

Software Testing
ISTQB / ISEB Foundation Exam Practice

1 Tổng quan	2 Lifecycle	3 Kiểm thử tĩnh
4. Kiểm thử động	5. Quản lý	6 Công cụ hỗ trợ

Công cụ hỗ trợ kiểm thử (CAST)

1	2	3
4	5	6

ISTQB / ISEB Foundation Exam Practice

Tool support

Nội dung

Phân loại công cụ

Tại sao ghi lại/làm lại không phải tự động hóa kiểm thử

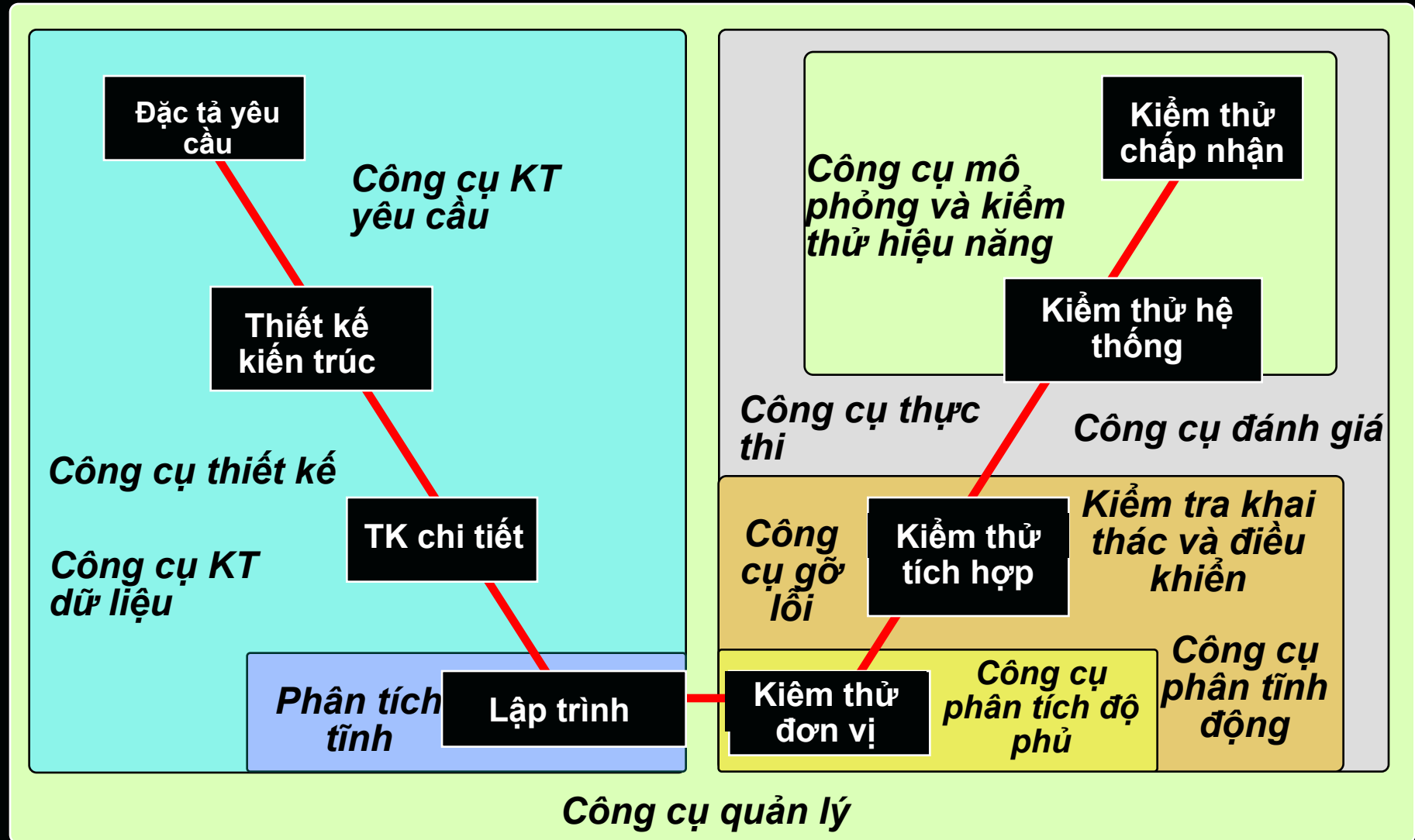
Tự động hóa và kiểm thử là kỹ năng riêng biệt

Best practice

Phân loại các công cụ kiểm thử

1. Công cụ kiểm thử yêu cầu
2. Công cụ phân tích tĩnh
3. Công cụ kiểm thử thiết kế
4. Công cụ kiểm thử dữ liệu chuẩn
5. Công cụ kiểm thử vận hành
6. Công cụ đánh giá các ca kiểm thử
7. Kiểm tra khai thác và điều khiển (test harnesses and drivers)
8. Công cụ kiểm thử hiệu năng
9. Công cụ phân tích động
10. Công cụ gỡ lỗi (debugging)
11. Công cụ quản lý
12. Công cụ phân tích độ bao phủ

Công cụ trong phát triển phần mềm



Công cụ kiểm thử yêu cầu

- **Hỗ trợ tự động cho việc thẩm định và công nhận những yêu cầu theo mẫu có sẵn**
 - Kiểm tra tính thống nhất
 - Linh động

Tool information available from:

Ovum Evaluates Software Testing Tools (subscription service)

CAST Report, 1999

World Wide Web

Công cụ phân tích tĩnh

- 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 (cyclomatic complexity)
 - Vấn đề khác: mức độ lồng nhau , dung lượng

Công cụ kiểm thử thiết kế

- Sinh ra các ca kiểm thử đầu vào
 - Từ một đặc tả chính thức
 - Từ mã nguồn

Công cụ kiểm thử dữ liệu chuẩn

- **Thao tác dữ liệu**

- 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

Các công cụ thực thi

- 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 được sử dụng trong tự động hóa kiểm thử hồi quy

Các công cụ thực thi (2)

- **Dựa vào thuộc tính**

- mô phỏng tương tác người dùng với sản phẩm
- 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)
- Ghi lại sự kiện cho chuyển động chuột, nhấp chuột, hay dữ liệu được nhập từ bàn phím
- Ghi lại các màn hình, hình ảnh, kí tự, trạng thái của đối tượng

Công cụ đánh giá

- 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

Kiểm tra khai thác và điều khiển

- Thường dùng cho những phần mềm chưa 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)

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**

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
 - allocation, use and de-allocation of resources, e.g. memory leaks
 - flag unassigned pointers or pointer arithmetic faults

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 thực thi;
 - Đánh dấu breakpoints hoặc watchpoint ở bất kỳ câu lệnh nào;
 - kiểm tra giá trị của biến hoặc dữ liệu khác.

Công cụ quản lý

- 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).

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.

1	2	3
4	5	6

ISTQB / ISEB Foundation Exam Practice

Công cụ hỗ trợ

Nội dung

Phân loại các công cụ hỗ trợ kiểm thử

Tại sao ghi lại/làm lại không phải tự động hóa kiểm thử

Tự động hóa và kiểm thử là hai kỹ năng riêng biệt

Best practice

Ưu điểm của việc ghi lại “kiểm thử thủ công”

- **Văn bản hóa những gì người kiểm thử đã thực sự làm**
 - 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.
- **Tạo ra một kịch bản chi tiết**
 - Ghi lại các đầu vào thực tế;
 - Người hiểu biết kỹ thuật có thể sử dụng nó 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

Ghi lại kịch bản test

- **Sẽ rất khó hiểu**
 - Vì nó được viết bằng một ngôn ngữ lập trình
 - Trong quá trình bảo trì sẽ cần hiểu hơn, có thể đã được chú thích tự động.
- **Sẽ không chống chịu được nhiều thay đổi trong phần mềm**
 - VD: Một sự thay đổi giao diện đơn giản có thể tác động đến rất nhiều kịch bản.
- **Không bao gồm việc xác minh**
 - Có thể dễ dàng thêm một vài màn hình đơn giản dựa trên việc so sánh.

So sánh “seldom” vs So sánh “often”

Kiểm thử mạnh
Robust test

Kiểm thử nhạy
Sensitive

Dễ phản ứng với thay đổi

Công sức thực thi

Bỏ qua nhiều lỗi

Công sức phân tích lỗi

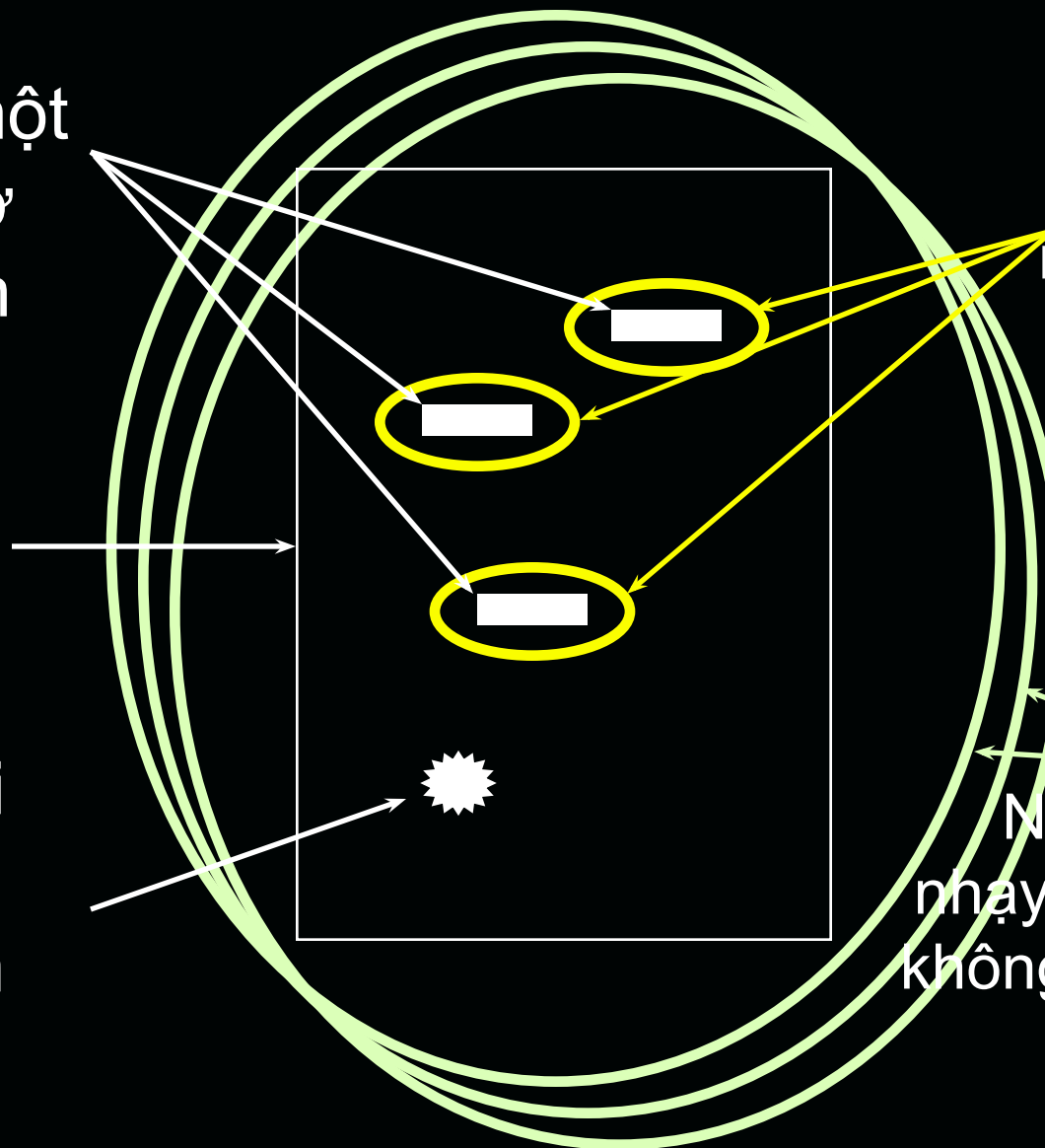
Không gian lưu trữ

Quá nhạy = dư thừa

3 tests, mỗi một test thay đổi ở một khía cạnh khác nhau

Test đầu ra

Những thay đổi không mong muốn xuất hiện ở mỗi test



Nếu tất cả test là mạnh, những thay đổi không mong muốn sẽ được bỏ qua

Nếu tất cả tests là nhạy, những thay đổi không mong muốn sẽ được hiển thị ra

Kiểm chứng tự động hóa

- **Có rất nhiều cách để làm**
 - thi hành động/ gửi(post), so sánh nhiều/ so sánh nhỏ, khả năng thay đổi / hiệu quả tìm bug.
- **Kịch bản có thể trở thành rất phức tạp**
 - càng dễ thay đổi, càng khó bảo trì.
- **Có nhiều tác vụ kéo theo**
 - Tốc độ và độ chính xác của công cụ sử dụng vô cùng quan trọng.
- **Thông thường, có sự kiểm chứng hơn có thể được làm xong**
 - Tự động hóa có thể dẫn tới kiểm thử tốt hơn (*điều này chưa được đảm bảo*)

1	2	3
4	5	6

ISTQB / ISEB Foundation Exam Practice

Công cụ hỗ
trợ

Nội dung

Phân loại các công cụ hỗ trợ kiểm thử

Tại sao ghi lại/ xem lại không phải tự động hóa kiểm thử

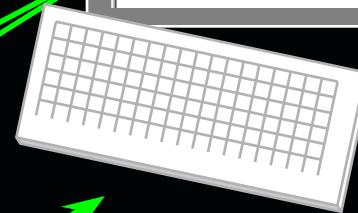
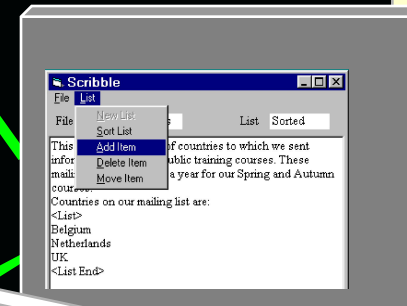
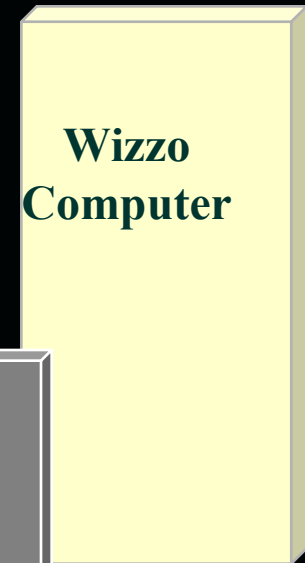
Tự động hóa và kiểm thử là hai kỹ năng riêng biệt

Best practice

Công sức tự động hóa

- **Công sức bỏ ra để tự động cho một test giao động rất lớn**
 - thông thường từ 2 đến 10 lần so với công sức test thủ công
- **và phụ thuộc vào:**
 - công cụ, kỹ năng, môi trường, phần mềm được test
 - tồn tại những cách kiểm thử thủ công có thể là
 - kiểm thử bằng tay không kích bản
 - kiểm thử bằng tay có kích bản trước (mơ hồ)
 - kiểm thử bằng tay có kích bản trước (chi tiết)

Kiểm thử thử công không kịch bản

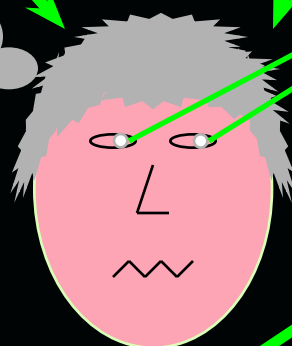


**Bước 4: kiểm tra
nó đã làm việc
ổn**

**Bước 1: Xác
định điều kiện
để kiểm thử**

**Bước 2:
Tính toán
những đầu
vào cụ thể**

Bước 3: Nhập input



Kiểm thử thủ công với kịch bản mơ hồ

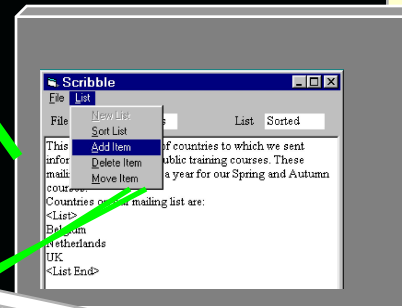
Step	Input	Expected Result	Pass
1	Run up Scribble	Document menu displayed	
2	Open file with sorted list	Menus displayed	
3	Select Add items to List	Item box displayed	
4	Add some items to List	Items added in order	
5	Move an item	Item moved, list is unsorted	
6	Add an item	Item added at end of List	
7	Delete item from List	Item deleted	
8	Delete item not in List	Error message displayed	
9	Save changes in new file	New file created	

Bước 1: Đọc
những gì phải
làm

Bước 2:
phân loại
input

Bước 4: Kiểm tra
HT chạy ok

Wizzo
Computer



Bước 3: Nhập input

Một kịch bản test thủ công (mơ hồ)

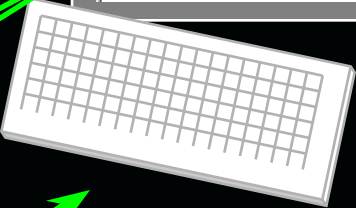
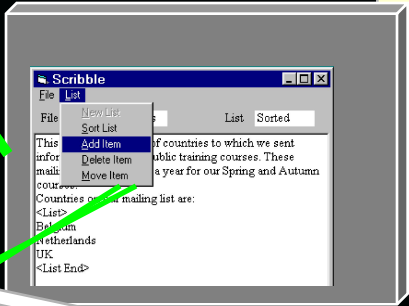
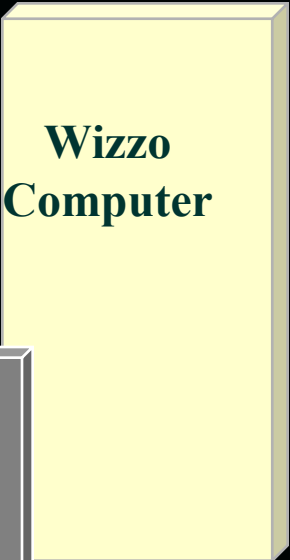
Step	Input	Expected Result	Pass
1	Run up Scribble	Document menu displayed	
2	Open file with sorted list	Menus displayed	
3	Select Add items to List	Item box displayed	
4	Add two items to List	Items added in order	
5	Move an item	Item moved, list is unsorted	
6	Add an item	Item added at end of List	
7	Delete item from List	Item deleted	
8	Delete item not in List	Error message displayed	
9	Save changes in new file	New file created	

Kiểm thử thủ công với kịch bản chi tiết

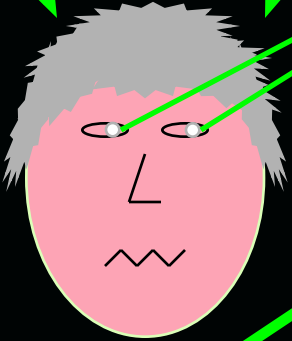
Step	Input	Expected Result	Pass
1	Double click "Scribble" icon.	Scribble opened, List menu disabled.	
2	Click on "File" menu.	File menu appears, options enabled: Open, New, Exit.	
3	Click on "Open" option.	"Open" dialogue box lists document "countries.dcm" in current folder.	
4	Select "countries.dcm" and click OK button.	"Open" dialogue box removed, file "countries.dcm" opened and displayed. List menu enabled.	

Bước 1:
đọc những gì cần làm

Bước 3: Kiểm tra nó đã làm việc ok



Bước 2: nhập input



1	2	3
4	5	6

ISTQB / ISEB Foundation Exam Practice

Công cụ hỗ
trợ

Nội dung

Phân loại các công cụ hỗ trợ kiểm thử

Tại sao ghi lại/xem lại không phải là kiểm thử tự động

Tự động hóa và kiểm thử là hai kỹ năng riêng biệt

Best practice

Đừng áp dụng tự động hóa trong dự án dài hạn

- **Các bộ test càng lớn hơn thì chi phí bảo trì mất nhiều hơn**
 - Công sức bảo trì sẽ được tích lũy dần dần trong khi không mang lại lợi ích.
- **Các bộ kiểm tra cũng có vòng đời của riêng nó**
 - các kiểm thử viên rời đi, hoặc người khác đến. Bộ test càng lớn hơn trong khi không ai biết chính xác tất cả những gì nó đã làm.
- **Những test không còn phù hợp lại được hoạt động**
 - tự động hóa trở thành cái kết cho chính nó

Duy trì kiểm soát

Best
practice

- **Duy trì lược bớt những test không quan trọng**
 - loại bỏ những test: dư thừa, lặp lại.
 - thách thức bổ sung những cái mới sau khi thay đổi..
- **Tính toán chi phí và lợi ích**
 - Chi phí bảo trì
 - Tiết kiệm thời gian công sức, tìm lỗi ra sao?

Đầu tư (Invest)

Best
practice

- **đảm bảo và bảo trì tài nguyên**
 - “ủng hộ” việc tự động hóa
 - hỗ trợ kỹ thuật
 - tư vấn/lời khuyên
- **kịch bản(scripting)**
 - phát triển và bảo trì thư viện
 - phương pháp phân chia dữ liệu, sử dụng lại

Nên tự động hóa tests

Best
practice

- **Chạy nhiều lần**
 - Kiểm thử hồi quy
- **Làm thủ công sẽ tốn kém**
 - Cần nhiều thời gian, chi phí nhân lực
 - kiểm thử đa người dùng, kiểm thử độ bền/độ tin cậy
- **Kiểm thử phi chức năng**
 - hiệu năng, an toàn an ninh và khả năng chịu tải...

Automat
e

Không nên tự động hóa kiểm thử

Best practice

- **Không chạy thường xuyên**
 - Nếu không cần thiết(không tốn kém khi chạy thử công)
 - Kiểm thử 1 lần (Trừ khi có thể xảy ra 1 số lần lặp và chi phí xây dựng có thể được giảm thiểu)
- **Không quan trọng**
 - Sẽ không tìm ra những vấn đề nghiêm trọng
- **Kiểm thử tính tiện dụng**
 - VD: màn hình này màu sắc đẹp?
- **Một số khía cạnh của các ứng dụng đa phương tiện**

Automate

1	2	3
4	5	6

ISTQB / ISEB Foundation Exam Practice

Công cụ hỗ trợ

Tổng kết: Những điểm chính

Có rất nhiều kiểu công cụ hỗ trợ khác nhau cho kiểm thử, bao gồm tất cả các giai đoạn trong phát triển phần mềm.

Tự động hóa đòi hỏi lập kế hoạch và công sức bỏ ra trước.

Xác định và áp dụng các phương pháp phù hợp với từng ngữ cảnh.