

Bài 1 – Tổng quan

Công cụ kiểm thử phần mềm



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Nội dung

- ☐ **Kiểm thử phần mềm là gì?**
- ☐ Tại sao kiểm thử quan trọng?
- ☐ Quy trình kiểm thử phần mềm
- ☐ Vai trò và thái độ
- ☐ Các cấp độ kiểm thử
- ☐ Các loại kiểm thử

Kiểm thử phần mềm là gì?

- Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm lỗi

The Art of Software Testing

- Là hoạt động kiểm tra xem phần mềm có chạy chính xác hay không (Verification) và có thoả mãn yêu cầu của khách hàng hay không (Validation) nhằm hướng tới mục tiêu Chất lượng cho phần mềm.

Xác minh và thẩm định

☐ Xác minh (Verification)

- ☐ Có đúng đặc tả, có đúng thiết kế

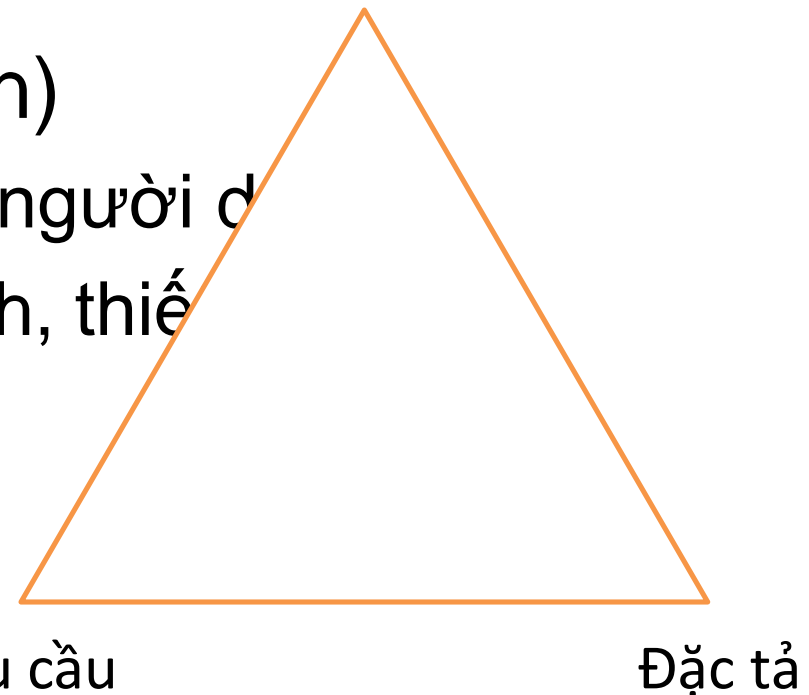
- ☐ Phát hiện lỗi lập trình

Phần mềm

☐ Thẩm định (Validation)

- ☐ Có đáp ứng nhu cầu người dùng

- ☐ Phát hiện lỗi phân tích, thiết kế



Xác minh và thẩm định

- V & V = Verification and Validation
 - Mục tiêu là phát hiện và sửa lỗi phần mềm, đánh giá tính đúng đắn được của phần mềm
 - Thứ tự thực hiện: Verification → Validation
 - Verification chiếm 80%, Validation chiếm 20% công việc
- Validation tác động 80% hiệu quả chung

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ **Tại sao kiểm thử quan trọng?**
- ☐ Quy trình kiểm thử phần mềm
- ☐ Vai trò và thái độ
- ☐ Các cấp độ kiểm thử
- ☐ Các loại kiểm thử

Tại sao có lỗi?

- Phần mềm viết bởi con người
 - Biết nhiều thứ, nhưng không phải mọi thứ
 - Có kỹ năng, nhưng không hoàn hảo
 - Luôn phạm sai lầm
- Làm việc dưới điều kiện căng thẳng để kịp bàn giao đúng tiến độ
 - Không có thời gian kiểm tra, giả định bị sai
 - Hệ thống chưa hoàn chỉnh

Chi phí lỗi

☐ Có thể rất lớn:

- ☐ Ariane 5: 7 tỉ đô la,

- ☐ Mariner space probe to Venus: 250 triệu đô la,

- ☐ American Airlines: 50 triệu đô la

☐ Có thể gây chết người:

- ☐ Therac-25

- ☐ Airbus & Korean Airlines

☐ Có thể không đáng kể:

- ☐ Lợi bất tiện, ảnh hưởng không nhìn thấy được

Tại sao kiểm thử cần thiết?

☐ Vì:

- ☐ Phần mềm luôn tồn tại lỗi
- ☐ Đánh giá độ tin cậy
- ☐ Chi phí lỗi có thể rất cao
- ☐ Tránh bị kiện từ khách hàng
- ☐ Giữ uy tín trong kinh doanh

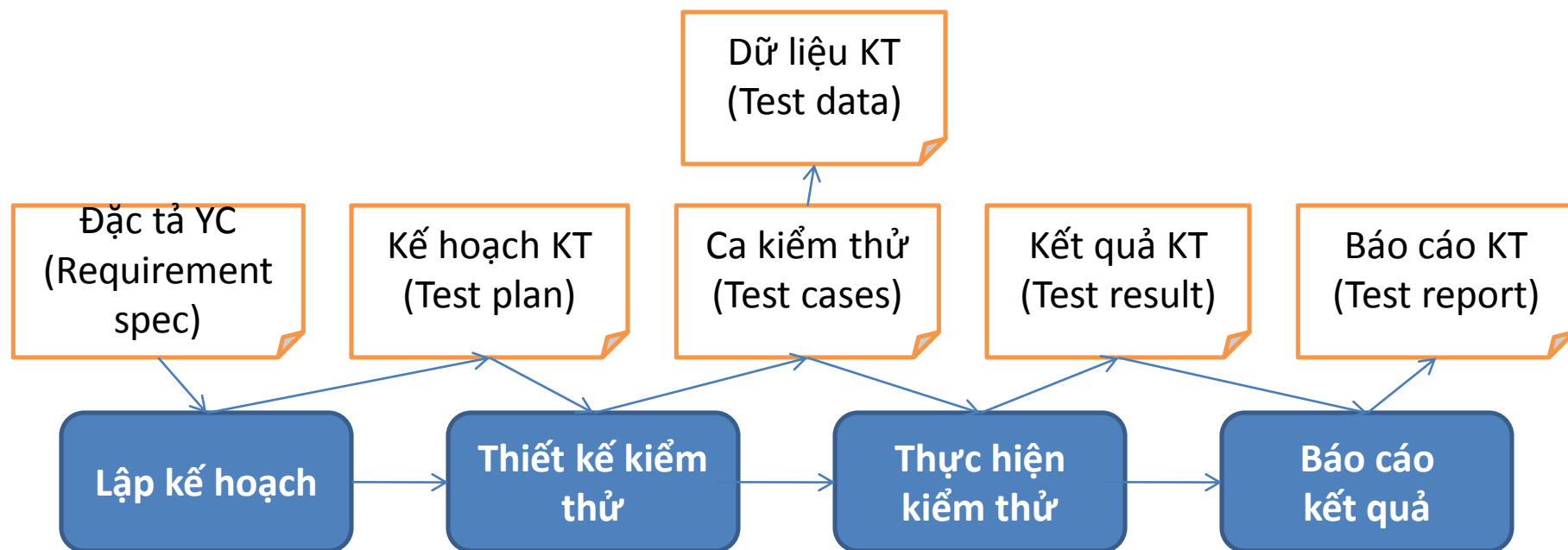
☐ Không vì:

- ☐ Lấp khoảng thời gian giữa ngày hoàn thành và ngày bàn giao
- ☐ Chứng minh là phần mềm không lỗi
- ☐ Kiểm thử là một phần của kế hoạch dự án

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ Tại sao kiểm thử quan trọng?
- ☐ **Quy trình kiểm thử phần mềm**
- ☐ Vai trò và thái độ
- ☐ Các cấp độ kiểm thử
- ☐ Các loại kiểm thử

Quy trình kiểm thử



Lập kế hoạch

- ☐ Mục đích: chỉ định, mô tả các chiến lược kiểm thử
- ☐ Kết quả: bản kế hoạch kiểm thử (Test plan)
- ☐ Nội dung TestPlan:
 - ☐ Giới thiệu
 - ☐ Yêu cầu
 - ☐ Chiến lược
 - ☐ Thời gian
 - ☐ Tài nguyên

Thiết kế

- Mục đích: bảo đảm tất cả các tình huống kiểm tra “quét” hết tất cả yêu cầu cần kiểm tra
- Kết quả: ca kiểm thử (Test cases), dữ liệu kiểm thử (Test data)
- Nội dung Testcase:
 - Dữ liệu kiểm thử
 - Các bước thực hiện
 - Đầu ra mong đợi

Thực hiện

- Mục đích: thực hiện các ca kiểm thử, ghi nhận kết quả
- Kết quả: bảng báo cáo (Test result) là kết quả thực tế của việc chạy testcase trên phần mềm

Báo cáo kiểm thử

- Mục đích: xem xét và đánh giá kết quả kiểm tra, liệt kê lỗi, chỉ định các yêu cầu thay đổi, thống kê số liệu
- Kết quả: báo cáo kiểm thử (Test report), có 2 loại
 - ▣ Test summary report: báo cáo thống kê
 - ▣ Bug report: báo cáo lỗi

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ Tại sao kiểm thử quan trọng?
- ☐ Qui trình kiểm thử phần mềm
- ☐ **Vai trò và thái độ**
- ☐ Các cấp độ kiểm thử
- ☐ Các loại kiểm thử

Vai trò QA, QC

☐ QC – Quality Control

- ☐ Những hoạt động, những kỹ thuật nhằm đảm bảo chất lượng sản phẩm.

☐ QA – Quality Assurance

- ☐ Những kế hoạch, những hoạt động mang tính hệ thống nhằm đảm bảo quá trình sản xuất sẽ tạo ra những sản phẩm có chất lượng.

Định nghĩa của ISO 9000

Công cụ kiểm thử phần mềm

Vai trò QA, QC

☐ QC

- ☐ Sản phẩm
- ☐ Phản ứng
- ☐ Tìm lỗi

☐ Ví dụ

- ☐ Kiểm duyệt
- ☐ Kiểm thử
- ☐ Thanh tra
- ☐ Kiểm tra lại

☐ QA

- ☐ Tiến trình
- ☐ Tiên đoán, ước tính
- ☐ Ngăn ngừa lỗi

☐ Ví dụ

- ☐ Đảm bảo chất lượng
- ☐ Định nghĩa tiến trình
- ☐ Chọn lựa công cụ
- ☐ Huấn luyện

Thái độ của Tester

- Cẩn thận (Cautious)
 - ▣ Phỏng đoán chứ không kết luận
 - ▣ Tập thừa nhận “Tôi không biết”
 - ▣ Có người khác kiểm tra lại
- Tò mò (Curious)
 - ▣ Điều gì xảy ra nếu ...?
 - ▣ Nó hoạt động như thế nào?
 - ▣ Tại sao nó xảy ra?
- Chỉ trích, phê phán (Critical)
 - ▣ Tiến hành phỏng đoán và bác bỏ
 - ▣ Tích cực tìm kiếm phản chứng
- Can đảm (Courageous)

Good testers are
hard to fool.

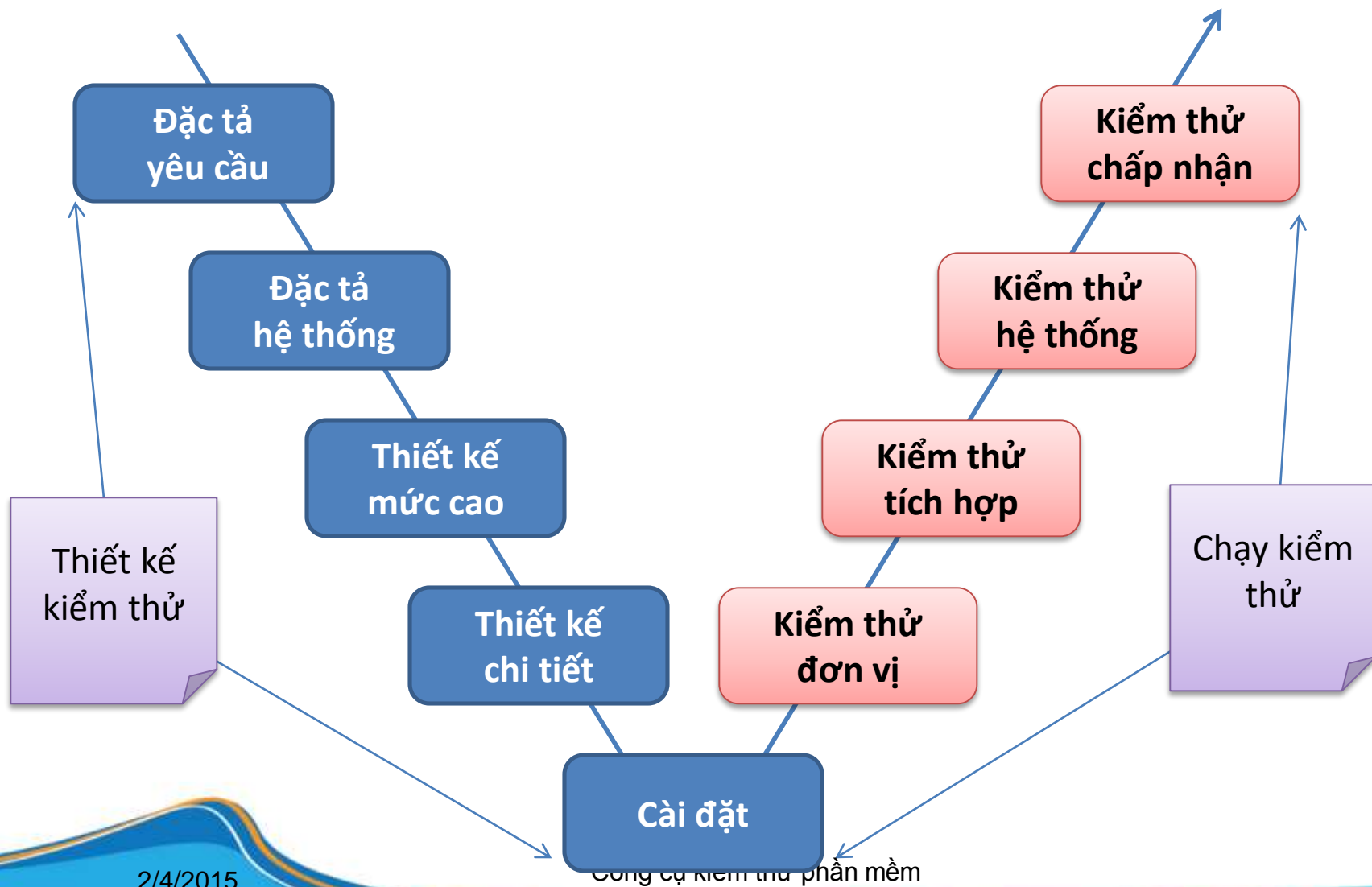
Kỹ năng của Tester

- ☐ Kỹ năng giao tiếp
- ☐ Kỹ năng đọc
- ☐ Kỹ năng giải quyết vấn đề
- ☐ Kỹ năng lập báo cáo
- ☐ Kỹ năng quản lý bản thân
- ☐ Kỹ năng ngoại ngữ

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ Tại sao kiểm thử quan trọng?
- ☐ Qui trình kiểm thử phần mềm
- ☐ Vai trò và thái độ
- ☐ **Các cấp độ kiểm thử**
- ☐ Các loại kiểm thử

Mô hình chữ V



Kiểm thử đơn vị – Unit testing

- ☐ Component/module/program testing
- ☐ Mỗi đơn vị được kiểm thử độc lập, trước khi tích hợp
- ☐ Mức thấp nhất và cụ thể, chi tiết nhất
- ☐ Mục tiêu:
 - ☐ Đảm bảo mã nguồn từng đơn vị đúng theo đặc tả
 - ☐ Bao gồm chức năng và phi chức năng
- ☐ Dựa trên:
 - ☐ Yêu cầu
 - ☐ Thiết kế đơn vị
 - ☐ Mã nguồn

Kiểm thử đơn vị – Unit testing

- ☐ Ai thực hiện?
 - ☐ Lập trình viên
- ☐ Báo cáo
 - ☐ Lỗi được sửa ngay, không cần báo cáo
- ☐ Công cụ
 - ☐ Viết trực tiếp mã nguồn
 - ☐ Unit test framework
 - ☐ Mocking framework
 - ☐ Dependency Injection and IoC containers

Kiểm thử đơn vị – Unit testing

- Test driven development
 - Test-first approach
 - Hướng tiếp cận phát triển phần mềm dựa trên Unit Test
 - Chuẩn bị và tự động hóa test case trước khi coding
 - Lập trình từng phần một → tất cả test case điều đạt

Kiểm thử tích hợp – Integration testing

- ☐ Kiểm tra hơn 2 đơn vị/hệ thống
- ☐ Mục tiêu:
 - ☐ Kiểm thử giao diện/sự tương tác giữa các đơn vị/hệ thống
 - ☐ Kiểm thử các tập không hoạt động độc lập
 - ☐ Kiểm thử chức năng và phi chức năng
- ☐ Dựa trên:
 - ☐ Thiết kế phần mềm
 - ☐ Kiến trúc phần mềm
 - ☐ Workflows/Use-cases



Kiểm thử tích hợp – Integration testing

- ☐ Hai cấp độ
 - ☐ Kiểm thử tích hợp đơn vị
 - ☐ Kiểm thử tích hợp hệ thống
- ☐ Ai thực hiện?
 - ☐ Người phát triển
 - ☐ Người thiết kế
 - ☐ Người kiểm thử độc lập

Kiểm thử hệ thống – System testing

- ☐ Là bước cuối cùng của kiểm thử tích hợp
- ☐ Kiểm thử hệ thống như một tổng thể
- ☐ Mục tiêu:
 - ☐ Phát hiện sai sót trong toàn bộ hệ thống chạy trên môi trường
 - ☐ Kiểm thử chức năng và phi chức năng
- ☐ Dựa trên:
 - ☐ Đặc tả yêu cầu phần mềm
 - ☐ Use case
 - ☐ Tài liệu hướng dẫn sử dụng



Kiểm thử hệ thống – System testing

- Ai thực hiện?

 - Thường và nên nhóm kiểm thử độc lập

- Phân loại

 - Kiểm thử chức năng – Functional testing

 - Kiểm thử phi chức năng – Non-functional testing

Kiểm thử chấp nhận – Acceptance testing

- ☐ Bước cuối cùng của validation
- ☐ Mục tiêu:
 - ☐ Xác nhận từ phía người dùng hệ thống đáp ứng đúng mong đợi của người dùng
- ☐ Dựa trên
 - ☐ Đặc tả yêu cầu
- ☐ Ai thực hiện?
 - ☐ Khách hàng/Người sử dụng
 - ☐ Có thể bao gồm kiểm thử viên



Kiểm thử chấp nhận – Acceptance testing

□ Alpha testing và Beta testing

□ Giống

- Khi phần mềm đã ổn định
- Nhận phản hồi về lỗi, mong đợi, đề xuất

□ Khác

- Alpha testing thực hiện tại môi trường phát triển
- Beta testing thực hiện tại môi trường thực tế

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ Tại sao kiểm thử quan trọng?
- ☐ Quy trình kiểm thử phần mềm
- ☐ Vai trò và thái độ
- ☐ Các cấp độ kiểm thử
- ☐ **Các loại kiểm thử**

Các loại kiểm thử

- Kiểm thử chức năng
 - ▣ Functional testing/Black-box testing
- Kiểm thử phi chức năng
 - ▣ Non-functional testing
- Kiểm thử cấu trúc
 - ▣ Structural testing/White-box testing
- Kiểm thử liên quan thay đổi
 - ▣ Confirmation testing/Re-testing & Regression testing

Kiểm thử chức năng

- ☐ Functional testing/Black-box testing
- ☐ Dựa trên đặc tả chức năng
- ☐ Phát hiện sai sót về chức năng
- ☐ Không quan tâm đến cách cài đặt

Kiểm thử chức năng

- Các kỹ thuật thiết kế
 - ▣ Phân hoạch tương đương (Equivalence partitioning)
 - ▣ Phân tích giá trị biên (Boundary value analysis)

Kiểm thử phi chức năng

- ☐ Kiểm thử hiệu năng – Performance testing
- ☐ Kiểm thử tính tiện dụng – Usability testing
- ☐ Kiểm thử bảo mật – Security testing
- ☐ Kiểm thử cấu hình/cài đặt – Configuration/Installation testing
- ☐ Kiểm thử sao lưu/khôi phục – Back-up/Recovery testing

Kiểm thử phi chức năng

- Kiểm thử hiệu năng – Performance testing
 - Kiểm thử khối lượng – Volume testing
 - Kiểm tra khả năng xử lý dữ liệu lớn của hệ thống
 - Kiểm thử tải/quá tải – Load/Stress testing
 - Kiểm tra yêu cầu về thời gian đáp ứng của hệ thống

Kiểm thử phi chức năng

- Kiểm thử tính tiện dụng – Usability testing
 - Dễ học, sử dụng đơn giản
 - Hiệu quả khi sử dụng
 - Giao diện đơn giản, đồng nhất
 - Hỗ trợ thông tin phản hồi
 - Ngăn ngừa lỗi
 - Liên kết tắt
 - Thông điệp báo lỗi tốt
 - ...

Kiểm thử phi chức năng

- Kiểm thử bảo mật – Security testing
 - Kiểm tra tính hợp lệ của việc truy xuất trong và ngoài chương trình

Kiểm thử phi chức năng

□ Kiểm thử cấu hình/cài đặt – Configuration/Installation testing

□ Kiểm tra cấu hình

- Phần cứng, môi trường phần mềm khác nhau
- Cấu hình bản thân phần mềm
- Độ độ nâng cấp phiên bản

□ Kiểm tra cài đặt

- Gói cài đặt (CD, mạng, ...)
- Uninstall

Kiểm thử phi chức năng

- Kiểm thử sao lưu/khôi phục – Back-up/Recovery testing
 - Kiểm tra khả năng sao lưu và khôi phục hệ thống từ sự cố

Kiểm thử cấu trúc

- ☐ Có nghiên cứu mã nguồn
- ☐ Phân tích thứ tự thực hiện các lệnh

Kiểm thử liên quan thay đổi

- ☐ Kiểm tra sau khi lỗi được sửa chữa
- ☐ Kiểm thử lại – Re-testing/Confirmation testing
 - ☐ Kiểm tra lại chính xác trường hợp kiểm thử đã phát hiện ra lỗi
 - ☐ Xác nhận lỗi đã được sửa chữa
 - Không bảo đảm lỗi mới không phát sinh
- ☐ Kiểm thử hồi qui – Regression testing
 - ☐ Kiểm tra lại tất cả các trường hợp kiểm thử đã thỏa trước đó
 - ☐ Tìm ra các lỗi mới phát sinh

Thảo luận

