

Kiểm thử phần mềm

Tổng quan



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
- ☐ Các nguyên lý tổng quát
- ☐ Vai trò, thái độ và kỹ năng

Kiểm thử phần mềm là gì?

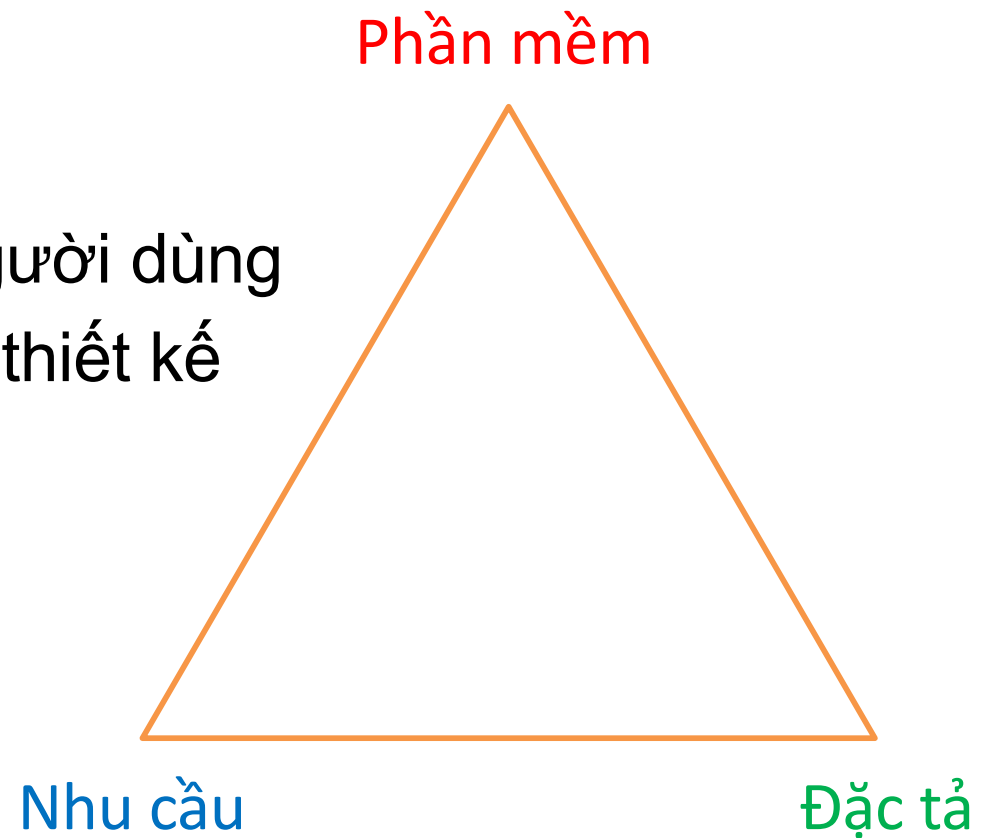
- Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm lỗi

The Art of Software Testing

- Là quá trình 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 kết
 - ▣ Phát hiện lỗi lập trình
- 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
- ☐ Các nguyên lý tổng quát
- ☐ Vai trò, thái độ và kỹ năng

Tại sao kiểm thử cần thiết?

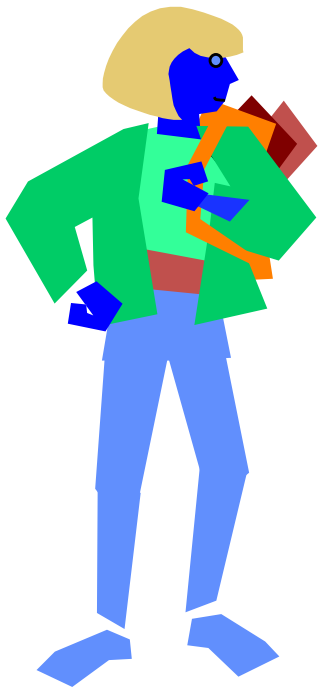
- ☐ Phần mềm luôn tồn tại lỗi
- ☐ Đánh giá độ tin cậy
- ☐ Chi phí lỗi có thể rất cao

Lỗi phần mềm

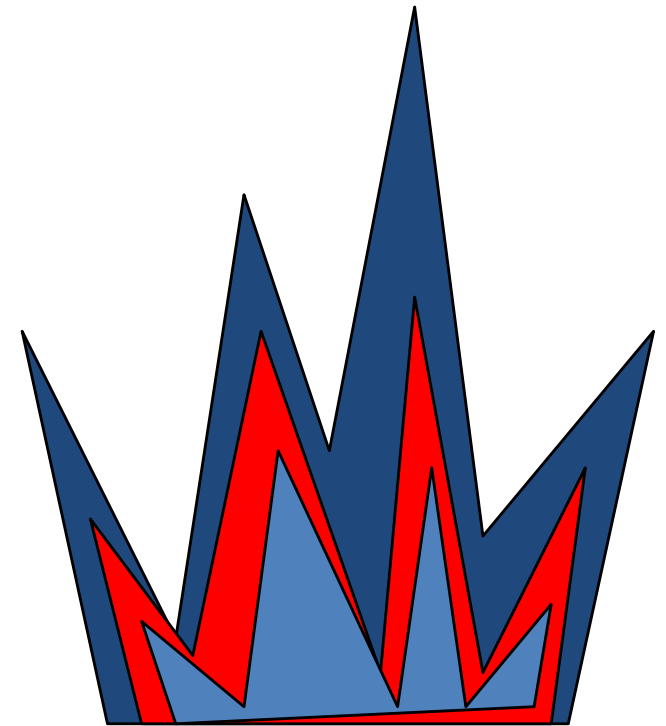
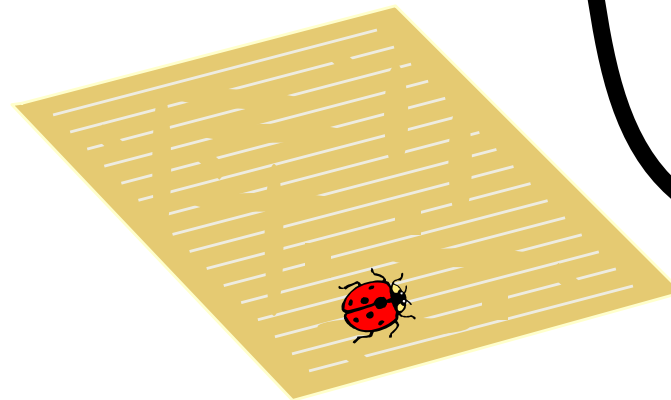
- ☐ Hành vi con người
→ Error (Mistake)
- ☐ Hệ quả xuất hiện trên chương trình
→ Bug (Fault/Defect)
- ☐ Khi thực thi chương trình
→ Failure
- ☐ Hệ quả không như mong đợi
→ Incident

Lỗi phần mềm

A person makes
an **error** ...



... that creates a
fault in the
software ...



... that can cause
a **failure**
in operation

Độ tin cậy (Reliability)

- ☐ Là xác suất phần mềm không phát sinh lỗi trong thời gian và điều kiện xác định
- ☐ Thảo luận
 - ☐ Liệu có một hệ thống không tồn tại lỗi?
 - ☐ Liệu một hệ thống đáng tin cậy nhưng vẫn tồn tại lỗi?
 - ☐ Liệu một hệ thống không lỗi là đáng tin cậy?

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: 500 triệu USD
 - ▣ Mariner space probe to Venus: 250 triệu USD
 - ▣ American Airlines: 50 triệu USD
- Có thể gây chết người
 - ▣ Therac-25
 - ▣ Airbus & Korean Airlines

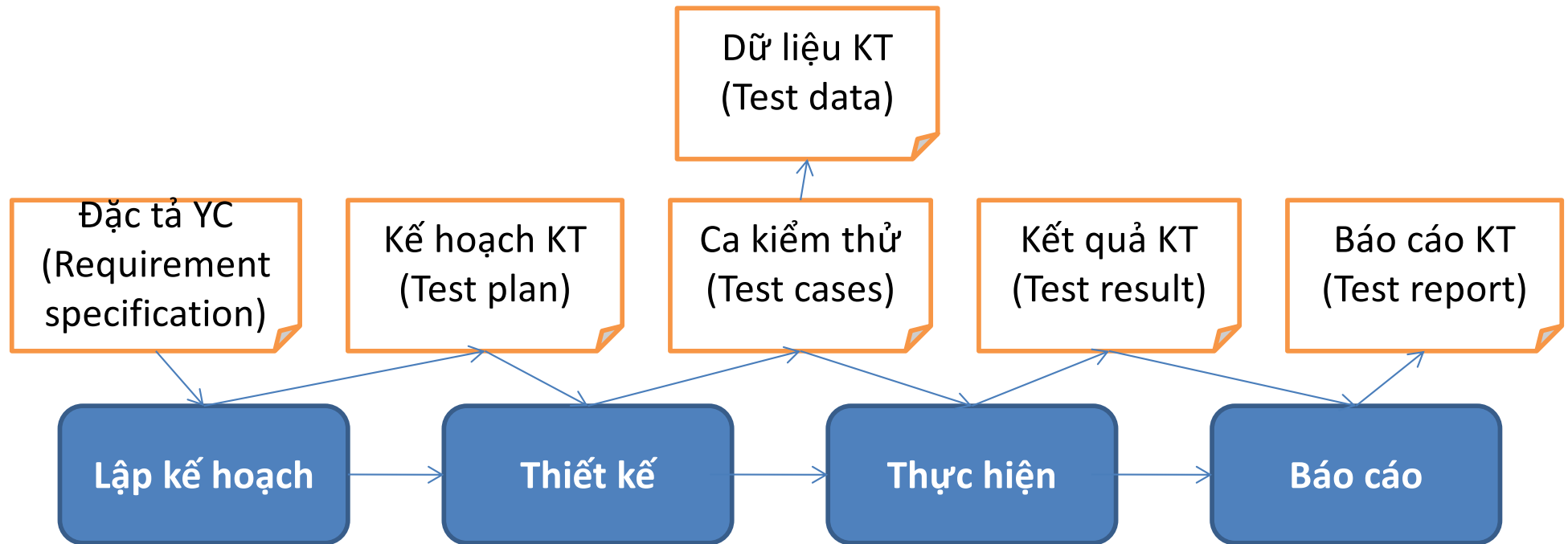
Chi phí lỗi

- Có thể không đáng kể
 - ▣ Bất tiện khi sử dụng
 - ▣ Ảnh hưởng không nhìn thấy được
- Không tuyến tính: 1 lỗi nhỏ nhưng có hậu quả lớn

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ Tại sao kiểm thử quan trọng?
- ☐ **Quy trình kiểm thử phần mềm**
- ☐ Các nguyên lý tổng quát
- ☐ Vai trò, thái độ và kỹ năng

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:
 - ☐ Giới thiệu
 - ☐ Yêu cầu
 - ☐ Chiến lược
 - ☐ Thời gian
 - ☐ Tài nguyên

Lập kế hoạch

Test
Policy

Test
Strategy

Company level

High Level
Test Plan

Project level (IEEE 829)
(one for each project)

Detailed
Test Plan

Test stage level (IEEE 829)
(one for each stage within a project,
e.g. Component, System, etc.)

Lập kế hoạch

- Các bước lập kế hoạch
 - Xác định yêu cầu kiểm thử
 - Khảo sát rủi ro
 - Xác định chiến lược kiểm thử
 - Xác định nhân lực, thiết bị
 - Lập kế hoạch chi tiết
 - Tổng hợp và tạo các bản kế hoạch kiểm tra
 - Xem xét các kế hoạch kiểm tra

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)

Thiết kế

- Các bước thiết kế ca kiểm thử
 - ▣ Xác định điều kiện cần thiết lập, mô tả dữ liệu đầu vào, kết quả mong chờ
 - ▣ Mô tả các bước chi tiết
 - ▣ Xem xét và khảo sát độ bao phủ
 - ▣ Xem xét test cases và các bước kiểm tra

Ca kiểm thử (test case)

- ☐ Test case: dữ liệu để kiểm tra hoạt động của chương trình
- ☐ Test case tốt: được thiết kế để phát hiện một lỗi của chương trình
- ☐ Kiểm thử thành công: phát hiện ra lỗi
- ☐ Mục đích
 - ☐ Chứng minh sự tồn tại của lỗi
 - ☐ Không chứng minh sự không có lỗi

Nội dung của test case

- Mô tả
 - ▣ Chức năng muốn kiểm thử
 - ▣ Dữ liệu đầu vào
 - ▣ Môi trường thử nghiệm
 - ▣ Thứ tự thao tác
- Kết quả mong muốn
 - ▣ Dữ liệu đầu ra
 - ▣ Màn hình, thời gian phản hồi
- Kết quả thực tế

Thực hiện

- ☐ Mục đích: thực hiện các ca kiểm thử, ghi nhận kết quả
- ☐ Kết quả: bảng báo cáo (Test result)

Thực hiện

- Các bước thực hiện kiểm thử
 - ▣ Xác lập và khởi động môi trường
 - ▣ Thực hiện các bước (bằng tay hoặc script)
 - ▣ Đánh giá quá trình kiểm thử
 - Hoàn tất chu kỳ → Thẩm định kết quả
 - Bị dừng hoặc treo → xác định nguyên nhân lỗi, khắc phục và lập lại
 - ▣ Thẩm định kết quả: bảo đảm kết quả nhận được là đáng tin cậy

Báo cáo

- ☐ 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)
- ☐ Lưu ý: bước đánh giá mang tính toàn cục

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ Tại sao kiểm thử quan trọng?
- ☐ Quy trình kiểm thử phần mềm
- ☐ **Các nguyên lý tổng quát**
- ☐ Vai trò, thái độ và kỹ năng

Các nguyên lý tổng quát

1. Phơi bày sự hiện diện của lỗi

- ❑ Cho thấy lỗi đang tồn tại
- ❑ Giảm xác suất lỗi chưa phát hiện

2. Không thể vét cạn hết các trường hợp

- ❑ Không thể kiểm nghiệm triệt để một phần mềm
- ❑ Thay vào đó:
 - Phân tích rủi ro
 - Độ ưu tiên

Các nguyên lý tổng quát

3. Kiểm tra sớm

- Nên bắt đầu sớm nhất có thể trong chu kỳ phát triển

4. Gom nhóm lỗi

- Nguyên lý Pareto: 20% module gây ra 80% lỗi
- → cô lập, tập trung những module khả nghi nhất

Các nguyên lý tổng quát

5. Nghịch lý thuốc trừ sâu (Pesticide paradox)

- ❑ Sử dụng cùng 1 kỹ thuật, ca kiểm thử nhiều lần → không tìm được lỗi mới
- ❑ Ca kiểm thử phải được xem xét và thay đổi thường xuyên

6. Phụ thuộc ngữ cảnh

- ❑ Thực hiện khác nhau trong các ngữ cảnh khác nhau

Các nguyên lý tổng quát

7. Ảo tưởng “không lỗi” (Absence-of-errors fallacy)

- ❑ Việc tìm và sửa chữa lỗi sẽ vô nghĩa nếu hệ thống được xây dựng xong vô dụng

Nội dung

- ☐ Kiểm thử phần mềm là gì?
- ☐ Tại sao kiểm thử quan trọng?
- ☐ Quy trình kiểm thử phần mềm
- ☐ Các nguyên lý tổng quát
- ☐ Vai trò, thái độ và kỹ năng

Vai trò

□ 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

Vai trò

☐ 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

- **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?

Thái độ

- ☐ 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

- ☐ 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

Thảo luận

