



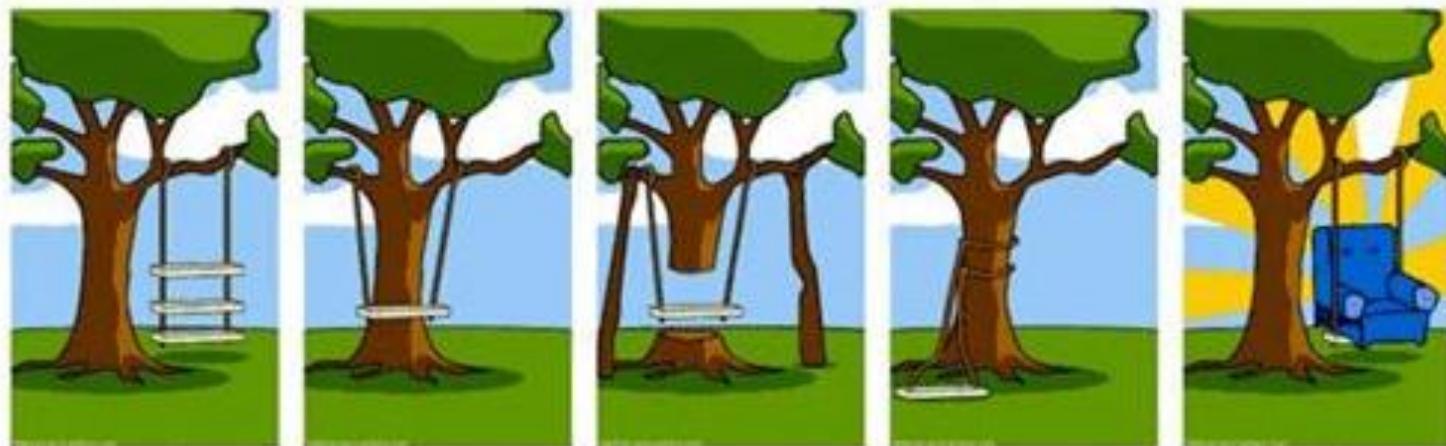
# KIỂM THỬ NÂNG CAO

## BÀI 3: KIỂM THỬ ĐƠN VỊ (P1)

## Nội dung bài học

- 7 chân lý trong kiểm thử phần mềm
- Test Level
- Unit Testing





What the customer said

What was understood

What was planned

What was developed

What was described by the business analyst



What was documented

What was deployed

The customer paid for...

How was the support

What the customer really needed



# 7 Chân lý trong kiểm thử phần mềm

# 7 CHÂN LÝ TRONG KIỂM THỬ PHẦN MỀM



Kiểm thử sẽ tìm ra lỗi hiện diện

Kiểm thử tất cả trường hợp tích hợp là không thể

Kiểm thử sớm

Sự tập trung của lỗi

Nghịch lý thuốc trừ sâu

Kiểm thử theo ngữ cảnh

Sự sai lầm về không có lỗi

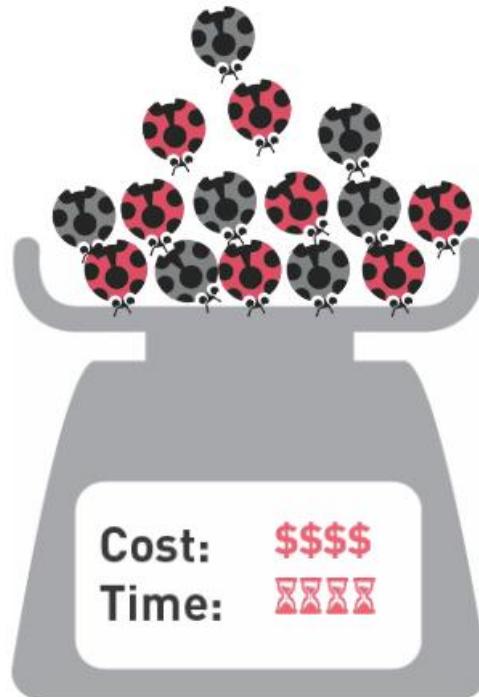
- Kiểm thử chỉ ra có lỗi tồn tại
- Giảm xác suất phát hiện lỗi qua các giai đoạn kiểm thử
- Không tìm thấy lỗi không thể kết luận phần mềm đạt chất lượng 100%



Tất cả thiên nga đều  
màu trắng ?

## #2 KHÔNG THỂ KIỂM THỬ TẤT CẢ

- ❑ Không thể kiểm thử hết tất cả
- ❑ Tập trung vào nội dung cần kiểm thử



Count Vowels & Consonants

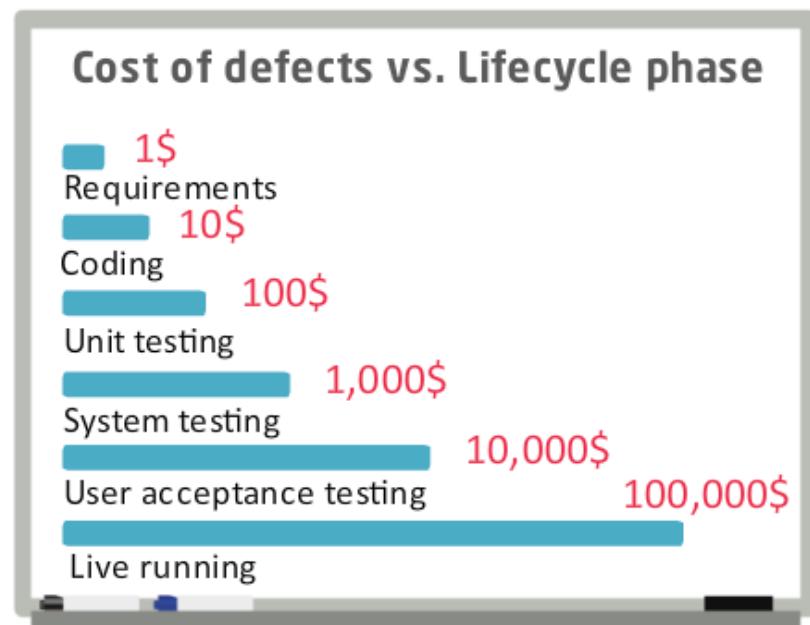
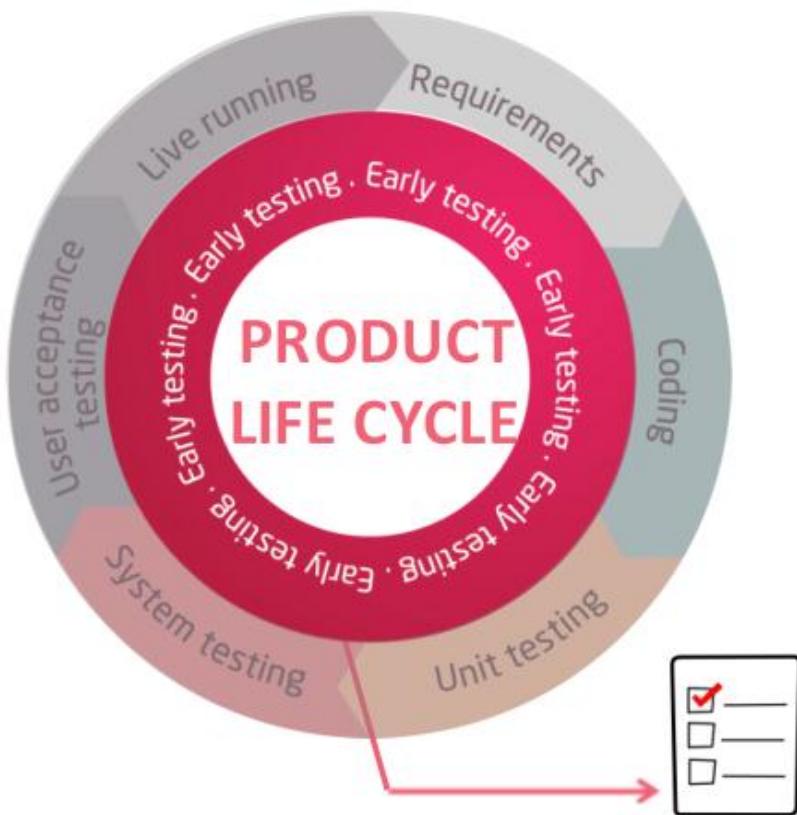
awake

Count Clear

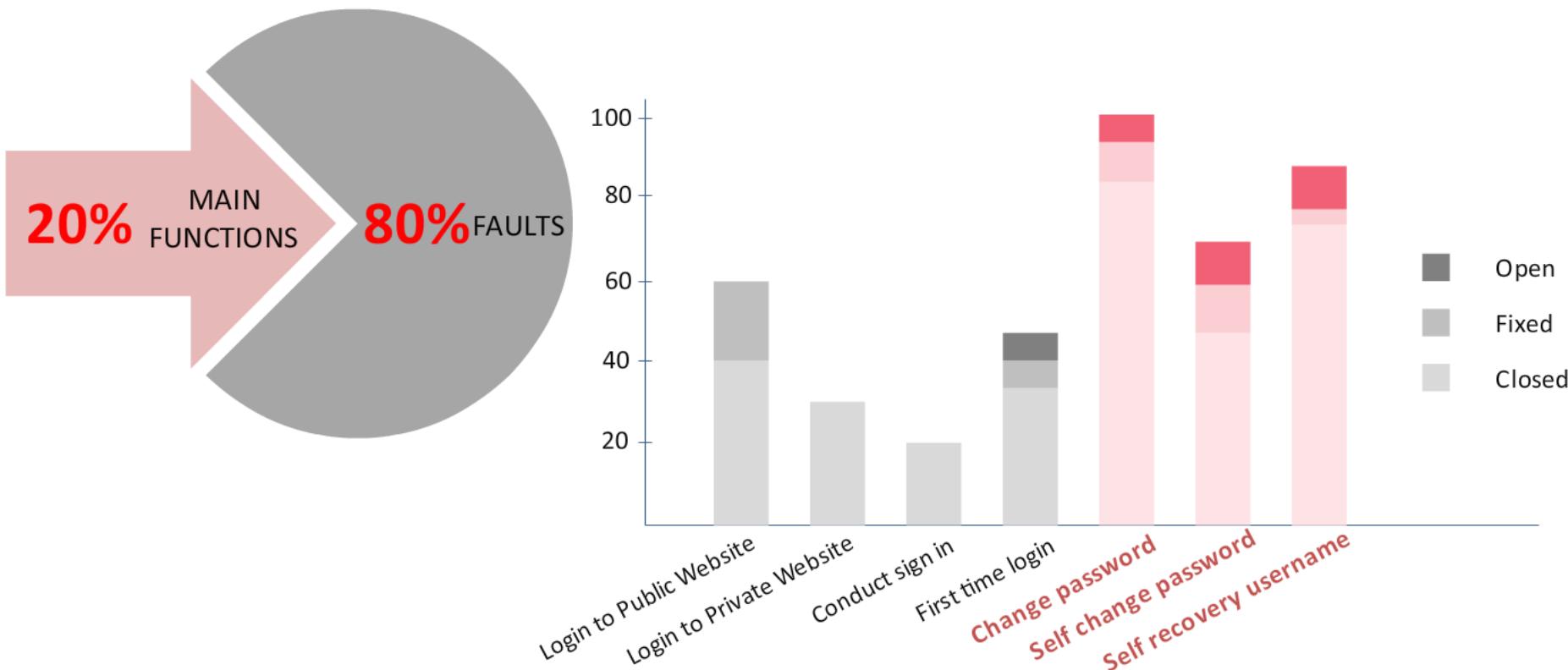
Vowels  
Vowels: aae  
Total: 3

26<sup>5</sup> trường hợp

- ❑ Các hoạt động kiểm thử nên bắt đầu càng sớm càng tốt

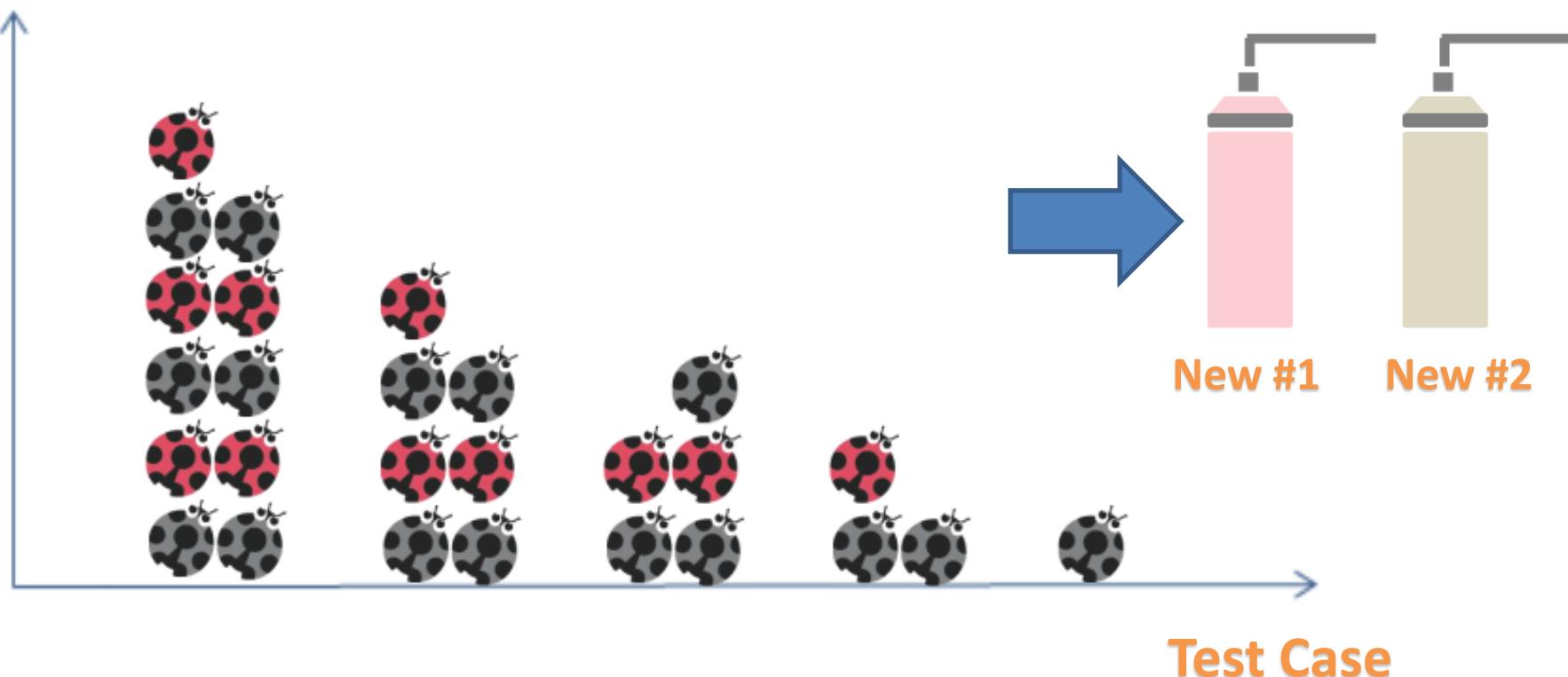


### ❑ Lỗi tập trung ở các chức năng chính



- ❑ Thay đổi phương thức kiểm thử để tìm lỗi mới

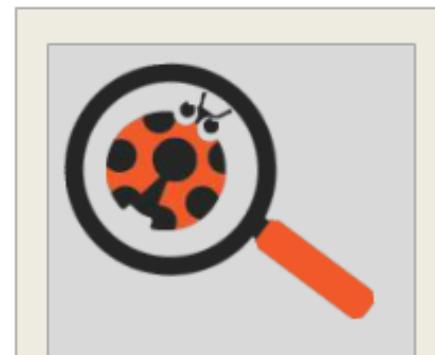
### Lỗi mới



- Kiểm thử phải theo ngữ cảnh của phần mềm



Test for safety



Test for  
correctness



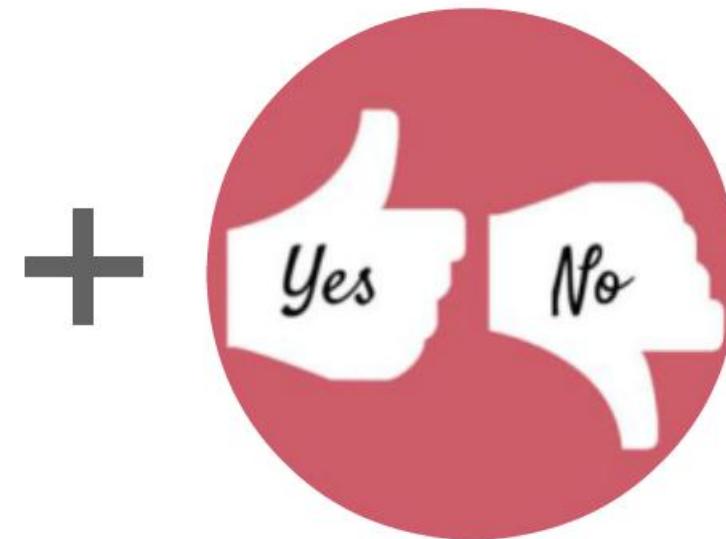
Test for design

## #7 SỰ SAI LẦM VỀ KHÔNG CÓ LỖI

- ☐ Kiểm thử tập trung vào trải nghiệm sử dụng phần mềm



**Finding defects**

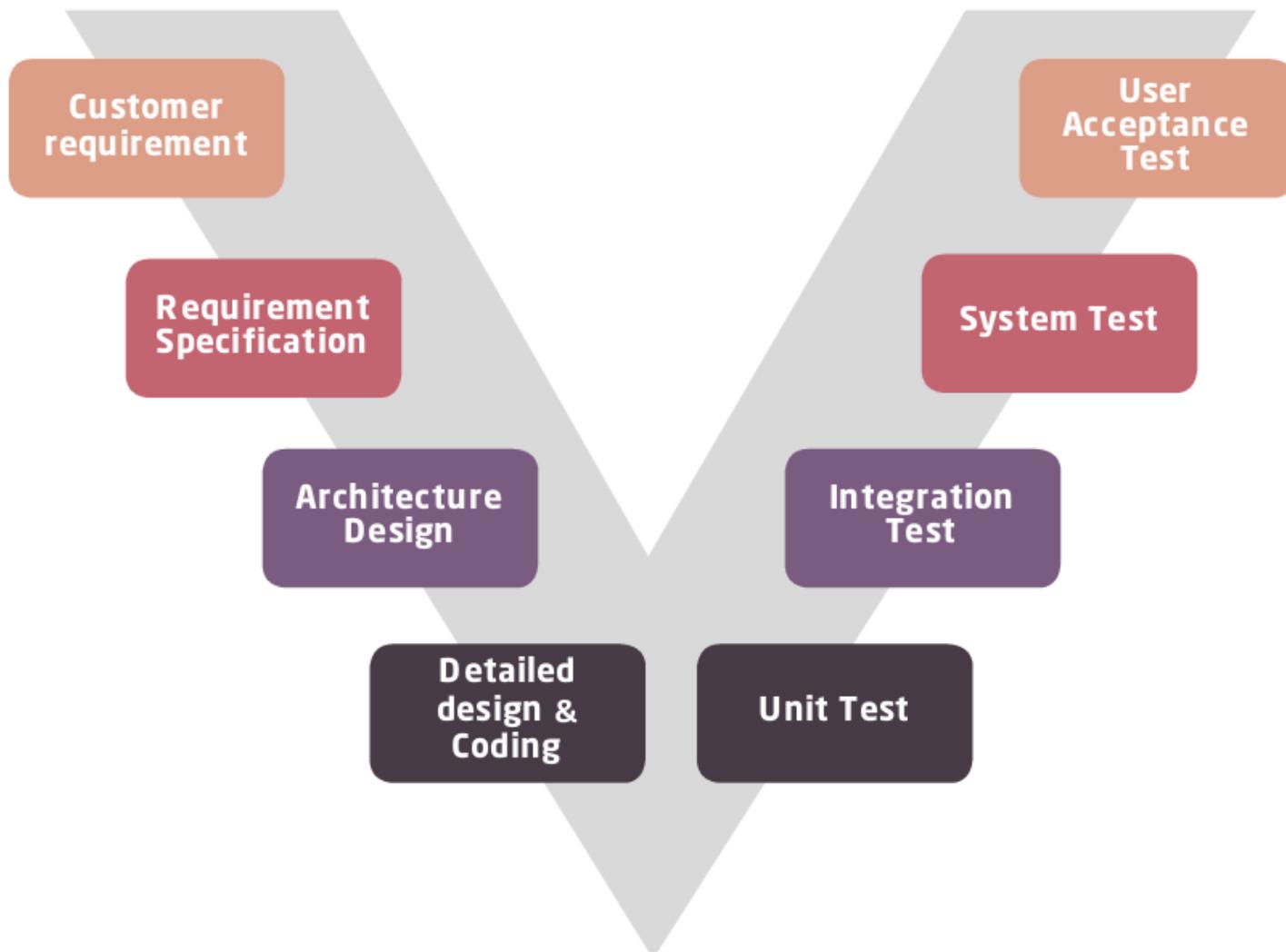


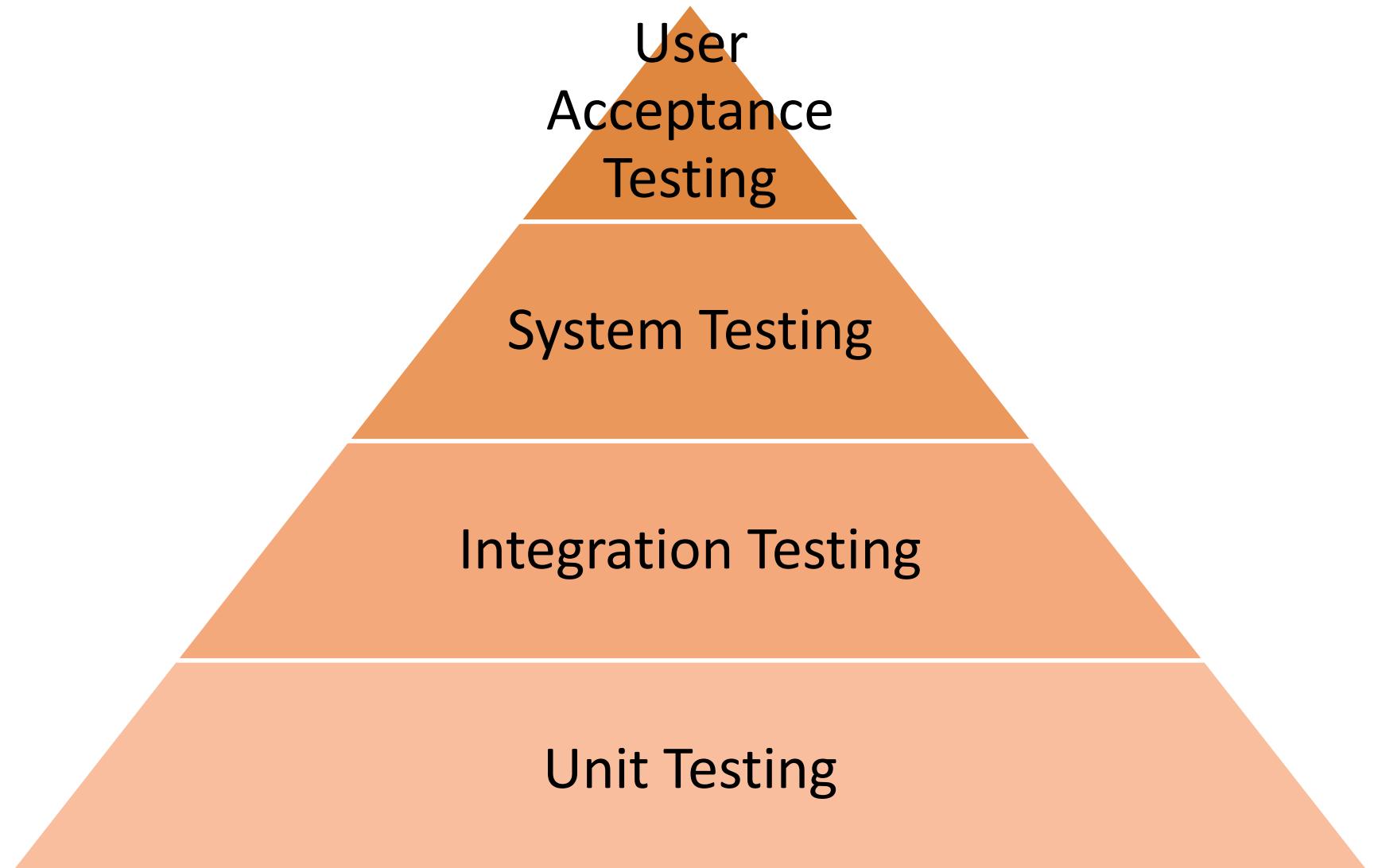
**User Acceptance Testing**



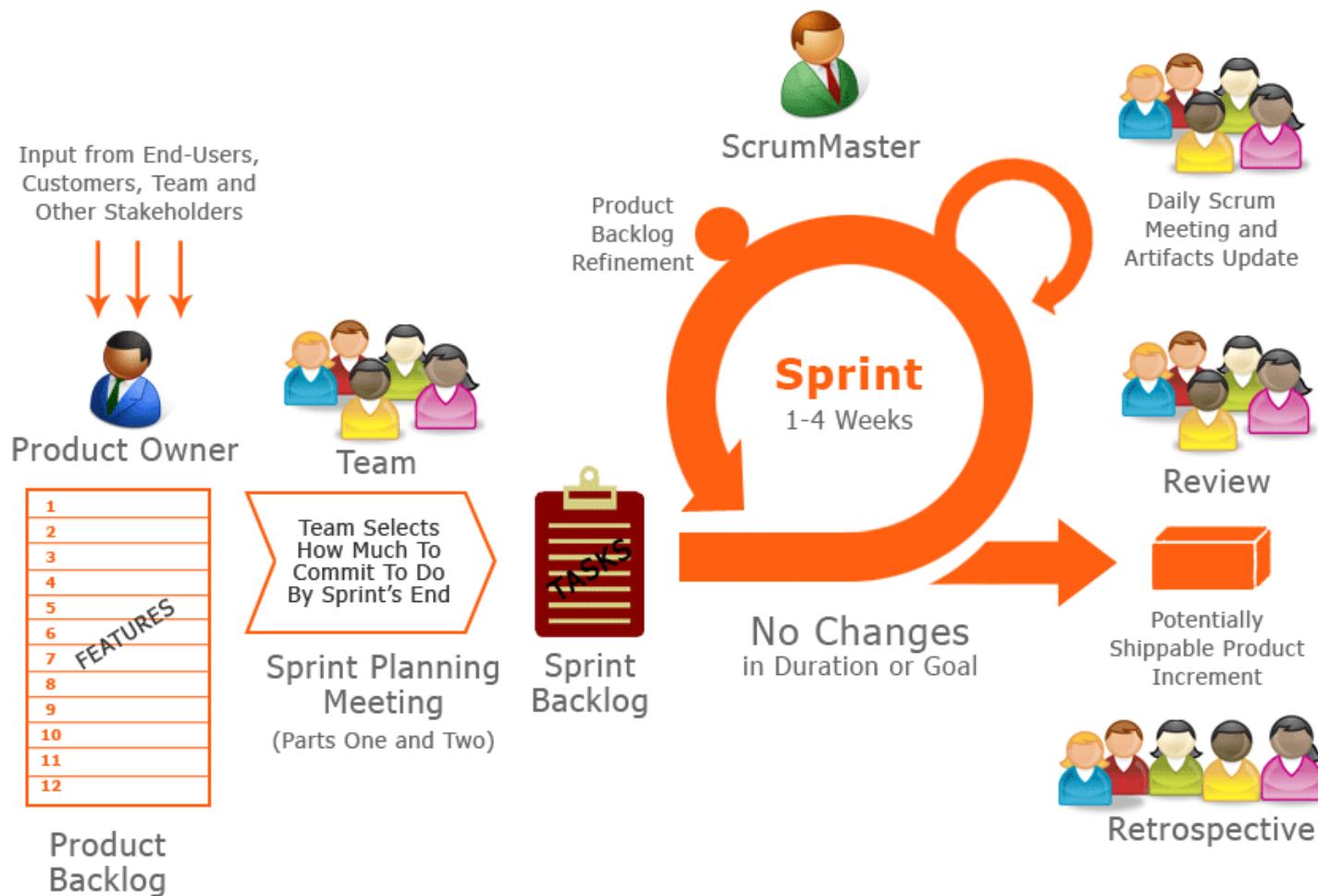
## Test Level

Kiểm thử nâng cao



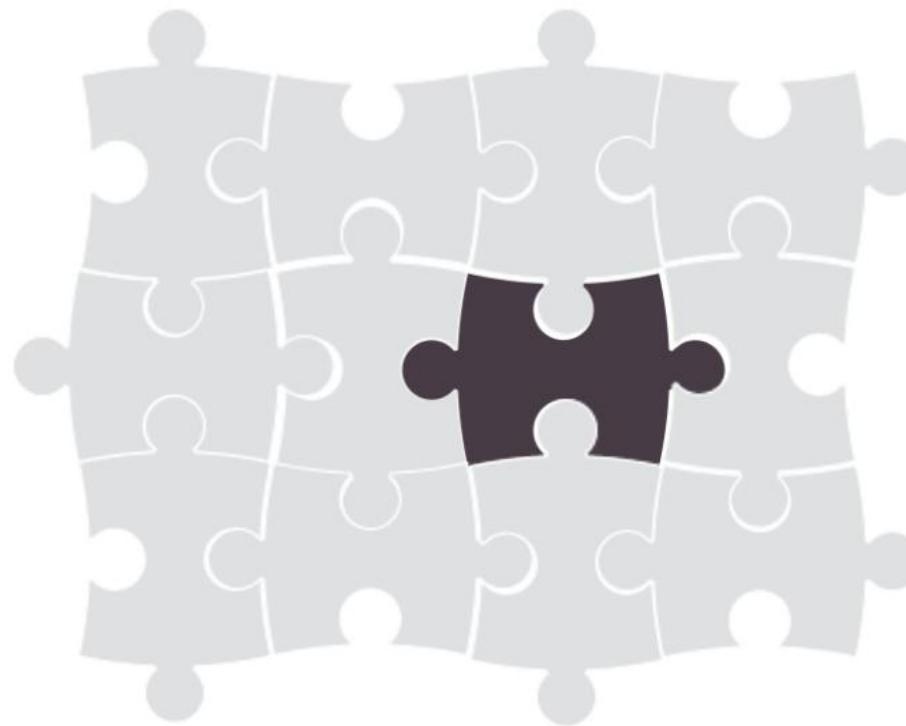


# QUY TRÌNH THỰC HIỆN SCRUM





# Kiểm thử đơn vị Unit Testing



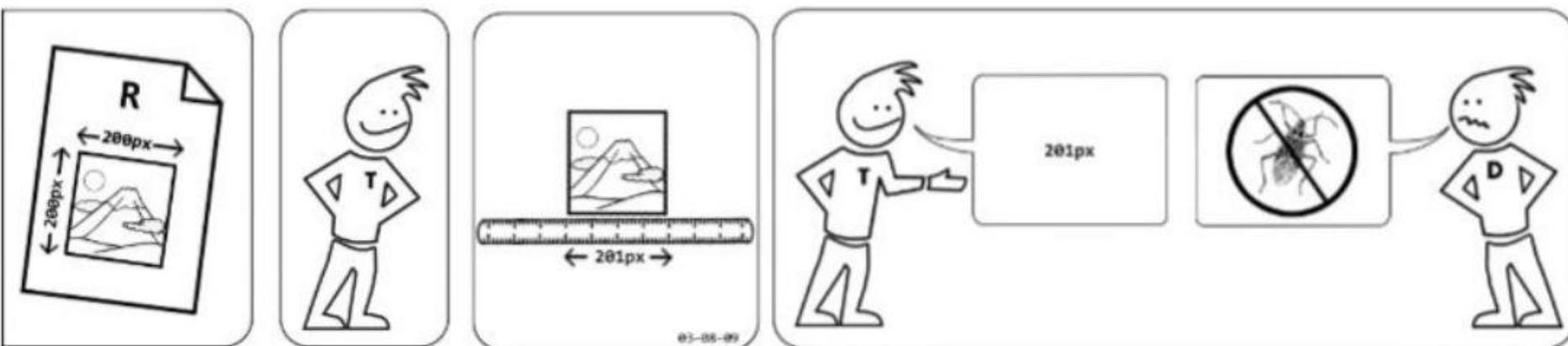
**Quá trình kiểm thử từng đơn vị phần mềm  
xem có hoạt động đúng như thiết kế hay  
không?**

- Một dòng lệnh
- Một hàm, phương thức
- Một module
- Một chương trình

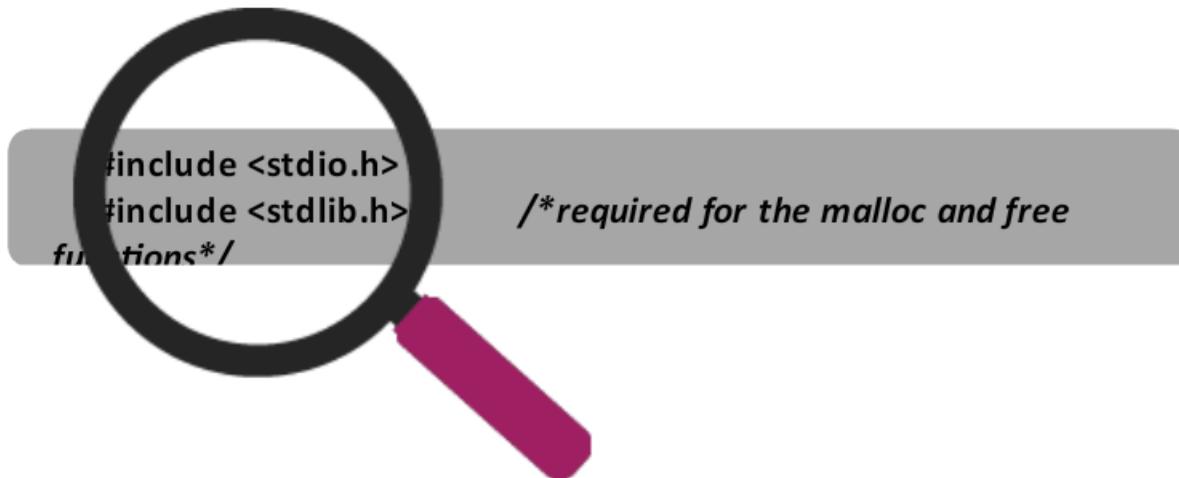


# TẠI SAO PHẢI THỰC HIỆN UNIT TESTING

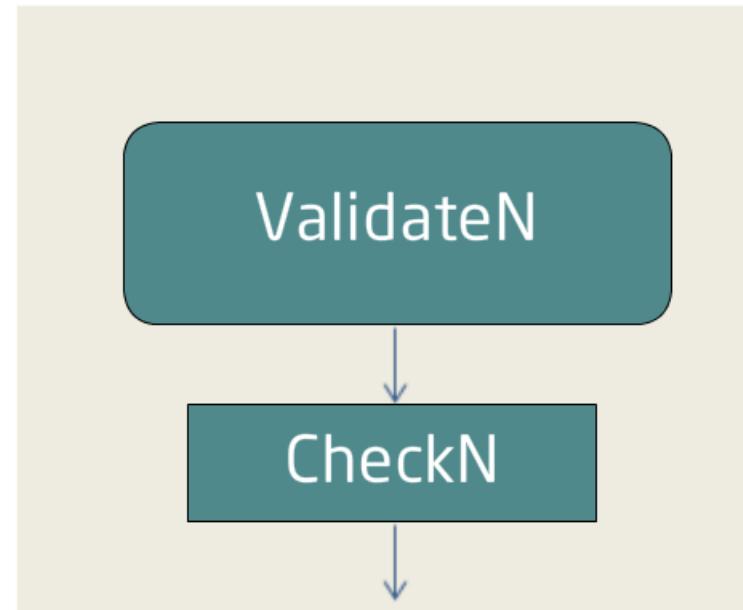
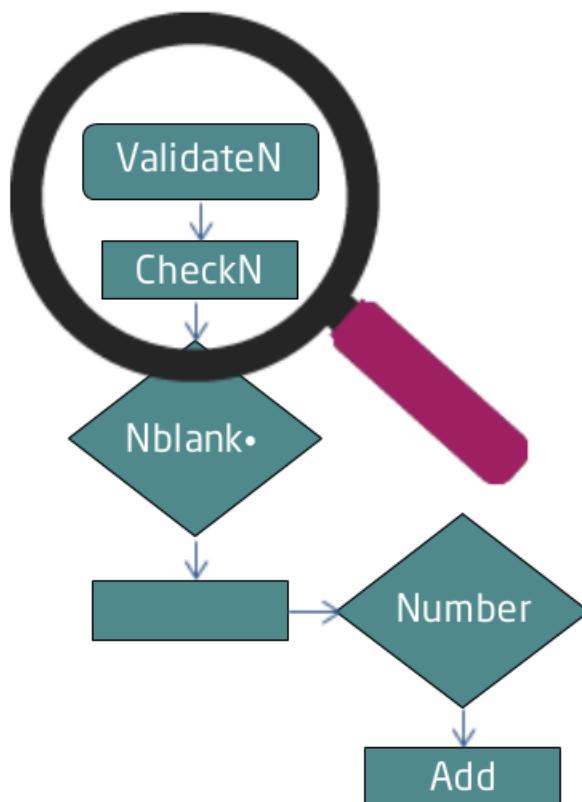
- Phát hiện lỗi nhanh
- Cải tiến quy trình và thiết kế sản phẩm
- Giảm chi phí sửa lỗi



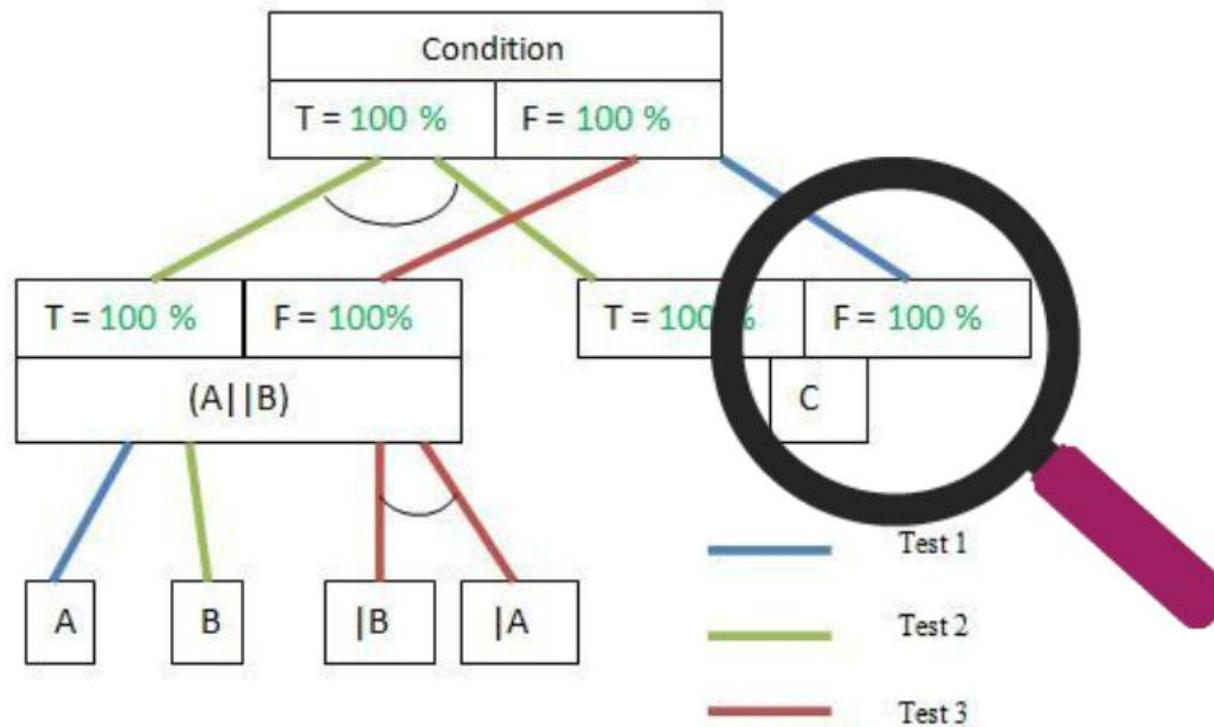
## Kiểm tra dòng lệnh



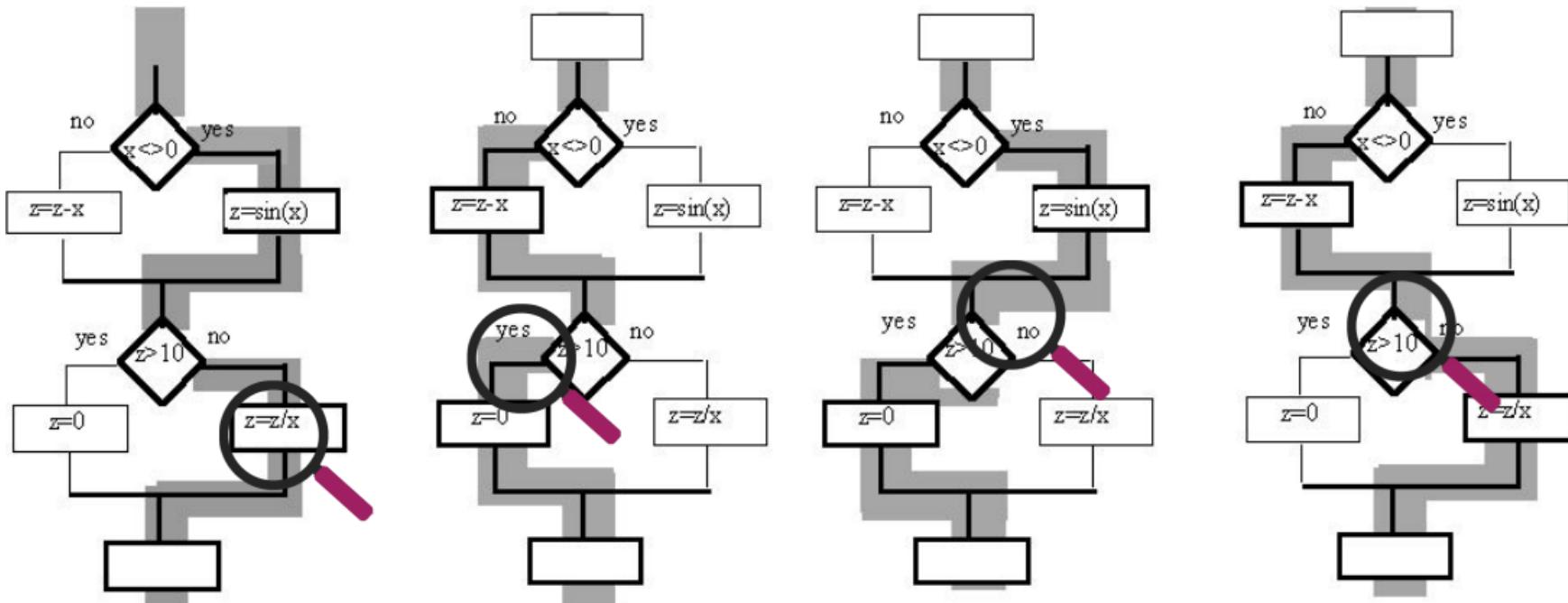
## ❑ Kiểm thử rẽ nhánh



## ☐ Kiểm thử theo điều kiện



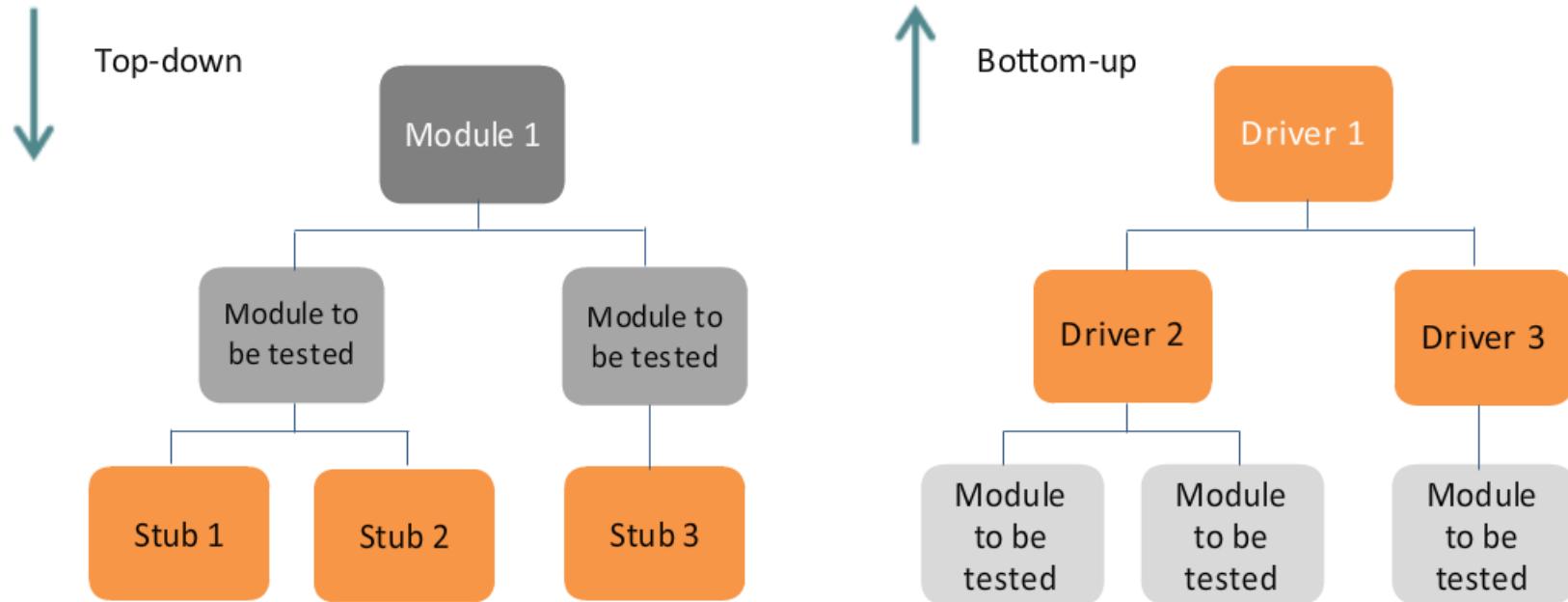
## ❑ Kiểm thử theo đường đi



## ☐ Lập trình viên thực hiện Unit Test

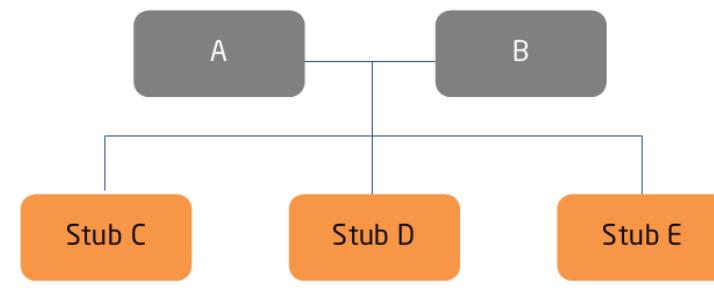


# THỰC HIỆN UNIT TESTING



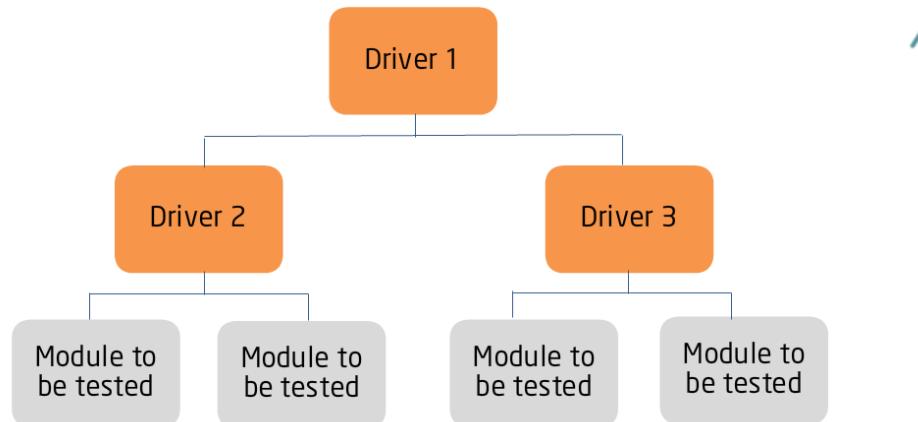
**Stub và Driver là các thuật ngữ thể hiện các thành phần thay thế khi chưa có để kiểm thử nội dung thực hiện**

- ❑ Thay thế các thành phần được gọi trong chương trình



```
void functionWeTest(params...) {  
    .....  
    int p = price(param1);  
    .....  
}  
void price(int param) { //this is the stub  
    return 10; // We don't care what the price is. We  
    just need a value so we can test the other function  
}
```

## ☐ Gọi các thành phần khác



```
void functionThatCallsPrice (params..) { //this is the driver
    int p = price(param1);
    printf("Price is: %d", p);
}

void price(int param) {
    //complex ecuations and DB interogations that
    determin the real price
}
```

## Tổng kết bài học

- ✓ Lập kế hoạch dự án scrum
- ✓ Tầm nhìn sản phẩm
- ✓ Tầm nhìn cấu trúc
- ✓ Quy trình kiểm thử
- ✓ Test planning
- ✓ Test control
- ✓ Test closure





KẾT THÚC