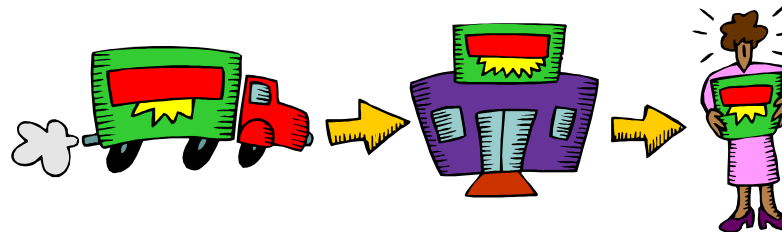


# Công nghệ phần mềm

## Tổng quan về kiểm thử và đảm bảo chất lượng phần mềm



# Nội dung

---

- Kiểm chứng và thẩm định
- Đảm bảo chất lượng
- Quy trình kiểm thử
- Ca kiểm thử
- Mô tả bài toán kiểm thử qua biểu đồ Venn
- Kiểm thử hộp đen và kiểm thử hộp trắng
- Một số công cụ hỗ trợ kiểm thử

# Yêu cầu khách hàng và đặc tả yêu cầu

---

- **PM được phát triển để phục vụ yêu cầu KH**
- **Yêu cầu khách hàng được biểu diễn bằng đặc tả yêu cầu**
- **Thất bại = PM không đáp ứng đúng như đặc tả**
  - Đặc tả sai?
  - Thiết kế sai?
  - Cài đặt sai?

# Verification and Validation

---

- **Verification (kiểm chứng)**
  - Kiểm tra sản phẩm có được cài đặt đúng thiết kế không?
  - Phát hiện lỗi lập trình so với thiết kế
- **Validation (Thẩm định)**
  - Kiểm tra xem sản phẩm có đáp ứng yêu cầu KH không? (chức năng và phi chức năng)
  - Tìm lỗi phân tích thiết kế
- **Verification -> Validation (V&V)**

# V&V: tĩnh và động

---

- V&V tĩnh:
  - Không thực thi/chạy chương trình
  - Xét duyệt yêu cầu, thiết kế, mã nguồn
  - Tiến hành ở mọi giai đoạn phát triển PM
  - Khó đánh giá tính hiệu quả của PM
- V&V động (Kiểm thử PM)
  - Thực thi/chạy chương trình
  - Là cách duy kiểm tra các yêu cầu phi chức năng

# Chất lượng và độ tin cậy

---

- **Chất lượng = sự thỏa mãn của sp so với đặc tả**
- **Chất lượng PM = “độ tốt, độ tuyệt hảo”**
  - Tính đúng đắn (đúng đặc tả)
  - Tính hiệu quả
  - Độ tin cậy
  - Khả kiểm thử
  - Dễ học, dễ sử dụng
  - Dễ bảo trì
  - ...
- **Độ tin cậy chỉ là một yếu tố để đánh giá chất lượng SP**
  - Là độ đo quan trọng

# Kiểm thử PM

---

- Là hoạt động chủ chốt nhằm đánh giá chất lượng
- Có thể chỉ ra lỗi, không thể khẳng định không còn lỗi
  - Có thể khẳng định hết lỗi bằng kiểm thử vét cạn, nhưng cách này không khả thi trên thực tế
- Một kiểm thử thành công là một kiểm thử phát hiện ra lỗi

# Các hoạt động kiểm thử

---





# Ca kiểm thử (test case)

---

Định danh (tên) của ca kiểm thử

Mục đích

Tiền điều kiện

Đầu vào

Đầu ra mong đợi

Hậu điều kiện

Lịch sử thực hiện ca kiểm thử:

*Ngày    Kết quả thực tế    Phiên bản    Kiểm thử viên*

# Bộ kiểm thử (các ca kiểm thử)

---

TC_id	Inputs	EO	Note
tc1	1 2 3	5	
tc2	4 5 8	20	
tc3	4 6 2	17	
...			

# Báo cáo kiểm thử

---

TC_id	Inputs	EO	Result
tc1	1 2 3	5	Passed
tc2	4 5 8	20	Failed
tc3	4 6 2	17	Failed
...			...

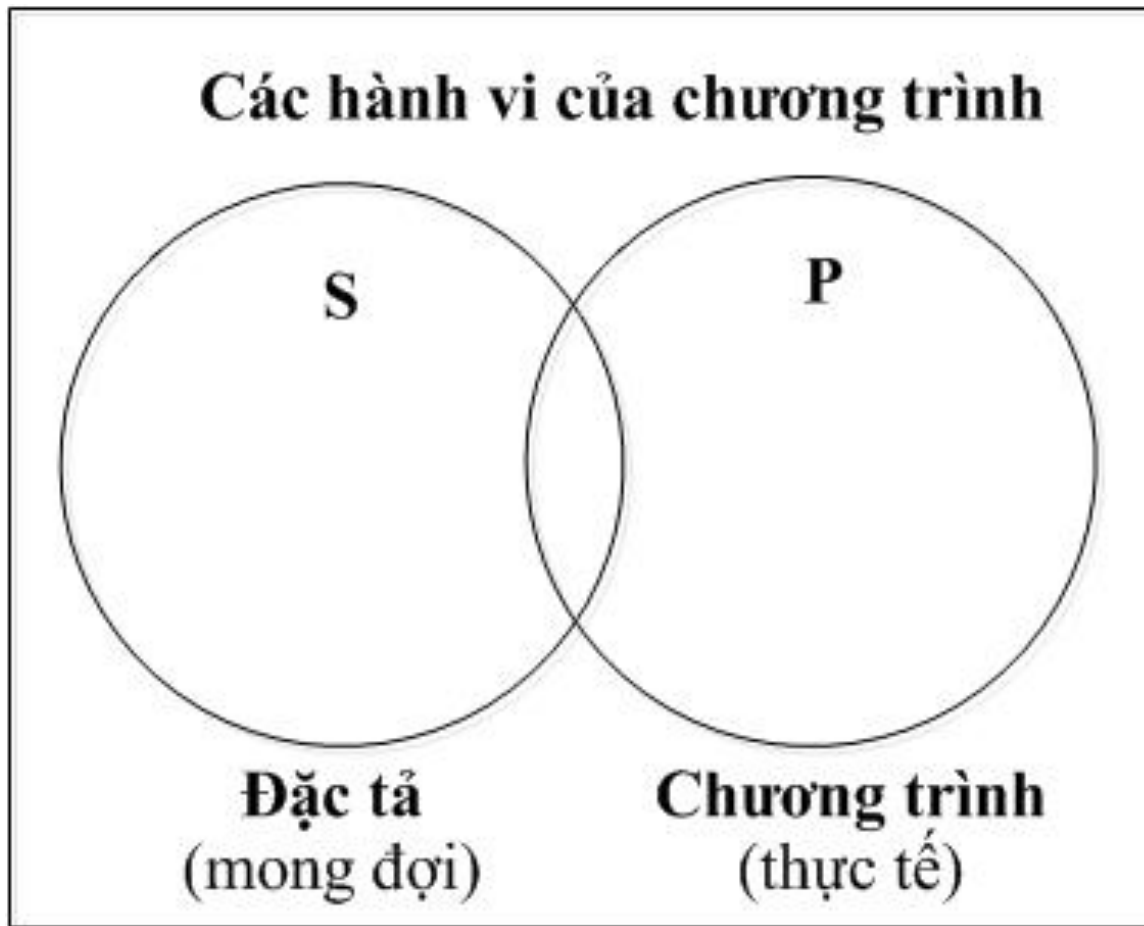
# Bộ kiểm thử tốt

---

- Chạy các ca kiểm thử với chương trình P
  - Bao phủ một số yêu cầu của P;
  - Bao phủ một phần chức năng của P
  - Bao phủ một phần trong cấu trúc của P
- => Tiêu chuẩn bao phủ sẽ định hướng thiết kế các ca kiểm thử
- Bộ kiểm thử tốt: đạt 100% tiêu chuẩn bao phủ (cho trước)

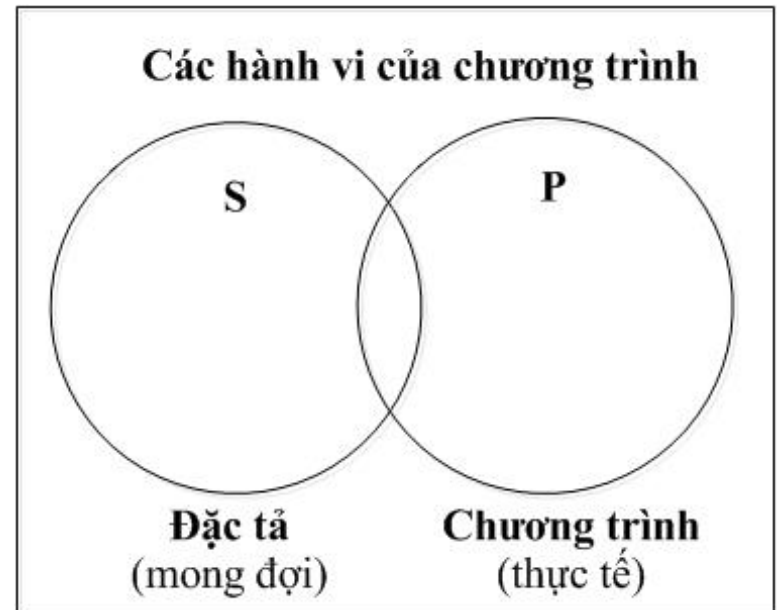
# Mô tả bài toán kiểm thử qua biểu đồ Venn

---



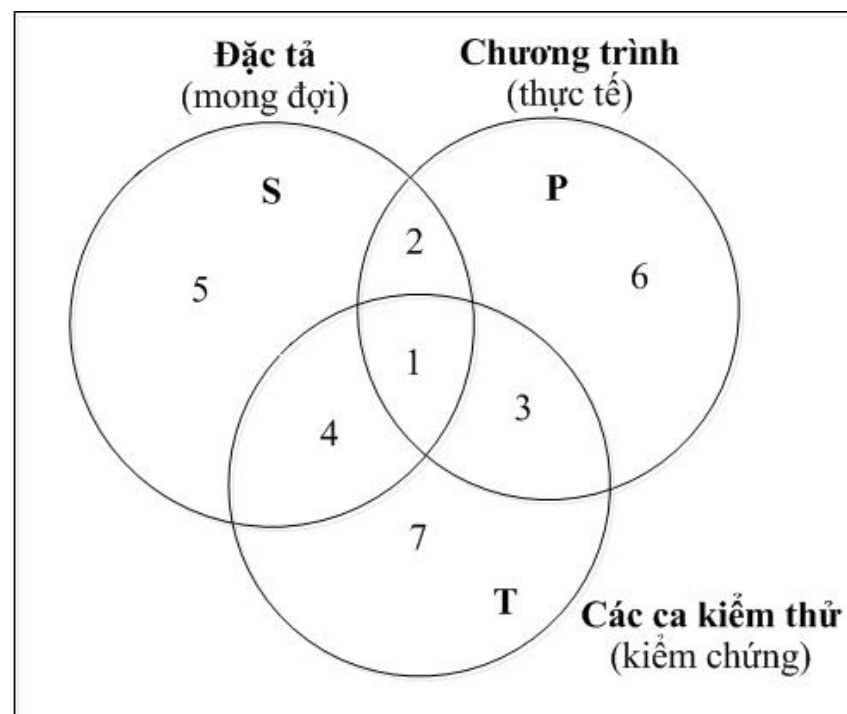
# Bài toán cần đối mặt của người kiểm thử

- Hành vi được đặc tả nhưng không được lập trình
  - sai lầm về bỏ quên
- Hành vi được lập trình mà không được đặc tả
  - Sai lầm về nhiệm vụ
- Giao giữa S và P là phần đúng đắn



# Bài toán cần đổi mặt của người kiểm thử

- 1:
- 2:
- 3:
- 4:
- 5:
- 6:
- 7:



# Kiểm thử hộp đen

---

- Còn gọi là kiểm thử hàm, kiểm thử chức năng
- Tập trung vào hành vi vào/ra. Với đầu vào đã biết ra có thể đoán/tính đầu ra, rồi kiểm tra chương trình có tạo kết quả như ta đoán/tính.
  - Không thể kiểm thử hết các bộ dữ liệu đầu vào
- Bài toán đặt ra là giảm số lượng ca kiểm thử bằng việc chia không gian đầu vào thành các miền tương đương
  - Sau đó chọn một ca kiểm thử từ mỗi miền tương đương này.



# Kiểm thử hộp trắng

---

- Còn gọi là kiểm thử cấu trúc, kiểm thử logic
- Các tiêu chuẩn bao phủ
  - Dòng lệnh
    - Mọi lệnh đều được thử
  - Nhánh (if, while, ..)
    - Biểu thức điều kiện được thử với cả True và False
      - Các nhánh đều được chạy ít nhất một lần
  - Đường đi
    - Tất cả các khả năng chạy của chương trình
  - Vòng lặp
    - 0, 1, >1 lần

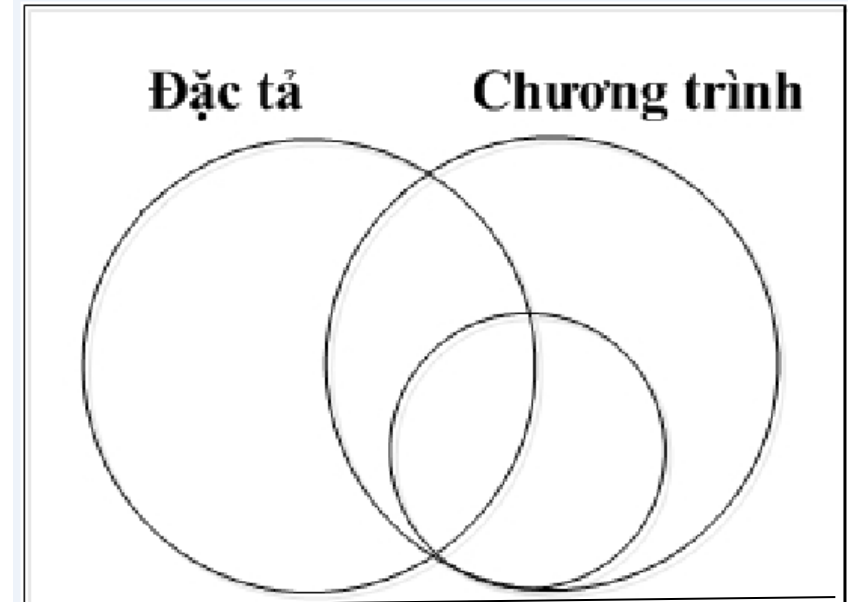
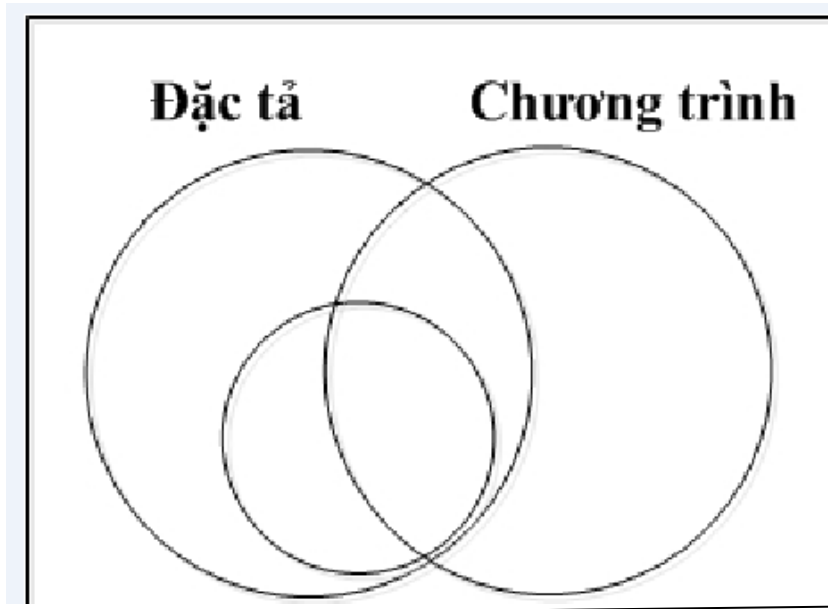
# So sánh kiểm thử hộp trắng và hộp đen

---

- Hộp trắng
  - Số đường đi nhiều khi là vô hạn
  - Kiểm tra những gì đã làm, không phải những gì cần được làm
  - Không thích hợp cho kiểm thử hệ thống và tích hợp
- Hộp đen
  - Dễ bùng nổ tổ hợp về số ca kiểm thử (dữ liệu đúng và dữ liệu sai)
  - Thường không chắc ca kiểm thử này có phát hiện được lỗi cụ thể hay không
  - Thích hợp cho tất cả các cấp độ kiểm thử
- Cần cả hai
- Kiểm thử hộp trắng và hộp đen là hai thái cực của kiểm thử
- Việc lựa chọn ca kiểm thử nằm giữa và phụ thuộc vào
  - Số đường đi logic có thể
  - Tính chất của dữ liệu đầu vào
  - Khối lượng tính toán
  - Độ phức tạp của cấu trúc dữ liệu và giải thuật
- Hai chiến lược là bổ sung cho nhau.

# So sánh kiểm thử hộp trắng và hộp đen

---



# Kiểm thử và gỡ lỗi (debugging)

---

- Kiểm thử
  - Khẳng định có lỗi
- Gỡ lỗi (debugging)
  - Định vị và sửa lỗi
- Gỡ lỗi thông thường phải lập giả thuyết về hành vi của chương trình và kiểm tra các giả thuyết này để tìm lỗi

# Phân loại các lỗi và sai

---

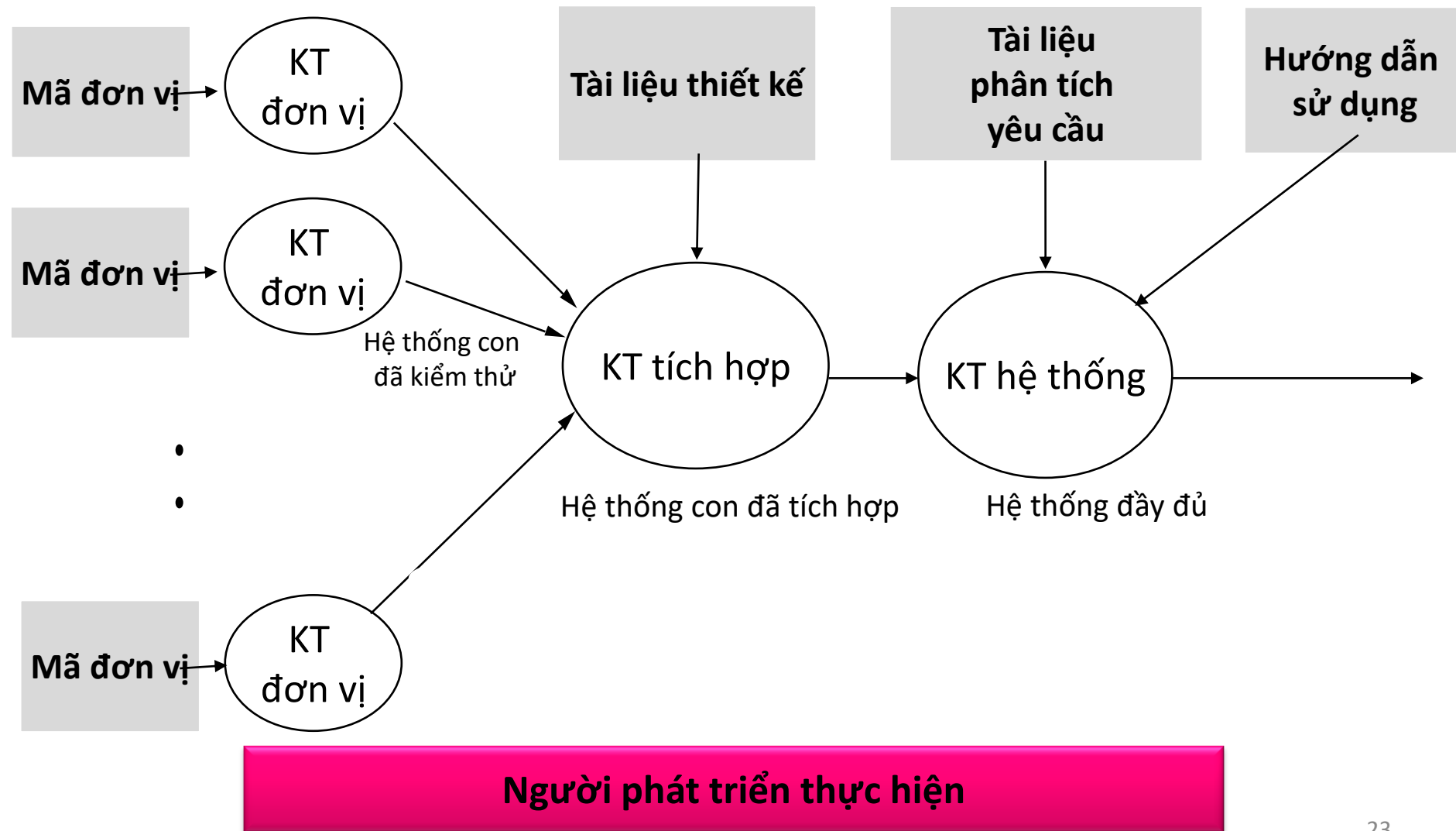
1	Nhẹ	Lỗi chính tả
2	Vừa	Hiểu lầm hoặc thừa thông tin
3	Khó chịu	Tên bị thiếu, cắt chữ hoặc hóa đơn có giá trị 0.0 đồng
4	Bực mình	Vài giao dịch không được xử lý
5	Nghiêm trọng	Mất giao dịch
6	Rất nghiêm trọng	Xử lý giao dịch sai
7	Cực kỳ nghiêm trọng	Lỗi rất nghiêm trọng xảy ra thường xuyên
8	Quá quất	Hủy hoại cơ sở dữ liệu
9	Thảm họa	Hệ thống bị tắt
10	Dịch họa	Thảm họa lây lan

# Các mức kiểm thử

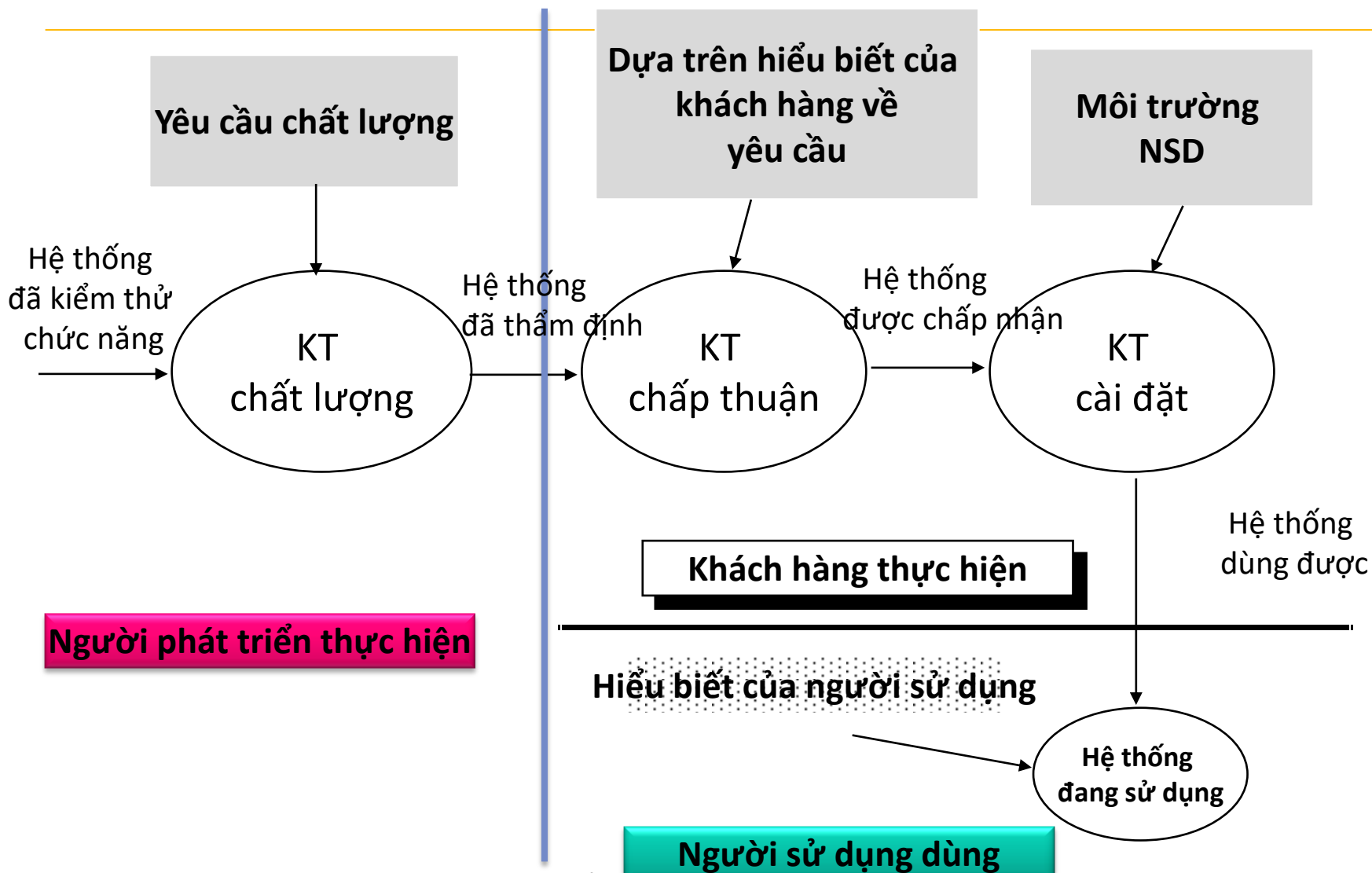
---

- Đơn vị
  - Tìm lỗi trong từng đơn vị
- Tích hợp
  - Tìm lỗi khi ghép các đơn vị
- Hệ thống
  - Tìm lỗi khi hệ thống đã tích hợp xong, trước khi phát hành, chuyển giao
- Chấp thuận
  - Người sử dụng dùng thử xem hệ thống đáp ứng đúng mong muốn chưa.
  - Còn gọi là kiểm thử alpha.

# Các mức kiểm thử



# Các mức kiểm thử (tiếp)



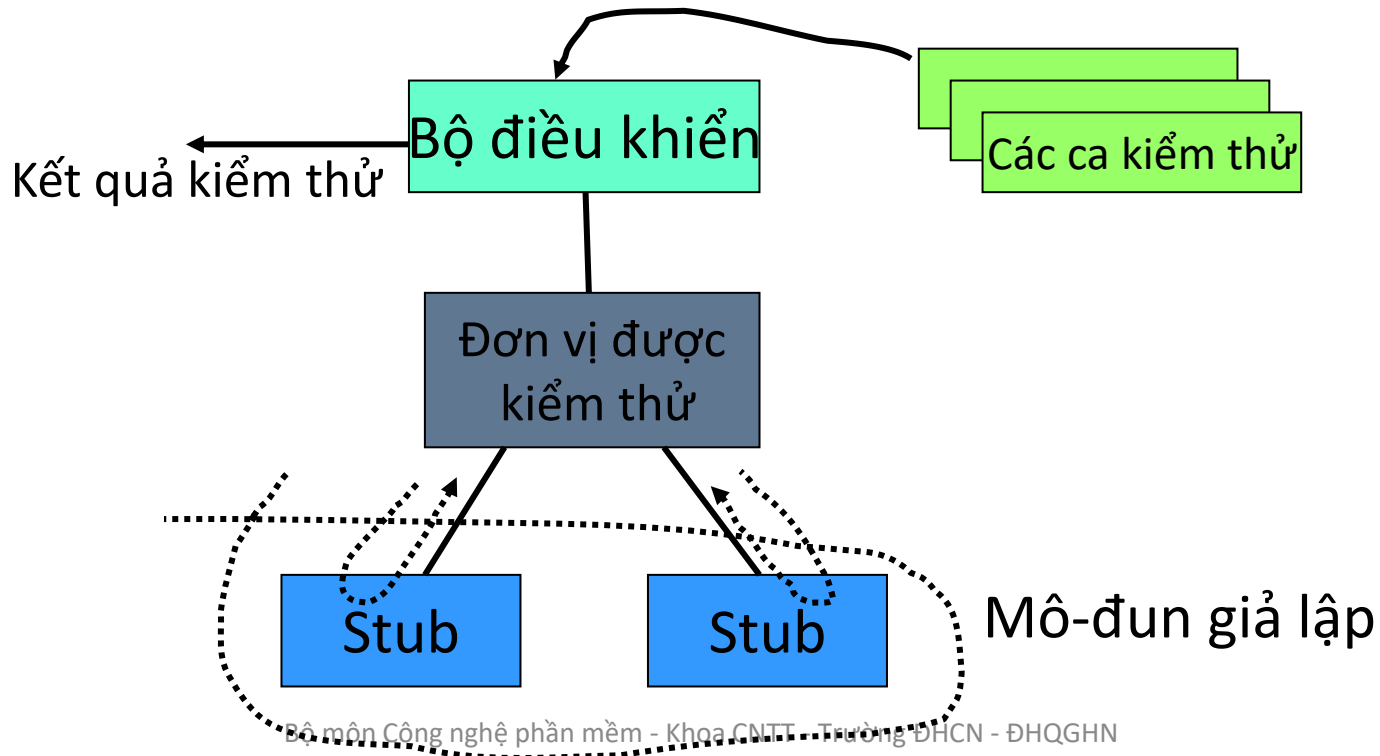


# Kiểm thử đơn vị

**Mục đích:** Tìm sự khác biệt giữa đặc tả và cài đặt của đơn vị

**Đơn vị:** các lớp, hàm, đối tượng, gói, mô-đun

**Môi trường kiểm thử đơn vị:**



# Kiểm thử tích hợp

---

- Mục tiêu:
  - Phát hiện vấn đề khi ghép các mô-đun/thành phần với nhau
- Các vấn đề
  - Bên trong: giữa các thành phần
    - Gọi: call/message passing/...
    - Tham số: kiểu, số lượng, thứ tự, giá trị
    - Kết quả trả về: ai, kiểu, trình tự
  - Bên ngoài:
    - Ngắt (wrong handler?)
    - Thời gian vào ra
  - Tương tác

# Kiểm thử hệ thống

---

- Liên quan đến các yếu tố bên ngoài hệ thống
- Không chỉ là kiểm tra chức năng
  - Khả dụng (usability)
    - Giao diện, thông báo, dễ học, dễ nhớ..
  - Hiệu năng
    - Khả năng đáp ứng/Tìm khả năng đáp ứng
  - Tài nguyên sử dụng

# Kiểm thử chấp thuận

---

- Có hai loại kiểm thử chấp nhận
  - Bởi cơ quan phát triển gọi là BAT
  - Bởi người dùng gọi là UAT
- Mục đích: kiểm tra sự hài lòng của người sử dụng
- Cơ sở: mong muốn của người dùng (không xét đến tài liệu đặc tả)
- Môi trường: thật
- Người thực hiện: bởi và cho người sử dụng
- Các ca kiểm thử:
  - Sử dụng lại từ kiểm thử hệ thống
  - Do người dùng thiết kế

# Khi nào nên dừng kiểm thử

---

- Hết thời gian, hết ngân sách
- Đạt mức độ bao phủ mong muốn
- Đạt tần suất hỏng hóc mong muốn

# Kiểm thử hồi qui

---

- Khi một hệ thống được chỉnh sửa (sửa lỗi, thêm/bớt chức năng,..) toàn bộ bộ kiểm thử cần phải chạy lại
  - Đảm bảo các tính năng đang hoạt động tốt không bị ảnh hưởng bởi chỉnh sửa
- Kiểm thử lại tự động trước khi lưu thay đổi vào kho (repo.)
- Cần các chiến lược kiểm thử tăng dần với hệ thống lớn

# Nhiều công cụ hỗ trợ các loại kiểm thử

---

- Kiểm thử đơn vị: **Achoo, JUnit, Pex/Moles, PyUnit, ...**
- Tự động kiểm thử: **TestComplete, ...**
- Kiểm thử hiệu năng và tải: **JMeter, ...**
- Kiểm thử giao diện: **Ranorex, Abbot, Guitar, ...**
- Kiểm thử tổ hợp: **AETG, FireEye, ...**
- Kiểm thử dựa trên mô hình: **Spec Explorer, ...**
- Phân tích bao phủ: **Corbertura, ...**
- Quản lý lỗi (defects): **Bugzilla, ...**
- ...

# Tổng kết

---

- Kiểm chứng và thẩm định
- Đảm bảo chất lượng
- Quy trình kiểm thử
- Ca kiểm thử
- Mô tả bài toán kiểm thử qua biểu đồ Venn
- Kiểm thử hộp đen và kiểm thử hộp trắng
- Một số công cụ hỗ trợ kiểm thử