TỔNG LƯỢC - ÔN NHANH CÁC KHÁI NIỆM CƠ BẢN CỦA SWT301

1.KIỂM THỬ PHẦN MỀM LÀ GÌ?

-So sánh giữa expected value và actual value. App xử lú tn , ra kq gì (actual value) và lẽ ra, hy vọng nó xử lí phải ra kết quả này (expected value)

- Kiểm tra xem App có được cài đặt, viết code, hành xử như thiết kế, như mô tả, như hứa hẹn với khách hang hay ko??

Hứa hẹn về App sẽ như thế nào nằm trong cuốn tài liệu dự án tên là:

SRS (Software Requirement Specification – SWR302)

FRS (Functional Requirement Specification)

- Đo lường xem hiệu năng hoạt động của app có ổn ko??

- Đo lường xem các đặc tính (attreibute) của App có đảm bảo đc ko

Chính là test/kiểm thử về Non-functional requirement

-Test cái “trải nghiệm” cảm giác “khi xài App”!!!

2. 7 viên ngọc rồng – 7 principles of software testing

- Giúp định hướng cách làm kiểm thử

3. Các level kiểm thử

- Tùy vào mức độ hoàn thiện code, mà ta có các công việc kiểm thử tương ứng. Code hoàn thiện đến đâu ta sẽ có các hoạt động kiểm thử đến đó.

Có 4 mức hoàn thiện code/ hoàn thiện App tương đương có 4 mức độ kiểm thử:

\* Mức 1: UNIT TESTING – KIỂM THỬ KHI VỪA XONG 1 HÀM/METHOS/CLASS

Dùng unit test framwork : Junit, Nunit,xUnit, TestNG, Mocha,..

\* Mức 2 : Integration testing – Kiểm thử tích hợp

Kiểm thử khi vừa xong 1 tính năng – vừa ráp UI vào với code UI+code +DB -> ra 1 chức năng

VD: Test chức năng Login, dùng tay test, hoặc Selenium

\* Mức 3 : System testing – Kiểm thử hệ thống

Kiểm thử khi App đã hoàn tất các tính năng , App hình thành rồi, giờ là lúc test các chức năng!!!

Dùng sức người để nhập info, nhấn nút, xem kq xử lí của App và kết luận tính năng chạy đúng, sai

>> Manual Testing

Dùng các thư viện phụ trợ, viết code, để tự động hóa quá trình nhập data vào App, tự động nhấn nút, tự động verify data về từ App, kết luận giúm đúng sai qua 2 màu xanh đỏ. Tăng tốc quá trình kiểm thử , và nếu cần kiểm thử lại thì cũng nhanh luôn , thứ đã từng kiểm thử trước thì cũng nhanh hơn,

Dùng code để tự động hóa Manual Testing

>> Automated Testing, Test Automation

Dùng Selenium(Webb app), Appium(Mobile, Desktop)

Code Selenium, tự động hóa việc kiểm thử, thay sức QC = chạy code ; code này đc gọi là test script!!!

Hôm bữa đã test và thấy chức năng chạy ổn , an hem tiếp tục upade App/code , vậy thì chức năng hôm bữa liệu còn ổn?? Phải test lại những chức năng đã từng test

Chỉ cần chạy lại code test là đử nhanh, xong nhanh

Test lại những thứ đã từng test -> Rgresstion test kiểm thử hồi quy

\* Mức 4: UAT – User acceptance testing

-Đưa App cho khách hang, bên đặt hang, user sử dụng “thử” để đánh giá , để tìm sai sót khiếm khuyết. Vượt qua được app đc user yêu thích, pass UAT, chính thức đc chấp nhận và đưa vào sử dụng – nghiệm thu

- Nếu là App Generic , App làm cho số đông: chính là phiên bản download trên mạng để dùng thử: alpha, beta, preview, rc,….

-Nếu là App Custom, Customized , Bespoke, cài đặt trên máy người đùng, khách hang, máy công ty đặt hang cho nhân viên dùng thử!!!

4. TESTING TYPES

- Đề cập đến cái mục tiêu cần kiểm thử, đối tượng cần kiểm thử

- Tức là trả lời câu hỏi “test cái gì” cho mục đích gì, test để đo lường điều gì?

Ví dụ: Test xem chức năng login chạy đúng như thiết kế hay ko (hộ trợ 3 hình thức:user-pass, GM,FB; khóa account x phút/giờ/ 4ever hay ko nếu login sai nhiều lần để xuất nút reset password, Signup hay ko tùy ngữ cảnh)

Test xem chức năng tạo mới đơn hang ở quầy thu ngân hành xử hợp lí ko, chạy ổn các case hay ko??

(có option tích điểm ko KHTT hay ko, nhập món mua bằng Barcode sanner , gõ phím, gõ lộn món thù phải làm sao??, thanh toán = cash, ví, ví nào??)

>>Muc tiêu : đo hiệu năng, - khả năng hành xử ,xử lí

Test xem trong 30 phút sales sập sàn, App chết ko ,treo ko ???

>>>Mục tiêu: đo hiệu năng – khả năng hành xử, xử lí

- Có 1 vài loại kiểm thử phổ biến - FUNCTIONAL Testing(SWR302)

- Kiểm thử Phi chức năng – non-funcional testing

-Đo lường các non-functinal reqs đạt hay ko>>

\* KIỂM THỬ KIẾN TRÚC APP, CODE CỦA APP ỔN KO ??

Structural Testing, Architectural Testing

- Đọc code, xem code!!

\* Kiểm thử hồi quy

-Test lại những thử đa từng test khi App có thay đổi code

\* Kiểm thử khi nâng cấp/bảo trì hệ thống

- Khi nâng cấp server, phần cứng , thiết bị, liệu app còn chạy ổn ko ??(Change Related Testing)

TESTING TYPES CÒN ĐƯỢC CHIA THÀNH VÔ SỐ LOẠI NHỎ HƠN

5. TESTING TECHNIQUES

- Là những hành động cụ thể để THỰC HIỆN HÓA TESTING LEVELS, TESTING TYPES

- Tức là muốn Test 1 chức năng chạy đúng hay sai??

-Tức là muốn Test 1 chức năng chạy đúng hay sai (Types) thì ta cần làm những việc gì

-Tức là muốn Test 1 hàm chạy đúng hay sai (Unit Test/Level) thì ta cần làm những việc gì??

- Teachniques can thiệp/ liên quan đến cả Levels, Types

Những hành động cụ thể để tiến hành việc kiểm thử, tìm bug, tìm sai sót của App, của code.

* Là những kĩ thuật thiết kế ra các Testcase, kĩ thuật tím ra những tình huống kiểm thử, tìm ra các nhánh các ngóc ngánh cần kiểm thử 1 hàm, 1 chức năng!!  
  Cần bn testcase để chứng minh rằng 1 hàm/method/class/ chức năng là ổn

Có 2 Loại kĩ thuật kiểm thử - 2 cách cụ thể tiến hành kiểm thử static testing và dynamic testing

5.1 Static testing

-Là kĩ thuật kiểm thử mà ko dùng đến App, ko chạy App , do có thể App chưa xong, code chưa xong.

-Là kĩ thuật kiểm thử dựa trên “Đọc , Review” các tài liệu của dự án , đó thường là : SRS, FRS, các diagram đã vẽ (ERD, Class diagram)

- Dân BA phối hợp với dân QC cùng xem các tài liệu dự án để tìm sai sót, để chỉnh sửa giúp mình

- Các kĩ thuật review document có thể xem là (xem CheatSheet)

- Walkthroughs

- Inspection

- Technical review

- Peer review

- …….

5.2 Dynamic testing

- Là kĩ thuật kiểm thử mà phải có App, có code , phải run đc App  
Chia thành 2 nhánh chính nhỏ hơn:

\*White-box testing (glass-box testing)  
Hộp trắng: bạn có cái đồng hồ, mặt lưng của nó trong suốt nhìn thấy bánh rang, bánh đà đang lắc qua lại, mặt trước thấy kim bánh rang đang xoay

-Cho nhìn thấy hoạt động bên trong.

- Mở code ra đọc , xem chạy thử

-Dùng Unit test

\*Black-box testing:

Hộp đen: đứng trước máy ATM rút tiền , nghe máy nó kêu, .. -> $ ra ta xài nhưng ko biết hoạt động bên trong thiết bị!!!

-Khong cho xem hoạt động bên trong

-Ko care về code chỉ care tính năng, chạy app qua ui, ko cần quan tâm sau ui code thế nào!!!

-Dùng Selenium test chức năng login!!

--==>Cho dù hộp đên hay hộp trắng, điều quan trọng nhất là các kĩ thuật thiết kế , nhận ra các test case, cần test những tình huống nào khi xài hàm, cần test những tình huống nào khi xài chức năng

VD: hàm getFactorial(int n) -> long cần test những tình huống nào ?? Có những case nào??

Case: n<20; n 2..20; n >20

Chức năng Login -> cần test những tình huống nào ??

Case: user- pass -> case nhở hơn, empty, user sai, pass sai/..

PB

CÓ CÁC CÁCH THỨC CƠ BẢN SAU ĐỂ THIẾT KẾ CÁC TESTCASE(chung cho cả Black- box, và White – box , thường dành cho Black-box nhiều hơn)

-EQUIVALENCE PARTITIONING – PHÂN VÙNG TƯƠNG ĐƯƠNG

-BOUNDARY VALUE ANALYSIS – PHÂN TÍCH GIÁ TRỊ BIÊN

-DESCISON TABLE – BẢNG QUYẾT ĐỊNH

-USE CASE TESTING – KIỂM THỬ TRÊN CÁC USE CASE

-STATE TRANSITION TESTING – Kiểm thử dựa trên sự thay đổi trạng thái

TESTING LEVELS vs. TESTING TYPES vs. TESTING TECHNIQUES

4 LEVELS Functional Testing

Non-functional Testing

Mức độ hoàn hiện ||Phân loại kiểm thử: | Black-box

Của code, của App ||-Dựa trên mục tiêu, đối| White -book

| Tượng cần kiểm thử |Kĩ thuật thiết kế các Test C

| -Mục tiêu cần kiểm thử |Kĩ thuật , cách thức làm

| Là gì , đo lường điều gì | cụ thể kiểm thử   
 | Đặt mục tiêu test | Hành động test

====================================

[NGOẠI TRUYỆN – ĐỀ THI PE]

* Đề thi PE cho bạn 1 tình huống về việc làm 1 cái App nào đó, ví dụ:

App quản lí trung tâm ngoại ngữ, App quản lí thư viện, App Quản lí tiệm thú cưng

* Đề thi giới thiệu rằng App có tính năng gì (UI + code + xử lí )
* Đề thi giới thiệu rằng App có hàm gì (code)

Đề thi yêu cầu dùng các kĩ thuật hộp đen đã new ở trên để thiết kế ra test case dùng cho cả 3 level viết app: Unit testing, Integration Test, System test

6.TESTING DOCUMENTATION

-Để kiểm thử 1 cái App thì gồm nhiều công ty liên quan đến QC Team và Dev Team

-QC Team phải:

- Lên kế hoạch phân bổ, phân chia các công việc/ đầu việc kiểm thử

Ai – QC nào sẽ tham gia ngày giở tham gia test, trang thiết bị cần thiết ,..

-Tham gia dự án từ sớm , từ lúc lấy requirement, để biết về cái app sắp phải test

-Review Specification(SRS, FRS) của BA – STATIC TESTING

-Thiết kế các test case (tình huống kiểm thử) – dùng các kĩ thuật ns ở trên

-Tiến hành kiểm thử - tìm ra BUG

- Ghi nhận kết quả kiểm thử : PASSED, FAILED, FAILED ghi cái BUG tìm thấy lên sever để nhóm dev thấy 🡪 Log bug!!!

Toàn bộ hành động của dân QC phải được ghi chép lại , trong file docx, excel, hoặc oniline tool(jira, Bugzilla, Testrail) theo chuẩn định sẵn, để đễ dàng liên thông, ai cũng hiểu

-chuẩn Document liên quan đến các hoạt động của nhóm test tên là IEEE-829(I triple e)

(các template có sẵn của Hội Điện tử Máy tính Hoa kì)