

## BÀI 1 - TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

## NỘI DUNG

- Tại sao phải kiểm thử phần mềm ?
- Kiểm thử phần mềm(Testing) là gì ?
- Phân biệt giữa kiểm soát và đảm bảo chất lượng
- Phân biệt Kiểm thử(Testing) và Gỡ lỗi (Debugging)
- Phân biệt Xác minh(Verification) và Xác nhận (Validation)
- Phân biệt Bug, Error, Fault, Failure
- 7 nguyên lý cơ bản của Testing
- Các giai đoạn trong quy trình kiểm thử phần mềm
- Tâm lý học trong kiểm thử phần mềm



## TẠI SAO PHẢI KIỂM THỬ PHẦN MỀM?

- Các vấn đề liên quan đến hệ thống phần mềm:
  - Phần mềm là 1 phần của cuộc sống hiện đại, từ những phần mềm nghiệp vụ đến những sản phẩm phục vụ con người.
  - Hầu hết mọi người đều có lúc trải qua việc phần mềm làm việc ko đúng mong đợi.
  - Phần mềm làm việc ko chính xác có thể dẫn đến nhiều rắc rối, ví dụ như mất tiền, thời gian, quan hệ, hoặc mạnh hơn là gây ra chấn thương và cái chết.

## TẠI SAO PHẢI KIỂM THỬ PHẦN MỀM? ...

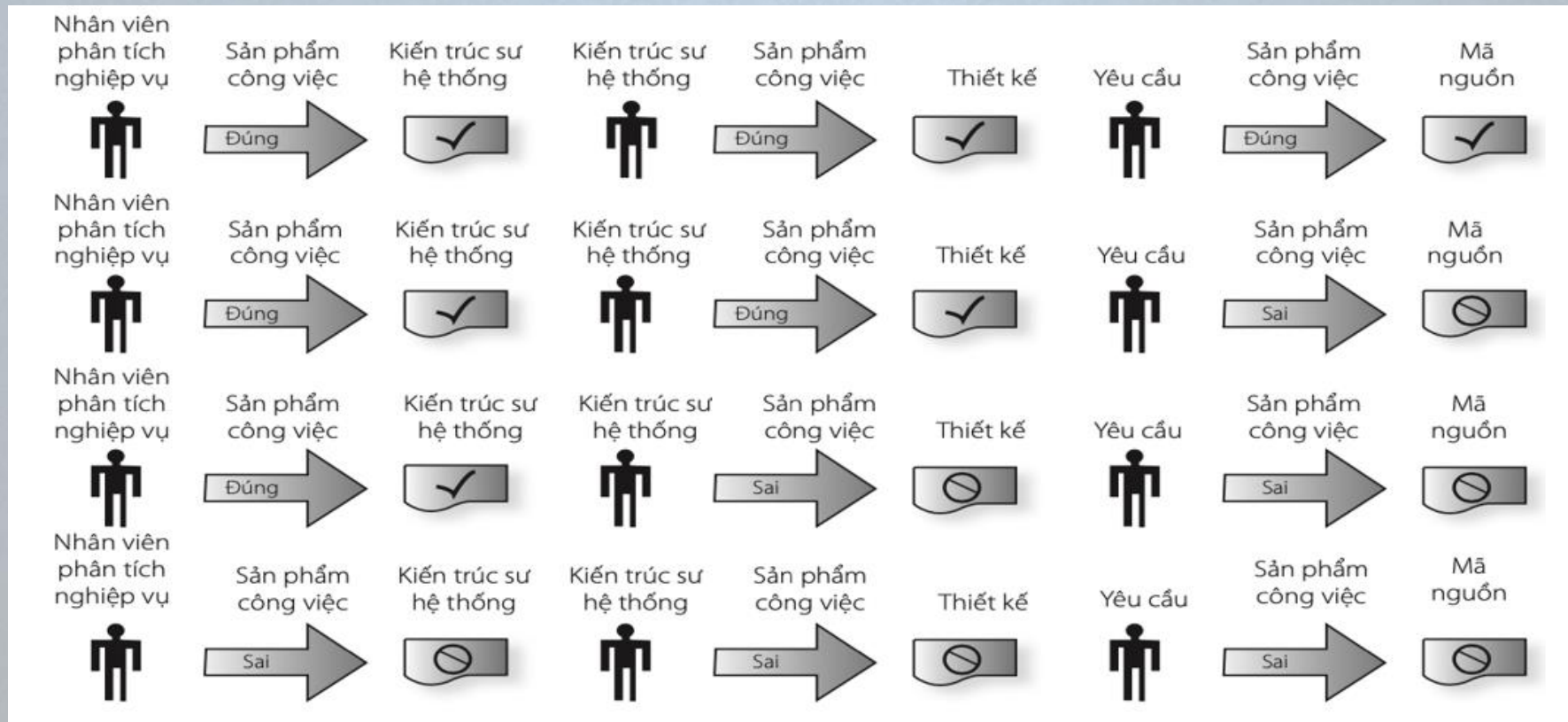
- Tại sao lại xảy ra lỗi phần mềm
  - Phần mềm được viết bởi con người
  - Con người giới hạn bởi kiến thức và kỹ năng
  - Con người tạo ra nhầm lẫn
  - Chịu sức ép rất lớn để bàn giao sản phẩm đúng hạn
  - Thiếu thời gian để kiểm tra
  - Hệ thống chưa hoàn thành





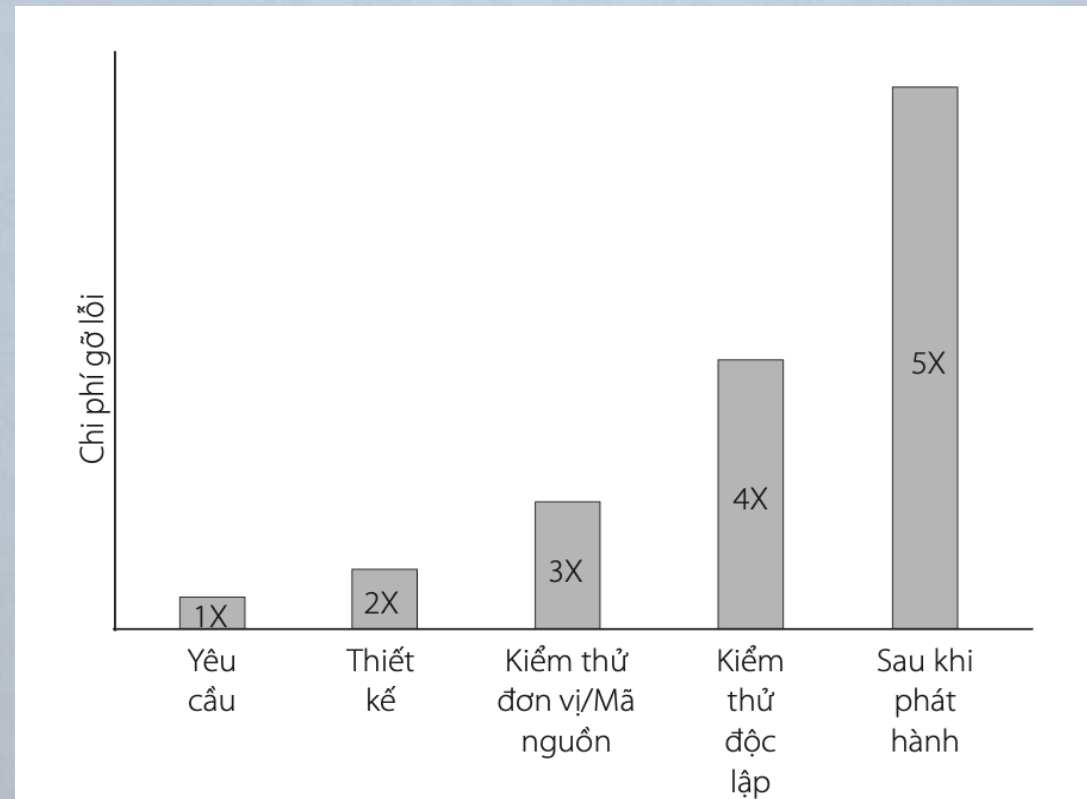
## TẠI SAO PHẢI KIỂM THỬ PHẦN MỀM? ...

- Lỗi phát sinh trong giai đoạn phát triển phần mềm



## TẠI SAO PHẢI KIỂM THỬ PHẦN MỀM? ...

- Tăng trưởng theo cấp số nhân chi phí khắc phục lỗi



## KIỂM THỬ PHẦN MỀM LÀ GÌ ?

- Chạy phần mềm chỉ là 1 phần không phải tất cả các hoạt động kiểm thử.
- Các hoạt động test tồn tại trước và sau khi chạy phần mềm bao gồm: lên kế hoạch và kiểm soát, chọn điều kiện test, thiết kế và chạy test case, kiểm tra kết quả, đánh giá tiêu chí kết thúc, báo cáo trong quy trình test và các hoạt động sau khi giai đoạn test hoàn thành.
- Test thì bao gồm cả review tài liệu, source code.



## MỤC TIÊU CỦA KIỂM THỬ

- Tìm lỗi
- Thu thập sự tự tin vào chất lượng
- Cung cấp thông tin để ra quyết định
- Ngăn ngừa lỗi



# MỤC ĐÍCH TRONG TỪNG GIAI ĐOẠN TEST

- **In development testing:** Test trong giai đoạn phát triển phần mềm mục đích chính là tìm được càng nhiều lỗi có thể và có thể sửa sớm
- **In acceptance testing:** Test trong giai đoạn nghiệm thu là để xác nhận xem hệ thống đã làm việc đúng như mong đợi chưa, thu thập sự tự tin rằng là phần mềm đã đúng với yêu cầu
- **In some cases:** Trong một vài trường hợp thì là để đánh giá chất lượng của phần mềm, để đưa ra thông tin cho nhà đầu tư về rủi ro của việc phát hành hệ thống tại thời điểm đó
- **Maintenance testing:** Test trong giai đoạn bảo trì phải bao gồm test để đảm bảo ko có lỗi mới xuất hiện trong quá trình thay đổi, chỉnh sửa
- **During operational testing:** Trong quá trình test vận hành thì mục đích chính là đánh giá các đặc tính chất lượng của hệ thống như độ tin cậy, tính sẵn sàng

## PHÂN BIỆT GIỮA KIỂM SOÁT CHẤT LƯỢNG VÀ ĐẢM BẢO CHẤT LƯỢNG

### Kiểm soát chất lượng

- Kiểm thử
- Kiểm duyệt
- Kiểm tra

### Đảm bảo chất lượng

- Kiểm toán lại chất lượng
- Kiểm tra và duyệt quy trình
- Đưa ra quy trình đảm bảo tiêu chuẩn



# PHÂN BIỆT TESTING VÀ DEBUGGING

## Testing

- Hoạt động tìm lỗi
- Được thực hiện bởi kiểm thử viên
- Tìm được càng nhiều lỗi càng tốt

## Debugging

- Hoạt động sửa lỗi
- Được thực hiện bởi lập trình viên
- Loại bỏ những lỗi tìm được

# PHÂN BIỆT VERIFICATION VÀ VALIDATION

- Verification (Xác minh)
  - Đánh giá các sản phẩm trung gian, kiểm tra xem nó có đáp ứng các yêu cầu cụ thể của từng giai đoạn
  - Kiểm tra xem sản phẩm có được xây dựng đúng theo yêu cầu và đặc điểm kỹ thuật thiết kế không.
  - Được thực hiện mà không cần chạy phần mềm.



# PHÂN BIỆT VERIFICATION VÀ VALIDATION...

- Các hoạt động Verification
  - Xác minh yêu cầu: tham gia review các yêu cầu.
  - Xác minh thiết kế: tham gia đánh giá của tất cả các tài liệu thiết kế.
  - Kiểm tra code: thực hiện review code
  - Xác minh tài liệu: kiểm tra hướng dẫn sử dụng và các tài liệu liên quan khác.

## VÍ DỤ VỀ VERIFICATION

- Một công ty phần mềm đang phát triển một ứng dụng đặt vé máy bay trực tuyến. Khách hàng yêu cầu hệ thống phải cho phép người dùng tìm kiếm chuyến bay, đặt vé, thanh toán trực tuyến và nhận vé qua email.
- Verification (Xác minh) trong quá trình phát triển phần mềm?

## VÍ DỤ VỀ VERIFICATION ...

Verification đảm bảo rằng phần mềm được xây dựng đúng theo yêu cầu đã đặt ra. Các hoạt động có thể bao gồm:

- Xác minh yêu cầu: Kiểm tra tài liệu đặc tả có mô tả đầy đủ các chức năng tìm kiếm, đặt vé, thanh toán, gửi email vé.
- Xác minh thiết kế :
  - Kiểm tra thiết kế hệ thống có hỗ trợ việc tìm kiếm và đặt vé hợp lý không.
  - Kiểm tra cơ sở dữ liệu có đủ bảng để lưu thông tin vé, hành khách, chuyến bay không.
- Kiểm tra code: Kiểm tra mã đảm bảo không có lỗi cú pháp hoặc cấu trúc sai.

## PHÂN BIỆT VERIFICATION VÀ VALIDATION...

- Validation (Thẩm định)
  - Đánh giá sản phẩm cuối cùng để kiểm tra xem nó có đáp ứng được yêu cầu nghiệp vụ hay không.
  - Xác định xem phần mềm có phù hợp với nhu cầu sử dụng và đáp ứng yêu cầu nghiệp vụ hay không.
  - Được thực hiện cùng với việc chạy phần mềm.



## PHÂN BIỆT VERIFICATION VÀ VALIDATION...

- Các hoạt động Validation
  - Chuẩn bị các tài liệu test requirement, test case và các thông số test khác để phân tích các kết quả test.
  - Đánh giá rằng các test requirement, các test case và các thông số kỹ thuật khác phản ánh yêu cầu và phù hợp để sử dụng.
  - Test các giá trị biên, ngưỡng hệ thống và các chức năng.
  - Test các thông báo lỗi và trong trường hợp có bất kỳ lỗi nào, ứng dụng sẽ kết thúc.
  - Kiểm tra xem phần mềm có đáp ứng các yêu cầu nghiệp vụ và phù hợp để sử dụng hay không.
  - Test các thông báo lỗi và trong trường hợp có bất kỳ lỗi nào, ứng dụng sẽ kết thúc.
  - Kiểm tra xem phần mềm có đáp ứng các yêu cầu nghiệp vụ và phù hợp để sử dụng hay không.



## VÍ DỤ VỀ VALIDATION

- Một công ty phần mềm đang phát triển một ứng dụng đặt vé máy bay trực tuyến. Khách hàng yêu cầu hệ thống phải cho phép người dùng tìm kiếm chuyến bay, đặt vé, thanh toán trực tuyến và nhận vé qua email.
- Validation (Thẩm định) sau khi hoàn thành phần mềm?

## VÍ DỤ VỀ VALIDATION...

Validation đảm bảo rằng phần mềm thực sự đáp ứng nhu cầu của người dùng và hoạt động đúng trong thực tế. Các hoạt động có thể bao gồm:

- Kiểm thử chức năng
  - Kiểm tra người dùng có thể tìm kiếm chuyến bay và đặt vé thành công không.
  - Kiểm tra quá trình thanh toán có hoạt động chính xác không (ví dụ: nhập thẻ tín dụng hợp lệ và xác nhận thanh toán thành công).
- Kiểm thử hệ thống
  - Mô phỏng số lượng lớn người dùng truy cập cùng lúc để kiểm tra hiệu suất hệ thống.
  - Kiểm tra toàn bộ quy trình từ tìm kiếm, đặt vé, thanh toán, đến nhận email xác nhận.
- Kiểm thử người dùng
  - Mời nhóm người dùng thực tế thử nghiệm ứng dụng để xem liệu họ có thể dễ dàng tìm kiếm chuyến bay và đặt vé không.

# PHÂN BIỆT ERROR, FAULT, FAILURE, BUG

- **Lỗi (error)**

- Là lỗi do con người gây ra (thường là các lập trình viên)
- **Error** phần mềm có thể là lỗi cú pháp hoặc lỗi logic

- **Sai sót (fault)**

- **Fault** của phần mềm không phải lúc nào cũng do **error** phần mềm
- Có thể có **fault** do dư thừa hoặc bỏ sót yêu cầu phần mềm (từ khâu khảo sát, phân tích, đưa ra yêu cầu phần mềm bị thừa hoặc bị sót so với yêu cầu của khách hàng)



# PHÂN BIỆT ERROR, FAULT, FAILURE, BUG ...

- **Hỏng hóc (failure)**

- Một **fault** của phần mềm dẫn đến **failure** khi **fault** đó bị phát hiện
- Một **fault** của phần mềm nếu không bị phát hiện hoặc không gây ảnh hưởng tới phần mềm thì sẽ không được coi là **failure**.

- **Bug**

- Bug là một **fault** được phát hiện trong quá trình kiểm thử. Khi tester kiểm tra và phát hiện rằng phần mềm không hoạt động đúng như mong đợi, họ sẽ ghi nhận một **bug**.

## 7 NGUYÊN LÝ CƠ BẢN TRONG KIỂM THỬ PHẦN MỀM

- Nguyên lý 1 : Kiểm thử cho thấy sự hiện diện của lỗi
- Nguyên lý 2: Kiểm thử vét cạn (toàn bộ) là không thể
- Nguyên lý 3: Kiểm thử sớm tiết kiệm thời gian và tiền bạc
- Nguyên lý 4: Phân cụm lỗi cùng nhau
- Nguyên lý 5: Đề phòng nghịch lý thuốc trừ sâu
- Nguyên lý 6: Kiểm thử phụ thuộc vào bối cảnh
- Nguyên lý 7: Ảo tưởng về sự vắng mặt của lỗi



## 7 NGUYÊN LÝ CƠ BẢN

- Nguyên lý 1: Kiểm thử cho thấy sự hiện diện của lỗi
  - Kiểm thử phần mềm chỉ có thể giúp phát hiện sự tồn tại của lỗi, nhưng không thể chứng minh rằng phần mềm không có lỗi."
  - Nói cách khác, dù một phần mềm có thể vượt qua tất cả các ca kiểm thử, không có nghĩa là nó hoàn toàn không có lỗi. Các lỗi có thể vẫn tồn tại nhưng chưa được phát hiện.

=>

- Kiểm thử giúp phát hiện lỗi, nhưng không thể chứng minh rằng phần mềm không còn lỗi nào.
- Ngay cả khi phần mềm vượt qua tất cả các ca kiểm thử, vẫn có thể có lỗi chưa được phát hiện. Do đó, kiểm thử nên kết hợp nhiều phương pháp khác nhau như kiểm thử hồi quy, kiểm thử bảo mật, kiểm thử tải,... để tăng khả năng phát hiện lỗi.



## 7 NGUYÊN LÝ CƠ BẢN

- Nguyên lý 2: Kiểm thử vét cạn (toàn bộ) là không thể
  - "Không thể kiểm thử hết tất cả các trường hợp đầu vào, trạng thái và đường đi trong phần mềm."
  - Điều này có nghĩa là kiểm thử toàn bộ (exhaustive testing) là không khả thi, bởi vì số lượng tình huống đầu vào của phần mềm có thể là vô hạn hoặc quá lớn để kiểm thử hết trong thời gian hợp lý.

=>

- Kiểm thử toàn bộ là không khả thi, do đó tester cần áp dụng các kỹ thuật chọn lọc như phân vùng tương đương, bảng quyết định, kiểm thử dựa trên rủi ro để tối ưu việc kiểm thử.
- Mục tiêu của kiểm thử không phải là kiểm thử hết tất cả mọi trường hợp, mà là tìm ra lỗi quan trọng một cách hiệu quả nhất.

## 7 NGUYÊN LÝ CƠ BẢN

- Nguyên lý 3: Kiểm thử sớm tiết kiệm thời gian và tiền bạc
    - “Việc phát hiện và sửa lỗi càng sớm trong vòng đời phát triển phần mềm thì chi phí sửa lỗi càng thấp.”
    - Nếu lỗi được phát hiện ngay từ giai đoạn đầu (ví dụ: khi thu thập yêu cầu hoặc thiết kế), chi phí sửa lỗi sẽ ít hơn nhiều so với khi lỗi xuất hiện ở giai đoạn phát triển, kiểm thử hoặc thậm chí sau khi phần mềm đã triển khai.
- => Tester nên tham gia từ sớm trong quá trình phát triển phần mềm để đảm bảo chất lượng cao nhất.

## 7 NGUYÊN LÝ CƠ BẢN

- Nguyên lý 4: Phân cụm lỗi cùng nhau

- Một số lượng nhỏ các mô-đun thường chứa hầu hết các lỗi phát hiện được trong giai đoạn kiểm thử trước khi phát hành, hoặc chịu trách nhiệm về hầu hết các hỏng hóc trong quá trình hoạt động. Các cụm lỗi dự đoán được và bị phát hiện khi kiểm thử và vận hành là đầu vào quan trọng cho phân tích rủi ro được áp dụng trong lần kiểm thử.

=> Nếu có nguồn lực hạn chế, tester nên tập trung kiểm thử kỹ hơn ở các module có lịch sử nhiều lỗi hoặc mới được cập nhật.



## 7 NGUYÊN LÝ CƠ BẢN

- Nguyên lý 5: Đề phòng nghịch lý thuốc trừ sâu
  - Nếu cùng một tập các trường hợp kiểm thử, được lặp đi lặp lại trong một thời gian dài sẽ không còn hiệu quả để phát hiện lỗi nữa.
  - Giống như thuốc trừ sâu, nếu sử dụng một loại thuốc trừ sâu nhiều lần trên cùng một loại sâu bệnh, sâu bệnh sẽ dần thích nghi và thuốc sẽ mất tác dụng. Trong kiểm thử phần mềm, nếu tester chỉ sử dụng một bộ test cố định, hệ thống có thể dần trở nên ổn định với các lỗi mà bộ test đó có thể phát hiện, nhưng lỗi mới có thể xuất hiện mà không bị phát hiện.

=> Nghịch lý thuốc trừ sâu nhắc nhở tester rằng **chạy lại cùng một bộ test nhiều lần sẽ không giúp phát hiện lỗi mới**. Để kiểm thử hiệu quả, cần **thay đổi, mở rộng và cập nhật liên tục các test case** nhằm phát hiện nhiều lỗi tiềm ẩn hơn.

## 7 NGUYÊN LÝ CƠ BẢN

- Nguyên lý 6: Kiểm thử phụ thuộc vào bối cảnh

- Nghĩa là chiến lược, phương pháp, kỹ thuật kiểm thử có thể thay đổi tùy theo từng dự án, lĩnh vực, yêu cầu cụ thể của phần mềm. Không có một phương pháp kiểm thử nào phù hợp cho tất cả phần mềm.

=> Kiểm thử phần mềm không có một quy trình cố định áp dụng cho tất cả dự án. Tester phải linh hoạt và hiểu rõ bối cảnh của phần mềm để lựa chọn phương pháp kiểm thử phù hợp.

## 7 NGUYÊN LÝ CƠ BẢN

- Nguyên lý 7: Ảo tưởng về sự vắng mặt của lỗi
  - “Việc một phần mềm không có lỗi không có nghĩa là nó đáp ứng đúng nhu cầu của người dùng.”
  - Điều này có nghĩa là ngay cả khi phần mềm đã được kiểm thử kỹ lưỡng và không còn lỗi kỹ thuật nào, nó vẫn có thể không đáp ứng được mục đích sử dụng, yêu cầu của khách hàng hoặc thị trường.



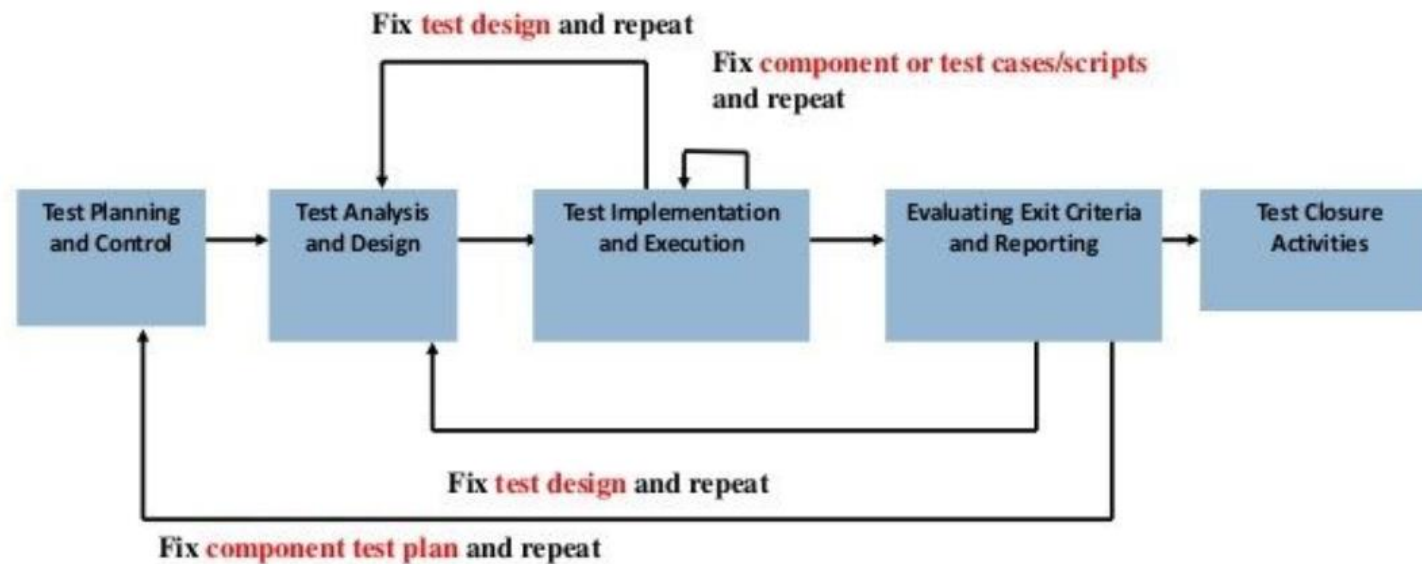
=>

- Hiểu rõ yêu cầu của khách hàng/người dùng: Không chỉ tập trung vào lỗi kỹ thuật mà còn phải kiểm thử xem phần mềm có đáp ứng đúng yêu cầu hay không.
- Kiểm thử khả năng sử dụng (Usability Testing): Đảm bảo phần mềm dễ dùng, thân thiện với người dùng.
- Lắng nghe phản hồi từ người dùng thực tế: Thu thập ý kiến từ khách hàng để cải tiến sản phẩm.
- Không chỉ tập trung vào kiểm thử chức năng: Cần kiểm thử toàn diện, bao gồm hiệu suất, bảo mật, khả năng mở rộng và trải nghiệm người dùng.



## CÁC GIAI ĐOẠN TRONG QUY TRÌNH KIỂM THỬ PHẦN MỀM

### Fundamental Test Process



## TEST PLANNING

- Lập kế hoạch kiểm thử: xác định mục tiêu của việc test và chỉ rõ các hoạt động test để đạt được mục tiêu trọng tâm.
- Nhiệm vụ chính
  - Xác định phạm vi, rủi ro và mục tiêu của kiểm thử
  - Xác định phương pháp tiếp cận test ( kỹ thuật, tỷ lệ bao phủ, xác định và giao tiếp giữa các đội tham gia vào test)
  - Triển khai kế hoạch chiến lược test
  - Xác định các nguồn lực cần thiết ( con người, môi trường, PC, công cụ)
  - Lập lịch trình cho các giai đoạn
  - Xác định các tiêu chí kết thúc

## TEST CONTROL

- Kiểm soát và giám sát kiểm thử: diễn ra liên tục để so sánh giữa tiến độ thực tế và kế hoạch, báo cáo trạng thái (gồm cả những cái lệch so với kế hoạch).
- Kiểm soát và giám sát kiểm thử phải đưa ra được các hành động cần thiết để đạt được mục tiêu của dự án.
- Các hoạt động kiểm soát và giám sát kiểm thử được thực hiện và kiểm tra suốt cả dự án.
- Kế hoạch kiểm thử phải được cập nhật phù hợp với những hoạt động kiểm soát và giám sát.



## TEST CONTROL ...

- Nhiệm vụ chính
  - Đo đạc và phân tích kết quả của hoạt động review và kiểm thử
  - Theo dõi và ghi chép tiến độ, độ bao phủ test, tiêu chí kết thúc
  - Cung cấp thông tin của test cho việc lựa chọn, đánh giá
  - Đưa ra các hành động khắc phục, sửa chữa
  - Ra quyết định

## TEST ANALYSIS

- Phân tích kiểm thử: Xác nhận các test case thông qua việc phân tích cơ sở kiểm thử xác định các thuộc tính có thể kiểm thử và điều kiện kiểm thử liên quan.
- Nhiệm vụ chính
  - Xem xét các cơ sở kiểm thử (yêu cầu, các rủi ro, báo cáo phân tích rủi ro, kiến trúc, thiết kế, đặc tả giao diện).
  - Đánh giá khả năng kiểm thử của dựa trên cơ sở kiểm thử và các đối tượng kiểm thử.
  - Xác định và sắp xếp thứ tự ưu tiên của các điều kiện kiểm thử dựa vào phân tích các danh sách kiểm thử, đặc tả, hiệu ứng và cấu trúc của phần mềm.



## TEST DESIGN

- Thiết kế kiểm thử: Xác định dữ liệu đầu vào và đầu ra cụ thể để chạy phần mềm dựa trên các điều kiện kiểm thử và dữ liệu cụ thể đã được phân tích.
- Nhiệm vụ chính
  - Thiết kế và sắp xếp thứ tự ưu tiên của Test case.
  - Xác định các dữ liệu test để phục vụ cho test conditions and test cases.
  - Xác định môi trường test cần thiết lập và các yêu cầu về hạ tầng, công cụ.
  - Tạo truy xuất hai chiều giữa cơ sở kiểm thử, điều kiện kiểm thử, trường hợp kiểm thử và thủ tục kiểm thử.

# TEST DESIGN

Test Case Type	Description	Test Step	Expected Result	Status
Functionality	Area should accommodate up to 20 characters	Input up to 20 characters	All 20 characters in the request should be appropriate	Pass or Fail
Security	Verify password rules are working	Create a new password in accordance with rules	The user's password will be accepted if it adheres to the rules	Pass or Fail
Usability	Ensure all links are working properly	Have users click on various links on the page	Links will take users to another web page according to the on-page URL	Pass or Fail



## TEST IMPLEMENTATION

- Triển khai kiểm thử: Chuẩn bị phần mềm kiểm thử cần thiết để thực thi kiểm thử, dựa trên phân tích và thiết kế kiểm thử.
- Nhiệm vụ chính
  - Chuẩn hoá, xây dựng và sắp xếp ưu tiên test case (gồm cả xác định test data).
  - Xây dựng và ưu tiên các thủ tục test, tạo test data, chuẩn bị thủ tục và viết test scripts tự động test.
  - Tạo bộ test suites từ thủ tục test cho việc chạy test.
  - Kiểm tra các môi trường được thiết lập đúng chưa.
  - Kiểm tra và cập nhật ma trận theo dõi.

## TEST EXECUTION

- Thực thi kiểm thử: chạy kiểm thử trên một hệ thống hoặc thành phần đang được kiểm thử, tạo ra kết quả thực tế.
- Nhiệm vụ chính
  - Chạy các thủ tục test bằng tay hoặc tự động bằng tool.
  - Ghi nhận kết quả chạy test và các vấn đề phát hiện được, phiên bản được test, công cụ test và tài sản test.
  - So sánh kết quả thực tế và kết quả mong đợi.
  - Báo cáo lỗi và phân tích chúng để tìm ra nguyên nhân (lỗi trong code, trong test data, trong tài liệu test hoặc sai ở cách thực hiện test).
  - Lặp lại các hoạt động test để xác nhận các sửa lỗi là phù hợp (re-test), kiểm tra những phần xung quanh không xảy ra lỗi nữa (regression test).

## TEST EXIT CRITERIA AND REPORTING

- Tiêu chí kết thúc và báo cáo kiểm thử: được thực hiện khi việc chạy test đã đạt được đến mục tiêu trong kế hoạch, nó nên được làm ở mỗi level của test.
- Nhiệm vụ chính
  - Kiểm tra bao phủ 100% requirement
  - Kiểm tra tất cả các lỗi ưu tiên được close
  - Kiểm tra các mức độ bao phủ tối đa cần đạt được
  - Kiểm tra thời gian và ngân sách đã tiêu tốn
  - Kiểm tra khả năng xuất hiện bug tối thiểu
  - Đánh giá xem có cần phải test thêm nữa không hoặc phải sửa tiêu chí kết thúc hay không.
  - Viết báo cáo test gửi nhà đầu tư.



# TEST CLOSURE

- Là hoạt động thu thập dữ liệu từ các hoạt động test đã hoàn thành để tập trung lại các kinh nghiệm, tài sản test.
- Là hoạt động diễn ra ở mỗi giai đoạn của dự án, khi hệ thống được phát hành hoặc khi dự án test hoàn thành, giai đoạn kết thúc, phát hành sản phẩm.



## TEST CLOSURE

- Nhiệm vụ chính kết thúc kiểm tra
  - Kiểm tra các sản phẩm được lên kế hoạch phát hành đã phát hành chưa
  - Đóng các báo cáo lỗi hoặc nâng cao các ghi chú thay đổi cho phần còn sót lại.
  - Tài liệu hoá.
  - Hoàn thiện và thu hồi tài sản test, môi trường và hạ tầng việc tái sử dụng.
  - Bàn giao tài sản test cho tổ chức.
  - Phân tích các bài học kinh nghiệm để xác định các thay đổi cần thiết cho các ban hành trong tương lai hoặc dự án khác.
  - Sử dụng các thông tin thu thập để cải tiến hoạt động test.



## TÂM LÝ HỌC TRONG KIỂM THỬ

- Các mức độ độc lập trong kiểm thử
- Giao tiếp với những phần còn lại của dự án

## CÁC MỨC ĐỘ ĐỘC LẬP TRONG KIỂM THỬ

- Các trường hợp Test không được xây dựng bởi người viết ra phần mềm
- Test phải được xây dựng bởi người khác
- Test cần được xây dựng bởi người từ những phòng ban khác
- Test cần được xây dựng bởi người từ những tổ chức khác

## GIAO TIẾP VỚI PHẦN CÒN LẠI DỰ ÁN

- Việc chỉ ra lỗi được nhiều đối tượng coi như là 1 hành động chỉ trích đến sản phẩm và tác giả. Vì thế mà test thường bị coi là hoạt động phá hoại, ngay cả khi đó là 1 hoạt động mang tính xây dựng trong quản lý rủi ro.
- Để tìm ra lỗi trong phần mềm yêu cầu cần phải có sự tò mò, bí quan, con mắt nghiêm trọng, tập trung vào chi tiết, giao tiếp tốt với các nhóm phát triển và có kinh nghiệm để dự đoán được lỗi.



## GIAO TIẾP VỚI PHẦN CÒN LẠI DỰ ÁN...

- Tester và test leader cần có kỹ năng giao tiếp tốt để trao đổi về các thông tin thực tế của lỗi, tiến độ, rủi ro trên tinh thần xây dựng.
- Giao tiếp những điểm tìm được về sản phẩm 1 cách khách quan, tập trung vào thực tế mà không có sự thù ghét cá nhân.
- Giải thích bằng các hiểu biết về những gì chúng ta phải làm gì với nó hoặc sửa nó tốt hơn.
- Khởi đầu bằng sự hợp tác không phải là trận chiến. Luôn nhắc mọi người về mục tiêu chung là giúp hệ thống đạt chất lượng tốt hơn.

