

ĐẠI HỌC QUỐC GIA HÀ NỘI - ĐẠI HỌC CÔNG NGHỆ  
Bộ môn Công nghệ phần mềm

---

**BÀI GIẢNG**  
**ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**  
**VÀ KIỂM THỬ**

*Nguyễn Văn Vy*

Email: [vynv@vnu.edu.vn](mailto:vynv@vnu.edu.vn), mobile: **0912.505.291**

**Hà nội - 2005**

# Phần II

*Nguyễn Văn Vy*



## KIỂM THỬ PHẦN MỀM

# Nội dung – Tài liệu

Nguyễn Văn Vỹ



- Khái niệm kiểm thử
- Các loại kiểm thử
- Thẩm định và xác minh

- ❖ Roger S. Pressman. *Software Engineering, a Practitioner's Approach*. 3th Edition, McGraw-Hill, 1992, Bản dịch của Ngô Trung Việt, Phần 4, tập 4 (Chương 17, 18, 23 –bản 2001)
- ❖ Ian Sommerville. *Software Engineering*, Sixth Edition, Addison Wesley, 2001, Phần 5 và 6. chương 20
- ❖ E.M.Bennatan, *Software Project Management : a practitioner's approach*, McGRAW-HILL Book Company, 2001

# H. Kiểm thử thẩm định

*Nguyễn Văn Vy*



- Khi kết thúc kiểm thử tích hợp thì phần mềm đã hoàn toàn được lắp ráp trong một gói, các sai giao diện đã được bộc lộ và chỉnh sửa, và một loạt các kiểm thử phần mềm cuối cùng bắt đầu - **kiểm thử thẩm định**.
- Thẩm định là thắng lợi nếu các chức năng phần mềm ở một chừng mức nào đó là có thể thoả mãn mong đợi hợp lý của người đặt hàng
- Mục tiêu thẩm định: xem phần mềm có đáp ứng được yêu cầu khách hàng không?

# h1. Khái niệm kiểm thử thẩm định(t)

*Nguyễn Văn Vy*



- Cái “mong đợi hợp lý” của khách hàng đã được xác định trong **Đặc tả yêu cầu phần mềm** (mô tả các tính chất người dùng nhìn thấy được), bao gồm cả mô tả được gọi là **tiêu chuẩn kiểm thử phần mềm**
- Thẩm định phần mềm được thực hiện thông qua một loạt các **kiểm thử hộp đen** để thuyết minh sự phù hợp với các yêu cầu.

# h1. Khái niệm kiểm thử thẩm định(t)

*Nguyễn Văn Vy*



- Một kế hoạch kiểm thử phác ra những lớp kiểm thử cần tiến hành và một thủ tục kiểm thử xác định các ca kiểm thử sẽ được dùng để thuyết minh sự phù hợp với các yêu cầu.
- Cả kế hoạch này và thủ tục này được thiết kế để bảo đảm rằng tất cả các yêu cầu được thỏa mãn, các yêu cầu thi hành đạt được, tài liệu đúng đắn và các yêu cầu khác được thỏa.

## h2. Tiêu chuẩn kiểm thử thẩm định

*Nguyễn Văn Vy*



□ Sau mỗi ca kiểm thử phần mềm ở vào một trong hai trường hợp sau:

- Các đặc tính chức năng hoặc sự thực hiện phù hợp với đặc tả và được chấp nhận.
- Một lệch lạc so với đặc tả được phát hiện và một danh sách các khiếm khuyết được tạo ra.

Ít khi các sai sót được chỉnh sửa trong giai đoạn này. Thường phải thảo luận lại với khách hàng để thiết lập một phương pháp để giải quyết các lệch lạc đó.

## h3. Rà soát cầu hình

*Nguyễn Văn Vy*



- Một yếu tố quan trọng của quá trình thẩm định là **rà soát cầu hình** (đôi khi được gọi là **kiểm toán**). Rà soát này bảo đảm rằng các yếu tố của **cấu hình** phần mềm đã thực sự được phát triển, lập danh mục, và có các chi tiết cần thiết để trợ giúp pha bảo trì của vòng đời phần mềm



## h4. Kiểm thử Alpha và Beta

*Nguyễn Văn Vy*



- ❑ Người phát triển không thể đoán trước được khách hàng thực sự dùng một chương trình như thế nào.
- ❑ Các chỉ dẫn sử dụng có thể bị hiểu lầm.
- ❑ Các tổ hợp dữ liệu lạ có thể bị sử dụng định kỳ.
- ❑ Đầu ra là rõ ràng đối với người kiểm thử nhưng có thể lại khó hiểu đối với người dùng.

# H4.1. Kiểm thử Alpha & Beta 1 khách

*Nguyễn Văn Vy*



- Khi các kiểm thử phần mềm dành cho một người đặt hàng thì một loạt hoạt động được tiến hành để chỉ 1 khách hàng thử nghiệm mọi yêu cầu.
- Tiến hành kiểm thử này là người sử dụng đầu cuối chứ không phải là người đặt hàng.
- kiểm thử chấp thuận có thể tiến hành vài tuần hoặc vài tháng một lần, nhờ đó mà bộc lộ được các lỗi tích lũy làm suy giảm hệ thống theo thời gian.

## H4.2. Kiểm thử Alpha & Beta n khách

*Nguyễn Văn Vy*



- Khi phần mềm dành cho nhiều người đặt hàng thì kiểm thử chấp thuận bởi một khách hàng là không thực tế. Và phải dùng quá trình **kiểm thử alpha và kiểm thử beta** cho nhiều người tiến hành để bộc lộ các sai mà có lẽ chỉ các người sử dụng đầu cuối mới có thể phát hiện được.

## H4.3. Kiểm thử Alpha

*Nguyễn Văn Vy*



- ❑ **kiểm thử alpha** được bên phát triển tiến hành . Phần mềm sẽ được người dùng dùng trong bối cảnh tự nhiên để người phát triển “nhòm qua vai” người sử dụng và báo cáo các sai và các vấn đề sử dụng (vì thế còn gọi là kiểm thử sau lượng).
- ❑ kiểm thử alpha được tiến hành trong một môi trường được điều khiển (theo kế hoạch của người phát triển).

## H4.4. Kiểm thử Beta

*Nguyễn Văn Vy*



- ❑ **kiểm thử beta** được nhiều người đặt hàng tiến hành , không có mặt Người phát triển.
- ❑ kiểm thử beta là áp dụng trong môi trường thực, không có sự kiểm soát của phía người phát triển.
- ❑ Khách hàng sẽ báo cáo tất cả các vấn đề (thực hoặc tưởng tượng) mà họ gặp trong quá trình kiểm thử cho người phát triển một cách định kỳ.
- ❑ Theo các báo đó Người phát triển cải biên và chuẩn bị phân phối bản phát hành bản hoàn thiện cho toàn bộ những người đặt hàng.

## h5. Kiểm thử hệ thống

*Nguyễn Văn Vy*



- ❑ Hệ thống dựa vào máy tính do nhiều bên xây dựng, người phát triển phần mềm chỉ là một
- ❑ Việc kiểm thử hệ thống dễ có nguy cơ “**đổ lỗi cho nhau**”.
- ❑ Người phát triển phần mềm cần đoán trước các vấn đề giao diện có thể nảy ra, và
- ❑ Phát hiện các thiết kế đường xử lý sai thông qua kiểm thử tất cả các thông tin đến từ các phần tử khác của hệ thống.

## h5. Kiểm thử hệ thống (t)

*Nguyễn Văn Vy*



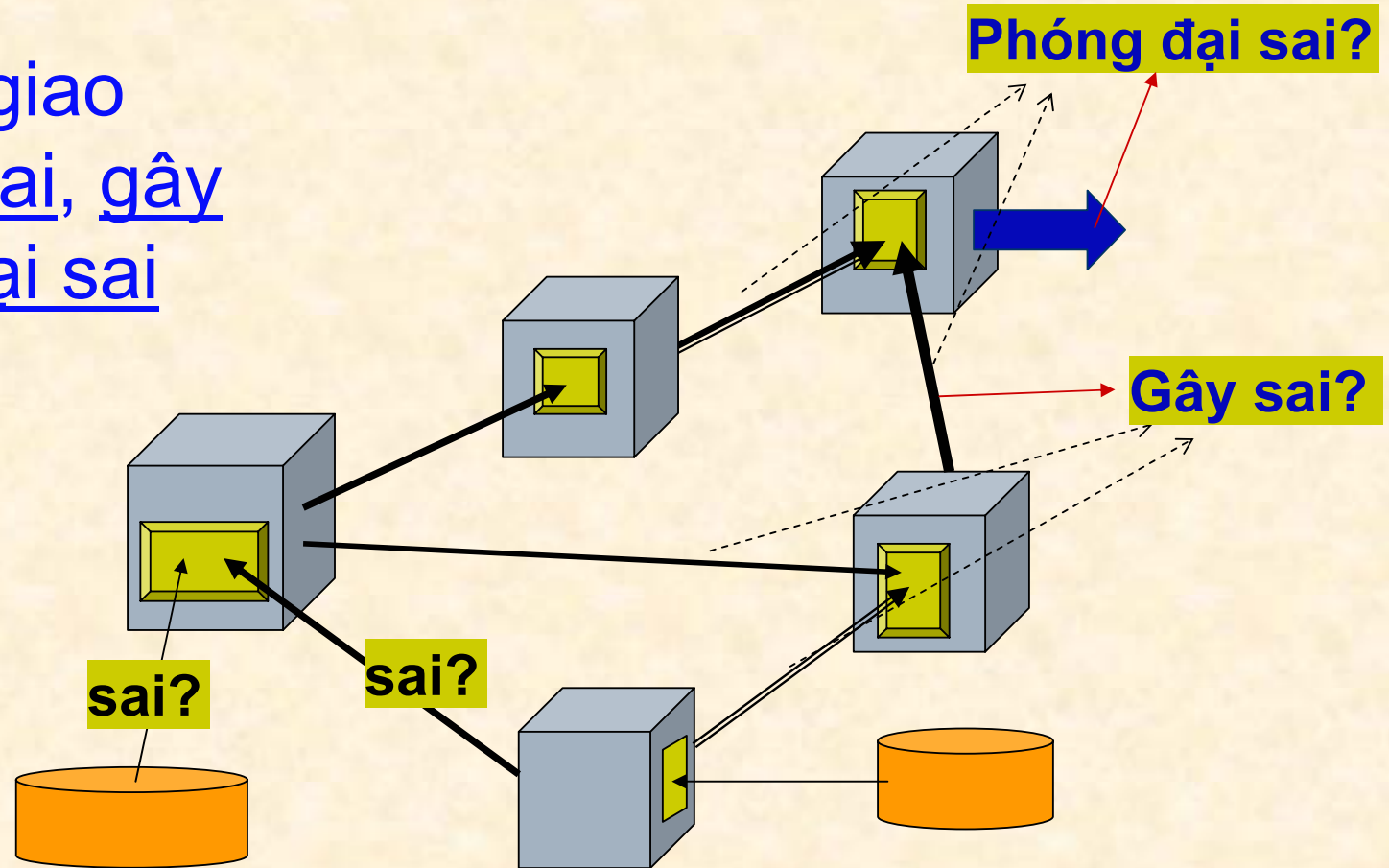
- ❑ Tiến hành các kiểm thử mô phỏng các dữ liệu xấu hoặc các sai tiềm tàng khác tại giao diện phần mềm.
- ❑ Báo cáo các kết quả kiểm thử để làm chứng cứ phòng ngừa đổ lỗi cho nhau.
- ❑ Những người tham gia vào trong việc hoạch định và thiết kế các kiểm thử hệ thống sao cho kế hoạch và kiểm thử bảo đảm phần mềm được kiểm thử đầy đủ

# h5. Kiểm thử hệ thống (t)

Nguyễn Văn Vy



Dữ liệu qua giao diện có thể sai, gây sai, phóng đại sai





## H6. Kiểm thử hồi phục

*Nguyễn Văn Vy*



- ❑ Nhiều hệ thống cần phải phục hồi sau lỗi, để tiếp tục xử lý trong một thời gian đã đặc tả trước:
- ❑ Có trường hợp, hệ thống cần thử lỗi: nghĩa là xử lý lỗi bắt buộc không được làm ngừng hoạt động của toàn hệ thống.
- ❑ Trường hợp khác, lỗi phải được khắc phục dần theo từng chu kỳ đã đặc tả.
- ❑ kiểm thử hồi phục là kiểm thử bất phân mềm phải thất bại để xem có hồi phục được hay không.

## h6. Kiểm thử hồi phục (t)

*Nguyễn Văn Vỹ*



Có 2 cách hồi phục:

- Phục hồi tự động: bằng khởi động lại (cơ chế checkpoint). Sau khi phục hồi dữ liệu, hệ thống tự khởi động lại thì được đánh giá là đúng đắn.
- Phục hồi có sự can thiệp của con người: phải đánh giá thời gian trung bình để sửa chữa và xác định xem liệu nó đã ở trong giới hạn chấp nhận được không?

## h7. Kiểm thử an ninh

*Nguyễn Văn Vỹ*



- ❑ Kiểm tra cơ chế bảo vệ được xây dựng trong hệ thống xem có đạt hiệu quả trước các đột nhập hay không?
- ❑ Xét tất cả các loại đột nhập có thể “trước mặt”, “ngang sườn” và “sau lưng”.
- ❑ Khi thử nghiệm an ninh, người kiểm thử sẽ đóng vai trò của kẻ Đột nhập.

## h7. Kiểm thử an ninh(t)

Nguyễn Văn Vỹ



- Về nguyên tắc: Mọi đột nhập là có thể nếu đủ thời gian và nguồn lực.
- Bài toán thiết kế hệ thống đặt ra là:
  - làm cho việc đột nhập tốn phí nhiều hơn giá trị thu được do đột nhập
  - Công sức bỏ ra xây dựng công cụ bảo vệ phải ít hơn giá trị mất đi nếu bị đột nhập

***Chi phí công cụ bảo vệ < lợi ích do bảo vệ khỏi đột nhập***  
***Chi phí để đột nhập > lợi ích thu được từ đột nhập***

## h8. Kiểm thử áp lực

Nguyễn Văn Vy



- Các kỹ thuật hộp trắng và hộp đen được dùng để đánh giá chức năng và sự thi hành của chương trình bình thường.
- Kiểm thử áp lực là vận hành hệ thống đòi hỏi nguồn lực với số lượng, tần suất và cường độ dị thường.
- Một loại khác của thử nghiệm áp lực là kiểm thử nhay cảm: cố gắng làm bộc lộ các tổ hợp dữ liệu (lớp dữ liệu vào có hiệu lực) hay sự kiện mà có thể gây ra việc xử lý không ổn định hoặc không chính xác.

## h9. Kiểm thử thi hành

Nguyễn Văn Vỹ



- Với các hệ nhúng & hệ thời gian thực, phần mềm cung cấp chức năng nhưng không phù hợp với các yêu cầu thi hành đều là không chấp nhận được.
- kiểm thử thi hành được thiết kế để kiểm thử việc thi hành (run-time) của phần mềm khi hệ thống được tích hợp.
- kiểm thử thi hành xuất hiện trong tất cả các bước của quá trình kiểm thử, tuy nhiên chỉ đến khi tất cả các phần tử của hệ thống đã được tích hợp thì kiểm thử mới có thể thực sự là chắc chắn.

## h9. Kiểm thử thi hành(t)

*Nguyễn Văn Vy*



- kiểm thử thi hành đôi khi gắn liền với kiểm thử áp lực vì cả hai thường đòi hỏi các dụng cụ phần cứng và phần mềm. Đó là do cần đo sự tổng hợp nguồn lực (trong, ngoài). Nhờ dụng cụ ngoại lai để thể giám sát các khoảng vận hành, các sự kiện ngắt (log) khi nó xuất hiện, có thể lấy mẫu các trạng thái máy.
- kiểm thử có thể làm bộc lộ các tình thế dẫn đến sự suy giảm hoặc thất bại hệ thống tiềm ẩn.

# I. Nghệ thuật gỡ rối

*Nguyễn Văn Vỹ*



- ❑ Kiểm thử thành công dẫn đến việc gỡ lỗi, kết quả của kiểm thử thường mới chỉ ra cho kỹ sư phần mềm những triệu chứng của vấn đề. Có thể chưa rõ nguyên nhân: Biểu lộ bên ngoài của sai & nguyên nhân bên trong của sai có thể không có quan hệ rõ ràng.
- ❑ Gỡ lỗi không phải là kiểm thử. Mà là tìm nguyên nhân gây lỗi để loại trừ lỗi - khác với kiểm thử



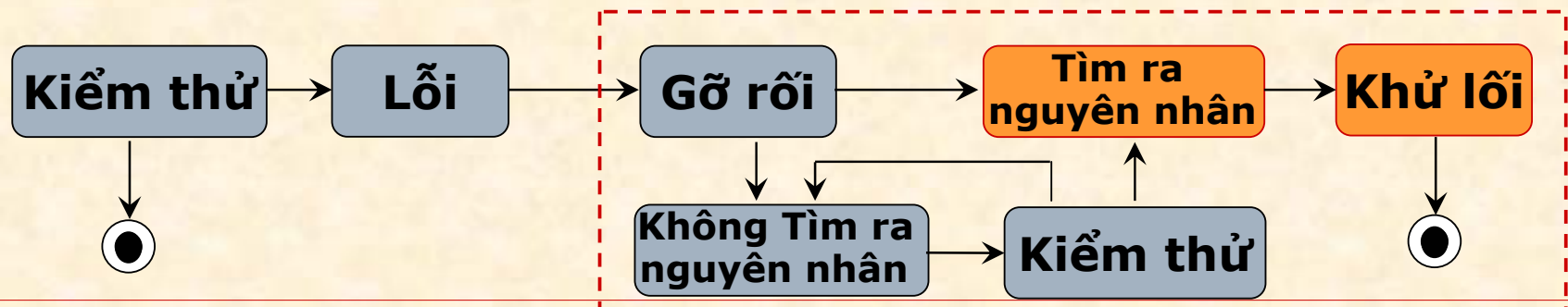
# I1. Tiến trình gỡ rối

Nguyễn Văn Vỹ

□ Quá trình gỡ rối luôn dẫn tới hai khả năng:

- Tìm ra nguyên nhân, chỉnh sửa và khử được lỗi.
- Không tìm được nguyên nhân.

Trường hợp này, người gỡ lỗi có thể nghi ngờ một số nguyên nhân nào đó và cần thiết kể một ca kiểm thử để giúp việc thẩm định nghi ngờ và như vậy công việc tìm sai lại dẫn đến tiếp tục kiểm thử như một vòng lặp.



# I1. Tiến trình gỡ rối (t)

*Nguyễn Văn Vỹ*



## □ Gỡ lỗi là khó khăn:

- Triệu chứng có thể xa nguyên nhân (về không gian).
- Triệu chứng có thể tạm thời biến mất khi có một sai khác được sửa.
- Triệu chứng có thể thực gây ra bởi sai của người dùng mà không dễ theo dõi được.
- Triệu chứng có thể là kết quả của vấn đề thời gian chứ không phải là vấn đề xử lý.

# I1. Tiến trình gỡ rối (t)

*Nguyễn Văn Vỹ*

- Có thể khó tái thể hiện các điều kiện đầu vào (ứng dụng thời gian thực, thứ tự đầu vào không xác định).
- Triệu chứng có thể là bị gián đoạn (các hệ nhúng với liên kết phần cứng và phần mềm không tháo rời được).
- Triệu chứng có thể do các nguyên nhân được phân tán trong một số các nhiệm vụ chạy trên các bộ xử lý khác nhau.

## I2. Vấn đề tâm lý trong gỡ lỗi

*Nguyễn Văn Vỹ*



- ❑ Nhiều chứng cứ cho rằng tài gỡ lỗi là một thứ bẩm sinh của con người.
- ❑ Người này thì giỏi gỡ lỗi, kẻ khác lại không; và rất khó dạy và khó học gỡ lỗi.
- ❑ Tuy nhiên người ta cũng đã có vài đề xuất phương cách gỡ lỗi.

# 13. Các cách thức gỡ rối

*Nguyễn Văn Vỹ*



- Nói chung có 3 lựa chọn cách gỡ lỗi:
  - Vũ dũng vô mưu.
  - Lăn theo vệt cũ.
  - Loại trừ dần các nguyên nhân.

# 13.1. Phương cách vũ dũng vô mưu

*Nguyễn Văn Vỹ*



- ❑ Phương cách vũ dũng vô mưu là phương sách chung nhất nhưng lại ít hiệu quả nhất; áp dụng phương sách này khi chẳng còn cách nào khác.
- ❑ Triết lý “hãy để cho máy tính tìm ra sai”: xem xét tất cả, ghi lại tất cả với hy vọng tìm ra trong đồng thông tin khổng lồ đó cái nguyên nhân của sai!
- ❑ Một cách nghĩ thô thiển, nhiều trường hợp không khả thi

## 13.2. Phương cách lần theo vết cũ

*Nguyễn Văn Vỹ*



- Phương cách lùi theo vết cũ cũng là ít thông dụng, và có thể dùng thắng lợi trong các chương trình nhỏ.
- Bắt đầu lần ngược từ chỗ thấy triệu chứng sai trong mã nguồn (bằng tay!) cho tới khi định vị được sai.
- Khi số dòng lệnh tăng lên thì số con đường đi ngược có thể nhiều đến mức không quản lý nổi.

## 13.3. Phương cách loại trừ nguyên nhân

*Nguyễn Văn Vỹ*



- Cách làm: tổ chức lại dữ liệu liên quan đến xuất hiện sai để cô lập nguyên nhân tiềm ẩn.
- Một giả thiết nguyên nhân được đưa ra từ dữ liệu trên được dùng để chứng tỏ hoặc phản chứng giả thiết đó. một danh sách các nguyên nhân tiềm ẩn được vạch ra và các kiểm thử được tiến hành để loại trừ các nguyên nhân trong danh sách đó.
- Nếu kiểm thử chỉ ra một nguyên nhân nào đó có hứa hẹn thì tinh chế tiếp dữ liệu liên quan để **cô lập lỗi**



## 13.3. Phương cách loại trừ nguyên nhân(t)

*Nguyễn Văn Vỹ*



- Mỗi khi tìm thấy lỗi thì phải chỉnh sửa. Nhưng nên nhớ rằng sửa một sai có thể gây ra nhiều sai khác. Làm sao để sửa triệt để?
- Trước khi sửa nên tự hỏi để tìm giải pháp:
  - Nguyên nhân này có thể sinh ra ở phần khác của chương trình hay không? (trong nhiều tình thế, một khiếm khuyết chương trình có thể gây ra một mẫu sai logic mà nó có thể được sinh ra ở một nơi khác).

## 13.3. Phương cách loại trừ nguyên nhân(t)

*Nguyễn Văn Vy*



- Lỗi nào có thể được sinh ra tiếp? Trước khi chỉnh sửa nên xét lại mã nguồn (tốt hơn hết là xét lại thiết kế) để đánh giá sự gắn kết giữa logic và cấu trúc dữ liệu.
- Làm gì để ngăn cản lỗi này ngay từ chỗ đầu tiên? (thường có nhiều giải pháp để nhà thiết kế lựa chọn).

## m. Quản lý cầu hình

*Nguyễn Văn Vỹ*



### □ Quản lý cấu hình phần mềm:

- Là tập các hoạt động để quản lý các thay đổi của phần mềm trong suốt vòng đời của nó.
- Một loại hoạt động bảo đảm chất lượng phần mềm, được áp dụng cho tất cả các pha của kỹ nghệ
- Bao trùm suốt tiến trình phát triển và tiến hóa của phần mềm.

# m1. Nội dung quản lý cầu hình

*Nguyễn Văn Vy*



- **Nội dung** quản lý cầu hình phần mềm bao gồm:
  - Xác định các thay đổi.
  - Kiểm soát các thay đổi.
  - Bảo đảm các thay đổi đã được thực hiện.
  - Báo cáo các thay đổi cho người quan tâm..

## m2. Phân biệt với bảo trì

*Nguyễn Văn Vỹ*



- Quản lý cấu hình khác bảo trì phần mềm:
  - **Bảo trì phần mềm** là các hoạt động kỹ nghệ xuất hiện sau khi phân phát phần mềm và nó đi vào hoạt động.
  - **Quản lý cấu hình phần mềm** là các hoạt động theo dõi và kiểm soát , từ bắt đầu dự án phát triển phần mềm và chỉ kết thúc khi mà phần mềm không hoạt động nữa.

## m3. Thành phần của phần mềm

*Nguyễn Văn Vỹ*



- Kết quả của tiến trình kỹ nghệ phần mềm là các thông tin có thể được chia thành 3 loại:
  - Các chương trình máy tính (cả mức nguồn và mức chạy được).
  - Các tài liệu mô tả chương trình máy tính đó (nhắm đến cả những người thực hành kỹ thuật lẫn những người dùng).
  - Các cấu trúc dữ liệu (cả bên trong và ngoài chương trình).

# m4. Cầu hình phần mềm

*Nguyễn Văn Vy*



- Các khoản mục cấu thành lên các thành phần của phần mềm được sản ra như là những chế tác của tiến trình kỹ nghệ phần mềm được tập hợp lại trong một cái tên chung gọi là cầu hình phần mềm.
- Các chế tác này có nhiều mức khác nhau:
  - Bộ phận - tổng thể (phạm vi)
  - Chưa hoàn thiện – hoàn thiện (theo tiến trình, chất lượng)
  - Ở các mức tiến hóa khác nhau (các phiên bản)

# m5. Công cụ quản lý cấu hình

Nguyễn Văn Vy

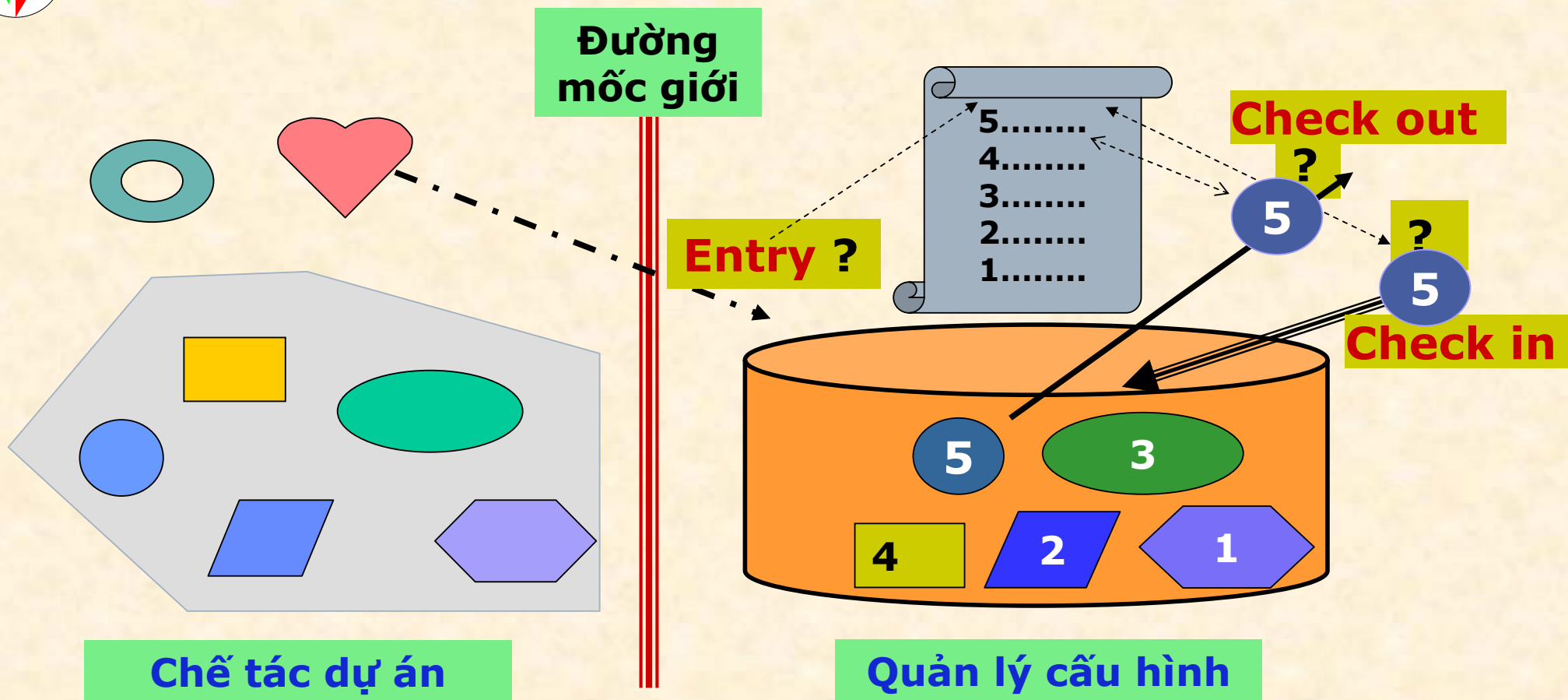


- Các đường mốc giới là ranh giới được đặt ra:
  - Trước nó thì cấu hình có thể thay đổi nhanh chóng và không chính thức.
  - Sau nó thì cấu hình có thể thay đổi, nhưng cần các thủ tục đặc biệt và chính thức để có thể đánh giá và kiểm thử từng thay đổi một.
- Đường mốc giới để đánh dấu việc phân phát **một vài khoản mục cấu hình** phần mềm - SCI.
- Tại đường mốc các **khoản mục cấu hình** phần mềm tương ứng **được đưa vào cơ sở dữ liệu dự án**



# m5. Công cụ quản lý cấu hình (t)

Nguyễn Văn Vy



# m6. Các khoản mục cấu hình phần mềm

*Nguyễn Văn Vỹ*



- Đặc tả hệ thống
- Kế hoạch dự án phần mềm.
- Đặc tả yêu cầu:
  - Đặc tả yêu cầu phần mềm.
  - Nguyên mẫu thi hành được hoặc nguyên mẫu “giấy tờ”
  - Sổ tay sử dụng sơ đẳng

# m6. Các khoản mục cấu hình phần mềm(t)

*Nguyễn Văn Vỹ*



## □ Đặc tả thiết kế:

- Đặc tả thiết kế dữ liệu.
- Đặc tả thiết kế kiến trúc.
- Đặc tả thiết kế môđun.
- Đặc tả thiết kế giao diện.
- Đặc tả đối tượng (nếu dùng kỹ thuật hướng đối tượng)

# m6. Các khoản mục cấu hình phần mềm(t)

*Nguyễn Văn Vy*



- Mã nguồn và kiểm thử :
  - Kế hoạch và thủ tục kiểm thử .
  - Các ca kiểm thử & các kết quả được ghi lại.
  - Các sổ tay vận hành & sổ tay lắp đặt.
  - Chương trình thi hành được.
  - Các môđun & mã thi hành được
  - Các môđun đã liên kết

# m6. Các khoản mục cấu hình phần mềm(t)

*Nguyễn Văn Vỹ*



- Mô tả cơ sở dữ liệu
  - Lược đồ & cấu trúc các file
  - Nội dung hồ sơ ban đầu
- Sổ tay người sử dụng
- Các tài liệu bảo trì
  - Các báo cáo những vấn đề phần mềm
  - Các yêu cầu bảo trì
  - Đặt thay đổi kỹ nghệ
  - Các chuẩn & các thủ tục cho kỹ nghệ phần mềm.

# m7. Sự hình thành quản lý cấu hình

*Nguyễn Văn Vỹ*



- Trách nhiệm nguyên thủy của **quản lý cấu hình phần mềm – SCM** là kiểm soát các thay đổi
- Sau này thêm các trách nhiệm:
  - Xác định các khoản mục cấu hình, các version của phần mềm;
  - kiểm toán cấu hình phần mềm nhằm bảo đảm phần mềm đã được phát triển đúng và
  - báo cáo mọi thay đổi đã được áp dụng cho cấu hình đó.

# m8. Nhiệm vụ quản lý cấu hình

*Nguyễn Văn Vy*



- 5 nhiệm vụ cụ thể quản lý cấu hình phần mềm:
  - Xác định cấu hình
  - kiểm soát version
  - kiểm soát đổi thay,
  - kiểm toán cấu hình,
  - báo cáo thay đổi.
  
- Mọi cuộc thảo luận về quản lý cấu hình phần mềm cần đưa ra các câu hỏi:

## m8. Nhiệm vụ quản lý cấu hình(t)

*Nguyễn Văn Vỹ*



1. Làm thế nào để tổ chức minh định và quản lý được nhiều version của chương trình sao cho nó có thể thay đổi được để thích nghi một cách hiệu quả?
2. Làm thế nào để tổ chức kiểm soát được các đổi thay phần mềm trước và sau khi phân phát cho người đặt hàng?



## m8. Nhiệm vụ quản lý cấu hình(t)

*Nguyễn Văn Vỹ*



3. Ai chịu trách nhiệm việc chấp thuận và đặt thứ tự ưu tiên của các đổi thay?
4. Làm thế nào có thể bảo đảm rằng việc đổi thay đã được thực hiện đúng?
5. Dùng cơ cấu nào để đánh giá các đổi thay khác?

# m9. Xác định đối tượng cấu hình FM

*Nguyễn Văn Vỹ*



- Cần đặt tên không trùng cho các khoản mục cấu hình phần mềm, để kiểm soát quản lý và tổ chức lại theo phương cách hướng đối tượng.
- Có hai loại đối tượng:
  - Đối tượng cơ bản là một “đơn vị văn bản”, được kỹ sư phần mềm tạo ra trong quá trình phân tích thiết kế, lập mã và kiểm thử .
  - Đối tượng hỗn hợp được cấu thành từ các đối tượng

## m9. Xác định đối tượng cấu hình FM(t)

*Nguyễn Văn Vỹ*



- Mỗi đối tượng có một bộ các đặc tính mà thể hiện của nó là duy nhất: tên, mô tả, danh sách các nguồn lực, sự hiện thực hoá.
- Mô tả đối tượng bằng một danh sách các khoản mục dữ liệu:
  - Kiểu khoản mục cấu hình phần mềm (tài liệu hay chương trình hay dữ liệu).
  - Chứng thư dự án (thuộc phần nào trong dự án).
  - Thông tin đổi thay và/hoặc thông tin version

## m9. Xác định đối tượng cấu hình FM(t)

*Nguyễn Văn Vỹ*



- Nguồn lực là tất cả các thực thể được cung cấp, xử lý, tham khảo, và các thứ khác được đối tượng cần đến.
- Mối quan hệ giữa các đối tượng là **quan hệ bộ phận – toàn bộ**. Ta có đồ thị các đối tượng.
- Một quan hệ khác là **quan hệ liên quan** với nhau (<interrelated>).
- Để kiểm soát đổi thay của đối tượng ta cần đến đồ thị tiến hoá cho từng đối tượng, nó mô tả lịch sử đổi thay của đối tượng đó.

# m10. Kiểm soát phiên bản

*Nguyễn Văn Vy*



- **Kiểm soát phiên bản** = tổ hợp các thủ tục & các công cụ để quản lý các phiên bản khác nhau của các đối tượng cấu hình (đã được tạo ra trong tiến trình kỹ nghệ phần mềm).
- **Quản lý cấu hình** cho phép người sử dụng đặc tả các cấu hình thay thế của hệ thống phần mềm = lựa chọn các phiên bản thích hợp và gắn kết với các thuộc tính; nhờ đó mà cho phép đặc tả một cấu hình bằng mô tả tập các thuộc tính mong muốn.

# m10. Kiểm soát phiên bản (t)

*Nguyễn Văn Vy*



- ❑ Để xây dựng một biến thể thích hợp của một phiên bản của một chương trình, mỗi thành phần của phiên bản được gán một “bộ thuộc tính” - là một danh sách các đặc điểm
- ❑ Một phiên bản hay biến thể được xây dựng cần xác định thành phần nào được dùng hay cần thay đổi.
- ❑ Một cách khác để hình thành khái niệm về quan hệ giữa các thành phần, các biến thể, các phiên bản là biểu diễn chúng như là một vụng (pool) đối tượng.

# m10. Kiểm soát phiên bản(1)

*Nguyễn Văn Vy*



- ❑ Mỗi thành phần được cấu tạo bởi một bộ các đối tượng trong cùng một mức xét duyệt.
- ❑ Mỗi biến thể là một bộ các đối tượng trong cùng một mức xét duyệt.
- ❑ Xác định khi các thay đổi mỗi phiên bản chủ yếu đã được thực hiện đối với một vài đối tượng.
- ❑ Đã có một số các công cụ như: SCCS, RCS và hiện đại hơn như: NSE, DSEE.

# m11. Kiểm soát sự thay đổi

*Nguyễn Văn Vỹ*



- ❑ **Tại sao?** Khi phát triển phần mềm lớn, không kiểm soát được các thay đổi sẽ dễ dẫn đến hỗn độn.
- ❑ **Phương cách:** cần phối hợp cả các thủ tục , con người , các công cụ tự động nhằm cung cấp một cơ chế để kiểm soát sự đổi thay.
- ❑ **Tiến trình** kiểm soát đổi thay có thể mô tả như biểu đồ sau:



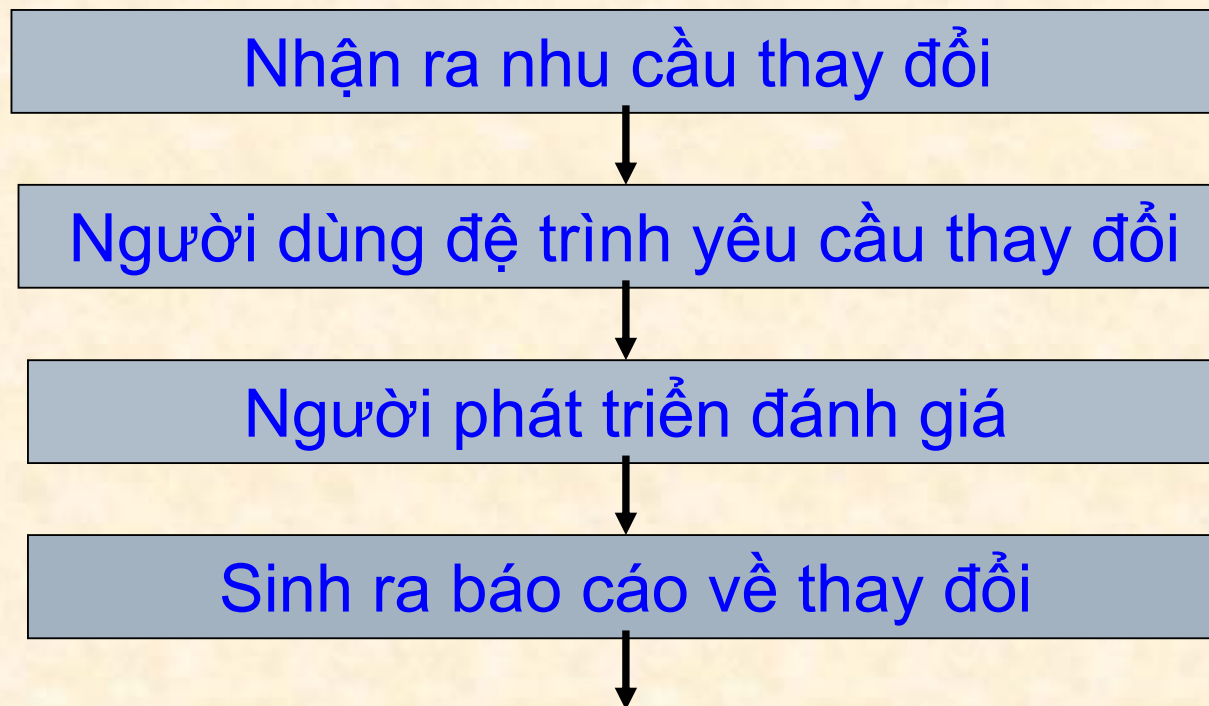


# m11.1. Tiến trình kiểm soát sự thay đổi

*Nguyễn Văn Vy*



- ❑ Biểu đồ tiến trình kiểm soát thay đổi:

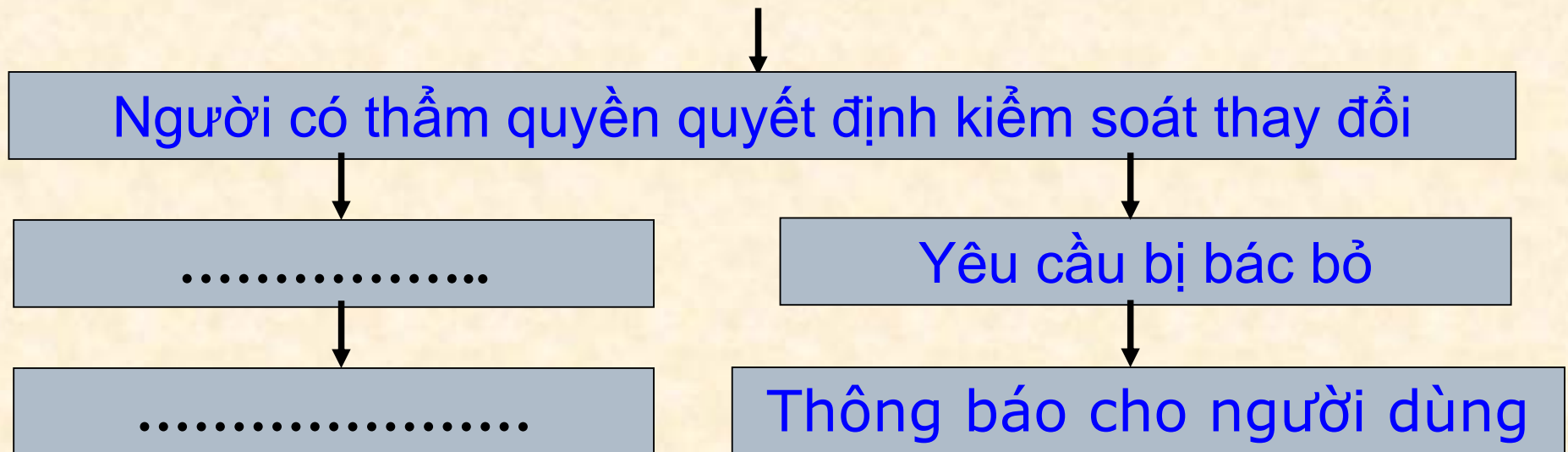


# m11.1 Tiến trình kiểm soát sự thay đổi(t)

*Nguyễn Văn Vy*



- ❑ Quá trình kiểm soát đổi thay như sau:

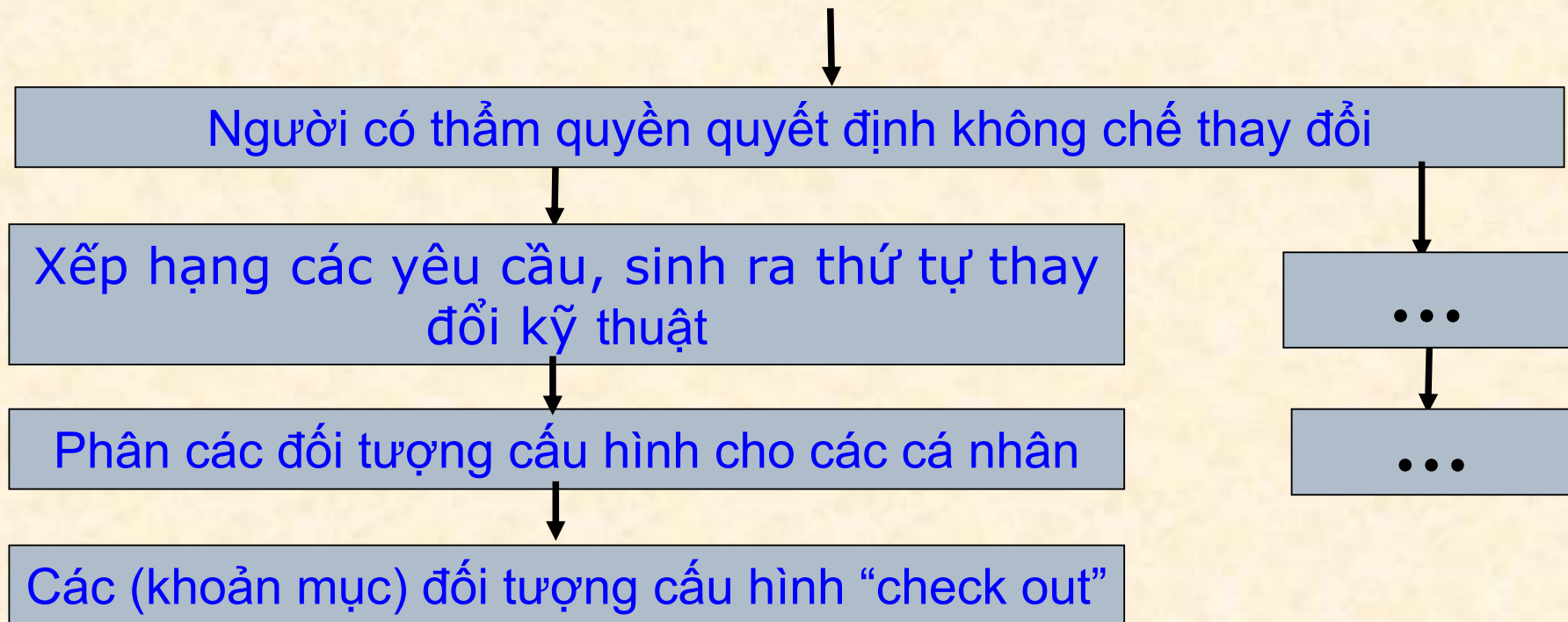


# m11.1 Tiến trình kiểm soát sự thay đổi(t)

*Nguyễn Văn Vỹ*



- ❑ Quá trình kiểm soát đổi thay như sau:



# m11.1 Tiến trình kiểm soát sự thay đổi(t)

*Nguyễn Văn Vy*

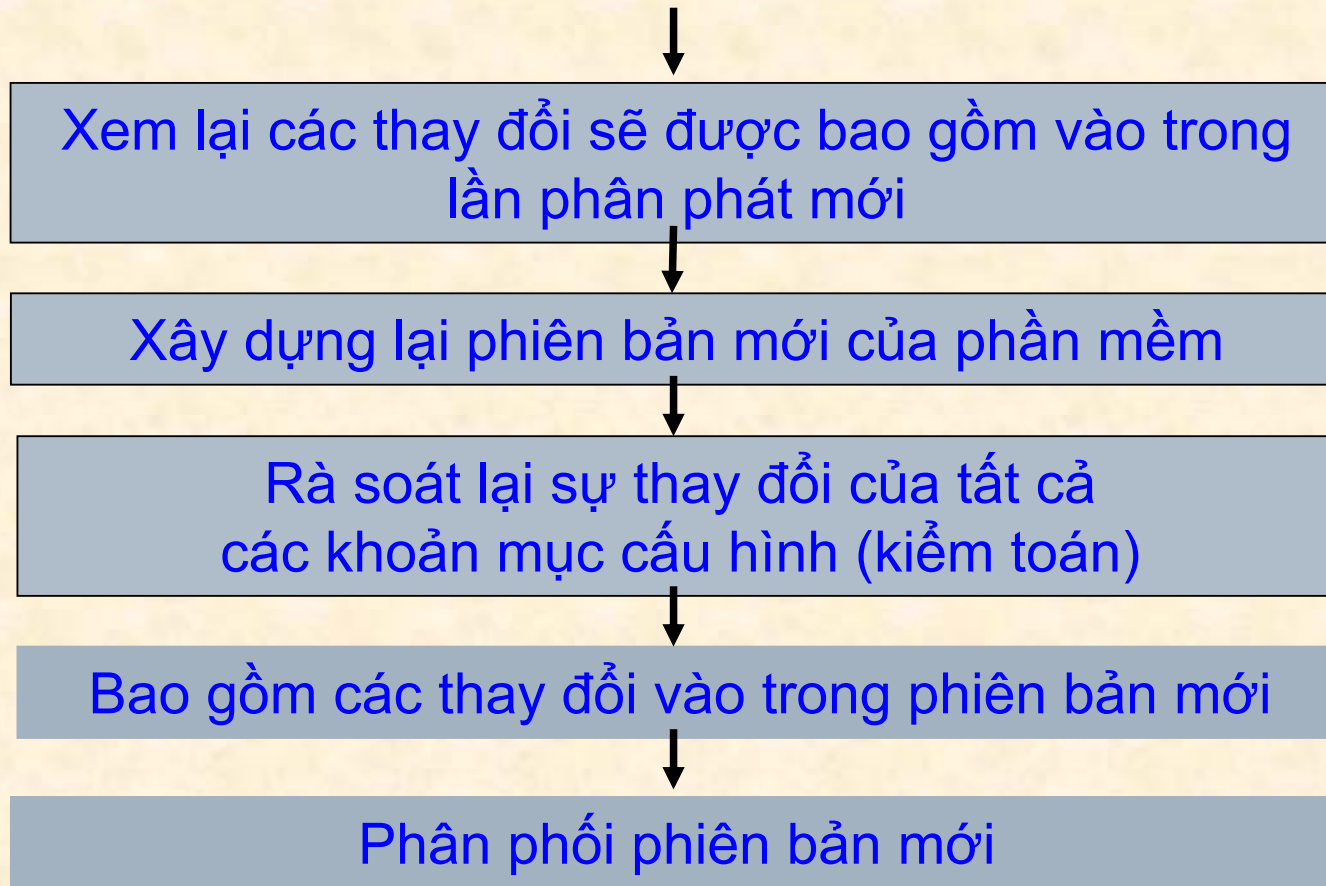


- ❑ Quá trình kiểm soát đổi thay như sau:



# m11.1 Tiến trình kiểm soát sự thay đổi(t)

*Nguyễn Văn Vỹ*



# m11.2 Hoạt động kiểm soát sự thay đổi

*Nguyễn Văn Vy*



- Khi một yêu cầu thay đổi được đệ trình cần định giá trị nhằm:
  - sử dụng kỹ thuật thích
  - hiệu ứng phụ có thể có, ảnh hưởng lên tổng thể các đối tượng cấu hình khác & lên các chức năng hệ thống & lên chi phí dự án vì thay đổi đó..
- Kết quả đánh giá về sự đổi thay được báo cáo cho người có **thẩm quyền kiểm soát đổi thay – CCA** -, có quyền tối hậu về tình trạng & quyết định đổi thay.

## m11.2. Hoạt động kiểm soát sự thay đổi(t)

*Nguyễn Văn Vỹ*

- Một mệnh lệnh đổi thay kỹ thuật (ECO) được tạo ra cho từng đổi thay được chấp thuận.
- Lệnh này mô tả các đổi thay cần làm, các ràng buộc phải tuân theo, các tiêu chuẩn rà soát và kiểm toán.
- Đối tượng cần thay đổi được “check out” khỏi cơ sở dữ liệu dự án, được thay đổi, và các hoạt động bảo đảm chất lượng phần mềm cần được áp dụng.
- Sau đó đối tượng này được “check in” vào cơ sở dữ liệu & một cơ chế kiểm soát phiên bản được sử dụng.

## m11.2. Hoạt động kiểm soát sự thay đổi(t)

*Nguyễn Văn Vy*



- ❑ Quá trình “**check in**” và “**check out**” thực thi hai điều quan trọng của kiểm soát đổi thay là: kiểm soát truy cập và kiểm soát đồng bộ.
- ❑ **kiểm soát truy cập** quản trị những cái gì mà người kỹ sư có thẩm quyền truy cập và cải biên một đối tượng cấu hình đặc biệt.
- ❑ **kiểm soát đồng bộ** giúp ta bảo đảm rằng các đổi thay song song, hoàn thành bởi những người khác nhau không viết đè lên nhau.



## m11.2. Hoạt động kiểm soát sự thay đổi(t)

*Nguyễn Văn Vỹ*



- “check out” một đối tượng cấu hình dựa trên yêu cầu đổi thay đã được chấp thuận và thứ tự đổi thay kỹ thuật cho phép người kỹ sư phần mềm.
- Một chức năng kiểm soát truy cập sẽ bảo đảm rằng người kỹ sư phần mềm đó có thẩm quyền “check out” cái đối tượng đó & các kiểm soát đồng bộ sẽ khoá đối tượng đó trong cơ sở dữ liệu dự án sao cho không thể cập nhật được dữ liệu này cho đến khi phiên bản “check out” hiện thời được thay thế.

## m11.2. Hoạt động kiểm soát sự thay đổi(t)

*Nguyễn Văn Vy*



- Chú ý rằng các bản sao khác có thể được “check out” nhưng không thể được cập nhật.
- Một bản sao của đối tượng đường mốc (gọi là phiên bản trích ly) được cải biên bởi kỹ sư phần mềm. Sau bảo đảm chất lượng phần mềm và kiểm thử thì phiên bản đã được cải biên của đối tượng đó được “check in” cả đối tượng và đường mốc mới này được mở khoá.

## m11.2. Hoạt động kiểm soát sự thay đổi(t)

*Nguyễn Văn Vỹ*



- đường mốc được tạo ra khi đối tượng đó đã được rà soát kỹ thuật chính thức & đã được chấp thuận
- Sau đó kiểm soát đổi thay mức dự án được thực thi.  
Trước hết người phát triển phải được sự chấp thuận của người quản lý dự án (nếu dự án là cục bộ) hoặc được sự chấp thuận của người có thẩm quyền kiểm soát đổi thay (nếu đổi thay ảnh hưởng tới các khoản mục cấu hình phần mềm khác).

## m11.3. Vấn đề trong kiểm soát sự thay đổi

*Nguyễn Văn Vỹ*



- ❑ Quá trình kiểm soát thay đổi với nhiều thủ tục quá chặt chẽ có nguy cơ tạo ra quan liêu .
- ❑ Không tổ chức và kiểm tra tốt, sự kiểm soát thay đổi có thể dẫn đến cản trở tiến trình phát triển
- ❑ Hầu hết các người phát triển phần mềm đều có cơ chế kiểm soát đổi thay (cũng nhiều người chẳng có!), họ đã tạo ra một số tầng kiểm soát để tránh những vấn đề nói trên.

## m11.4. Thực tiễn kiểm soát sự thay đổi

*Nguyễn Văn Vỹ*



- Trong một số trường hợp sự tạo sinh nhiều thủ tục chính thức: các yêu cầu, các báo cáo thay, và các thứ tự đổi thay kỹ thuật là được bỏ qua. Tuy nhiên việc đánh giá từng đổi thay vẫn được tiến hành và tất cả các đổi thay vẫn được theo dõi và được rà soát.
- kiểm soát đổi thay chính thức được hình thành khi sản phẩm đã được phân phát cho khách hàng.
- Thẩm quyền kiểm soát thay đổi đóng một vai trò tích cực trong tầng thứ hai và thứ ba. Phụ thuộc vào kích cỡ và đặc tính của dự án phần mềm

# m11.4. Thực tiễn kiểm soát sự thay đổi(t)

*Nguyễn Văn Vy*



- Người có thẩm quyền cần có cái nhìn tổng thể, đánh giá được ảnh hưởng của thay đổi đối với các yếu tố bên ngoài các khoản mục cấu hình phần mềm:

Thay đổi ảnh hưởng tới:

- phần cứng như thế nào?
- sự thực thi như thế nào?
- sự nhìn nhận của khách hàng đối với sản phẩm như thế nào?
- chất lượng và độ tin cậy của sản phẩm như thế nào?, và nhiều câu hỏi khác nữa

# n. Kiểm toán cấu hình

*Nguyễn Văn Vỹ*



- Minh định, kiểm soát phiên bản, kiểm soát thay đổi, giúp cho người phát triển duy trì một trật tự, tránh được tình thế hỗn độn.
- Tuy nhiên ngay cả các cơ cấu kiểm soát thành công nhất cũng chỉ theo dõi một thay đổi cho đến khi một đối tượng cấu hình kỹ thuật được sinh ra;
- làm thế nào để bảo đảm sự đổi đó thực sự được thực thi? Cần có hai hoạt động:
  - rà soát kỹ thuật chính thức,
  - kiểm toán cấu hình.

## n. Kiểm toán cấu hình (t)

Nguyễn Văn Vỹ



- **Rà soát kỹ thuật chính thức** tập trung vào sự đúng đắn kỹ thuật của đối tượng cấu hình được cải biên.

Người rà soát đánh giá:

- sự phù hợp với các khoản mục cấu hình khác,
  - sự bỏ sót và các hiệu ứng phụ khác.
- **Kiểm toán cấu hình phần mềm** bổ sung cho rà soát kỹ thuật chính thức bằng cách đánh giá một đối tượng cấu hình trên các đặc tính mà thường không được xét trong quá trình rà soát.



## n. Kiểm toán cấu hình (t)

*Nguyễn Văn Vỹ*



- Kiểm toán hỏi và trả lời các câu hỏi sau:
- Thay đổi được đặc tả trong trật tự thay đổi kỹ thuật đã được thực hiện hay chưa?, Nhưng cải biên phụ nào đã được phối hợp?
  - Rà soát kỹ thuật chính thức đã được tiến hành để đánh giá sự đúng đắn kỹ thuật hay chưa?
  - Các chuẩn kỹ nghệ phần mềm đã thực sự được tuân thủ chưa?

## n. Kiểm toán cấu hình (t)

*Nguyễn Văn Vy*



- Kiểm toán hỏi và trả lời các câu hỏi sau (tiếp):
  - Thay đổi đó đã nổi bật lên trong các khoản mục phần mềm chưa? Có đặc tả **ngày & tác giả** của thay đổi đó? **Các thuộc tính** của đối tượng cấu hình đó đã phản ánh được đổi thay đó chưa?
  - Các thủ tục quản lý cấu hình để giám sát, ghi lại và báo cáo về nó có được tuân thủ hay không?
  - Tất cả các khoản mục cấu hình phần mềm liên quan đã thực sự được cập nhật chưa?

# n. Kiểm toán cấu hình (t)

*Nguyễn Văn Vy*



- Trong một vài trường hợp các câu hỏi kiểm toán như là một phần của rà soát kỹ thuật phần mềm; nhưng khi quản lý cấu hình phần mềm là một hoạt động chính thức thì kiểm toán quản lý cấu hình phần mềm được tiến hành tách ra bởi một nhóm bảo đảm chất lượng

## o. Báo cáo hiện trạng

*Nguyễn Văn Vỹ*



- Báo cáo tình trạng cấu hình - CSR - (còn gọi là kiểm toán tình trạng) là một nhiệm vụ quản lý cấu hình phần mềm; nó trả lời các câu hỏi sau:
  - Cái gì đã xảy ra?
  - Ai làm?
  - Xảy ra khi nào?
  - Sẽ còn ảnh hưởng nào khác nữa?
- Thông tin báo cáo tình trạng cấu hình gắn chặt với các nhiệm vụ trong phần kiểm soát đổi thay

## o. Báo cáo hiện trạng(t)

*Nguyễn Văn Vỹ*



- Mỗi khi một khoản mục cấu hình phần mềm được án định mới hoặc được cập nhật chứng thư thì một **nội dung (entry)** được tạo ra.
- Mỗi khi một khoản mục cấu hình phần mềm được chấp thuận bởi thẩm quyền thay đổi cấu hình (có một mệnh lệnh thay đổi kỹ nghệ được phát ra) thì một **nội dung (entry)** được tạo ra
- Mỗi khi một kiểm toán cấu hình được tiến hành thì các kết quả của nó được báo cáo như là một phần của nhiệm vụ báo cáo trạng thái cầu hình.

## o. Báo cáo hiện trạng(t)

*Nguyễn Văn Vy*



- Đầu ra của báo cáo tình trạng cấu hình có thể được đặt trên một cơ sở dữ liệu trực tuyến sao cho những người phát triển phần mềm hay bảo trì có thể truy cập các thông tin đổi thay.
- Báo cáo tình trạng cấu hình phần mềm đóng một vai trò cốt tử trong thắng lợi của một dự án phát triển phần mềm lớn.
- Các người phát triển phần mềm có thể cố gắng cải biên cùng một khoản mục cấu hình với những ý định khác nhau, thậm chí là mâu thuẫn nhau.

## o. Báo cáo hiện trạng(t)

*Nguyễn Văn Vỹ*



- Báo cáo tình trạng cấu hình giúp ta loại trừ vấn đề bất cập liên quan nhiều người bằng cách cải thiện giao tiếp giữa tất cả họ
- Vấn đề bất cập:
  - Đội kỹ nghệ phần mềm có thể tốn hàng tháng trời xây dựng phần mềm cho một đặc tả phần cứng đã không dùng nữa.
  - Một người nhận ra các hiệu ứng phụ nghiêm trọng do một đổi thay được đề xuất đã không ý thức được rằng nó đã được thực hiện.