



# KIỂM THỬ PHẦN MỀM

(Software Testing)



**GV: ThS. Nguyễn Thị Thanh Trúc**  
**Khoa: Công nghệ Phần mềm**  
**Email: [trucntt@uit.edu.vn](mailto:trucntt@uit.edu.vn)**

# BÀI 4: Các loại hình kiểm thử



- 4.1. Kiểm thử chức năng
- 4.2. Kiểm thử phi chức năng
- 4.3. Kiểm thử liên quan đến sự thay đổi

## 4.1. Kiểm thử chức năng



- Qui trình cố gắng tìm ra các khác biệt giữa đặc tả bên ngoài của phần mềm và thực tế mà phần mềm cung cấp.
- Đặc tả bên ngoài của phần mềm là đặc tả chính xác về hành vi của phần mềm theo góc nhìn của người dùng thấy.
- Các loại kiểm thử chức năng:
  - Kiểm thử chức năng của hệ thống
  - Kiểm thử tích hợp dữ liệu và cơ sở dữ liệu
  - Kiểm thử vòng lặp công việc
  - Kiểm thử kiểm soát truy cập
  - Kiểm thử giao diện

## 4.1.1. Kiểm thử chức năng hệ thống



- Mục tiêu của loại kiểm thử này là đảm bảo đúng mục tiêu của kiểm thử chức năng: nhập dữ liệu- xử lý lấy và kiểm tra kết quả trả về
- Kiểm tra sản phẩm phần mềm và các hoạt động của các chức năng bên trong sản phẩm đó bằng cách tương tác thông qua giao diện người dùng của sản phẩm
- Phân tích kết quả trả về

## 4.1.2. Kiểm thử giao diện



- **Mục tiêu:** kiểm tra giao diện của các chức năng trong một sản phẩm hệ thống hoạt động so với thiết kế
- Kiểm thử giao diện cần kiểm thử:
  - Liên kết hay chuyển tiếp
  - Cách thức truy cập (sử dụng phím tab, chuột...)
  - Kiểm tra các đối tượng trên màn hình:
    - Màu sắc
    - Vị trí
    - Kích thước chữ
    - Kiểu đối tượng
    - ....

### 4.1.3. Kiểm thử tích hợp dữ liệu và CSDL



- Kiểm tra các chức năng của một sản phẩm hay hệ thống phần mềm hoạt động đúng không sau khi sản phẩm pm đó đã có sự tích hợp hay đưa dữ liệu cũ, dữ liệu đã có sẵn từ bên ngoài vào sản phẩm.
- Đảm bảo các chức năng của hệ thống mới sử dụng được các dữ liệu cũ

## 4.1.4. Kiểm thử vòng lặp công việc



- Đảm bảo hoạt động của các công việc được chạy tự động theo lịch đã đặt trước không do người dùng tác động

## 4.1.5. Kiểm thử kiểm soát truy cập



- Đảm bảo các tác nhân, người sử dụng chỉ có thể truy cập vào đúng chức năng họ được phép truy cập
- Đảm bảo chỉ những người dùng được phân quyền truy cập hệ thống mới có thể truy cập vào hệ thống và thông qua các gateway thích hợp



	Navigate Pages	Draw/Markup	Import Content	Sync Rights	Delete Pages & other users content	Permission Controls	View/Book Settings
Owner	✓	✓	✓	✓	✓	✓	✓
Moderator	✓	✓	✓	✓	✓		
Collaborator	✓	✓	✓				
Reviewer	✓	✓					
Viewer	✓*						

\* A viewer can only navigate pages if Sync Mode is OFF.



## 4.2. Kiểm thử phi chức năng



- Tập trung vào kiểm thử sản phẩm, hệ thống phần mềm cần kiểm thử có những đặc tính tốt như thế nào (how well)
- Kiểm thử phi chức năng có thể được sử dụng ở mọi cấp độ kiểm thử nhưng thường được sử dụng hiệu quả nhất trong cấp độ kiểm thử hệ thống và kiểm thử chấp nhận sản phẩm.
- Các loại kiểm thử phi chức năng (4 loại thường dùng)
  - Kiểm thử hiệu năng (performance testing)
  - Kiểm thử tải trọng (load testing)
  - Kiểm thử tập trung (stress testing)
  - Kiểm thử với lượng dữ liệu lớn (volume testing)

## 4.2.1. Kiểm thử hiệu năng



- Bảo đảm tối ưu việc phân bổ tài nguyên hệ thống (ví dụ bộ nhớ) nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn...
- Thí dụ:
  - trình chiếu phim full HD không chiếu kịp 20frame/sec.
  - trình nén dữ liệu không nén dữ liệu kịp với tốc độ đưa ra.
  - trình soạn thảo văn bản không nhận và xử lý kịp các ký tự được nhập bởi người dùng.
  - trình ghi DVD không tạo dữ liệu ghi kịp tốc độ mà ổ DVD yêu cầu...

## 4.2.2. Kiểm thử tải trọng



- Kiểm thử tải trọng (kiểm thử đồng thời): tập trung vào xác định đặc tính hiệu suất của hệ thống hay sản phẩm phần mềm trong điều kiện tải hay upload cụ thể
- Bảo đảm hệ thống vận hành đúng dưới áp lực cao (ví dụ nhiều người truy xuất cùng lúc).
- Ví dụ: Hệ thống hỗ trợ cho việc truy cập, giải quyết 3000 yêu cầu trong một ngày, đáp ứng việc sử dụng của 350 người sử dụng đồng thời từ 9h30 đến 11h am

## 4.2.3. Kiểm thử tập trung



- Stress Test tập trung vào các trạng thái tới hạn, các "điểm chết", các tình huống bất thường như đang giao dịch thì ngắt kết nối (xuất hiện nhiều trong kiểm tra thiết bị như POS, ATM...)...

## 4.2.3. Kiểm thử với lượng dữ liệu lớn



- Kiểm thử tập trung vào việc xác định hoặc xác nhận đặc tính hiệu suất của hệ thống hoặc ứng dụng được kiểm thử trong điều kiện hệ thống có lượng dữ liệu rất lớn
- Dữ liệu lớn có thể là cơ sở dữ liệu lớn hoặc dữ liệu trong file upload lên hệ thống có dung lượng lớn

## 4.4. Kiểm thử liên quan đến sự thay đổi

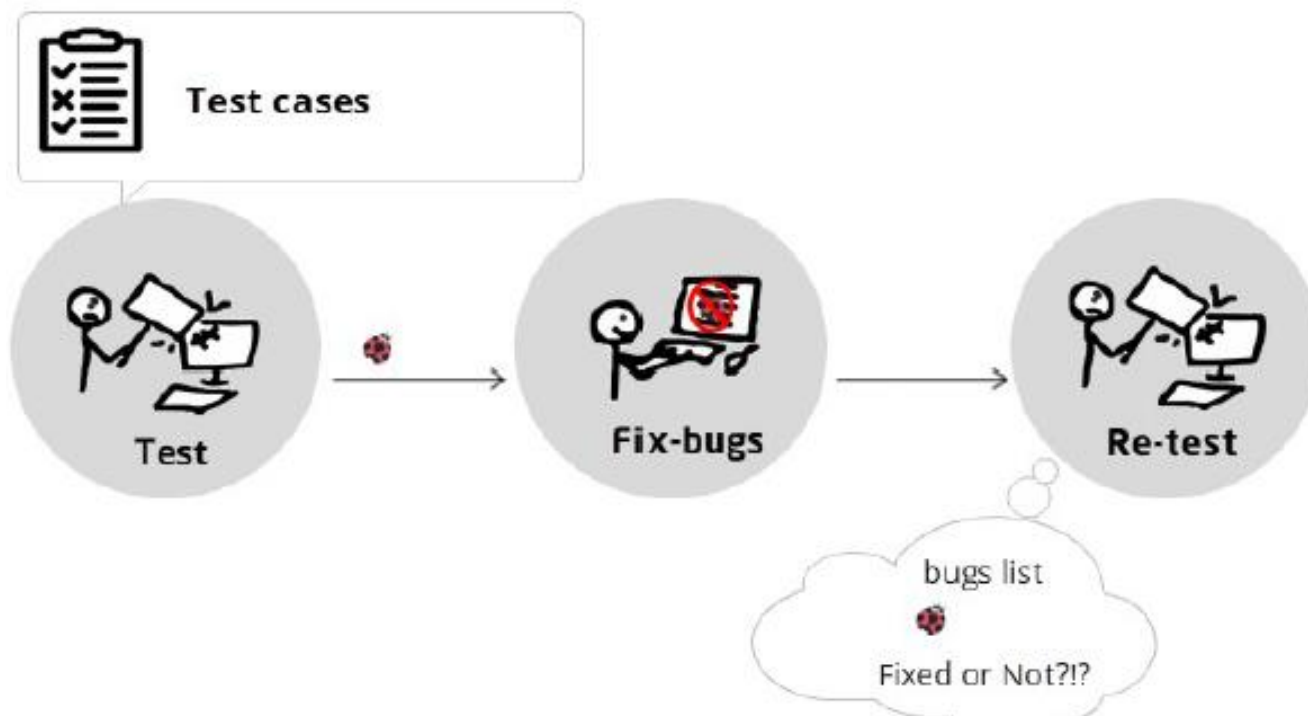


- Thực hiện hoạt động kiểm thử khi có sự thay đổi trên hoặc trong sản phẩm phần mềm.
- Sự thay đổi của sản phẩm phần mềm có thể là:
  - Sửa chữa các lỗi tìm được
  - Sản phẩm được nâng cấp, được thay đổi về chức năng
- Các loại kiểm thử liên quan đến sự thay đổi:
  - Kiểm thử lại
  - Kiểm thử hồi quy

## 4.4.1. Kiểm thử lại (kiểm thử xác nhận)



- Chỉ kiểm thử những test case chưa pass



## 4.4.1. Kiểm thử lại (kiểm thử xác nhận)



- Khi thực hiện kiểm thử xác nhận cần chú ý:
- Thực hiện kiểm thử đúng các bước như trong mô tả trường hợp kiểm thử:
  - Đúng các tập đầu vào
  - Đúng các dữ liệu
  - Đúng môi trường kiểm thử



## 4.4.2. Kiểm thử hồi quy



- Kiểm thử hồi quy để đảm bảo rằng những thay đổi mới không làm ảnh hưởng đến những phần đã hoàn thiện trước đó
- Kiểm thử hồi quy thường được thực hiện tự động

Have requirement changes



- After the Software has changed
- When the Environment has changed



Regression Testing



Validation of change

## 4.4.2. Kiểm thử hồi quy



- Các phép thử hồi quy được chia làm 3 loại:
  - **Các phép thử đại diện:** thực hiện tất cả chức năng của pm
  - **Các phép thử bổ sung:** tập trung vào chức năng dễ bị ảnh hưởng nhất khi có thay đổi
  - **Các phép thử tập trung:** tập trung vào thành phần pm bị thay đổi



# Kiểm thử hồi quy



- Việc kết hợp các module lại với nhau có thể ảnh hưởng đến vòng lặp điều khiển, cấu trúc dữ liệu hay I/O chia sẻ trong một số module
- Điều đó làm lộ ra một số lỗi không thể phát hiện được khi tiến hành kiểm thử theo đơn vị

## → Phải kiểm thử hồi quy khi tích hợp

- Kiểm thử hồi quy có thể được tiến hành thủ công bằng cách thực hiện lại các test-case đã tạo ra. Hoặc có thể dùng một công cụ capture-playback để thực hiện tự động