ZADANIE\_1

1. *Podaj swoją definicję błędu?*

Wadliwa część oprogramowania, kodu powodująca nieprawidłowe działanie programu lub jego całkowitą dysfunkcję

1. *Napisz w kilku zdaniach, w jakim celu testujemy oprogramowanie.*

Oprogramowanie testuje się po to, aby wyeliminować jak największą ilość istniejących lub potencjalnych błędów mogacych pojawić się podczas docelowej, produkcyjnej pracy programu/aplikacji.

1. *Kiedy możemy uznać, że produkt jest w 100% przetestowany?*

Trudno stwierdzić czy w ogóle można wysnuć tak smiałą tezę. Z uwagi na skomplikowanie i zaawansowanie obecnych urządzeń - a co z tym związane – skomplikowaniem funkcjonujących na nich aplikacji w mojej ocenie brak jest realnej możliwości sprawdzenia wszelkich mozliwych przypadków i potencjalnych scenariuszy prowadzących do zaistnienia błędu.

Możemy siebie oszukiwać, że produkt jest już przetestowany na 10 minut przed uruchomieniem programu/aplikacji na produkcji karmiąc się jednocześnie nadzieją, że nie pojawi się żaden critical :)

1. *Opisz/Narysuj cykl życia znalezionego błędu.*

Opiszę to na podstawie moich niewielkich doświaczeń z testami aplikacji - w pewnym Towarzystwie Ubezpieczeniowym (A.D 2014)

BŁĄD----> NAJPIERW GROMKIE: "*NIE DZIAŁA!!*"---->TICKET W JIRA – DO CHŁOPCÓW Z IT---> 3 MANDAY'E/1.500$ później: *"SPRAWDŹ, NIE ZAMYKAŁEM TICKETA -jak nie działa to przypisz do Darka"---->TESTY WG. SCENARIUSZY TESTOWYCH (dwie opcje):*

*---->JEŚLI JEST OK, BŁĄD NIE POJAWIA SIĘ = ZAMYKAMY TICKET W JIRA, PRZECHODZIMY DO DALSZYCH CZYNNOŚCI – był błąd, nie ma błędu :)*

*---->BŁĄD POJAWIA SIĘ DALEJ ---> PRZYPISUJEMY TICKET DO DARKA I CZEKAMY NA ODPOWIEDŹ*

Tak czy owak każda droga prowadzi do wyeliminowania błędu jak najmniejszym nakładem z jak najlepszym skutkiem

1. *Opisz wady i zalety testów manualnych i automatycznych.*

*TESTY MANUALNE*

*- WADY: podatność na przysłowiowy czynnik ludzki, ograniczona zasobami ludzkimi i wielkością ekspresy do kawy w firmie produktywność, szybkośc realizacji*

*- ZALETY: relatywnie niedrogie i nie wymagające dużych nakładów finansowych, możliwośc odtworzenia typowo ludzkich zachowań i ludzkiej skłonności do zepsucia czegoś co działa, brak konieczności angażowania programistów ($$$)*

*TESTY AUTOMATYCZNE*

- WADY: cena, wymagana znajomośc programowania, możliwość wykonywania testów wydajnościowych np. 1000 logowań jednocześnie, możliwość wykorzystania w innych projektach po odpowiedniej adaptacji

- ZALETY: brak podatności na błędy (jeśli programista się wykazał), możliwość testowania 24/7, szybkość wykonywania, powtarzalność,

1. *Na jakim etapie wytwarzania oprogramowania tester powinien rozpocząć*

*testowanie?*

Moim zdaniem tester powinien być obecny "*od początku"* już na etapie projekowania aplikacji aby móc zacząć intelektualnie pracować nad przypadkami testowymi. Testy jako testy same w sobie powinny rozpocząć się niezwłocznie od momentu pierwszego uruchomienia aplikacji/programu

1. *Czym są testy pozytywne, a czym negatywne?*

Testowanie pozytywne to rodzaj testowania mający na celu doprowadzenie do oczekiwanego rezultatu – scenariusze mają doprowadzić np. do przejścia do okna *"settings"* w aplikacji z każdego innego dostępnego okna aplikacji

Testowanie negatywne to rodzaj testowania mający na celu doprowadzenie do zainicjowania błędu, zawieszenia się aplikacji, przejścia na niewłaściwe okno np. kliknięcie 10 razy na button w jak najkrótszym czasie, klikanie pomiędzy oknami bez ładu i składu :)

Tak mi się przynajmniej wydaje, takie Jin i Jang.

SCENARIUSZ TESTOWY NUMER 1:

SPRAWDZENIE FUNKCJI "SEARCH" – sprawdzenie funkcjonalności wyszukiwarki ofert pracy na witrynie internetowej Klienta

PRZYPADKI TESTOWE:

1.

* sprawdzenie funkcjonalności wyszukiwania na podstawie słów kluczowych POLE "KEYWORD"
* 1. Wpisać treść w pole "KEYWORD" 2. OCZEKIWANY REZULTAT: pokazuje sie lista rozwijana zawierająca treść, jaka została wpisana. W przypadku braku takiej treści lista rozwijana nie wyświetla się

2.

* sprawdzenie funkcjonalności pola LOCATION
* 1. Kliknąć w pole location. 2. OCZEKIWANY REZULTAT – pojawia się lista rozwijana z lokalizacjami tj. nazwami miast i państw
* 1. Kliknać w pole location, po pojawieniu sie rozwijanej listy wybrać pierwszą pozycje z listy. 2. Nacisnąć przycisk serach. 3. OCZEKIWANY REZULTAT: poniżej pojawiają się hyperlinki do ofert pracy znajdujących sie wybranej lokalizacji. W przypadku braku ofert pracy w danym regionie pojawia się komunikat wskazany przez Klienta – UWAGA – powtórzyć czynności dla każdej lokalizacji z listy rozwijanej

Mógłbym tak stworzyć jeszcze kilkadziesiąć róznych scenariuszy dla jednej tylko funkcji – ale niestety praca na kontrakcie B2b + dwójka dzieci bardzo "dba" aby mój czas wolny był poddany odpowiedniemu reżimowi.

1. Niestety nie miałem nigdy styczności z testami obciążeniowymi, wydajnościowymi ale zakładam, że tutaj niezbedny będzie automatyzm.

A. TEST WYDAJNOŚCIOWY – przygotowany uprzednio przez testera automatyzującego/progrmistę program rozpoczyna logowanie się na witrynie od 1 logowania, co 3ms dodaje kolejne logowanie w na zasadzie liczba poprzenich logowań +1. Test trwa do momentu :

- uzyskania 1000 jednoczesnych logowań z zachowaniem czasu lowowania, funkcjonalności i bezpieczeństwa logowania <=3ms – przy pozytywnym wyniku program wstrzymuje działanie wyświetlając komunikat o pozytywnym wyniku testu

- przekroczenia czasu logowania pojedynczego użytkownika powyżej 3ms przy liczbie jednoczesnych logowań poniżej 1000 (przy zachowaniu funkcjonalności i bezpieczeństwa logowania) - w tej sytuacji program wstrzymuje działanie i wyświetla komunikat o przekroczonym czasie logowanie

B+C. przygotowany uprzednio przez testera automatyzującego/progrmistę program rozpoczyna logowanie się na witrynie od 1000 jednoczesnych logowań. Co 10 ms dodaje kolejne logowanie na zasadzie liczba poprzednich +1 aż do uzyskania 1500 jednoczesnych logowań przy zachowaniu funkcjonalności i bezpieczeństwa logowania. W przypadku pozytywnego wyniku program ponownie rozpoczyna logowania na zasadzie liczba porzednich logowań +1 - zaczynajac od 1000 jednoczesnych logowań natomiast czas pomiędzy logowaniami jest zmniejszany o 1ms do momentu uzyskania przerwy pomiędzy logowaniami 3ms. W przypadku negatywnego wyniku testu czas pomiędzy kolejnymi pojedyńczymi logowaniami na zasadzie poprzednia liczba jednoczesnych zalogowań + 1, jest zwiększany o 1 ms zaczynająć od 10ms – aż do uzyskania pozytywnego wyniku testu jednoczesnych logowań w liczbie 1500.

1. Załączę log z baterii na GIT HUB do 23:59 2 grudnia 2021 roku, – niestety ale mój prywatny komputer nie za bardzo chce się połączyć z moim telefonem, natomiast na służbowym komputerze nie chcę wykonywać takich czynności. Liczę na akceptację czasu
2. Dzieci są najlepszymi testerami, gdyż mają nieszablonowe pomysły, nie zakykają się w schematach i spontanicznie potrafią wykonać czynności, których dorosły człowiek by nie wykonał z uwagi na wypracowane zachowania, doświadczenie itd.
3. Testowanie kończy się jedynie w momencie wygaszenia, usunięcia danego programu/ aplikacji. Zawsze można dodać kolejną funkcjonalność, zmienić jakikolwiek element a to wymaga kolejnych testów
4. Gdyby nie moje bogate doświadczenie zawodowe być może miałbym problem z odpowiedzią na pytanie i proszę mi wierzyć lub nie – nie googlowałem tego :) Okrągły właz nigdy nie wpadnie do studzienki, gdyż pierścień montażowy ma mniejszą średnicę niż studzienka – jest to fizycznie nie możliwe:)