**BẢN BÁO CÁO BÀI TẬP TUẦN 9 – TÀI LIỆU ISTQB (SLIDE 06)**

**CÔNG CỤ HỖ TRỢ KIỂM THỬ (CAST)**

* **Bộ môn:** Kiểm thử và đảm bảo chất lượng phần mềm.
* **Giảng viên:** PGS.TS Trương Anh Hoàng
* **Nhóm 42:**
* Ngô Quốc Thắng
* Phan Bá Mỹ
* Nguyễn Hoàng Quân
* Phạm Văn Bộ

Năm học 2015 – 2016

Nội dung

[1. Phân loại công cụ 4](#_Toc436577996)

[1.1. Công cụ trong phát triển phần mềm 4](#_Toc436577997)

[1.2. Phân loại các công cụ 4](#_Toc436577998)

[1.2.1. Công cụ kiểm thử yêu cầu 5](#_Toc436577999)

[1.2.2. Công cụ phân tích tĩnh 5](#_Toc436578000)

[1.2.3. Công cụ kiểm thử thiết kế 5](#_Toc436578001)

[1.2.4. Công cụ kiểm thử dữ liệu 5](#_Toc436578002)

[1.2.5. Công cụ kiểm thử vận hành 5](#_Toc436578003)

[1.2.6. Công cụ so sánh 6](#_Toc436578004)

[1.2.7. Kiểm thử harnesses và driver 6](#_Toc436578005)

[1.2.8. Công cụ kiểm thử hiệu năng 6](#_Toc436578006)

[1.2.9. Công cụ phân tích động 7](#_Toc436578007)

[1.2.10. Công cụ gỡ lỗi 7](#_Toc436578008)

[1.2.11. Công cụ quản lí 7](#_Toc436578009)

[1.2.12. Công cụ phân tích độ bao phủ 7](#_Toc436578010)

[2. Tại sao ghi lại/xem xét lại không phải là tự động hóa trong kiểm thử 7](#_Toc436578011)

[2.1. Lợi ích của việc ghi lại những ca “kiểm thử bằng tay” 7](#_Toc436578012)

[2.2. Lưu kịch bản test 8](#_Toc436578013)

[2.3. Kiểm thử mạnh và kiểm thử nhạy 8](#_Toc436578014)

[2.4. Quá chi tiết là dư thừa 9](#_Toc436578015)

[2.5. Kiểm chứng tự động hóa 9](#_Toc436578016)

[3. Tự động hóa và kiểm thử là hai kĩ năng đặc biệt 9](#_Toc436578017)

[3.1. Nỗ lực tự động hóa 9](#_Toc436578018)

[3.1.1. Kiểm thử bằng tay không có kịch bản trước 10](#_Toc436578019)

[3.1.2. Kiểm thử bằng tay với kịch bản không rõ ràng 11](#_Toc436578020)

[3.1.3. Kiểm thử bằng tay với kịch bản chi tiết. 12](#_Toc436578021)

[4. Best practice 13](#_Toc436578022)

[4.1. Không tự động hóa quá dài hạn 13](#_Toc436578023)

[4.2. Duy trì kiểm soát 13](#_Toc436578024)

[4.3. Đầu tư 13](#_Toc436578025)

[4.4. Nên tự động hóa khi nào? 14](#_Toc436578026)

[4.5. Trường hợp không nên tự động hóa? 14](#_Toc436578027)

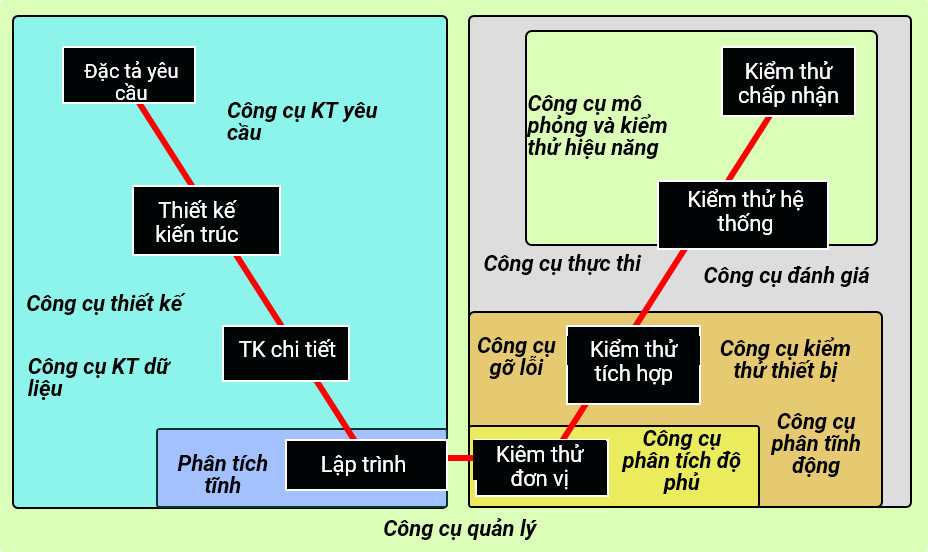
[5. Tổng kết 14](#_Toc436578028)

# 1. Phân loại công cụ

## 1.1. Công cụ trong phát triển phần mềm

Các công cụ được sử dụng trong quá trình phát triển phần mềm được thể hiện ở

Hình 1.



Hình 1.

## 1.2. Phân loại các công cụ

* Công cụ kiểm thử yêu cầu
* Công cụ phân tích tĩnh
* Công cụ kiểm thử thiết kế
* Công cụ kiểm thử dữ liệu
* Công cụ kiểm thử vận hành
* Công cụ so sánh
* Kiểm thử harnesses và driver
* Công cụ kiểm thử hiệu năng
* Công cụ phân tích động
* Công cụ gỡ lỗi (debugging)
* Công cụ quản lí
* Công cụ phân tích độ bao phủ

### 1.2.1. Công cụ kiểm thử yêu cầu

Công cụ kiểm thử này hỗ trợ tự động cho việc xác minh và phê chuẩn những yêu cầu theo mẫu có sẵn, kiểm tra tính nhất quán.

### 1.2.2. Công cụ phân tích tĩnh

Công cụ kiểm thử mang một số đặc điểm sau:

* Cung cấp 1 phần thông tin và chất lượng phần mềm
* Mã nguồn sẽ được phân tích để tìm các lỗi hay gặp mà không cần phải thực thi
* Mục tiêu đo lường:
* Độ phức tạp lốc xoáy
* Vấn đề khác: mức độ lồng nhau, dung lượng

### 1.2.3. Công cụ kiểm thử thiết kế

Công cụ này được sử dụng nhằm sinh ra các ca kiểm thử đầu vào, có thể từ đặc tả chính thức hoặc từ mã nguồn.

### 1.2.4. Công cụ kiểm thử dữ liệu

Công cụ kiểm thử phục vụ thao tác với dữ liệu có những đặc điểm:

* Được chọn từ những tệp hoặc cơ sở dữ liệu đã có
* Được tạo ra theo một số chuẩn
* Được chỉnh sửa từ những nguồn khác nhau

### 1.2.5. Công cụ kiểm thử vận hành

* Thực hiện/thực thi các ca kiểm thử
* Giám sát việc thực hiện từng ca kiểm thử
* Kịch bản kiểm tra được viết bằng ngôn ngữ của chương trình
* Bộ giá trị đầu vào, giá trị đầu ra mong muốn, và giá trị đầu ra thực tế nhằm tạo báo cáo kiểm thử
* Thường thấy trong tự động hóa kiểm thử hồi quy
* Dựa vào thuộc tính
  + Mô phỏng tương tác người dùng với các thiết bị đầu cuối
  + Bắt sự kiện cho các phím tổ hợp và màn hình hồi đáp.
* GUI (Giao diện người dùng)
  + Mô phỏng tương tác người dùng cho những ứng dụng WIMP(Windows, Icons, Mouse, Pointer)
  + Bắt sự kiện cho chuyển động chuột, nhấp chuột, hay dữ liệu được nhập từ bàn phím
  + Bắt các màn hình, hình ảnh, kí tự, trạng thái của đối tượng

### 1.2.6. Công cụ so sánh

* Phát hiện sự sai khác giữa kết quả kiểm thử thực tế và kết quả mong muốn
* screens, characters, bitmaps
* masking and filtering
* Các công cụ kiểm thử thực thi thường tích hợp khả năng đánh giá
* Công cụ đánh giá đứng tác biệt là để phục vụ cho các tệp tin hoặc các dữ liệu

### 1.2.7. Kiểm thử harnesses và driver

* Thường dùng cho những phần mềm không có giao diện người dùng
* Thường chạy một nhóm các ca kiểm thử tự động hoặc đánh giá
* Đôi khi được sinh ra theo số đo
* Sử dụng các bộ phỏng (bởi vì khi thực nghiệm trong môi trường thực tế sẽ tốn kém và nguy hiểm)

### 1.2.8. Công cụ kiểm thử hiệu năng

* Load generation
* Chạy ứng dụng qua giao diện người dùng hoặc test harness
* Khả năng chịu tải
* Đo các giao dịch
* Thời gian hồi đáp cho yêu cầu người dùng thông qua giao diện
* Báo cáo dựa trên các bản ghi, đồ thị so sánh thời gian tải và thời gian hồi đáp

### 1.2.9. Công cụ phân tích động

* Hỗ trợ cả kiểm thử đơn vị và kiểm thử tích hợp
* Cho phép sinh ra các ca kiểm thử từ mã nguồn và thực thi chúng nhằm phát hiện các lỗi lập trình

### 1.2.10. Công cụ gỡ lỗi

* Cho phép định vị các lỗi được phát hiện bởi một ca kiểm thử.
* Cho phép kiểm tra chạy chương trình ở mức chi tiết
* Chạy từng câu lệnh
* Đánh dấu breakpoints hoặc watchpoint ở bất kỳ câu lệnh nào
* Kiểm tra giá trị của biến hoặc dữ liệu khác.

### 1.2.11. Công cụ quản lí

* Quản lý của phần kiểm thử: kế hoạch, chi tiết và kết quả kiểm thử;
* Quản lý dự án trong quy trình kiểm thử, ví dụ như ước lượng, kế hoạch test và kết quả ghi lại;
* Công cụ quản lý thay đổi (bao gồm thứ tự công việc để phân bổ, sửa chữa và kiểm tra lại);
* Truy hồi (kiểm tra lại đến yêu cầu và thiết kế).

### 1.2.12. Công cụ phân tích độ bao phủ

* Mục đích: xác định những phần nào trong cấu trúc phần mềm đã được kiểm thử trong cấu trúc phần mềm;
* Trong kiểm thử tĩnh, mã nguồn sẽ được đánh giá độ bao phủ của 1 bộ test.
* Công cụ ghi lại những phần được bao phủ, cũng như không được bao phủ bởi bộ test (theo dòng) và báo cáo lại;
* Các loại độ bao phủ khác nhau: bao phủ dòng lệnh, bao phủ nhánh, bao phủ điều kiện.

# 2. Tại sao ghi lại/xem xét lại không phải là tự động hóa trong kiểm thử

## 2.1. Lợi ích của việc ghi lại những ca “kiểm thử bằng tay”

Việc văn bản hóa những gì người kiểm thử đã thực hiện đem lại nhiều lợi ích:

* Hữu dụng cho việc bắt những bộ test tùy biến (ví dụ: người sử dụng cuối)
* Có thể phát hiện lỗi phần mềm để làm lại.

Một kịch bản chi tiết sẽ:

* Ghi lại các đầu vào thực tế
* Có thể được sử dụng bởi người hiểu biết kỹ thuật để thực thi bộ test tự động đảm bảo hơn.

Lý tưởng cho các task thực hiện 1 lần

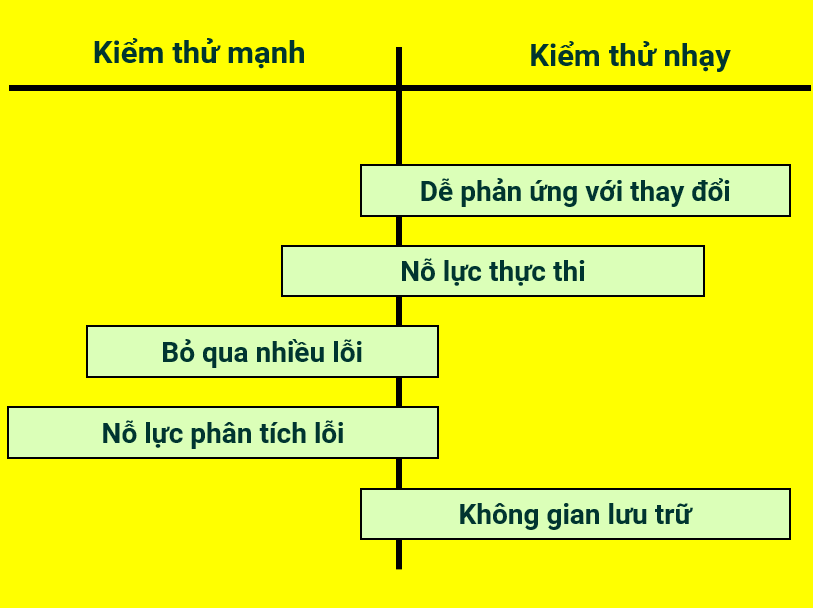
* Chẳng hạn như nhập một dữ liệu dài phức tạp

## 2.2. Lưu kịch bản test

Kịch bản test được lưu lại sẽ khá khó hiểu vì suy cho cùng nó được viết bằng một ngôn ngữ lập trình, hơn nữa kịch bản cũng sẽ không đáp ứng được nhiều thay đổi trong phần mềm bởi một sự thay đổi đơn giản trong giao diện cũng có thể tác động đến rất nhiều kịch bản.

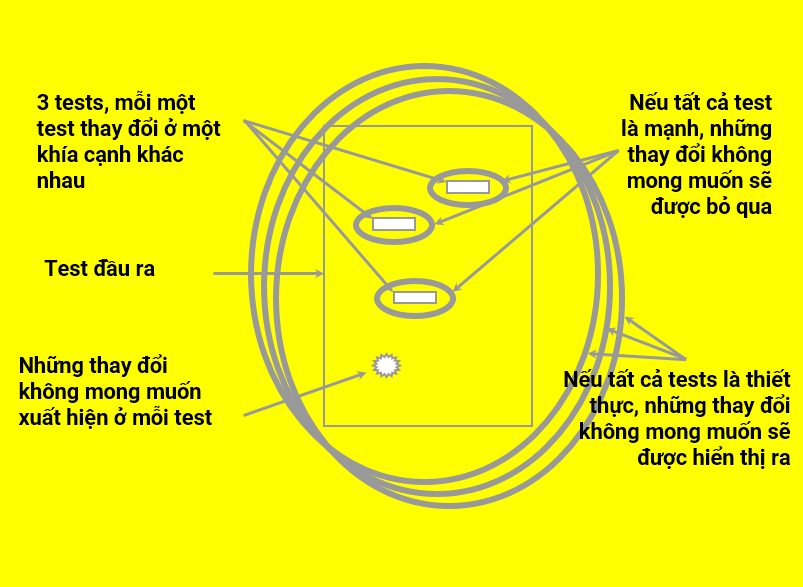
## 2.3. Kiểm thử mạnh và kiểm thử nhạy

So sánh kiểm thử mạnh và kiểm thử nhạy.



Hình 2.

## 2.4. Kiểm thử quá chi tiết là dư thừa



Hình 3.

## 2.5. Kiểm chứng tự động hóa

Kiểm chứng tự động hóa có thể được thực hiện bởi nhiều cách như: so sánh nhiều/so sánh ít, khả năng thay đổi/hiệu quả phát hiện lỗi. Kịch bản sẽ ngày càng trở nên phức tạp và sẽ có nhiều tác vụ kéo theo.

Thông thường nhiều sự kiểm chứng hơn có thể được hoàn thành, sự tự động hóa sẽ làm cho việc kiểm thử đạt kết quả tốt hơn.

# 3. Tự động hóa và kiểm thử là hai kĩ năng đặc biệt

## 3.1. Nỗ lực tự động hóa

Công sức bỏ ra cho một mỗi ca kiểm thử tự động thường dao động trong một khoảng rất lớn, thường là từ 2 đến 10 lần so với kiểm thử bằng tay.

Việc này phụ thuộc vào công cụ kiểm thử, môi trường, kĩ năng người sử dụng và phần mềm cần kiểm thử.

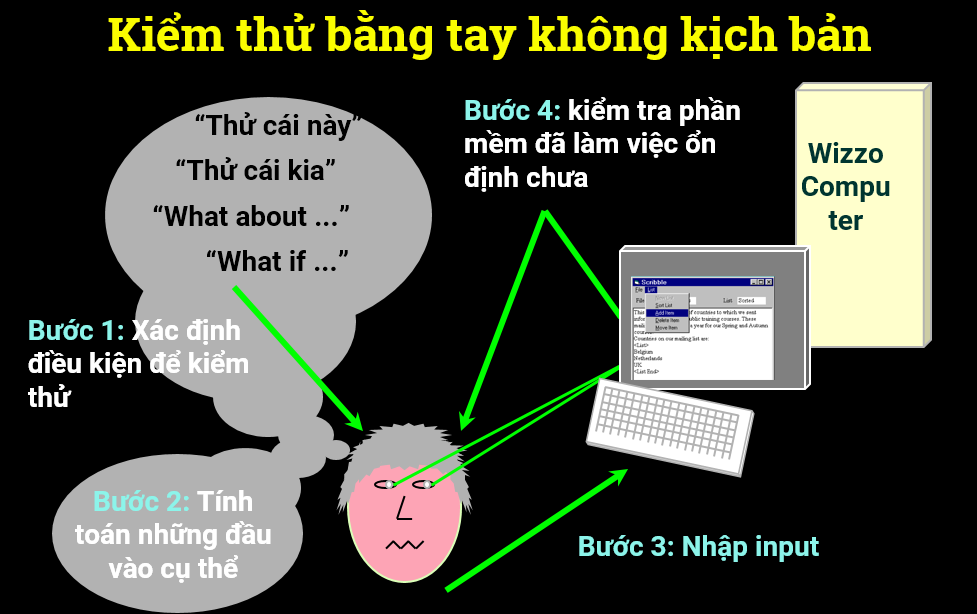
Những quy trình kiểm thử bằng tay hiện nay có thể kể đến:

* Kiểm thử bằng tay không có kịch bản trước
* Kiểm thử bằng tay có kịch bản (không rõ ràng, mơ hồ)
* Kiểm thử bằng tay có kịch bản (chi tiết)

### 3.1.1. Kiểm thử bằng tay không có kịch bản trước

Quy trình này gồm các bước thực hiện:

* Bước 1: Xác định điều kiện để kiểm thử.
* Bước 2: Tính toán những đầu vào cụ thể.
* Bước 3: Nhập đầu vào (input)
* Bước 4: Kiểm tra phần mềm liệu đã hoạt động tốt.

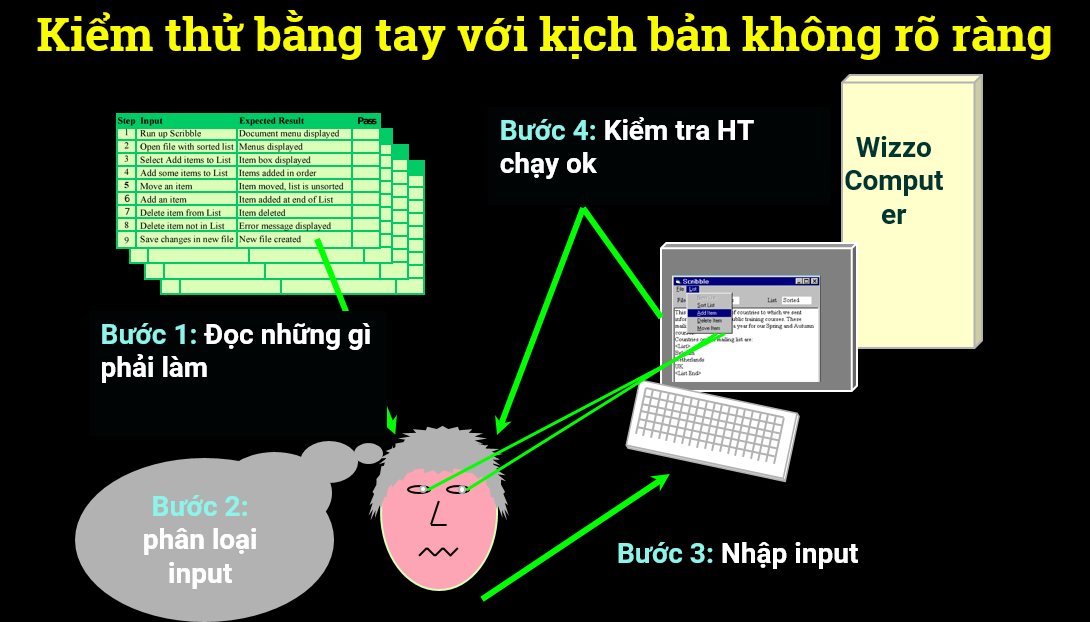


Hình 4.

### 3.1.2. Kiểm thử bằng tay với kịch bản không rõ ràng

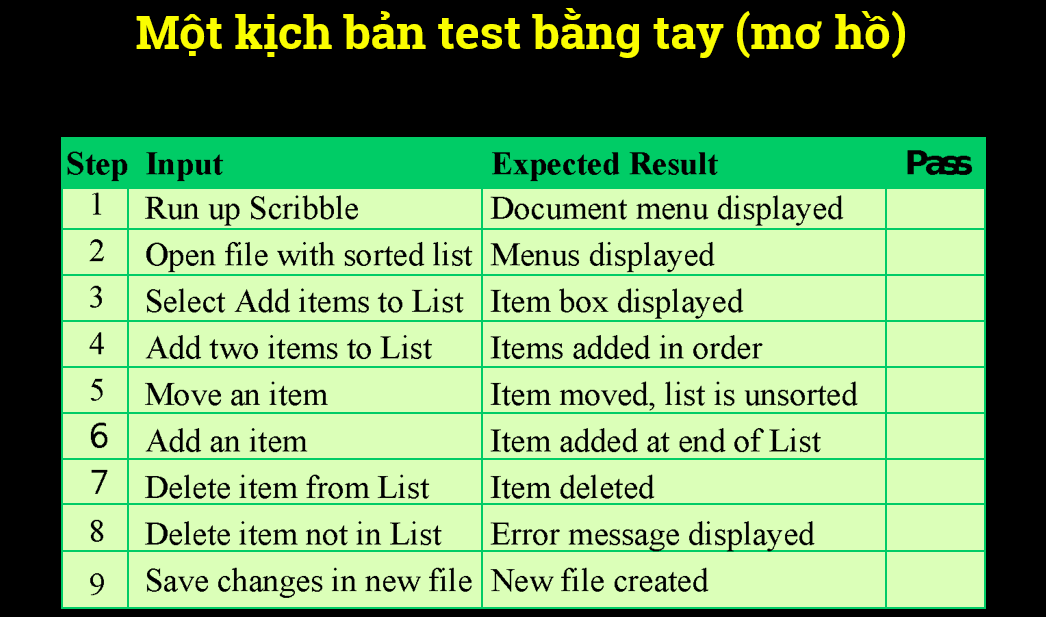
Quy trình này gồm các bước thực hiện:

* Bước 1: Đọc những gì phải làm trong việc kiểm thử
* Bước 2: Phân loại đầu vào (input)
* Bước 3: Nhập đầu vào
* Bước 4: Kiểm tra liệu phần mềm đã hoạt động tốt



Hình 5.

Dưới đây là một ví dụ cho một kịch bản còn thiếu chi tiết:

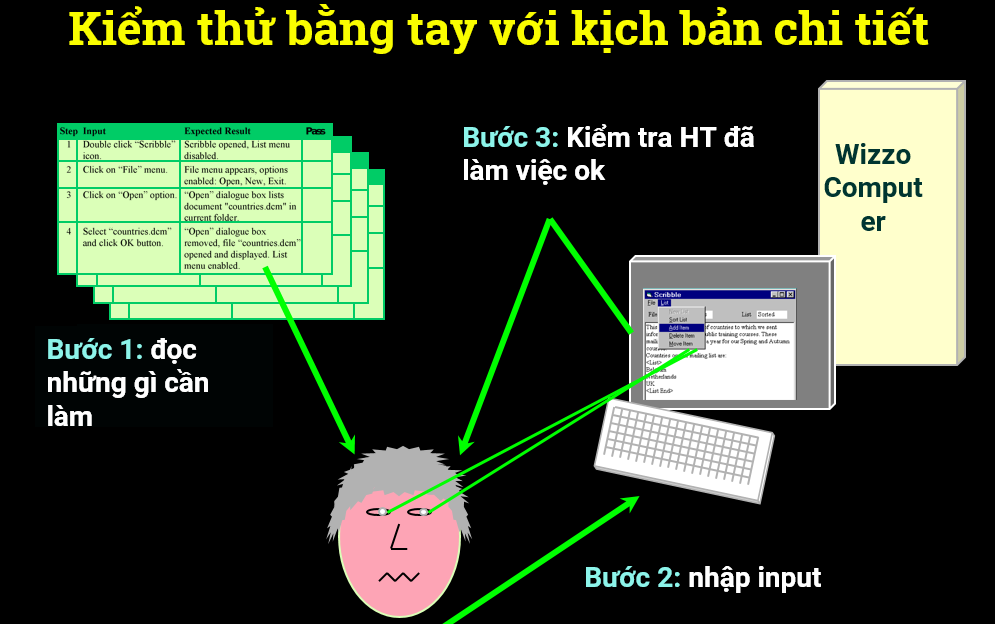


Hình 6.

### 3.1.3. Kiểm thử bằng tay với kịch bản chi tiết.

Quy trình này gồm các bước thực hiện:

* Bước 1: Đọc những gì phải làm trong việc kiểm thử
* Bước 2: Nhập đầu vào (input)
* Bước 3: Kiểm tra xem liệu hệ thống đã hoạt động tốt



Hình 7.

# 4. Best practice

## 4.1. Không tự động hóa quá dài hạn

Các bộ kiểm thử sẽ ngày một lớn cùng với đó chi phí bảo trì cũng sẽ tăng theo, công sức bỏ ra ngày một nhiều mà không mang lại lợi ích gì.

Các bộ kiểm thử cũng có vòng đời của nó, có những trường hợp có thể xảy ra như các kiểm thử viên rời đi, người mới đến, bộ kiểm thử sẽ ngày càng lớn trong khi không thể biết hết được chính xác những việc làm cũng từng người.

## 4.2. Duy trì kiểm soát

Trong suốt quá trình kiểm thử, những ca kiểm thử không quan trọng phải được lược bớt, đồng thời cần tính toán chi phí, lợi ích các hoạt động.

### 4.3. Đầu tư

Công việc kiểm thử cần có sự đầu tư lớn về chi phí, công sức:

* Đảm bảo và bảo trì tài nguyên.
* Ủng hộ việc tự động hóa
* Hỗ trợ kĩ thuật
* Tư vấn/cho lời khuyên
* Lên kịch bản
* Phát triển vào bảo trì thư viện
* Phương pháp phân chia dữ liệu, sử dụng lại

### 4.4. Nên tự động hóa khi nào?

Việc tự động hóa trong kiểm thử nên thực hiện khi chúng ta phải chạy các ca kiểm thử nhiều lần, khi ta nhận thấy làm thủ công sẽ rất tốn kém (về chi phí, thời gian, kiểm thử độ bền độ tin cậy).

Khi việc kiểm thử quá phức tạp và rắc rối, tức khó để biểu diễn được thủ công thì cũng nên hướng tới tự động hóa.

### 4.5. Trường hợp không nên tự động hóa?

Với những tác vụ mà việc chạy kiểm thử không quá thường xuyên hay các lỗi tìm ra khi kiểm thử không quá nghiêm trọng thì không nên thực hiện việc tự động hóa.

Hay đơn giản, chúng ta cần kiểm thử tính tiện dụng của sản phẩm (màu sắc có đẹp không, âm thanh hay không…) thì cũng không cần đến việc tự động hóa.

# 5. Tổng kết

Có rất nhiều loại công cụ khác nhau hỗ trợ cho kiểm thử, được sử dụng cho tất cả các giai đoạn trong quá tình phát triển phần mềm. Việc tự động hóa trong kiểm thử đòi hỏi việc lập kế hoạch chi tiết và công sức, chi phí, thời gian lớn. Việc xác định để lựa chọn áp dụng phương pháp nào trong kiểm thử là rất quan trọng, giúp cho kiểm thử phù hợp với từng giai đoạn và đạt hiệu quả cao.