# 1. Kiểm thử là gì

Là 1 quá trình thao tác thực hiện dẫn đến kiểm tra chương trình để:

+ tìm ra lỗi

+ chứng minh chương trình hoạt động đúng

# 2. Kiểm thử quan trọng

Bugs dẫn đến các vấn đề nghiêm trọng:

- thương tích, chết người

- thiệt hại kinh tế, uy tín đến doanh nghiệp

- Tốn thời gian tiền bạc

=> kiểm thử tìm ra lỗi -> đảm bảo chất lượng của phần mềm -> tăng độ tin tưởng và sự hài lòng của khách hàng

# 3. 7 nguyên tắc kiểm thử

## Exhanse testing is not impossible

Kiểm thử toàn bộ là không thể. Thay vì kiểm thử toàn bộ, chúng ta dựa vào đánh giá rủi do, độ ưu tiên, sử dụng kỹ thuật phù hợp.

## Defect clustering

Cụm lỗi: 1 số ít các mô đun chứa hầu hết các lỗi được phát hiện. Khoảng 80% các vấn đề tìm thấy trong 20% các modun

## Pesticide paradox

Nghịch lý thuốc trừ sâu: 1 bộ dl test sd lặp đi lặp lại sẽ không tìm được các lỗi mới -> cần xem xét cập nhật bộ dữ liệu test thường xuyên.

## Testing shows presence of defects

Kiểm thử chỉ ra sự tồn tại của lỗi, không thể chứng minh phần mềm không có lỗi, dẫn đến là giảm lỗi.

## Absence of error fallacy

Quan niệm sai lầm về phần mềm hết lỗi. có thể PM 98% không có lỗi Nhưng vẫn không được sử dụng nếu hệ thống được xác định trên yêu cầu sai

Kiểm thử không chỉ tra ra lỗi, mà còn để ktra xem PM có đáp ứng nhu cầu nghiệp vụ không.

## Early testing

Kiểm thử sớm -> chi phí sửa lỗi thấp hơn

## Testing is context dependent

Tùy thuộc vào ngữ cảnh: cách tiếp cận, phương pháp, kỹ thuật và các loại kiểm thử tùy thuộc vào ứng dụng

# 4. Quy trình kiểm thử phần mềm

Test planning & control

Test implementation & Execution

Evaluating & reporting

Test closure activities

## Test planning & control: lập kế hoạch và kiểm soát

Nhằm kiểm định và mô tả các loại kiểm thử sẽ triển khai và thực hiện gồm 2 hoạt động

+ lập kế hoạch kiểm thử: xác định phạm vi, rủi do, mục đích của hoạt động kiểm thử, phầm mềm kiểm thử, tài nguyên kiểm thử, lên lịch cho các hoạt động phân tích thiết kế kiểm thử, thực thi kiểm thử cũng như đánh giá kết quả, xác định thời gian kết thúc.

+ kiểm soát: theo dõi và ghi lại tiến độ, so sánh kết quả thực hiện với kế hoạch, cung cấp thông tin về kiểm thử tiến hành các hoạt động khắc phục cần thiết và đưa ra quyết định.

## Test analysis & design : phân tích và thiết kế

Nhằm xác định lại các test case và các báo cáo ktra chi tiết

+ kiểm tra lại các thông tin kiểm thử gồm các: yêu cầu tài liệu thiết kế, tài liệu phân tích rủi do…

+ xác định điều kiện kiểm thử

+ thiết kế test case

+ thiết kế môi trường kiểm thử, xác định cơ sở hạ tầng và công cụ cần thiết

## Test implementation & excution: hoàn thiện và thực hiện kiểm thử

- hoàn thiện: hoàn thiện và đánh độ ưu tiên cho các test case và tạo dl test, tạo ra một bộ test từ các test case hoàn thiện và kiểm tra môi trường

- Excution: thực hiện bộ test hoặc từng test case riêng lẻ, đảm bảo theo trình tự thời gian của dự án. Ghi lại kết quả test, phiên bản phần mềm sử dụng, so sánh kết quả test với kết quả mong đợi

## Evaluating & reporting: đánh giá điều kiện kết thúc và báo cáo kết quả

- đánh giá dựa vào tiêu trí :

+ số lượng test case passed

+ tỉ lệ lỗi < x%

+ deadlines

- > xem cần test thêm hoặc có cần thay đổi điều kiện kết thúc không

- viết báo cáo tóm tắt kiểm thử cho các bên liên quan

## Test closure activity: các hoạt động kết thúc kiểm thử

Kết thúc việc kiểm thử khi PM sẵn sàng chuyển giao hoặc dự án bị hủy bỏ

Cần đảm bảo rằng tất cả các sản phẩm được hoàn thiện, tất cả các báo cáo về sự cố được giải quyết. Hoàn thiện và lưu trữ bộ dl test để sử dụng sau này. Rút ra bài học cho các dự án sau này.

# 5. Mô hình phát triển phần mềm

- Thác nước: xác định yêu cầu-> phần tích -> thiết kế -> cài đặt -> kiểm thử -> vận hành và bảo trì

- V: verification- hoạt động review xác định sản phẩm đúng k? validatim: xác định đúng sản phẩm không

- W: chú trọng đến cả kiểm thử tĩnh: đánh giá kiểm tra, phân tích code

- Agile : linh hoạt, cải tiến

# 6. Level testing: có 4 level

## unit test

cô lập từng thành phần của chương trình -> tìm bugs, chứng minh các bộ phận riêng lẻ chính xác về các yêu cầu của chức năng.

## Intergration testing

- Phát hiện lỗi giao tiếp giữa các thành phần/ lỗi từng thành phần nếu có:

- Thành phần có thể là: module, Ứng dụng riêng lẻ, ứng dụng client server trên 1 mạng

-> tích hợp các unit riêng lẻ thành system -> ktra mức hệ thống

## System test

Kiểm thử hệ thống

Đánh giá phần mềm có tuân thử theo các yêu cầu đưa ra hay không

## Acceptant Test

Kiểm thử chấp nhận , kiểm thử hệ thống về khả năng chấp nhận được -> đánh giá phần mềm có đáp ứng được yêu cầu nghiệp vụ, có thể chấp nhận bàn giao không

Khách hàng thực hiện

- anpha: kiểm thử ngay tại nơi phát triển phần mềm, lập trình viên ghi nhận lỗi/phản hồi, đến lên kế hoạch sửa lỗi

- beta: gửi đến chỗ người dùng để kiểm nghiệm

# 7. Các loại kiểm thử (4) TEST TYPE

## function testing

Test chức năng: là một loại **kiểm thử hộp đen**, thực hiện bằng cách nhập ktra kết quả đầu ra, ít quan tâm đến cấu trúc bên trong của app

Test case dựa trên đặc tả của app phần mềm, thành phần đang test

Test những gì?

- mainline functions: test chức năng chính

- basic usability: user có dễ dàng thực hiện các thao tác không

- Error conditions: ktra xem nội dung thông báo lỗi thực hiện đúng k

- 4 báo cáo: xác định dữ liệu test -> xác định kết quả -> thực hiện test -> so sánh

## Test phi chức năng

Tập chung vào khía cạnh phi chức năng. Ví dụ: app làm việc trong điều kiện bình thường như thế nào? Hệ thống xử lý sao khi có nhiều người dùng truy cập đồng thời? tính bảo mật?

- một vài loại kiểm thử phi chức năng

+ kiểm thử hiệu năng

+ load testing: kiểm thử tải lượng

+ kiểm thử áp lực

+ kiểm thử tính dễ sử dụng

+ kiểm thử bảo mật

+ kiểm thử phần mềm trên các môi trường khác nhau

## Structure testing

Test cấu trúc hệ thống/ thành phần => **kiểm thử hộp trắng**

## Regrestion testing

Kiểm thử quy hồi: xác minh rằng phần mềm đã được phát triển, kiểm thử từ trước vẫn hoạt động chính xác sau khi khát triển hoặc giao tiếp với các phần mềm khác

# 8. Kỹ thuật kiểm thử hộp đen

- tập trung kiểm tra hệ thống làm gì? **Không** ktra hệ thống làm như thế nào

- tập trung vào chức năng của hệ thống

- sử dụng trong: test tích hợp, test hệ thống, test chấp nhận

## Kỹ thuật

1. phân vùng đối tượng: chia dl đầu vào thành các vùng tương đương nhau. Tất cả các giá trị trong một vùng tương đương sẽ cho 1 kq đầu ra giống nhau.

2. phân tích giá trị biến: tìm hiểu ranh giới giữa các điều kiện hợp lệ và không hợp lệ, check cận trên, cận dưới các điểm danh giới.

3. Bảng quyết định

4. Sơ đồ chuyển trạng thái – 4 phần: các trạng thái, sự dịch chuyển trạng thái, sự kiện kích hoạt trạng thái, kết quả của sự dịch chuyển

5. pairwise testing: cặp thử nghiệm

# 9. Cấu trúc 1 file test case

## Sheet cover

Thông tin dự án, thông tin sửa đổi, và version tài liệu

## Sheet testcase list

Thông tin môi trường test , danh sách các chức năng và link với sheet test case tương ứng

## Sheet test case

Test case của từng function

## Test report

Kết quả test của từng function, chỉ rõ coverage

# 10. Tổ chức các test case

## 1. nhóm testcase:

- group: tên chức năng lớn

- sub-group: tên chức năng nhỏ

- testcase: các testcase của chức năng nhỏ

## 2. cách chia nhóm

### access right

+ user nào được phép truy cập

+ how to truy cập

### UI:

+ số lượng item

+ vị trí font chữ, size, màu

+ trạng thái của các item: disable/ enable

+ check giá trị default

+ check giá trị store trong các combobox, listbox

### Validation:

+ check miền giá trị được phép

+ check miền giá trị không được phép

### Business:

Check nghiệp vụ chức năng

# 11. Testcase hiệu quả

- các bước đơn giản, rõ ràng, dễ hiểu

- testcase cần hợp lệ, đơn giản, vắn tắt

- testcase cần truy vết được

- testcase cần bảo trì được dễ dàng

- chuẩn bị dữ liệu test đa dạng

- chuẩn bị test case cho các case phi chức năng

# 12. bugs

Error / mistake: hoạt động của con người dẫn đến kết quả sai

Defect / bug / fault: sự hiện diện của lỗi tại thời điểm thực thi phần mềm

Failure ( thất bại): sự khác biệt giữa kết quả thực tế và kết quả mong đợi

Con người gây ra mistake / error trong code, tài liệu => dẫn đến bug trong software dẫn đến software bị failure

# 13. Các loại bug

- functional bug: lỗi nghiệp vụ, chức năng

- visual: giao diện

- content: lỗi đánh máy, ngữ pháp, lỗi bản địa

- performance: hệ thống chậm, treo

- crash: hệ thống tắt đột ngột

# 14. Vòng đời lỗi

Tester tìm thấy lỗi

Trạng thái : new

Chuyển bug cho quản lý dự án

Xem lỗi có hợp lệ không?

có

không

Có nằm trong phạm vi không?

Trạng thái : rejected (từ chối)

có

không

Lỗi này có trước đó chưa

Trạng thái re-open

Trạng thái: closed

không

có

Test case có pass không

Trạng thái : fixed

Tester test lại

Tiến hành đưa cho dev fix

Trạng thái = in-progress

chưa

Trạng thái : trùng lăp

Rồi

Trạng thái : Deferred (hoãn)

# 15. thông tin của một bug report

- tên

- tiền điều kiện

- các thao tác / các bước

- kết quả mong muốn

- kết quả thực tế

- attachment

# 16. Mô tả lỗi trên redmine

-tên

- màn hình + môi trường

- precondition

- step

- actual result

- expected result

# 17. test plan

Là tài liệu mô tả chiến lược kiểm thử, mục tiêu, lịch trình, ước lượng, và khả năng cung cấp nguồn nhân lực để ktra

=> giúp xác định nỗ lực cần thiết để xác nhận chất lượng của ứng dụng đang được kiểm thử.

- những người ngoài nhóm kiểm thử phát triển

- quản lý doanh nghiệp, khách hàng hiểu chi tiết về kiểm thử

- hướng dẫn suy nghĩ của chúng ta

- các khía cạnh quan trọng như: ước tính kiểm thử, phạm vi, chiến lược kiểm thử thực được ghi lại trong testplan

\* bao gồm:

- Phân tích sản phẩm

- xác định mục tiêu, tiêu trí kiểm thử

- hoạch định nguồn nhân lực

- lên kế hoạch môi trường kiểm thử

- lịch trình và dự đoán

- xác định kiểm thử

\* ước lượng:

- Resure: bao gồm con người, thiết bị, csvc, kinh phí

- Time: mỗi da đều có dealine

- ? skill

- lost

# 18.

- lỗi phát hiện càng muộn chi phí lớn vì có những lỗi sẽ thực hiện lại từ khâu thiết kế, rồi code lại

- static testing: thực hiện ở giai đoạn đầu, khi mà ta mới chỉ có các tài liệu yêu cầu, thiết kế mà chưa tiến hành viết code hay thực thi, thì việc ktra là chưa khả thi, nó đòi hỏi phải duyệt lại các yêu cầu và đặc tả để ktra tính logic mà không cần chạy chương trình

- dynamic testing: được thực hiện sau khi có code đã được building . dùng máy chạy chương trình để ktra trạng thái hoạt động của chương trình

- ktra dựa trên các testcase = thực hiện kiểm tra nhập giá trị đầu vào, trường hợp test thực tế và kết quả mong muốn

# 19. White box

Truy cập vào cấu trúc dl và giải thuật bên trong chương trình gồm: giao diện, API, bao phủ, lệnh, gán lỗi, hoán chuyển, static

Gray: truy cập cấu trúc dữ liệu và giải thuật bên trong nhưng chỉ được kiểm tra ở mức người sử dụng hay blackbox, việc thao tác dữ liệu ra vào là không rõ ràng.

# 20. Thứ tự test web

- phân trang theo top, button, right, left test trên từng khu vực

- top phần header có nhiều menu đến function: add, list, edit,..

- mệnh đề test