

Zadanie_1_QA

Zad 1. Czym się różnią testy funkcjonalne od нефункциональных?

Tabela 1. Różnice między testami funkcjonalnymi a нефункциональными

	testy funkcjonalne	testy нефункциональные
Testowanie wymagań	funkcjonalnych	нефункциональных
Dotyczą	tęgo jak system wykonuje funkcje, które ma za zadanie udostępniać (testowane są zaprojektowane funkcje systemu, czy system zachowuje się w zaprojektowany sposób)	cech jakościowych systemu, niezwiązanych z określoną funkcją bądź działaniem użytkownika
Przykłady	testowanie użyteczności, testy jednostkowe, testowanie zachowania systemu zgodnie z dokumentacją wymagań funkcjonalnych, testowanie zabezpieczeń, testowanie współdziałania, testowanie zdolności oprogramowania do współpracy z innymi modułami lub systemami, testy akceptacyjne użytkownika, testy dymne / testy kondycji, testy regresji, testowanie interfejsu oraz back-end'u.	testowanie zachowania systemu zgodnie z dokumentacją wymagań нефункциональных, testowanie wydajnościowe, testowanie obciążeniowe, testowanie przeciężeniowe, testowanie użyteczności, testowanie pielęgnowalności, testowanie niezawodności, testowanie przenaszalności.

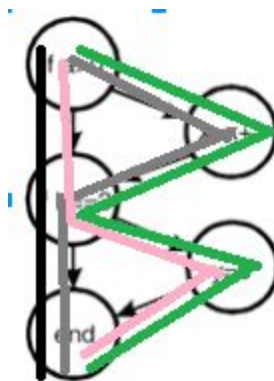
Zad 2. Co to są smoke testy i testy regresji? Kiedy je stosujemy?

Tabela 2. Smoke testy i testy regresji

	smoke testy	testy regresji
Co to jest?	Smoke test mówi nam, czy program/system da się uruchomić, czy jego interfejsy są dostępne i czy reagują na działania użytkownika.	Test regresji mówi nam, czy aplikacja działa po dokonaniu w niej modyfikacji, poprawieniu błędów lub po dodaniu nowej funkcjonalności.
Kiedy stosowany?	<ul style="list-style-type: none">- tuż przed oddaniem wersji aplikacji (programiści)- przed zaakceptowaniem otrzymanej do testów aplikacji (testerzy)	<ul style="list-style-type: none">- po smoke testach lub testach typu sanity

Zad 3. Ile przypadków testowych potrzeba, aby pokryć wszystkie możliwości?

Potrzebne są 4 przypadki testowe.



Zad 4. Co to jest testowanie zwinne?

Jest to metoda testowania stosowana w projektach wytwarzanych w oparciu o metodyki agile'owe (np. scrum). Kładzie nacisk na paradygmat "najpierw przygotuj testy" oraz traktuje samo wytwarzanie oprogramowania jako klienta testowania.

Zad 5. Dany jest input „wiek”, który przyjmuje wartości od 18 do 60. Twoim zadaniem jest przetestować go za pomocą techniki wartości brzegowych. Jakie wartości wpisujesz do inputu, podaj wszystkie liczby, które wpisujesz.

Liczby to: 18, 60, 17, 61, 40

Zakładając że wiek (x) mieści się w przedziale 18 do 60 z tymi wartościami włącznie, czyli $18 \leq x \leq 60$, przetestowałabym wartości minimalne i maksymalne z zakresu, czyli 18 oraz 60 (mógł wkraść się błąd taki, że $x > 18$ zamiast $x \geq 18$ oraz $x < 60$ oraz $x \leq 60$). Oprócz wymienionych wartości przetestowałabym również 17 (ponieważ mógł być błąd, taki że zamiast $x \geq 18$ zostało zaimplementowane $x < 18$), 61 (ponieważ mógł być błąd, taki że zamiast $x \leq 60$ zostało zaimplementowane $x > 60$) oraz którąś z wartości środkowych np. 40 (ponieważ mógł być błąd, taki że zamiast $x \leq 60$ i $x \geq 18$ zostało zaimplementowane $x = 18$ i $x = 60$).

Zad 6. Dołączasz do projektu w trakcie developmentu aplikacji, do której nie ma dokumentacji...

Zadałabym pytania związane z wymaganiami dotyczącymi logowania, czyli:

- czy wielkość znaków w nazwie użytkownika ma znaczenie,
- co ma się dzieć, gdy użytkownik o danej nazwie nie istnieje,
- czy wielkość znaków w hasle ma znaczenie,
- co ma się dzieć po niepoprawnym wpisaniu hasła (jeżeli ma być "blokowane" konto, to po ilu razach niepoprawnego wpisania hasła),
- czy hasło przy wpisywaniu powinno być "chowane" (hasło symbolizując gwiazdki bądź inne znaczki).
- jakie są warunki wstępne rozpoczęcia testowania logowania,
- skąd pobierane są dane użytkownika (założmy, że jest o baza danych: jak nazywają się pola nazwy użytkownika oraz hasła),
- jakie są dane testowe (nazwa użytkownika i hasło),
- czy do zalogowania się, niezbędne jest połączenie z Internetem (jeżeli tak, to co ma się dzieć jeżeli połączenie z siecią jest niestabilne / brak połączenia z Internetem),
- czy istnieje możliwość zalogowania się do jednego konta z dwóch lub więcej urządzeń jednocześnie (jeżeli nie, to jak powinna się zachowywać aplikacja w sytuacji, gdy będę chciała zalogować się do konta z dwóch urządzeń jednocześnie),
- w jakich sytuacjach powinny pojawić się komunikaty błędów,
- jakie konkretnie komunikaty powinny się wyświetlać przy błędach w wypełnianiu formularza logowania,

- czy istnieje możliwość zalogowania się jako gość (jeżeli tak, to jak ma to wyglądać).

Zadałabym również pytania do osobnych, ale bezpośrednio związanych "ficzerów" (myślę, że tego typu informacje byłyby również przydatne), np.:

- czy jest możliwość przywracania hasła (jeżeli tak, to jak wygląda proces przywracania hasła) ,
- jak wygląda wylogowanie z aplikacji ,
- czy istnieje tzw. pytanie pomocnicze ,
- czy istnieje możliwość zmiany hasła, jak tak to w jaki sposób,
- jak długo ma trwać sesja użytkownika zalogowanego (czy w ogóle są względem długości sesji jakieś ograniczenia, co ma się dzieć po ewentualnym wygaśnięciu sesji),
- czy istnieje możliwość zapamiętywania hasła i użytkownika (jeżeli tak to jakie warunki muszą być spełnione).

Spytałabym się też o przebieg rejestracji. Przy testowaniu logowania (nawet posiadając jakieś konto testowe do logowania) chciałabym sprawdzić, czy po dokonaniu rejestracji, będę miała dostęp do utworzonego przeze mnie konta. Byłyby to na przykład pytania typu:

- jaka jest minimalna i maksymalna długość hasła użytkownika,
- czy w hasle mogą występować znaki specjalne i liczby (jeżeli tak to czy obowiązkowym jest zastosowanie znaków specjalnych/liczb w nazwie (jeżeli tak to jaka jest minimalna i maksymalna ilość tych znaków)),
- jaka jest minimalna i maksymalna długość nazwy użytkownika,
- czy w nazwie użytkownika mogą występować znaki specjalne i liczby (jeżeli tak to czy obowiązkowym jest zastosowanie znaków specjalnych/liczb w nazwie (jeżeli tak to jaka jest minimalna i maksymalna ilość tych znaków)),
- czy nazwa użytkownika jest e-mailem (wtedy trzeba wykryć znak @ i sprawdzić poprawność domeny email).

Zad 7. Co jest celem testowania?

Celem testowania jest:

- weryfikacja działań deweloperów (względem wymagań co do aplikacji),
- walidacja zgodności oprogramowania z wymaganiami użytkownika,
- podnoszenie jakości oprogramowania ("uwalnianie" oprogramowania od bug'ów"),
- dostarczenie wiedzy na temat stanu produktu,
- zmniejszenie ryzyka wystąpienia awarii w czasie użytkownika oprogramowania,
- ochrona przed stratą czasu (szybkie wykrycie błędu jest mniej czasochłonne do naprawienia, niż po dodaniu np. kolejnych funkcjonalności)
- ochrona przed stratą pieniędzy (czas to pieniądz :)),
- ochrona przed utratą reputacji (wystąpienie awarii bądź błędów może nieść negatywne skutki dla firmy posiadającej system)

Zadanie praktyczne

Numer założonego taska: **id 0000092**

PS. zostało wybrane "urgent" ze względu na fakt, że na naszej uczelni są studenci nieznający języka polskiego. Wtenczas, nie rozumieją oni co znajduje się na lewym panelu strony co może im znacznie utrudnić pracę z e-dziekanatem. Należy to szybko naprawić.

Zadanie_2_QA

Zad 1. Oceń, czy możliwe jest przetestowanie aplikacji w 100%? Odpowiedź uzasadnij.

Przetestowanie aplikacji w 100% nie jest możliwe. Związane jest to między innymi z:

- ograniczeniem czasowym (deadline na oddanie projektu) i ograniczeniem budżetu przeznaczonym na testy,
- brakiem możliwości wprowadzenia wszystkich możliwych wartości dla wszystkich możliwych kombinacji,
- brakiem możliwości przetestowania aplikacji w każdym możliwym środowisku oraz dla wszystkich konfiguracji sprzętu,
- zmianą, dodawaniem nowych wymagań,
- dodawaniem nowych funkcjonalności, rozbudowaniem systemu,
- nieprzewidywalnością zachowań użytkowników systemu,

Mniejsze projekty "łatwiej" dokładniej przetestować niż duże, rozbudowane funkcjonalnie systemy (o dużej, skomplikowanej infrastrukturze). Należy również pamiętać, że testowanie nie udowadnia braku błędów.

Zad 2. Kiedy można zakończyć testowanie?

Testowanie **trzeba** zakończyć wtedy, gdy:

- kończy nam się deadline (trzeba oddać projekt), bądź kończy się budżet na testowanie,
- współpracujący system jest uruchomiony (dla poprawnego działania bądź uzyskania pewnych korzyści biznesowych nasz system musi być również wdrożony),
- klient chce żebyśmy przestali testować.

Testowanie **można** zakończyć wtedy, gdy:

- ustalony procent przypadków testowych nie wykrywa błędów,
- krzywa wykrywania błędów spadła poniżej ustalonego progu,
- pokrycie kodu/funkcjonalności/wymagań, osiągnęło założony poziom,
- oprogramowanie spełnia założone wymagania jakości,
- korzyści z kontynuowania testów przestają rekompensować koszty ponoszone na testowanie.

Zad 3. „It's not a bug, it's a feature”. Spróbuj uzasadnić różnicę jednym zdaniem

Bug to niecelowe/przypadkowe/niezaplanowane/niezamierzone zachowanie systemu (będące wynikiem błędnej interpretacji dokumentacji bądź braku interpretacji dokumentacji), feature to celowe/zamierzone, udokumentowane zachowanie systemu.

Zadanie praktyczne

ID	1.0
Tytuł	Poprawne dodawanie kontentu do słownika
Środowisko	SoapUI

Warunek wstępny	Uruchomiona aplikacja (słownik), brak identycznej pary słów w bazie danych
Kroki do wykonania	0. Dodanie nowego requestu w operacji AddWord 1. Uzupełnienie parametru wordPLN (wartościami varchar 255) 2. Uzupełnienie parametru wordEng (wartościami varchar 255) 3. Wywołanie zapytania
Oczekiwany rezultat	Dodanie kontentu do słownika, zwrócenie komunikatu "Dodano do bazy danych"
Warunki końcowe	Kontent dodany do słownika

ID	2.0
Tytuł	Odmowa dodania kontentu do słownika
Środowisko	SoapUI
Warunek wstępny	Uruchomiona aplikacja (słownik), istnienie identycznej pary słów w bazie danych
Kroki do wykonania	0. Dodanie nowego requestu w operacji AddWord 1. Uzupełnienie parametru wordPLN (wartościami varchar 255) 2. Uzupełnienie parametru wordEng (wartościami varchar 255) 3. Wywołanie zapytania
Oczekiwany rezultat	Odmowa dodania kontentu do słownika, zwrócenie komunikatu "Ups, to już było"
Warunki końcowe	Kontent niedodany do słownika

ID	3.0
Tytuł	Błąd dodania kontentu do słownika
Środowisko	SoapUI
Warunek wstępny	Uruchomiona aplikacja (słownik)
Kroki do wykonania	0. Dodanie nowego requestu w operacji AddWord 1. Uzupełnienie parametru wordPLN (wartościami innymi niż varchar 255) 2. Uzupełnienie parametru wordEng (wartościami innymi niż varchar 255) 3. Wywołanie zapytania
Oczekiwany rezultat	Błąd dodania kontentu do słownika, zwrócenie komunikatu "could not execute statement"
Warunki końcowe	Kontent niedodany do słownika

ID	4.0
-----------	------------

Tytuł	Pobranie tłumaczenia słowa polskiego na angielskie
Środowisko	SoapUI
Warunek wstępny	Uruchomiona aplikacja (słownik), istnienie co najmniej jednej pary słów, gdzie wartość wordPLN jest taka sama jak wprowadzana przez nas wartość wordPLN
Kroki do wykonania	0. Dodanie nowego requestu w operacji TranslatePlnToEng 1. Uzupełnienie parametru wordPLN 2. Wywołanie zapytania
Oczekiwany rezultat	Zwrócenie wszystkich angielskich słów przypisanych do zadanego polskiego słowa
Warunki końcowe	Zwrócony rezultat (pobranie tłumaczenia) zgodny z bazą danych

ID	5.0
Tytuł	Błąd tłumaczenia słowa polskiego na angielskie
Środowisko	SoapUI
Warunek wstępny	Uruchomiona aplikacja (słownik), brak pary słów, w której wartość wordPLN jest taka sama jak wprowadzana przez nas wartość wordPLN
Kroki do wykonania	0. Dodanie nowego requestu w operacji TranslatePlnToEng 1. Uzupełnienie parametru wordPLN 2. Wywołanie zapytania
Oczekiwany rezultat	Błąd zwrócenie tłumaczenia, zwrócenie komunikatu "Nie znaleziono tłumaczenia"
Warunki końcowe	W bazie danych dla zadanego polskiego słowa nie znaleziono angielskiego odpowiednika

ID	6.0
Tytuł	Pobranie tłumaczenia słowa angielskiego na polskie
Środowisko	SoapUI
Warunek wstępny	Uruchomiona aplikacja (słownik), istnienie co najmniej jednej pary słów, gdzie wartość wordEng jest taka sama jak wprowadzana przez nas wartość wordEng
Kroki do wykonania	0. Dodanie nowego requestu w operacji TranslateEngToPln 1. Uzupełnienie parametru wordEng 2. Wywołanie zapytania
Oczekiwany rezultat	Zwrócenie wszystkich polskich słów przypisanych do zadanego angielskiego słowa
Warunki końcowe	Zwrócony rezultat (pobranie tłumaczenia) zgodny z bazą danych

ID	7.0
Tytuł	Błąd tłumaczenia słowa angielskiego na polskie
Środowisko	SoapUI
Warunek wstępny	Uruchomiona aplikacja (słownik), brak pary słów, w której wartość wordEng jest taka sama jak wprowadzana przez nas wartość wordEng
Kroki do wykonania	0. Dodanie nowego requestu w operacji TranslateEngToPln 1. Uzupełnienie parametru wordEng 2. Wywołanie zapytania
Oczekiwany rezultat	Błąd zwrócenia tłumaczenia, zwrócenie komunikatu "Nie znaleziono tłumaczenia"
Warunki końcowe	W bazie danych dla zadanego angielskiego słowa nie znaleziono polskiego odpowiednika

Zadanie_3_QA

Rozwiązania z tego zestawu zadań zostały dołączone w załączniku maila.