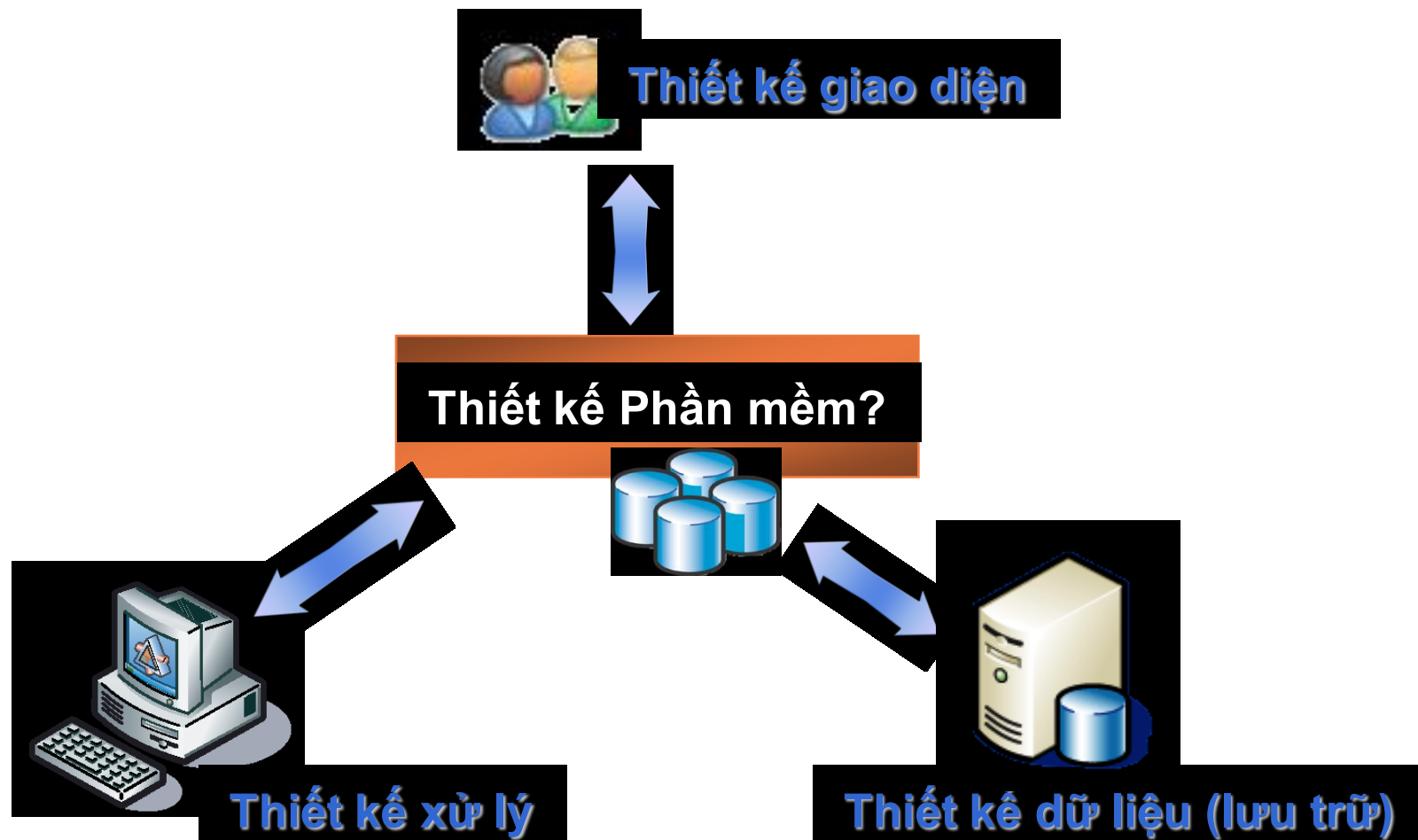
Hãy lập sơ đồ luồng dữ liệu



# Chương 3: Thiết kế Phần mềm



# Nội dung



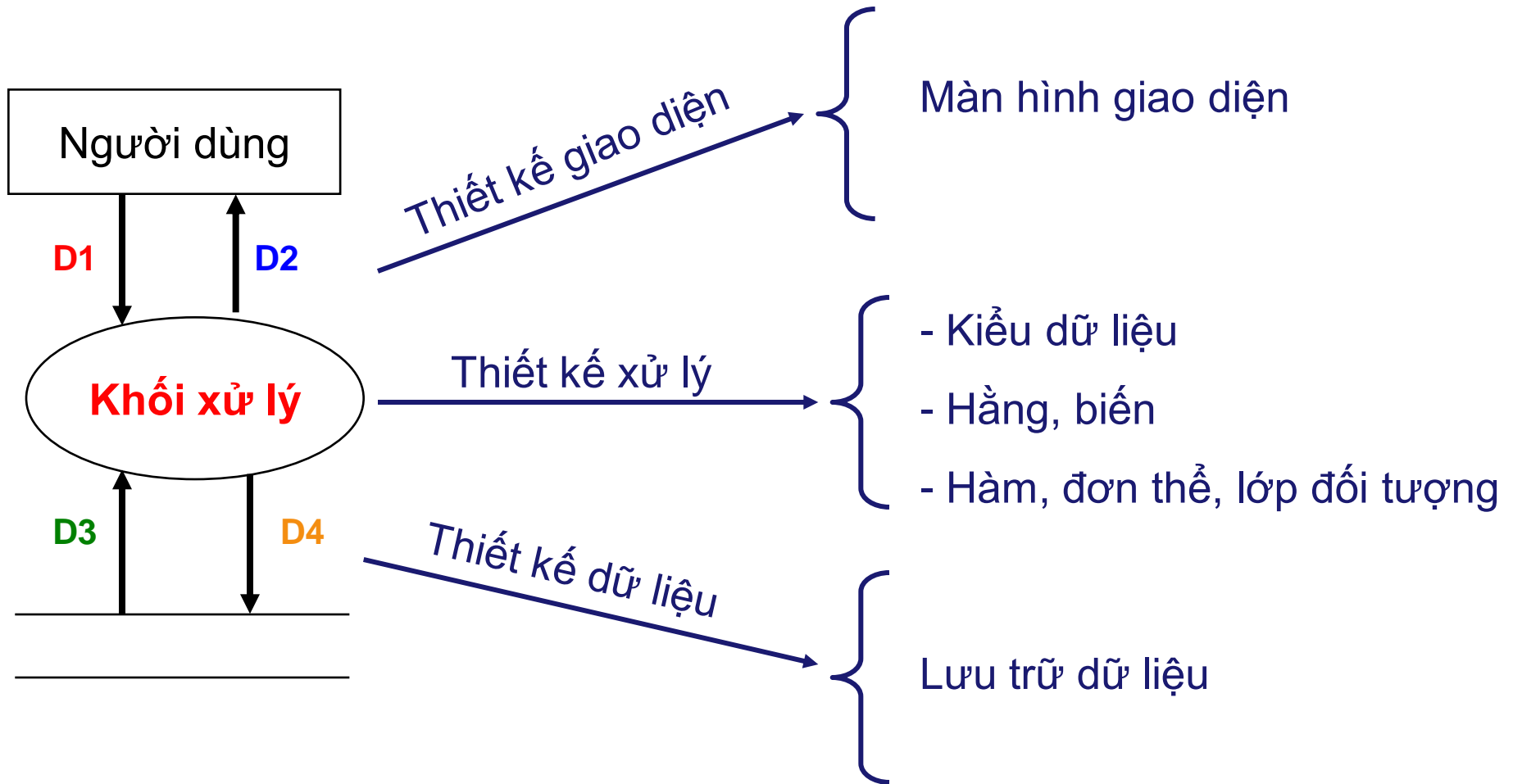
# Tổng quan

## ❖ Khái niệm:

- Thiết kế phần mềm là **mô tả chi tiết tổ chức, hoạt động** các đơn vị xử lý của phần mềm
- Dựa trên **kết quả** của việc **phân tích**
- Là **cơ sở** cho việc **thực hiện** phần mềm



# Tổng quan



# Thiết kế giao diện

- ❖ Thiết kế giao diện: **Mô tả chi tiết** cách thức **giao tiếp** giữa người sử dụng và phần mềm
- ❖ Màn hình giao diện
  - Nội dung
  - Hình thức trình bày
  - Biến cố phải xử lý
  - **Lập danh sách các biến cố:**

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
1			
...			



# Thiết kế giao diện

❖ Mô tả danh sách các thành phần của giao diện:

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
1						
...						

# Thiết kế dữ liệu (lưu trữ)

- ❖ Thiết kế dữ liệu: Mô tả chi tiết cách thức tổ chức, lưu trữ các dữ liệu của phần mềm
- ❖ Tổ chức lưu trữ
  - Bảng/Tập tin
  - Thuộc tính/Cấu trúc
  - Liên kết giữa các bảng/Tập tin
- ❖ Danh sách các bảng

STT	Bảng	Ý nghĩa	Ghi chú
1			
...			

# Thiết kế dữ liệu (lưu trữ)

## ❖ Danh sách các thuộc tính bảng X

STT	Thuộc tính	Kiểu	Ràng buộc	Giá trị Khởi động	Ghi chú
1					
...					

# Thiết kế Xử lý

- ❖ Thiết kế xử lý: **Mô tả chi tiết** hệ thống **các hàm xử lý** (cùng với các hằng, biến, kiểu liên quan) của phần mềm
- ❖ Danh sách các kiểu dữ liệu xử lý

STT	Kiểu dữ liệu	Ý nghĩa	Ghi chú
...	...	...	...

- ❖ Danh sách các thuộc tính kiểu dữ liệu X

STT	Thuộc tính	Kiểu	Ràng buộc	Giá trị khởi động	Ghi chú
...	...	...	...	...	...

# Thiết kế Xử lý

## ❖ Danh sách các biến

STT	Biến	Kiểu	Ý nghĩa	Ghi chú
...	...	...	...	...

## ❖ Danh sách các hằng

STT	Hằng	Kiểu	Giá trị	Ý nghĩa	Ghi chú
...	...	...	...	...	...

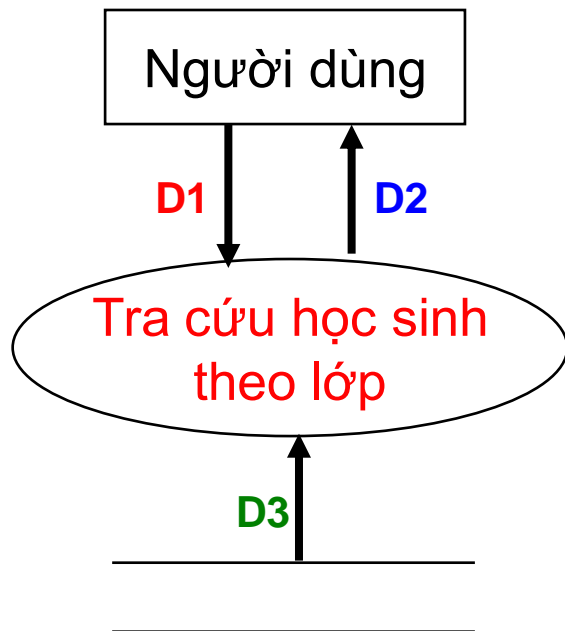
## ❖ Danh sách các hàm xử lý

STT	Hàm	Tham số	Kết quả trả về	Thuật giải	Ý nghĩa	Ghi chú
...	...	...	...	...	...	...

❖ Phân tích, thiết kế chức năng tra cứu học sinh theo lớp

## 1. Phân tích

### Sơ đồ luồng dữ liệu



❖ Giải thích:

- D1: Lớp được chọn
- D2: Danh sách học sinh trong lớp được chọn
- D3: Danh sách lớp, Danh sách học sinh

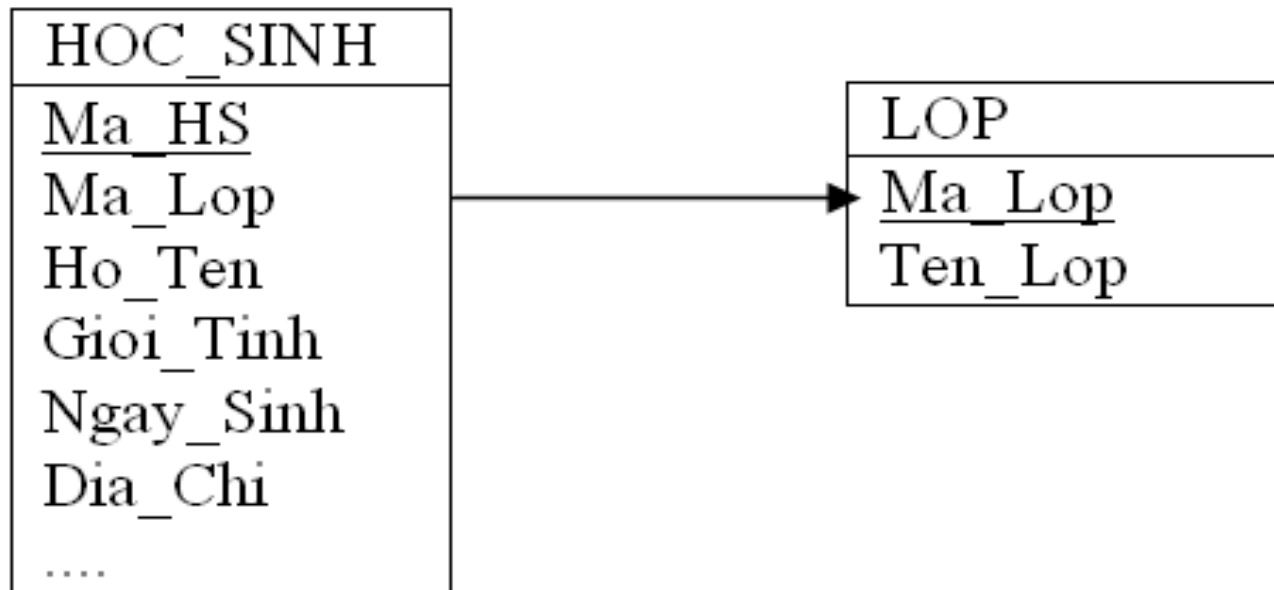
❖ Thuật toán xử lý

- Đọc D3
- Nhập D1
- Tính D2 (DS học sinh trong lớp được chọn)
- Xuất D2

## 2. Thiết kế

### a. Thiết kế dữ liệu (lưu trữ)

Dùng cơ sở dữ liệu quan hệ gồm hai bảng: HOC\_SINH, LOP



# Ví dụ (tt)

## 2. Thiết kế

### b. Thiết kế giao diện

Màn hình tra cứu học sinh theo lớp

Danh sách lớp

STT	Mã lớp	Tên lớp

Danh sách học sinh

STT	Họ tên	Giới tính	Ngày sinh	Địa chỉ

(0)

(1)

Cách 1



# Ví dụ (tt)

## 2. Thiết kế

### b. Thiết kế giao diện

Màn hình tra cứu học sinh theo lớp

Tên lớp  ▼

Danh sách học sinh

STT	Họ tên	Giới tính	Ngày sinh	Địa chỉ

(1)

(0)

Cách 2

# Ví dụ (tt)

## 2. Thiết kế

### b. Thiết kế giao diện

Màn hình tra cứu học sinh theo lớp

Tên lớp  ▼ (1)

Danh sách học sinh

STT	Họ tên

Hồ sơ học sinh

Họ tên   Nam

Ngày sinh

Địa chỉ

(0)

(2)

Cách 3

## 2. Thiết kế

### b. Thiết kế giao diện

Danh sách các biến cố

STT	Điều kiện kích hoạt	Xử lý
0	Khởi động màn hình	???
1	Chọn lớp	???
2	Chọn học sinh	???

# Ví dụ (tt)

## 2. Thiết kế

### b. Thiết kế xử lý

STT	Hàm	Tham số	Kiểu trả về	Thuật giải	Ghi chú
1	Xuất danh sách lớp			?	
2	Xuất danh sách học sinh			?	
...	...				

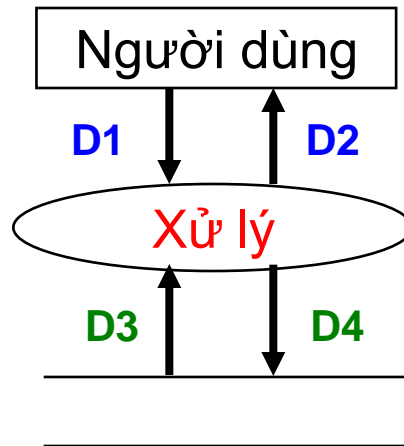
### 3. Bài tập: Mở rộng khi Trường quản lý phân cấp theo khối

# Thiết kế dữ liệu



# Thiết kế dữ liệu

- ❖ **Mục tiêu:** Mô tả cách thức **tổ chức lưu trữ** dữ liệu của phần mềm/hệ thống dựa trên phần mềm.



- ❖ D1, D2 được lưu trữ theo cách thức nào trong **bộ nhớ chính**?
- ❖ D3, D4 được lưu trữ theo cách thức nào trong **bộ nhớ phụ**?

- ❖ Cách thức lưu trữ dữ liệu trên **bộ nhớ chính**:
  - Kiểu cấu trúc
  - Kiểu mảng
  - Kiểu xâu
  - Kiểu cây
  - ...
- ❖ Cách thức lưu trữ dữ liệu trên **bộ nhớ phụ**:
  - Tập tin (Có cấu trúc/Không có cấu trúc)
  - Cơ sở dữ liệu (Quan hệ/Đối tượng)

## ❖ Các yêu cầu chất lượng:

- Tính đúng đắn: Lưu trữ đầy đủ và đúng ngữ nghĩa các thông tin có trong nghiệp vụ liên quan
- Tính tiến hóa: Lưu trữ thông tin về tổ chức và qui định có trong nghiệp vụ liên quan
- Tính hiệu quả: Lưu trữ tiết kiệm nhất không gian bộ nhớ, truy xuất nhanh nhất thông tin cần thiết
- Tính bảo mật: Lưu trữ thông tin về các người sử dụng phần mềm cùng với quyền hạn tương ứng



## ❖ Thuộc tính

- Là các đặc trưng mô tả về đối tượng

## ❖ Thực thể

## ❖ Sơ đồ logic

- Công cụ cho phép **mô tả trực quan** cách thức lưu trữ dữ liệu trên bộ nhớ phụ với việc sử dụng cơ sở dữ liệu quan hệ
- Liên kết giữa các bảng (**mối kết hợp**)

# Phân loại thuộc tính

- ❖ Thuộc tính **khóa**
- ❖ Thuộc tính có **giá trị rời rạc**
- ❖ Thuộc tính **đa trị**
- ❖ Thuộc tính là **đối tượng phụ**
- ❖ Thuộc tính **tính toán**

GiangVien			
<u>GVID</u>	<u>&lt;pi&gt;</u>	<u>TNBID</u>	<u>&lt;M&gt;</u>
MaGV		TNBCODE	
HocVi		TNBNAME	
ChucDanh		TNBNAME	
TenVT		TNBNAME	
Ho		TNBNAME	
Ten		TNBNAME	
DiaChi		TNBDESCRIPTION	
DienThoai		TNBPHONE	
Email		TNBPHONE	

# Phân loại thuộc tính

- ❖ Thuộc tính **khóa**
- ❖ Thuộc tính có **giá trị rời rạc**:
  - Thuộc tính phái của Sinh viên chỉ có 2 giá trị: **Nam hoặc Nữ**
  - Thuộc tính phái của Sinh viên chỉ có 2 giá trị: **0 hoặc 1 (0: Nam, 1: Nữ)**
  - Thuộc tính điểm học phần chỉ có các giá trị **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.**
  - Thuộc tính loại nhân viên chỉ có các giá trị: **Nhân viên văn phòng, nhân viên bán hàng, nhân viên sản xuất.**

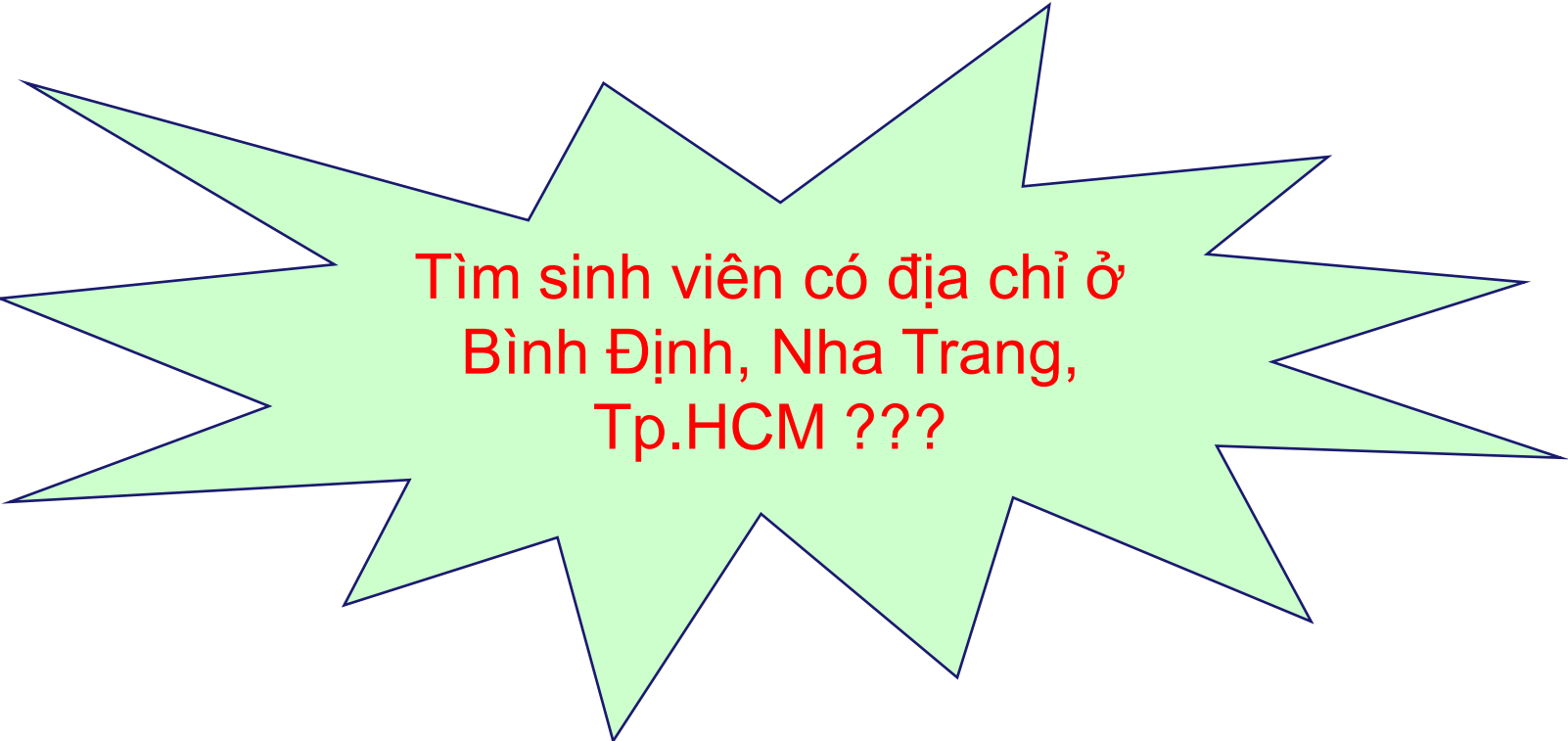
# Phân loại thuộc tính

## ❖ Thuộc tính đa trị:

- **Thuộc tính điện thoại của nhân viên** là thuộc tính đa trị:
  - 081234567
  - 09081234567,...
- **Thuộc tính email của sinh viên** cũng là một thuộc tính đa trị:
  - dungta@yahoo.com,
  - dungta@gmail.com,...

# Phân loại thuộc tính

❖ Thuộc tính là **đối tượng phụ**:



Tìm sinh viên có địa chỉ ở  
Bình Định, Nha Trang,  
Tp.HCM ???

# Phân loại thuộc tính

- ❖ Thuộc tính là **đối tượng phụ**:
  - **Thuộc tính địa chỉ** của sinh viên, nhân viên là một đối tượng phụ, bao gồm các thông tin:
    - Số nhà
    - Đường
    - Phường/xã
    - Quận/huyện
    - Tỉnh thành
  - **Thuộc tính ngày sinh** của sinh viên, nhân viên cũng là một đối tượng phụ gồm các thông tin: ngày, tháng, năm

# Phân loại thuộc tính

## ❖ Thuộc tính **tính toán**:

- Thuộc tính **thành tiền** trong hóa đơn là một thuộc tính tính toán, được tính bằng **tổng số lượng \* đơn giá** của các mặt hàng
- Thuộc tính **điểm trung bình** của học sinh, sinh viên cũng là một thuộc tính tính toán

- ❖ Quan hệ giữa các thực thể: Phân loại quan hệ dựa trên bản số
  - Quan hệ “1-1”
  - Quan hệ “1-nhiều”
  - Quan hệ “nhiều-nhiều”



# Ví dụ 1

- ❖ Xét phần mềm quản lý bán hàng với nghiệp vụ lập phiếu thu tiền của khách hàng

Biểu mẫu	Phiếu thu tiền	
Họ tên khách hàng:.....	CMND:.....	
Địa chỉ:.....	Ngày thu:.....	
Số tiền:.....	Lý do thu:.....	

**Qui định:** Tiền thu tối thiểu là 100.000đ, Có 3 lý do thu là A, B, C

Hãy lập sơ đồ logic dữ liệu

## Ví dụ 1 (tt)

- ❖ Cách 0: dùng 1 bảng
- ❖ Cách 1: dùng 2 bảng (PHIEU\_THU, LYDO\_THU)
- ❖ Cách 2: dùng 2 bảng (PHIEU\_THU, KHACH\_HANG)
- ❖ Cách 3: dùng 4 bảng

# Ví dụ 1 (tt)

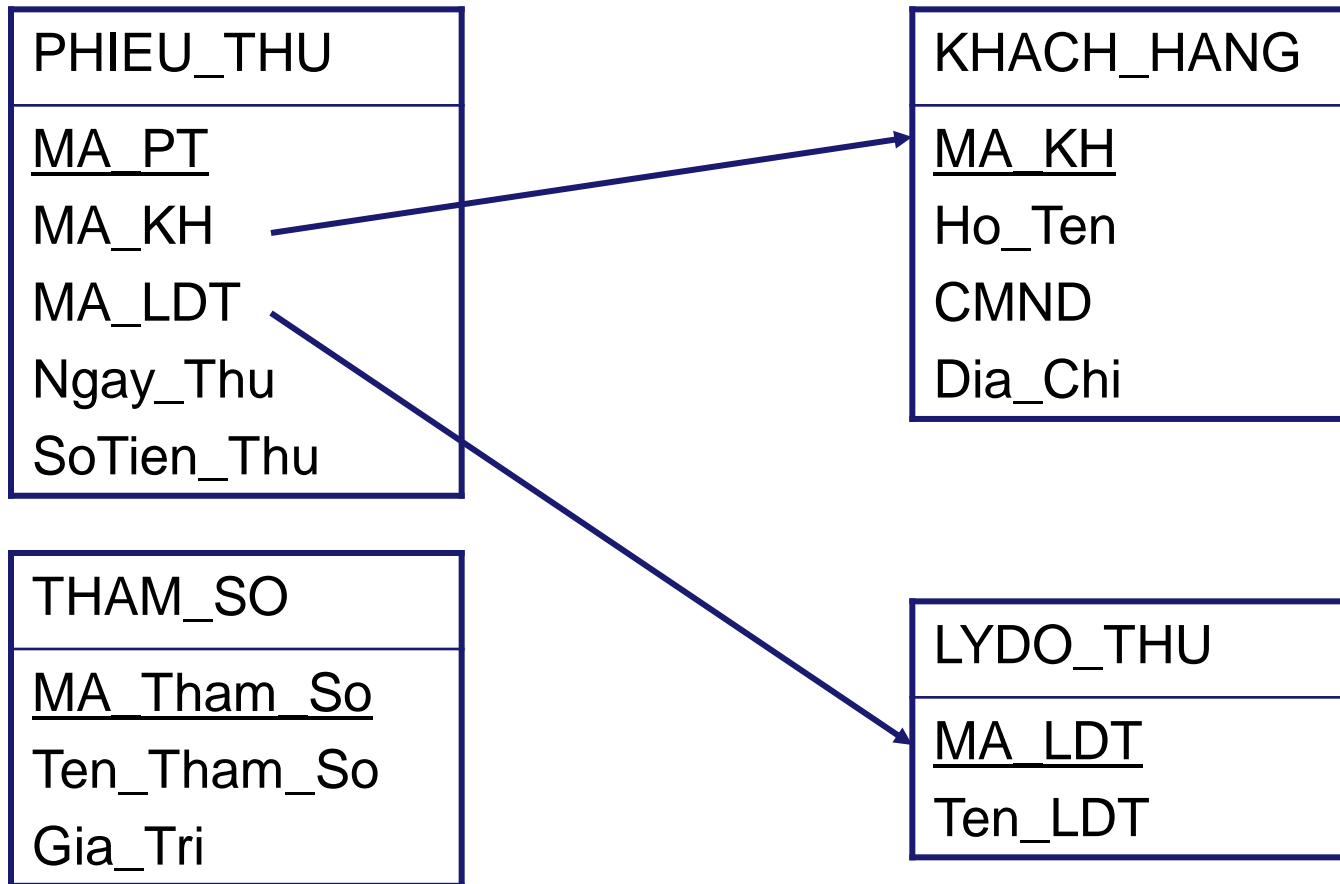


Table THAM\_SO dùng để lưu những qui định

## Ví dụ 2

- ❖ Xét phần mềm quản lý nhân sự với nghiệp vụ tiếp nhận nhân viên

Biểu mẫu	Hồ sơ nhân viên
Họ tên :.....	Giới tính:.....
Ngày sinh:.....	Địa chỉ:.....
Điện thoại:.....	Trình độ:.....
Đơn vị:.....	

**Qui định:** Công ty có 4 đơn vị X, Y, Z, T. Tuổi nhân viên nam từ 20 đến 60. Tuổi nhân viên nữ từ 18 đến 55. Có 4 trình độ: Sau đại học, đại học, cao đẳng, trung cấp.

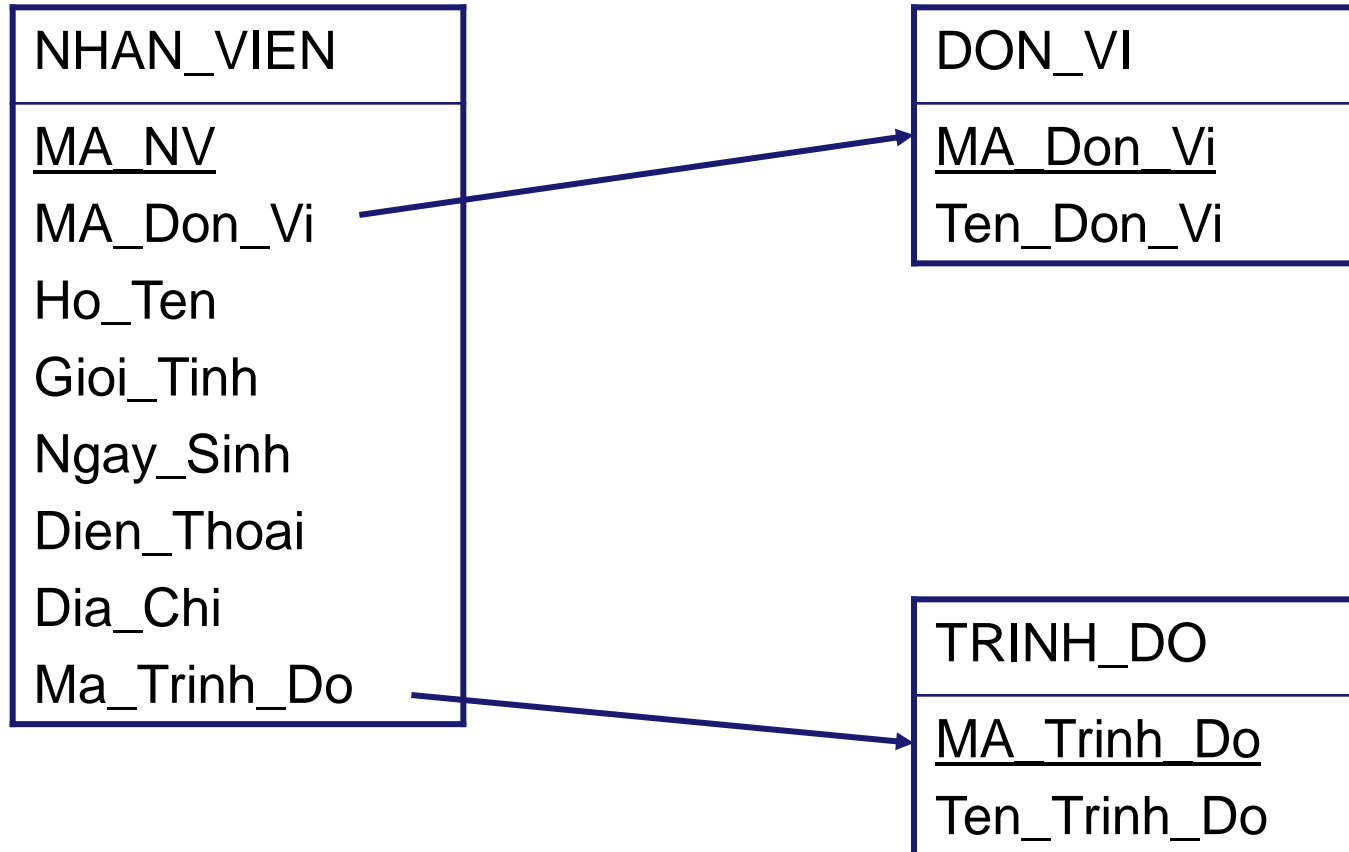
## Ví dụ 2 (tt)

Hãy lập sơ đồ logic dữ liệu cho yêu cầu nghiệp vụ trên trong các trường hợp sau:

- a. Không yêu cầu theo dõi quá trình chuyển nhân viên từ đơn vị này sang đơn vị khác
- b. Có yêu cầu theo dõi quá trình chuyển nhân viên
- c. Bổ sung qui định: Đơn vị X chỉ nhận các nhân viên có trình độ đại học và sau đại học

## Ví dụ 2 (tt)

a.



## Ví dụ 2 (tt)

❖ Table THAM\_SO có thể thiết kế theo nhiều cách

- Cách 1

Khóa	ThamSố#1	ThamSố#2	...	ThamSố#N
...	...	...	...	...

Mỗi tham số tương ứng với một cột trong bảng tham số

Bảng tham số thường chỉ gồm 1 dòng (chứa giá trị các tham số hiện hành)



Cần bổ sung tham số mới?

Cần vô hiệu hóa tác dụng của một tham số?

## Ví dụ 2 (tt)

- Cách 2

Được lưu  
dạng chuỗi

MãThamSố	TênThamSố	Kiểu	GiáTrị	TìnhTrạng
...	...	...	...	...

Mỗi tham số tương ứng với **một dòng** trong bảng tham số

**Giá trị hiện tại** của tham số được lưu bằng **dạng chuỗi**

Mỗi tham số cần **lưu trữ kiểu giá trị** để phần mềm “hiểu” đúng **nội dung** giá trị hiện tại của tham số



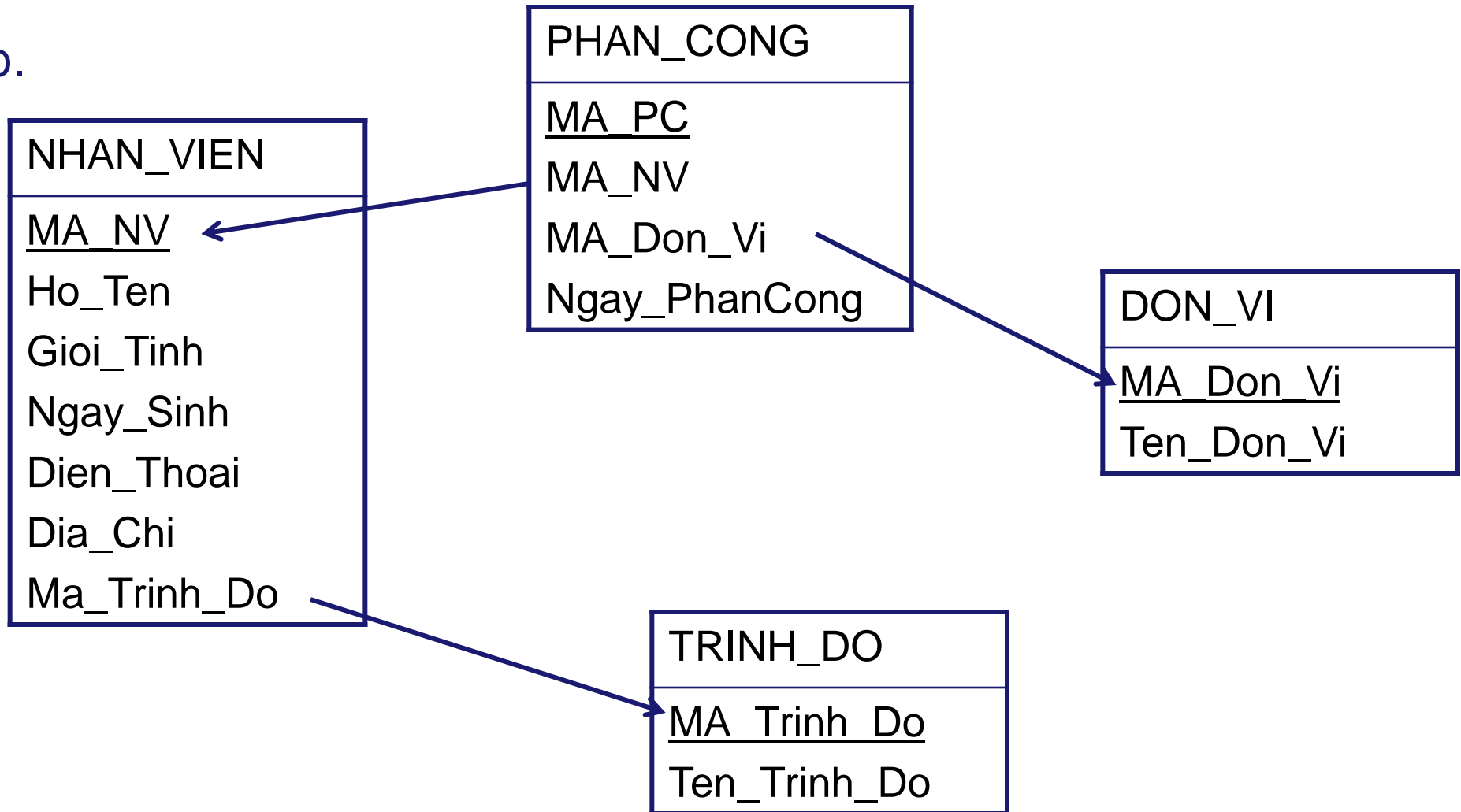
**Cần bổ sung tham số mới?**

**Cần vô hiệu hóa tác dụng của một tham số?**



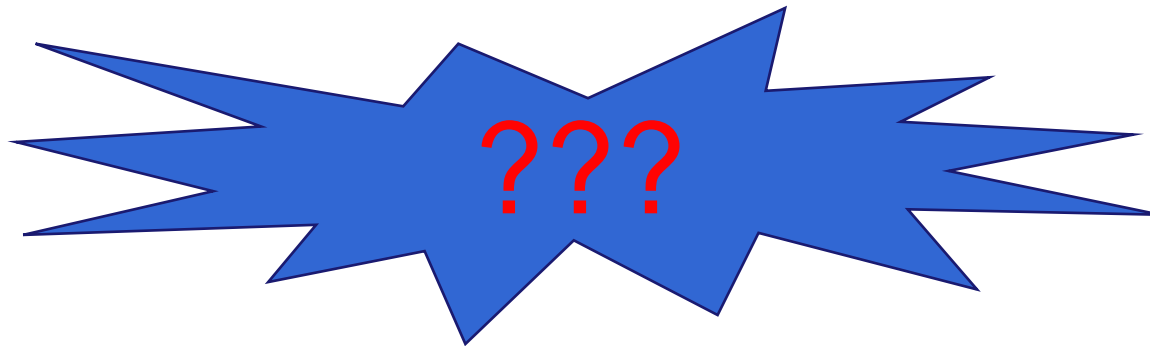
## Ví dụ 2 (tt)

b.



## Ví dụ 2 (tt)

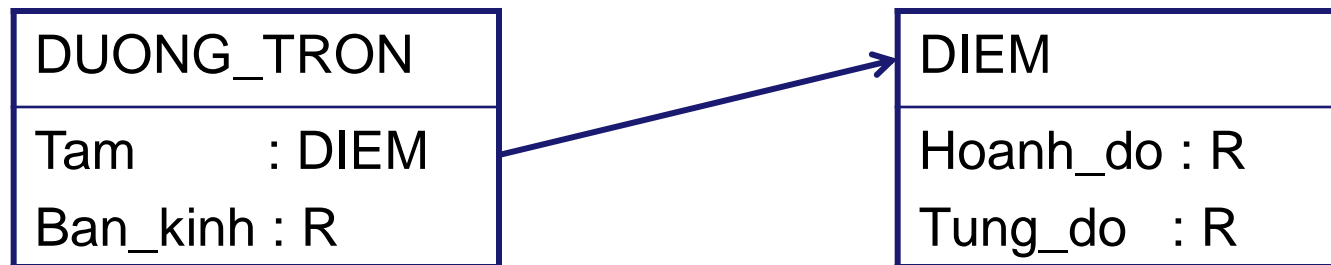
c. Bổ sung qui định: Đơn vị X chỉ nhận các nhân viên có trình độ đại học và sau đại học



- ❖ Khi thiết kế dữ liệu phải **chú ý** đến các vấn đề sau:
  - Không gian
  - Thời gian
  - Khối lượng dữ liệu phát sinh rất nhanh theo thời gian
  - Đáp ứng yêu cầu truy xuất nhanh
  - ....

# Sơ đồ kiểu dữ liệu

- ❖ Cho phép mô tả **cách thức tổ chức lưu trữ dữ liệu** trong **bộ nhớ chính**
- ❖ Ký hiệu: Tương tự như sơ đồ logic
- ❖ Ví dụ: Xét chức năng tính diện tích đường tròn



# Bài tập

1. Xét phần mềm quản lý học sinh với chức năng ghi nhận bảng điểm danh

**Bảng điểm danh**

Lớp:  ▼ Tháng:  ▼

STT	Họ tên	Vắng có phép	Vắng không phép

Hãy lập: sơ đồ luồng dữ liệu, sơ đồ logic dữ liệu

# Bài tập

2. Xét phần mềm quản lý bán hàng với chức năng lập hóa đơn bán hàng. Hãy lập sơ đồ logic dữ liệu

## Hóa đơn bán hàng

Họ tên khách hàng:..... CMND:.....

Địa chỉ:..... Ngày lập hóa đơn:  ▼

STT	Mặt hàng	Số lượng	Đơn giá	Thành tiền

**Qui định:** Số lượng bán phải nhỏ hơn số lượng tồn của mặt hàng tương ứng ít nhất 100.

# Bài tập

3. Xét phần mềm quản lý tiết kiệm với chức năng mở sổ tiết kiệm. Hãy lập sơ đồ logic dữ liệu

## Sổ tiết kiệm

Họ tên khách:.....	CMND:.....
Địa chỉ:.....	Loại tiết kiệm:.....
Ngày mở sổ:.....	Số tiền gửi:.....
Loại ngoại tệ:.....	

**Qui định:** Có 5 loại tiết kiệm: không kỳ hạn, kỳ hạn 3 tháng, 6 tháng, 9 tháng, 12 tháng. Số tiền gửi ít nhất là 200.000đ

Có 2 loại ngoại tệ: tiền VN và đồng USD.

Mỗi loại kỳ hạn có một lãi suất riêng ứng với từng ngoại tệ.

# Thiết kế giao diện





# Tầm quan trọng của giao diện



# Các thành phần của giao diện

Multimedia



Giao diện tương tác



Giao dịch điện tử

Giao diện

Giao diện nhập liệu

Giao diện kết xuất



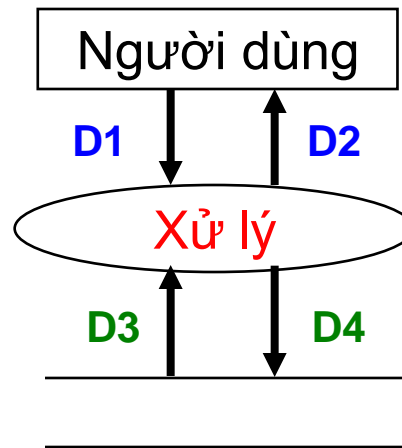
ệu điện tử

Xử lý thông tin

kiếm thông tin

# Mục tiêu

- ❖ **Mục tiêu:** mô tả chi tiết cách thức **giao tiếp** giữa người dùng và phần mềm trong quá trình thực hiện các nghiệp vụ liên quan



- Nội dung trình bày của D1, D2
- Hình thức trình bày của D1, D2
- Biến cố phải xử lý

# Các yêu cầu chất lượng

## ❖ Tính tiện dụng

- Dễ học
- Thự tự nhập trực quan, dễ sử dụng

## ❖ Tính hiệu quả

- Quy trình nhập đơn giản nhất, tự nhiên nhất
- Tận dụng những bước xử lý trên màn hình
- Tránh thêm những công đoạn (thao tác) thừa (không cần thiết)
- Tốc độ thực hiện nhanh

## ❖ Tính tiến hóa

- Các tùy chọn về nội dung
- Hình thức trình bày
- Biến cố phải xử lý



# Các yêu cầu chất lượng

- ❖ Phải quan sát **thói quen** của người sử dụng
- ❖ Thói quen phải tôn trọng => **bắt buộc phải tôn trọng**
  - **Quen** với **phím enter** khi nhập liệu
- ❖ Thói quen **chưa hợp lý hoặc mâu thuẫn** với kỹ thuật  
=> Phải trao đổi, thuyết phục và thống nhất với người sử dụng

## 1. Bổ sung nội dung

- Hướng dẫn sử dụng
- Thuộc tính tính toán
- Thông tin chi tiết liên quan

## 2. Tăng tốc thao tác

- Dùng giá trị định sẵn
- Chuyển ô nhập liệu thành cột nhập liệu
- Sử dụng giá trị thay thế
- ...

## 3. Xử lý lỗi

- Thông báo lỗi chính xác, cơ hội sửa lỗi
- Hạn chế lỗi
- Cấm tuyệt đối lỗi

## 4. Bổ sung, thay thế hình thức trình bày

- Dùng biểu tượng
- Dùng thực đơn động
- Dùng cây
- Dùng sơ đồ
- Thao tác trực tiếp

- ❖ Xét phần mềm Quản lý học sinh với chức năng tiếp nhận học sinh mới

## Tiếp nhận học sinh

Họ tên:..... Giới tính:.....  
Ngày sinh:..... Địa chỉ:.....  
Lớp:.....

**Qui định:** Họ tên phải có. Tuổi từ 15-20. Trường có 20 lớp và 3 khối. Khối 10 có 8 lớp, Khối 11 có 7 lớp. Khối 12 có 5 lớp

- ❖ Hãy thiết kế dữ liệu và giao diện



# Ví dụ

## Tiếp nhận học sinh

Họ tên

Nam

Ngày sinh

Lớp



Địa chỉ

Ghi

## Danh sách học sinh đã tiếp nhận

STT	Mã HS	Tên HS	Giới tính	Ngày sinh
...	...	...	...	...

## ❖ Mô tả các thành phần của giao diện

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
1	Lb_Tieu_de	A_Label	Tiêu đề màn hình			
2	Lb_Hoten	A_Label	Tiêu đề họ tên			
3	Txt_Hoten	A_Textbox	Text box nhập họ tên			
4	Ch_Phai	A_Checkbox				
5	Lb_Ngaysinh	A_Datetime				
...	...	...	...	...	...	...

- ❖ Xét phần mềm quản lý giải bóng đá với các yêu cầu sau:
  - Tiếp nhận đăng ký tham gia
  - Xếp lịch thi đấu
  - Ghi nhận kết quả thi đấu
  - Lập bảng xếp hạng
- ❖ Giải gồm có 4 đội tham gia.
  - Thông tin về mỗi đội bao gồm: Tên đội, danh sách các cầu thủ của đội
  - Thông tin về mỗi cầu thủ bao gồm: Tên cầu thủ, vị trí sở trường
- ❖ Xếp lịch thi đấu
  - Các đội thi đấu vòng tròn hai lượt. Thông tin về trận đấu được xếp lịch: Hai đội bóng tham dự, ngày giờ thi đấu, sân thi đấu (giải diễn ra trên hai sân: A, B)

- ❖ Ghi nhận kết quả thi đấu, chỉ yêu cầu ghi nhận:
    - Tỷ số
    - **Tổng số thẻ vàng, thẻ đỏ**
  - ❖ Lập bảng xếp hạng:
    - Hạng được dựa trên các tiêu chí sau:
      - Điểm số
      - Hiệu số
      - Số bàn thắng
    - Điểm cho mỗi trận thua là 0, hòa là 1, thắng là 3
1. Hãy lập sơ đồ luồng dữ liệu cho mỗi yêu cầu
  2. Lập sơ đồ logic dữ liệu
  3. Thiết kế màn hình giao diện cho từng yêu cầu.

# Giao diện ghi nhận KQTĐ

## ❖ Cách 1:

GHI NHẬN KẾT QUẢ TRẬN ĐẤU

Trận đấu	<input type="text"/>	-	<input type="text"/>
Tỷ số	<input type="text"/>	-	<input type="text"/>
			<input type="button" value="Ghi"/>

# Giao diện ghi nhận KQTĐ

## ❖ Cách 2:

**GHI NHẬN KẾT QUẢ TRẬN ĐẤU**

Đội bóng 1	Đội bóng 2
<input type="text"/>	<input type="text"/>
▼	▼
Tỷ số	
<input type="text"/>	<input type="text"/>
<input type="button" value="Ghi"/>	

# Giao diện ghi nhận KQTD

❖ Cách 3:

GHI NHẬN KẾT QUẢ TRẬN ĐẤU		
Đội bóng 1	Đội bóng 2	Tỷ số

Ghi

# Giao diện ghi nhận KQTD

❖ Cách 4:

## GHI NHẬN KẾT QUẢ TRẬN ĐẤU

STT	Trận đấu	Tỷ số

Ghi



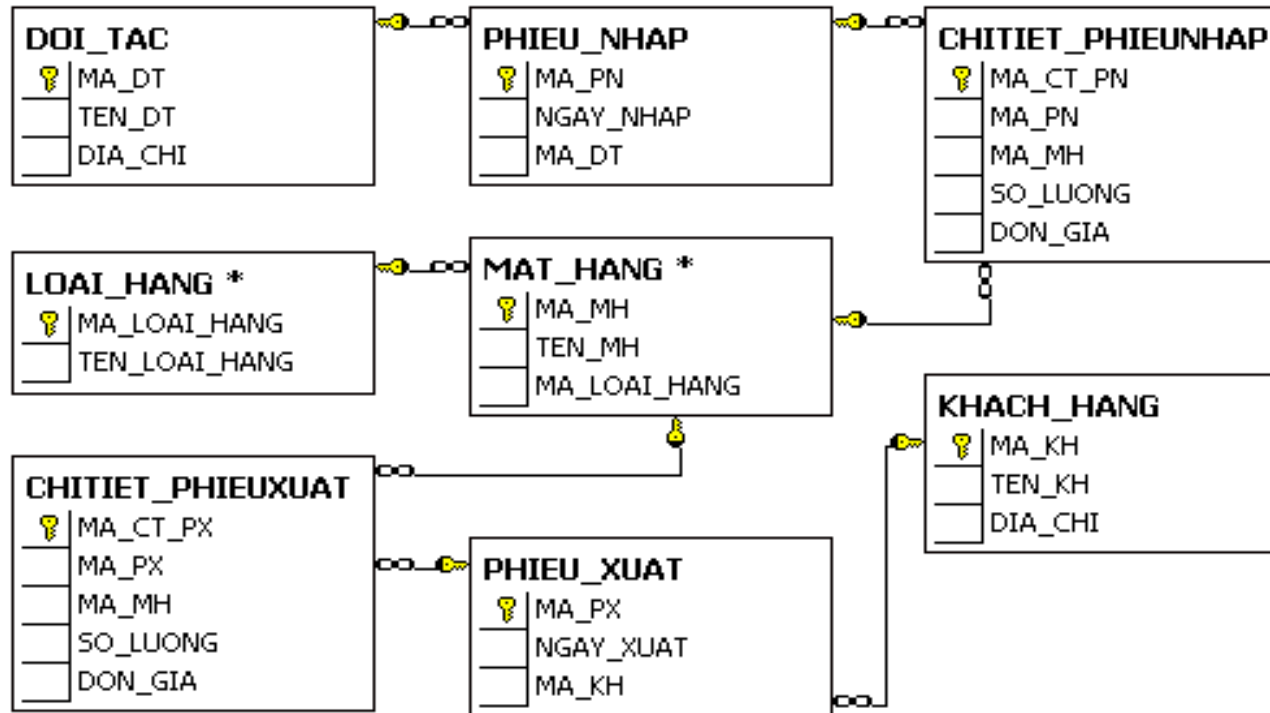
# Giao diện ghi nhận KQTĐ

## ❖ Cách 5:

GHI NHẬN KẾT QUẢ TRẬN ĐẤU				
	Đội bóng 1	Đội bóng 2	...	Đội bóng k
Đội bóng 1	<del> </del>			
Đội bóng 2		<del> </del>		
....			<del> </del>	
Đội bóng k				<del> </del>

# Bài tập

❖ Cho CSDL của phần mềm quản lý bán hàng như sau:



❖ Hãy thiết kế giao diện và mô tả các thành phần của giao diện:

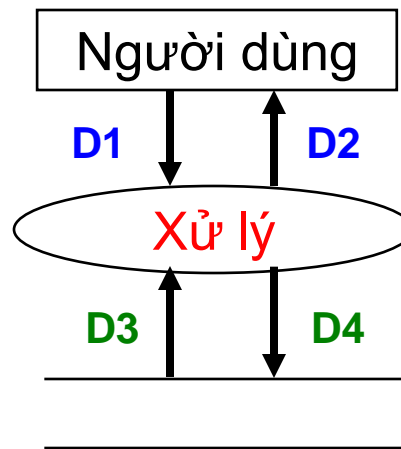
- Lập phiếu nhập, lập phiếu xuất
- Tra cứu hàng hóa

# Thiết kế Xử lý



# Thiết kế xử lý

- ❖ Mục tiêu: Mô tả chi tiết hệ thống các hàm xử lý của phần mềm



Mô tả hàm xử lý

- Tổ chức: đơn thể/đối tượng
- Thông tin chi tiết:
  - Tên
  - Tham số
  - Kết quả
  - Dữ liệu toàn cục
- Sự phối hợp

## ❖ Kết quả:

- Danh cách các đơn thể/đối tượng
- Danh sách các hàm của mỗi đơn thể/đối tượng
- Các sơ đồ phối hợp

## ❖ Yêu cầu thiết kế:

- Tính đúng đắn
- Tính dễ bảo trì
- Tính tái sử dụng
- Tính dễ mang chuyển

## ❖ Kỹ thuật thiết kế:

- Phân rã/tích hợp
- Tham số hóa
- Đối tượng hóa

## ❖ Sơ đồ phối hợp:

- Mô tả **cách thức phối hợp** (gọi thực hiện) **giữa các hàm**

# Thiết kế xử lý

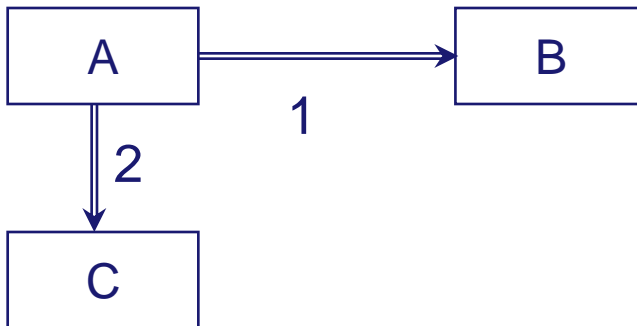
## ❖ Ký hiệu:



Hàm xử lý



Hàm của đơn thể



A có gọi đến B, C theo thứ tự, không chuyển tham số, không nhận kết quả

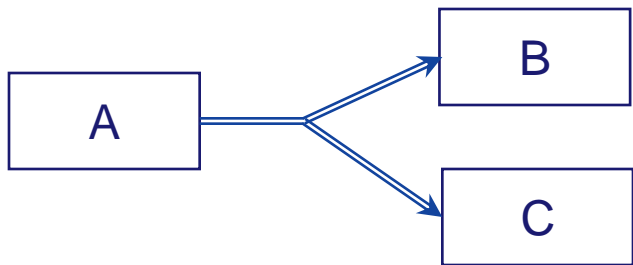


A gọi đến B có chuyển tham số, không nhận kết quả

# Thiết kế xử lý



A gọi đến B không chuyển tham số, nhưng nhận kết quả



A gọi đến B hoặc C



A gọi đến B nhiều lần (ít nhất là 0 lần)



A gọi đến B nhiều lần (ít nhất là 1 lần)

**Chú ý:** Có n biến cố phải xử lý → Tương ứng có n sơ đồ phối hợp



# Ví dụ

- ❖ Xét màn hình tiếp nhận học sinh mới như sau:

**Tiếp nhận học sinh**

Họ tên  Nam

Ngày sinh  Lớp  ▼

Địa chỉ

Danh sách học sinh đã tiếp nhận

STT	Mã HS	Tên HS	Giới tính	Ngày sinh
...	...	...	...	...

## ❖ Hãy:

- Mô tả các biến cố
- Lập danh sách các hàm xử lý
- Lập các sơ đồ phối hợp (Khi có ít nhất hai hàm tham gia trong cùng một biến cố)

# Ví dụ - Mô tả biến cố

- ❖ Biến cố 0:
  - Khởi động màn hình
- ❖ Biến cố 1:
  - Kiểm tra tuổi học sinh hợp lệ (tuổi từ 15 đến 20)
- ❖ Biến cố 2:
  - Khi chọn một lớp học trên combobox
- ❖ Biến cố 3:
  - Kiểm tra dữ liệu hợp lệ và ghi

# Ví dụ - Danh sách các biến cố

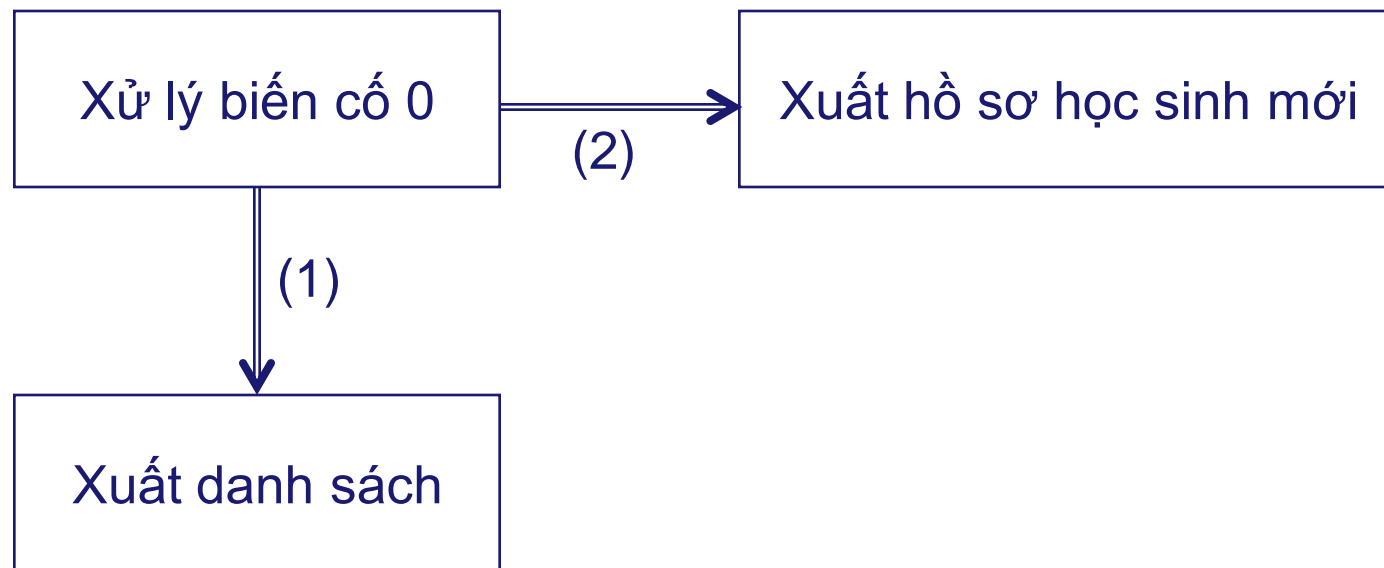
BC	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khởi động màn hình	<ul style="list-style-type: none"><li>- Đọc danh sách lớp, danh sách học sinh, tham số</li><li>- Xuất danh sách lớp, danh sách học sinh, hồ sơ học sinh mới</li></ul>	
1	Kết thúc nhập ngày sinh	<ul style="list-style-type: none"><li>- Kiểm tra ngày sinh hợp lệ và xuất thông báo lỗi nếu không hợp lệ</li></ul>	Tuổi theo qui định 15→20
2	Kết thúc chọn lớp	<ul style="list-style-type: none"><li>- Ghi nhận vị trí của lớp được chọn trong danh sách lớp</li></ul>	Chuẩn bị khi ghi hồ sơ
3	Nhút nút ghi	<ul style="list-style-type: none"><li>- Kiểm tra hồ sơ hợp lệ</li><li>- Nếu hợp lệ thì nhập hồ sơ học sinh và ghi hồ sơ học sinh. Xuất thông báo</li></ul>	Mã và tên phải khác rỗng

# Ví dụ - Danh sách các hàm xử lý

STT	Tên hàm	Tham số	Kết quả	Ý nghĩa	Ghi chú
...					

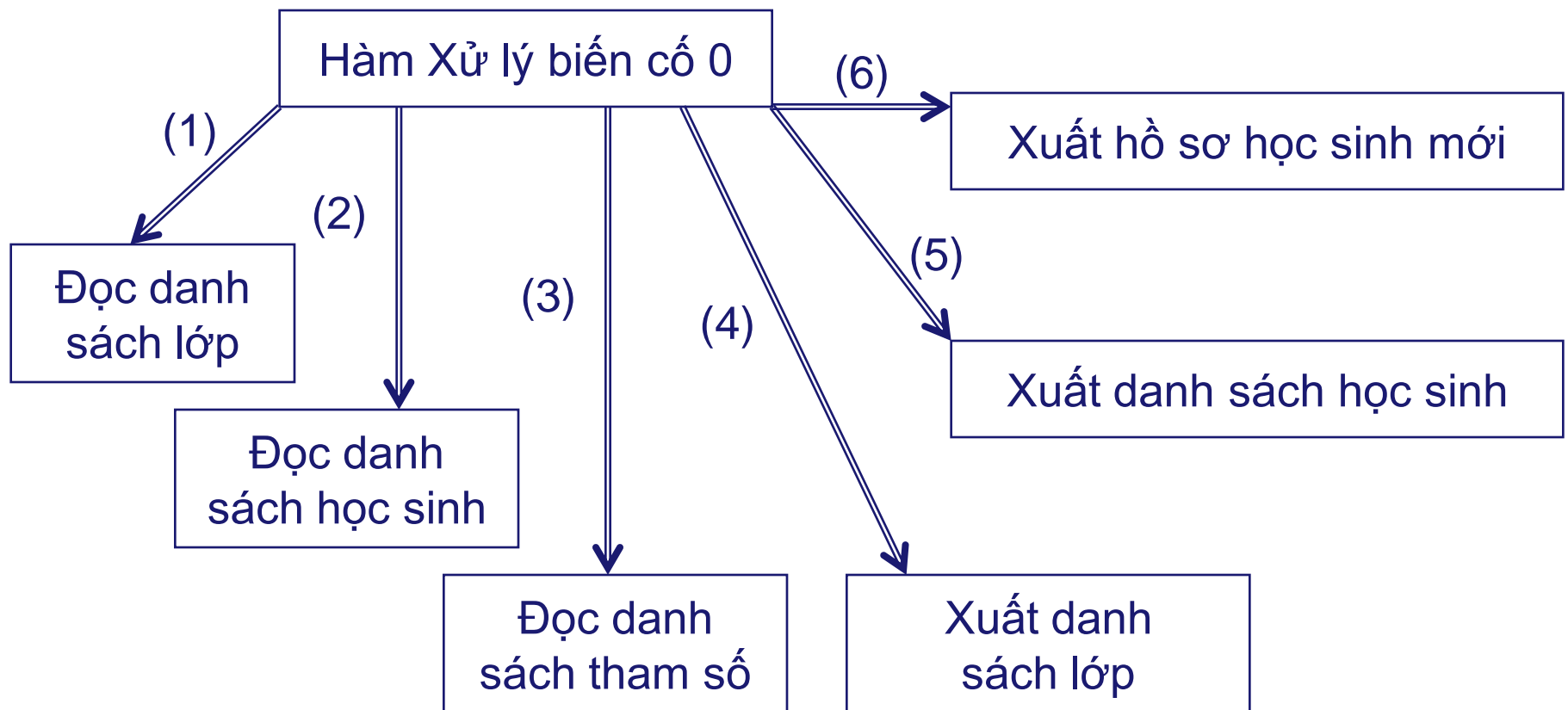
# Ví dụ - Lập sơ đồ phối hợp

## ❖ Xử lý biến cố 0

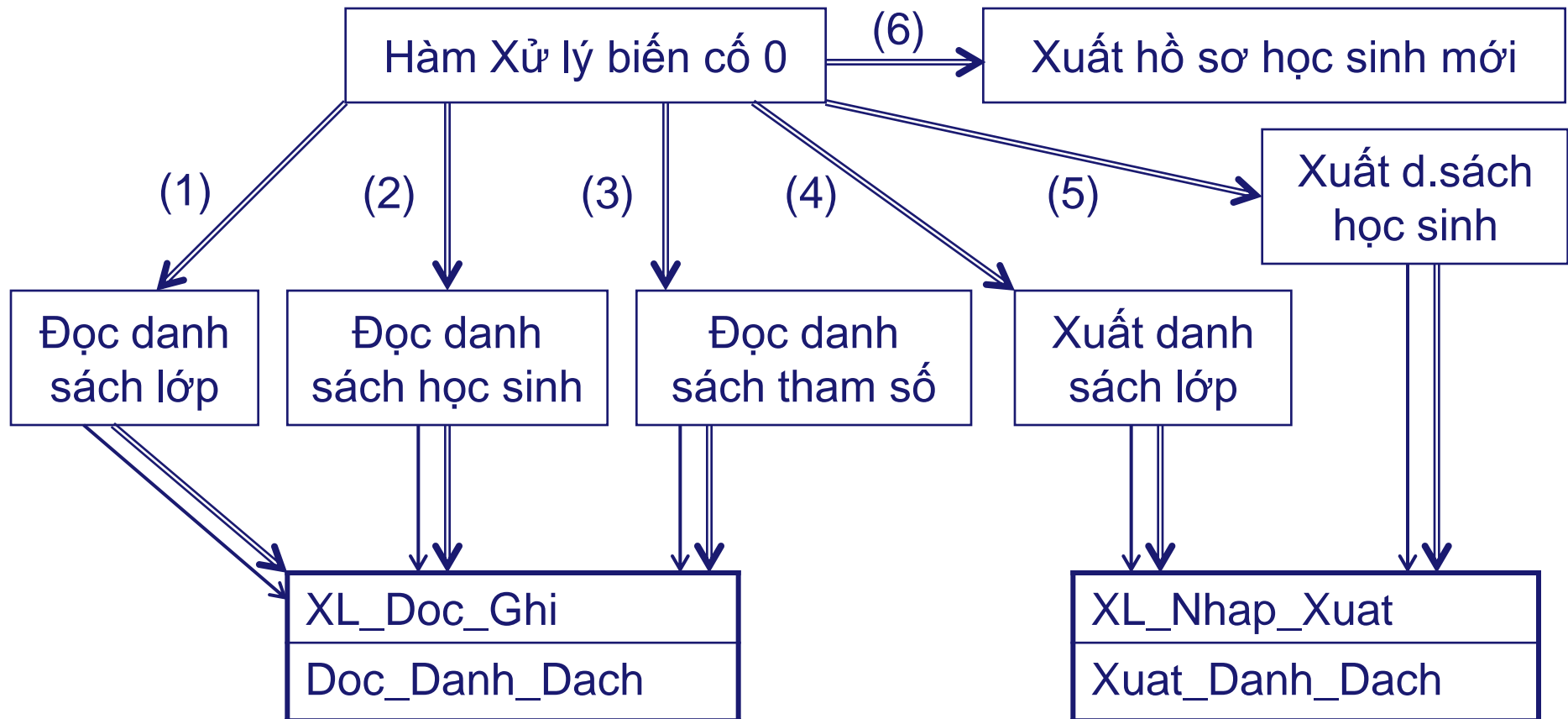


# Ví dụ - Lập sơ đồ phối hợp

## ❖ Xử lý biến cố 0: **Phân rã hàm**



# Ví dụ - Lập sơ đồ phối hợp





# Ví dụ - Lập sơ đồ phối hợp

- ❖ Xử lý biến cố 1
  - Dùng 1 hàm
- ❖ Xử lý biến cố 2
  - ?
- ❖ Xử lý biến cố 3
  - ?



# Bài tập

1. Lập sơ đồ phối hợp cho các biến cố còn lại trong ví dụ trên.
2. Đánh giá các sơ đồ phối hợp dựa trên các tính chất
  - Tính đúng đắn
  - Tính tái sử dụng
  - Tính dễ bảo trì
  - Tính dễ mang chuyển
  - ...



# Chương 4: Cài đặt Phần mềm



- ❖ Giới thiệu
- ❖ Kỹ năng lập trình
- ❖ Phương pháp
  - Lập trình **cấu trúc**
  - Lập trình **hướng đối tượng**
- ❖ Kiến trúc mô hình **1 lớp, 2 lớp, 3 lớp**
- ❖ Sử dụng công cụ Visual Source Safe để **tổ chức, quản lý, chia sẻ** source code.

- ❖ Cài đặt?
  - Là quá trình chuyển đổi từ **thiết kế** chi tiết sang **mã lệnh**.
- ❖ Lựa chọn **ngôn ngữ lập trình**:
  - Phụ thuộc vào cấu hình máy
  - Phụ thuộc vào số lượng ngôn ngữ lập trình sẵn có
  - Phụ thuộc vào thói quen sử dụng ngôn ngữ lập trình
  - Phụ thuộc vào khách hàng
  - ...
- ❖ **Đánh giá rủi ro** khi chọn ngôn ngữ lập trình



# Kỹ năng lập trình

- ❖ Hiểu rõ ngôn ngữ (language-specific)
- ❖ Sử dụng tên biến thích hợp và có nghĩa
  - Tên biến phải rõ ràng, tránh nhầm lẫn
- ❖ Nên có các chú thích bên trong mô-đun
- ❖ Mã lệnh chuẩn
  - Thống nhất về cách đặt tên Mô-đun, tên hàm, tên biến,...
- ❖ Khả năng tái sử dụng

- ❖ Thông tin tối thiểu của một mô-đun:
  - Tên mô-đun
  - Mô tả vắn tắt các công việc mô-đun phải thực hiện
  - Tên lập trình viên
  - Ngày viết
  - Ngày chỉnh sửa
  - Danh sách các tham số
  - Danh sách các biến
  - ...





# Lập trình tuyến tính

- ❖ Khi các phần mềm còn rất đơn giản:
  - Chương trình được viết tuần tự với các câu lệnh thực hiện từ đầu đến cuối.
- ❖ Tuy nhiên:
  - Khoa học máy tính ngày càng phát triển.
  - Các phần mềm đòi hỏi ngày càng phức tạp và lớn hơn rất nhiều.
- ❖ Phương pháp lập trình tuyến tính kém hiệu quả ?

# Lập trình cấu trúc

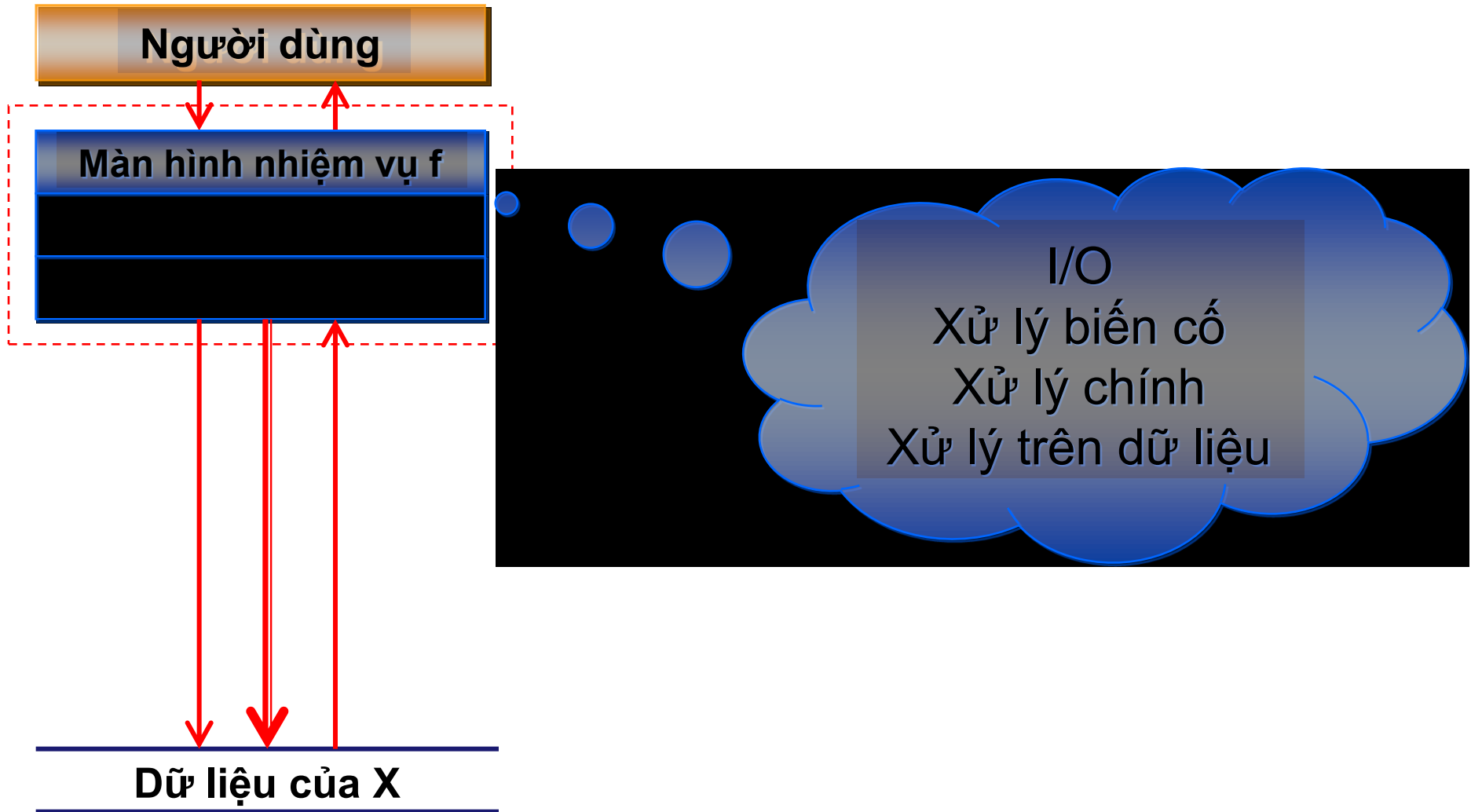
- ❖ Phương pháp lập trình **thủ tục** hay lập trình **cấu trúc**
  - Hệ thống chia các chức năng (hàm) thành các chức năng nhỏ hơn.
  - Chương trình được tổ chức thành các chương trình con
  - Chương trình = Cấu trúc dữ liệu + giải thuật
- ❖ Tổ chức dữ liệu như thế nào?
- ❖ Khi thay đổi cấu trúc dữ liệu?



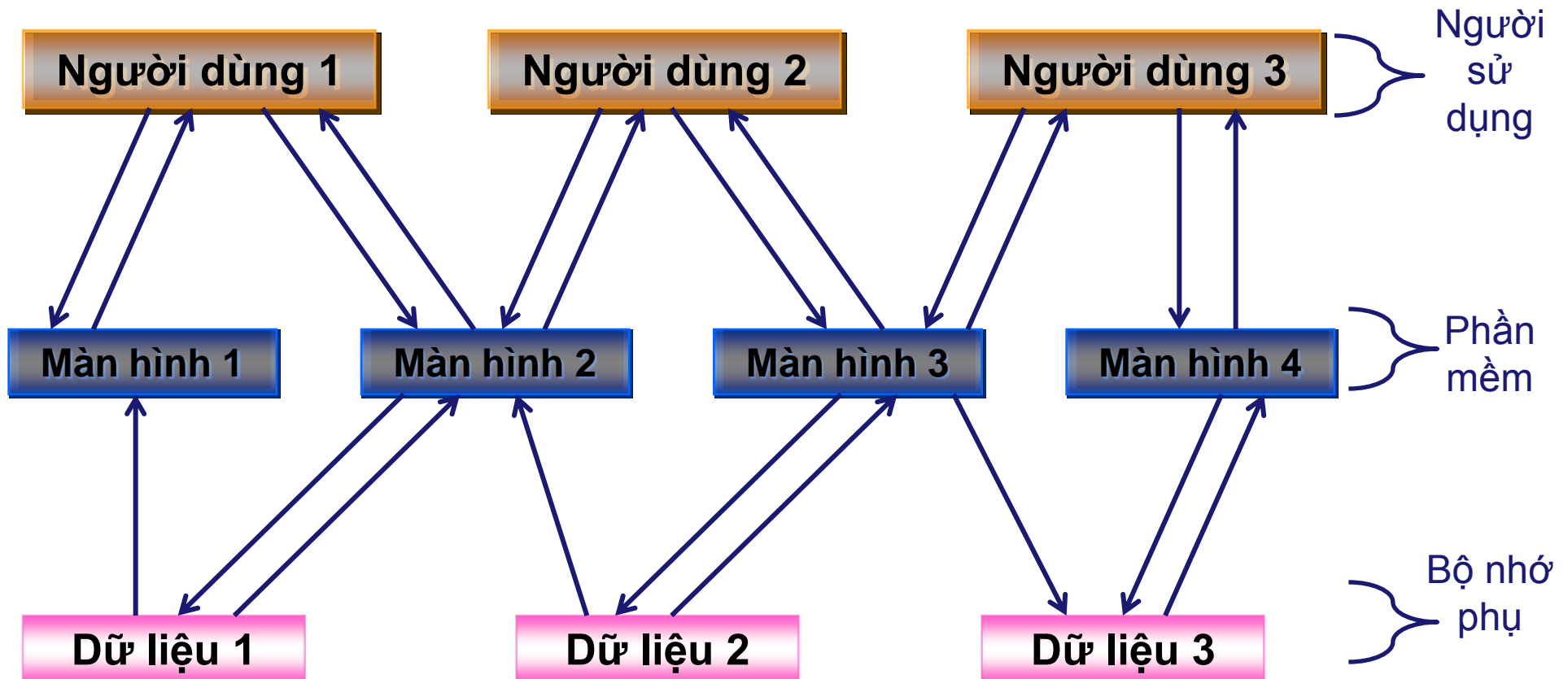
# Lập trình Hướng đối tượng

- ❖ Lập trình hướng đối tượng – Lập trình định hướng đối tượng - OOP
  - Là phương pháp lập trình lấy đối tượng làm nền tảng để xây dựng thuật giải, xây dựng chương trình.
  - Dữ liệu + Hành vi của dữ liệu = Đối tượng
- ❖ Cách tiếp cận gần gũi và thực tế

# Mô hình kiến trúc 1 tầng (1 layer)

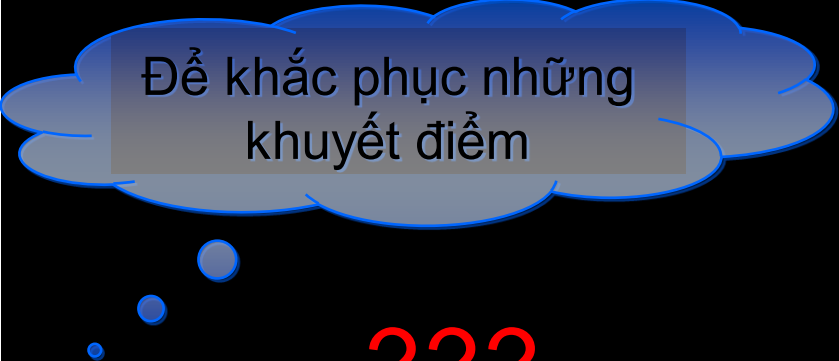


# Mô hình kiến trúc 1 tầng (1 layer)



# Mô hình kiến trúc 1 tầng (1 layer)

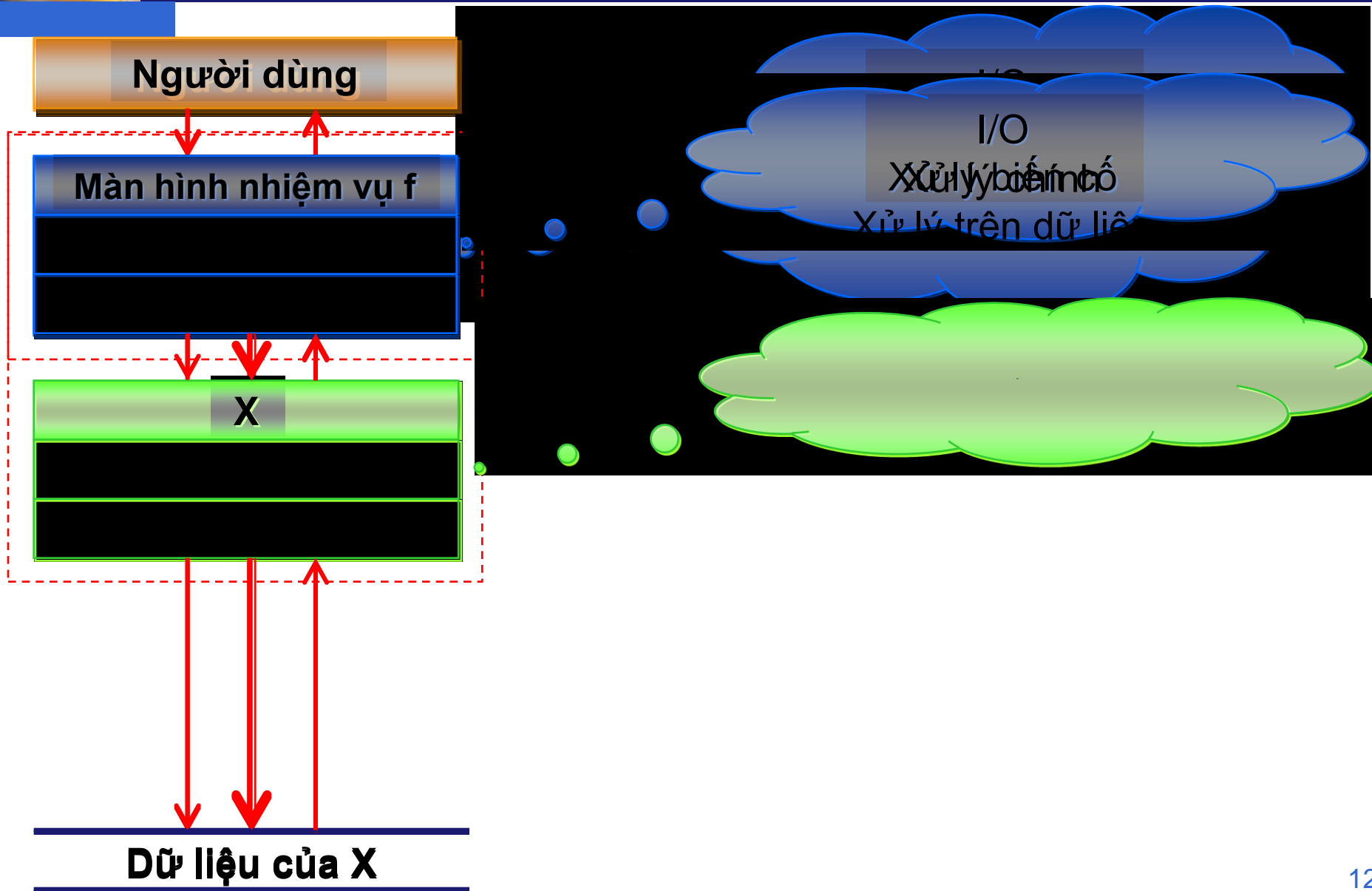
- ❖ Hệ thống trên bao gồm:
  - 3 người sử dụng
  - 4 đơn vị xử lý
  - 3 đơn vị lưu trữ
- ❖ Đặc điểm: Không có sự phân loại các xử lý
- ❖ Ưu điểm: Thiết kế và lập trình nhanh
- ❖ Nhược điểm:
  - Mỗi đơn vị xử lý phức tạp
  - Khó bảo trì
  - Không có tính tái sử dụng



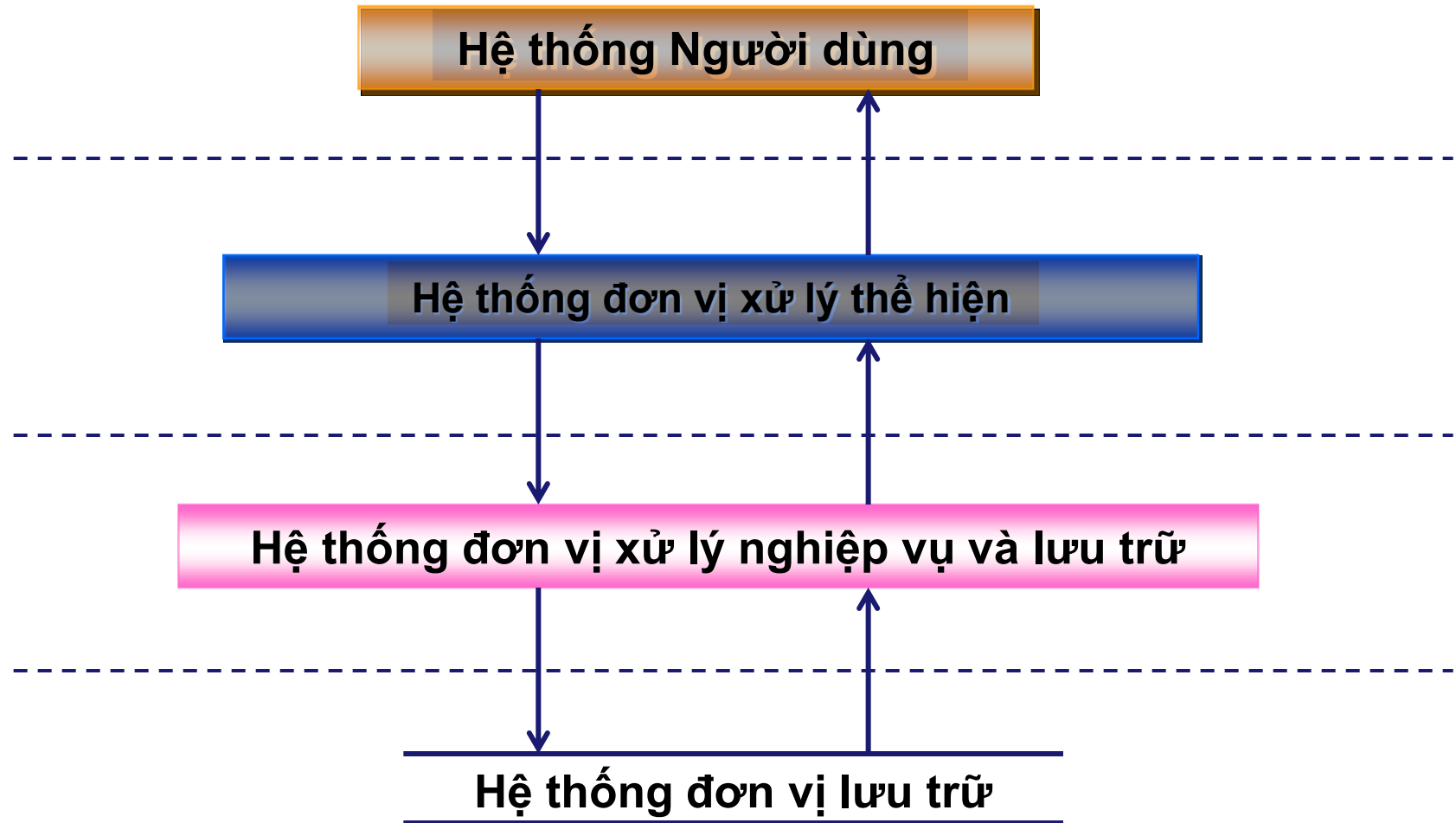
Để khắc phục những  
khuyết điểm

???

# Mô hình kiến trúc 2 tầng (2 layer)



# Mô hình kiến trúc 2 tầng (2 layer)



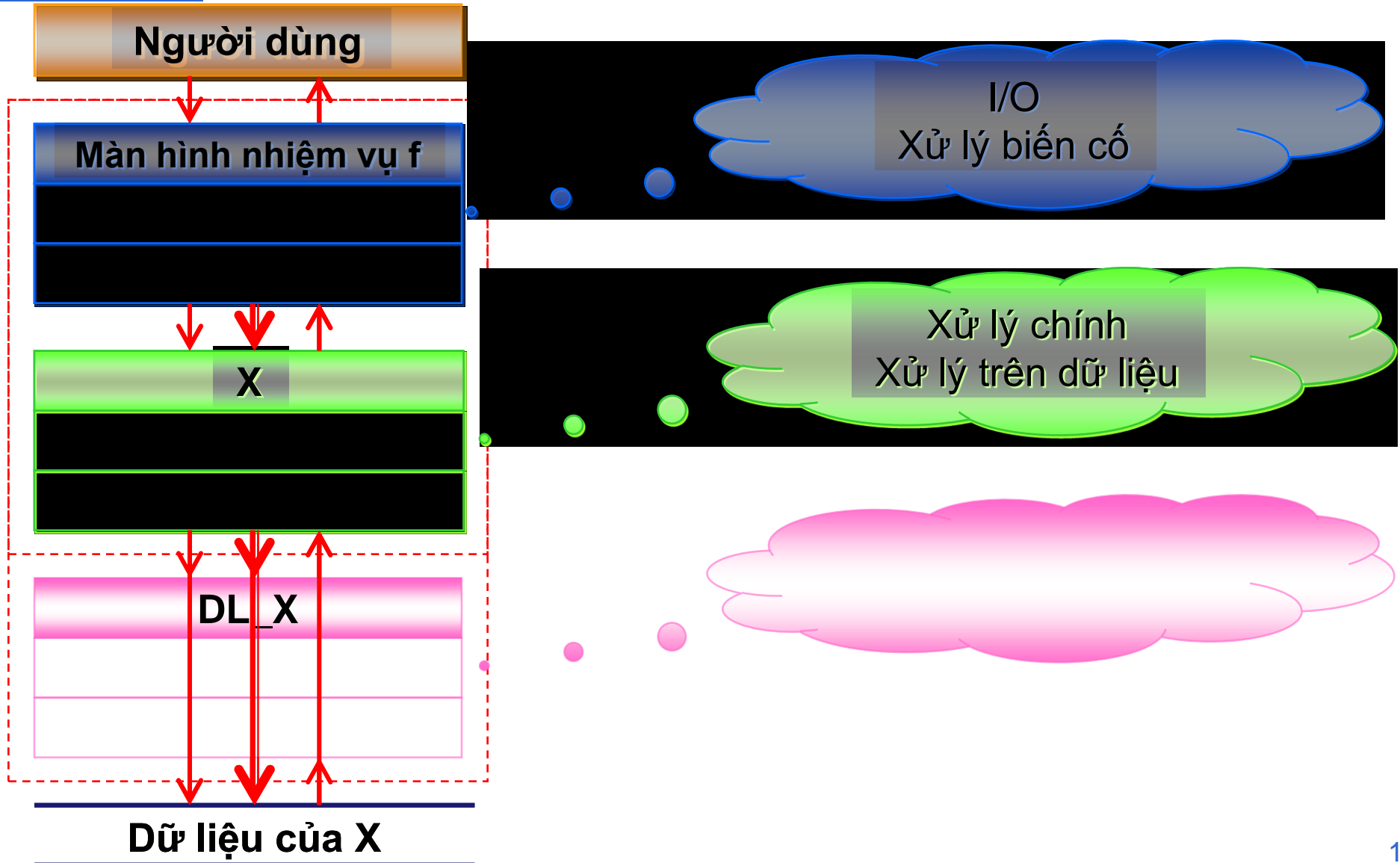




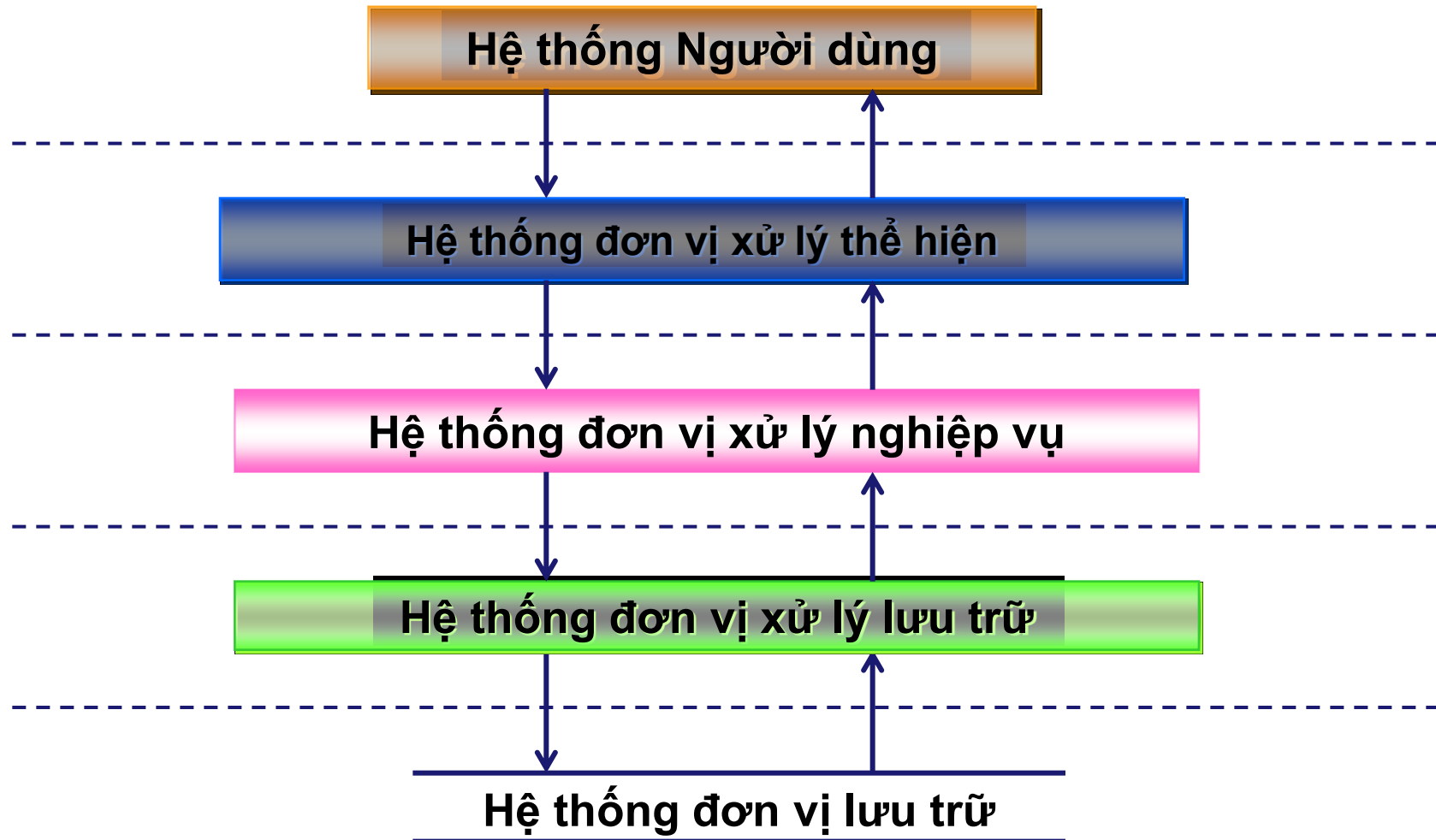
## Mô hình kiến trúc 2 tầng (2 layer)

- ❖ Đặc điểm: Các đơn vị xử lý được phân thành 2 loại
  - Loại 1: Các đơn vị xử lý chuyên biệt về giao tiếp người dùng
  - Loại 2: Các đơn vị xử lý nghiệp vụ (kiểm tra, tính toán), lưu trữ (đọc, ghi)
- ❖ Ưu điểm, khuyết điểm ?

# Mô hình kiến trúc 3 tầng (3 layer)



# Mô hình kiến trúc 3 tầng (3 layer)



# Giới thiệu Microsoft Visual Source Safe

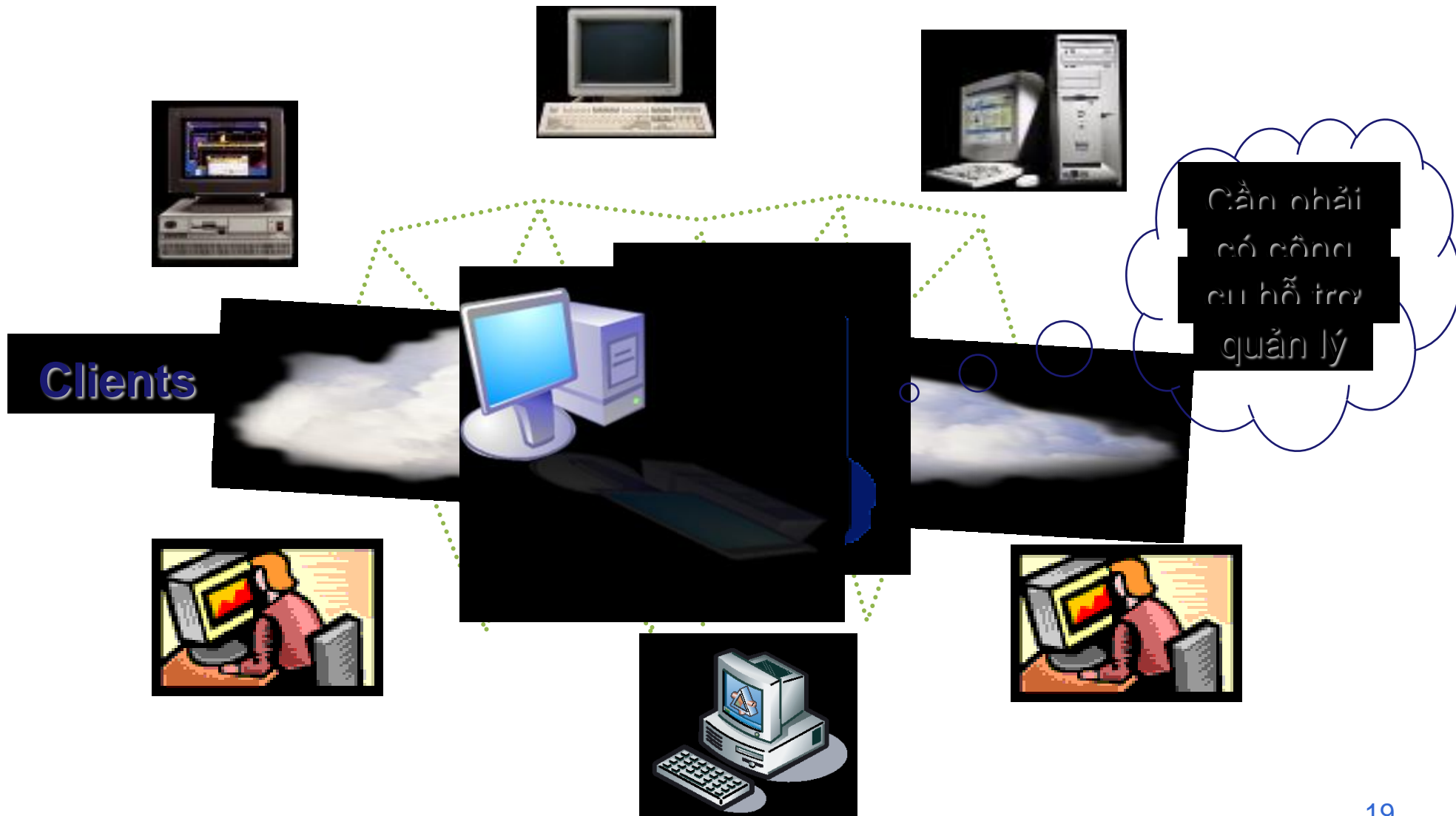


# Giới thiệu

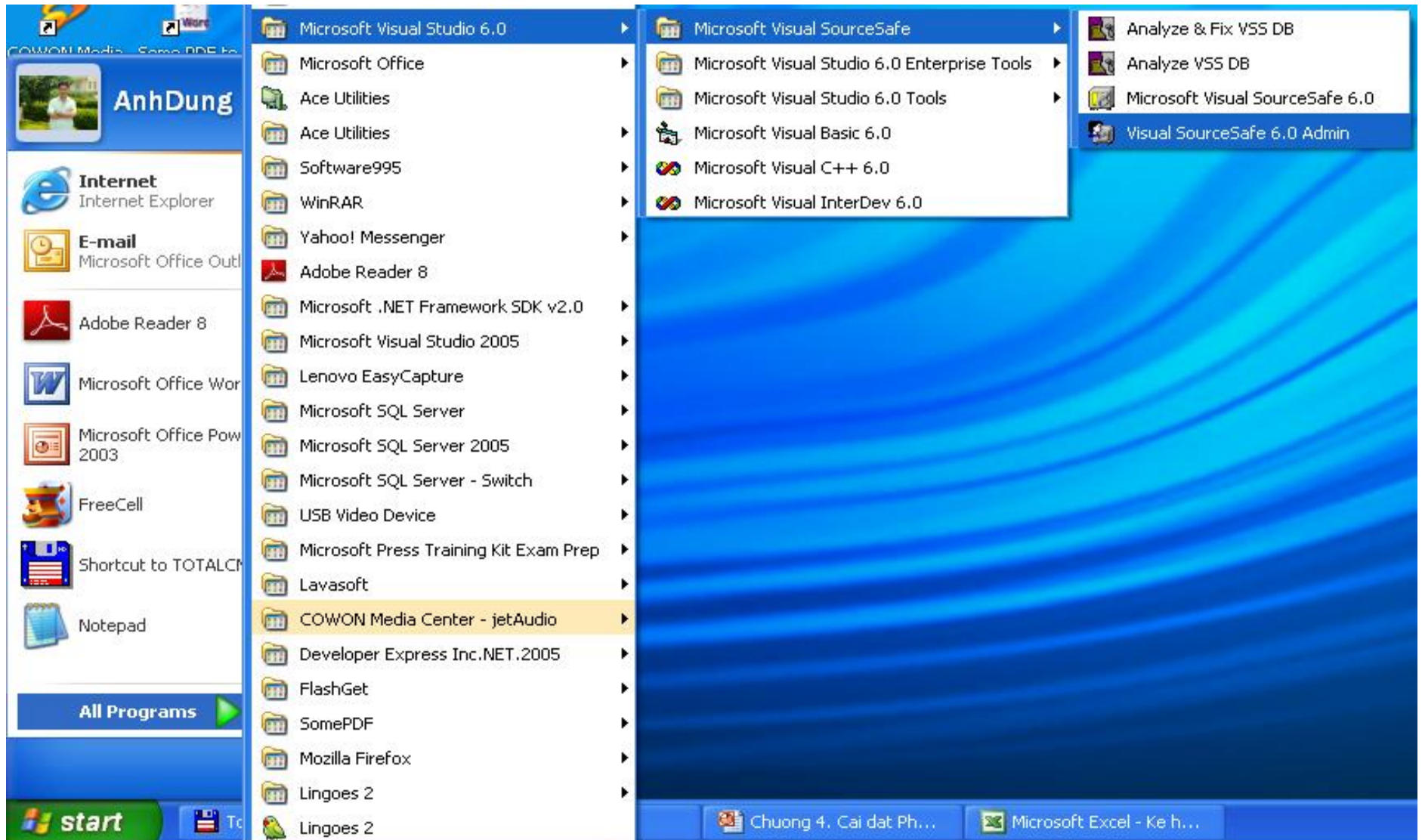
Qui mô dự án phần mềm lớn  
Nhiều người cùng tham gia  
Tổ chức, quản lý ???



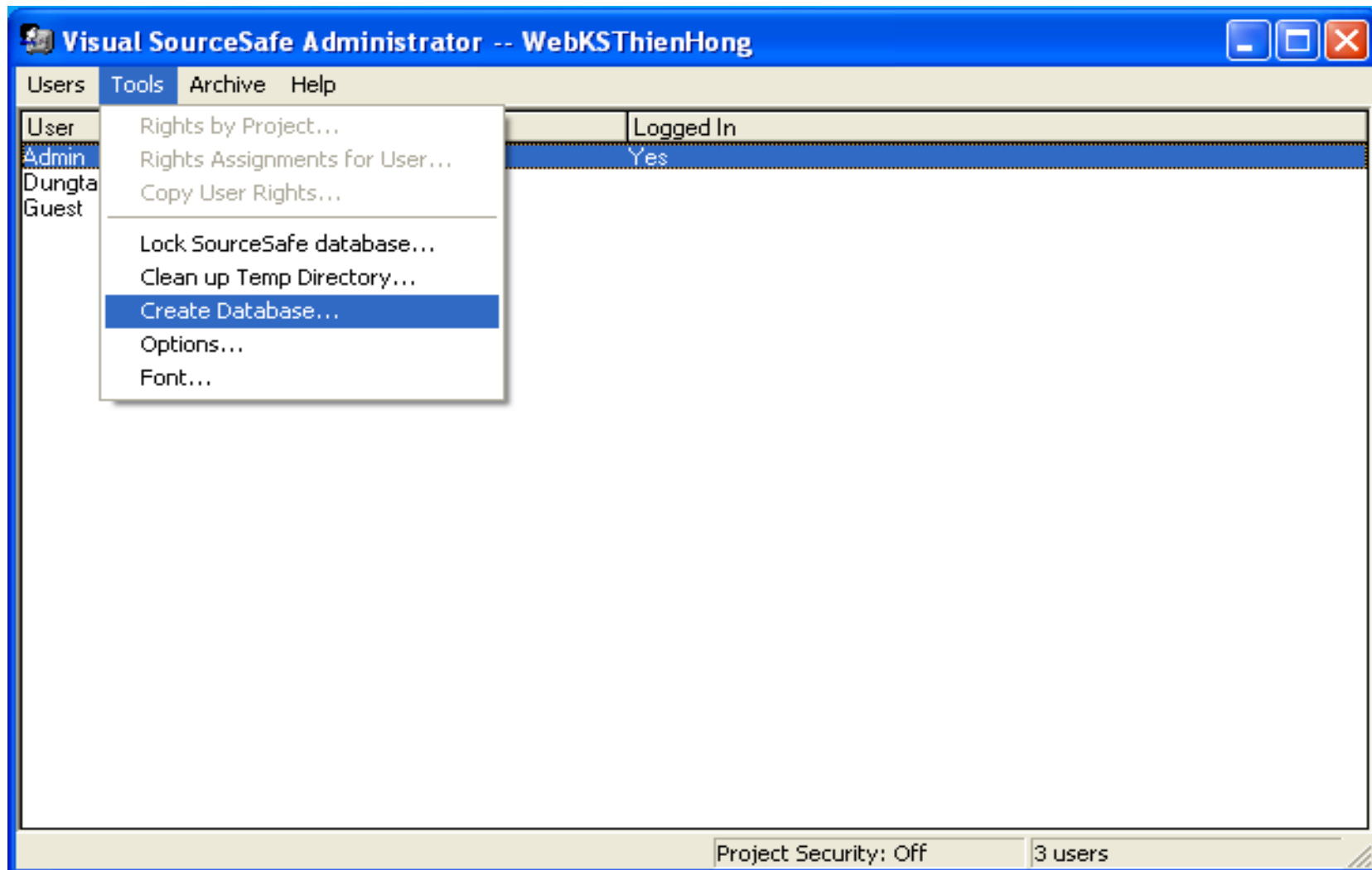
# Giới thiệu



# Giới thiệu

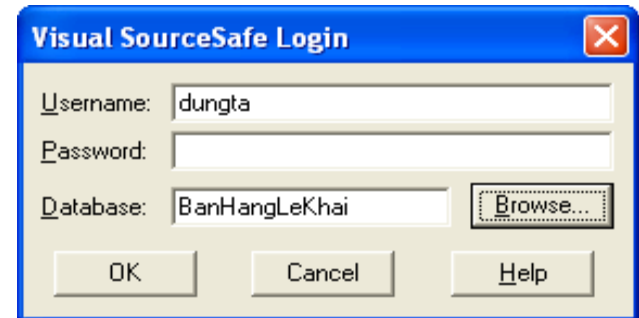
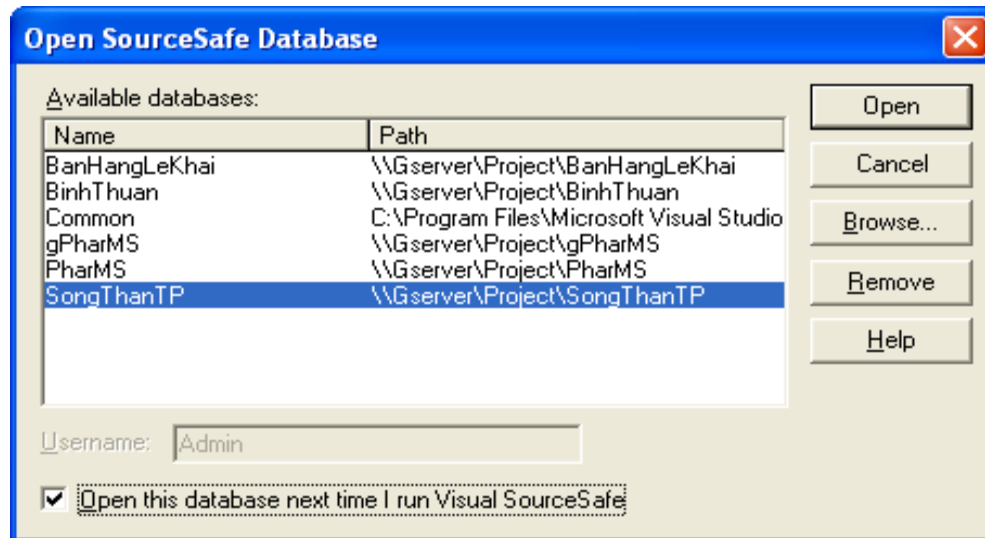


# Visual SourceSafe Admin



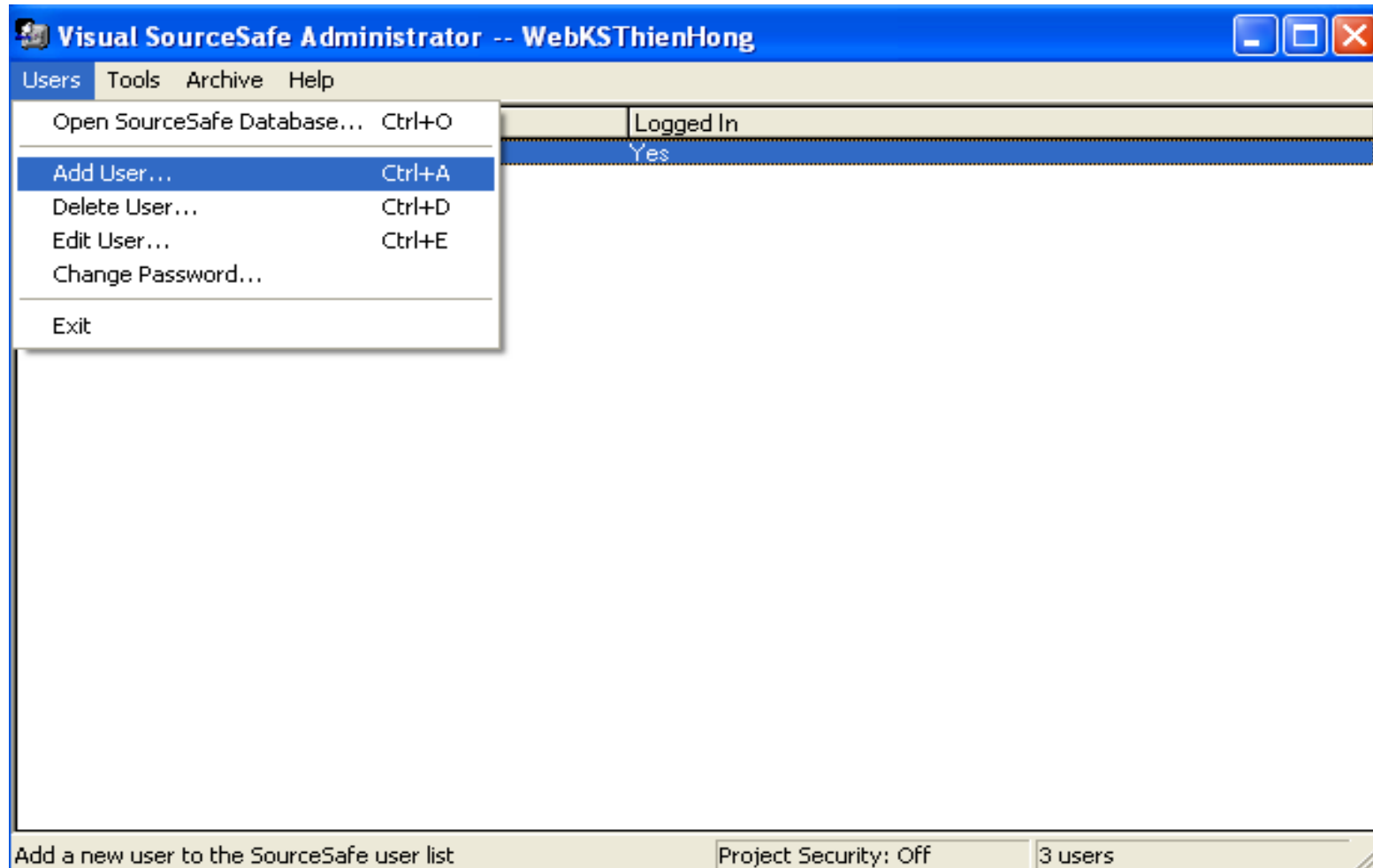


# Visual SourceSafe Admin



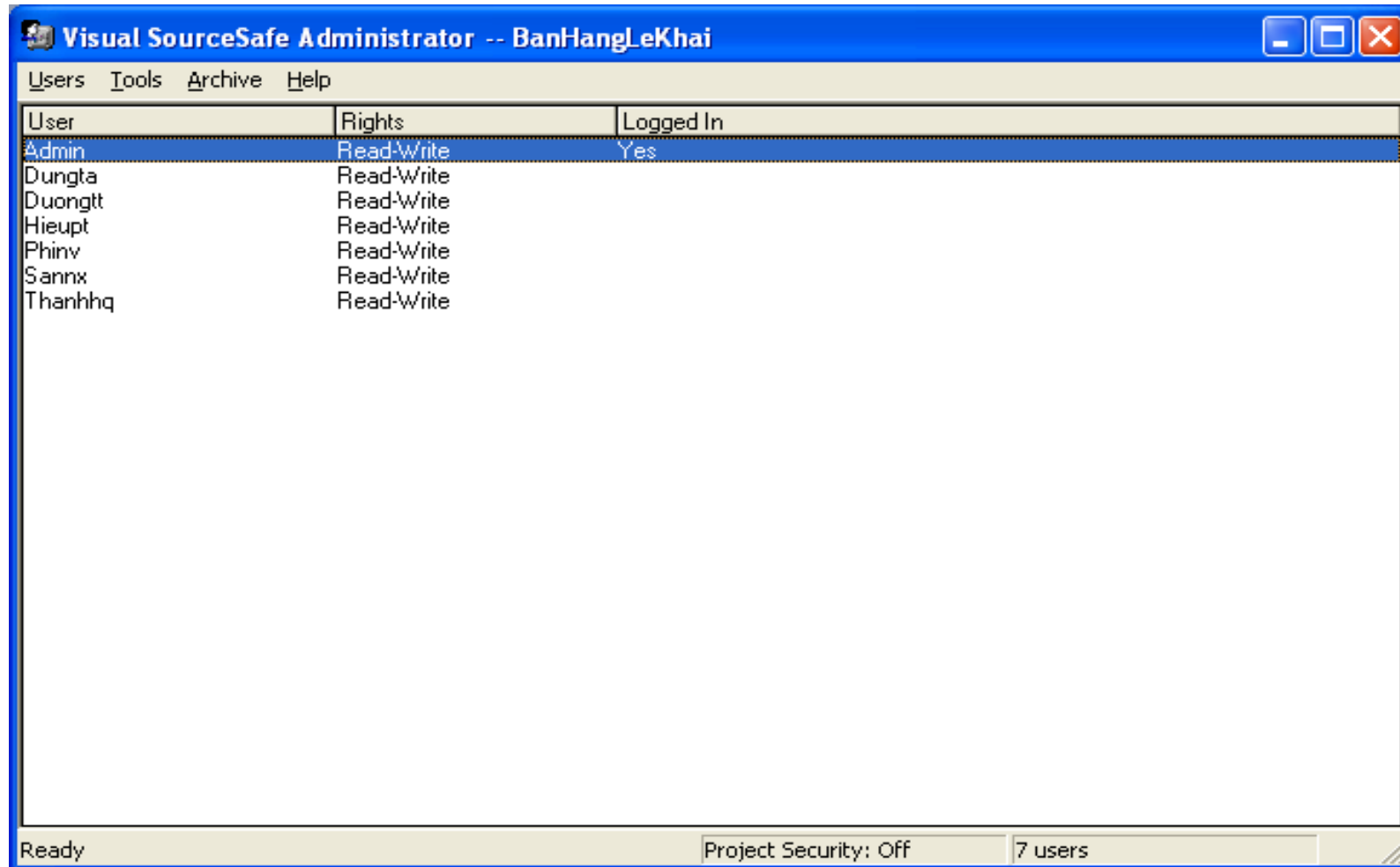
# Visual SourceSafe Admin

## ❖ Quản lý user

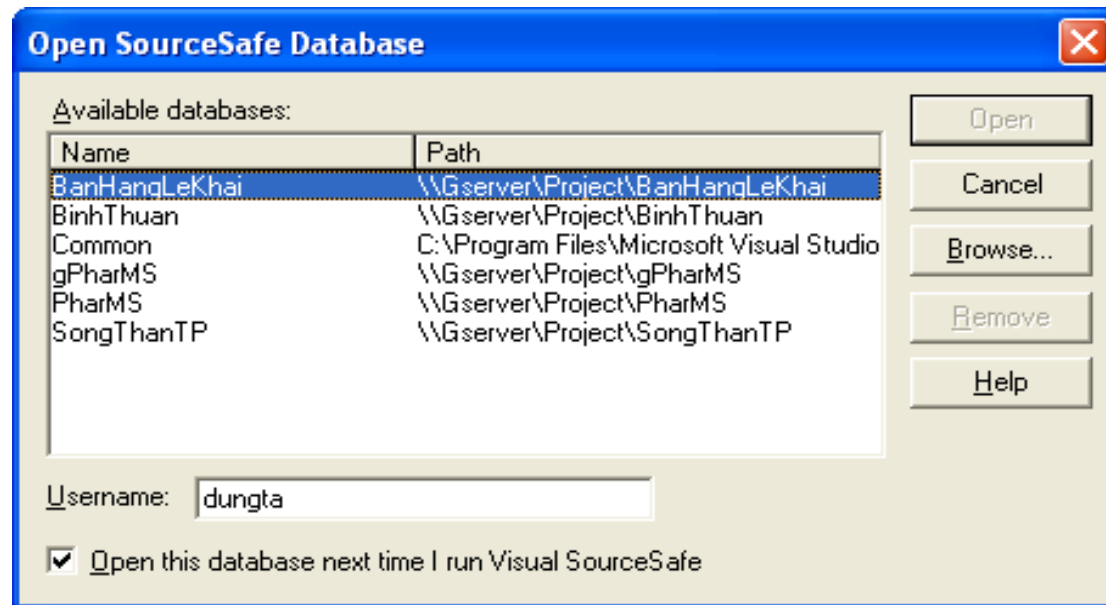


# Visual SourceSafe Admin

## ❖ Quản lý user



# Microsoft Visual SourceSafe



# Microsoft Visual SourceSafe

The screenshot shows the Microsoft Visual SourceSafe Explorer interface. The window title is "Visual SourceSafe Explorer -- BanHangLeKhai". The menu bar includes File, Edit, View, SourceSafe, Tools, Web, and Help. The toolbar contains various icons for file operations. The left pane shows a project tree with the following structure:

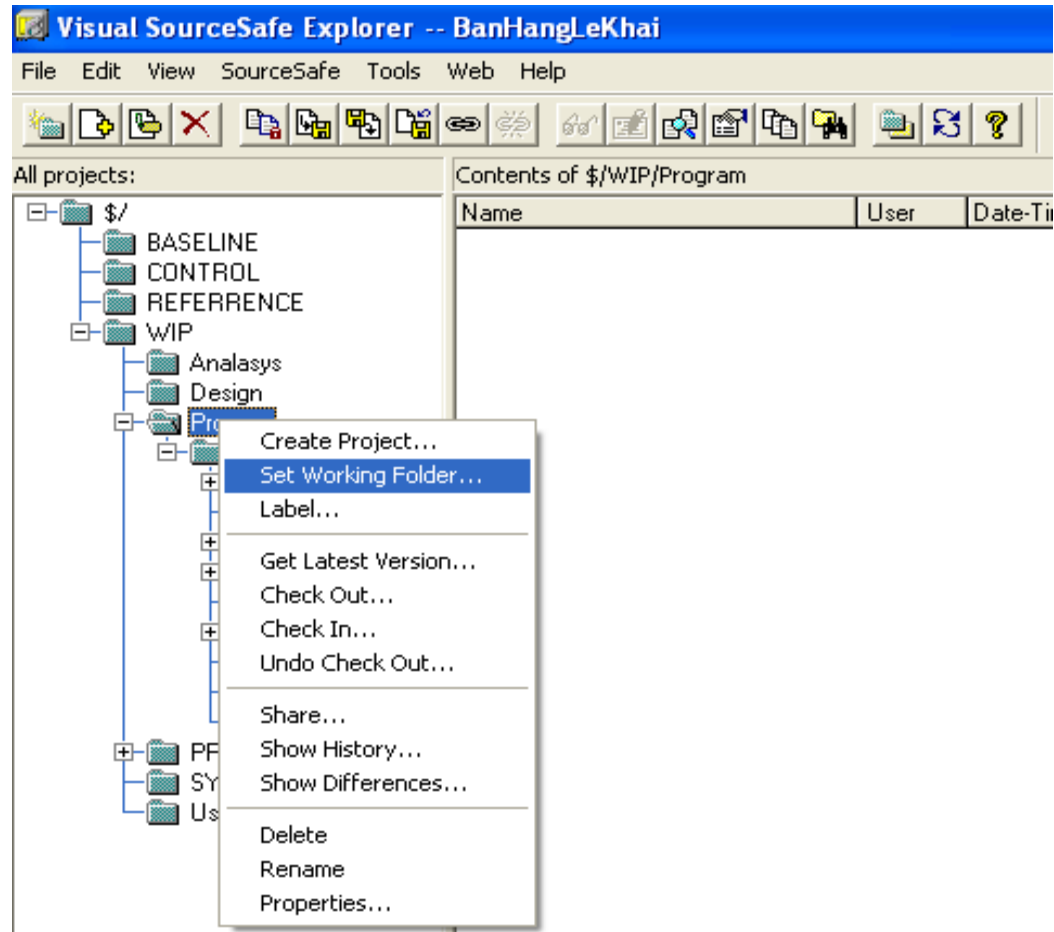
- \$/
  - BASLINE
  - CONTROL
  - REFERENCE
  - WIP
    - Analasys
    - Design
    - Program
      - PharMS
        - bin
        - Classes
        - Commons
        - Form
        - Icon
        - obj
        - OfficelInterop
        - Report
        - Systems
  - PROJECT MANAGERM
  - SYSTEM TEST
  - Users

The right pane shows a file list for the path `$/WIP/Program/PharMS/Classes`. The table below represents the data shown in the file list:

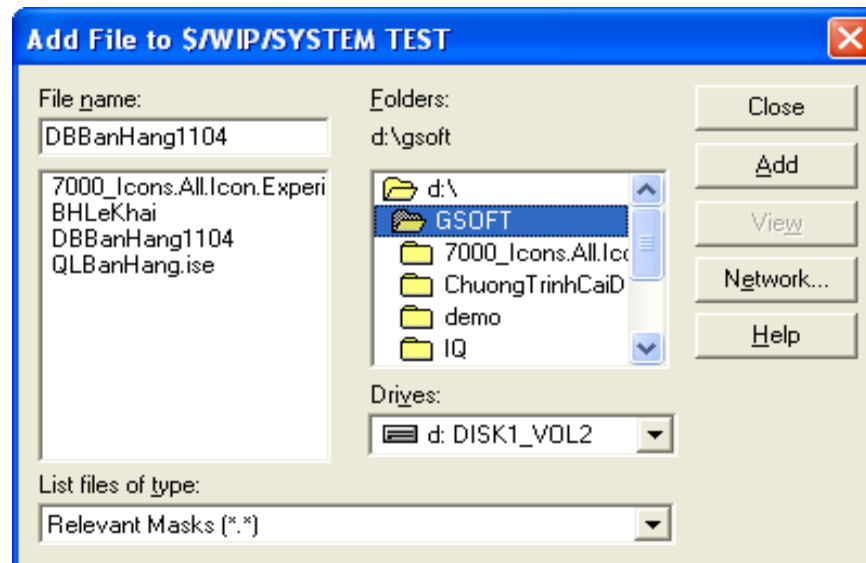
Name	User	Date-Time	Check Out Folder
clsDM_HangHoa.cs	Dungta	11/04/08 9:58p	D:\GSOFT\Projects\Ban...
clsCT_ChungTuKeToan.cs		9/04/08 4:31p	
clsCT_CHUNGTUTONGHOP.cs		9/04/08 5:00p	
clsCT_CHUNGTUVATTU.cs		25/03/08 4:07p	
clsDM_KhachHang.cs		28/03/08 11:18a	
clsDM_KhoHang.cs		17/01/08 1:08p	
clsDM_LoaiChungTu.cs		21/01/08 3:13p	
clsDM_NhapXuat.cs		13/02/08 8:47a	
clsDM_NhomHang.cs		13/02/08 8:47a	
clsDM_NhomKhachHang.cs		21/01/08 1:08p	
clsDM_PhuongXa.cs		13/02/08 8:47a	
clsDM_QuanHuyen.cs		13/02/08 8:47a	
clsDM_TaiKhoan.cs		13/02/08 8:47a	
clsDM_ThamSo.cs		15/02/08 8:14a	
clsDM_TinhThanhPho.cs		13/02/08 8:47a	
clsDMNhanVien.cs		31/03/08 3:50p	
clsDMTRINHDOUC.cs		31/03/08 3:42p	
clsHoaDonBanHang.cs		10/04/08 4:55p	
clsSD_CongNo.cs		11/01/08 10:40p	
clsSD_HangHoa.cs		11/01/08 10:40p	

The status bar at the bottom shows "Ready", the user "dungta", "Sort: User", and "21 Items".

# Microsoft Visual SourceSafe



# Microsoft Visual SourceSafe



# Microsoft Visual SourceSafe

Visual SourceSafe Explorer -- BanHangLeKhai

File Edit View SourceSafe Tools Web Help

All projects:

- \$/
  - BASELINE
  - CONTROL
  - REFERENCE
  - WIP
    - Analasys
    - Design
    - Program
      - PharMS
        - bin
        - Classes
        - Commons
        - Form
        - Icon
        - obj
        - OfficelInterop
        - Report
        - Systems
- PROJECT MANAGERM
- SYSTEM TEST
- Users

Contents of \$/WIP/Program/PharMS/Classes

Name	User	Date-Time	Check Out Folder
clsDM_Han	a	11/04/08 9:58p	D:\GSOFT\Projects\BanHangLe
clsCT_Chur		9/04/08 4:31p	
clsCT_CHU		9/04/08 5:00p	
clsCT_CHU		25/03/08 4:07p	
clsDM_Kha		28/03/08 11:18a	
clsDM_Kho		17/01/08 1:08p	
clsDM_Loi		21/01/08 3:13p	
clsDM_Nha		13/02/08 8:47a	
clsDM_Nho		13/02/08 8:47a	
clsDM_Nho		21/01/08 1:08p	
clsDM_Phu		13/02/08 8:47a	
clsDM_Qua		13/02/08 8:47a	
clsDM_TaiK		13/02/08 8:47a	
clsDM_ThamSo.cs		15/02/08 8:14a	
clsDM_TinhThanhPho.cs		13/02/08 8:47a	
clsDMNhanVien.cs		31/03/08 3:50p	
clsDMTRINHUUOC.cs		31/03/08 3:42p	
clsHoaDonBanHang.cs		10/04/08 4:55p	
clsSD_CongNo.cs		11/01/08 10:40p	
clsSD_HangHoa.cs		11/01/08 10:40p	
clsSD_TaiKhoan.cs		8/04/08 4:53p	





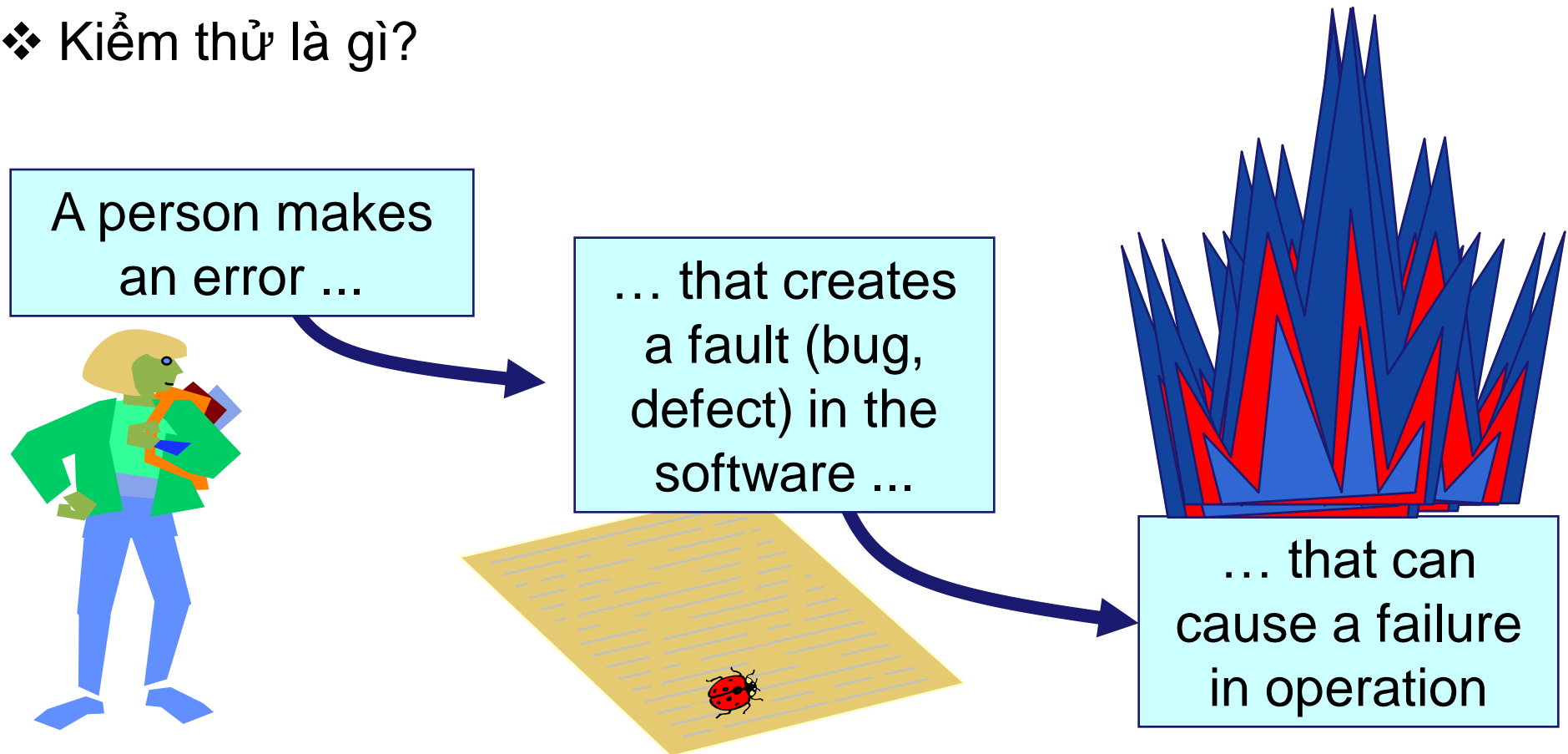
# Chương 5: Kiểm thử (Testing)



- ❖ Khái niệm kiểm thử phần mềm
- ❖ Một số đặc điểm của kiểm thử phần mềm
- ❖ Tại sao kiểm thử lại cần thiết?
- ❖ Quy trình kiểm thử
- ❖ Các mức độ test
- ❖ Kỹ thuật thiết kế test
- ❖ Vai trò của Tester
- ❖ Công việc Tester
- ❖ Tài liệu tham khảo

# Khái niệm kiểm thử phần mềm

## ❖ Kiểm thử là gì?





# Khái niệm kiểm thử phần mềm

❖ Kiểm thử phần mềm là quá trình thực thi phần mềm với mục tiêu tìm ra lỗi

Glen Myers, 1979

→ Khẳng định được chất lượng của phần mềm đang xây dựng

Hetzel, 1988



# Một số đặc điểm kiểm thử PM

- ❖ Kiểm thử phần mềm giúp tìm ra được sự hiện diện của lỗi nhưng không thể chỉ ra sự vắng mặt của lỗi

Dijkstra

- ❖ Mọi phương pháp được dùng để ngăn ngừa hoặc tìm ra lỗi đều sót lại những lỗi khó phát hiện hơn

Beizer

- ❖ Điều gì xảy ra nếu việc kiểm thử không tìm được lỗi trong phần mềm hoặc phát hiện quá ít lỗi
  - Phần mềm có chất lượng quá tốt
  - Quy trình/Đội ngũ kiểm thử hoạt động không hiệu quả

# Tại sao kiểm thử lại cần thiết?

- ❖ Thông thường thì phần mềm **không hoạt động như mong muốn** → **lãng phí tiền bạc, thời gian, uy tín** của doanh nghiệp, thậm chí có thể gây nên thương tích hay cái chết.
- ❖ Ví dụ:
  - Website công ty có nhiều lỗi chính tả trong câu chữ → Khách hàng có thể lãng tránh công ty với lý do công ty trông có vẻ không chuyên nghiệp.
  - Một phần mềm tính toán lượng thuốc trừ sâu dùng cho cây trồng, vì lý do tính sai số lượng lên gấp 10 lần → Nông dân phải bỏ nhiều tiền mua, cây trồng hư hại, môi trường sống, nguồn nước bị ảnh hưởng,.....

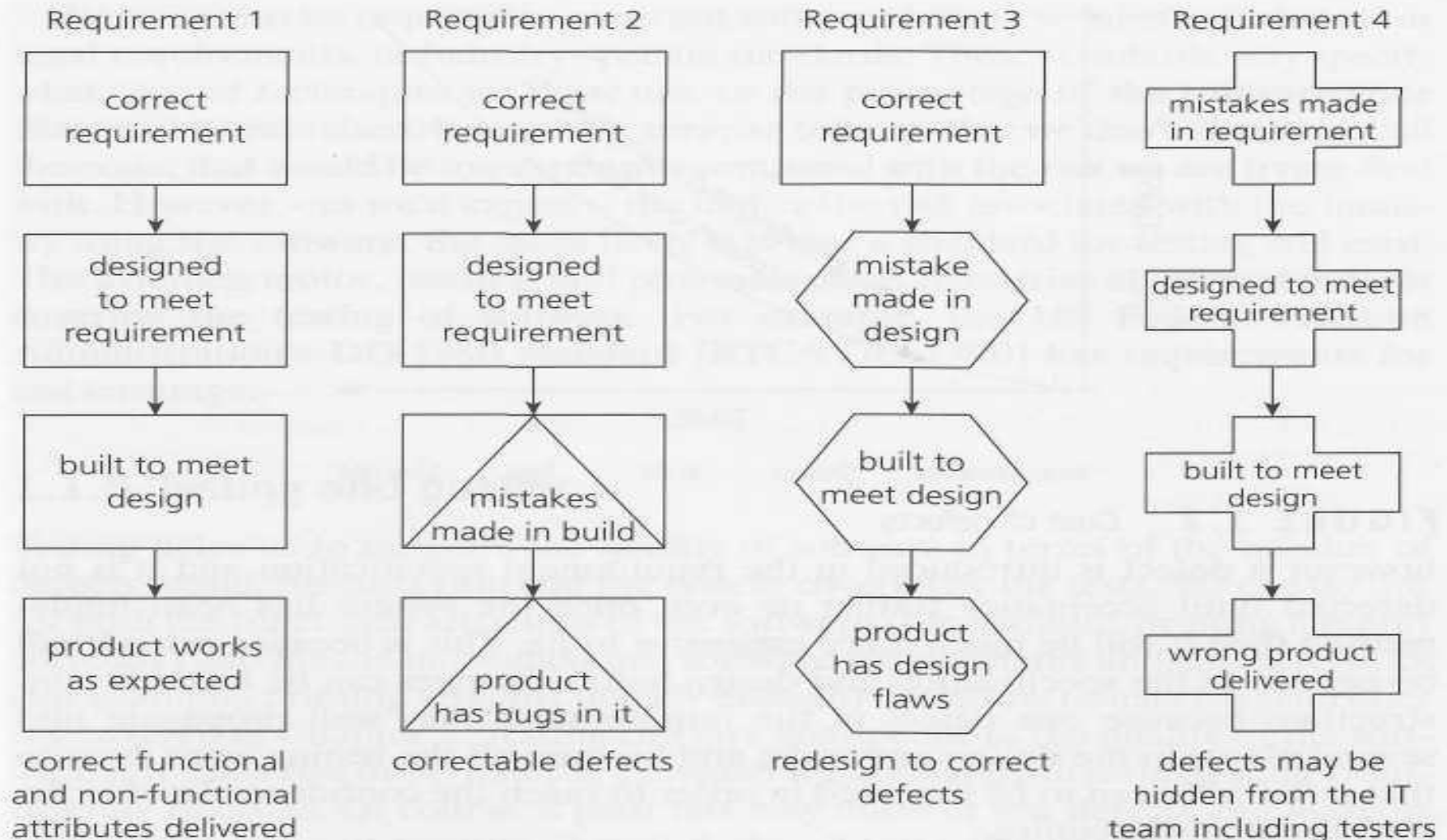


# Tại sao kiểm thử lại cần thiết?

- ❖ Kiểm thử phần mềm → chất lượng phần mềm được nâng cao.
- ❖ Chúng ta có thể đánh giá chất lượng phần mềm dựa vào số lượng lỗi tìm thấy và các đặc tính như: tính đúng đắn, tính dễ sử dụng, tính dễ bảo trì,...
- ❖ Kiểm thử có thể đem lại sự tin tưởng đối với chất lượng phần mềm nếu có ít lỗi hoặc không có lỗi nào được tìm thấy. Nếu lỗi tìm thấy và được sửa thì chất lượng phần mềm càng được tăng → Giảm chi phí trong quá trình phát triển, nâng cấp, bảo trì phần mềm



# Lỗi tăng lên khi nào?

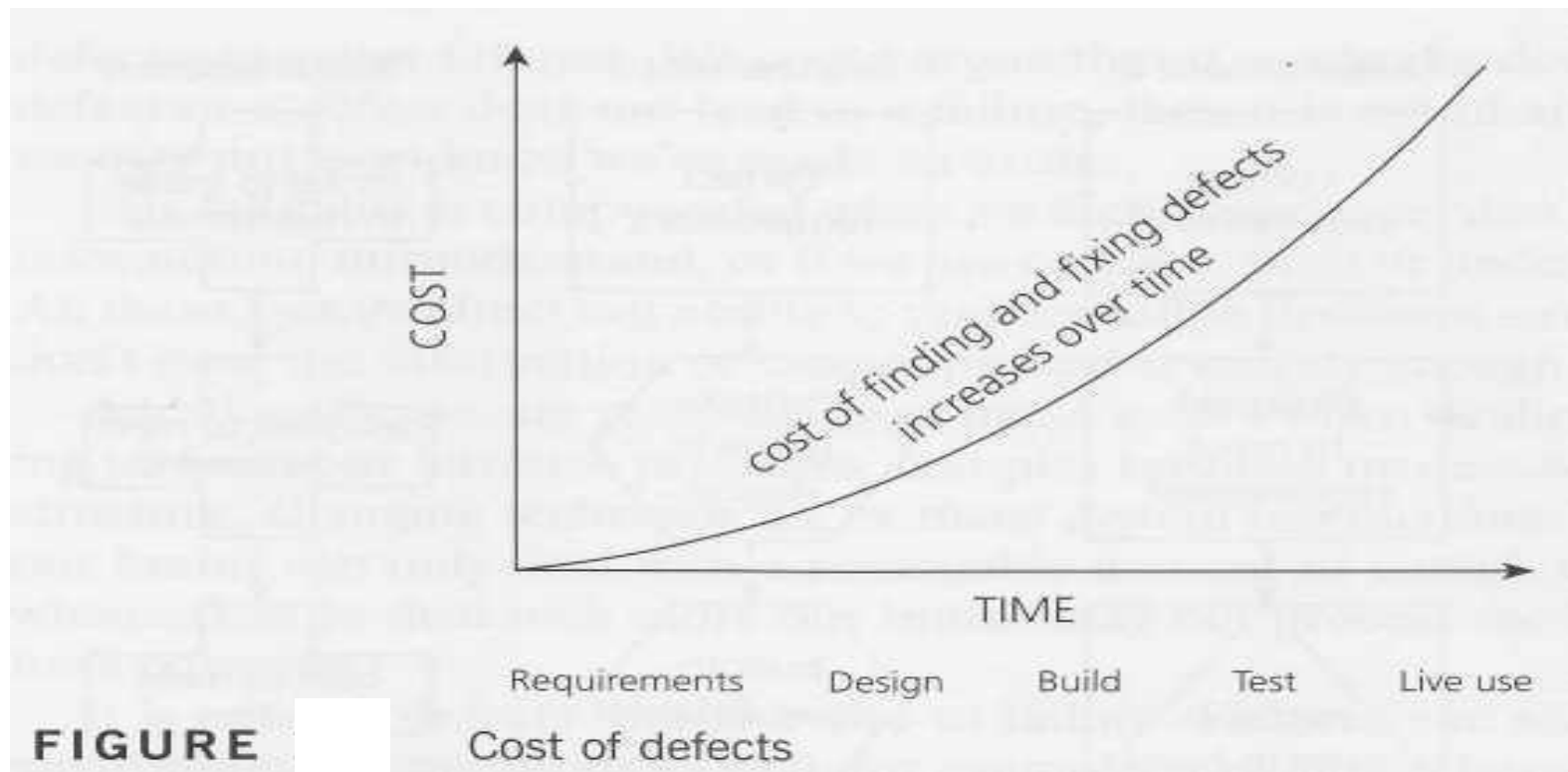


**FIGURE**

Types of error and defect

# Lỗi tăng lên khi nào?

- ❖ Chi phí cho việc tìm thấy và sửa lỗi **tăng dần** trong suốt chu kỳ sống của phần mềm. Lỗi tìm thấy **càng sớm** thì **chi phí** để sửa **càng thấp** và ngược lại.





# Các hoạt động của kiểm thử

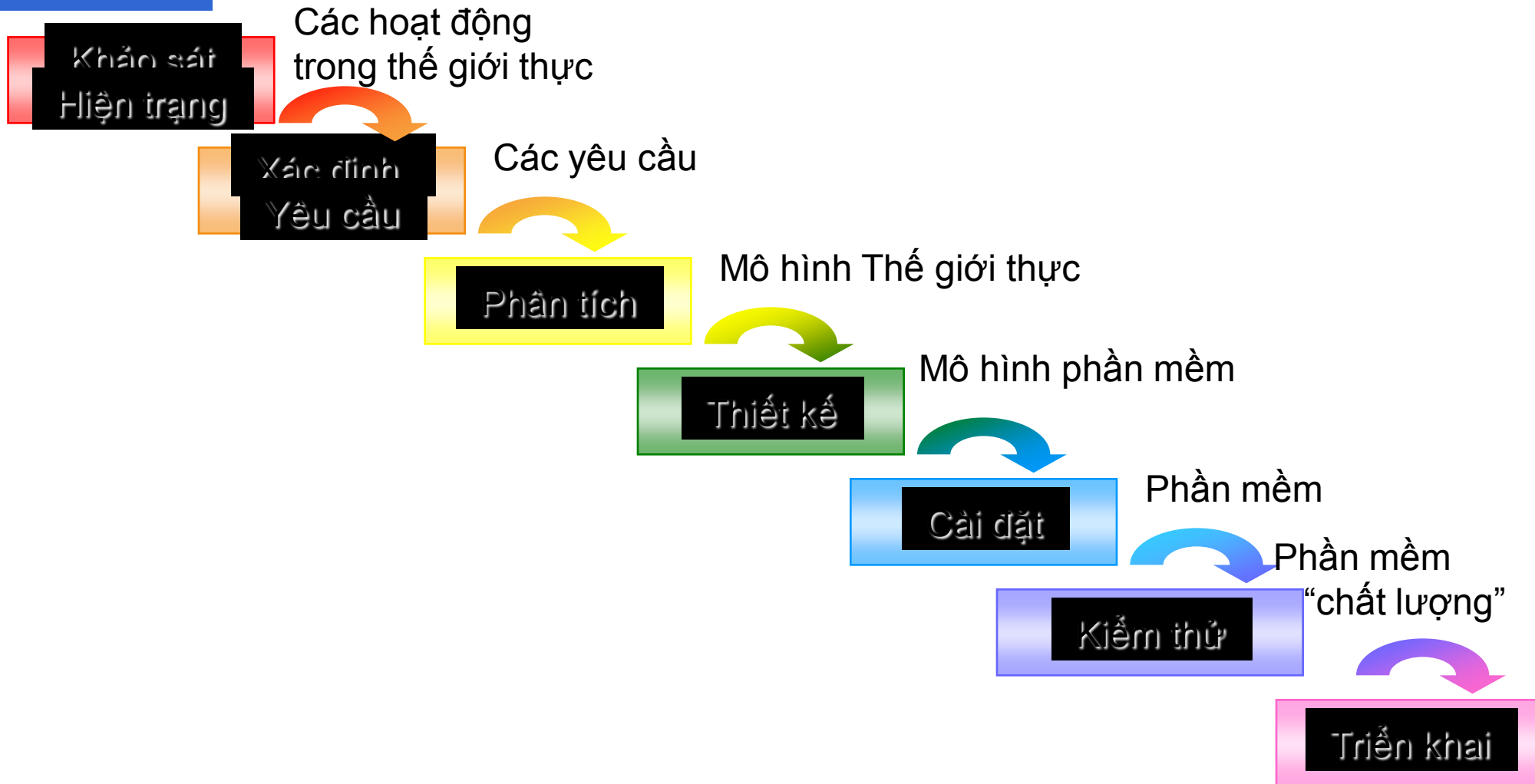
- ❖ Các hoạt động của kiểm thử tồn tại cả trước và sau khi thực thi phần mềm như:
  - Lập kế hoạch test (test plan)
  - Chọn các điều kiện test (test conditions)
  - Thiết kế các trường hợp test (test cases)
  - Kiểm tra kết quả, ước lượng khi nào thì dừng test.
  - Báo cáo kết quả test.



# Vai trò kiểm thử

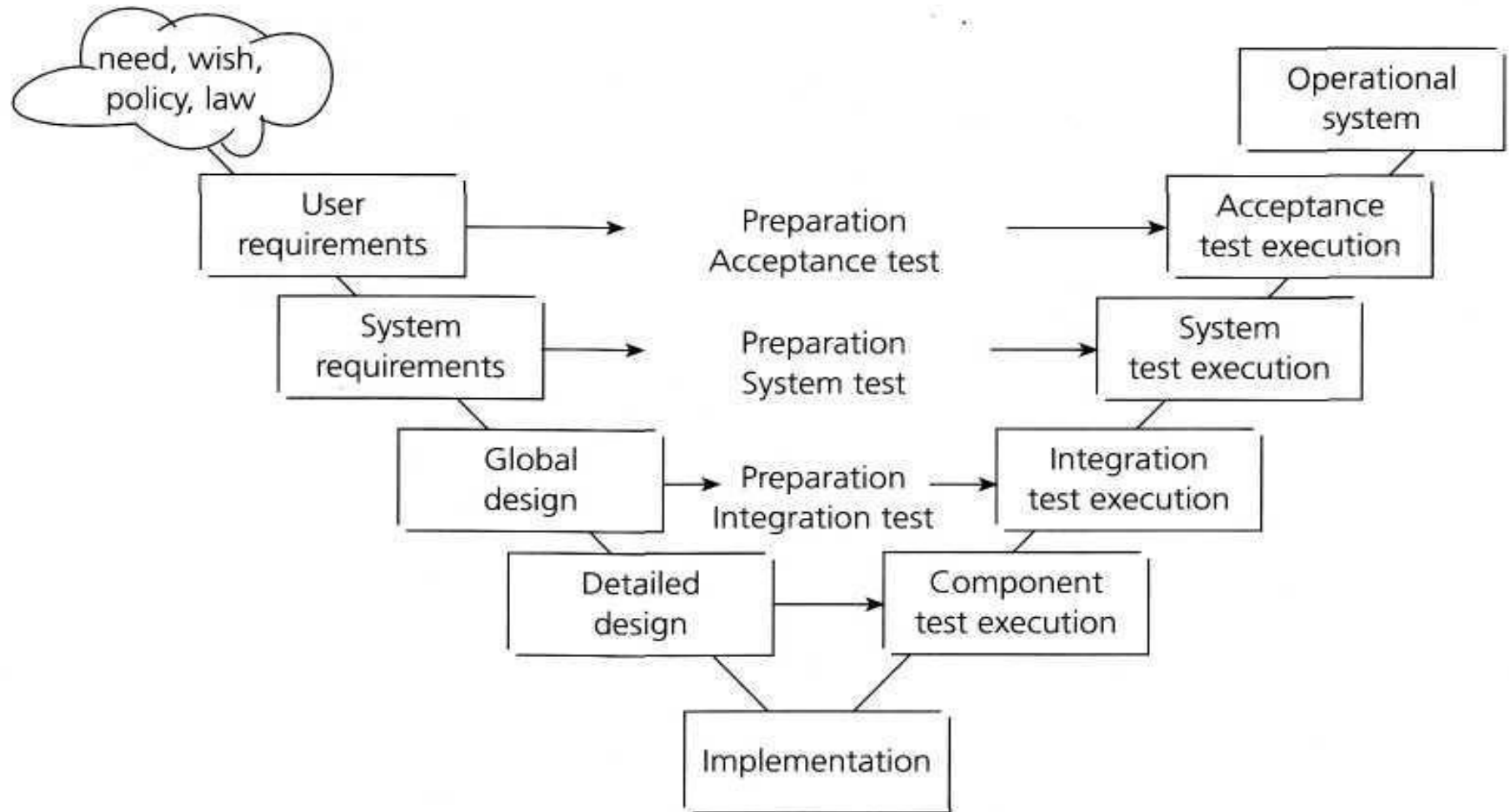
- ❖ Vai trò kiểm thử trong suốt quy trình sống của phần mềm
  - Kiểm thử **không tồn tại độc lập**.
  - Các hoạt động của kiểm thử luôn **gắn liền** với các **hoạt động phát triển phần mềm**.
  - Các **mô hình phát triển phần mềm khác nhau** cần các cách **tiếp cận test khác nhau**.

# Mô hình thác nước



Waterfall

# Mô hình chữ V

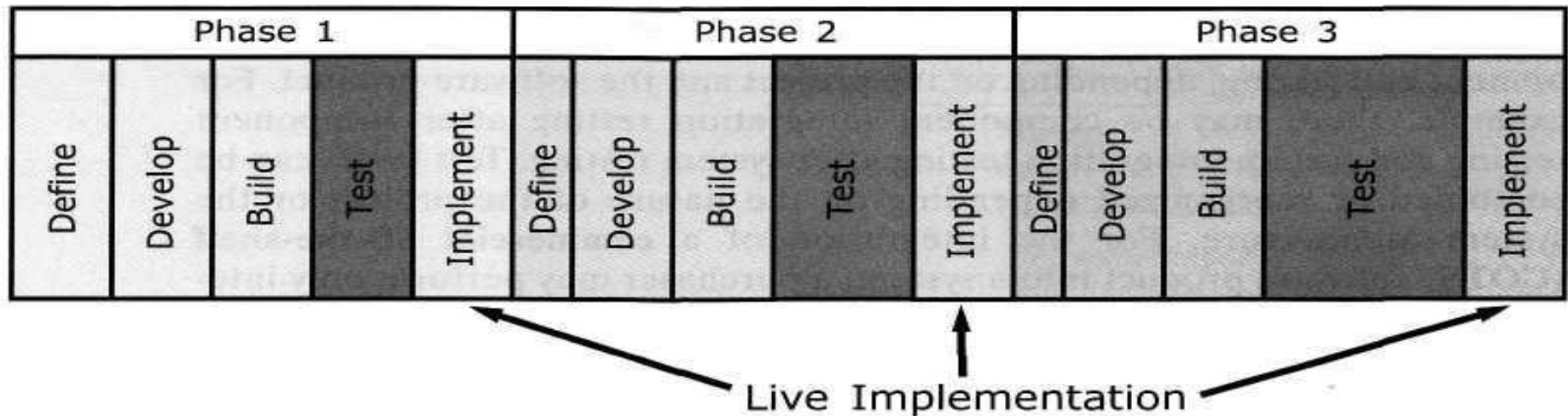


**FIGURE**

V-model

# Mô hình phát triển lặp

- ❖ Chúng ta có thể chia nhỏ phần mềm ra làm **nhiều giai đoạn** thay vì làm một lần từ đầu đến cuối. Mô hình này cần các hoạt động test như: **test chức năng mới**, **test lặp lại** cho những chức năng cũ, và **integration test** cho cả phần cũ và phần mới.

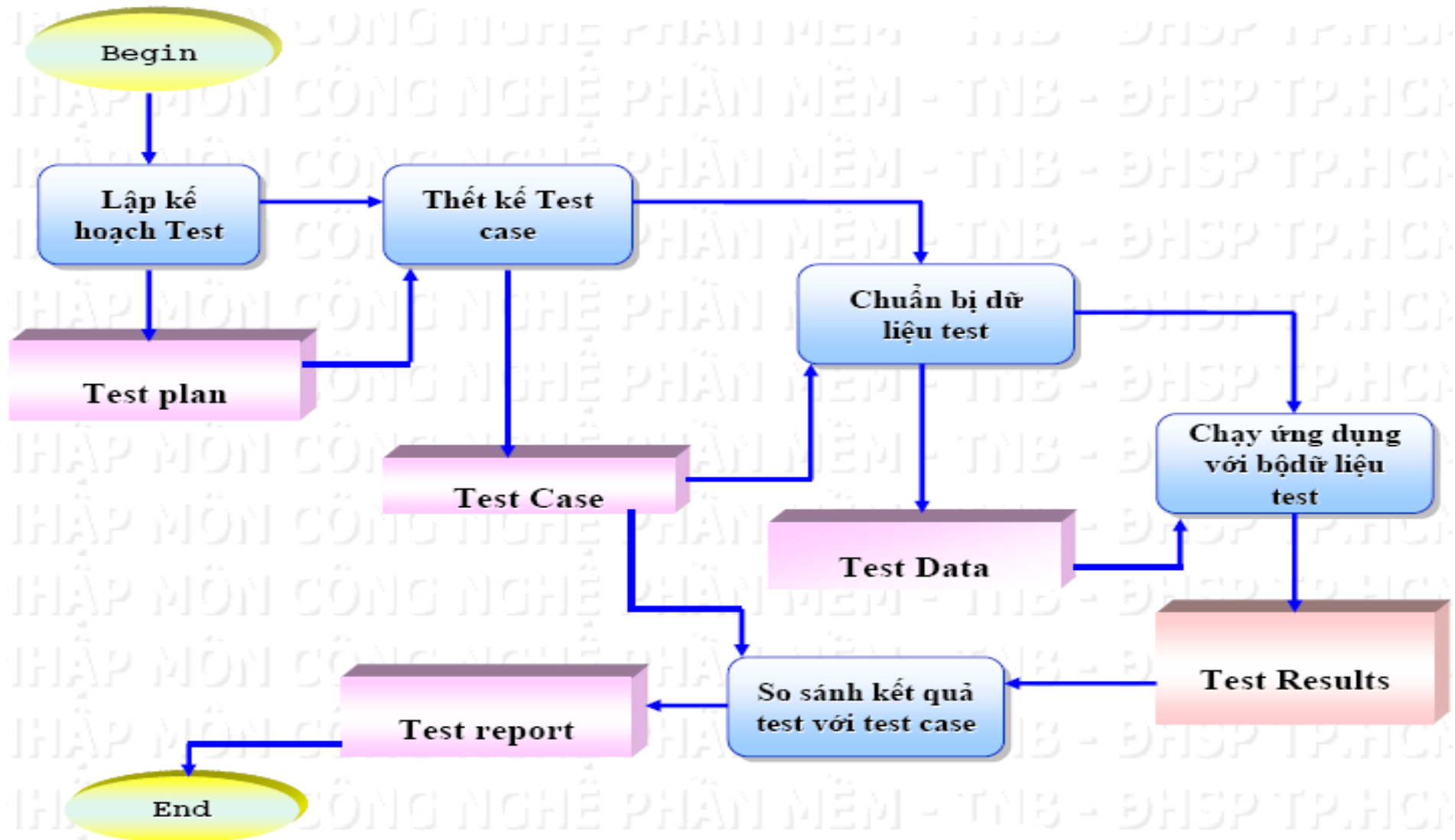


**FIGURE**

Iterative development model



# Quy trình kiểm thử phần mềm







# Các mức độ test (Test levels)

## ❖ Component testing (unit testing):

- Tìm lỗi trong các component của phần mềm như: modules, programs, objects, classes,...
- Ai thực hiện?

## ❖ Integration testing:

- Test sự kết hợp của các component, sự tác động của các phần khác nhau trong một hệ thống, sự kết hợp của các hệ thống với nhau,...



# Các mức độ test (Test levels)

- ❖ **System testing:** Test hệ thống.
  - Đảm bảo rằng hệ thống (sau khi tích hợp) thỏa mãn tất cả các yêu cầu của người sử dụng
  - Tập trung vào việc phát hiện các lỗi xảy ra trên toàn hệ thống
- ❖ **Acceptance testing:** Test phần mềm đứng theo góc độ người dùng để xác định phần mềm có được chấp nhận hay không.

# Một số kỹ thuật test

## ❖ Test tĩnh:

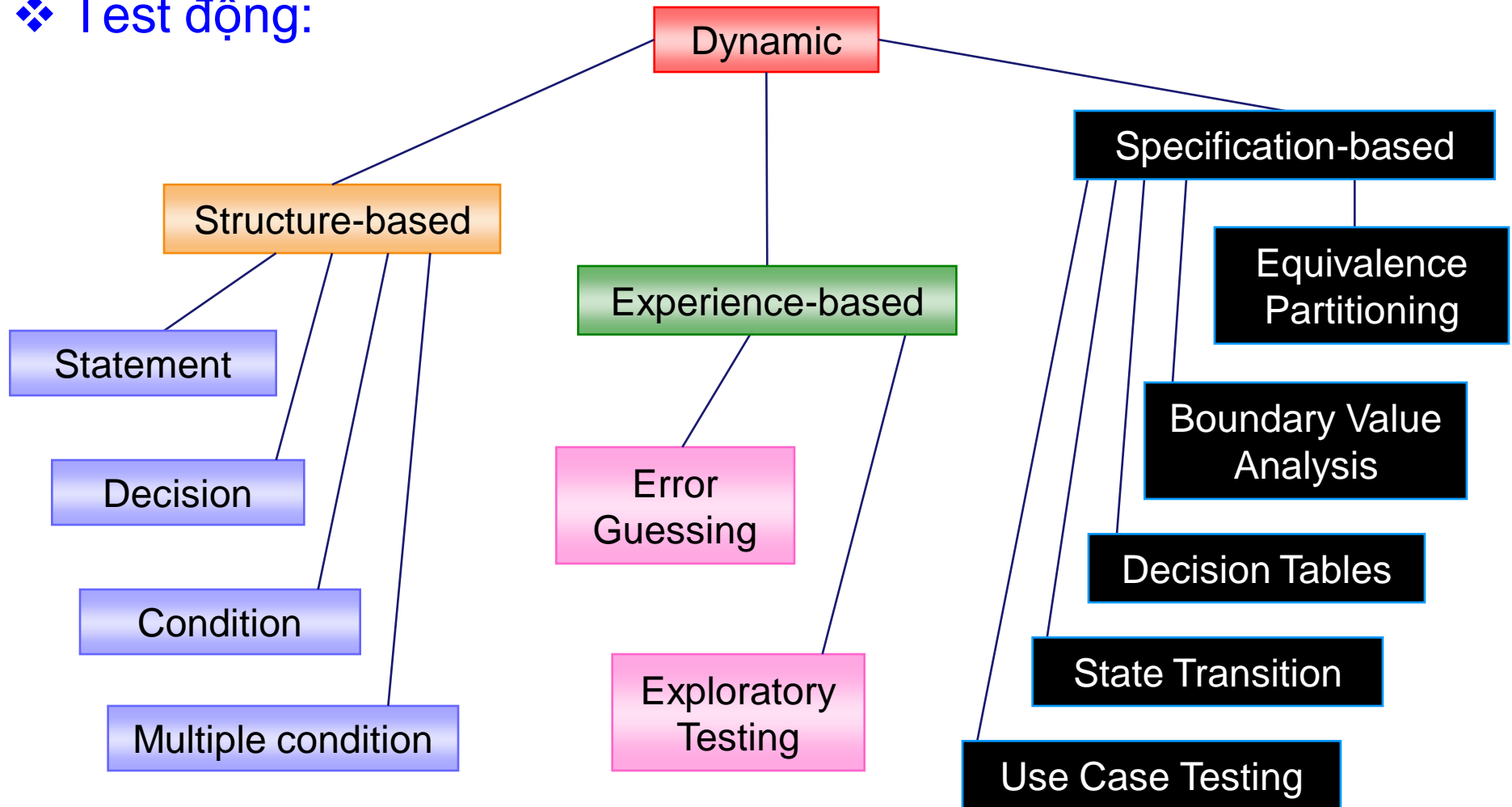
- Dựa vào việc kiểm tra tài liệu, source code,... mà **không cần phải thực thi phần mềm**.
- Các lỗi được tìm thấy trong quá trình kiểm tra có thể dễ dàng được loại bỏ và chi phí rẻ hơn nhiều so với khi tìm thấy trong **test động**. Một số lợi ích khi thực hiện việc kiểm tra (reviews):
  - Lỗi sớm được tìm thấy và sửa chữa
  - Giảm thời gian lập trình
  - Giảm thời gian và chi phí test

## ❖ Test tĩnh (tt):

- Các tài liệu được kiểm thử:
  - Tài liệu đặc tả yêu cầu
  - Tài liệu đặc tả thiết kế
  - Sơ đồ luồng dữ liệu
  - Mô hình ER
  - Source code
  - Test case
  - ...

# Một số kỹ thuật test

## ❖ Test động:



# Một số kỹ thuật test

## ❖ Test động:

- Test dựa trên mô tả (specification-based) hay còn gọi **test chức năng** (functional testing): Test những gì mà phần mềm phải làm, không cần biết phần mềm làm như thế nào (kỹ thuật **black box**)
- Test dựa trên cấu trúc (structure-based) hay còn gọi **test phi chức năng** (non-functional testing): Test phần mềm hoạt động như thế nào (kỹ thuật **white box**)
- Test dựa trên kinh nghiệm (experience-based): đòi hỏi sự hiểu biết, kỹ năng và kinh nghiệm của người test

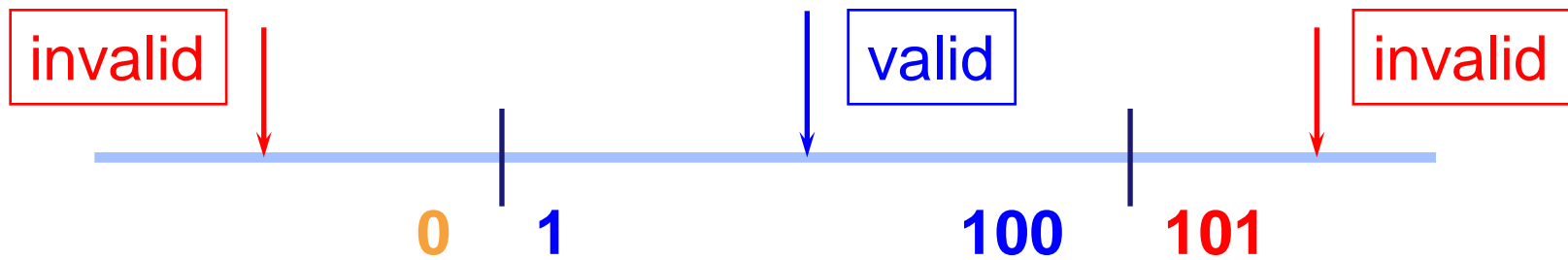
# Kỹ thuật specification-based

- ❖ Kỹ thuật phân vùng tương đương – EP (Equivalence Partitioning)
  - Ví dụ: một textbox chỉ cho phép nhập số nguyên từ 1 đến 100
  - → Ta không thể nhập tất cả các giá trị từ 1 đến 100
  - Ý tưởng của kỹ thuật này: chia (partition) đầu vào thành những nhóm tương đương nhau (equivalence). Nếu một giá trị trong nhóm hoạt động đúng thì tất cả các giá trị trong nhóm đó cũng hoạt động đúng và ngược lại.

# Kỹ thuật specification-based

## ❖ Kỹ thuật phân vùng tương đương – EP (tt)

- Trong ví dụ trên dùng kỹ thuật phân vùng tương đương, chia làm 3 phân vùng như sau:



- Như vậy chỉ cần chọn 3 test case để test trường hợp này: -5, 55, 102 hoặc 0, 10, 1000, ...



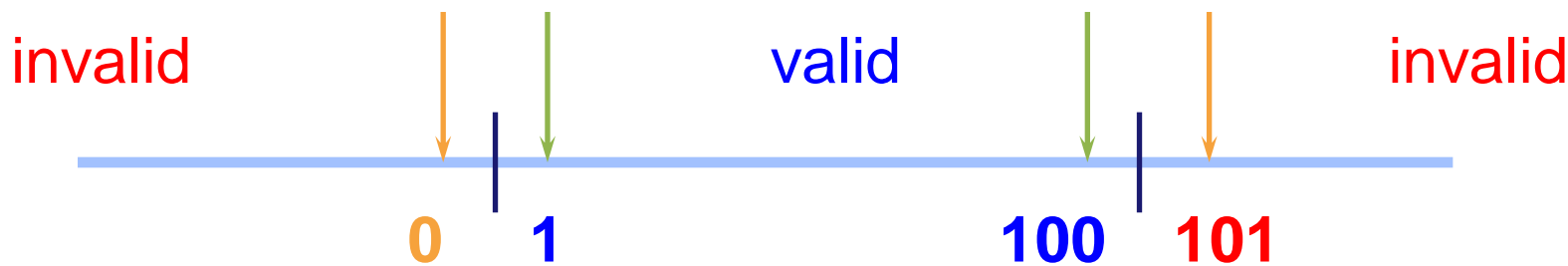
# Kỹ thuật specification-based

## ❖ Kỹ thuật phân vùng tương đương – EP (tt)

- Tuy nhiên nếu ta nhập vào **số thập phân** (55.5) hay một **ký tự không phải là số** (abc)?
- Trong trường hợp trên có thể chia làm 5 phân vùng như sau:
  - Các số nguyên từ 1 đến 100
  - Các số nguyên nhỏ hơn 1
  - Các số nguyên lớn hơn 100
  - Không phải số
  - Số thập phân
- Như vậy, việc phân vùng có đúng và đủ hay không là tùy thuộc vào kinh nghiệm của tester.

# Kỹ thuật Boundary Value Analysis

- ❖ Kỹ thuật phân tích giá trị giới hạn - BVA (Boundary Value Analysis)
  - Kỹ thuật BVA sẽ chọn các giá trị nằm tại các điểm giới hạn của phân vùng.



- Áp dụng kỹ thuật BVA cần 4 test case để test trường hợp này: 0,1,10,101

# Kỹ thuật EP & BVA

- ❖ Xét ví dụ: Một ngân hàng trả lãi cho khách hàng dựa vào số tiền còn lại trong tài khoản. Nếu số tiền từ 0 đến 100\$ thì trả 3% lãi, từ lớn hơn 100 \$ đến nhỏ hơn 1000\$ trả 5% lãi, từ 1000\$ trở lên trả 7% lãi.

- Dùng kỹ thuật EP:

Invalid partition	Valid (for 3% interest)	Valid (for 5%)	Valid (for 7%)
-\$0.01	\$0.00	\$100.00	\$100.01 \$999.99 \$1000.00

- Kỹ thuật EP: -0.44, 55.00, 777.50, 1200.00
- Kỹ thuật BVA: -0.01, 0.00, 100.00, 100.01, 999.99, 1000.00



# Tại sao phải kết hợp BVA và EP

- ❖ Mỗi giá trị giới hạn đều nằm trong một phân vùng nào đó. Nếu chỉ sử dụng giá trị giới hạn thì ta cũng có thể test luôn phân vùng đó.
- ❖ Tuy nhiên vấn đề đặt ra là nếu như giá trị đó sai thì nghĩa là giá trị giới hạn bị sai hay là cả phân vùng bị sai. Hơn nữa, nếu chỉ sử dụng giá trị giới hạn thì không đem lại sự tin tưởng cho người dùng vì chúng ta chỉ sử dụng những giá trị đặc biệt thay vì sử dụng giá trị thông thường.
- ❖ Vì vậy, **cần phải kết hợp cả BVA và EP**

# Ví dụ

Customer Name

2-64 chars.

Account number

6 digits, 1st non-zero

Loan amount requested

£500 to £9000

Term of loan

1 to 30 years

Monthly repayment

Minimum £10

**Term:**

**Repayment:**

**Interest rate:**

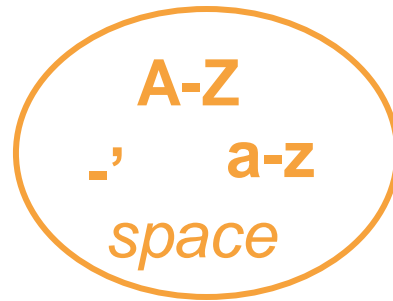
**Total paid back:**

# Customer name

Number of characters:



Valid characters:



Conditions	Valid Partitions	Invalid Partitions	Valid Boundaries	Invalid Boundaries
Customer name	2 to 64 chars valid chars	< 2 chars > 64 chars invalid chars	2 chars 64 chars	1 chars 65 chars 0 chars

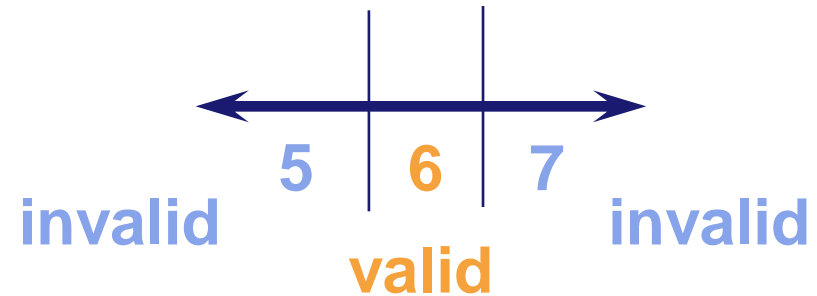
# Account number

first character:

valid: non-zero

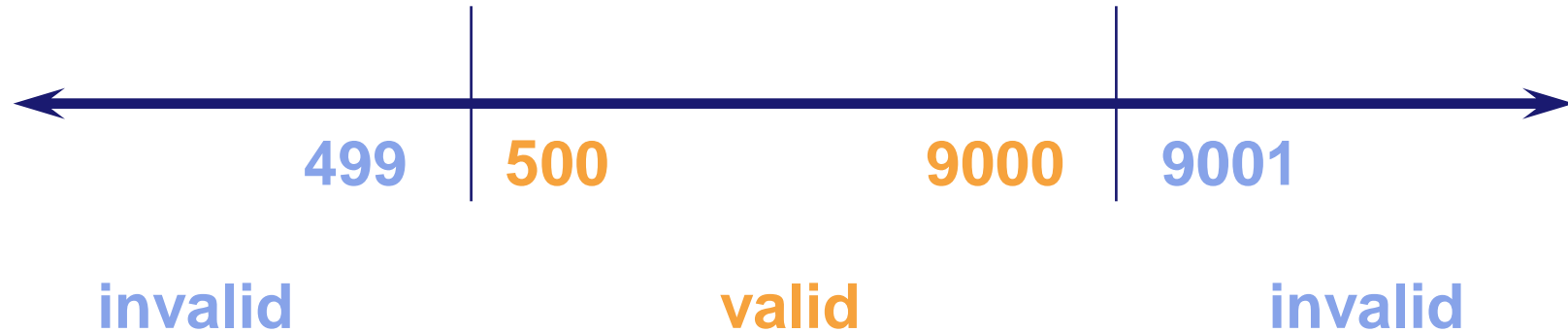
invalid: zero

number of digits:



Conditions	Valid Partitions	Invalid Partitions	Valid Boundaries	Invalid Boundaries
Account number	6 digits 1 <sup>st</sup> non-zero	< 6 digits > 6 digits 1 <sup>st</sup> digit = 0 non-digit	100000 999999	5 digits 7 digits 0 digits

# Loan amount



Conditions	Valid Partitions	Invalid Partitions	Valid Boundaries	Invalid Boundaries
Loan amount	500 - 9000	< 500 > 9000 0 non-numeric null	500 9000	499 9001



# Condition template

Conditions	Valid Partitions	Tag	Invalid Partitions	Tag	Valid Boundaries	Tag	Invalid Boundaries	Tag
Customer name	2 - 64 chars valid chars	V1	< 2 chars	X1	2 chars	B1	1 char	D1
		V2	> 64 chars	X2	64 chars	B2	65 chars	D2
			invalid char	X3			0 chars	D3
Account number	6 digits 1 <sup>st</sup> non-zero	V3	< 6 digits	X4	100000	B3	5 digits	D4
		V4	> 6 digits	X5	999999	B4	7 digits	D5
			1 <sup>st</sup> digit = 0	X6			0 digits	D6
			non-digit	X7				
Loan amount	500 - 9000	V5	< 500	X8	500	B5	499	D7
			>9000	X9	9000	B6	9001	D8
			0	X10				
			non-integer	X11				
			null	X12				

# Design test cases

Test Case	Description	Expected Outcome	New Tags Covered
1	Name: John Smith Acc no: 123456 Loan: 2500 Term: 3 years	Term: 3 years Repayment: 79.86 Interest rate: 10% Total paid: 2874.96	V1, V2, V3, V4, V5 .....
2	Name: AB Acc no: 100000 Loan: 500 Term: 1 year	Term: 1 year Repayment: 44.80 Interest rate: 7.5% Total paid: 537.60	B1, B3, B5, .....



# Vai trò Tester

- ❖ Kiểm lỗi phần mềm
- ❖ Kiểm lỗi bản đóng gói
- ❖ Kiểm lỗi tài liệu
  - User guide
  - Installation Guide
  - Release Notes
  - Troubleshooting

- ❖ Chuẩn bị môi trường test
  - Windows XP, 2000, 2003
  - Linux
  - IE, FireFox, Netscape, Mozilla
  - Test Database, Test data
- ❖ Thiết kế Test case
- ❖ Thực hiện test các Test case trong từng môi trường khác nhau
- ❖ Mô tả Bug và chi tiết các bước để tạo ra bug
- ❖ Theo dõi quá trình Fix Bug
- ❖ Báo cáo kết quả test

## ❖ Testing Tools

- <http://www.aptest.com/resources.html>

## ❖ Testing Course

- <http://www.aptest.com/courses.html>
- <http://www.aptest.com/testtypes.html>
- <http://www.appperfect.com/products/windowstester.html>
- <http://www.openseminar.org/se/modules/7/index/screen.do>



# Chương 6: Đóng gói Phần mềm



- ❖ Quy trình đóng gói phần mềm.
- ❖ Sơ liệu mô tả đóng gói phần mềm
- ❖ Một số công cụ đóng gói thông dụng:
  - Installshield
  - Wise Install
  - Project Management Tool
    - Packer Role
  - ...



# Công cụ InstallShield

The image shows a Windows XP desktop environment. The Start menu is open, displaying a list of installed programs. The 'InstallShield' program is selected, and a context menu is visible over it, showing 'InstallShield X Tools' and 'InstallShield X Express'. A mouse cursor is hovering over 'InstallShield X Express', and a tooltip 'Launch InstallShield X Express' is displayed. The desktop background is blue with a wavy pattern. The taskbar at the bottom shows the Start button, several open applications, and the system tray with the time 4:43 PM.

Windows Update  
Bkav2006  
Ace Utilities  
Program Updates  
Accessories  
Games  
Startup  
Internet Explorer  
Windows Media Player  
Bach Khoa Antivirus  
Microsoft Web Publishing  
Microsoft Visual Studio 6.0  
Microsoft Office  
Ace Utilities  
Ace Utilities  
Software995  
WinRAR  
Yahoo! Messenger  
Adobe Reader 8  
Microsoft .NET Framework SDK v2.0  
Microsoft Visual Studio 2005  
Lenovo EasyCapture  
Microsoft SQL Server  
Microsoft SQL Server 2005  
Microsoft SQL Server - Switch  
Microsoft Press Training Kit Exam Prep  
Lavasoft  
COWON Media Center - jetAudio  
Developer Express Inc.NET.2005  
FlashGet  
SomePDF  
Mozilla Firefox  
Lingoes 2  
Lingoes 2

Easy Button  
NetBeans  
Microsoft DirectX SDK (March 2008)  
Developer Express Inc.NET.v3  
Rational Software  
Active Desktop Calendar  
AVG Anti-Spyware 7.5  
InstallShield  
InstallShield X Tools  
InstallShield X Express

Launch InstallShield X Express

start  
Microsoft PowerPoint ...  
4:43 PM

# Công cụ InstallShield

The screenshot displays the 'InstallShield Express' application window. The title bar reads 'IS InstallShield Express'. The menu bar includes 'File', 'Edit', 'View', 'Go', 'Project', 'Build', 'Tools', and 'Help'. The toolbar contains various icons for file operations and project management. The main content area is titled 'Start Page' and features the 'InstallShield X' logo on the left. A large 'get started' banner is visible in the background. Below the banner is a table with columns 'name' and 'modified'. The table contains one entry: 'VIDu.ise' with a modified date of 'Wednesday, October 22, 2008 7:32:42 PM'. On the left side, there are two panels: 'Project Tasks' with options like 'Create a new project', 'Open an existing project', and 'Browse sample projects'; and 'Help Topics' with questions like 'Where can I find help for error or warning codes?' and 'What is the difference between patching and upgrading?'. On the right side, there are two panels: 'Resources' with links to 'Help Library', 'Support Central', 'InstallShield Community', 'Technical FAQs', 'Webinars', and 'Downloads'; and 'Contact Us' with links to 'Give Us Feedback' and 'Join our Customer Experience Improvement Program'. At the bottom, there are three columns with icons and text: 'Getting Started' (with a hand icon), 'Upcoming Webinars' (with a calendar icon), and 'Top Knowledge Base Articles' (with a question mark icon). The status bar at the bottom left shows 'Ready'.

IS InstallShield Express

File Edit View Go Project Build Tools Help

Start Page Start Page IS

## InstallShield X

get started

name	modified
VIDu.ise	Wednesday, October 22, 2008 7:32:42 PM

**Project Tasks**

- Create a **new** project
- Open an **existing** project
- Browse **sample** projects

**Help Topics**

- Where can I find help for error or warning codes?
- What is the difference between patching and upgrading?

**Resources**

- Help Library
- Support Central
- InstallShield Community
- Technical FAQs
- Webinars
- Downloads

**Contact Us**

- Give Us Feedback
- Join our **Customer Experience Improvement Program** and help improve the quality, reliability, and performance of InstallShield

**Getting Started**

Not sure where to begin? Click the Getting Started link to enter the world of installation creation technology using InstallShield X Express Edition.

**Upcoming Webinars**

InstallShield webinars are free Internet seminars that will help you gain the most from your tools. View the latest webinars here.

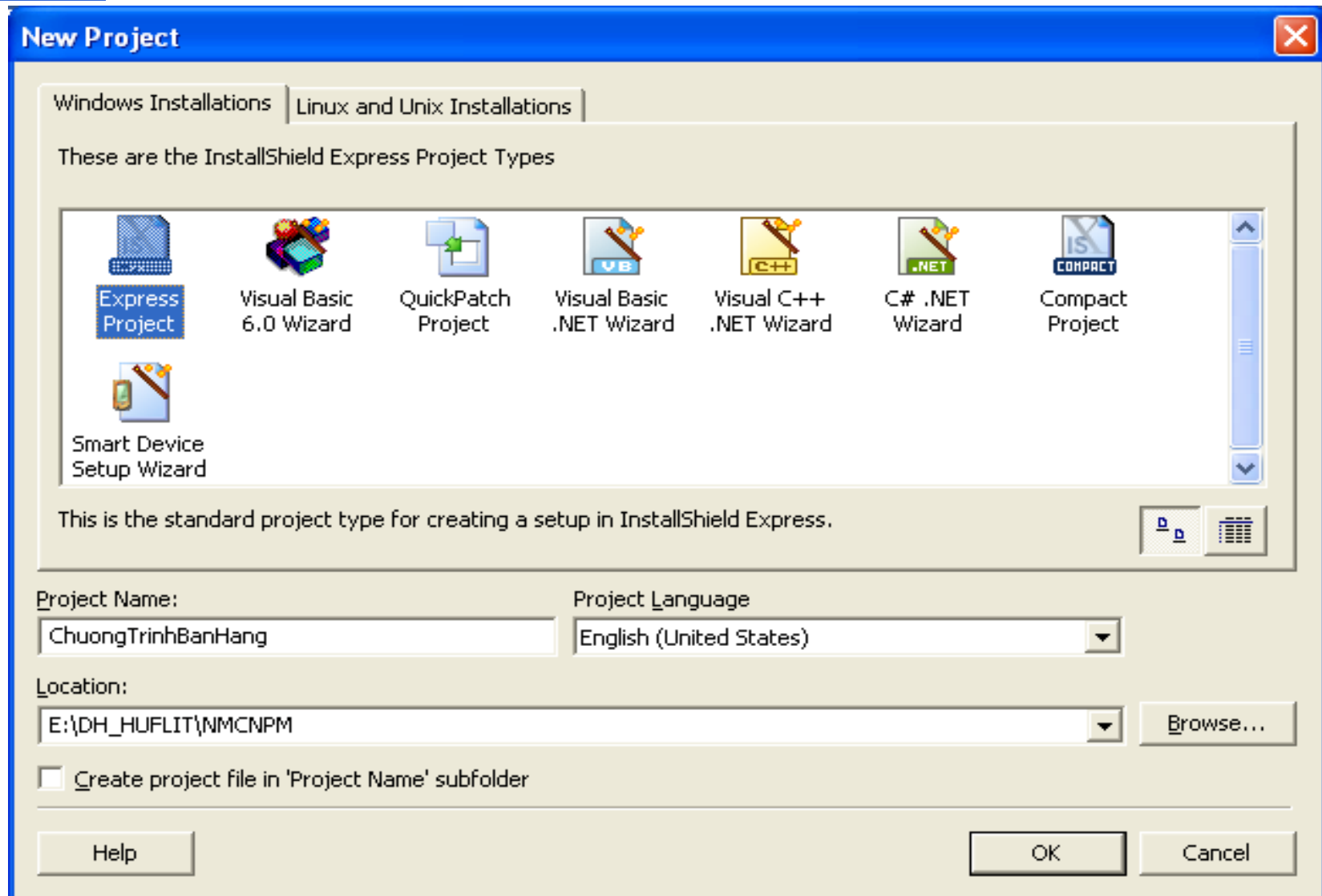
**Top Knowledge Base Articles**

Click an article ID to read that article.

- Q104371 - INFO: Redistributable Files
- Q104962 - INFO: The BDE 5.1.1 Merge Module
- Q107094 - ERRDOC: Run-time Error 1607

Ready

# Công cụ InstallShield



# Công cụ InstallShield

IS ChuongTrinhBanHang - InstallShield Express [Express Project]

File Edit View Go Project Build Tools Help

Start Page Project Assistant Installation Designer Project Assistant

## project assistant

The Project Assistant will guide you through the process of building your installation. You can use the Project Assistant to create a basic installation, or to build the foundation for an advanced installation. To access all the features and all the power of InstallShield, use the Installation Designer.

**Installation Requirements**    **Installation Architecture**    **Installation Interview**

Installation Architecture  
NewFeature1

Installation Configuration

Application Architecture

Application Files    Application Shortcuts    Application Registry

Click **Home** to return to this page.    Click **Next** to get started using the Project Assistant.

Application Information    Installation Requirements    Installation Architecture    Application Files    Application Shortcuts    Application Registry    Installation Interview    Build Installation

Ready

# Công cụ InstallShield

The screenshot displays the InstallShield Express [Express Project] window. The title bar reads "IS ChuongTrinhBanHang - InstallShield Express [Express Project]". The menu bar includes "File", "Edit", "View", "Go", "Project", "Build", "Tools", and "Help". The toolbar contains various icons for file operations and project management. The main window is divided into three tabs: "Start Page", "Project Assistant", and "Installation Designer". The "Project Assistant" tab is active, showing the "Application Information" page. The page title is "Application Information" with the subtitle "Specify general application information." The main content area contains the following text and form fields:

The Application Information page assists you in requesting general information about your application. This information is used from within the installation as well as from the Add or Remove Programs in the Windows Control Panel.

Specify your company name:

Specify your application name:

Specify your application version:

Specify your company web address:

Would you like to automatically notify your end users when you create updates to your application?  
 Yes  No

Select the icon to display with your application in Add or Remove Programs.

The left sidebar contains three sections: "More Options" with "Edit the default installation location.", "Other Places" with "General Information" and "Update Service", and "Help Links" with four links: "Learn more about Add/Remove Programs.", "Learn about how the company name and product name are used by the installation.", "Tell me more about the installation life cycle.", and "Tell me more about automatic update notification."

The bottom taskbar shows several icons and labels: "Application Information", "Installation Requirements", "Installation Architecture", "Application Files", "Application Shortcuts", "Application Registry", "Installation Interview", and "Build Installation". The system tray at the bottom left shows "Ready".

# Công cụ InstallShield

The screenshot shows the 'Project Assistant' window for an 'Express Project' in InstallShield. The window title is 'ChuongTrinhBanHang - InstallShield Express [Express Project]'. The menu bar includes 'File', 'Edit', 'View', 'Go', 'Project', 'Build', 'Tools', and 'Help'. The toolbar contains various icons for navigation and actions. The 'Installation Designer' tab is active, and the 'Project Assistant' pane is open on the right.

**Installation Requirements**  
Specify the software and operating systems required by your installation.

**More Options**  
Create a custom software condition.

**Other Places**  
Requirements

**Help Links**  
How can I create custom requirements?  
When does the installation check for requirements?

Does your application require any specific operating systems?  
 Yes  No

<input checked="" type="checkbox"/> Windows 95	<input checked="" type="checkbox"/> Windows NT 4.0
<input checked="" type="checkbox"/> Windows 98	<input checked="" type="checkbox"/> Windows 2000
<input checked="" type="checkbox"/> Windows 98 SE	<input checked="" type="checkbox"/> Windows XP
<input checked="" type="checkbox"/> Windows Me	<input checked="" type="checkbox"/> Windows 2003 Server

Does your application require any software to be installed on the machine?  
 Yes  No

Required Software	Runtime Message
<input type="checkbox"/> MDAC Version 2.6	
<input type="checkbox"/> Adobe Acrobat 5 Reader Inst...	
<input type="checkbox"/> Internet Explorer 6.0	
<input type="checkbox"/> Internet Explorer 5.0	
<input type="checkbox"/> MDAC Version 2.7	

The taskbar at the bottom shows the 'Ready' state and several open applications: 'start', 'Borland C++ for DOS', 'FreeCell Game #31376', 'Total Commander 6.0...', 'Microsoft PowerPoint ...', and 'ChuongTrinhBanHang...'. The system clock shows 4:52 PM.



# Công cụ InstallShield

The screenshot displays the InstallShield Express software interface. The title bar reads "IS ChuongTrinhBanHang - InstallShield Express [Express Project]". The menu bar includes "File", "Edit", "View", "Go", "Project", "Build", "Tools", and "Help". The toolbar contains various icons for file operations and project management. The main window is divided into several sections:

- Start Page**: Contains instructions for adding application files to the installation.
- Project Assistant**: A sidebar on the left with sections for "More Options" (Launch Windows Explorer, Scan Visual Basic 6 project), "Other Places" (Files and Folders, Feature Files), and "Help Links" (How do I add files to a fixed folder location?, How can I add more "variable" folders to the defaults already being displayed to me?).
- Installation Designer**: The main workspace showing a tree view of the destination computer structure. The tree includes folders for [AppDataFolder], [CommonFilesFolder], and [ProgramFilesFolder], with a sub-folder [ProgramFilesFolder] containing a folder named "DH-CNTT" and a sub-folder "ChuongTrinhBanHang [INSTALLDIR]".
- Table**: A table with columns for "Name", "Size", "Link To", "Modified", and "Destir".
- Buttons**: "Add Files" and "Add Folders" buttons are located at the bottom right of the main workspace, with "Add Files" circled in red.

The bottom of the window features a taskbar with icons for "Application Information", "Installation Requirements", "Installation Architecture", "Application Files", "Application Shortcuts", "Application Registry", "Installation Interview", and "Build Installation". The status bar at the bottom left shows "Ready".

# Công cụ InstallShield

The screenshot displays the InstallShield Express software interface. The title bar reads "ChuongTrinhBanHang - InstallShield Express [Express Project]". The menu bar includes "File", "Edit", "View", "Go", "Project", "Build", "Tools", and "Help". The toolbar contains various icons for file operations and project management. The "Project Assistant" tab is active, showing the "Application Shortcuts" configuration page.

**Application Shortcuts**  
Define your application shortcuts.

**More Options**  
 Create an uninstallation shortcut.

**Other Places**  
Shortcuts  
File Extensions

**Help Links**  
What is a file extension?  
How do I create shortcuts to files not in my installation?

Application shortcuts allow users to launch your application from the Windows Start menu.  
By default, InstallShield creates shortcuts to the executable files you have included in your installation. You can delete these default shortcuts as well as create shortcuts for other files in your installation.

File Name	Create shortcut in Start Menu	Create shortcut on Desktop	Use alternate shortcut Icon:
ChuongTrinhBanHang.exe	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uninstall ChuongTrinhBanHang	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Buttons: New, Rename, Delete

Buttons: Browse..., Associate a file extension with the shortcut'

Taskbar: Application Information, Installation Requirements, Installation Architecture, Application Files, **Application Shortcuts**, Application Registry, Installation Interview, Build Installation

Ready



# Công cụ InstallShield

The screenshot displays the InstallShield Express 'Project Assistant' window. The title bar reads 'ChuongTrinhBanHang - InstallShield Express [Express Project]'. The menu bar includes 'File', 'Edit', 'View', 'Go', 'Project', 'Build', 'Tools', and 'Help'. The 'Installation Designer' tab is active, showing the 'Build Installation' section. The main area contains instructions: 'The Project Assistant is now ready to build your installation. Select the installation image types you want to create and click **Build Installations**.' Three options are listed: 'Single Executable' (checked), 'Internet', and 'CD-ROM'. Each option has a description and links for 'Optional distribution settings' and 'Open release folder'. The 'Build Installations' button at the bottom is highlighted with a red circle. The taskbar at the bottom shows various application icons and the system tray with the time '4:58 PM'.

ChuongTrinhBanHang - InstallShield Express [Express Project]

File Edit View Go Project Build Tools Help

Start Page Project Assistant Installation Designer Project Assistant

**Build Installation**  
Build your installation disk images.

**Other Places**  
Releases  
Distribute

**Help Links**  
I've finished the Project Assistant steps. What's next?

The Project Assistant is now ready to build your installation. Select the installation image types you want to create and click **Build Installations**.

**Single Executable**  
A Single Executable installation is an installation that contains all of your application files and settings, as well as all of the subordinate installation-related files in a single Setup.exe.  
[Optional distribution settings](#) [Open release folder](#)

Internet  
An Internet installation is optimized for Internet downloading. Your users will launch it from a Web page called Install.htm, which is generated for you.  
[Optional distribution settings](#) [Open release folder](#)

CD-ROM  
A CD-ROM installation is one in which all of the installation and application files are left uncompressed on the disk image. The build engine automatically creates multiple disk images for CD-ROM installations that are larger than 650 MB.  
[Optional distribution settings](#) [Open release folder](#)

**Build Installations**

Application Information Installation Requirements Installation Architecture Application Files Application Shortcuts Application Registry Installation Interview **Build Installation**

Ready

start Borland C++ for DOS FreeCell Game #31376 Total Commander 6.0... Microsoft PowerPoint ... ChuongTrinhBanHang... 4:58 PM



# Chương 7: Vận hành và bảo trì Phần mềm



- ❖ Bảo trì phần mềm là gì?
- ❖ Tại sao cần phải bảo trì
- ❖ Phân loại các kiểu bảo trì
- ❖ Khó khăn
- ❖ Các nhân tố ảnh hưởng đến chi phí bảo trì
- ❖ Bảo trì phần mềm hướng đối tượng



# Bảo trì phần mềm là gì?

- ❖ 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.

- ❖ Giai đoạn bảo trì:
  - Bắt đầu sau khi khách hàng đã chấp thuận sản phẩm và cần có các thay đổi trên sản phẩm
- ❖ Các thể hiện của bảo trì:
  - Mã nguồn, tài liệu, hướng dẫn sử dụng
- ❖ Còn gọi là tiến triển (evolution) để chỉ rõ sự phát triển của sản phẩm thay vì gọi đó là bảo trì

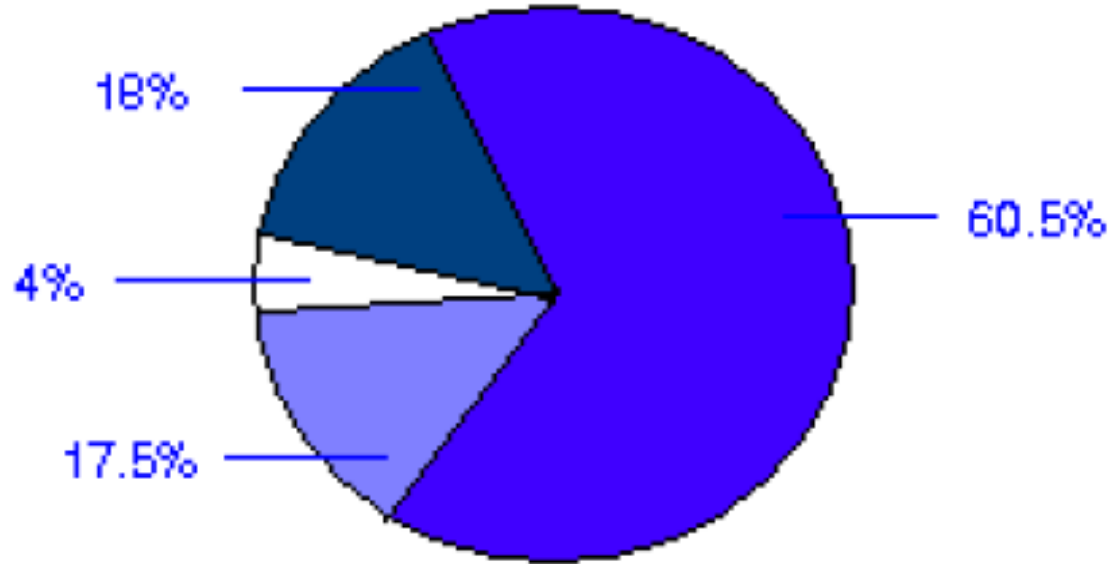


# Tại sao cần phải bảo trì

- ❖ **Hiệu chỉnh:** Các **lỗi** về đặc tả, thiết kế, tài liệu, mã nguồn,...
- ❖ **Hoàn thiện:** **Thay đổi** nhằm hoàn thiện hiệu năng của sản phẩm
  - Ví dụ: Khách hàng yêu cầu thêm một số chức năng hay sửa đổi sản phẩm để tăng tốc độ xử lý.
- ❖ **Thích ứng:** Các **thay đổi** nhằm đáp ứng những thay đổi trong môi trường mà sản phẩm đang vận hành.
  - Ví dụ: thay đổi trình biên dịch, hệ điều hành, phần cứng,...

# Tại sao cần phải bảo trì

- ❖ Được xem như là dịch vụ hậu mãi, **giữ khách hàng** bằng cách cung cấp những dịch vụ bảo trì tốt nhất.



■ Hiệu chỉnh ■ Hoàn thiện ■ Thích ứng ■ Các dạng khác

Các khoảng thời gian cho mỗi dạng bảo trì





# Phân loại bảo trì phần mềm

## ❖ 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.



# Khó khăn

- ❖ Đây là vấn đề khó khăn và nhiều thách thức của một sản phẩm phần mềm.
  - Liên quan đến tất cả các giai đoạn trong tiến trình xây dựng phần mềm.
- ❖ Chi phí bảo trì?
  - Phụ thuộc vào số lượng các thay đổi
  - Chi phí thay đổi phụ thuộc vào khả năng bảo trì
- ❖ Chi phí bảo trì bị ảnh hưởng bởi cả tác nhân **kỹ thuật và phi kỹ thuật**.



# Khó khăn

- ❖ Nếu bảo trì càng nhiều, sẽ càng làm thay đổi cấu trúc phần mềm và do đó sẽ làm cho việc bảo trì càng trở lên khó khăn hơn.
- ❖ Phần mềm có tuổi thọ càng cao thì càng phải cần chi phí cao hơn (vì sử dụng các ngôn ngữ và chương trình dịch cũ ...)



# Các nhân tố ảnh hưởng

- ❖ 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.
- ❖ 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 lên khó hiểu và thay đổi nhiều.



# Bảo trì phần mềm hướng đối tượng

## ❖ Dễ dàng bảo trì các đối tượng

- Do các khái niệm độc lập nên dễ dàng xác định vị trí nhằm hiệu chỉnh hay nâng cao
- Các thay đổi chỉ ảnh hưởng bên trong đối tượng nên **giảm** thiểu các lỗi hồi qui.

## ❖ Khó khăn?

