

# ***Kỹ nghệ phần mềm***

## ***Software Engeneering***

*Đại học Kinh doanh và Công nghệ Hà Nội*  
*Khoa CNTT*

*GV: Đào Thị Phương*

*Email: [phuongdt102@gmail.com](mailto:phuongdt102@gmail.com)*

*Page fb: [facebook.com/it.hubt](https://facebook.com/it.hubt)*

*Phone: 0946.866.817*

# Bài 9: Xác minh & thẩm định



## Nội dung

- Khái niệm xác minh, thẩm định
- Rà soát phần mềm
- Kiểm thử phần mềm

# TÀI LIỆU THAM KHẢO



1. Nguyễn Văn Vy, Nguyễn Việt Hà. *Giáo trình kỹ nghệ phần mềm*. Nhà xuất bản Đại học Quốc gia Hà nội, 2008
2. Grady Booch, James Rumbaugh, Ivar Jacobson. *The Unified Modeling language User Guid*. Addison-Wesley, 1998.
3. M. Ould. *Managing Software Quality and Business Risk*, John Wiley and Sons, 1999.
4. Roger S.Pressman, *Software Engineering, a Practitioner's Approach*. Fifth Edition, McGraw Hill, 2001.
5. Ian Sommerville, *Software Engineering*. Sixth Edition, Addison-Wasley, 2001.
6. Nguyễn Văn Vy. *Phân tích thiết kế hệ thống thông tin hiện đại. Hướng cấu trúc và hướng đối tượng*, NXB Thống kê, 2002, Hà Nội.

# Khái niệm xác minh & thẩm định



- Xác minh (*Verification*)

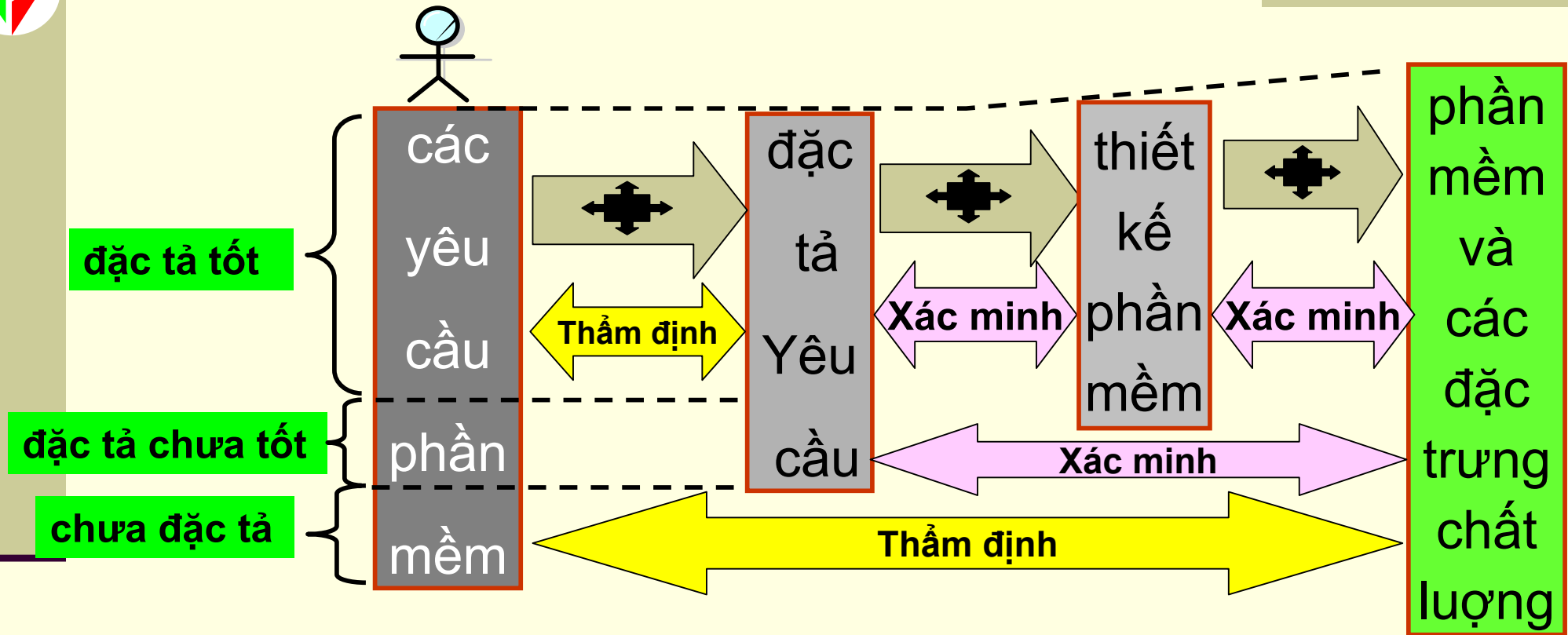
Kiểm tra xem phần mềm làm ra có đúng đặc tả (yêu cầu, thiết kế) hay không

- Thẩm định (*Validation*)

kiểm tra xem phần mềm có đáp ứng yêu cầu người dùng không

❖ Đây là 2 hoạt động cốt yếu để đảm bảo chất lượng phần mềm, diễn ra suốt quá trình phát triển

# Hoạt động kiểm chứng phần mềm

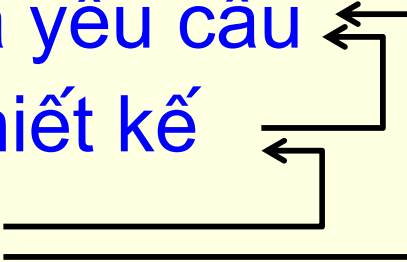


**Thẩm định và xác minh thực hiện ở mọi giai đoạn phát triển, với sản phẩm khác nhau, do đối tượng khác nhau thực hiện**

# Các hoạt động xác minh



## ■ Cơ sở cho hoạt động xác minh

- ◆ Bản đặc tả yêu cầu
  - ◆ Các bản thiết kế
  - ◆ Mã nguồn
- 

## ■ Hoạt động xác minh

- ◆ Rà soát (thanh tra, xét duyệt, kiểm toán)
- ◆ Kiểm thử (đơn vị, tích hợp, hệ thống)

# Các hoạt động thẩm định



- Cơ sở cho hoạt động xác minh
  - ◆ Bản đặc tả yêu cầu
  - ◆ Mã nguồn
- Hoạt động xác minh
  - ◆ Rà soát (thanh tra, xét duyệt)
  - ◆ Kiểm toán
  - ◆ Kiểm thử thẩm định(chấp nhận)
- ❖ Hai hoạt động chính của thẩm định và xác minh là: **rà soát** và **kiểm thử**.

# Thẩm định/xác minh tính



- ◆ rà soát, xét duyệt các tài liệu phần mềm: *kế hoạch, yêu cầu, thiết kế, mã nguồn*
- ◆ phát hiện một số loại lỗi nhất định
- ◆ khó đánh giá tính hiệu quả của sản phẩm



# Thẩm định/xác minh động



- thực hiện trên cơ sở cho vận hành sản phẩm phần mềm:
  - ◆ Làm mẫu yêu cầu
  - ◆ Vận hành chương trình (kiểm thử)
  - ◆ Mô phỏng hệ thống
- người phát triển/người dùng trực tiếp kiểm tra đánh giá
- phát hiện mọi lỗi và khiếm khuyết phần mềm, hiệu quả cao

# Rà soát phần mềm



- **Rà soát** là xem xét, đánh giá sản phẩm được tiến hành mỗi giai đoạn để phát hiện ra những khiếm khuyết cần sửa trước khi sang giai đoạn sau
- **Mục tiêu:**
  - Chỉ ra các khiếm khuyết cần phải cải thiện.
  - Khẳng định những sản phẩm đạt yêu cầu.
  - Kiểm soát việc đạt chất lượng kỹ thuật tối thiểu của sản phẩm (có diện mạo không đổi, ổn định)
- **Áp dụng** tại các thời điểm khác nhau trong quá trình phát triển phần mềm.

# Các hình thức rà soát



- Các kiểu rà soát:
  - ◆ Thanh tra
  - ◆ Hợp xét duyệt không chính thức,
  - ◆ Hợp chính thức với các thành viên: khách hàng, nhà quản lý, nhân viên kỹ thuật. (rà soát kỹ thuật chính thức – formal technical review: FTR)
- FTR chủ yếu do các kỹ sư phần mềm thực hiện (là một phương tiện hiệu quả để cải thiện chất lượng)

# Rà soát kỹ thuật chính thức

*(formal technical review - FTR)*



- **Rà soát kỹ thuật chính thức** là hoạt động bảo đảm chất lượng phần mềm do những người đang tham gia phát triển thực hiện.
- Mục tiêu cụ thể là:
  - ◆ Phát hiện các lỗi trong chức năng, logic (chương trình) và triển khai (implementation).
  - ◆ Kiểm thử sự phù hợp của phần mềm với yêu cầu
  - ◆ Khẳng định phần đã đạt yêu cầu

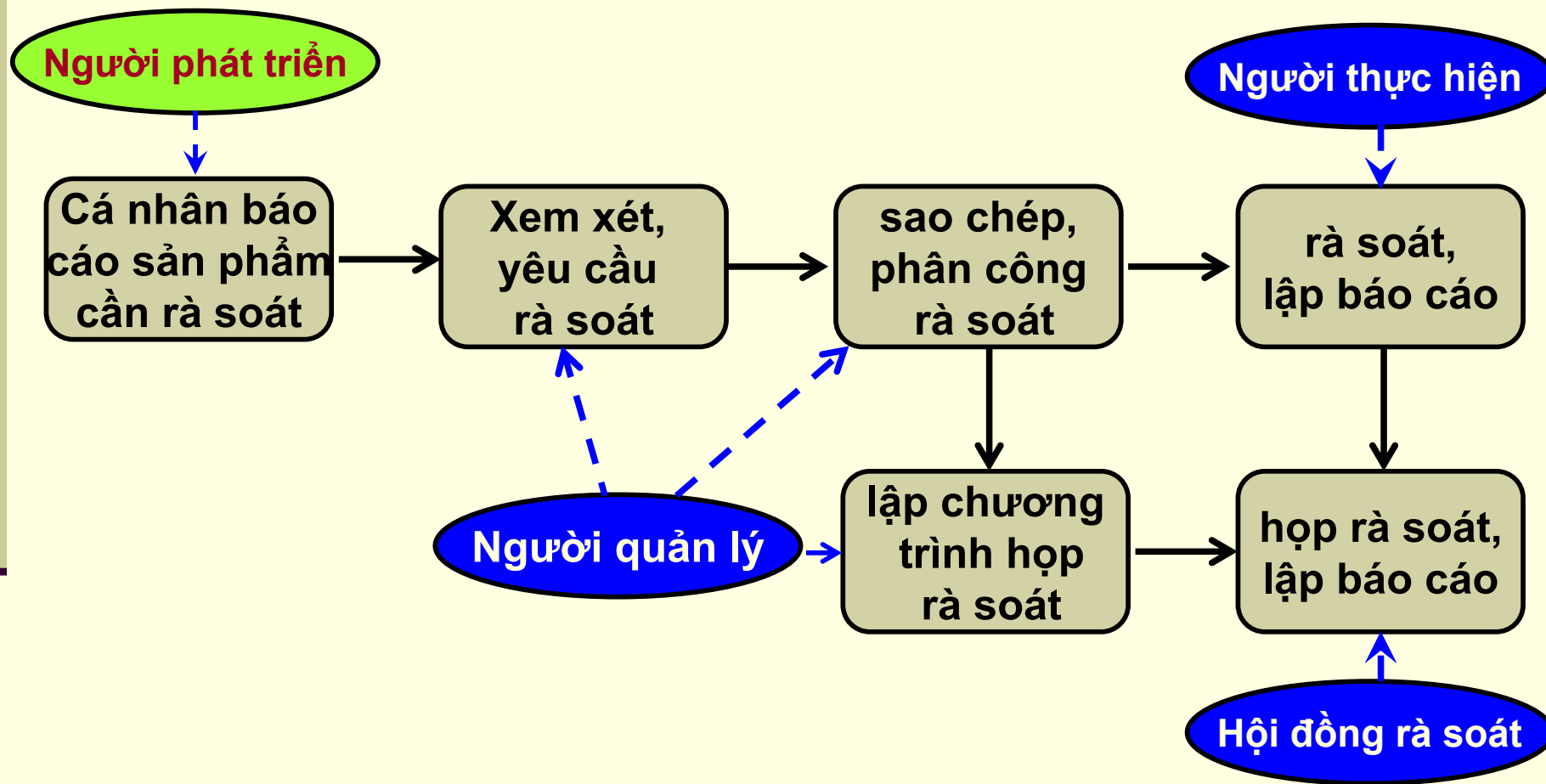
# Mục tiêu rà soát kỹ thuật chính thức



## ■ Mục tiêu cụ thể (t):

- ◆ Bảo đảm FM phù hợp với các chuẩn đã định
- ◆ Đảm bảo FM được phát triển theo một cách thức nhất quán (uniform manner)
- ◆ Làm cho dự án dễ quản lý hơn
- ◆ Ngoài ra, làm cơ sở huấn luyện các kỹ sư trẻ và có ích ngay cả cho những kỹ sư đã có kinh nghiệm

# Tiến trình hoạt động rà soát



# Cuộc họp rà soát



- Thành phần: lãnh đạo rà soát, các cá nhân rà soát và người tạo ra sản phẩm được rà soát (+ khách).
- Kết luận đưa ra 1 trong 3 quyết định sau:
  - Chấp nhận sản phẩm không cần chỉnh sửa
  - Khước từ sản phẩm vì những lỗi nghiêm trọng
  - Chấp nhận cho chỉnh sửa sản phẩm, sau khi chỉnh sửa phải rà soát lại
- Mọi thành viên tham gia cuộc họp phải ký vào quyết định

# Sản phẩm rà soát



- **Sản phẩm** cuộc họp rà soát:
  - 1 Báo cáo các vấn đề nảy sinh do cá nhân rà soát nêu ra
  - 1 danh sách các vấn đề cần giải quyết
  - 1 bản tổng kết cuộc họp
- **Bản tổng kết** họp rà soát phải chỉ rõ:
  - Đã rà soát cái gì
  - Ai rà soát
  - Tìm thấy cái gì và Kết luận ra sao



# Sản phẩm rà soát (t)



## ■ Danh sách các vấn đề tồn tại phục vụ:

- ◆ Nhận ra các vùng có vấn đề trong sản phẩm được rà soát
- ◆ Dùng như 1 danh sách các khoản mục để chỉ cho các người làm ra sản phẩm cần chỉnh sửa
- ◆ Thiết lập thủ tục để bảo đảm rằng các khoản mục trong danh sách đó sẽ được chỉnh sửa thực sự

# Tiến hành rà soát



- Mọi sản phẩm được tạo ra ở mỗi bước đều được rà soát (không chỉ sản phẩm cuối cùng)
- Tiến trình phát triển chung nhất gồm 4 -5 giai đoạn:
  - ◆ Kỹ nghệ hệ thống (*KH triển khai*)
  - ◆ Phân tích, xác định yêu cầu phần mềm (*đặc tả yêu cầu*)
  - ◆ Thiết kế phần mềm (*thiết kế*)
  - ◆ Lập mã (*mã nguồn*)
  - ◆ Kiểm thử phần mềm (*kế hoạch kiểm thử*)
  - ◆ Bảo trì (*kế hoạch bảo trì*)

Rà soát bám sát theo sản phẩm của các giai đoạn này

# Các danh mục sản phẩm cần rà soát



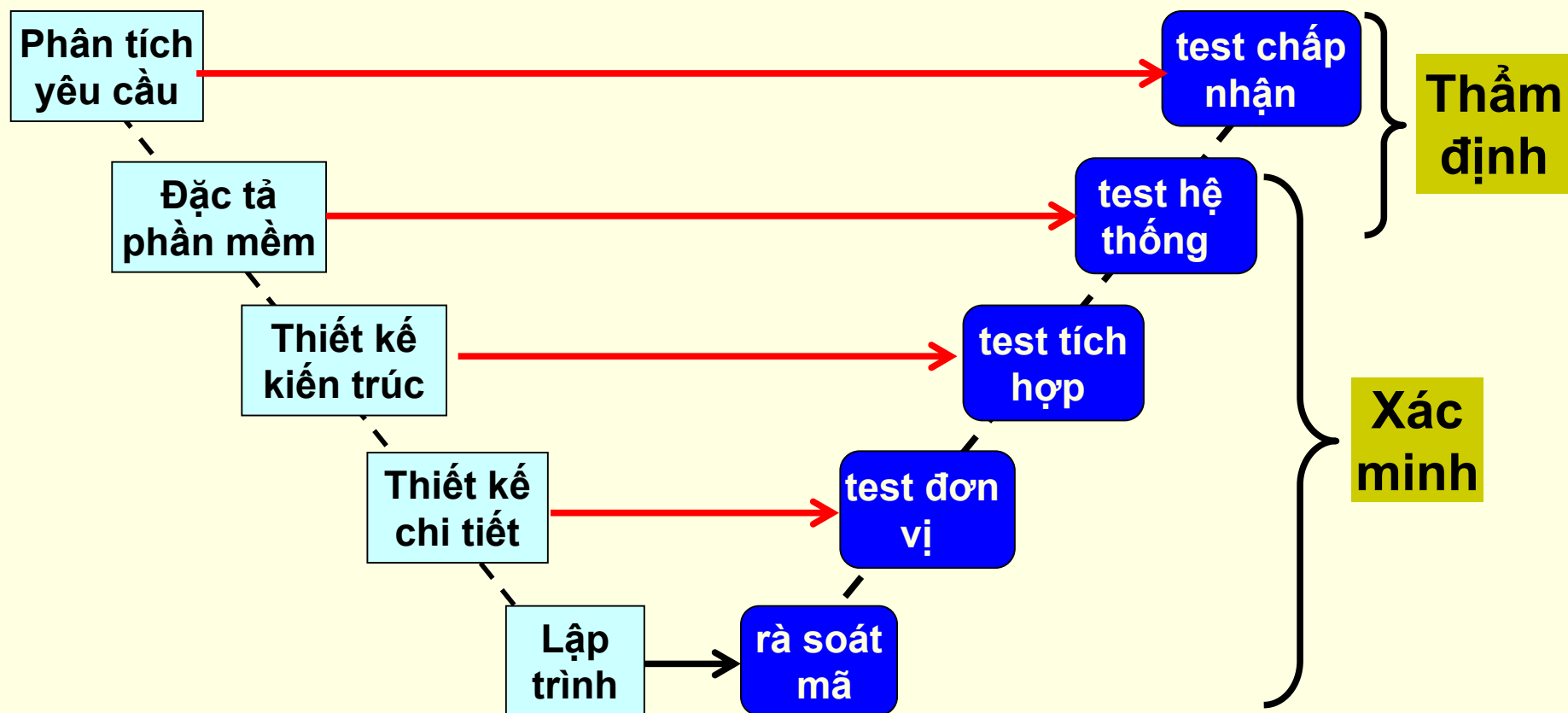
- Danh mục rà soát **kỹ nghệ hệ thống**
- Danh mục rà soát **lập kế hoạch dự án**
- Danh mục rà soát **phân tích yêu cầu phần mềm**
- Danh mục rà soát **thiết kế phần mềm**
- Danh mục rà soát **khâu lập mã**
- Danh mục rà soát **kiểm thử phần mềm**
- Danh mục rà soát **bảo trì phần mềm**

# Kiểm thử phần mềm – software testing



- ***Kiểm thử là tổ chức vận hành phần mềm 1 cách có kế hoạch và phương pháp để tìm ra lỗi***
- Cần vận hành như thế nào để:
  - ◆ hiệu suất tìm ra lỗi là cao nhất ?
  - ◆ chi phí (*thời gian, công sức*) ít nhất?
- Nội dung hoạt động kiểm thử bao gồm:
  - ◆ Kế hoạch kiểm thử
  - ◆ phương pháp kiểm thử,
  - ◆ chiến lược kiểm thử và
  - ◆ kỹ thuật sử dụng

# Mô hình chữ V - Các mức kiểm thử



# Các loại kiểm thử



Tương ứng với mô hình chữ V có các loại kiểm thử:

- ◆ Kiểm thử đơn vị (*unit testing*)
- ◆ Kiểm thử tích hợp (*integration testing*)
- ◆ Kiểm thử hệ thống (*system testing*)
  - Kiểm thử phục hồi (*recovery testing*)
  - Kiểm thử áp lực (*stress testing*)
  - Kiểm thử thi hành (*performance testing*)
  - Kiểm thử an ninh (*security testing*)
- ◆ Kiểm thử thẩm định/chấp nhận (*acceptance testing: alpha testing, beta testing*)

# Kế hoạch kiểm thử



- Kế hoạch kiểm thử tổng thể:
  1. Giới thiệu chung
    - Mô tả hệ thống cần kiểm thử
    - Các mục tiêu kiểm thử
    - Phương pháp sử dụng
    - Tài liệu hỗ trợ
  2. Kế hoạch
    - Thời gian, địa điểm
    - Tài liệu kiểm thử: các ca kiểm thử, tiến trình, lịch trình
    - Điều kiện
  3. Các yêu cầu: phần cứng, phần mềm, nhân sự
  4. Kiểm soát quá trình kiểm thử

# Phương pháp và chiến lược kiểm thử

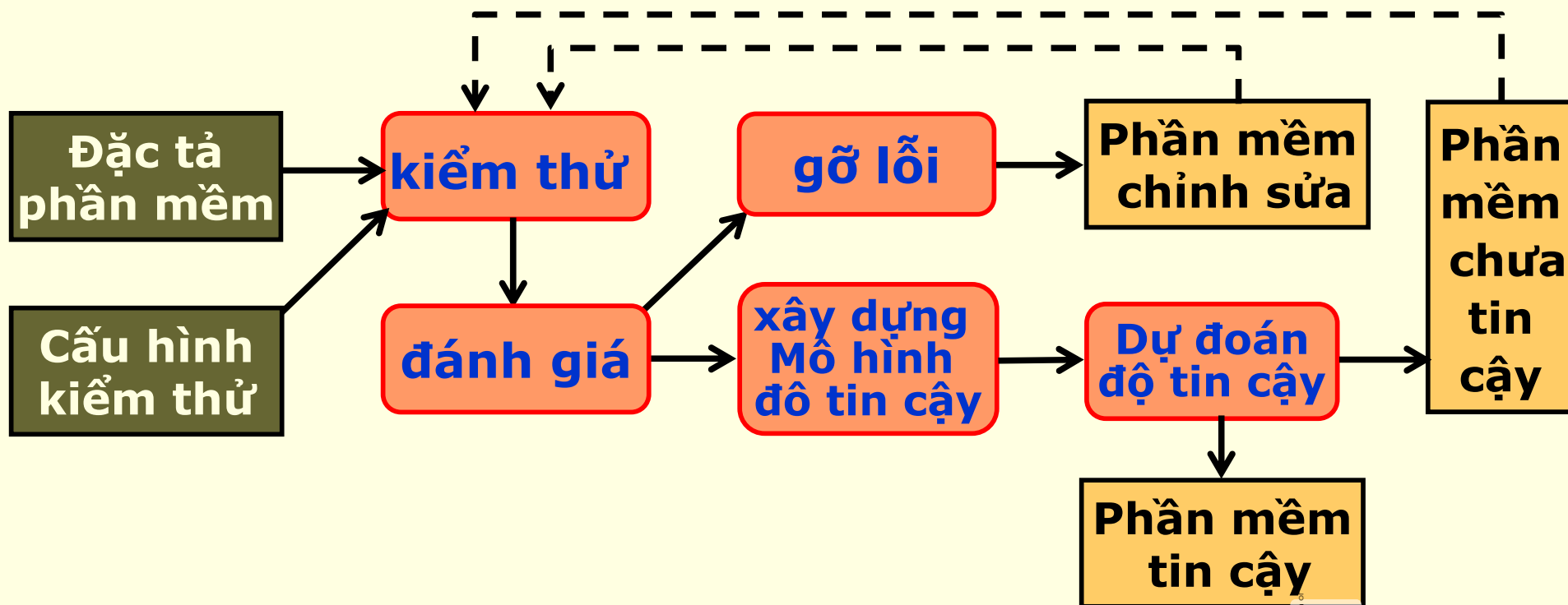


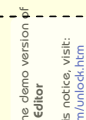
- Hai phương pháp phổ biến:
  - ◆ Kiểm thử hộp trắng (**white box**)
  - ◆ Kiểm thử hộp đen (**black box**)
- Các chiến lược Kiểm thử
  - ◆ Ứng dụng cho **các mức** & **loại kiểm thử** khác nhau.
  - ◆ Một số chiến lược:
    - Chiến lược nhánh & toán tử quan hệ: BRO(đơn vị)
    - Kiểm thử từ trên xuống/dưới lên/lai (tích hợp)
    - Kiểm thử vụ nổ lớn (big bang – tích hợp)
    - Kiểm thử hồi quy (tích hợp)
    - Kiểm thử luồng sợi (hệ thời gian thực)



# Biểu đồ dòng thông tin kiểm thử

- Sơ đồ **dòng thông tin** của tiến trình kiểm thử





# Khái niệm về thiết kế ca kiểm thử



Mục tiêu thiết kế ca kiểm thử nhằm:

- ◆ tìm ra nhiều sai nhất
- ◆ với nỗ lực & thời gian nhỏ nhất.

Các phương pháp tốt phải cho một cơ chế:

- ◆ bảo đảm tính đầy đủ (không sót phần nào) và
- ◆ cung cấp khả năng thật sự phát hiện được các sai
- ◆ Ca kiểm thử hiệu quả là ca kiểm thử phát hiện ra ít nhất 1 lỗi

# Kiểm thử hộp trắng



## ■ Khái niệm kiểm thử hộp trắng

- Đối tượng: mã nguồn
- Mức: các mô đun đơn vị
- Nội dung là khám xét:
  - ◆ các chi tiết thủ tục (thuật toán)
  - ◆ con đường logic (luồng điều khiển)
  - ◆ các trạng thái của chương trình (dữ liệu).

# Yêu cầu kiểm thử hộp trắng



## ■ Yêu cầu đặt ra:

- ◆ Mọi con đường độc lập trong một mô đun cần được thực hiện ít nhất một lần.
- ◆ Mọi ràng buộc logic được thực hiện cả hai phía đúng (true) & phía sai (false).
- ◆ Tất cả các vòng lặp ở biên của nó & cả các biên vận hành phải được thực hiện.
- ◆ Mọi cấu trúc dữ liệu nội tại được dùng để bảo đảm hiệu lực thi hành của nó

# Các kỹ thuật sử dụng



1. Đồ thị dòng (Tom McCabe đưa ra đầu tiên).
2. Ma trận kiểm thử (số đường đi, trọng số).
3. Điều kiện logic – chiến lược miền và BRO
4. Điều khiển theo dòng dữ liệu
5. Các cấu trúc chu trình – giá trị đặc trưng

# Ví dụ: cấu trúc điều khiển 1 chương trình

- xét biểu đồ điều khiển của một chương trình

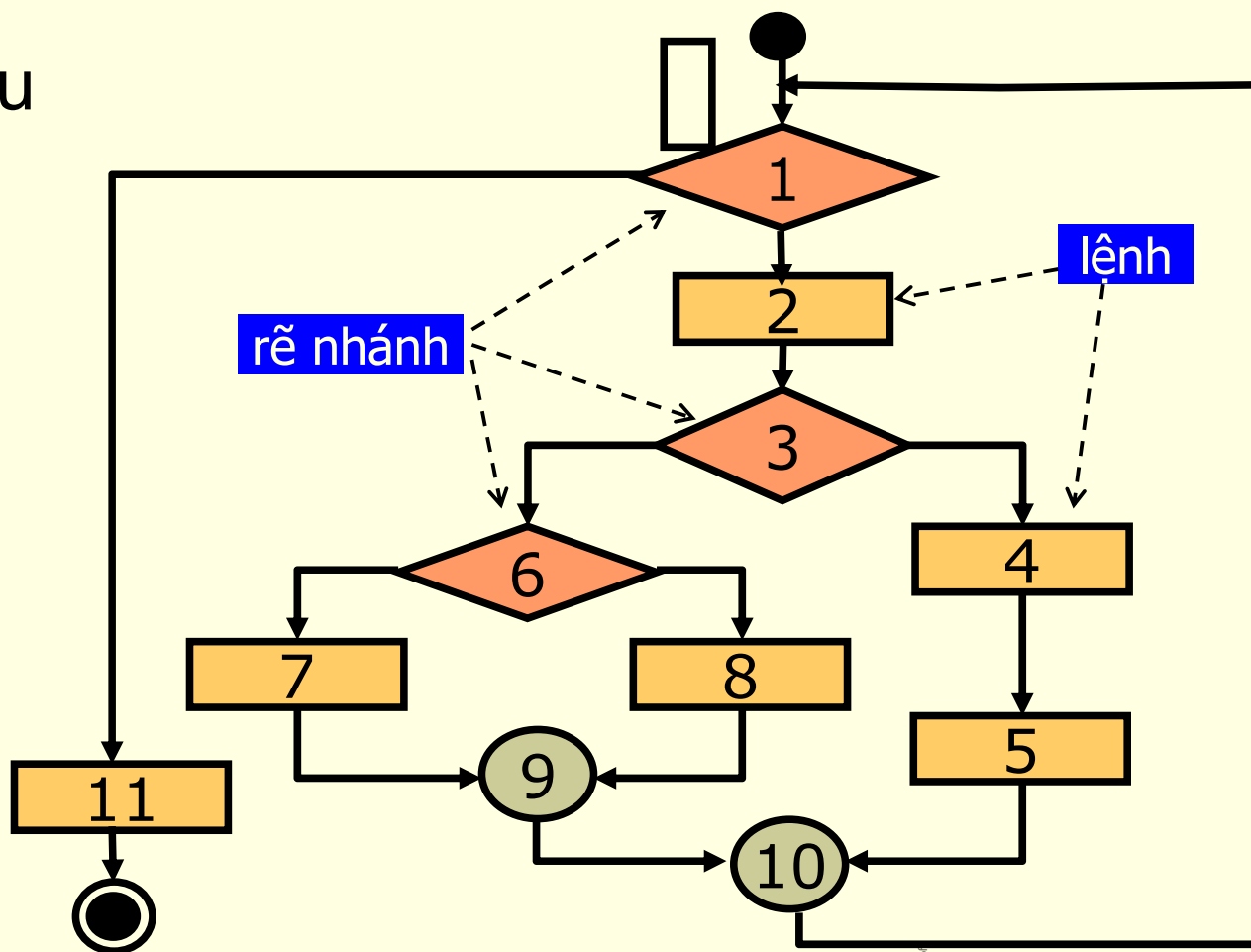
4 đường đi độc lập:

1-11

1-2-3-4-5-10

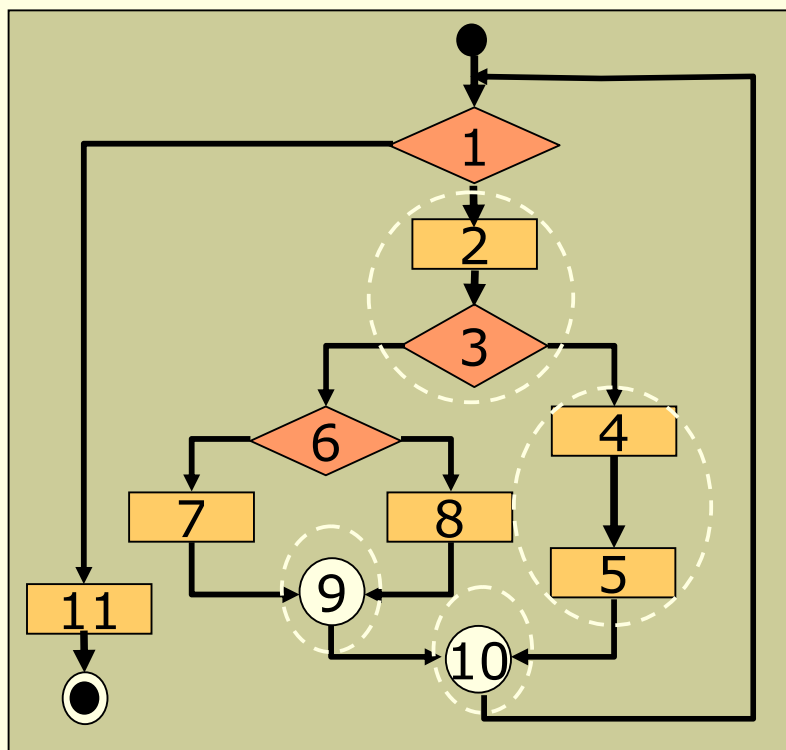
1-2-3-6-7-9-10

1-2-3-6-8-9-10

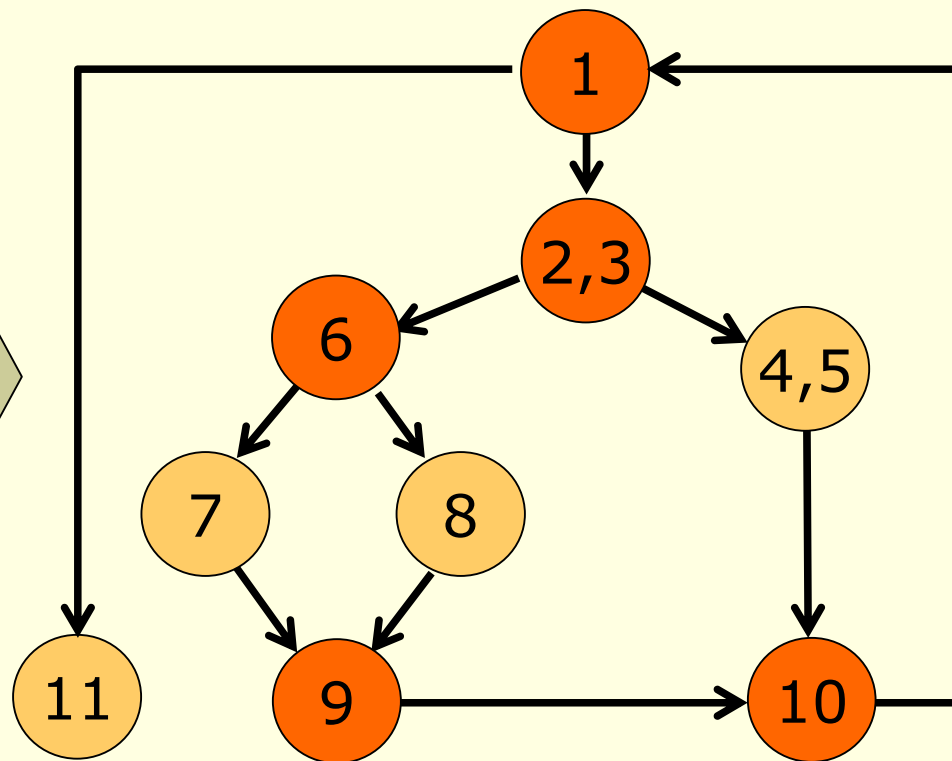


# Đồ thị dòng của chương trình

## ■ luồng điều khiển



## ■ đồ thị dòng





# Độ phức tạp của chu trình



- Để mọi lệnh đều được kiểm thử ít nhất một lần, cần tìm được tất cả các đường điều khiển độc lập trong chương trình (khác với các đường khác ít nhất một lệnh).
- Số các đường độc lập của 1 chương trình là giới hạn trên số các kiểm thử cần phải thực hiện. Nó được gọi là độ phức tạp chu trình của chương trình
- Các đường độc lập của 1 chương trình trùng với các đường độc lập của đồ thị dòng (tìm đơn giản hơn).

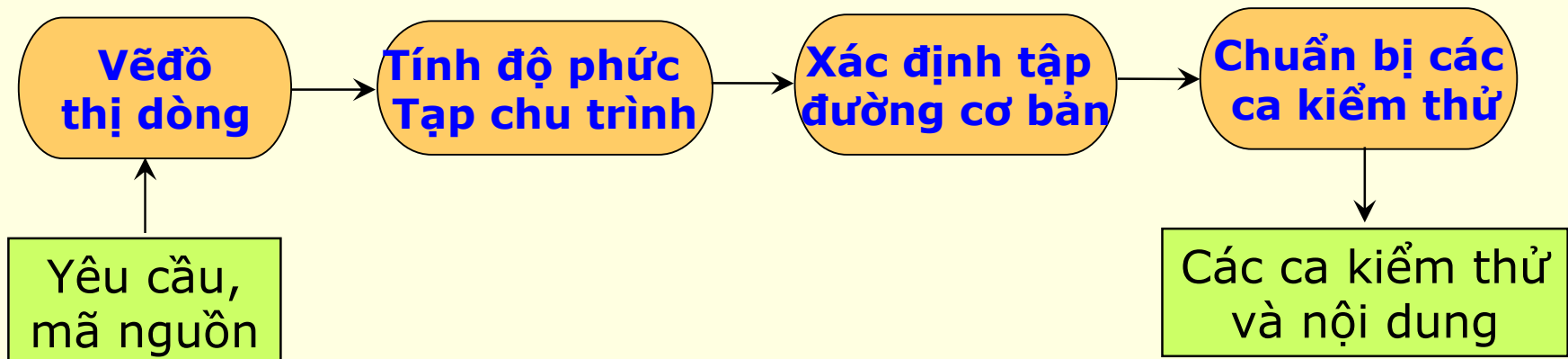
# Tính độ phức tạp chu trình từ đồ thị dòng



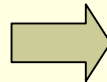
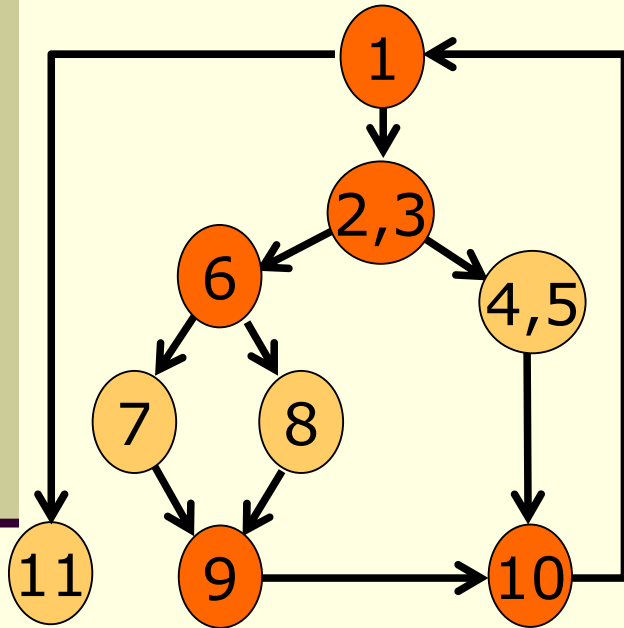
- Đồ thị dòng trên gồm: **9** nút, trong đó: **5** nút là vị tự, **11** cung, Chia mặt phẳng thành **4** miền
- Độ phức tạp chu trình  $V(G)$  của đồ thị  $G$  được tính theo các cách sau:
  - $V(G) = E - N + 2 = 11 - 9 + 2 = 4$
  - $V(G) = \text{số miền phẳng} = 4$
  - $V(G) = P - 1 = 5 - 1 = 4$

Trong đó:  $E$  = số cung;  $N$  = số nút;  $P$  = số nút vị tự

# Xác định các ca kiểm thử



# Ví dụ ma trận kiểm thử



	1	23	45	6	7	8	9	10	11
1		1							1
23			1	1					
45								1	
6					1	1			
7							1		
8							1		
9								1	
10	1								
11									

= A

# Ví dụ ma trận kiểm thử (t)

$A^2 =$

	1	23	45	6	7	8	9	10	11
1			1	1					
23					1	1			
45	1								
6							2		
7								1	
8								1	
9									
10									
11		1							1

- ♦ số trong ma trận cho biết số con đường có hai cạnh đi qua cung đó
- ♦ Ma trận này giúp lựa chọn đường để lập ca kiểm thử

# Điều kiện logic và các chiến lược



Trong chương trình, mỗi rẽ nhánh xác định bằng biểu thức logic. Điều kiện logic có thể là:

- Điều kiện đơn là 1 biến Bool (có thể có toán tử phủ định):  $X$
- Điều kiện đơn là biểu thức quan hệ giữa 2 biểu thức số học

$C = (A \Theta B)$ , với  $\Theta$  là phép so sánh:  $<, \leq, =, >, \geq$  hay  $\neq$   
 $A, B$  là biểu thức số học

- Điều kiện phức hợp cấu thành từ hơn 1 điều kiện đơn nhờ các toán tử Bool: hoặc ( $\cup$ ), và ( $\cap$ ), phủ định ( $\neg$ )

$D = X_1 \& X_2 \& \dots X_n$ , trong đó  $X_i$  là điều kiện đơn,  $\&$  là toán tử bool

# Chiến lược phân nhánh, miền và BRO



- Kiểm thử từng điều kiện logic trong chương trình.
- Kiểm thử không chỉ phát hiện sai trong điều kiện mà còn là phát hiện sai khác của chương trình liên quan.
- **Nguyên tắc kiểm thử nhánh:** với mỗi điều kiện phức hợp  $C$ , thì mỗi nhánh "true" và "false" của  $C$ , mỗi điều kiện đơn trong  $C$  phải được kiểm thử ít nhất một lần.
- **Chiến lược kiểm thử miền** cần **3 hoặc 4** kiểm thử cho một biểu thức quan hệ gồm các trường hợp:  $<$ ,  $>$ ,  $=$  và có thể  $\neq$  nữa.
- **Chiến lược BRO** là kết hợp 2 chiến lược trên
- ◆ **Làm sao chỉ ra tất cả các trường hợp cần kiểm thử?**

## Ví dụ: Chiến lược BRO – tạo ràng buộc2

- Xét điều kiện  $C$  là hội biến Bool và biểu thức quan hệ:

$$C = A \cup (B = E)$$

Khi đó các ràng buộc của  $C$  là các cặp  $(t,t)$ ,  $(t,f)$  &  $(f,t)$ ; với  $(B = E)$  có giá trị  $t$  tương ứng với "=", và giá trị  $f$  tương ứng với "<" hoặc ">"; Bởi vậy tập các đầu vào để kiểm tra  $C$  phải gồm 4 phần tử:

$(t,=)$ ,  $(t,<)$ ,  $(t,>)$  và  $(f,=)$ .

- Đầu vào phủ các ràng buộc  $C$  này bảo đảm phát hiện được mọi sai biến Bool hoặc toán tử quan hệ trong  $C$ .

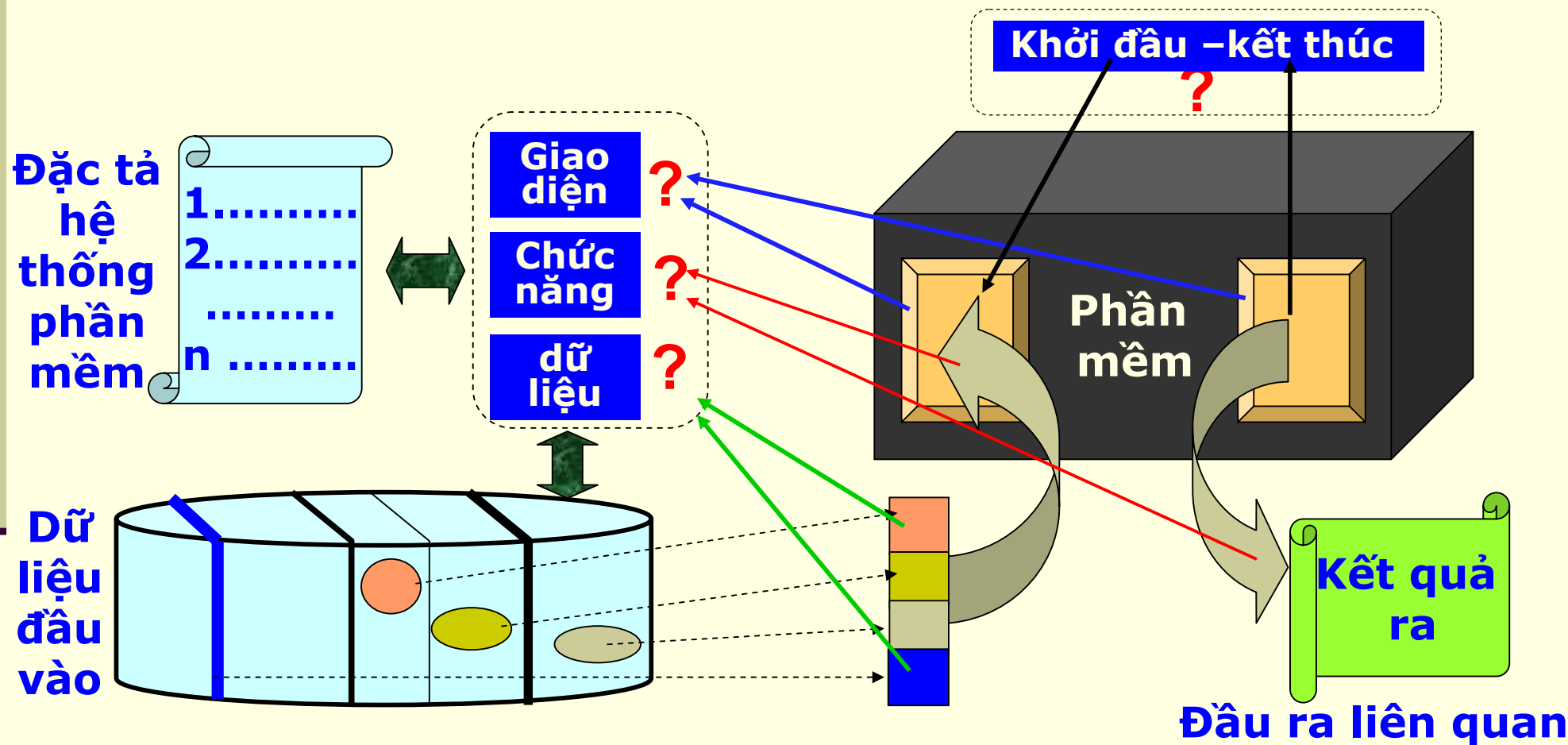


# Kiểm thử hộp đen – khái niệm



- Đối tượng kiểm thử xem như hộp đen, thông qua giao diện để đưa dữ liệu vào và nhận thông tin ra
- Là kiểm thử **yêu cầu chức năng**
- Đối tượng: **mô đun, hệ con, toàn hệ thống**
- Đặc trưng:
  - ◆ Thuyết minh: các chức năng đủ & vận hành đúng
  - ◆ Thực hiện: qua giao diện
  - ◆ Cơ sở: đặc tả, điều kiện vào/ra và cấu trúc dữ liệu
  - ◆ Ít chú ý tới cấu trúc logic nội tại của nó

# Mô hình khái niệm kiểm thử hộp đen



# Mục đích kiểm thử hộp đen



## □ Tìm các loại sai liên quan:

- ◆ **Chức năng:** đủ, đúng đắn
- ◆ **Giao diện:** vào, ra: đủ, phù hợp, đúng, tiện lợi
- ◆ **Cấu trúc, truy cập dữ liệu:** thông suốt, đúng đắn
- ◆ **Thực thi:** trôi chảy, kịp thời, chịu lỗi, phục hồi được
- ◆ **Khởi đầu - kết thúc:** mỗi tiến trình bình thường

# Các kỹ thuật kiểm thử hộp đen



## 1. Phân hoạch tương đương

- ◆ Chia tập dữ liệu thành từng lớp tương đương
- ◆ Mỗi lớp hoặc là đúng hay sai, chỉ cần kiểm tra 1 số giá trị đặc trưng của nó → rút được số ca kiểm thử

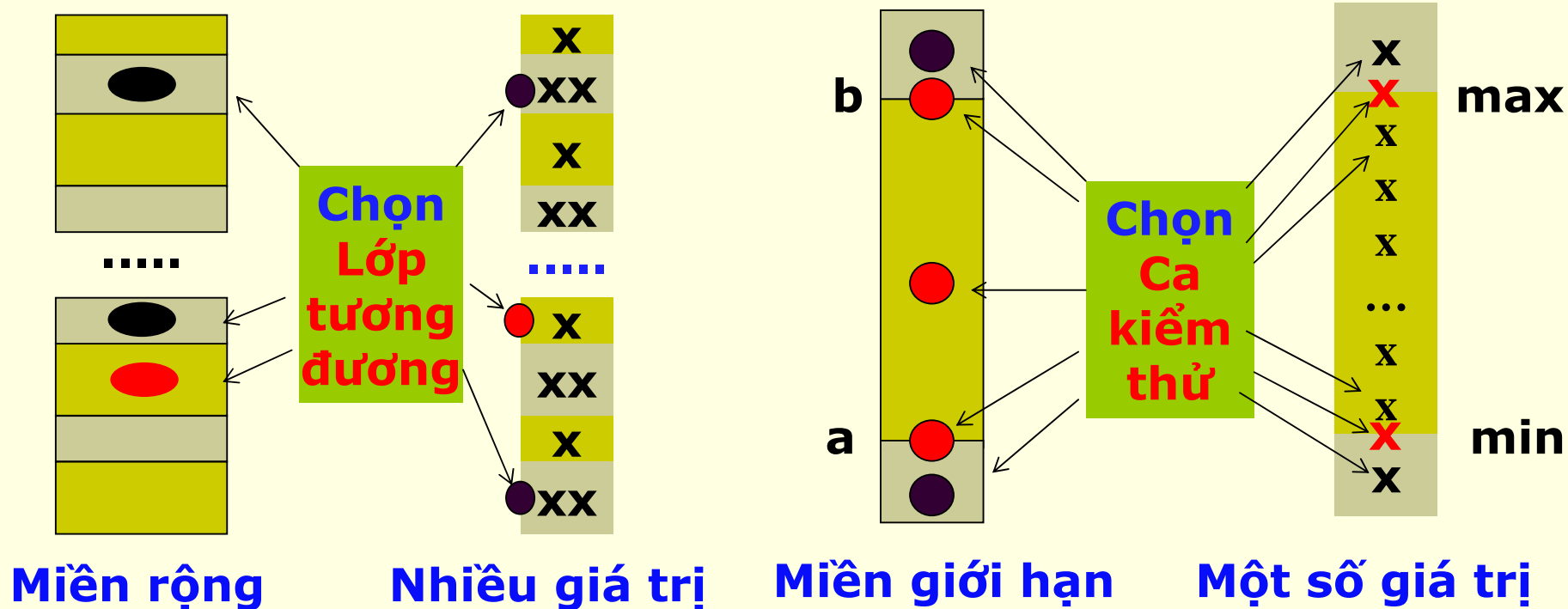
## 2. Phân tích giá trị biên

- ◆ Các sai thường ở giá trị biên
- ◆ Lựa chọn các giá trị biên của lớp phân hoạch để kiểm thử

## 3. Đồ thị nhân quả

- ◆ Lập các đồ thị nhân quả làm cơ sở xây dựng ca kiểm thử

# Mô hình phân hoạch & phân tích giá trị biên

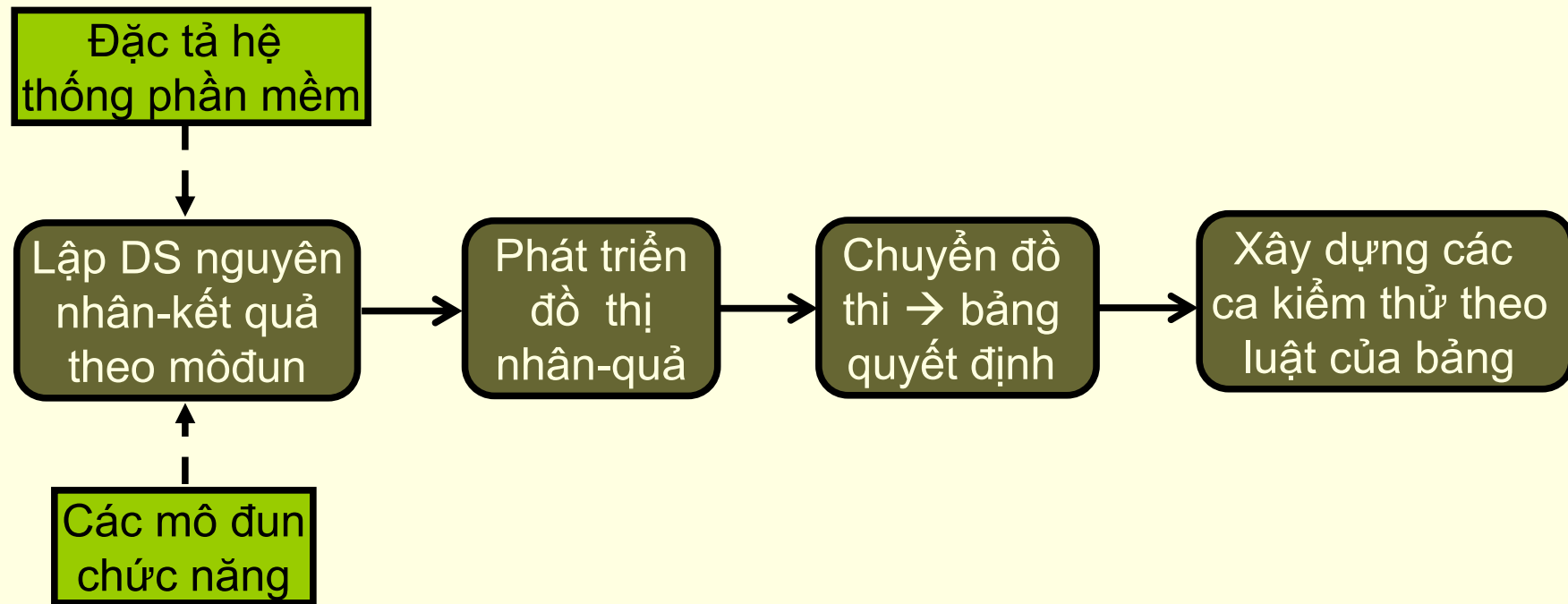


# Kỹ thuật đồ thị nhân quả



- Là một kỹ thuật để thiết kế ca kiểm thử
- Cung cấp một biểu diễn chính xác giữa các điều kiện logic (đầu vào) và các hành động tương ứng (đầu ra- kết quả).
- Được xây dựng dựa trên các mô đun chức năng, logic tiến trình và đặc tả hệ thống
- Kỹ thuật gồm 4 bước

# Tiến trình kỹ thuật nhân quả



# Ví dụ: kỹ thuật đồ thị nhân quả



## □ Danh sách nhân quả theo modul

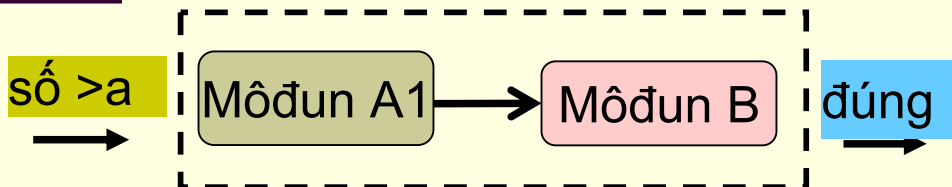
Modul	Nguyên nhân	Kết quả	Định danh
A	Số $> a$	đúng	A1
	Số $\geq a$	ngghi ngờ	A2
	Số $= a$	ngghi ngờ	A3
	Số $< a$	sai	A4
B	Số nguyên	đúng	B1



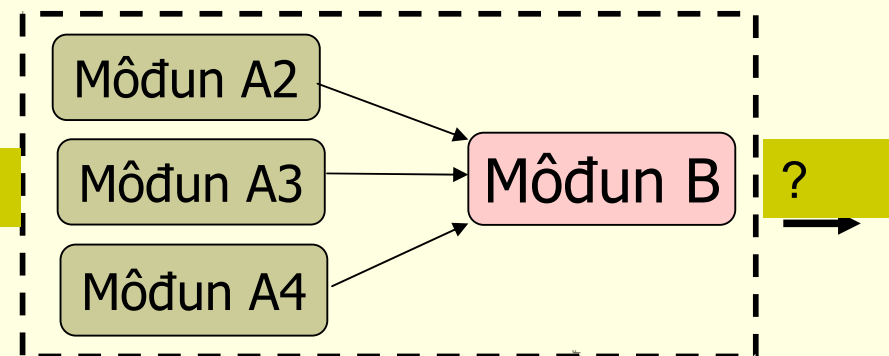
# Ví dụ: bảng quyết định đồ thị nhân quả

Định danh	Điều kiện	đúng	ngghi ngờ	sai
A1	Số > a	X		
A2,A3	Số ≥ a		X	
A4	Số < a			X
B1	Số nguyên	X		
...	...	..	..	..

**Ca 1: A1 & B**



**Ca 2: A2,A3,A4 & B1**



# Kiểm thử đơn vị



- Đối tượng: các mô đun đơn vị chương trình
- Nội dung kiểm thử:
  - ◆ giao diện: dữ liệu qua giao diện, dữ liệu vào ra
  - ◆ cấu trúc dữ liệu sử dụng cục bộ
  - ◆ đường điều khiển
  - ◆ điều kiện logic
  - ◆ phép toán xử lý
- Phương pháp sử dụng: Phương pháp hộp trắng
- Kỹ thuật: các kỹ thuật phương pháp hộp trắng và bộ lái, cưỡng

# Kiểm thử tích hợp – khái niệm



- Kiểm thử tích hợp (*integration testing*) nhằm nhận được 1 bộ phận chức năng hay 1 hệ con tốt.
- Một kỹ thuật có tính hệ thống để xây dựng cấu trúc chương trình: từ các môđun đã kiểm, xây dựng cấu trúc chương trình đảm bảo tuân theo thiết kế.
- Có hai cách tích hợp
  - ◆ Tích hợp dần: *từ trên xuống, dưới lên, kẹp*
  - ◆ Tích hợp đồng thời 1 lúc: *“big bang”*
- Phương pháp: *phương pháp hộp đen*
- Kỹ thuật: *bộ lái, cuộn*

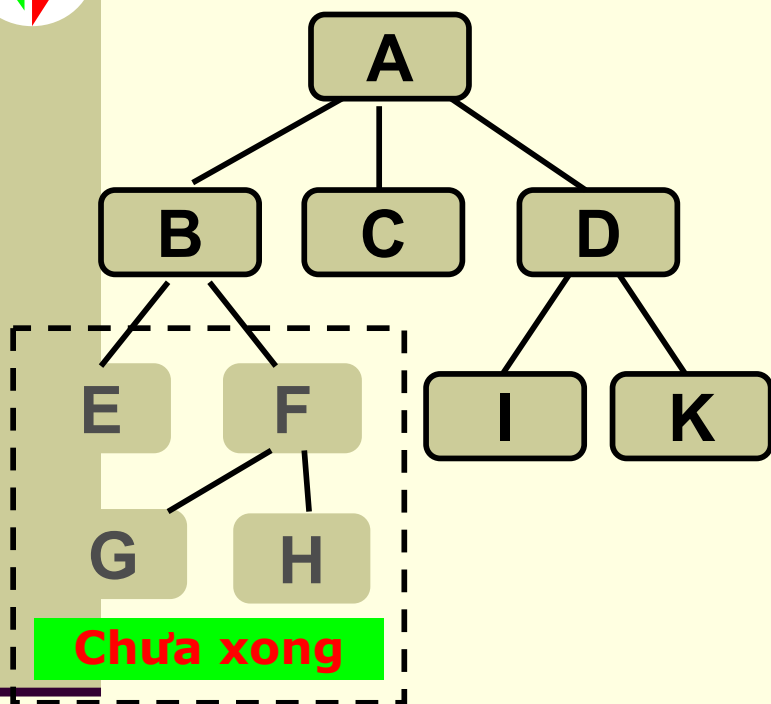
# Các sai gặp khi tích hợp mô đun



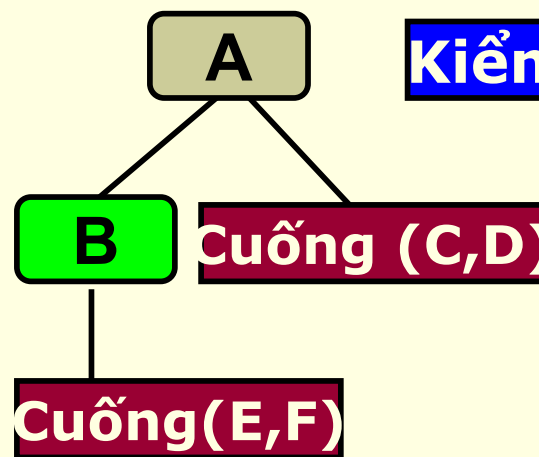
Các sai có thể gặp khi tích hợp :

- Dữ liệu bị mất khi đi qua một giao diện.
- Hiệu ứng bất lợi 1 môđun vô tình gây ra đối các môđun khác.
- Sự kết hợp các chức năng phụ có thể không sinh ra chức năng chính mong muốn.
- Sự phóng đại các sai sót riêng rẽ có thể bị đến mức không chấp nhận được.
- Vấn đề của cấu trúc dữ liệu toàn cục có thể để lộ ra .

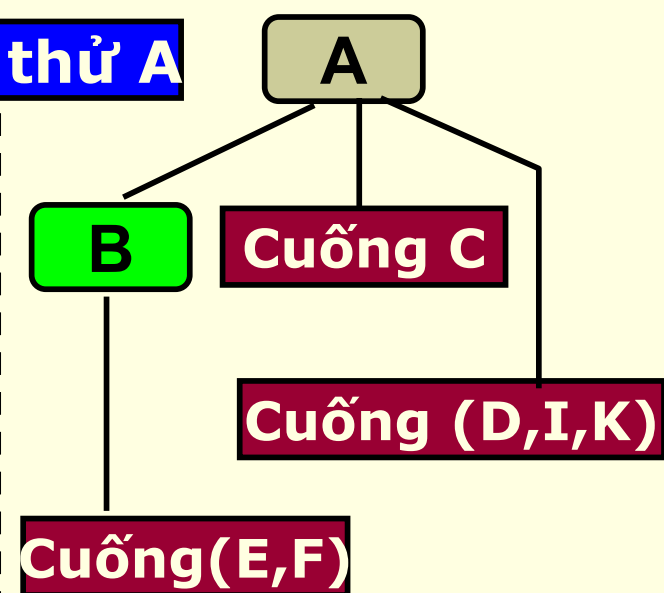
# Sơ đồ - tích hợp trên xuống



Hệ cần kiểm thử

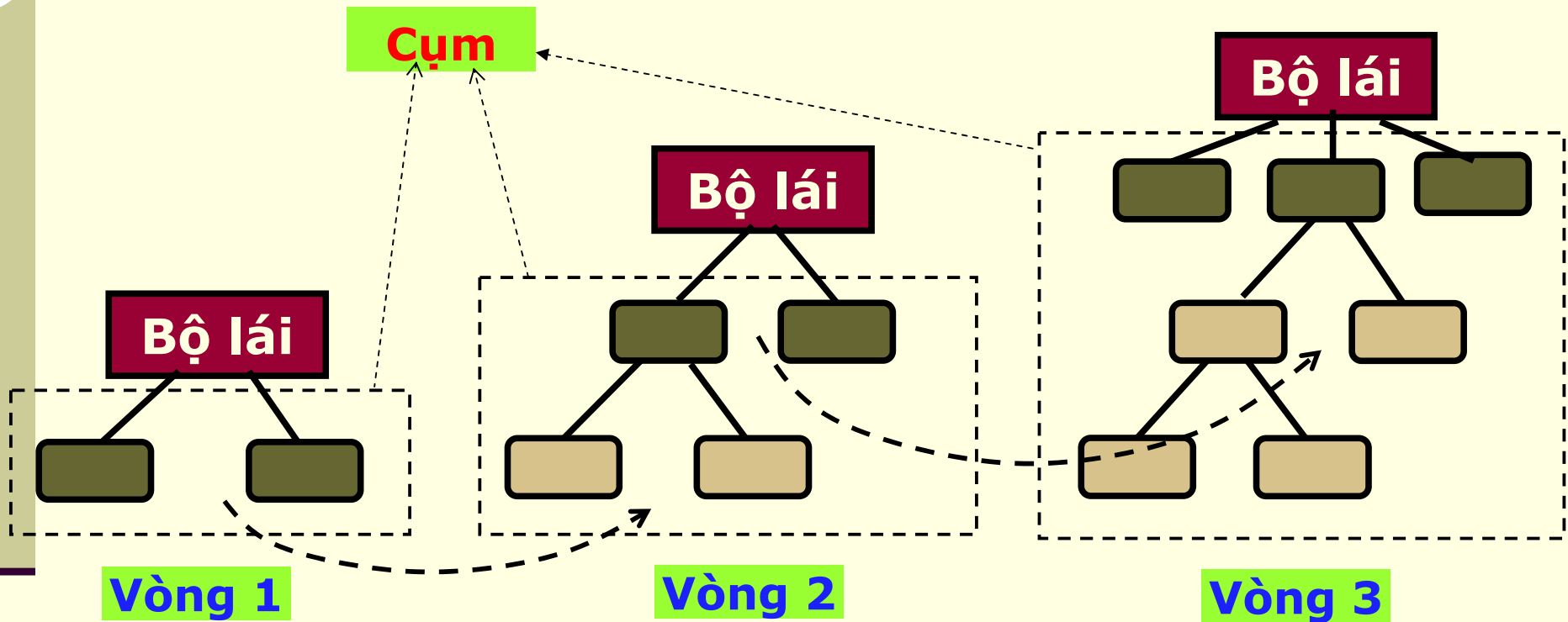


Kết hợp theo  
chiều rộng



Kết hợp theo  
chiều sâu

# Sơ đồ - tích hợp dưới lên



# Nhận xét phương pháp tích hợp



## ❑ Chiến lược Big bang

- ◆ dùng cho chương trình nhỏ
- ◆ phức tạp, không hiệu quả

## ❑ Chiến lược trên-xuống

- ◆ nhược điểm: cần các cuống
- ◆ những khó khăn kèm theo cuống.
- ◆ có ngay chức năng điều khiển hệ thống.

## ❑ Chiến lược dưới –lên:

- ◆ luôn chưa có chương trình chỉnh thể
- ◆ thiết kế ca kiểm thử dễ và không cần cuống.

# Kiểm thử hệ thống



## ■ Khái niệm kiểm thử hệ thống

- ❑ **Hệ thống** dựa trên máy tính (phần cứng & phần mềm) **do nhiều bên xây dựng**, người phát triển phần mềm chỉ là một. Chúng cần được kiểm tra tổng thể
- ❑ Những sai cần kiểm tra:
  - ◆ **Các dữ liệu qua giao diện** của các thành phần được kiểm thử
  - ◆ **Đường xử lý** liên kết các thành phần
  - ◆ **Sự tích hợp lỗi** từ các thành phần khác nhau
  - ◆ **Những hạn chế khác** đến năng lực do ảnh hưởng từ các thành phần: chịu lỗi, an toàn, thực thi



# Các loại kiểm thử hệ thống



## 1. Kiểm thử chức năng (*mức hệ thống*)

bao gồm các chức năng giao diện, các chức năng mức người dùng hay đầu ra cuối cùng khỏi hệ thống

## 2. Kiểm thử phục hồi (*chịu lỗi*)

- ◆ kiểm thử phục hồi là bắt phần mềm phải thất bại để xem khả năng phục hồi của nó đến đâu. Có 2 mức phục hồi: phục hồi tự động hay cần đến sự can thiệp của con người
- ◆ Độ tin cậy là một độ đo đánh giá khả năng phục hồi

# Các loại kiểm thử hệ thống



## 3. Kiểm thử an ninh (*sức chịu tấn công*)

- ◆ kiểm tra mọi cơ chế bảo vệ được xây dựng xem có đạt hiệu quả đề ra trước các đột nhập hay không?
- ◆ người kiểm thử đóng vai trò của kẻ đột nhập thực hiện mọi đột nhập có thể để đánh giá.

## 4. Kiểm thử thi hành (*thông suốt, kịp thời*)

- ◆ kiểm thử thi hành được thiết kế để kiểm tra sự vận hành của phần mềm khi hệ thống được tích hợp.
- ◆ Việc thi hành đúng bao gồm cả số lượng, chất lượng (hoạt động và hiệu năng)

# Các loại kiểm thử hệ thống



## 5. Kiểm thử chịu tải (*qui mô, giá trị nhạy cảm*)

- ♦ là vận hành hệ thống khi sử dụng nguồn lực với số lượng, tần suất và cường độ **dị thường**.
- ♦ Ví dụ: vận hành 1 cơ sở dữ liệu với số bản ghi cực lớn, vận hành hệ điều hành mạng với số máy nhiều dần.

# Kiểm thử chấp nhận- thẩm định



- ❑ **Mục tiêu:** xem phần mềm có đáp ứng được yêu cầu khách hàng/người dùng không?
- ❑ Thực hiện thông qua 1 loạt các **kiểm thử hộp đen**
- ❑ Kế hoạch & thủ tục được thiết kế bảo đảm rằng:
  - ◆ Tất cả các yêu cầu được **thoả mãn**,
  - ◆ Các yêu cầu thi hành đã **chính xác**,
  - ◆ Tài liệu **đúng đắn** và
  - ◆ Các yêu cầu khác là **thoả đáng**.
- ❑ Có hai loại: **kiểm thử Alpha** và **kiểm thử Beta**

# Kiểm thử Alpha và kiểm thử Beta



## ■ Kiểm thử alpha do phát triển tiến hành:

- ◆ Phần mềm được người dùng thực hiện trong bối cảnh “tự nhiên”, trong một môi trường được điều khiển
- ◆ Người phát triển “nhòm qua vai” người sử dụng để báo cáo các sai và các vấn đề sử dụng (vì thế còn gọi là **kiểm thử sau lưng**). Dữ liệu thường là dữ liệu mô phỏng

## ■ Kiểm thử Beta do khách hàng tiến hành

- ◆ Tiến hành trong môi trường thực
- ◆ Khách hàng báo cáo tất cả các vấn đề họ gặp trong quá trình kiểm thử cho người phát triển 1 cách định kỳ.

# Câu hỏi củng cố



1. Định nghĩa thẩm định và xác minh?
2. Sự khác nhau giữa thẩm định, xác minh tĩnh và động?
3. Các hoạt động chính của thẩm định và xác minh là gì?
4. Rà soát là gì? Có những loại rà soát nào?
5. Mục tiêu, đối tượng của rà soát kỹ thuật chính thức
6. Tiến trình rà soát kỹ thuật chính thức?
7. Thành phần, nội dung, kết quả hợp rà soát chính thức?
8. Nêu danh mục các sản phẩm cần rà soát?
9. Kiểm thử phần mềm là gì? Nội dung của hoạt động kiểm thử gồm những gì?

# Câu hỏi củng cố



10. Trình bày đề cương nội dung kế hoạch kiểm thử?
11. Vẽ sơ đồ dòng thông tin kiểm thử?
12. Vẽ sơ đồ tiến trình thực hiện một ca kiểm thử
13. Có những loại kiểm thử nào? Đối tượng của nó là gì?
14. Có những phương pháp và chiến lược kiểm thử nào?
15. Trình bày tóm tắt mỗi phương pháp kiểm thử: đối tượng, những sai cần kiểm tra, các chiến lược và kỹ thuật sử dụng?
16. Trình bày các loại hình kiểm thử: định nghĩa, mục tiêu, các sai cần kiểm tra, chiến lược và kỹ thuật sử dụng, cách thức tiến hành?

# Câu hỏi và thảo luận

