

**Nhập môn**  
**Công nghệ học Phần mềm**  
**Introduction to Software Engineering**

**Department of Software Engineering**  
**Faculty of Information Technology**  
**Hanoi University of Technology**  
**TEL: 04-8682595 FAX: 04-8692906**  
**Email: [cnpm@it-hut.edu.vn](mailto:cnpm@it-hut.edu.vn)**

# Phần V

## Kiểm thử và Bảo trì

### Test and Maintenance

#### Chương 9: Phương pháp kiểm thử

##### 9.1 Khái niệm kiểm thử

##### 9.2 Phương pháp thử

##### 9.3 Kỹ thuật thiết kế trường hợp thử

##### 9.4 Phương pháp thử các môđun

# 9.1 Khái niệm kiểm thử

## Định nghĩa kiểm thử:

- Là mấu chốt của đảm bảo chất lượng phần mềm
- Là tiến trình (và là nghệ thuật) nhằm phát hiện lỗi bằng việc xem xét lại đặc tả, thiết kế và mã hoá.
- Kiểm thử thành công là phát hiện ra lỗi; kiểm thử không phát hiện ra lỗi là kiểm thử dở (Sue A.Conger- The New SE)

# Những khó khăn khi kiểm thử

- **Nâng cao chất lượng phần mềm nhưng không vượt quá chất lượng khi thiết kế: chỉ phát hiện các lỗi tiềm tàng và sửa chúng**
- **Phát hiện lỗi bị hạn chế do thủ công là chính**
- **Dễ bị ảnh hưởng tâm lý khi kiểm thử**
- **Khó đảm bảo tính đầy đủ của kiểm thử**

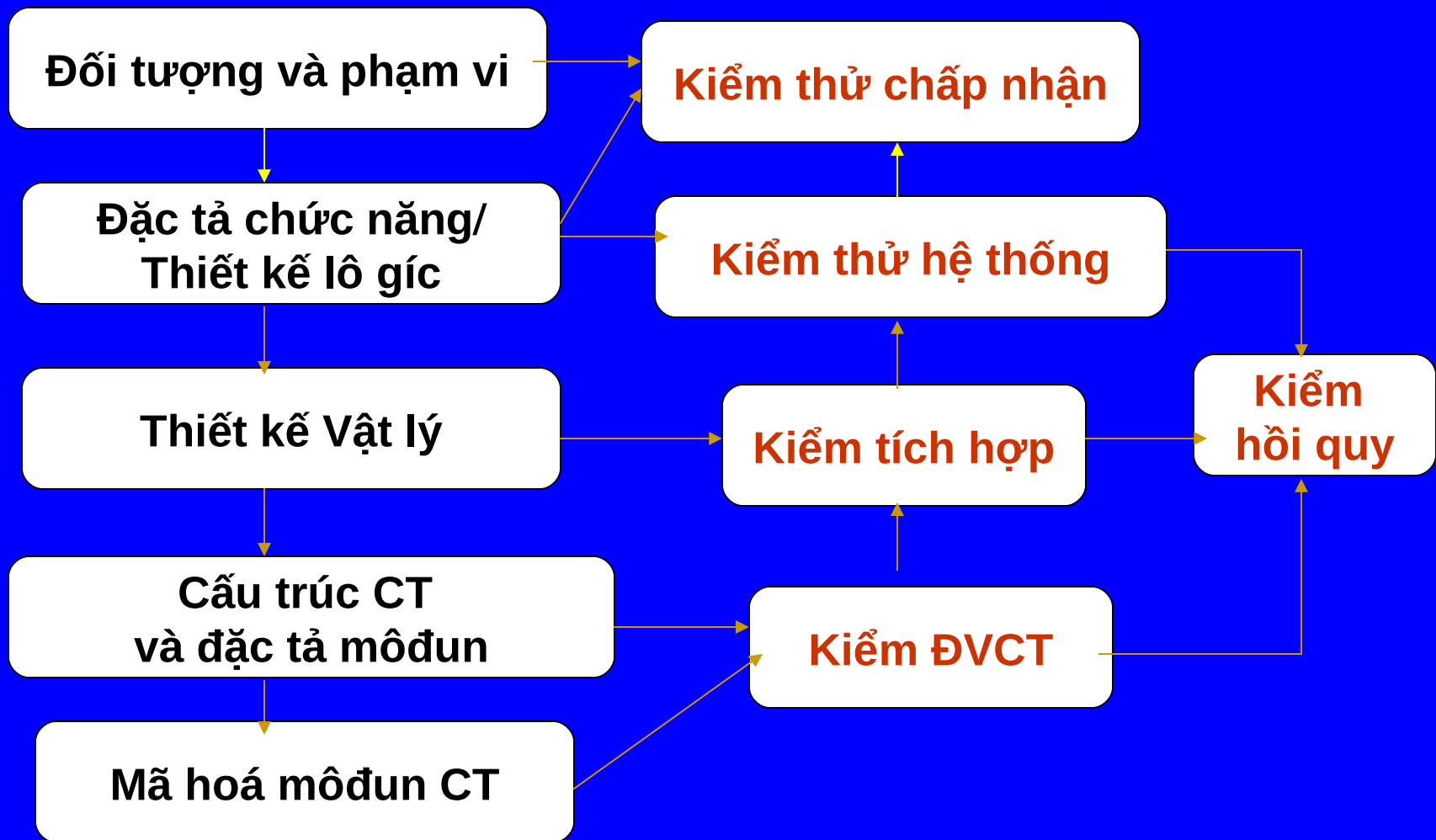
# 6 điểm lưu ý khi kiểm thử

- (1) **Chất lượng phần mềm do khâu thiết kế quyết định là chủ yếu, chứ không phải khâu kiểm thử**
- (2) **Tính dễ kiểm thử phụ thuộc vào cấu trúc chương trình**
- (3) **Người kiểm thử và người phát triển nên khác nhau**

# 6 điểm lưu ý khi kiểm thử (tiếp)

- (4) Dữ liệu thử cho kết quả bình thường thì không có ý nghĩa nhiều, cần có những dữ liệu kiểm thử mà phát hiện ra lỗi
- (5) Khi thiết kế trường hợp thử, không chỉ dữ liệu kiểm thử nhập vào, mà phải thiết kế trước cả dữ liệu kết quả sẽ có
- (6) Khi phát sinh thêm trường hợp thử thì nên thử lại những trường hợp thử trước đó để tránh ảnh hưởng lan truyền sóng

# Tương ứng giữa vòng đời dự án và kiểm thử



## 9.2 Phương pháp thử: thử tĩnh

- **Kiểm thử trên bàn hay Kiểm thử tĩnh: giấy và bút trên bàn, kiểm tra logic, lần từng chi tiết ngay sau khi lập trình xong**
- **Đi xuyên suốt (walk through)**
- **Thanh tra (inspection)**



# Kiểm thử trên máy

- **Gỡ lỗi bằng máy (machine debug) hay kiểm thử động: Dùng máy chạy chương trình để điều tra trạng thái từng động tác của chương trình**
- **9 bước của trình tự kiểm thử bằng máy**

# Trình tự kiểm thử bằng máy

- (1) Thiết kế trường hợp thử theo thử trên bàn
- (2) Trường hợp thử phải có cả kết quả kỳ vọng sẽ thu được
- (3) Dịch chương trình nguồn và tạo môđun tải để thực hiện
- (4) Khi trường hợp thử có xử lý tệp vào-ra, phải làm trước trên bàn việc xác định miền của các tệp

# Trình tự kiểm thử bằng máy (tiếp)

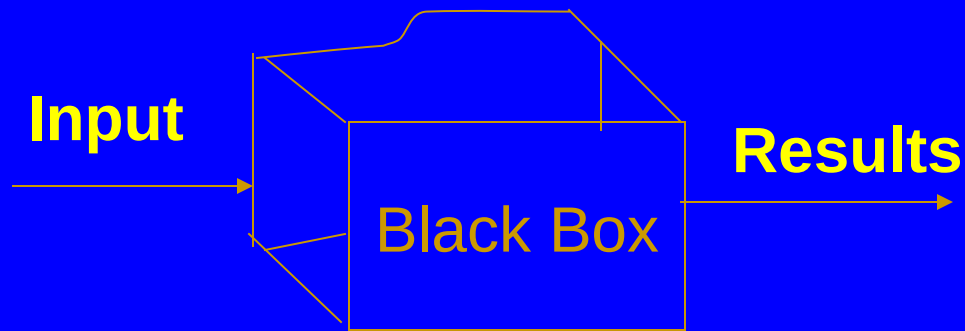
- (5) Nhập dữ liệu đã thiết kế cho trường hợp kiểm thử
- (6) Điều chỉnh môi trường thực hiện môđun tải (tạo thử tục đưa các tệp truy cập tệp vào chương trình)
- (7) Thực hiện môđun tải và ghi nhận kết quả
- (8) Xác nhận kết quả với kết quả kỳ vọng
- (9) Lặp lại thao tác (5)-(8)

## 9.3 Kỹ thuật thiết kế trường hợp thử

- Kỹ thuật thiết kế trường hợp thử dựa trên đặc tả bên ngoài của chương trình: Kiểm thử hộp đen (Black box test): WHAT ?
- Kỹ thuật thiết kế trường hợp thử dựa trên đặc tả bên trong của chương trình: Kiểm thử hộp trắng (white box test): HOW ?
- Kiểm thử Top-Down hay Bottom-Up

# Kiểm thử hộp đen

- Phân đoạn tương đương
- Phân tích giá trị biên
- Đoán lỗi



Black box Data Testing Strategy

# Phương pháp phân đoạn tương đương (Equivalence Partition)

- Mục đích: giảm số lượng test bằng cách chọn các tập dữ liệu đại diện
- Thực hiện: Chia dữ liệu vào thành các đoạn, mỗi đoạn đại diện cho một số dữ liệu  $\Rightarrow$  việc kiểm thử chỉ thực hiện trên đại diện đó
- Ưu điểm: Test theo mức trừu tượng hơn là trường. áp dụng: màn hình, menu hay mức quá trình

## Phương pháp phân tích giá trị biên (Boundary value analysis)

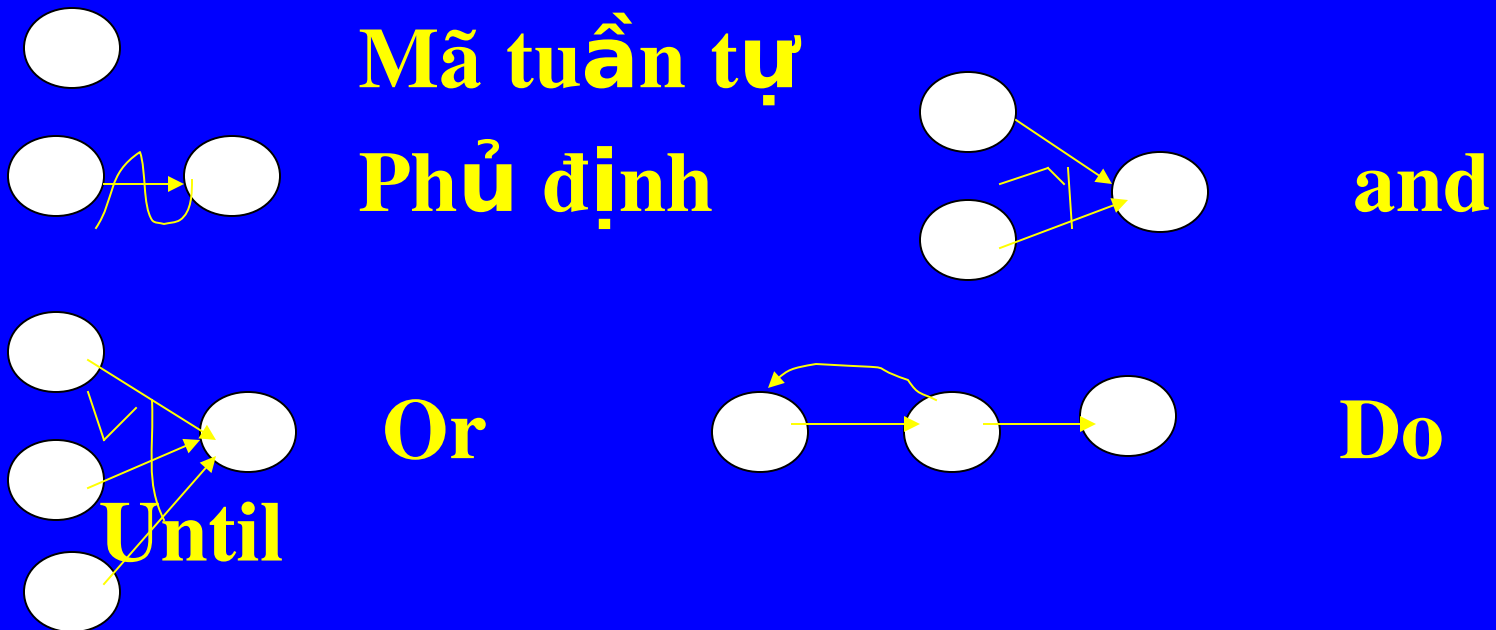
- Là 1 trường hợp riêng của phân đoạn
- Ví dụ: nếu miền dữ liệu là tháng thì giá trị 0 hay >12 là không hợp lệ
- Thường sử dụng trong kiểm thử môđun

# Phương pháp đoán lỗi (Error Guessing)

- Dựa vào trực giác và kinh nghiệm
- Thí dụ lỗi chia cho 0. Nếu môđun có phép chia thì phải kiểm thử lỗi này
- Nhược điểm: không phát hiện hết lỗi

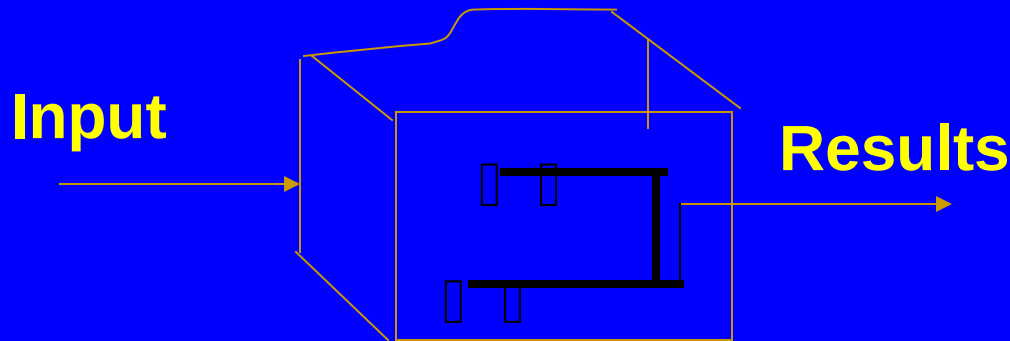


# Phương pháp đồ thị nguyên nhân - kết quả (Cause-effect Graphing)



# Kiểm thử hộp trắng

- Bó các lệnh
- Bó các rẽ nhánh
- Bó các điều kiện
- Bó các điều kiện - rẽ nhánh



White Box Data Testing Strategy

# Trình tự thiết kế

- **Kiểm thử môđun**
- **Kiểm thử tích hợp**
  - **Kiểm thử tích hợp trên xuống**
  - **Kiểm thử tích hợp dưới lên**
  - **Kiểm thử hồi qui**

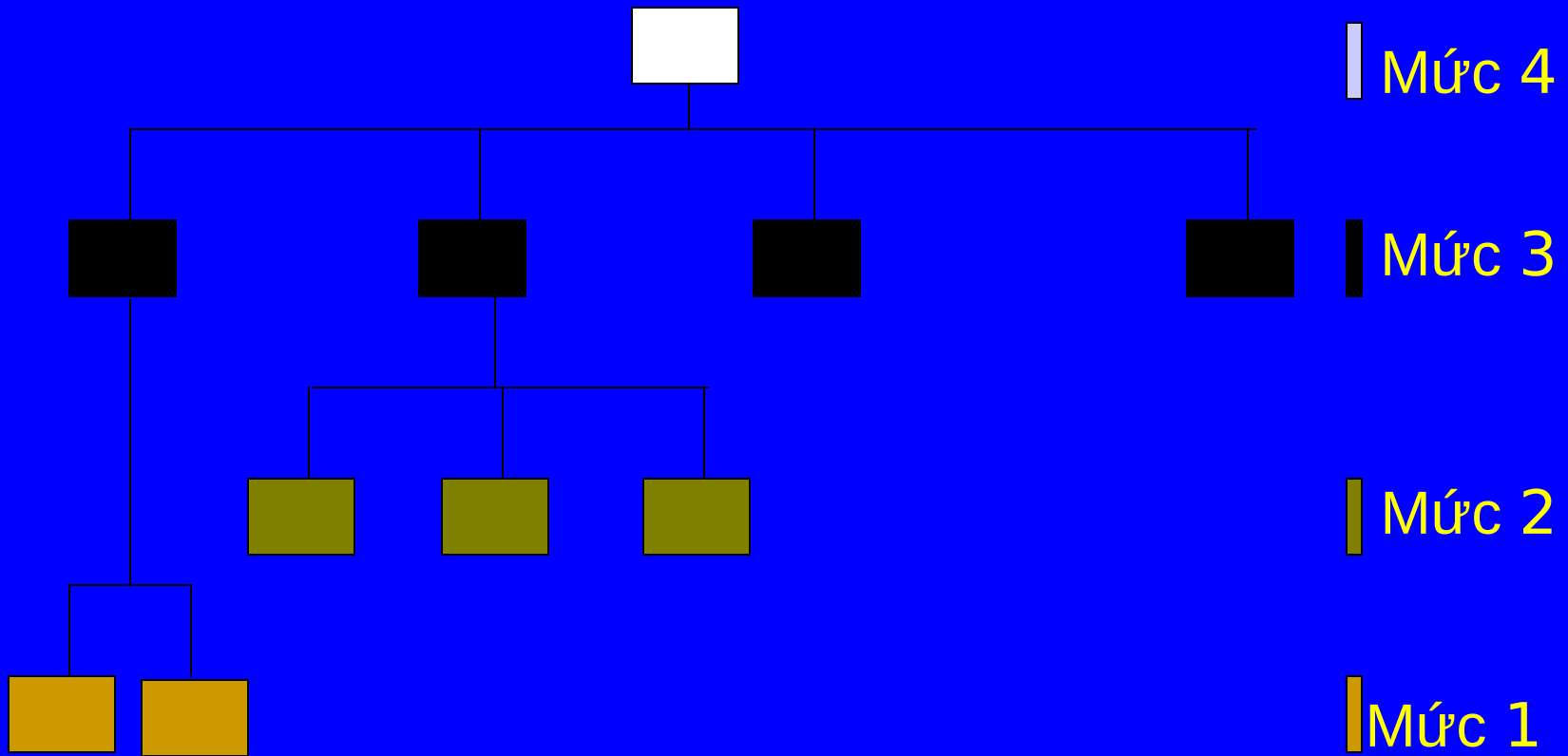
# 9.4 Kỹ thuật kiểm thử môđun

- **Kiểm thử tích hợp môđun**
  - **Kiểm thử dưới lên (Bottom-up Test)**
  - **Kiểm thử trên xuống (Top-down Test)**
  - **Kiểm thử cột trụ (Big bung Test)**
  - **Kiểm thử kẹp (Sandwich Test)**

# Bottom-up Test

- Các môđun mức thấp được tổ hợp vào các chùm thực hiện một chức năng con
- Viết trình điều khiển phối hợp vào/ ra và kiểm thử
- Kiểm thử chùm/bó
- Loại bỏ trình điều khiển và chuyển lên mức trên

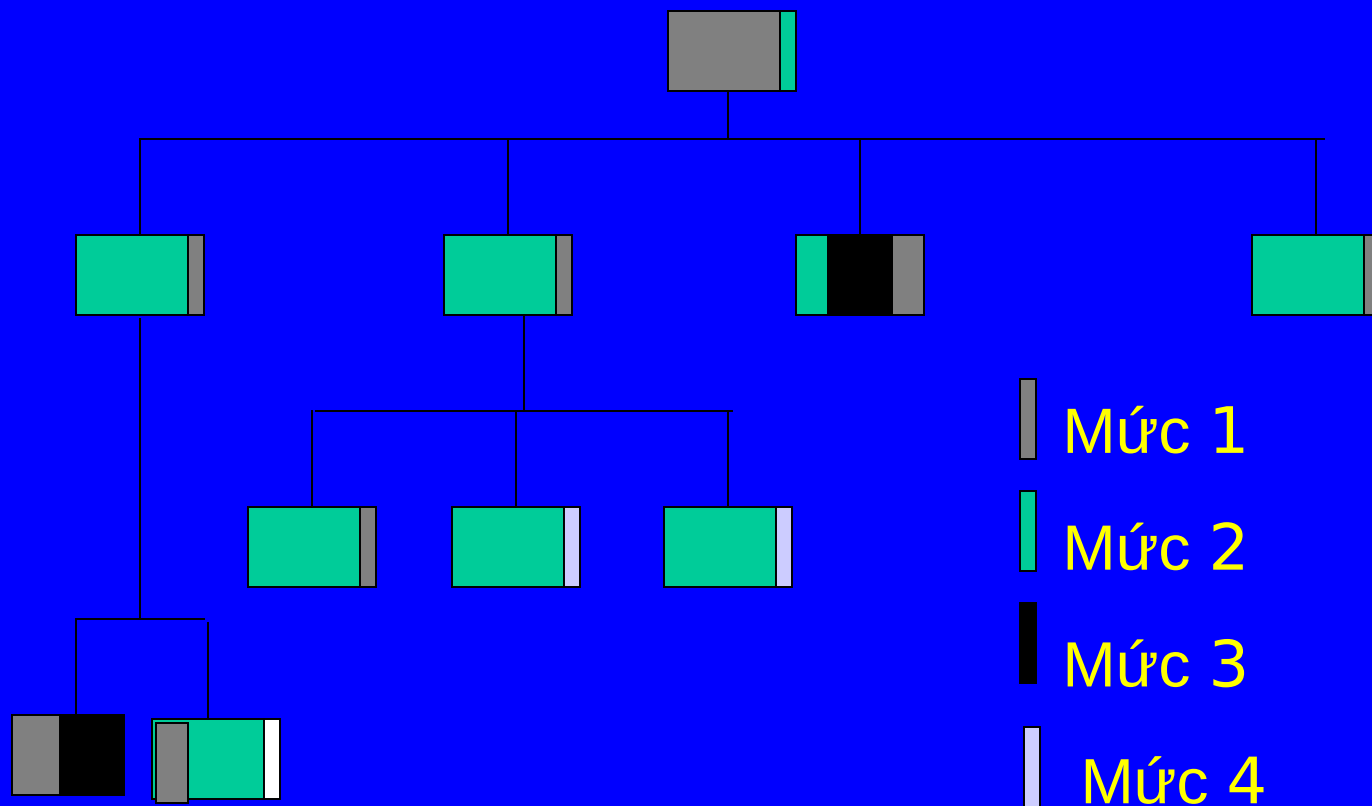
# Bottom-up Test (Tiếp)



# Top-down Test

- **Môđun điều khiển chính được dùng như trình điều khiển kiểm thử, gắn các nút con trực tiếp vào nó**
- **Thay các nút con bằng các môđun thực tại (theo chiều sâu / ngang)**
- **Kiểm thử từng môđun được gắn vào**
- **Các 1 nút thử xong được thử tiếp nút khác**
- **Kiểm thử hồi quy**

# Top-down Test (tiếp)





# Big bung Test

- Tích hợp không tăng dần
- Tất các các môđun đều được tổ hợp trước
- Toàn bộ chương trình được kiểm thử tổng thể
- Khó khăn: khó cô lập lỗi, khi chữa xong lỗi này có thể lỗi mới lại phát sinh

# Sandwich Test

- Tích hợp trên xuống cho các mức trên cấu trúc chương trình
- Tích hợp dưới lên cho các mức phụ thuộc

# Kiểm thử hệ thống

- **Kiểm thử phục hồi: bắt buộc phần mềm hỏng nhiều cách để kiểm chứng phục hồi**
- **Kiểm thử an toàn: kiểm chứng cơ chế bảo vệ**
- **Kiểm thử gay cấn**
- **Kiểm thử hiệu năng**

# Chương 10: Phương pháp bảo trì Maintenance Methods

**10.1 Bảo trì là gì?**

**10.2 Trình tự nghiệp vụ bảo trì**

**10.3 Những vấn đề về bảo trì hiện nay**

# 10.1 Bảo trì là gì?

- **Định nghĩa:** Bảo trì là công việc tu sửa, thay đổi phần mềm đã được phát triển (chương trình, dữ liệu, JCL, các loại tư liệu đặc tả, . . .) theo những lý do nào đó
- **Các hình thái bảo trì: bảo trì để**
  - Tu chỉnh
  - Thích hợp
  - Cải tiến
  - Phòng ngừa

# Bảo trì để tu sửa

- Là bảo trì khắc phục những khiếm khuyết có trong phần mềm
- Một số nguyên nhân điển hình
  - Kỹ sư phần mềm và khách hiểu nhầm nhau
  - Lỗi tiềm ẩn của phần mềm do sơ ý của lập trình hoặc khi kiểm thử chưa bao quát hết
  - Vấn đề tính năng của phần mềm: không đáp ứng được yêu cầu về bộ nhớ, tệp, . . . Thiết kế sai, biên tập sai . . .
  - Thiếu chuẩn hóa trong phát triển phần mềm (trước đó)

# Bảo trì để tu sửa (tiếp)

- **Kỹ nghệ ngược (Reverse Engineering):**  
dò lại thiết kế để tu sửa
- **Những lưu ý**
  - Mức trừu tượng
  - Tính đầy đủ
  - Tính tương tác
  - Tính định hướng

# Bảo trì để thích hợp

- Là tu chỉnh phần mềm theo thay đổi của môi trường bên ngoài nhằm duy trì và quản lý phần mềm theo vòng đời của nó
- Thay đổi phần mềm thích nghi với môi trường: công nghệ phần cứng, môi trường phần mềm
- Những nguyên nhân chính:
  - Thay đổi về phần cứng (ngoại vi, máy chủ, . . .)
  - Thay đổi về phần mềm (môi trường): đổi OS
  - Thay đổi cấu trúc tệp hoặc mở rộng CSDL



# Bảo trì để cải tiến

- Là việc tu chỉnh hệ phần mềm theo các yêu cầu ngày càng hoàn thiện hơn, đầy đủ hơn, hợp lý hơn
- Những nguyên nhân chính:
  - Do muốn nâng cao hiệu suất nên thường hay cải tiến phương thức truy cập tệp
  - Mở rộng thêm chức năng mới cho hệ thống
  - Cải tiến quản lý kéo theo cải tiến tư liệu vận hành và trình tự công việc
  - Thay đổi người dùng hoặc thay đổi thao tác

# Bảo trì để cải tiến (tiếp)

- Còn gọi là tái kỹ nghệ (re-engineering)
- Mục đích: đưa ra một thiết kế cùng chức năng nhưng có chất lượng cao hơn
- Các bước thực hiện:
  - Xây dựng lưu đồ phần mềm
  - Suy dẫn ra biểu thức Bun cho từng dãy xử lý
  - Biên dịch bảng chân lí
  - Tái cấu trúc phần mềm

# Bảo trì để phòng ngừa

- Là công việc tu chỉnh chương trình có tính đến tương lai của phần mềm đó sẽ mở rộng và thay đổi như thế nào
- Thực ra trong khi thiết kế phần mềm đã phải tính đến tính mở rộng của nó, nên thực tế ít khi ta gặp bảo trì phòng ngừa nếu như phần mềm được thiết kế tốt

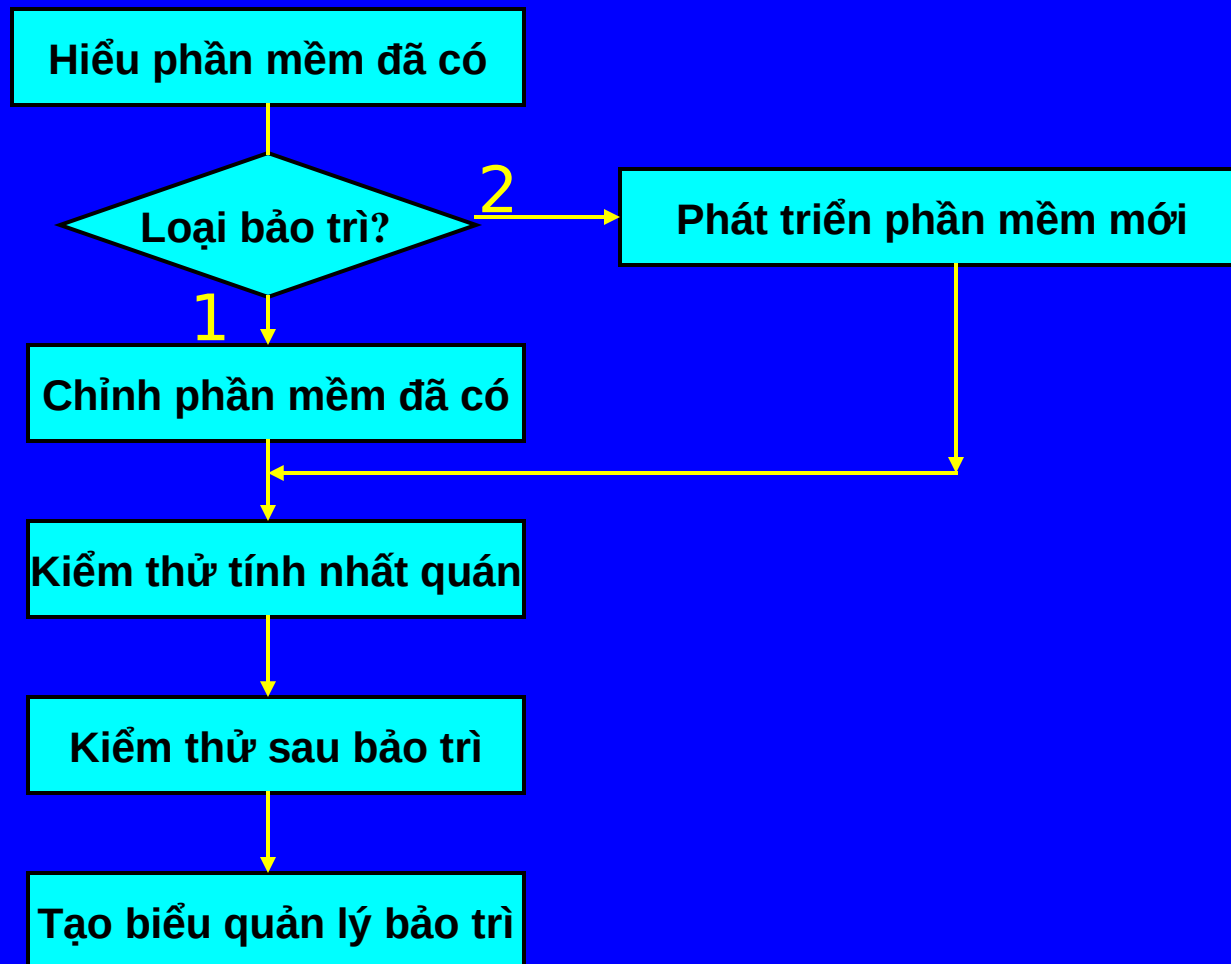
# Bảo trì để phòng ngừa (tiếp)

- Mục đích: sửa đổi để thích hợp với yêu cầu thay đổi sẽ có của người dùng
- Thực hiện những thay đổi trên thiết kế không tương minh
- Hiểu hoạt động bên trong chương trình
- Thiết kế / lập trình lại
- Sử dụng công cụ CASE

## 10.2 Trình tự nghiệp vụ bảo trì

- Quy trình bảo trì là gì ? Đó là quá trình trong vòng đời của phần mềm, cũng tuân theo các pha phân tích, thiết kế, phát triển và kiểm thử từ khi phát sinh vấn đề cho đến khi giải quyết xong
- Thao tác bảo trì: Gồm 2 loại
  - Tu chỉnh cái đã có (loại 1)
  - Thêm cái mới (loại 2)

# Sơ đồ bảo trì



# Hiểu phần mềm đã có

- Theo tài liệu nắm chắc các chức năng
- Theo tài liệu chi tiết hãy nắm vững đặc tả chi tiết, điều kiện kiểm thử, ...
- Đọc chương trình nguồn, hiểu trình tự xử lý chi tiết của hệ thống

**3 việc trên đều là công việc thực thi trên bàn**

# Tu sửa phần mềm đã có

- Bảo trì chương trình nguồn, tạo các môđun mới và dịch lại
- Thực hiện kiểm thử unit và tu chỉnh những mục liên quan có trong tư liệu đặc tả
- Chú ý theo sát tác động của môđun được sửa đến các thành phần khác trong hệ thống



# Phát triển phần mềm mới

- Khi thêm chức năng mới phải phát triển chương trình cho phù hợp với yêu cầu
- Cần tiến hành từ thiết kế, lập trình, gỡ lỗi và kiểm thử unit
- Phản ảnh vào giao diện của phần mềm (thông báo, phiên bản, . . .)

# Kiểm chứng tính nhất quán bằng kiểm thử kết hợp

- Đưa đơn vị (unit) đã được kiểm thử vào hoạt động trong hệ thống
- Điều chỉnh sự tương tác giữa các môđun
- Dùng các dữ liệu trước đây khi kiểm thử để kiểm thử lại tính nhất quán
- Chú ý hiệu ứng làn sóng trong chỉnh sửa

# Kiểm tra khi hoàn thành bảo trì

- Kiểm tra nội dung mô tả có trong tư liệu đặc tả
- Cách ghi tư liệu có phù hợp với mô tả môi trường phần mềm mới hay không ?

# Lập biểu quản lý bảo trì

- Cần quản lý tình trạng bảo trì
- Lập biểu quản lý tình trạng bảo trì
  - Ngày tháng, giờ
  - Nguyên nhân
  - Tóm tắt cách khắc phục
  - Chi tiết khắc phục, hiệu ứng làn sóng
  - Người làm bảo trì
  - Số công

## 10.3 Những vấn đề lưu ý để bảo trì

**Phương pháp cải tiến thao tác bảo trì:**

- **Sáng kiến trong quy trình phát triển phần mềm**
- **Sáng kiến trong quy trình bảo trì phần mềm**
- **Phát triển những kỹ thuật mới cho bảo trì**

# Sáng kiến trong quy trình phát triển phần mềm

- (1) Chuẩn hóa mọi khâu trong phát triển phần mềm
- (2) Người bảo trì chủ chốt tham gia vào giai đoạn phân tích và thiết kế
- (3) Thiết kế để dễ bảo trì

# Sáng kiến trong quy trình bảo trì phần mềm

- (1) Sử dụng các công cụ hỗ trợ phát triển phần mềm
- (2) Chuẩn hóa thao tác bảo trì và thiết bị môi trường bảo trì
- (3) Lưu lại những thông tin sử dụng bảo trì
- (4) Dự án nên cử một người chủ chốt của mình làm công việc bảo trì sau khi dự án kết thúc giai đoạn phát triển

# Phát triển những kỹ thuật mới cho bảo trì

- Công cụ phần mềm hỗ trợ bảo trì
- Cơ sở dữ liệu cho bảo trì
- Quản lý tài liệu, quản lý dữ liệu, quản lý chương trình nguồn, quản lý dữ liệu thử, quản lý sử bảo trì
- Trạm bảo trì tính năng cao trong hệ thống mạng lưới bảo trì với máy chủ thông minh