

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

**PHẦN II –
TIẾN TRÌNH PHẦN MỀM**
Bộ môn Công nghệ phần mềm,
Khoa CNTT&TT, Đại học Cần Thơ



1

Nội dung

- Phân tích và Đặc tả
- Thiết kế
- Lập trình
- Kiểm thử
- Triển khai hệ thống và Bảo trì
- Tiến trình RUP



2

TIẾN TRÌNH PHẦN MỀM

PHẦN II.3 – LẬP TRÌNH



3

Nội dung

- Các chuẩn và thủ tục lập trình
- Chọn ngôn ngữ lập trình
- Nguyên tắc lập trình
- Tài liệu lập trình



4

Các chuẩn và thủ tục lập trình



- Chuẩn và thủ tục giúp lập trình viên
 - Tổ chức các ý định và tránh các lỗi.
 - Các tài liệu theo chuẩn giúp ta quay lại công việc mà không mất dấu những gì đã làm.
 - Các tài liệu theo chuẩn giúp ta định vị các lỗi và tạo ra các thay đổi.
 - Dịch các thiết kế sang mã lệnh.

5

Các chuẩn và thủ tục lập trình



- Chuẩn và thủ tục giúp các thành viên khác
 - Giúp các thành viên khác hiểu được mã lệnh (do lập trình viên viết ra) làm gì và làm như thế nào nhằm thực hiện việc:
 - Tái sử dụng (lập trình viên khác).
 - Kiểm thử (kiểm thử viên).
 - Hiệu chỉnh hay hoàn thiện hệ thống (bảo trì viên).

6

Các chuẩn và thủ tục lập trình



- Chuẩn và thủ tục giúp tạo ra sự tương ứng trực tiếp giữa các thành phần thiết kế và các thành phần cài đặt.
 - Các đặc trưng của chương trình nên giống như các đặc trưng của thiết kế: **nối kết thấp**, **gắn kết cao** và các giao diện rõ ràng =>
 - Các giải thuật, chức năng, giao diện và cấu trúc dữ liệu có thể được theo vết từ thiết kế sang chương trình và ngược lại một cách dễ dàng.

7

Chọn ngôn ngữ lập trình



- Theo loại phần mềm
 - Phần mềm hệ thống: C, C++
 - Phần mềm thời gian thực: C, C++, Assembly
 - Phần mềm nhúng: C++, Java
 - Phần mềm nghiệp vụ (HTTT):
 - CSDL: Oracle, MySQL, SQL Server
 - NNLT: VB, Foxpro, VC++
 - Phần mềm trí tuệ nhân tạo: Prolog, Lisp
 - Phần mềm (dịch vụ) Web: PHP, ASP, Java Script
- Theo đặc trưng của ngôn ngữ
- Theo năng lực và kinh nghiệm của nhóm phát triển phần mềm
- Theo yêu cầu của khách hàng

8

Nguyên tắc lập trình



- Dù bất cứ ngôn ngữ lập trình nào được sử dụng, mỗi thành phần chương trình đều liên quan tới ít nhất 3 khía cạnh chính:
 - Cấu trúc điều khiển.
 - Giải thuật.
 - Cấu trúc dữ liệu.

9

Cấu trúc điều khiển



- Tạo ra mã lệnh dễ đọc
Ví dụ: Cho đoạn chương trình

```
benefit = minimum;
if (age < 75) goto A;
benefit = maximum;
goto C;
if (AGE < 65) goto B;
if (AGE < 55) goto C;
A: if (AGE < 65) goto B;
   benefit = benefit * 1.5 + bonus;
   goto C;
B:   if (age < 55) goto C;
   benefit = benefit * 1.5;
C:   next statement
```

Điều khiển của đoạn chương trình này bỏ qua một số câu lệnh =>
Rất khó đọc.

10

Cấu trúc điều khiển



- Ví dụ: Đoạn chương trình

```
benefit = minimum;
if (age < 75) goto A;
benefit = maximum;
goto C;
if (AGE < 65) goto B;
if (AGE < 55) goto C;
A:  if (AGE < 65) goto B;
    benefit = benefit * 1.5 + bonus;
    goto C;
B:  if (age < 55) goto C;
    benefit = benefit * 1.5;
C:  next statement
```

Được viết lại:

```
if (age < 55) benefit = minimum;
elseif (AGE < 65) benefit = minimum + bonus;
elseif (AGE < 75) benefit = minimum * 1.5 + bonus;
else benefit = maximum;
```

11

Cấu trúc điều khiển



- Xây dựng chương trình từ các khối mô đun nhằm làm cho hệ thống dễ hiểu, dễ kiểm thử và dễ bảo trì.
- Viết mã lệnh có tính tổng quát nhưng không làm ảnh hưởng đến sự thực hiện và sự hiệu biết.
- Sử dụng các tên tham số và các chú thích để biểu thị sự kết hợp giữa các thành phần.
- Tạo sự phụ thuộc giữa các thành phần một cách rõ ràng.

12

Giải thuật



- Mục tiêu chung: tăng hiệu quả thực hiện (tốc độ).
- Tính hiệu quả trong sự thực hiện có thể có các chi phí ẩn.
 - Tốn thời gian để viết mã lệnh thực thi nhanh hơn.
 - Tốn thời gian để kiểm thử mã lệnh.
 - Tốn thời gian để người dùng hiểu mã lệnh.
 - Tốn thời gian để hiệu chỉnh mã lệnh, nếu cần.

=> *Cân bằng thời gian thực thi với chất lượng thiết kế, chuẩn và các yêu cầu của khách hàng.*

13

Cấu trúc dữ liệu



- Ta nên định dạng và lưu trữ dữ liệu sao cho việc quản lý và thực hiện trên dữ liệu là không phức tạp.
- Các cấu trúc dữ liệu: danh sách, ngăn xếp, cây, v.v.
- Một số kỹ thuật sử dụng cấu trúc dữ liệu để tổ chức chương trình
 - Làm cho chương trình đơn giản.
 - Sử dụng cấu trúc dữ liệu để xác định cấu trúc chương trình.

14

Cấu trúc dữ liệu



- Làm cho chương trình đơn giản

Ví dụ: xác định thuế thu nhập

- Với \$10,000 thu nhập đầu tiên, thuế là 10%
- Với \$10,000 thu nhập tiếp theo (trên \$10,000), thuế là 12%
- Với \$10,000 thu nhập tiếp theo (trên \$20,000), thuế là 15%
- Với \$10,000 thu nhập tiếp theo (trên \$30,000), thuế là 18%
- Với bất cứ thu nhập nào trên \$40,000, thuế là 20%

15

Cấu trúc dữ liệu



```
tax = 0;
if (taxable_income == 0) goto EXIT;
if (taxable_income > 10000) tax = tax + 1000;
else {
    tax = tax + .10*taxable_income;
    goto EXIT;
}
if (taxable_income > 20000) tax = tax + 1200;
else {
    tax = tax + .12*(taxable_income-10000);
    goto EXIT;
}
if (taxable_income > 30000) tax = tax + 1500;
else {
    tax = tax + .15*(taxable_income-20000);
    goto EXIT;
}
if (taxable_income < 40000){
    tax = tax + .18*(taxable_income-30000);
    goto EXIT;
}
else
    tax = tax + 1800. + .20*(taxable_income-40000);
EXIT;
```

16

Cấu trúc dữ liệu



- Xác định bảng thuế

Bracket	Base	Percent
0	0	10
10,000	1000	12
20,000	2200	15
30,000	3700	18
40,000	55000	20

- Giải thuật được đơn giản hóa
for (int i=2, level=1; i <= 5; i++)
 if (taxable_income > bracket[i])
 level = level + 1;
tax = base[level]+percent[level]*(taxable_income-bracket[level]);

Cấu trúc dữ liệu



- Sử dụng cấu trúc dữ liệu để xác định cấu trúc chương trình
 - Cấu trúc dữ liệu chi phối tổ chức chương trình.
 - Cấu trúc dữ liệu có thể tác động đến việc chọn ngôn ngữ.
 - Nếu cấu trúc dữ liệu đệ quy, chọn ngôn ngữ cho phép lập trình các thủ tục đệ quy.
 - Nếu cấu trúc dữ liệu là danh sách, chọn ngôn ngữ Lisp.
 - ...

Nguyên tắc chung



- Cục bộ hóa input và output
 - Các phần đọc input và sinh ra output là những phần có khả năng phải thay đổi nhiều nhất khi phần cứng và phần mềm được sửa đổi.
 - Chúng được tách riêng khỏi phần còn lại của mã lệnh nhằm giúp ta dễ hiểu và dễ thay đổi hệ thống hơn.
- Tái sử dụng
 - Tái sử dụng của nhà sản xuất: tạo ra các thành phần mà chúng được tái sử dụng trong các ứng dụng nối tiếp.
 - Tái sử dụng của người tiêu thụ: tái sử dụng các thành phần mà ban đầu chúng được phát triển cho các dự án khác.

19

Nguyên tắc chung



- Một số lưu ý khi lập trình
 - Bắt ngoại lệ.
 - Quên cấp phát và thu hồi bộ nhớ.
 - Vấn đề làm tròn số.
 - Hạn chế gọi nhiều lần các hàm có qui mô nhỏ.
 - Truyền tham số.
 - Tránh dùng mảng nhiều chiều.
 - Sử dụng các phép toán nhanh.

20

Tài liệu chương trình



- Tài liệu chương trình: một tập các mô tả giải thích với người đọc về các chương trình làm gì và chúng làm như thế nào.
- Các loại tài liệu chương trình
 - Tài liệu nội là mô tả được viết trực tiếp trong mã lệnh và được sử dụng bởi những người đọc mã lệnh.
 - Tài liệu ngoại gồm những tài liệu khác và được sử dụng bởi những người không trực tiếp đọc mã lệnh.

21

Tài liệu chương trình



- Tài liệu nội
 - Khóai chú thích ở phần đầu của từng thành phần: ai, cái gì, tại sao, khi nào, như thế nào và ở đâu.
 - Các chú thích khác giúp ta hiểu chương trình rõ hơn: thêm thông tin mới, không phải diễn đạt lại những gì đã rõ ràng.
 - Nhãn cho câu lệnh và tên biến phải có nghĩa.
 - Định dạng để gia tăng sự hiểu biết: thụt lề, khoảng trắng.

22

Tài liệu chương trình



- Tài liệu ngoại
 - Mô tả vấn đề: mô tả các lựa chọn được xem xét để giải quyết vấn đề và tại sao một giải pháp cụ thể được chọn.
 - Mô tả giải thuật: giải thích từng giải thuật được sử dụng trong thành phần (công thức, ranh giới, các điều kiện đặc biệt, bắt lỗi, v.v).
 - Mô tả dữ liệu: từ điển dữ liệu.

23

TIẾN TRÌNH PHẦN MỀM

PHẦN II.4 – KIỂM THỬ



24

Nội dung

- Phần 1 – Kiểm thử chương trình
- Phần 2 – Kiểm thử hệ thống

25

Phần 1 - Nội dung

- Các lỗi phần mềm
- Các vấn đề trong kiểm thử
- Kiểm thử đơn vị
- Kiểm thử tích hợp
- Lập kế hoạch kiểm thử
- Các công cụ kiểm thử tự động

26

Lỗi phần mềm



- Mục đích của kiểm thử: phát hiện ra các lỗi.
- Một kiểm thử là thành công chỉ khi lỗi được phát hiện.
 - **Nhận dạng lỗi** là quá trình xác định lỗi nào đã gây ra sự thất bại.
 - **Hiệu chỉnh lỗi** là quá trình tạo ra các thay đổi đối với hệ thống nhằm loại bỏ các lỗi.

27

Lỗi phần mềm



- Các loại lỗi
 - Lỗi giải thuật (các lỗi giải thuật điển hình)
 - Lỗi giải thuật xuất hiện khi logic hay giải thuật của thành phần **không** tạo ra kết xuất đúng với đầu vào xác định vì đã có sai sót trong các bước xử lý như:
 - Rẽ nhánh quá sớm
 - Rẽ nhánh quá trễ
 - Kiểm thử cho điều kiện sai
 - Quên khởi tạo giá trị ban đầu của biến hay đặt vòng lặp bất biến
 - Quên kiểm thử cho điều kiện đặc biệt
 - So sánh giá trị của các kiểu dữ liệu không phù hợp
 - Lỗi cú pháp

28

Lỗi phần mềm



- Các loại lỗi
 - Lỗi về độ chính xác và tính toán
 - Sự thực hiện của một công thức là sai hoặc không tính ra kết quả với độ chính xác mong muốn.
 - Lỗi tài liệu
 - Tài liệu không phù hợp với cái mà chương trình làm.
 - Lỗi về quá tải hay dung lượng
 - Sự thực hiện của hệ thống là không thể chấp nhận khi đạt đến các ranh giới nào đó.

29

Lỗi phần mềm



- Các loại lỗi
 - Lỗi hợp tác và thời gian.
 - Lỗi thực hiện.
 - Hệ thống không thực hiện với tốc độ được mô tả
 - Lỗi quy trình và chuẩn.

30

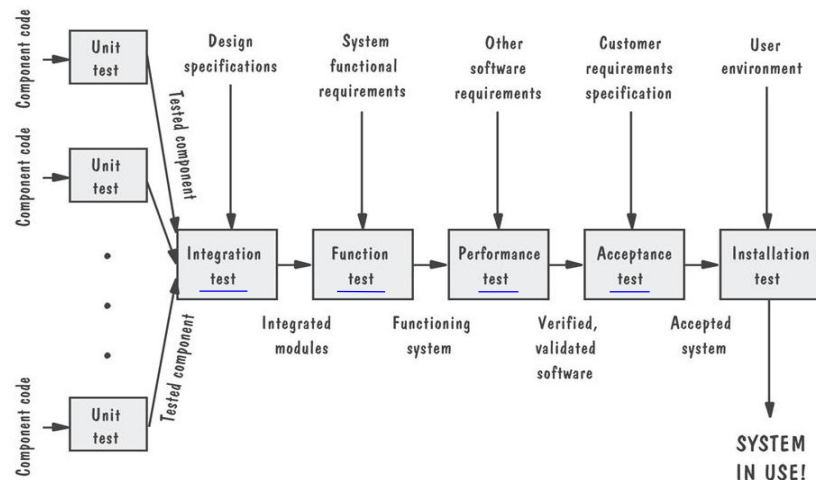
Các vấn đề trong kiểm thử

- Tổ chức kiểm thử
 - Kiểm thử đơn vị
 - Kiểm thử tích hợp
 - Kiểm thử chức năng
 - Kiểm thử sự thực hiện
 - Kiểm thử chấp nhận
 - Kiểm thử cài đặt

31

Các vấn đề trong kiểm thử

- Tổ chức kiểm thử



Các vấn đề trong kiểm thử



- Quan điểm kiểm thử
 - Các chương trình được xem như những thành phần của một hệ thống lớn hơn, không như tài sản của những người đã viết ra chúng.

33

Các vấn đề trong kiểm thử



- Người kiểm thử: nhóm kiểm thử độc lập
 - Tránh sự mâu thuẫn.
 - Tăng tính khách quan.
 - Cho phép kiểm thử và lập trình đồng thời.

34

Các vấn đề trong kiểm thử



- Tổng quan về các phương pháp kiểm thử
 - Kiểm thử hộp đóng hay hộp đen: sử dụng **chức năng** của đối tượng kiểm thử để kiểm thử theo nhiều cách.
 - Kiểm thử hộp mở hay hộp trắng: sử dụng **cấu trúc** của đối tượng kiểm thử để kiểm thử theo nhiều cách.

35

Các vấn đề trong kiểm thử



- Kiểm thử hộp đen
 - Đặc điểm
 - Đối tượng kiểm thử như một hộp đen, thông qua giao diện để đưa dữ liệu vào và nhận dữ liệu ra.



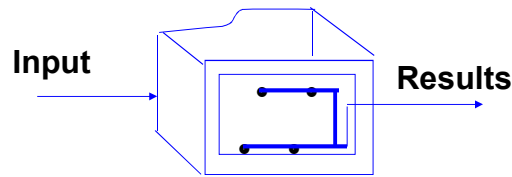
- Độc lập với các ràng buộc bị tác động bởi cấu trúc bên trong và tính logic của đối tượng.

36

Các vấn đề trong kiểm thử



- Kiểm thử hộp trắng
 - Phương pháp thiết kế trường hợp kiểm thử sử dụng cấu trúc điều khiển của thiết kế thuật toán để dẫn ra các trường hợp kiểm thử.



37

Các vấn đề trong kiểm thử



- Các yếu tố ảnh hưởng đến việc chọn phương pháp kiểm thử:
 - Số đường logic có thể có.
 - Trạng thái của dữ liệu input.
 - Số lượng tính toán có liên quan.
 - Độ phức tạp của giải thuật.

38

Kiểm thử đơn vị



- Kiểm thử đơn vị: kiểm thử từng thành phần hay từng mô đun của phần mềm.
 - Ví dụ: C++ - phương thức, lớp; C/Pascal – hàm;
- Các bước kiểm thử đơn vị
 - Xem lại mã lệnh.
 - Biên dịch mã lệnh và loại bỏ bất cứ lỗi cú pháp nào còn lại
 - Phát triển các trường hợp kiểm thử để chỉ ra rằng input được chuyển đổi chính xác thành output mong đợi.

39

Kiểm thử đơn vị



- Các bước chọn trường hợp kiểm thử
 - Xác định các mục tiêu kiểm thử
 - Lựa chọn các trường hợp kiểm thử
 - Định nghĩa một kiểm thử

40

Kiểm thử đơn vị



- Tính toàn diện của kiểm thử
 - Kiểm thử câu lệnh
 - Kiểm thử rẽ nhánh
 - Kiểm thử đường đi
 - Kiểm thử sử dụng sự định nghĩa
 - Kiểm thử tất cả sử dụng
 - Kiểm thử tất cả sử dụng vị từ / kiểm thử một số sử dụng tính toán
 - Kiểm thử một số sử dụng vị từ / kiểm thử tất cả sử dụng tính toán

41

Kiểm thử đơn vị - Kiểm thử đường đi

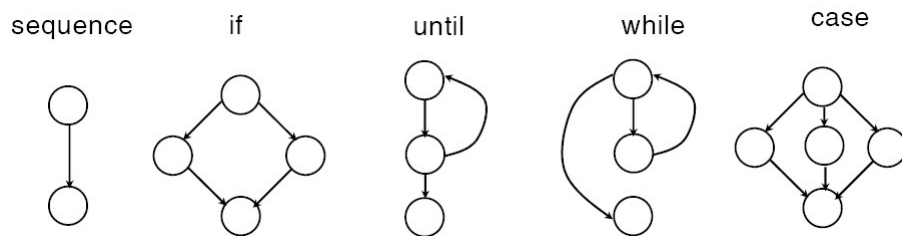


- Kiểm thử đường đi: Kiểm thử tất cả các hướng đi có thể
 - Thiết lập đồ thị dòng chảy.
 - Liệt kê các đường thực thi độc lập cơ bản.
 - Sinh các trường hợp kiểm thử cho các đường thực thi đó.

42

Kiểm thử đường đi

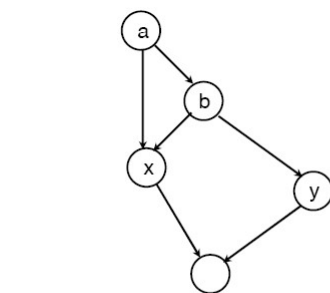
- Thiết lập đồ thị dòng chảy
 - Mỗi nút hình tròn biểu diễn một hoặc một vài tác vụ.
 - Cạnh có hướng miêu tả đường thực thi.
 - Đồ thị dòng chảy được xây dựng từ lưu đồ thuật giải.



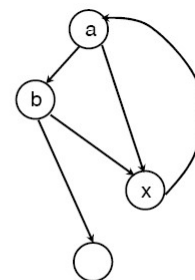
43

Kiểm thử đường đi

- Thiết lập đồ thị dòng chảy
 - Phải phân rã tất cả các điều kiện phức trở thành các điều kiện đơn.



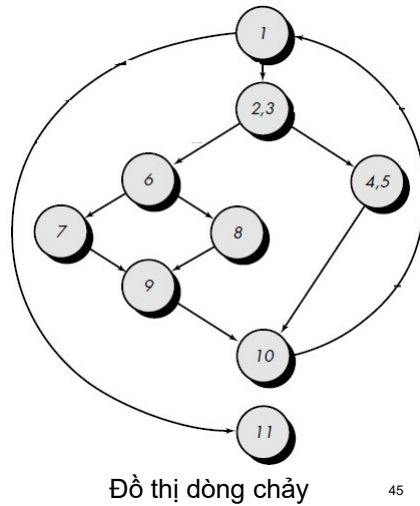
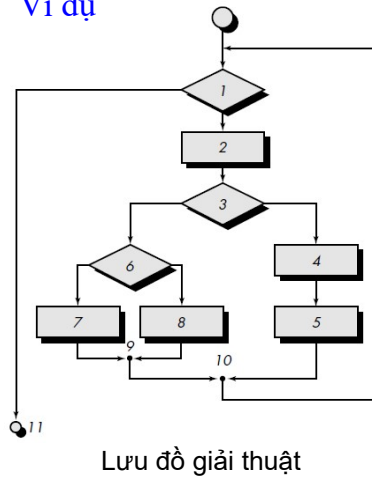
if a and b then y else x



while a or b do x

Kiểm thử đường đi

- Ví dụ



45

Kiểm thử đường đi

- Liệt kê các **đường độc lập cơ bản**

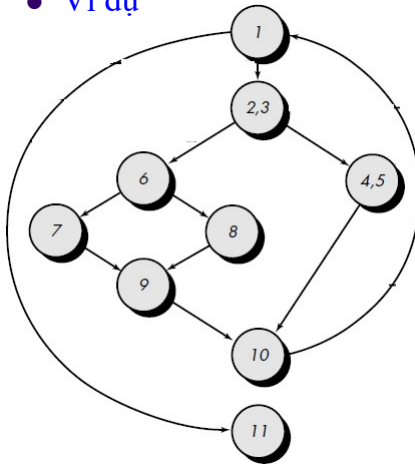
- Từ nút bắt đầu đến nút kết thúc, các đường thực thi cơ bản được liệt kê theo một thứ tự nào đó để đảm bảo rằng đường đang liệt kê ít nhất đi qua một cạnh chưa được duyệt qua bởi các đường đã liệt kê trước đó.
- Tổng số đường thực thi cơ bản độc lập nhau được tính bằng $V = P + 1$ trong đó P là số nút phân nhánh.

46

Kiểm thử đường đi



- Ví dụ



Đồ thị dòng chảy

Các đường độc lập cơ bản:

1. 1 - 11
2. 1 - 2,3 - 4,5 - 10 - 1 - 11
3. 1 - 2,3 - 6 - 7 - 9 - 10 - 1 - 11
4. 1 - 2,3 - 6 - 8 - 9 - 10 - 1 - 11

47

Kiểm thử đường đi

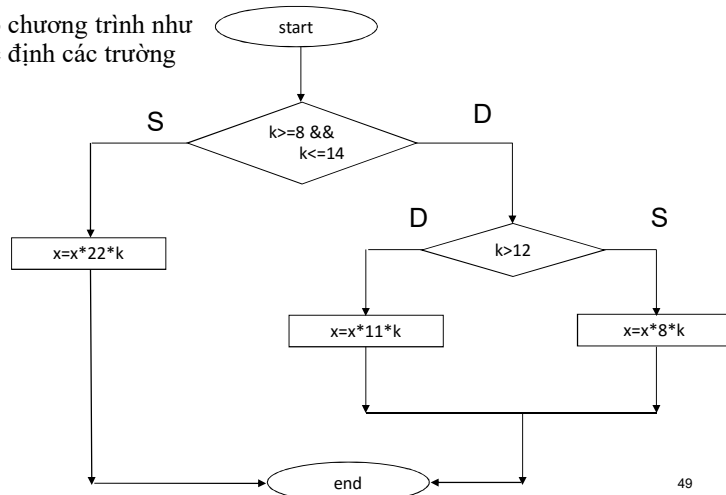


- Sinh ra các trường hợp kiểm thử (test-case)
 - Thiết lập một test-case cho mỗi đường thực thi cơ bản.
 - Dựa vào giải thuật để
 - Đưa ra dữ liệu input.
 - Tính ra dữ liệu output hay đáp ứng mong đợi của giải thuật.
 - Có thể không tạo ra được test-case cho một đường thực thi nào đó.

48

Kiểm thử đường đi

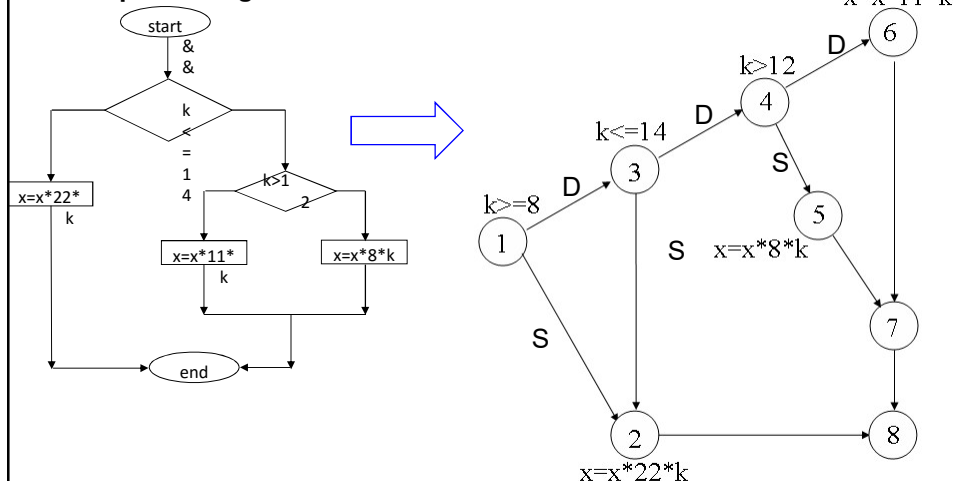
Ví dụ: cho lưu đồ chương trình như hình bên, hãy xác định các trường hợp kiểm thử?



49

Kiểm thử đường đi

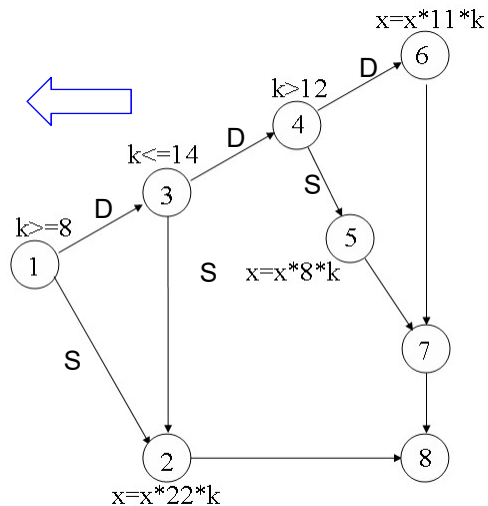
Xác định đường đi



Kiểm thử đường đi

Lập các Trường hợp kiểm thử

Đường đi	Tham số	Kết quả
1-2-8	k=5, x=10	1100
1-3-2-8	k=15, x=10	3300
1-3-4-5-7-8	k=10, x=10	800
1-3-4-6-7-8	k=13, x=10	1430



Kiểm thử tích hợp

- Kiểm thử tích hợp nhằm phát hiện các lỗi liên quan đến sự giao tiếp giữa các thành phần.

- Các dạng kiểm thử tích hợp

- Kiểm thử từ dưới lên
- Kiểm thử từ trên xuống
- Kiểm thử từ trên xuống được hiệu chỉnh
- Kiểm thử sandwich (kẹp)
- Kiểm thử sandwich được hiệu chỉnh
- Kiểm thử Big-bang

Tích hợp dần

Tích hợp đồng thời một lúc

Kiểm thử tích hợp



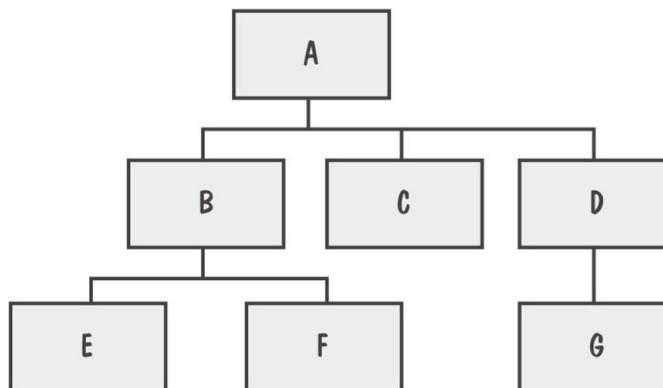
- Thuật ngữ
 - **Driver**: một chương trình mà nó gọi một thành phần (component) cụ thể và truyền một trường hợp kiểm thử cho nó.
 - **Stub**: một chương trình (có mục đích đặc biệt) mô phỏng hoạt động của thành phần bị thiếu.

53

Kiểm thử tích hợp



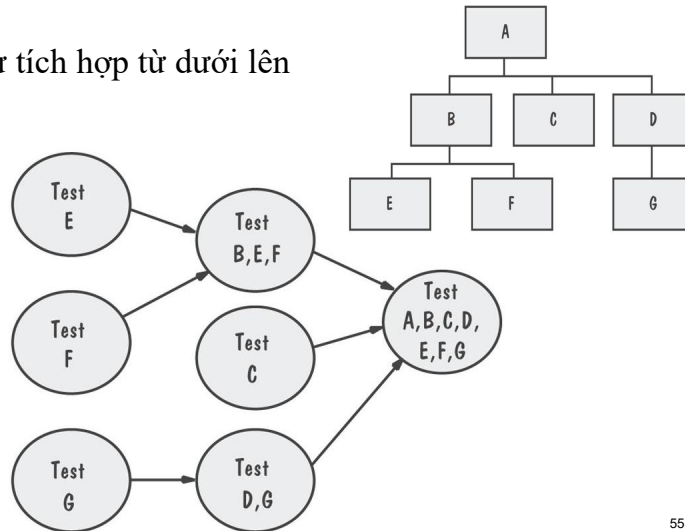
- **Ví dụ**: Cho một hệ thống với các thành phần được phân cấp



54

Kiểm thử tích hợp

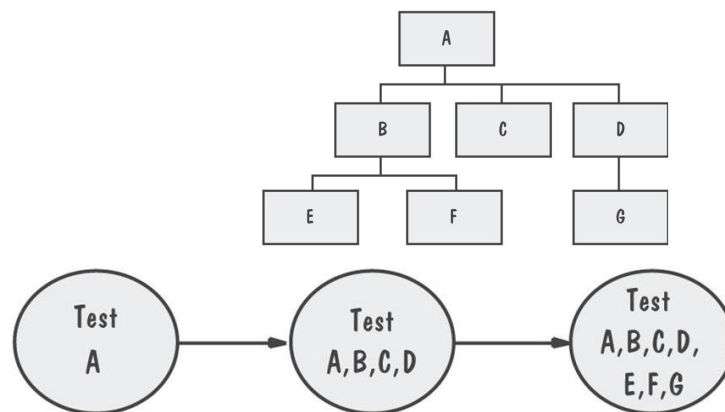
- Kiểm thử tích hợp từ dưới lên



55

Kiểm thử tích hợp

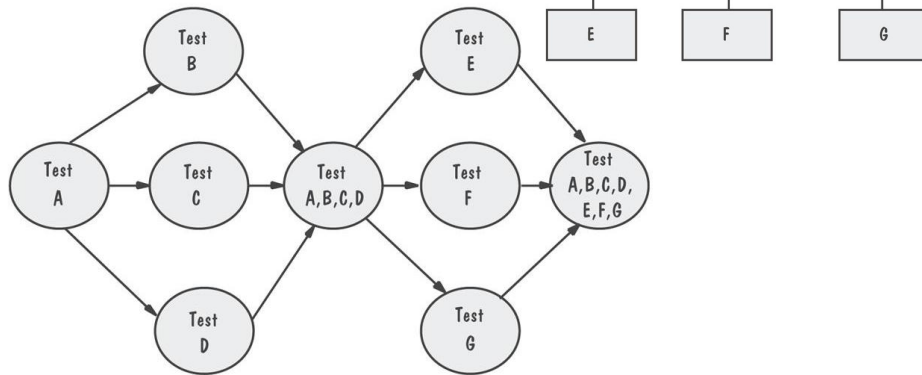
- Kiểm thử tích hợp từ trên xuống



56

Kiểm thử tích hợp

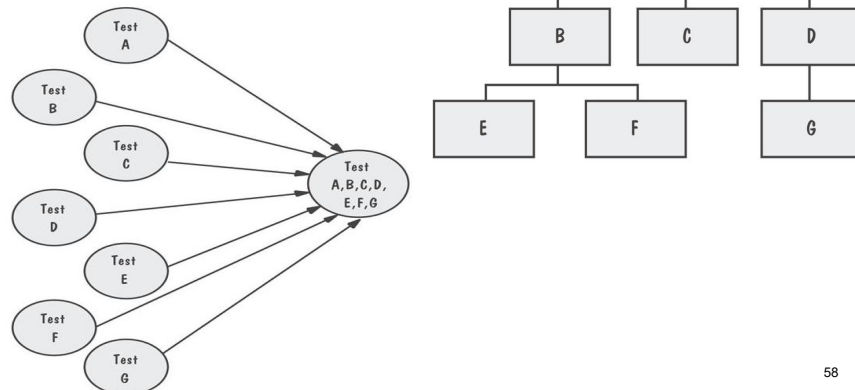
- Kiểm thử tích hợp từ trên xuống được hiệu chỉnh



57

Kiểm thử tích hợp

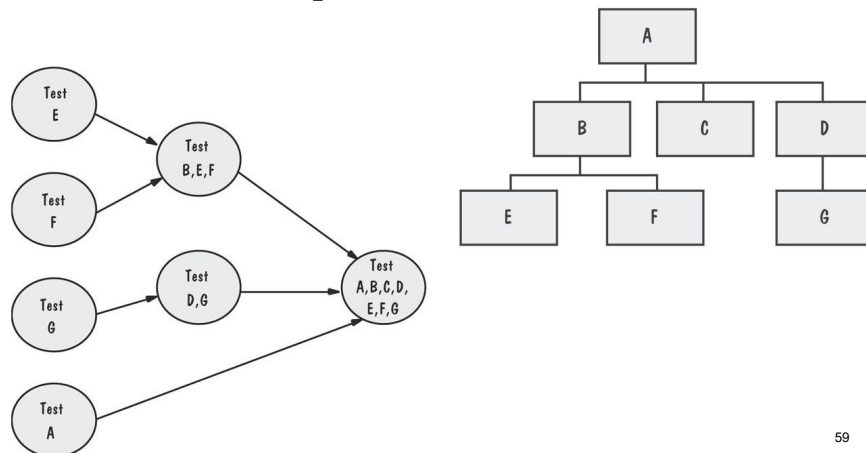
- Kiểm thử tích hợp Big Bang



58

Kiểm thử tích hợp

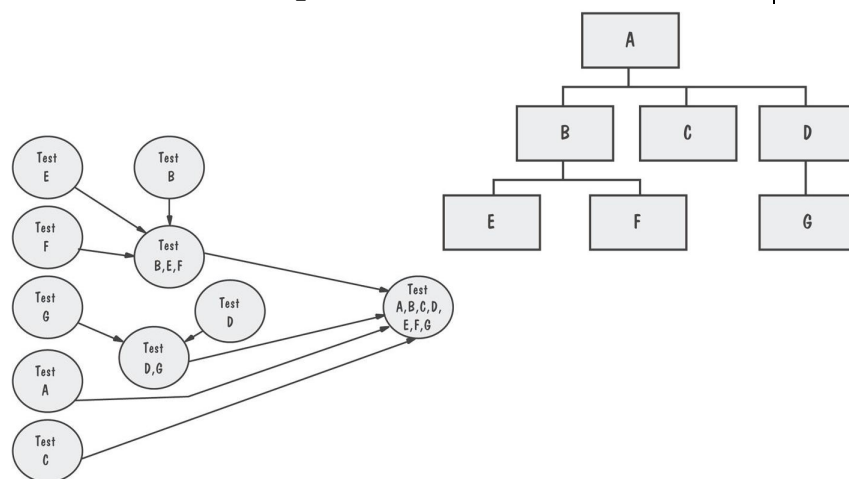
- Kiểm thử tích hợp sandwich



59

Kiểm thử tích hợp

- Kiểm thử tích hợp sandwich được hiệu chỉnh



60

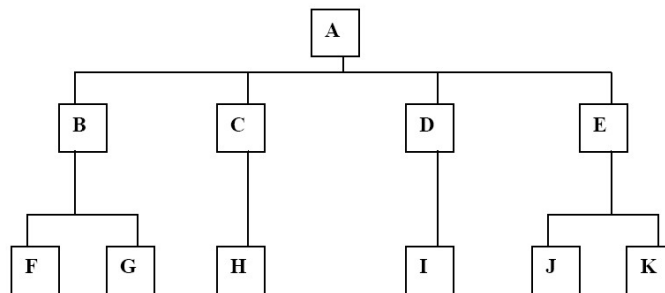
Kiểm thử tích hợp



• So sánh các dạng kiểm thử tích hợp

	Từ dưới lên	Từ trên xuống	Từ trên xuống được hiệu chỉnh	Big-bang	Sandwich	Sandwich được hiệu chỉnh
Sự tích hợp	Early	Early	Early	Late	Early	Early
Thời gian cho lập trình công việc cơ bản	Late	Early	Early	Late	Early	Early
Các driver thành phần được cần đến	Yes	No	Yes	Yes	Yes	Yes
Các Stub được cần	No	Yes	Yes	Yes	Yes	Yes
Quan hệ song song của công việc lúc bắt đầu	Medium	Low	Medium	High	Medium	High
Khả năng kiểm thử các đường dẫn cụ thể	Easy	Hard	Easy	Easy	Medium	Easy
Khả năng lập kế hoạch và kiểm soát chuỗi sự kiện	Easy	Hard	Hard	Easy	Hard	Hard

Bài tập



Xác định dạng kiểm thử tích hợp?

Phần đọc thêm



63

Lập kế hoạch kiểm thử



- Thiết lập các mục tiêu kiểm thử
- Thiết kế các trường hợp kiểm thử
- Viết các trường hợp kiểm thử
- Kiểm tra các trường hợp kiểm thử
- Thực thi các kiểm thử
- Đánh giá các kết quả kiểm thử

64

Lập kế hoạch kiểm thử



- Mục đích của kế hoạch
Kế hoạch kiểm thử giải thích:
 - Ai thực hiện kiểm thử
 - Tại sao các kiểm thử được thực hiện
 - Cách thức các kiểm thử được kiểm soát
 - Khi nào các kiểm thử được thực hiện

65

Lập kế hoạch kiểm thử



- Nội dung của kế hoạch:
 - Các mục tiêu của kiểm thử
 - Cách thực hiện kiểm thử
 - Chuẩn được sử dụng để xác định khi nào kiểm thử hoàn thành

66

Công cụ kiểm thử tự động



- Công cụ phân tích mã lệnh
 - Phân tích tĩnh
 - Bộ phân tích mã lệnh
 - Bộ kiểm tra cấu trúc
 - Bộ phân tích dữ liệu
 - Bộ kiểm tra trình tự
 - Phân tích động
 - Các bộ giám sát chương trình: xem và báo cáo lại các hoạt động của chương trình

67

Công cụ kiểm thử tự động



- Công cụ thực hiện kiểm thử
 - Capture và replay
 - Các stub và driver
 - Môi trường kiểm thử tự động
- Bộ sinh các trường hợp kiểm thử

68

Phần 2 - Nội dung



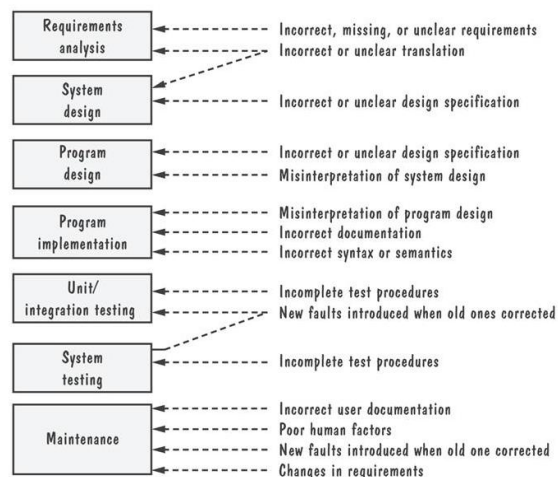
- Các nguyên lý của kiểm thử hệ thống
- Kiểm thử chức năng
- Kiểm thử sự thực thi
- Tính tin cậy, tính sẵn có và tính có thể bảo trì
- Kiểm thử chấp nhận
- Kiểm thử sự cài đặt
- Tài liệu kiểm thử

69

Các nguyên lý của kiểm thử hệ thống



- Nguồn gốc của các lỗi phần mềm trong suốt quá trình phát triển



Các nguyên lý của kiểm thử hệ thống



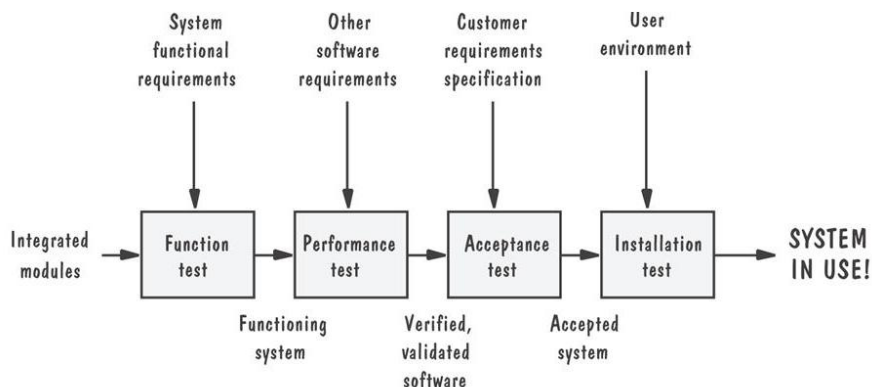
- Quy trình kiểm thử hệ thống
 - Kiểm thử chức năng: hệ thống được tích hợp có thực hiện như được cam kết trong đặc tả yêu cầu?
 - Kiểm thử thực hiện: các yếu tố phi chức năng có được đáp ứng?
 - Kiểm thử chấp nhận: hệ thống có phải là cái mà khách hàng mong đợi?
 - Kiểm thử sự cài đặt: hệ thống có vận hành ở chỗ khách hàng không?

71

Các nguyên lý của kiểm thử hệ thống



- Các bước trong quy trình kiểm thử hệ thống



Các nguyên lý của kiểm thử hệ thống



- Những kỹ thuật được sử dụng trong kiểm thử hệ thống
 - Kế hoạch tích hợp hay xây dựng.
 - Kiểm thử hồi quy.
 - Quản lý cấu hình.

73

Các nguyên lý của kiểm thử hệ thống



- Kế hoạch tích hợp hay xây dựng
 - Mô tả các hệ thống con (spin) được kiểm thử.
 - Mô tả cách thức, nơi chốn, thời gian và người thực hiện các kiểm thử.

74

Các nguyên lý của kiểm thử hệ thống



- Ví dụ về kế hoạch xây dựng cho hệ thống viễn thông

Spin	Chức năng	Bắt đầu kiểm thử	Kết thúc kiểm thử
0	Exchange	1 September	15 September
1	Area code	30 September	15 October
2	State/province/district	25 October	5 November
3	Country	10 November	20 November
4	International	1 December	15 December

75

Các nguyên lý của kiểm thử hệ thống



- Kiểm thử hồi quy
 - Nhận dạng các lỗi mới mà chúng được gây ra bởi các lỗi hiện tại được hiệu chỉnh.
 - Kiểm tra phiên bản hay phát hành mới để xác nhận rằng nó vẫn thực hiện cùng các chức năng theo cùng cách như phiên bản hay phát hành cũ.

76

Các nguyên lý của kiểm thử hệ thống



- Các bước kiểm thử hồi quy
 - Chèn vào mã lệnh mới.
 - Kiểm thử các chức năng được biết sẽ bị ảnh hưởng bởi mã lệnh mới.
 - Kiểm thử các chức năng cần thiết của phiên bản m để xác nhận rằng chúng vẫn hoạt động chính xác.
 - Tiếp tục kiểm thử chức năng của phiên bản $m + 1$.

77

Các nguyên lý của kiểm thử hệ thống



- Nhóm kiểm thử
 - Kiểm thử viên chuyên nghiệp: tổ chức và thực hiện các kiểm thử.
 - Nhà phân tích: người đã tạo ra các đặc tả.
 - Nhà thiết kế hệ thống: hiểu giải pháp được đề nghị.
 - Chuyên gia quản lý cấu hình: giúp kiểm soát các sửa chữa.
 - Người dùng: đánh giá các phát sinh.

78

Kiểm thử chức năng



- Mục đích và vai trò
 - So sánh sự thực hiện thực tế của hệ thống với các yêu cầu của nó.
 - Phát triển các trường hợp kiểm thử dựa trên tài liệu yêu cầu.



79

Kiểm thử chức năng



- Tạo ra các trường hợp kiểm thử chức năng
 - Vẽ đồ thị nhân quả từ các yêu cầu
 - Chuyển đồ thị thành bảng quyết định
 - Mỗi cột trong bảng quyết định tương ứng với một trường hợp kiểm thử chức năng

80

Kiểm thử chức năng



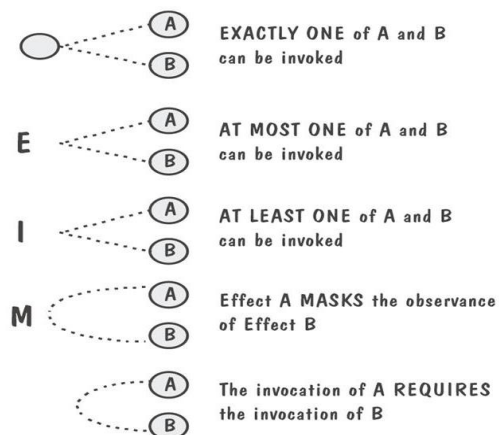
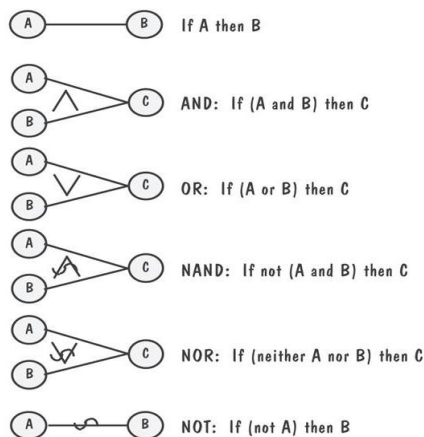
- Đồ thị nhân quả
 - Một đồ thị logic phản ánh quan hệ logic giữa nguyên nhân (các input) và kết quả (output hay sự biến đổi)
 - Vẽ đồ thị nhân quả từ các yêu cầu:
 - Các yêu cầu được phân tách sao cho mỗi yêu cầu mô tả một chức năng.
 - Mô tả các nguyên nhân, các kết quả (đó là các nút trong đồ thị với các nhân nằm bên trái và các quả nằm bên phải).
 - Vẽ mối quan hệ giữa chúng.

81

Kiểm thử chức năng



- Ký hiệu của đồ thị nhân quả
- Ký hiệu bổ sung của đồ thị nhân quả



Kiểm thử sự thực thi



- Mục đích và vai trò
 - Được sử dụng để kiểm tra:
 - Sự tính toán
 - Tốc độ đáp ứng
 - Độ chính xác của kết quả
 - Khả năng truy cập dữ liệu
 - Được thiết kế và quản lý bởi nhóm kiểm thử.



83

Kiểm thử sự thực thi



- Các loại kiểm thử sự thực thi
 - Kiểm thử hiệu suất
 - Kiểm thử dung lượng CTDL
 - Kiểm thử cấu hình
 - Kiểm thử tính tương thích
 - Kiểm thử hồi quy
 - Kiểm thử sự bảo mật
 - Kiểm thử sự điều hòa thời gian
 - Kiểm thử môi trường
 - Kiểm thử chất lượng
 - Kiểm thử sự hồi phục
 - Kiểm thử bảo trì
 - Kiểm thử tài liệu
 - Kiểm thử tính dễ sử dụng

84

Tính tin cậy, tính sẵn có và tính có thể bảo trì



- Định nghĩa

- Tính tin cậy của phần mềm: vận hành mà không có lỗi dưới một điều kiện xác định trong một khoảng thời gian cho trước.
- Tính sẵn có của phần mềm: vận hành thành công theo sự đặc tả tại một điểm thời gian xác định.
- Tính có thể bảo trì của phần mềm: với một điều kiện sử dụng xác định, một hoạt động bảo trì có thể được thực hiện trong khoảng thời gian, thủ tục và tài nguyên xác định.

85

Tính tin cậy, tính sẵn có và tính có thể bảo trì



- Các mức độ khác nhau về lỗi

- Thảm khốc: gây ra sự chết chóc hoặc sự thất bại của hệ thống.
- Then chốt: gây ra tổn hại rất xấu hay sự hư hại hệ thống chính dẫn đến sự thất bại về nhiệm vụ.
- Lề: gây ra tổn hại phụ hay sự hư hại hệ thống phụ dẫn đến sự ngừng trệ, mất tính sẵn có hay giảm sút trong công việc.
- Thứ yếu: không đủ nghiêm trọng để gây ra tổn hại hay sự hư hại hệ thống nhưng sẽ dẫn đến việc bảo trì và sửa chữa không có kế hoạch.

86

Kiểm thử chấp nhận



- Mục đích và vai trò
 - Cho phép khách hàng và người dùng xác định xem hệ thống được xây dựng có đáp ứng được yêu cầu và sự mong đợi của họ hay không.
 - Được viết, quản lý và đánh giá bởi khách hàng.

87

Kiểm thử chấp nhận



- Các loại kiểm thử chấp nhận
 - Kiểm thử thử nghiệm (pilot): cài đặt hệ thống trên cơ sở thực nghiệm.
 - Kiểm thử alpha: kiểm thử của người dùng trong tổ chức hay công ty phát triển phần mềm.
 - Kiểm thử beta: kiểm thử của khách hàng.
 - Kiểm thử song song: một hệ thống mới vận hành song song với hệ thống cũ.



88

Kiểm thử sự cài đặt



- Trước khi kiểm thử
 - Cấu hình hệ thống
 - Gắn số và loại thiết bị
 - Thiết lập sự giao tiếp với hệ thống khác
- Kiểm thử
 - Kiểm thử hồi quy: kiểm tra rằng hệ thống được cài đặt một cách chính xác và hoạt động.

89

Tài liệu kiểm thử

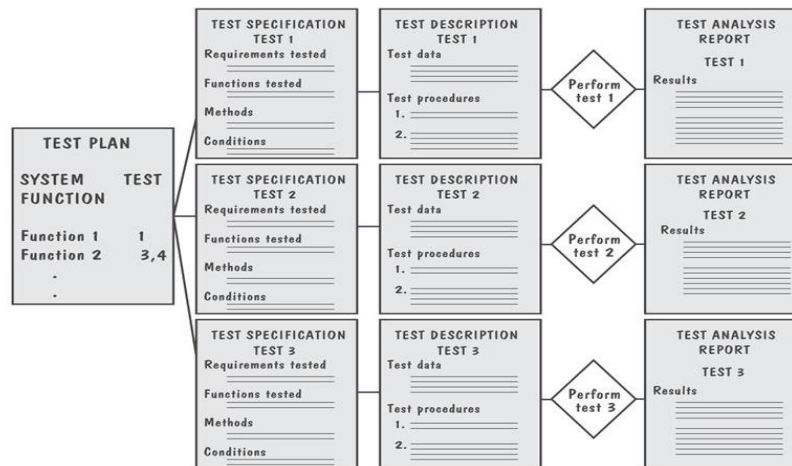


- Kế hoạch kiểm thử: mô tả hệ thống và kế hoạch để kiểm tra tất cả các chức năng và các đặc trưng.
- Đặc tả và đánh giá kiểm thử: chi tiết từng kiểm thử và định nghĩa tiêu chí để đánh giá từng đặc điểm.
- Mô tả kiểm thử: thủ tục và dữ liệu kiểm thử cho từng kiểm thử.
- Báo cáo phân tích kiểm thử: các kết quả của từng kiểm thử.

90

Tài liệu kiểm thử

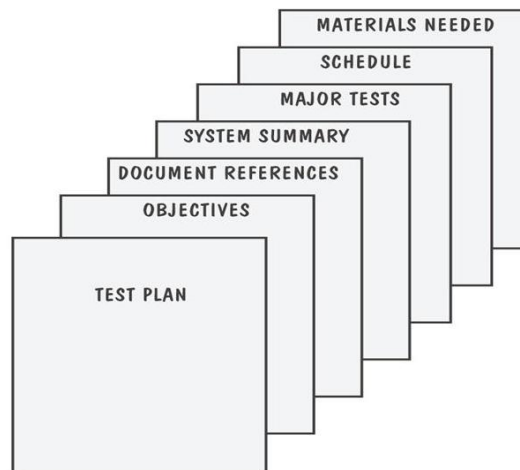
- Các tài liệu được tạo ra trong suốt sự kiểm thử



91

Tài liệu kiểm thử

- Các phần của một kế hoạch kiểm thử



92

Tài liệu kiểm thử



- Sự mô tả kiểm thử gồm:
 - Phương tiện kiểm soát
 - Dữ liệu
 - Thủ tục

93

Tài liệu kiểm thử



- Báo cáo phân tích kiểm thử
 - Ghi kết quả kiểm thử.
 - Cung cấp thông tin cần để sao chép lại sự thất bại, định vị và sửa tận gốc vấn đề.
 - Cung cấp thông tin cần thiết để xác định xem dự án có hoàn thành hay không.
 - Thiết lập sự tin cậy trong sự thực thi của hệ thống.

94

Tài liệu kiểm thử



- Biểu mẫu báo cáo vấn đề
 - Vị trí: vấn đề đã xuất hiện ở đâu?
 - Thời gian: nó đã xuất hiện khi nào?
 - Dấu hiệu: cái được quan sát?
 - Kết quả cuối cùng: các hệ quả?
 - Kỹ thuật: nó đã xuất hiện như thế nào?
 - Nguyên nhân: tại sao nó xuất hiện?
 - Tính khốc liệt: mức độ mà người dùng hay doanh nghiệp bị ảnh hưởng?
 - Chi phí: nó tốn bao nhiêu?

95

TIẾN TRÌNH PHẦN MỀM

PHẦN II.5 – TRIỂN KHAI & BẢO TRÌ



96

Nội dung

- Triển khai hệ thống
- Bảo trì



97

Triển khai hệ thống

- Triển khai hệ thống:
 - Không chỉ đặt hệ thống vào vị trí.
 - Còn giúp người dùng hiểu và cảm thấy thoải mái với hệ thống.
 - Huấn luyện (phân loại đối tượng, loại huấn luyện, phương tiện trợ giúp, nguyên tắc huấn luyện)
 - Tài liệu (phân loại độc giả, loại tài liệu)



98

Huấn luyện



- Phân loại người sử dụng hệ thống
 - Người dùng: sử dụng các chức năng của hệ thống chính.
 - Điều hành viên: thực hiện các chức năng bổ sung.
 - Tạo ra các bản sao dự phòng của các tập tin dữ liệu.
 - Xác định những người truy xuất vào hệ thống.

99

Huấn luyện



- Các chức năng của người dùng và điều hành viên

Chức năng của người dùng	Chức năng của điều hành viên
Thao tác trên các tập tin dữ liệu	Chấp nhận truy xuất của người dùng
Mô phỏng các hoạt động	Chấp nhận truy xuất tập tin
Phân tích dữ liệu	Thực hiện sao lưu
Giao tiếp dữ liệu	Cài đặt các thiết bị mới
Vẽ đồ thị và biểu đồ	Cài đặt phần mềm mới
	Phục hồi các tập tin hư

100

Huấn luyện



- Các kiểu huấn luyện
 - Huấn luyện người dùng
 - Huấn luyện điều hành viên
 - Yêu cầu huấn luyện đặc biệt

101

Huấn luyện



- Huấn luyện người dùng
 - Giới thiệu các chức năng cơ bản
 - Quản lý hồ sơ: tạo, xóa, truy lục, sắp xếp hồ sơ.
 - Sự điều hướng qua hệ thống.
 - Không cần kỹ thuật bên trong (giải thuật, cấu trúc dữ liệu, ...).
 - Liên hệ cách các chức năng được thực hiện hiện tại, cách thực hiện sau đó bằng hệ thống mới.
 - Cần chú ý đến tính khó của học hỏi chuyển giao.

102

Huấn luyện



- Huấn luyện điều hành viên
 - Tập trung vào việc huấn luyện điều hành viên quen với các chức năng hỗ trợ của hệ thống và xác định cách thức hệ thống làm việc (hơn là cái mà hệ thống làm).
 - Thực hiện huấn luyện theo hai mức
 - Cách đưa ra và thực thi hệ thống mới.
 - Cách hỗ trợ người dùng.

103

Huấn luyện



- Yêu cầu huấn luyện đặc biệt
 - Người sử dụng thường xuyên và không thường xuyên.
 - Người sử dụng mới.

104

Huấn luyện



- Phương tiện trợ giúp huấn luyện
 - Tài liệu
 - Tài liệu hình thức, sách hướng dẫn.
 - Một tỷ lệ nhỏ người dùng đọc chúng.
 - Trợ giúp trực tuyến và biểu tượng
 - Biểu tượng cho đối tượng và chức năng.
 - Sách hướng dẫn trực tuyến cung cấp các liên kết siêu văn bản.
 - Các thể hiện và các lớp
 - Tổ chức lớp học; Sử dụng thiết bị đa phương tiện (nghe, nhìn).
 - Người dùng thành thạo (và người được huấn luyện)

105

Huấn luyện



- Nguyên tắc huấn luyện
 - Hiểu sở thích cá nhân, kiểu làm việc và áp lực tổ chức.
 - Cần xem xét các kiểu học viên khác nhau.
 - Hệ thống được đặc thù hóa
 - Chia nội dung huấn luyện thành các bài ở dạng ngắn.
 - Xác định kiểu huấn luyện dựa trên vị trí của học viên.

106

Tài liệu



- Hiểu thính giả
 - Phân nhóm thính giả
 - Người dùng
 - Điều hành viên
 - Nhân viên của khách hàng
 - Các thành viên khác của nhóm phát triển
 - Thiết kế tài liệu khác nhau cho các nhóm thính giả khác nhau.

107

Tài liệu



- Các loại tài liệu
 - Sách hướng dẫn người dùng
 - Sách hướng dẫn điều hành viên
 - Hướng dẫn hệ thống chung
 - Hướng dẫn học
 - Tài liệu khác: hướng dẫn lập trình viên

108

Tài liệu



- Sách hướng dẫn người dùng
 - Bắt đầu bằng mục tiêu chung, tiến tới mô tả chức năng chi tiết.
 - Mục tiêu và mục đích của hệ thống.
 - Các chức năng và khả năng của hệ thống.
 - Các đặc điểm, đặc trưng và thuận lợi của hệ thống.

109

Tài liệu



- Sách hướng dẫn điều hành viên
 - Cấu hình phần cứng, phần mềm.
 - Các phương pháp chấp nhận và từ chối truy xuất đối với người dùng.
 - Các thủ tục thêm để bổ sung và loại bỏ các ngoại vi ra khỏi hệ thống.
 - Các kỹ thuật để sao chép và dự phòng các tập tin và tài liệu.

110

Tài liệu



- Hướng dẫn hệ thống chung
 - Hệ thống chi tiết hóa theo cách mà khách hàng có thể hiểu.
 - Cấu hình phần cứng và phần mềm của hệ thống.

111

Tài liệu



- Hướng dẫn học
 - Hướng dẫn học được tự động hóa, từng bước một bằng đa phương tiện.

112

Tài liệu



- Hướng dẫn lập trình viên
 - Khái quát về cách thức phần cứng, phần mềm được cấu hình.
 - Các thành phần của phần mềm được chi tiết hóa và các chức năng của chúng được thực hiện.
 - Các chức năng hỗ trợ hệ thống.
 - Các cải tiến của hệ thống.

113

Tài liệu



- Gỡ rối và giúp đỡ người dùng
 - Các chỉ dẫn khi hệ thống không thực hiện.
 - Các tài liệu tham khảo.
 - Các tập tin giúp đỡ trực tuyến.

114

Bảo trì



- Định nghĩa
- Phân loại bảo trì
- Framework của bảo trì phần mềm (BTPM)
- Các vấn đề then chốt trong BTPM
- Quy trình BTPM
- Các kỹ thuật và công cụ trong BTPM

115

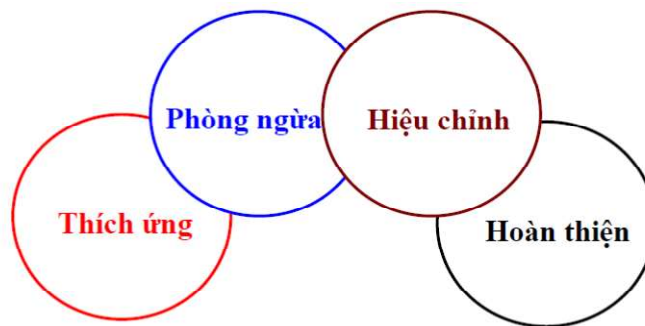
Định nghĩa



- Bảo trì phần mềm (Software Maintenance)
 - [IEEE 1219] Bảo trì phần mềm là sự sửa đổi một sản phẩm phần mềm sau khi phát hành nhằm hiệu chỉnh lỗi, cải thiện sự thực thi hay các đặc tính khác, hay làm cho sản phẩm thích ứng với môi trường bị thay đổi.
 - [ISO/IEC/IEEE 14764] Bảo trì phần mềm là sản phẩm phần mềm phải trải qua sự sửa đổi về mã lệnh và các tài liệu liên quan do có vấn đề hay nhu cầu cải tiến. Mục đích là sửa đổi sản phẩm phần mềm hiện có mà vẫn giữ được tính toàn vẹn của nó.

116

Các loại bảo trì phần mềm

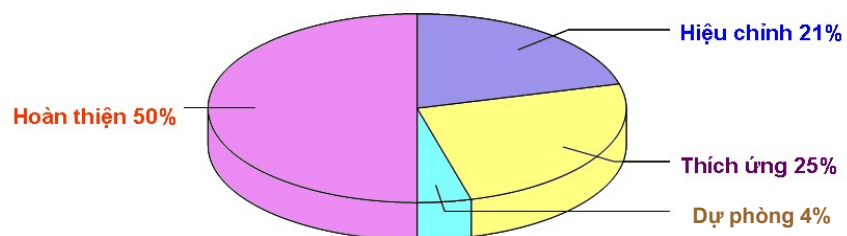


117

Các loại bảo trì phần mềm

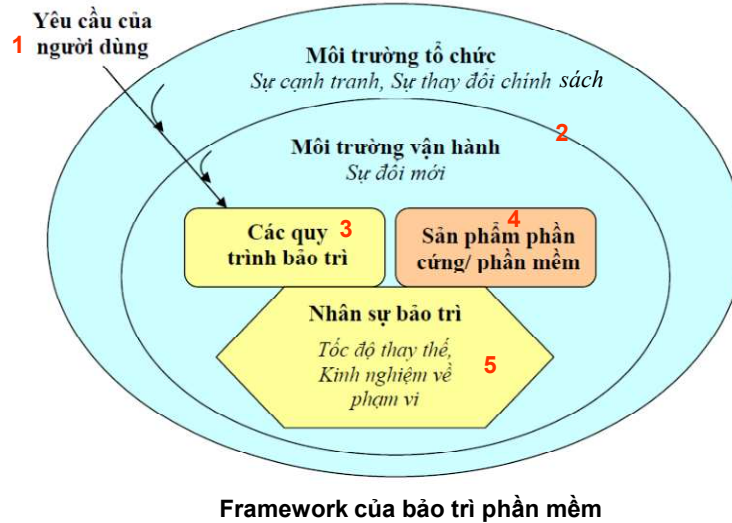


- Công sức bảo trì



118

Framework của bảo trì phần mềm



119

Các vấn đề then chốt trong BTPM

- Các vấn đề về kỹ thuật: Sự hiểu biết có giới hạn, Kiểm thử, Phân tích sự tác động, Tính có thể bảo trì.
- Các vấn đề trong quản lý: Bố trí nhân sự, Các vấn đề về quy trình, Chọn nhóm bảo trì, Gia công bảo trì phần mềm.
- Dự đoán chi phí phần mềm
- Phép đo bảo trì phần mềm

120

Quy trình bảo trì



- *Quy trình bảo trì phần mềm* (Software Maintenance Process): Các chuỗi các hoạt động được thực hiện để đem lại sự thay đổi trong suốt sự bảo trì.
- Những quy trình bảo trì phổ biến được mô tả trong các chuẩn
 - IEEE 1219
 - ISO/IEC/IEEE 14764

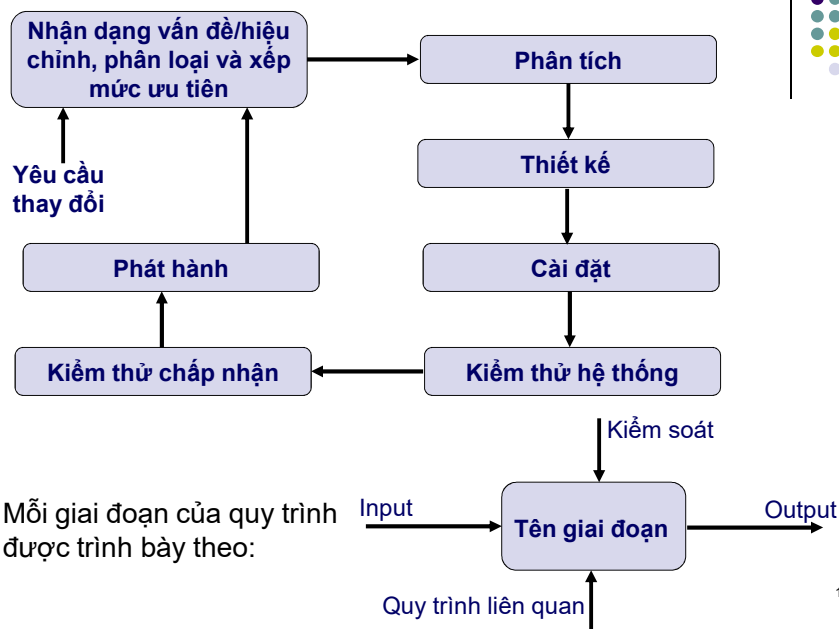
121

Phân đọc thêm



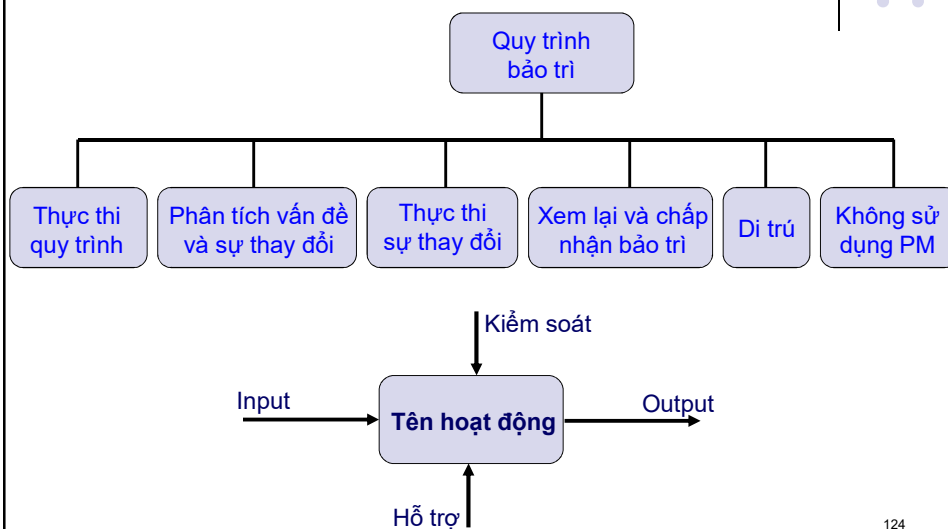
122

Quy trình BT theo chuẩn IEEE 1219



123

Quy trình BT theo chuẩn ISO/IEC 14764



124

Kỹ thuật và công cụ bảo trì



- Sự hiểu biết về chương trình
- Kỹ thuật đảo ngược (Reverse Engineering)
- Kỹ thuật tái kiến tạo (ReEngineering)
- Các công cụ bảo trì

125

Sự hiểu biết về chương trình



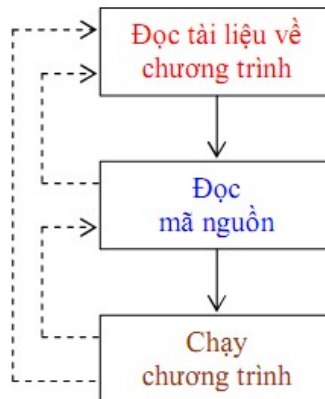
- Mục đích cuối cùng của việc đọc và hiểu chương trình là có thể thực hiện thành công các thay đổi được yêu cầu.
- Điều này đòi hỏi phải có kiến thức về **những điểm đặc trưng của một hệ thống phần mềm**:
 - Phạm vi vấn đề
 - Ảnh hưởng của sự thực hiện
 - Quan hệ nhân - quả
 - Quan hệ sản phẩm - môi trường
 - Các điểm đặc trưng hỗ trợ việc ra quyết định

126

Sự hiểu biết về chương trình



- Các hoạt động liên quan đến việc hiểu chương trình



127

Định nghĩa



- **Kỹ thuật đảo ngược** là quy trình phân tích một hệ thống chủ thể để:
 - Nhận dạng các thành phần của hệ thống và các mối quan hệ giữa chúng
 - Tạo ra các biểu diễn của hệ thống ở một dạng khác hay ở các mức trừu tượng cao hơn

128

Kỹ thuật đảo ngược



- **Kỹ thuật đảo ngược** là quy trình phân tích một hệ thống chủ thể để:
 - Nhận dạng các thành phần của hệ thống và các mối quan hệ giữa chúng
 - Tạo ra các biểu diễn của hệ thống ở một dạng khác hay ở các mức trừu tượng cao hơn

129

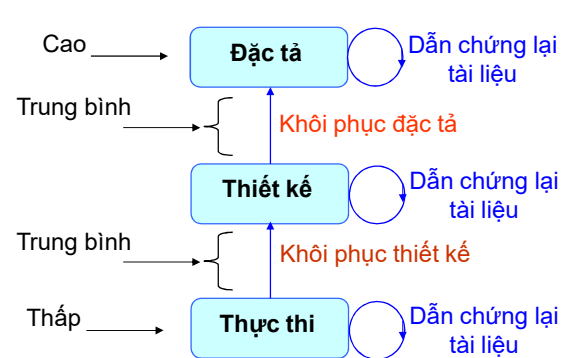
Kỹ thuật đảo ngược



Các mức của kỹ thuật đảo ngược

Mức trừu tượng

Các giai đoạn



Kỹ thuật
đảo ngược

130

Kỹ thuật tái kiến tạo



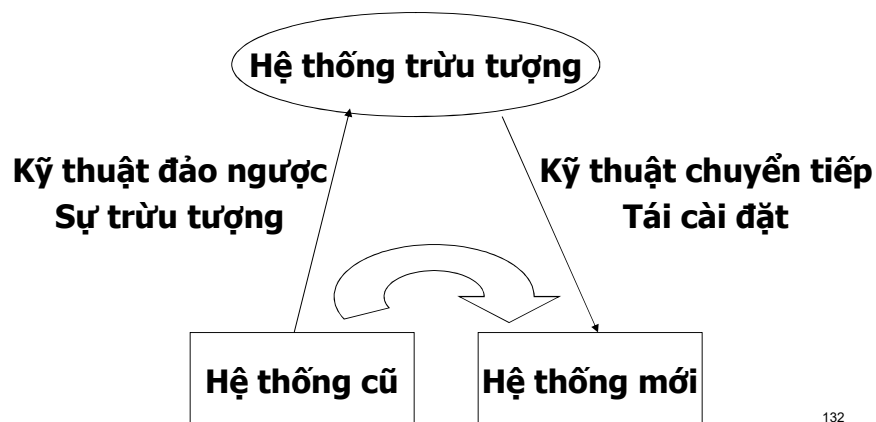
- Kỹ thuật tái kiến tạo
 - Là quy trình kiểm tra và sửa đổi hệ thống đích để thực thi các thay đổi mong muốn.
 - Gồm hai bước:
 - Kỹ thuật đảo ngược
 - Kỹ thuật chuyển tiếp

131

Kỹ thuật tái kiến tạo



- Kỹ thuật tái kiến tạo



132

Các loại công cụ bảo trì



- Về nguyên tắc, ta có thể phân loại công cụ bảo trì phần mềm dựa trên công việc mà chúng hỗ trợ.
- Các công việc giúp phân loại công cụ bảo trì:
 - Hiểu chương trình, kỹ thuật đảo ngược
 - Kiểm thử
 - Quản lý cấu hình
 - Lập tài liệu và đo lường
- Trong thực tế, ta khó có được phân loại tốt vì trạng thái tự nhiên đa dạng hóa và có liên quan với nhau của các hoạt động bảo trì phần mềm

133