



ĐẢM BẢO CHẤT LƯỢNG & KIỂM THỬ PHẦN MỀM

CHƯƠNG 1

TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

02/2022



MỤC TIÊU CHƯƠNG

- **Hiểu rõ chất lượng của phần mềm**
- **Nhân tố, sản phẩm tạo ra phần mềm chất lượng tốt**
- **Hiểu rõ khái niệm Software Testing**
- **Mục tiêu của Testing**
- **Phân biệt Software Testing trong lý thuyết và thực tế**

- **Chất lượng phần mềm**
- **Software Testing**
- **Tại sao phải kiểm thử phần mềm**
- **Nguyên lý trong kiểm thử phần mềm**
- **Một số khái niệm liên quan**

CHẤT LƯỢNG PHẦN MỀM

- **Quan niệm về chất lượng phần mềm**
- **Tam giác chất lượng**
- **Định nghĩa Software Testing**
- **Mục tiêu hoạt động kiểm thử phần mềm**
- **Các quan niệm khác nhau giữa lý thuyết và thực tế về Software Testing**

Quality in different views

What do you think of the IPHONE 12?



I really like it!

I don't like it!



Manufacturer view

Nhà sản xuất



Meet the requirements
specification

Product view



Meet attributes of a product

User view



Fit for use

Đáp ứng **đạt điểm kỹ thuật yêu cầu**
Đáp ứng các **thuộc tính** của một sản phẩm
Phù hợp để **sử dụng**

Chất lượng phần mềm

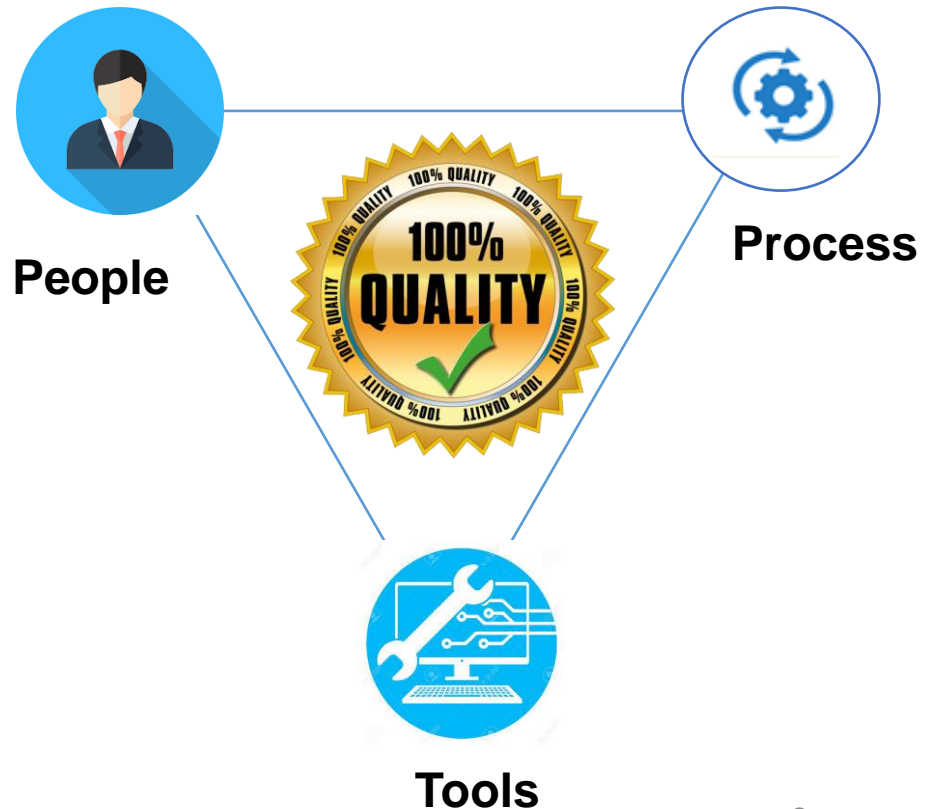
- Quan niệm về chất lượng phần mềm
- **Tam giác chất lượng**
- Định nghĩa Software Testing
- Mục tiêu hoạt động kiểm thử phần mềm
- Các quan niệm khác nhau giữa lý thuyết và thực tế về Software Testing
- Một số khái niệm liên quan

The Quality Triangle

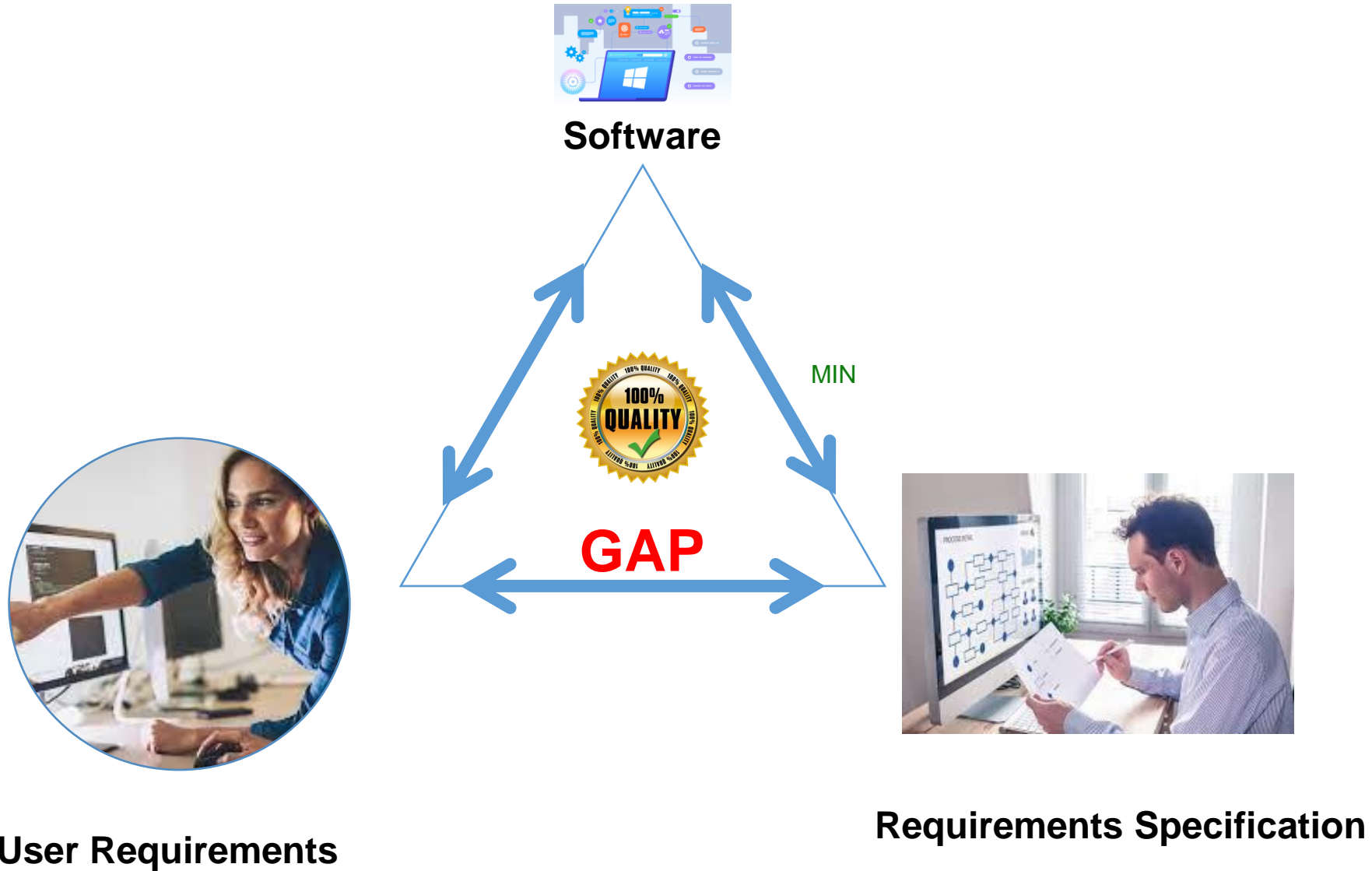


It is wrong to assume that
people alone are responsible
for poor quality

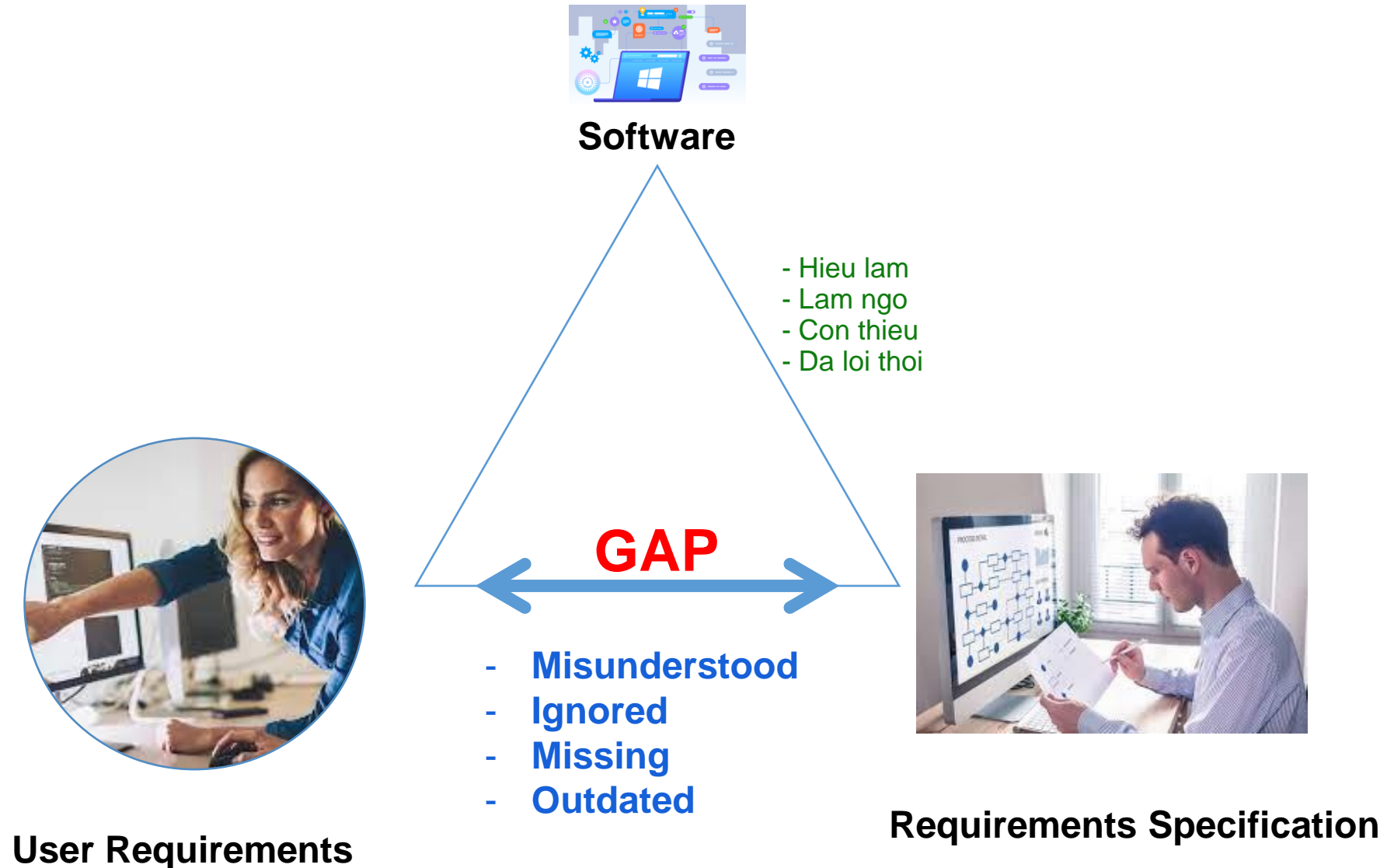
Thật sai lầm khi cho rằng một mình con
người chịu trách nhiệm về chất lượng kém



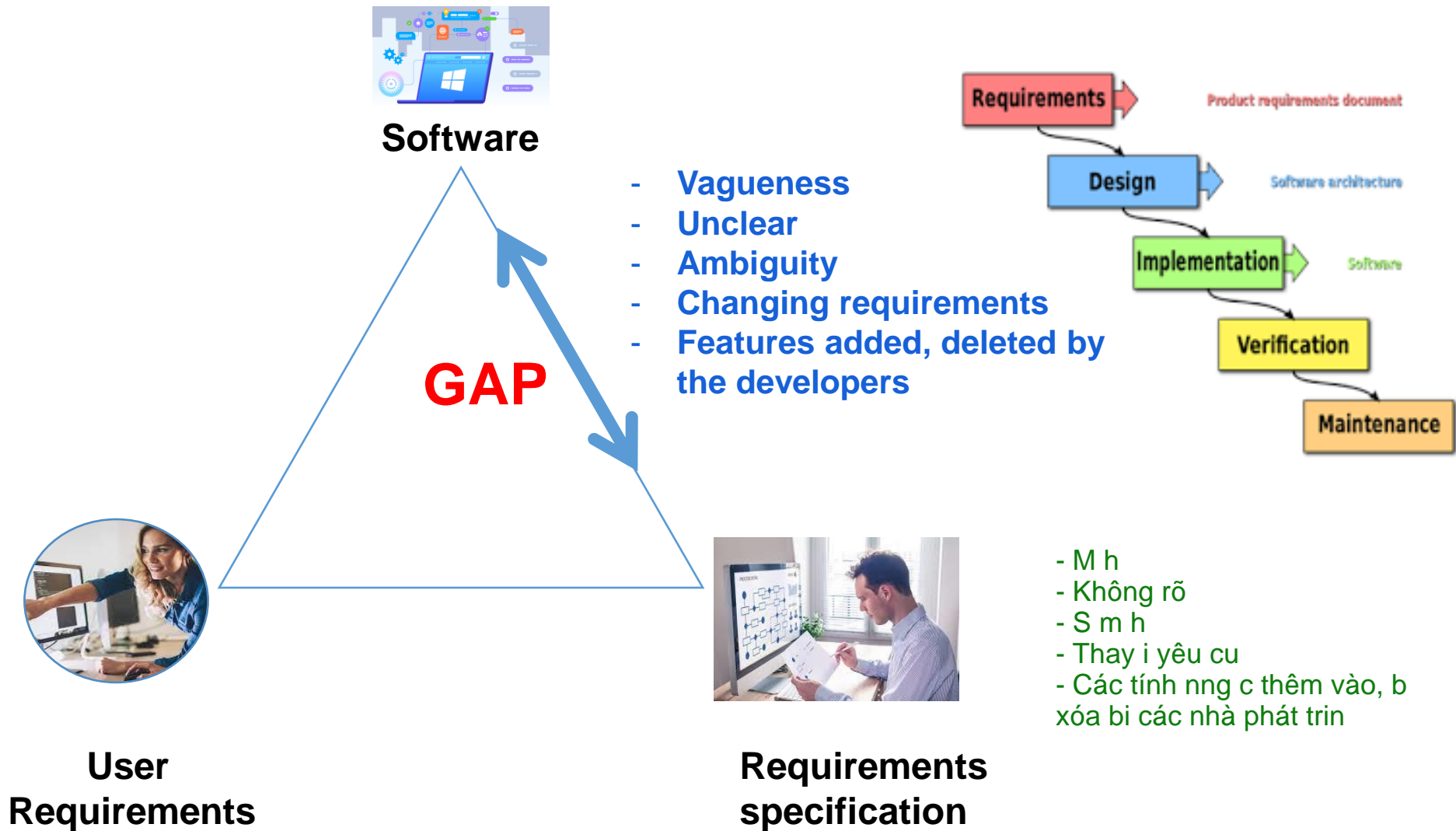
The Quality Triangle



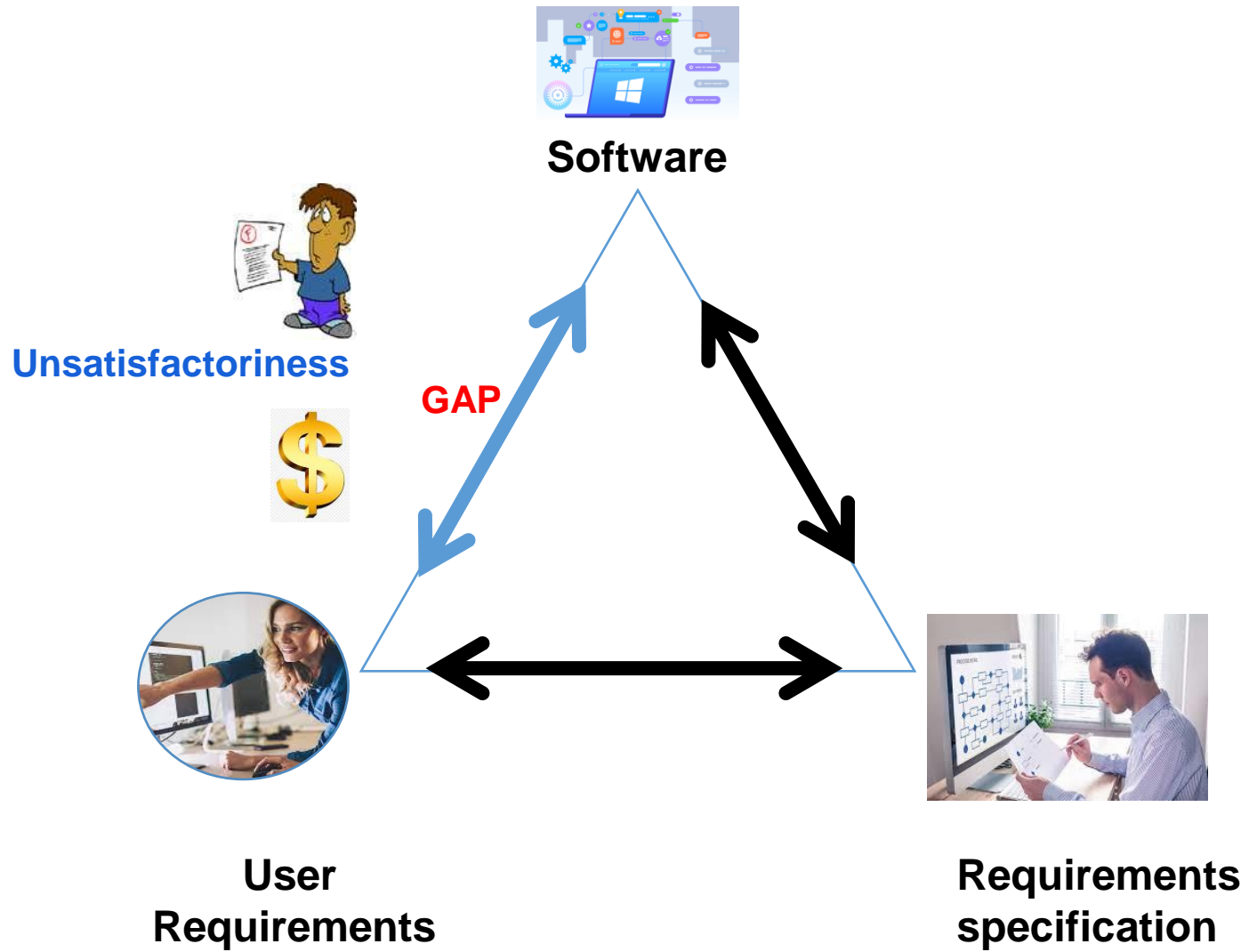
The User requirements – Requirements Specification Gap



The Requirements Specification – Software Gap



The Software - User requirements Gap



Close Gap between three views

Static Test

- Early requirement validation
- Reviews, inspection
- Configuration management

Dynamic Test

- Unit testing
- Integration testing
- System testing
- Acceptance testing

ống Không cách gia ba ch xem
Kim tra tn

- Xác thc yêu cu sm
- ánh giá, kim tra
- Qun lý cu hình
- Kim tra ng
- Kim tra n v
- Th nghiệm hi nhp
- Th nghiệm h thng
- Kim tra chp nhn

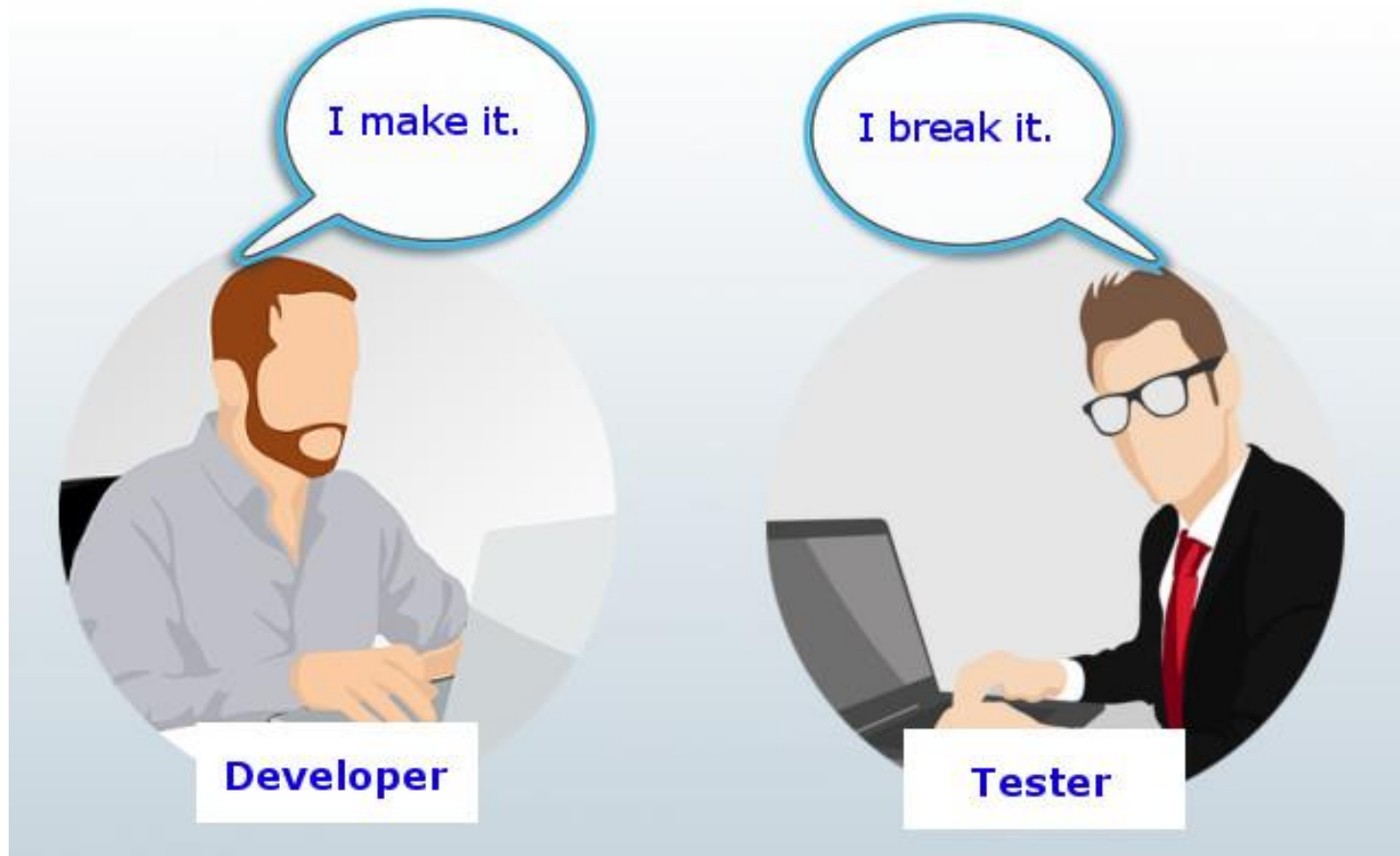


Improve the processes that create the product.

- Chất lượng phần mềm
- **Software Testing**
- Tại sao phải kiểm thử phần mềm
- Nguyên lý trong kiểm thử phần mềm
- Một số khái niệm liên quan

Mindset của Dev và Tester

- Vai trò của Tester là không cần thiết và một developer có thể tự kiểm tra mã code do mình viết?



What is Software Testing?

Testing is the **process** of exercising or **evaluating** a system or system component by manual or automated means to **verify** that it **satisfies specified requirements**

Kim tra là quá trình thực hiện học ánh giá mt h thng

(ANSI/IEEE Standard 729, 1983)

What is Software Testing?

The **process** consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and **evaluation** of software products and related work products to **determine** that they **satisfy specified requirements**, to demonstrate that they are fit for purpose and to detect defects.

(ISTQB Glossary, 2018)

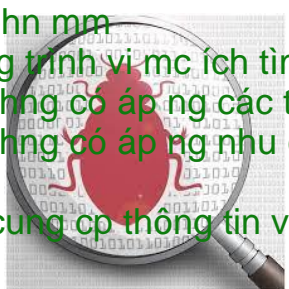
Quá trình bao gồm tất cả các hoạt động trong vòng đời, các hoạt động tĩnh và động, quan tâm đến việc lập kế hoạch, chuẩn bị và đánh giá các sản phẩm phần mềm và các sản phẩm công việc liên quan xác nhận rằng chúng đáp ứng các yêu cầu cụ thể, chứng minh rằng chúng phù hợp với mục đích và phát hiện khuyết tật.

Software Testing Objectives

What's Software Testing for?

- Execute a program with the intent of **finding defects** to
 - ✓ Determine whether system **meets specifications**
 - ✓ Determine whether system **meets user's needs**
 - ✓ Gain confidence and provide information about the **level of quality**
 - ✓ **Prevent defects**
- Continuously **improve the testing process**

Mục tiêu kiểm thử phần mềm
Thử nghiệm một chương trình vì mục đích tìm ra các khuyết tật
Xác định xem hệ thống có đáp ứng các thông số kỹ thuật hay không
Xác định xem hệ thống có đáp ứng nhu cầu của người dùng hay không
Cơ sở tin và cung cấp thông tin về mức độ tin cậy
Ngăn ngừa các khuyết tật
Kiểm thử phần mềm làm gì?
Liên tục cải tiến quy trình thử nghiệm



Testing Myth & Fact

Myth 1

- Testing is **easy**

Fact

- Testing can be **extremely challenging**
- Testing an application for possible use cases with minimum test cases requires high analytical skills

Testing Myth & Fact

Myth 2

- **Anyone** can do manual testing

Fact

- Testing requires **many skill sets**

Testing Myth & Fact

Myth 3

- Testing ensures **100%** defect free product

Fact

- Testing attempts to find as many defects as possible
- Identifying all possible defects is impossible

Testing Myth & Fact

Myth 4

- Automated testing is **more powerful** than manual testing

Fact

- 100% test automation cannot be done
- Manual testing is also essential

- Chất lượng phần mềm
- Software Testing
- **Tại sao phải kiểm thử phần mềm**
- Nguyên lý trong kiểm thử phần mềm
- Một số khái niệm liên quan

Why is testing?



Why is testing?

- Tại sao việc kiểm thử phần mềm là cần thiết?
- Lỗi trong các sản phẩm phần mềm
- Khi nào lỗi xuất hiện?
- Chi phí lỗi thay đổi như thế nào

Why is testing necessary?

- Because software is likely to have **FAULTS**
- To learn about the **RELIABILITY** of the software
- Because failures can be very **EXPENSIVE** or even **DANGEROUS**
- To avoid being **SUED** by customers
- To stay in Bussiness

Software failures

- Loss of money
- Loss of time
- Loss of business reputation
- Injury/Death

Loss of Money

and **P1 MILLION** Millions And Millions In Cash Prizes!

HERE'S HOW:

1. Look under specially-marked crowns and/or resealable caps of Pepsi, 7 Up, Mirinda and Mountain Dew. These crowns are marked with:
 - A 3-digit number ranging from 001 to 999;
 - Cash prizes in denominations of P1,000, P10,000, P50,000, P100,000 and P1,000,000;
 - A security code.
2. Everyday, from Monday to Friday starting February 17 up to May 8, 1992, a winning 3-digit number will be announced on national and local TV, radio and newspapers.
3. All holders of crowns and/or caps bearing the winning 3-digit number will win the corresponding cash prize printed on the crowns/caps.
4. All winning numbers are valid throughout the promo period, not just on the days the numbers are announced. For example, if any day during the promo period you open a crown that bears a winning number already previously announced, you win. Likewise, if today you open a crown bearing a number that has not been announced yet, keep your crown! It could win any day during the promo period! All winning numbers are randomly selected by computer.
5. Watch out for the winning numbers to be announced everyday! Check with your nearest sari-sari store for a record of all the winning numbers.
6. This promo is valid in all areas covered by Pepsi-Cola plants and warehouses nationwide.
7. Cash prizes worth P1,000 can be redeemed from all Pepsi-Cola plants and warehouses nationwide. Cash prizes P10,000 and up can be redeemed from all Pepsi-Cola plants only. Prizes not claimed within sixty (60) days after the end of promo will be automatically forfeited. Promo period is from February 17 to May 8, 1992.
8. All cash prizes are tax-free. Pepsi-Cola will shoulder any withholding tax due on the prizes.
9. Employees of Pepsi-Cola Products Philippines, Inc. (PGPPI), PepsiCo Inc., Pepsi-Cola Far East Trade Development Co., Inc., their advertising and public relations agencies, and relatives to the second degree of consanguinity and/or affinity are disqualified from joining the promotion.

The owner of this store will write in the winning numbers below.

MARCH 2	MARCH 9	MARCH 16	MARCH 23	MARCH 30	APRIL 6	APRIL 13	APRIL 20
401	901	347	180	703	762	425	355



Loss of Business Reputation

Nissan triệu hồi hơn 1 triệu xe khởi thị trường do lỗi phần mềm trong các máy dò cảm ứng túi khí

5/1996, một lỗi phần mềm → tài khoản ngân hàng của 823 khách hàng của một ngân hàng lớn ở Hoa Kỳ được ghi có 920 triệu đô la Mỹ

Users don't like bugs



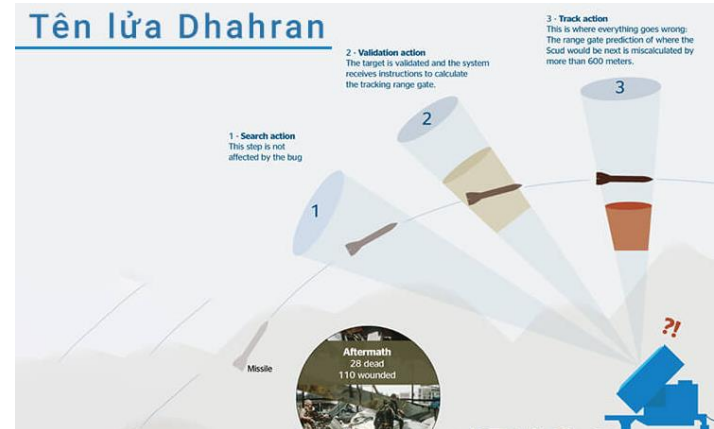
Injury/Death

26/4/1994, Lỗi phần mềm làm máy bay Air300 A300 của China Airlines bị rơi, 264 người mất



2/1991, Chiến tranh vùng Vịnh, một tên lửa của Iraq đã đi lọt vào căn cứ Mỹ ở Dhahran làm chết 28 lính Mỹ

- Không tính toán đến sai số pin.
- Sau 1 giờ: đồng hồ sai 1 mili giây
- Sau 100 giờ: sai 0,33 giây
- 0,33 giây thì radar có thể phát hiện được tên lửa nhưng hệ thống phòng thủ không kịp khai hỏa



Defect & Bug in Software Projects

- Error (mistake) Loi do hanh dong con nguoi gay ra
- Fault
- Failure
- Bug
- Defect (fault, bug)
- Incident

Defect & Bug in Software Projects



Error

- A **human action** that produces an incorrect result
- Lỗi xảy ra khi có tác động lên sản phẩm gây ra một kết quả sai lệch

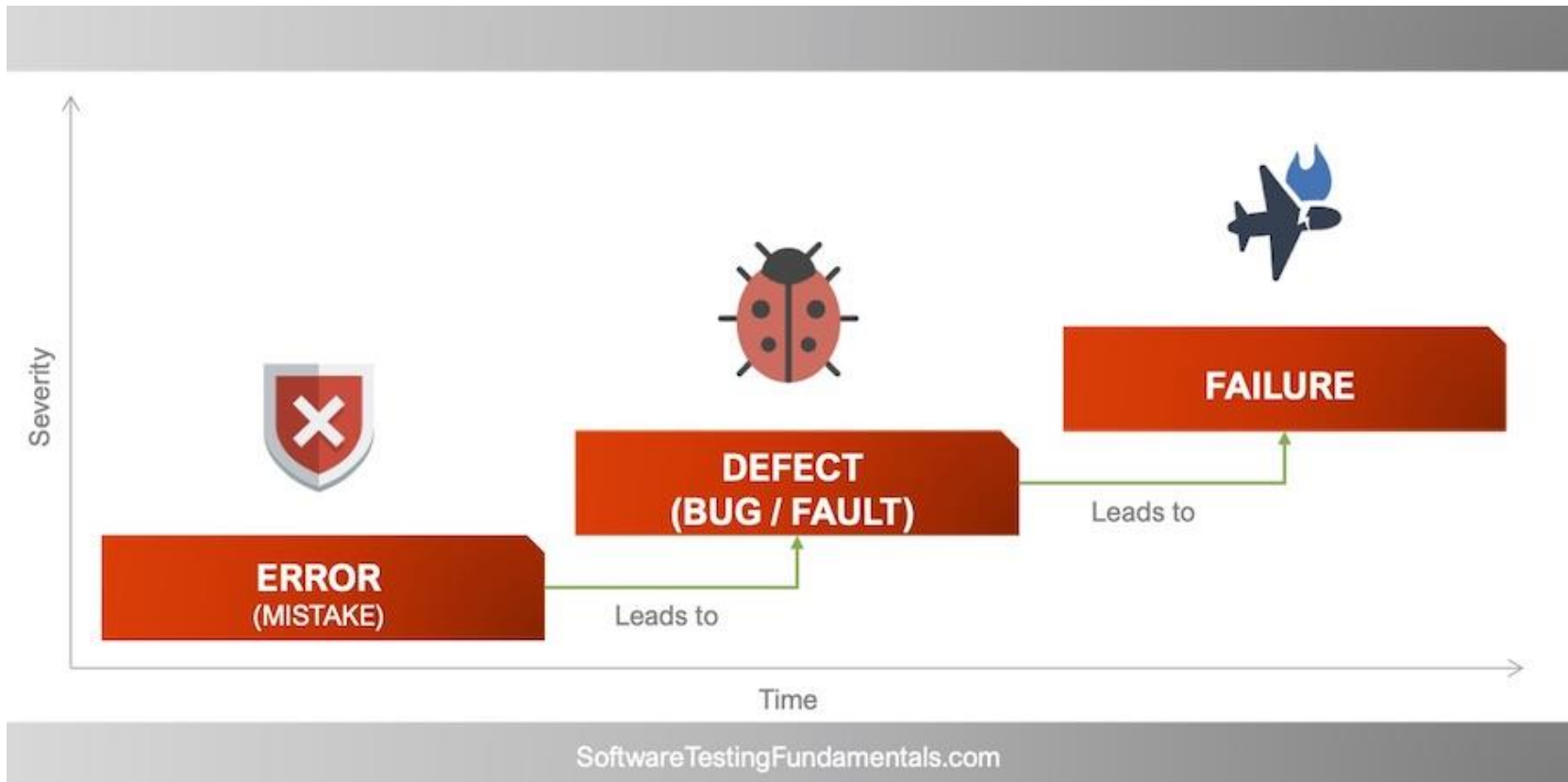
Fault

- An incorrect step, process, or data definition.
- Lỗi xảy ra khi làm sai các step, process, hoặc chuẩn bị dữ liệu

Failure

- A **behavioral deviation** from user requirement or product specification
- Lỗi khi có kết quả sai lệch so với yêu cầu đặc tả

Relation of error, fault and failure



Defect/ Bug/ Incident Management

When do defects arise?



How the
customer
explained it.



How the Project
Manager
Understood it.



How the
Engineer
Designed it.



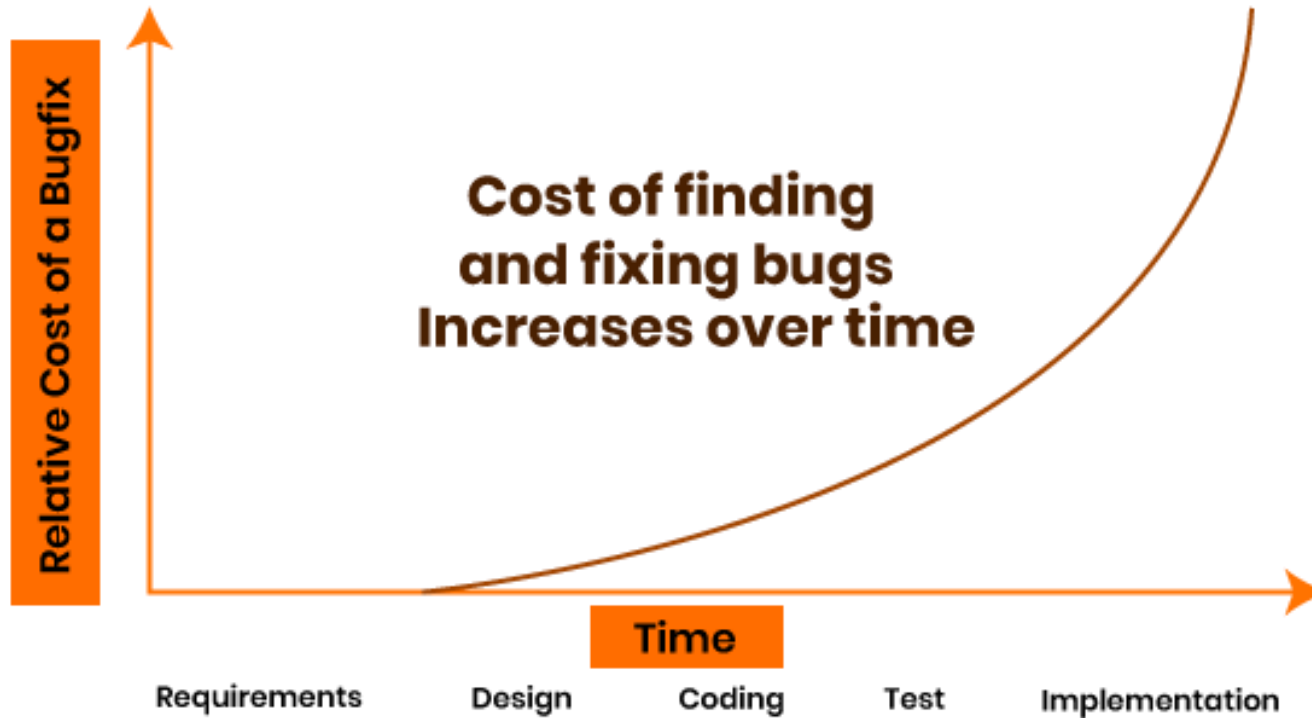
How the
Technician
Built it.



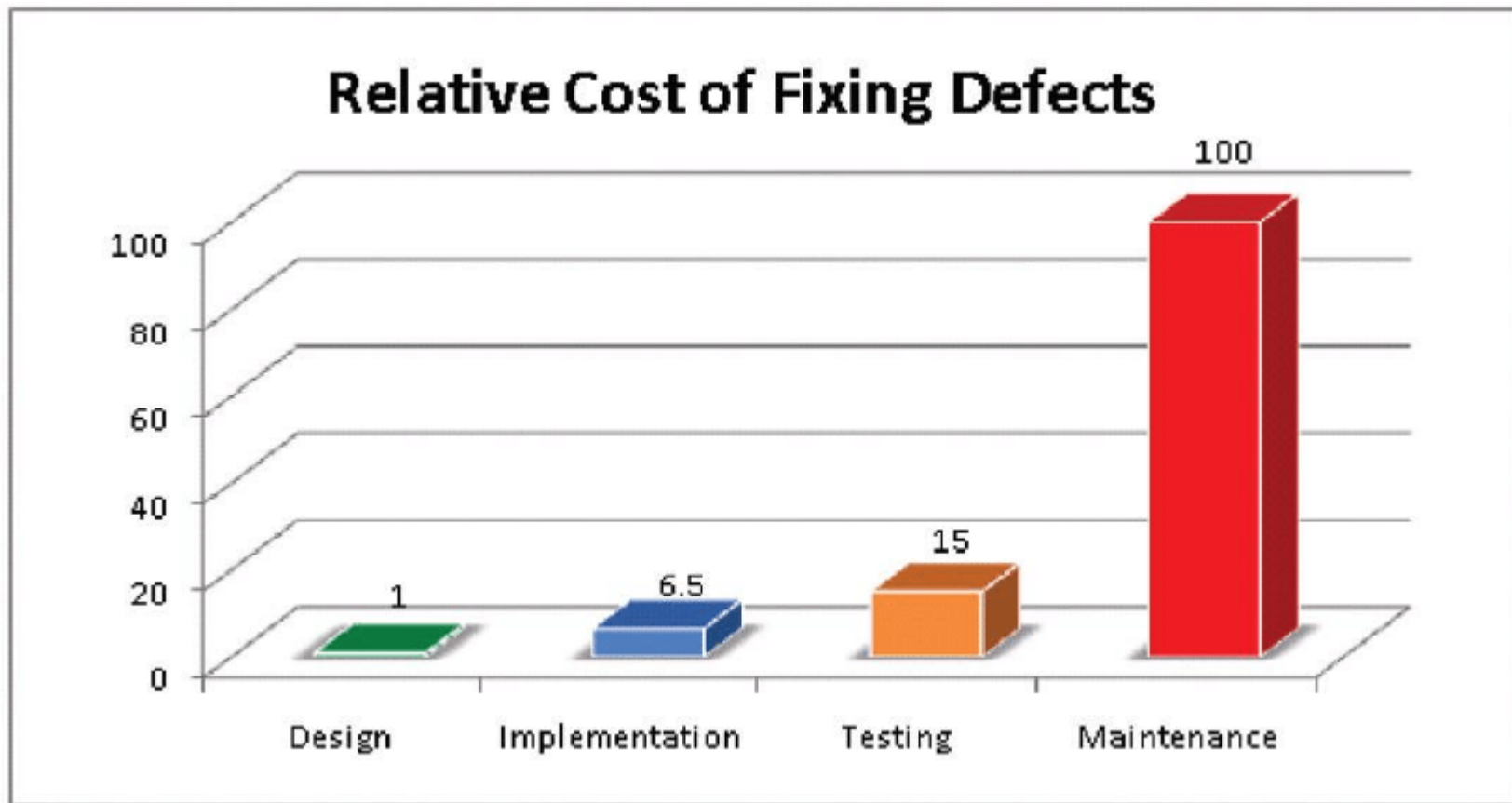
How the
Customer really
wanted it.

Cost of defect

How does it cost to fix a bug?



Cost of defect & Lifecycle phase

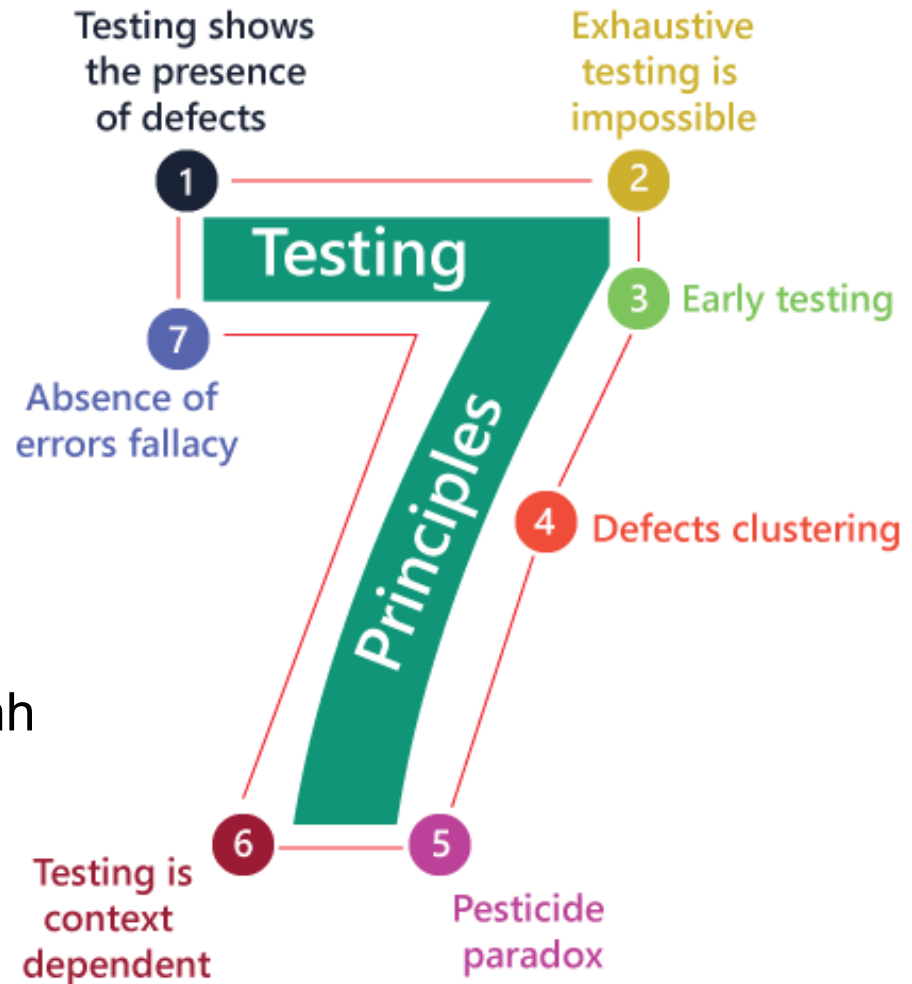


IBM System Science Institute Relative Cost of Fixing Defects

- Chất lượng phần mềm
- Software Testing
- Tại sao phải kiểm thử phần mềm?
- **Nguyên lý trong kiểm thử phần mềm**
- Một số khái niệm liên quan

7 basic principles

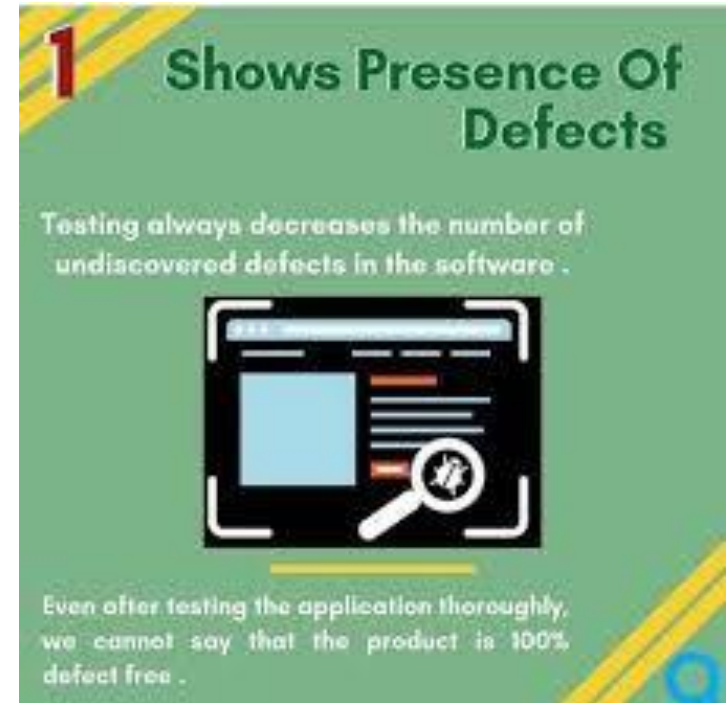
1. Kiểm thử nhằm chỉ ra lỗi
2. Test toàn diện là không thể
3. Kiểm thử càng sớm càng tốt
4. Sự tập trung của lỗi
5. Nguyên lý thuốc trừ sâu
6. Kiểm thử phụ thuộc vào bối cảnh
7. Sự sai lầm về việc không có lỗi



Testing shows presence of defects

No defects found \neq No defects present

- Kiểm thử chỉ ra sự hiện diện của lỗi, chứ không phải phần mềm không có lỗi.
- Kiểm thử viên không cần thiết phải nỗ lực chứng minh rằng lỗi đó không xuất hiện/ bug phát sinh rằng chúng ... vắng mặt.



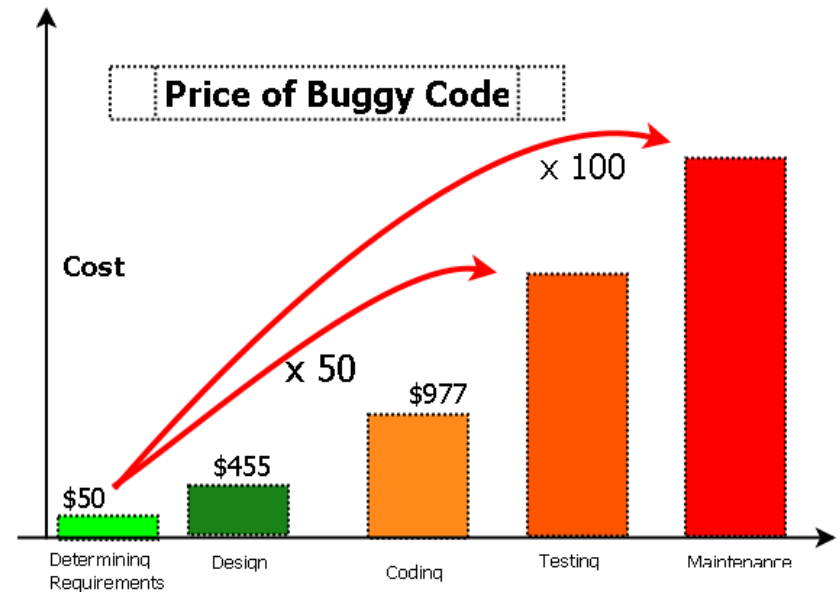
Exhaustive testing is impossible

- Kiểm thử toàn bộ là điều không thể.
- Phân tích rủi ro.
- Phân độ ưu tiên.
- Sử dụng thuần thục các kỹ thuật test, lúc nào dùng kỹ thuật nào → tăng thúc đẩy việc giảm thiểu bug xảy ra.



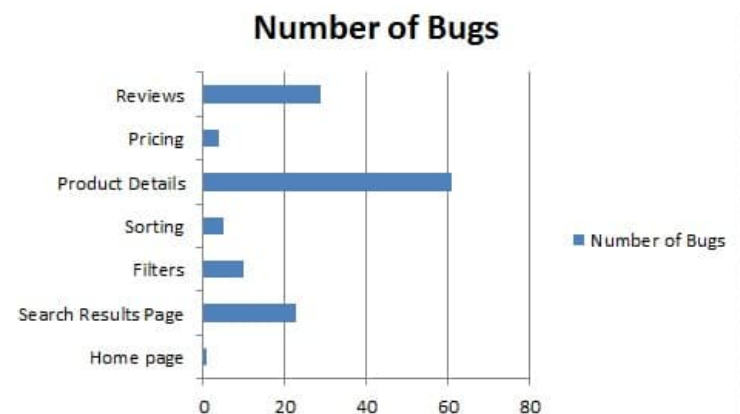
Early testing

- Test sớm giúp tiết kiệm thời gian và tiền bạc
- Static test và dynamic test nên được bắt đầu càng sớm càng tốt
- Shift Left Testing
→ giảm hoặc loại bỏ các thay đổi tốn kém



Defect clustering: defect density

- Lỗi thường được phân bố tập trung
 - Nguyên lý Pareto: 80% số lượng lỗi được tìm thấy trong 20% tính năng của hệ thống.
 - Nguyên tắc tổ gián
 - Nguyên lý 2: không thể → cần analysis (phân tích) và priorities (tính toán mức độ ưu tiên) để quyết định tập trung vào test chỗ nào.
- Defect Clustering sẽ xảy ra hiện tượng "kháng thuốc → hiệu ứng thuốc trừ sâu



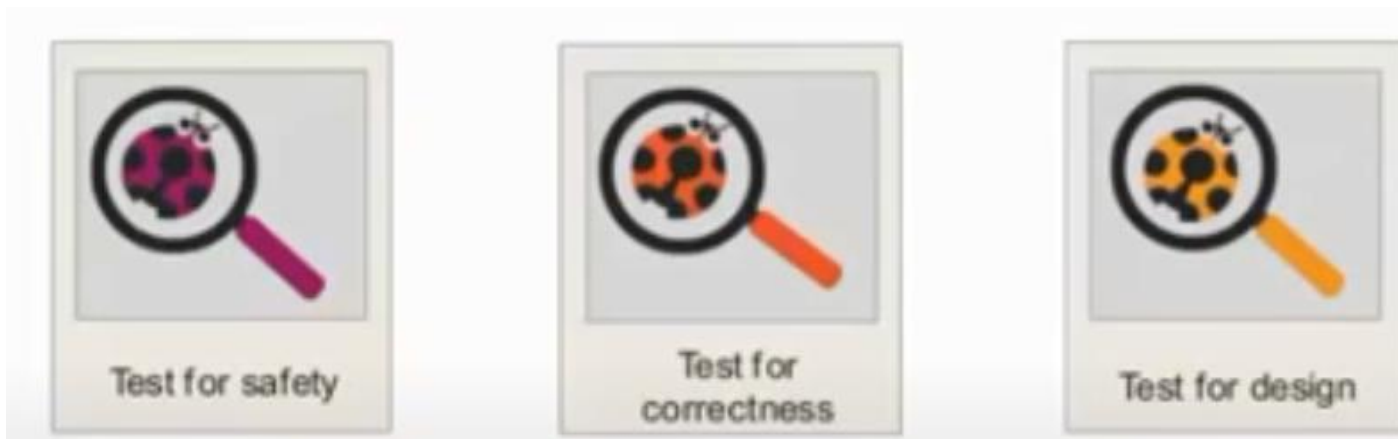
Pesticide paradox

- Đề phòng nghịch lý thuốc trừ sâu.
- Test cases phải được reviewed và sửa đổi thường xuyên, thêm các test cases mới để giúp tìm ra nhiều lỗi hơn.
- Liên tục cải thiện các phương pháp hiện có để kiểm thử được hiệu quả hơn



Testing is context dependence

- Kiểm thử được thực hiện khác nhau trong các bối cảnh khác nhau.
- Sử dụng phương pháp tiếp cận, kỹ thuật, loại kiểm thử khác nhau dựa trên từng loại ứng dụng khác nhau.
- Ví dụ: kiểm thử trong dự án Agile # dự án Waterfall



Absence of error fallacy

- Sự vắng mặt của lỗi là một ảo tưởng.

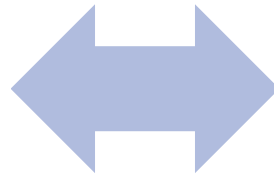


- Chất lượng phần mềm
- Software Testing
- Tại sao phải kiểm thử phần mềm?
- Nguyên lý trong kiểm thử phần mềm
- **Một số khái niệm liên quan**

Một số khái niệm liên quan

- **Testing & Debugging**
- **Verification & Validation**
- **QA, QC, Tester**
- **Mindset của Dev và Tester**

Testing & Debugging



Testing

- Detect a defect
- Done by Tester (Test team)
- Find as many defects as possible

Debugging

- Fix that defect
- Done by Developer (Development team)
- Remove those defects

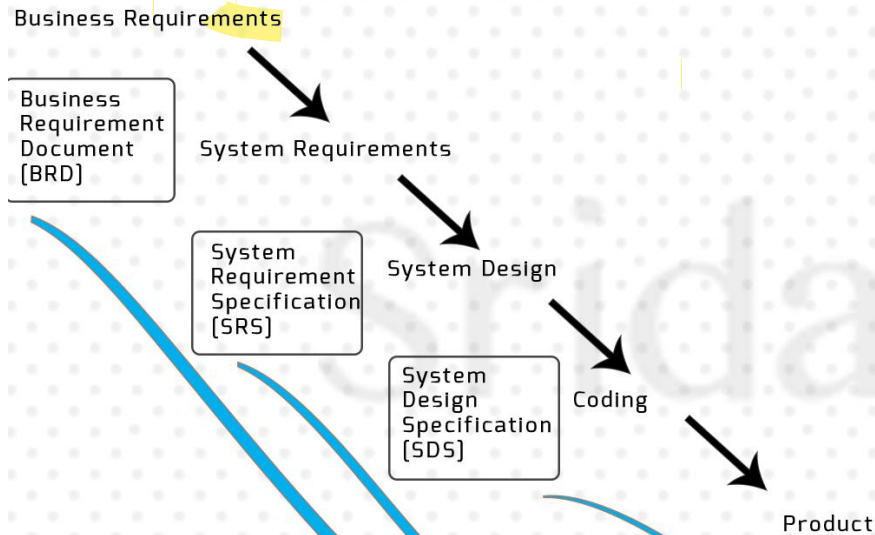
Verification & Validation

"Are we building the **product right**?"

"Are we building the **right product**?"

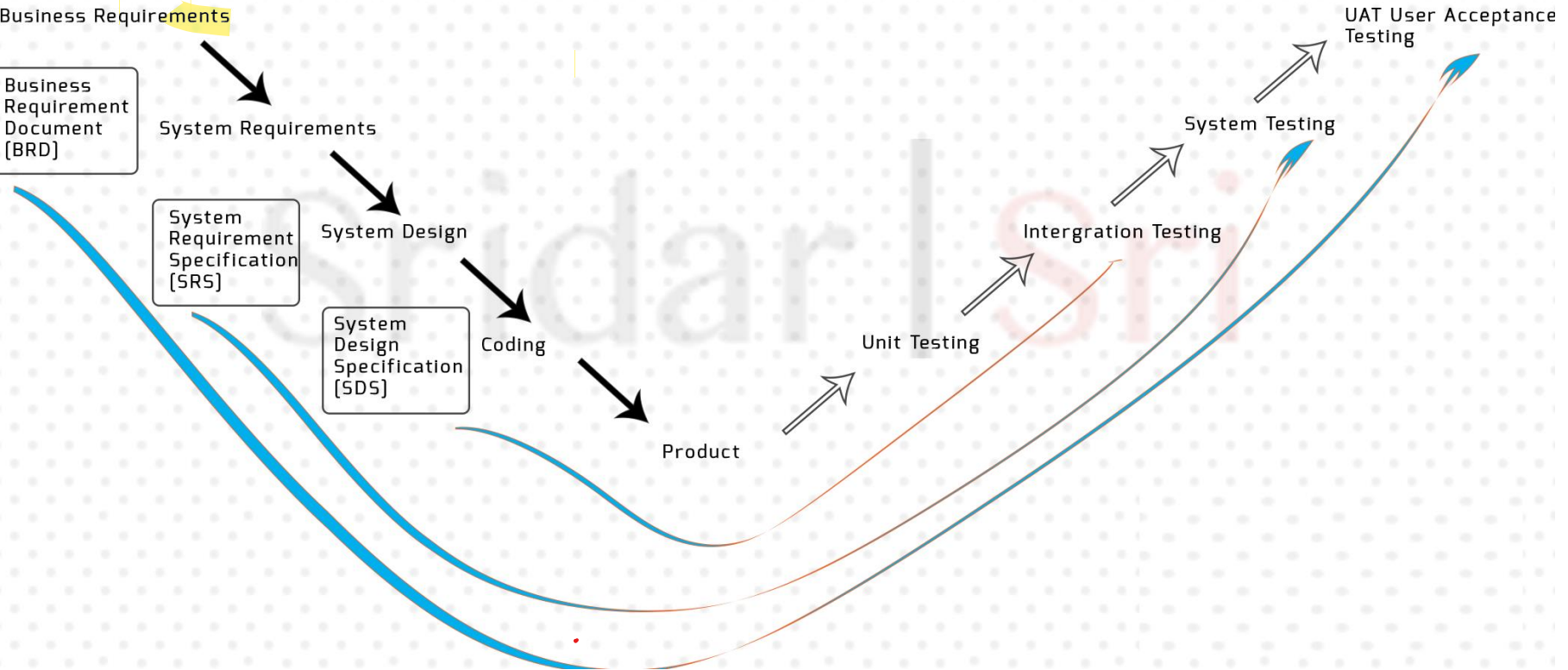
VERIFICATION

Static Testing



VALIDATION

Dynamic Testing



Các khái niệm liên quan - Verification

- **Verification:** Thanh kiểm tra phần mềm là qui trình **xác định** xem sản phẩm của 1 công đoạn trong qui trình phát triển phần mềm có thoả mãn các yêu cầu đặt ra trong công đoạn trước không (Ta có đang xây dựng đúng đắn sản phẩm không?)
- Thanh kiểm tra phần mềm thường là hoạt động kỹ thuật vì nó dùng các kiến thức về các artifacts, các yêu cầu, các đặc tả rời rạc của phần mềm.
- Các hoạt động Thanh kiểm tra phần mềm bao gồm kiểm thử (testing) và xem lại (reviews).

Các khái niệm liên quan - Validation

- **Validation:** Kiểm định phần mềm là qui trình **đánh giá** phần mềm ở cuối chu kỳ phát triển để đảm bảo sự bằng lòng sử dụng của khách hàng (Ta có xây dựng phần mềm đúng theo yêu cầu khách hàng?).
- Các hoạt động kiểm định được dùng để đánh giá xem các tính chất được hiện thực trong phần mềm có thỏa mãn các yêu cầu khách hàng và có thể theo dõi với các yêu cầu khách hàng?
- Kiểm định phần mềm thường phụ thuộc vào kiến thức của lĩnh vực mà phần mềm xử lý.

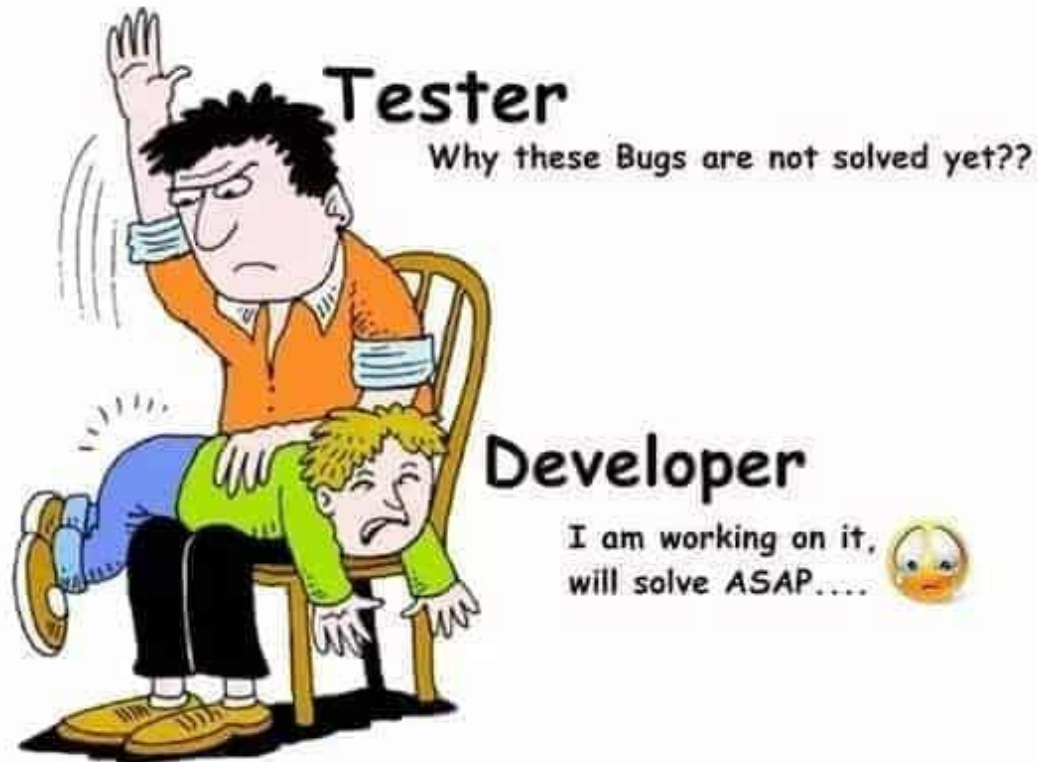
Các khái niệm liên quan

- **QA (Quality Assurance = Kiểm định chất lượng):**
 - Người đặt ra các quy định, nguyên tắc để dự án được thành công.
- **QC (Quality Control = Điều khiển chất lượng)**
 - Người thi thành các quy định, nguyên tắc, đảm bảo sản phẩm cuối cùng đạt đúng các quy định, nguyên tắc mà QA đặt ra.
 - QC: tập trung vào sản phẩm, kiểm thử sản phẩm.
- **Tester (hoặc Test Engineer) = QC Engineer:** kiểm thử, tìm các trường hợp còn thiếu sót hay lỗi so với yêu cầu.



Các khái niệm liên quan

- **Dữ liệu kiểm thử** (test data): Dữ liệu cần cung cấp để phần mềm có thể thực thi để kiểm thử;
- **Kịch bản kiểm thử** (test scenario): Các bước thực hiện khi kiểm thử.



Các khái niệm liên quan

- **Test case:** chứa các thông tin cần thiết để kiểm thử thành phần phần mềm theo mục tiêu xác định, gồm bộ 3 thông tin:
 - Tập dữ liệu đầu vào (**input**): gồm các giá trị dữ liệu cần thiết để thành phần phần mềm dùng và xử lý
 - Tập kết quả kỳ vọng (**output**): kết quả mong muốn sau khi thành phần phần mềm xử lý dữ liệu nhập
 - Trạng thái thành phần phần mềm: được tạo ra bởi các giá trị prefix và postfix.
- **Tập các testcase:** tập hợp các testcase mà ta có ý định dùng để kiểm thử thành phần phần mềm để minh chứng rằng thành phần phần mềm có đúng các hành vi mong muốn.

Nguyên tắc cơ bản về kiểm thử

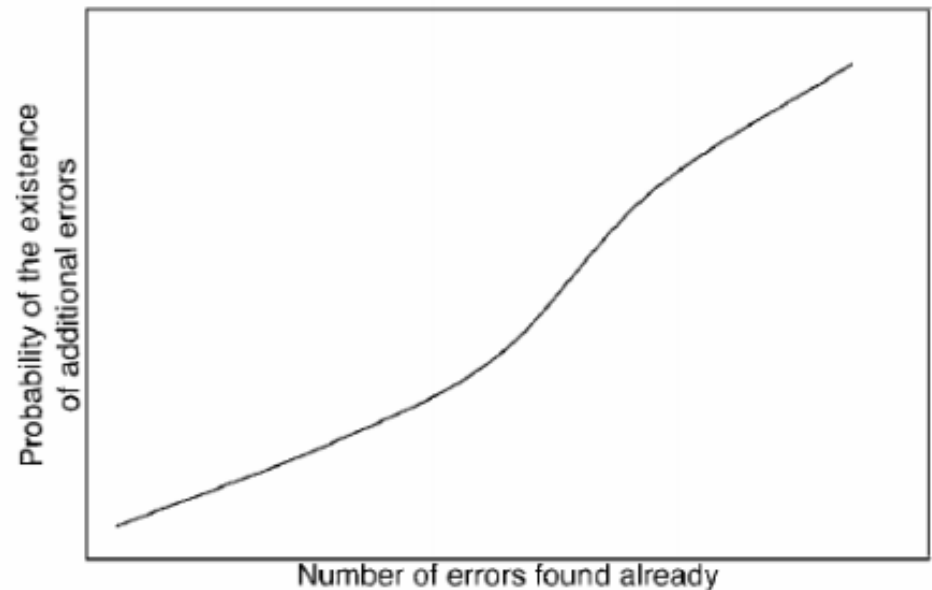
- Thông tin thiết yếu của mỗi testcase là kết quả hay dữ liệu xuất kỳ vọng.
 - Nếu kết quả kỳ vọng của testcase không được định nghĩa rõ ràng, người ta sẽ giải thích kết quả sai (plausible) thành kết quả đúng bởi vì hiện tượng “the eye seeing what it wants to see.”
- 1 test case phải chứa 2 thành phần thiết yếu:
- đặc tả về điều kiện dữ liệu nhập.
 - đặc tả chính xác về kết quả đúng của chương trình tương ứng với dữ liệu nhập

Nguyên tắc cơ bản về kiểm thử

- Việc kiểm thử đòi hỏi tính độc lập: lập trình viên nên tránh việc kiểm thử các thành phần phần mềm do mình viết.
- Các vấn đề tâm lý :
 - Chương trình có thể chứa các lỗi do lập trình viên hiểu sai về đặc tả/ phát biểu vấn đề.
 - Tổ chức lập trình không nên kiểm thử các chương trình của tổ chức mình viết.
 - Thanh tra 1 cách xuyên suốt các kết quả kiểm thử.
- Phải thiết kế đủ các test case cho cả 2 trường hợp: dữ liệu đầu vào hợp lệ và dữ liệu đầu vào không hợp lệ và chờ đợi.

Nguyên tắc cơ bản về kiểm thử

- Xem xét chương trình xem nó không thực hiện những điều mong muốn, xem nó có làm những điều không mong muốn
- Tránh các testcase "throwaway" trừ phi chương trình thật sự là "throwaway".
- Xác suất xuất hiện nhiều lỗi hơn trong 1 section phần mềm tỉ lệ thuận với số lỗi đã phát hiện được trong section đó.



Nguyên tắc cơ bản về kiểm thử

- Không nên lập kế hoạch nỗ lực kiểm thử dựa trên giả định ngầm rằng phần mềm không có lỗi.
- Kiểm thử phần mềm nên bắt đầu từ các thành phần nhỏ đơn giản rồi đến các thành phần ngày càng lớn hơn.
- Kiểm thử theo kiểu vét cạn là không thể.
- Nên hoạch định qui trình kiểm thử trước khi bắt đầu thực hiện kiểm thử.
- Kiểm thử là 1 tác vụ rất thách thức đòi hỏi sự sáng tạo và trí tuệ.

HỎI & ĐÁP



Chương kế tiếp...

QUY TRÌNH KIỂM THỬ PHẦN MỀM