

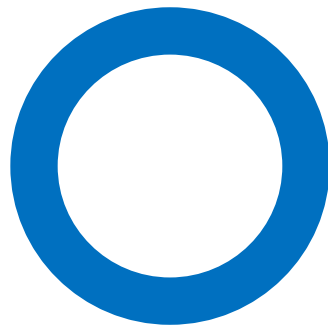
Tổng quan

Công cụ kiểm chứng phần mềm

Hồ Tuấn Thanh –

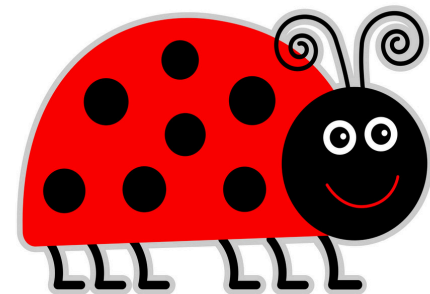
htthanh@fit.hcmus.edu.vn

Giới thiệu



Bug / Defect

- ☐ Không đáp ứng yêu cầu phần mềm (software requirement).
- ☐ Không đáp ứng mong đợi của người dùng (end user expectation).
- ☐ Error, flaw, failure or fault.
- ☐ Người dùng, phần mềm, library, OS.



Cho ví dụ vài bug
của chính bạn

Bug / Defect

- ❑ A program that contains a large number of bugs is said to be **buggy**.
- ❑ Reports detailing bugs in software are known as **bug reports**.
- ❑ Applications for tracking bugs are known as **bug tracking tools**.
- ❑ The process of finding the cause of bugs is known as **debugging**.

Software requirements

1.1 ĐỀ TÀI QUẢN LÝ THƯ VIỆN

1.1.1 Danh sách các yêu cầu

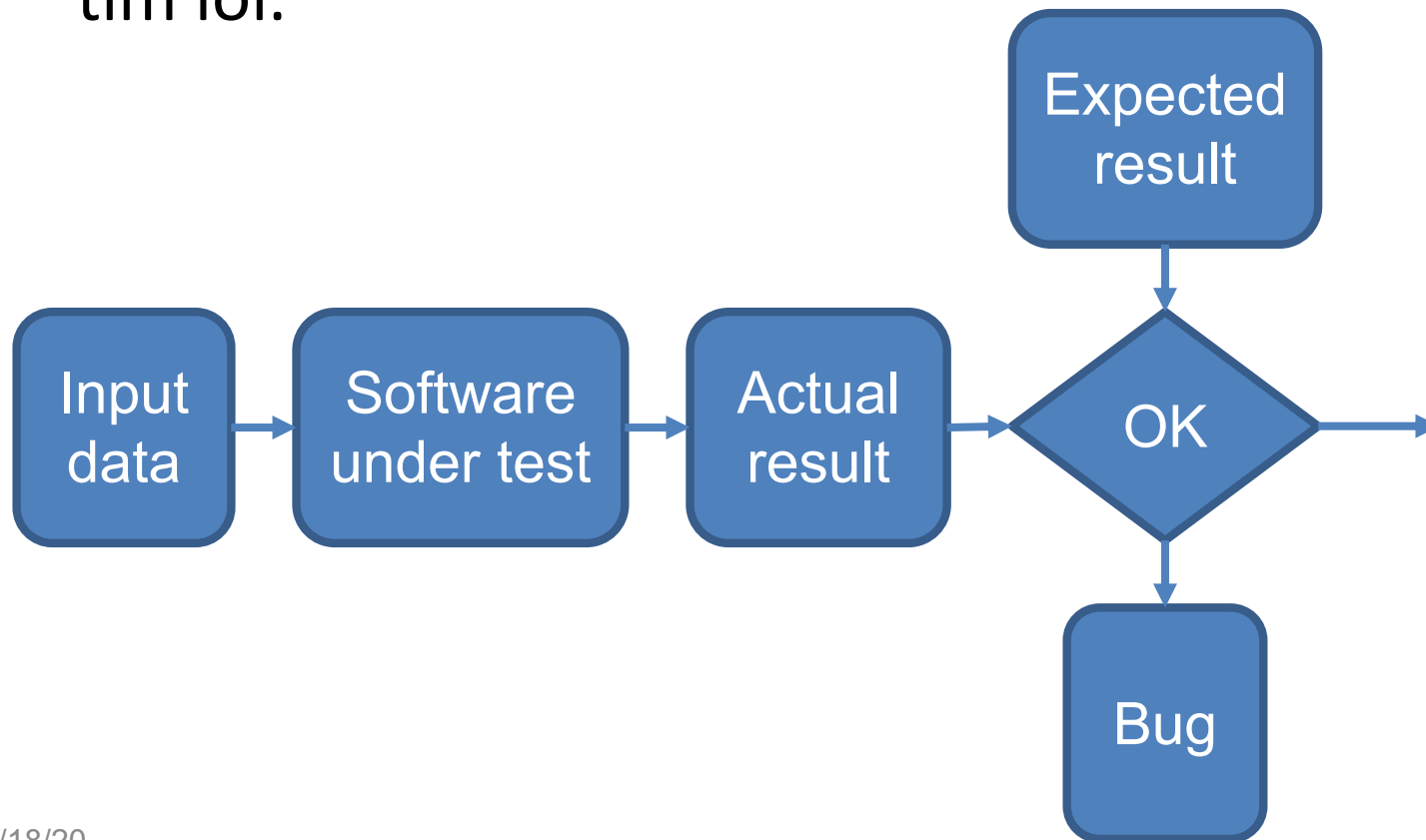
STT	Tên yêu cầu	Biểu mẫu	Qui định	Ghi chú
1	Tiếp nhận nhân viên	BM1	QĐ1	
2	Lập thẻ đọc giả	BM2	QĐ2	
3	Tiếp nhận sách mới	BM3	QĐ3	
4	Tra cứu sách	BM4		
5	Cho mượn sách	BM5	QĐ5	
6	Nhận trả sách	BM6	QĐ6	
7	Lập phiếu thu tiền phạt	BM7	QĐ7	
8	Ghi nhận mất sách	BM8	QĐ8	
9	Thanh lý sách	BM9	QĐ9	
10	Lập báo cáo	BM10		

End user expectation

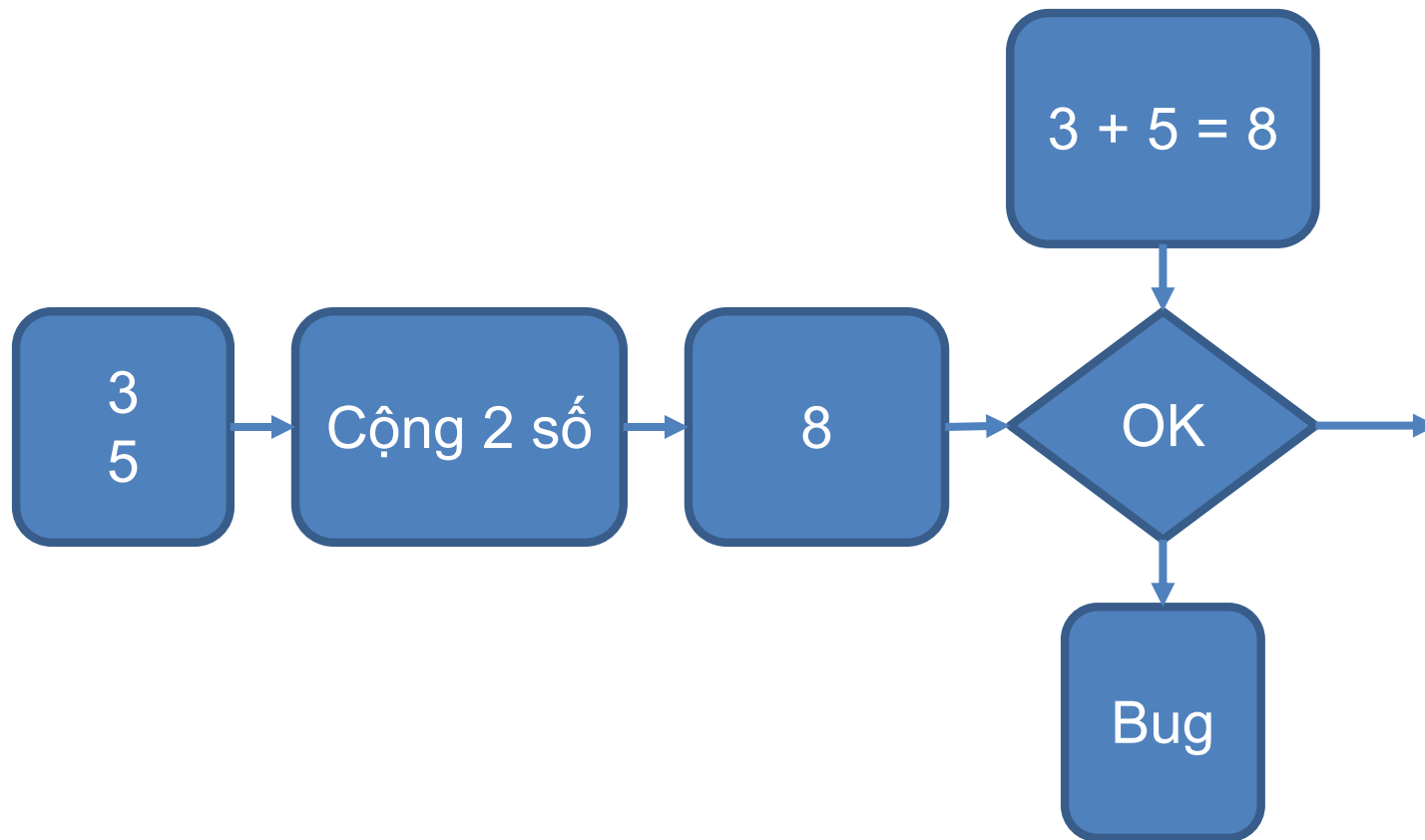
- ☐ Giao diện quá xấu.
- ☐ Sao không kiểm tra lỗi nhập liệu.
- ☐ ...

Software testing

- ❑ Quá trình thực thi một chương trình với mục đích tìm lỗi.



Ví dụ



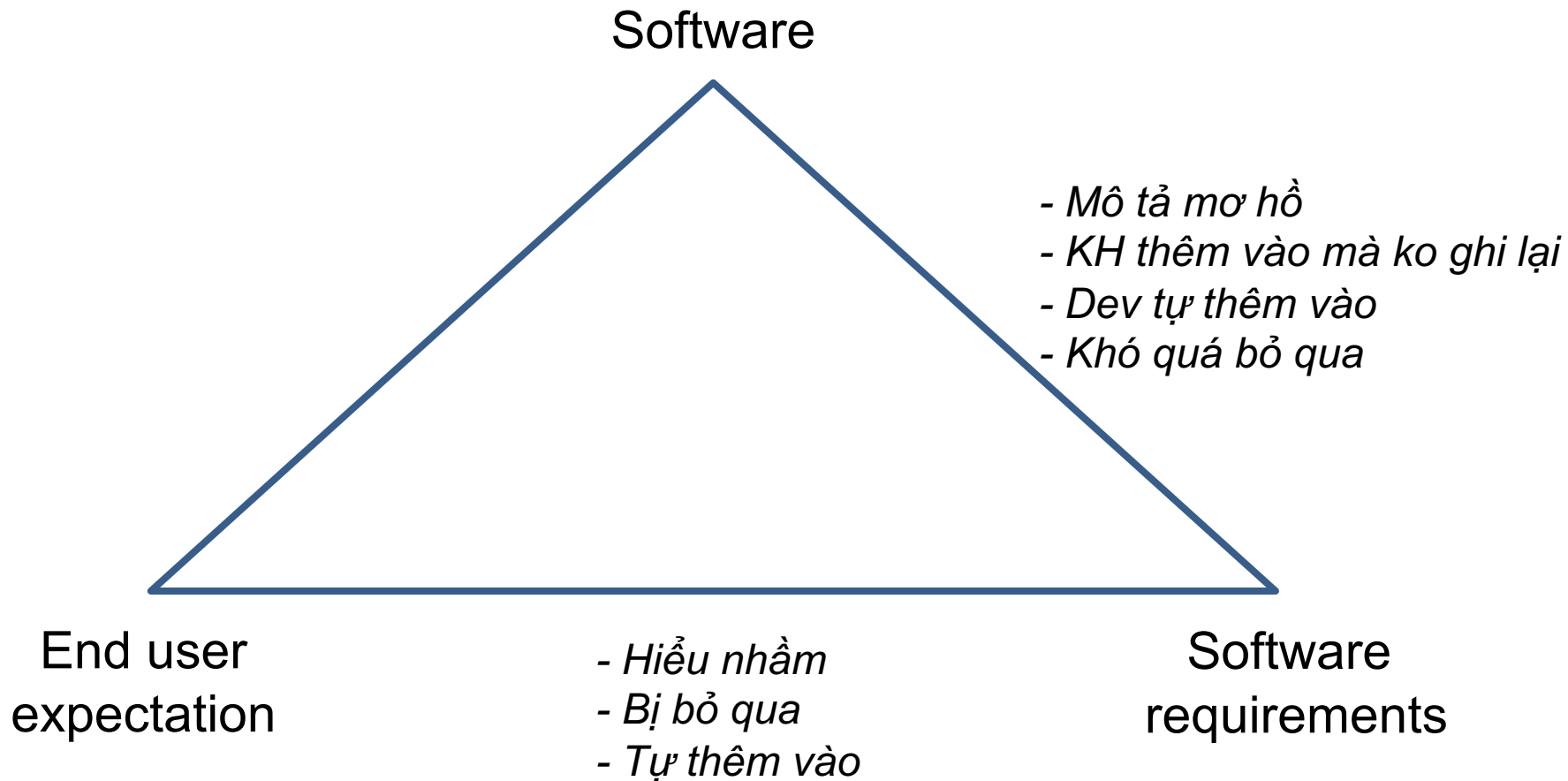
Bạn đã bao giờ
test phần mềm nào chưa?

Thế bạn học môn này
để làm gì?

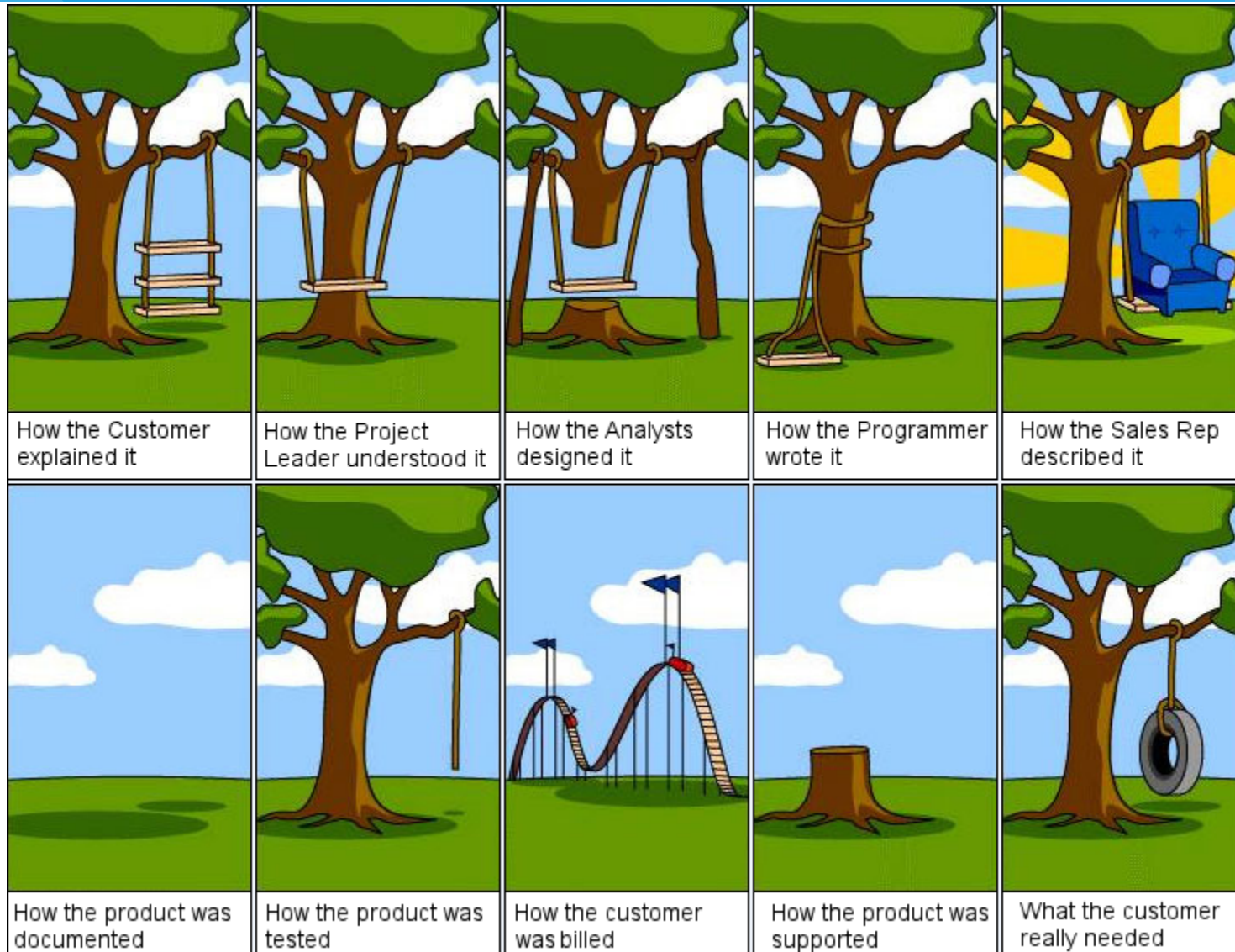
Software testing

- ❑ Verification: xác minh xem phần mềm có chạy chính xác hay không (so với software requirements).
- ❑ Validation: thẩm định xem phần mềm có thỏa mãn yêu cầu của khách hàng hay không (end user expectation).
- ❑ Nhằm đảm bảo chất lượng phần mềm (software quality).

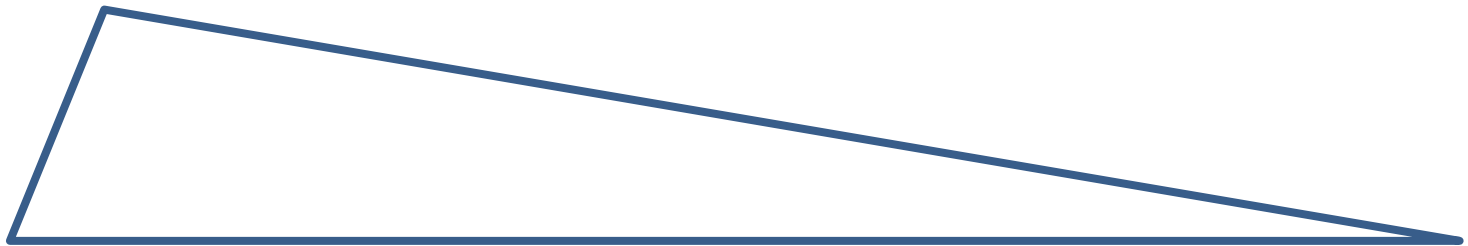
Quality triangle



Triangle quality

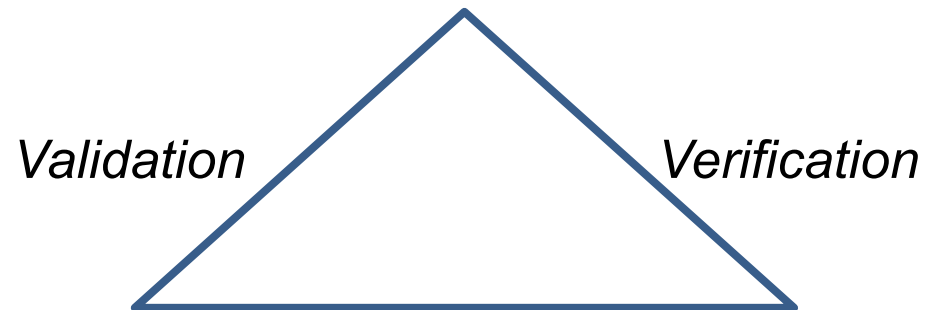


Quality triangle



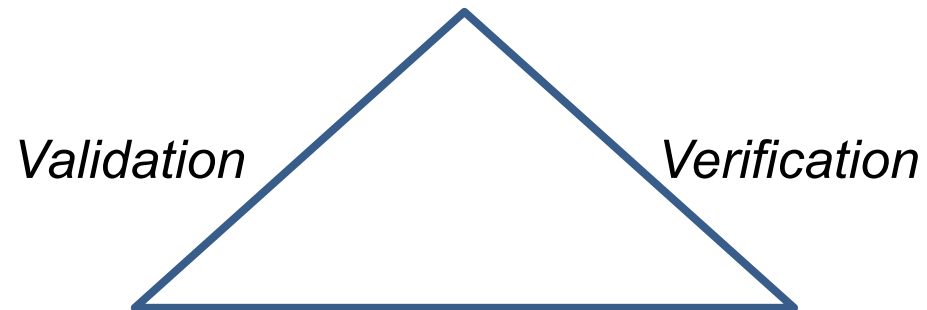
Verification vs Validation

- ❑ Verification: vs software requirements.
- ❑ Validation: vs end user expectation.
- ❑ Verification: build system CORRECTLY.
- ❑ Validation: build CORRECT system.



Verification vs Validation

- ❑ Verification: 80% khối lượng công việc.
- ❑ Validation: 20% khối lượng công việc.
- ❑ Validation: 80% hiệu quả chung.



Verification vs Validation

- ❑ Verification: điều kiện cần.
- ❑ Validation: điều kiện đủ.

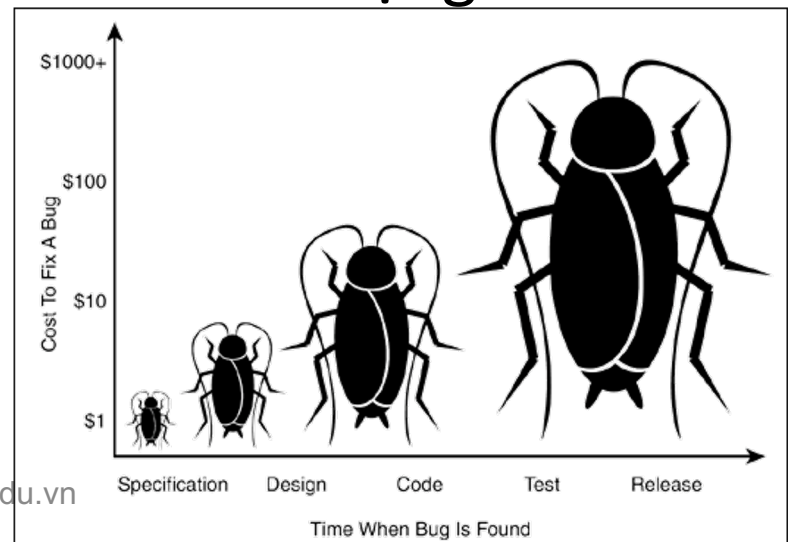
Tại sao có bug?

- ❑ Phần mềm được viết ra bởi con người.
 - Biết nhiều thứ, nhưng không phải mọi thứ.
 - Có kĩ năng, nhưng không hoàn hảo.
 - Luôn lun phạm sai lầm.

- ❑ Làm việc dưới áp lực căng thẳng để kịp bàn giao sản phẩm đúng tiến độ.
 - Không có thời gian kiểm tra.
 - Giả định sai.
 - Hệ thống chưa hoàn chỉnh.

Chi phí lỗi (cost)

- ❑ Tăng theo cấp số mũ (exponential).
- ❑ Lỗi phát hiện càng trễ, chi phí sửa càng cao.
- ❑ Xin lỗi khách hàng, sửa lại, discount cho khách hàng, đền tiền, bồi thường cho nhân mạng.



Cho ví dụ vài lỗi
bạn đọc được trên web

Should

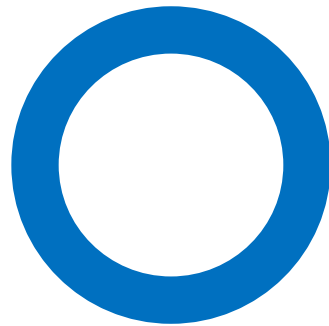
- ☐ Phần mềm luôn tồn tại lỗi.
- ☐ Chi phí lỗi có thể rất cao.
- ☐ Đảm bảo chất lượng phần mềm.
- ☐ Tránh bị khách hàng than phiền.
- ☐ Giữ uy tín cho công ty.

- ☐ Đạo đức.

Should not

- ☐ Lấp đầy khoảng thời gian giữa ngày hoàn thành dự án và ngày bàn giao sản phẩm.
- ☐ Chứng minh phần mềm ko lỗi.

8 nguyên lí



8 software testing principles

1. Testing shows presence of defects.

- Chỉ giúp giảm số lỗi chưa phát hiện ra.
- Không thể chứng minh phần mềm ko còn lỗi.
- *M\$ có dám tuyên bố Windows ko có bug?*

2. Exhaustive testing is impossible.

- Không thể vét cạn mọi trường hợp.
- Hãy phân tích rủi ro, đặt độ ưu tiên.
- *Cần test bao nhiêu trường hợp cho chương trình cộng 2 số nguyên?*

8 software testing principles

3. Early testing.

- Nên bắt đầu sớm quá trình kiểm thử.

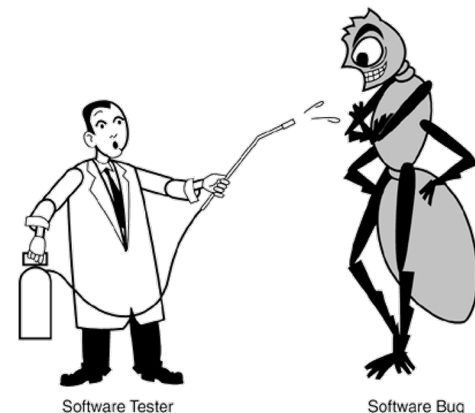
4. Defect clustering.

- Nguyên lí Pareto (nguyên lí 80-20).
- 80% lỗi xảy ra ở 20% module.
- Cô lập, tập trung vào những module khả nghi.
- *Web bán hàng, nên test phần nào trước?*

8 software testing principles

5. Pesticide paradox

- Nghịch lí thuốc trừ sâu.
- Sử dụng cùng một test technique, cùng một test case để kiểm thử nhiều lần thì sẽ ko thì được lỗi mới.
- Test case phải được xem xét và thay đổi thường xuyên.
- *Đổi cách làm, đổi người mới, đổi project.*



8 software testing principles

6. Testing is context depending

- Phụ thuộc vào ngữ cảnh khác nhau.

7. Absence-of-errors fallacy

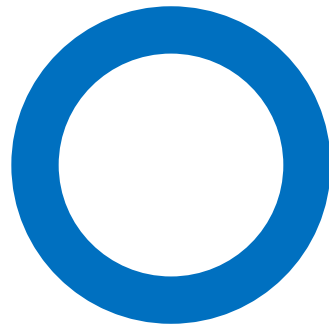
- Hệ thống xây dựng ra mà ko sử dụng được, ko đáp ứng yêu cầu khách hàng, thì việc tìm và sửa lỗi chẳng giúp ích gì.

8 software testing principles

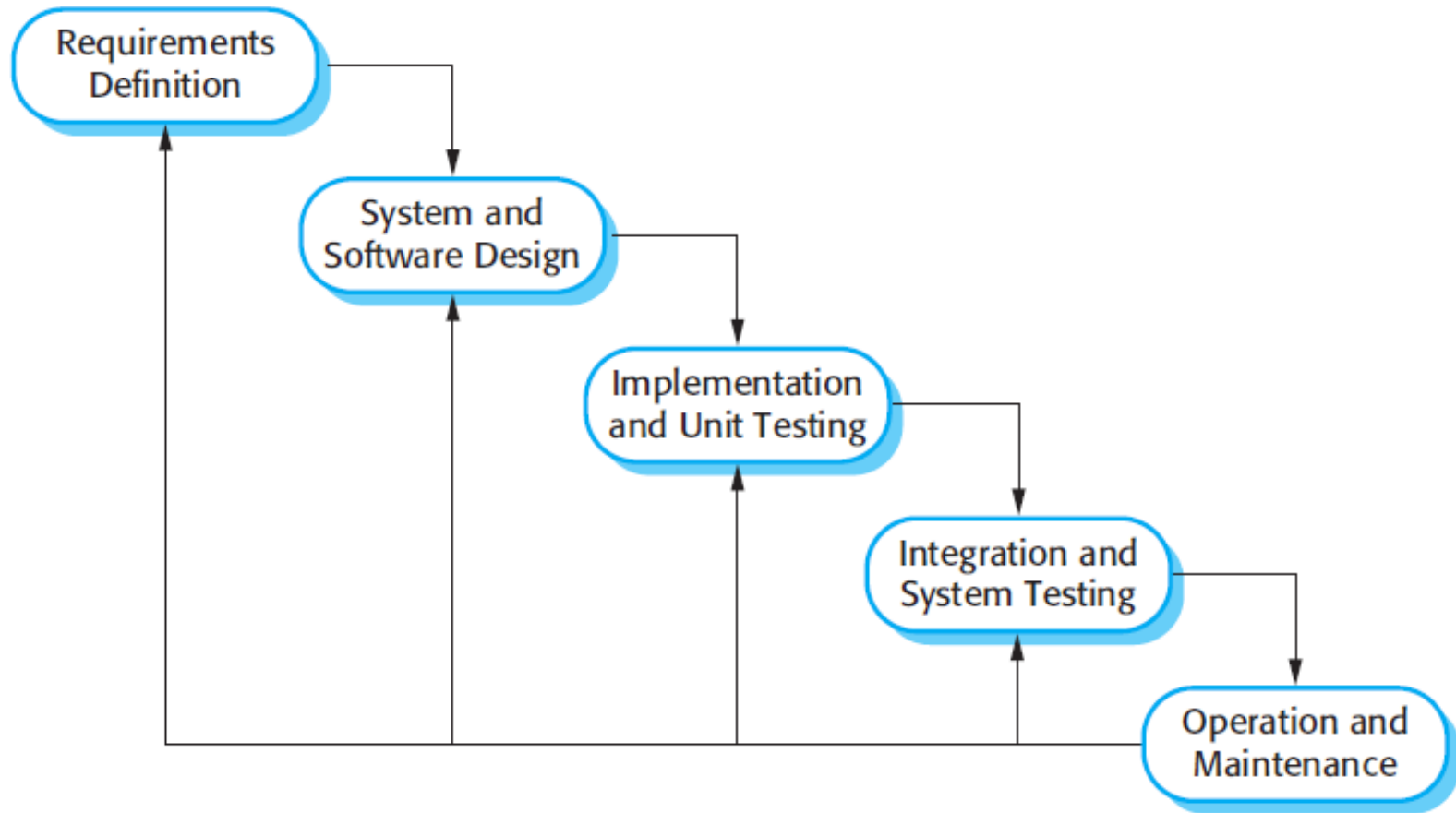
8. Not all the bugs you find will be fixed.
- Ko đủ thời gian.
 - Ko phải là bug.
 - Quá nguy hiểm để fix.
 - Ko đáng để fix.

It's not a bug,
but a undocumented feature

Qui trình

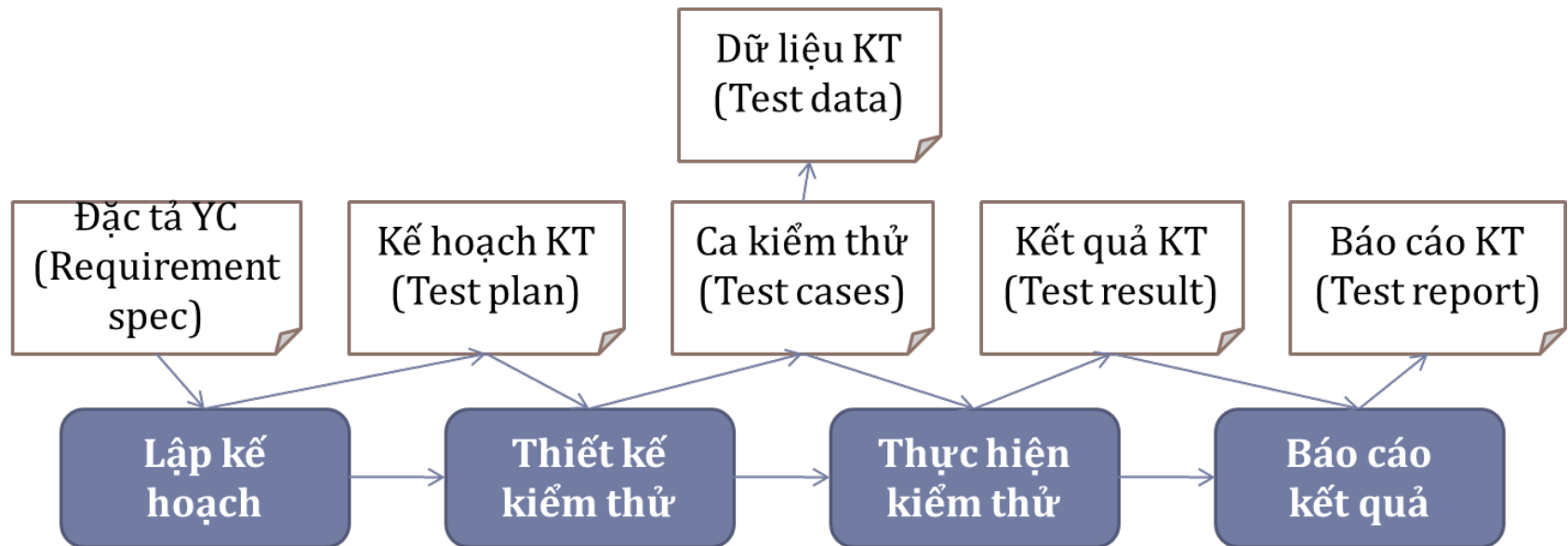


Software process



Trong quá trình developer
code, thì tester làm gì?

Software testing process



Planning

- ❑ Mục tiêu: mô tả software under test, chiến lược, thời gian, tài nguyên.
- ❑ Đầu vào:
 - Tài liệu đặc tả yêu cầu (software requirement document).
 - Tài liệu kế hoạch dự án (project plan document).
- ❑ Đầu ra:
 - Tài liệu kế hoạch kiểm thử (test plan document).
- ❑ Nội dung:
 - Giới thiệu.
 - Chiến lược.
 - Thời gian.
 - Tài nguyên.

Designing

- ❑ Mục tiêu: bảo đảm tất cả các tình huống kiểm tra “quét” hết tất cả các yêu cầu cần kiểm tra
- ❑ Đầu vào:
 - Software requirement document.
 - Project plan document.
- ❑ Đầu ra:
 - Tài liệu trường hợp kiểm thử (test case document).
- ❑ Nội dung:
 - Test data.
 - Các bước thực hiện (step).
 - Đầu ra mong đợi (expected result).

Executing

- ❑ Mục tiêu: thực hiện các test case, ghi nhận kết quả.
- ❑ Đầu vào:
 - Software requirement document.
 - Test case document.
 - Application.
- ❑ Đầu vào:
 - Báo cáo kết quả kiểm thử (test result document).
 - Báo cáo lỗi (bug report document).
- ❑ Nội dung: ghi nhận kết quả thực tế khi chạy các test case.

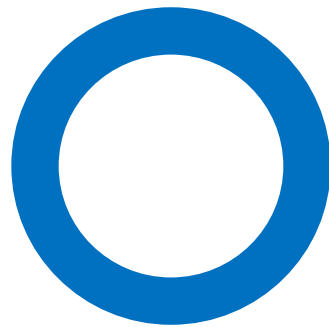
Reporting

- ❑ Mục tiêu: xem xét và đánh giá kết quả kiểm tra, liệt kê lỗi, chỉ định các yêu cầu thay đổi, tổng kê số liệu.

- ❑ Đầu vào:
 - Test result document.
 - Bug report document.

- ❑ Đầu ra:
 - Báo cáo thống kê (test report document).
 - Bug report document.

Vai trò



Quality Control (QC)

- ❑ Thực hiện những hoạt động, áp dụng các kĩ thuật nhằm đảm bảo chất lượng sản phẩm.

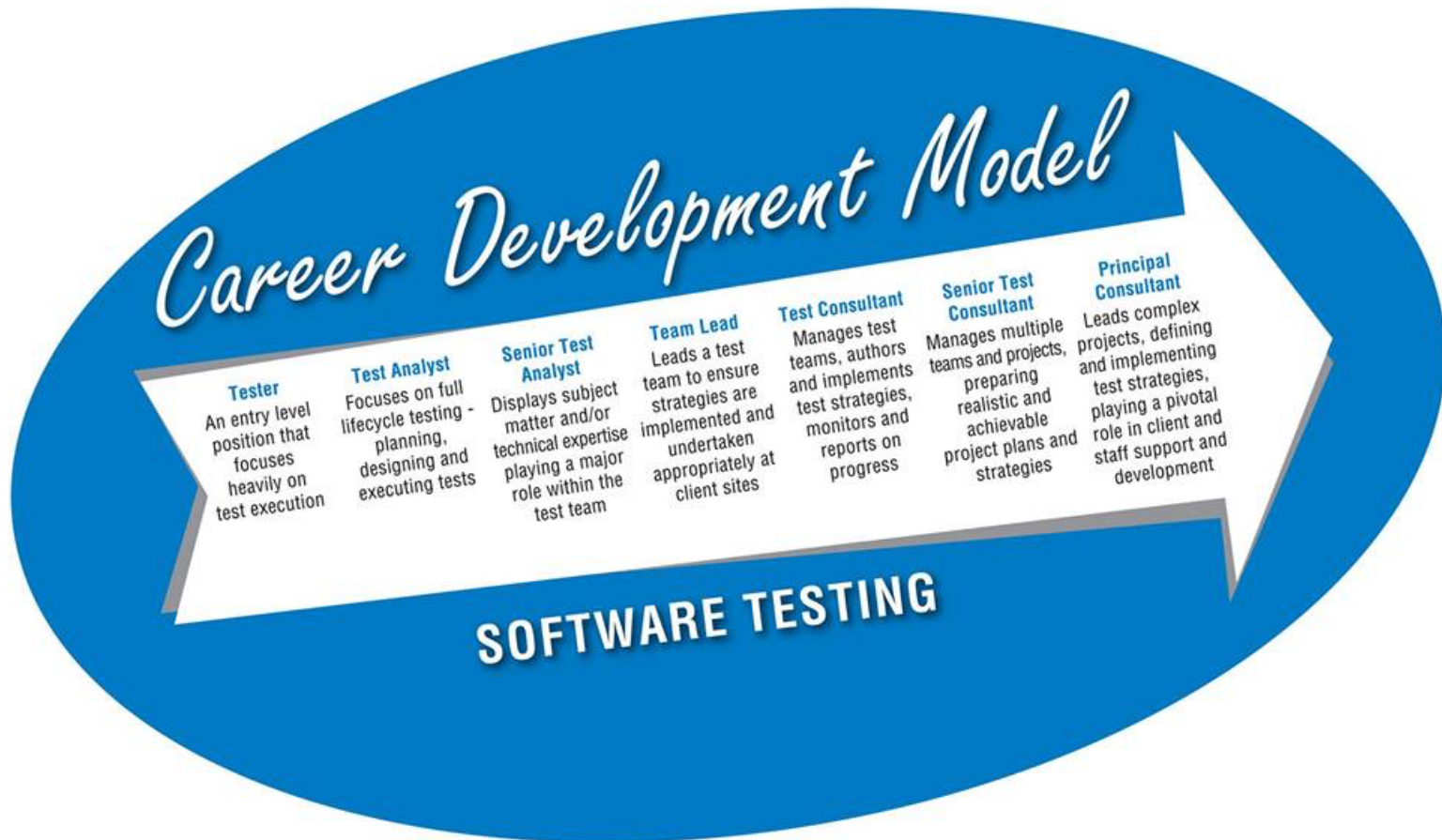
Quality Assurance (QA)

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

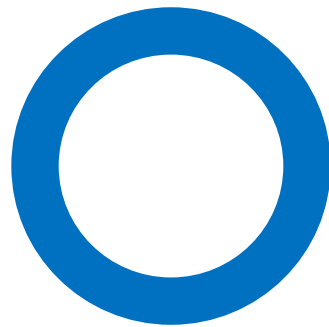
Titles of testers

- ☐ **Fresher**
- ☐ Test Engineer
- ☐ Quality Analyst
- ☐ Testing Analyst
- ☐ Automation Engineer
- ☐ Test Architect
- ☐ Test Automation Developer
- ☐ Test Manager
- ☐ Director of Test
- ☐ Senior Test Engineer

Career path



Kĩ năng, thái độ



Kĩ năng

- ☐ Kĩ năng giao tiếp.
- ☐ Kĩ năng đọc.
- ☐ Kĩ năng giải quyết vấn đề.
- ☐ Kĩ năng lập báo cáo.
- ☐ Kĩ năng quản lí bản thân.
- ☐ Kĩ năng ngoại ngữ.

Thái độ

- ❑ **Cẩn thận (Cautious)**
 - Phỏng đoán chứ ko kết luận
 - Tập thừa nhận “Tôi ko 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)

