Test Plan MarKol

1. **WPROWADZENIE**

1.1 Cel
Celem projektu jest aplikacja e-commerce.

1.2 Przegld projektu
Projekt zakada dziaajc aplikacj internetow e-commerce. Zainteresowany moe zakupi wybrany produkt poprzez jego umieszczenie w koszyku, edytowanie koszyka oraz moliwo dokonania finalizacji transakcji.

1.3 Zespó
zespó QA- MT, AT, TM

zespó DEV: FE, BE. PO, SM, PM

Zarzd firmy zlecajcej projekt

2. **STRATEGIA TESTÓW**

2.1 Cel testu

Przetestowanie wymaga biznesowych aplikacji, dodanie do koszyka, edycja koszyka, finalizacja zakupu

Przetestowanie funkcjonalne i niefunkcjonalne aplikacji

Retesty

Raport z defektów

2.2 Zaoenia testowania

Aplikacja dostosowana do trybu deskop oraz mobile, zgodnie z zaoeniami biznesu

Dostp do narzdzi i sprztu potrzebnego do przeprowadzenia testów,

2.3 Poziomy i rodzaje testów

Testy jednostkowe - testy przeprowadzane przez Developerów, testy automatyczne (SeleniumWebdriver)

Testy integracyjne - urzdzenia testowe + licencja Browserstack, Postman, testy niefunkcjonalne - wydajnoci (GATLINK)

Testy systemowe - rodowisko testowe zblione do produkcyjnego, uytecznoci, regresji, funkcjonalnoci

Testy akceptacyjne - formalne potwierdzenie wykonania oprogramowania odpowiedniej jakoci

2.3.1 Testy jednostkowe

Cel: sprawdzenie kodu, dziaania funkcji aplikacji dodanie do koszyka, edycja koszyka, finalizacja zakupu

Zakres: modu

Testerzy: zespó developerów

Metoda: testy automatyczne SeleniumWebdriver

Czas: cay okres projektu

2.3.2 Testy integracyjne i systemowe

Cel:

- zmniejszenie ryzyka,
- zapobieganie przedostawaniu si defektów na wysze poziomy testowania,

Zakres: interfejs uytkownika UI

Testerzy: QA

Metoda: testy manualne,

Czas: po integracji moduów

2.3.3 Testy akceptacyjne

Cel: zgodno z wymaganiami projektowymi- dodanie do koszyka, edycja koszyka, finalizacja zakupu

Zakres: aplikacja e-commerce - dodanie do koszyka, edycja koszyka, finalizacja zakupu

Testerzy: PO, Zarzd firmy zlecajcej projekt

Metoda: testy manualne

Czas: przed wdroeniem na produkcje

2.3.4 Testy exploracyjne

Cel: zapoznanie z aplikacj - dodanie do koszyka, edycja koszyka, finalizacja zakupu

Zakres: gotowa aplikacja

Testerzy: zespó testesrki

Metoda: testy manualne

Czas: produkt gotowy

2.3.5 Testy funkcjonalne

Cel: przetestowanie funkcji,

Zakres: dodanie do koszyka, edycja koszyka, finalizacja zakupu

Testerzy: zespó testerski

Metoda: testy automatyczne, testy manualne

Czas: gdy funkcje s gotowe

2.3.6 Testy niefunkcjonalne

Cel: test wydajnociowy aplikacji (niefunkcjonalne)

Zakres: prdko przenaszalnoci

Testerzy: zespó testerski

Metoda: testy manualne, Jmeter

Czas: gdy funkcja jest gotowa

2.3.7 Automatyczne testy regresyjne

Cel: testy regresji

Zakres: funkcja dodania do koszyka

Testerzy: zespó testerski

Metoda: Selenium Webdriver

Czas: gdy aplikacja jest gotowa

2.4 Wyniki testów

Test Plan

Raport defektów

Przypadki testowe

Podsumowanie testów

2.5 Test effort estimation

Dziaania w zakresie zapewnienia jakoci	Czas testów (MD)

Analiza wymaga biznesowych	3MD
Opracowanie Plan Test	3MD
Opracowanie przypadków testowych	6MD
Wykonanie przypadków testowych	8MD
Opracowanie raportu defektów	3MD
Retesty	3MD
Testy regresyjne	3MD
Raport z podsumowania testów	3MD

3. **STRATEGIA REALIZACJI**

3.1 Kryteria wejcia i wyjcia

Kryteria wejcia:

aplikacja gotowa do testów

dane testowe dostpne

rodowisko dostpne

Kryteria wyjcia:

wszystkie zaoone przypadki testowe zostay wykonane

opracowany raport kocowy

testy akceptacyjne zakoczone sukcesem

3.3 Walidacja i zarzdzanie defektami

stworzenie raportu defektu

zaoenie buga

przekazanie buga do programisty

debugowanie defektu

naprawienie bdu przez programist

wykonanie retestu

gdy bd wystpuje to wraca do debugowania, gdy bd naprawiony = zamkniecie zgoszenia

3.4. Metryka testów

Metryka	Formua
Procent pozytywnych przypadków testowych	(liczba pozytywnych testów / cakowita liczba wykonanych testów) X 100
Procentowy udzia bdów krytycznych	(bdy krytyczne / cakowita liczba zgoszonych bdów) X 100
Efektywno projektowania testów	Liczba zaprojektowanych testów /Cakowity czas

4. **PROCES ZARZDZANIA TESTAMI**

4.1 Narzdzie do zarzdzania testami
Do zarzdzania testami wykorzystywana bdzie JIRA, defekty bd zgaszane jako BUG w JIRA

4.2 Proces projektowania testów
Do zarzdzania przypadkami testowymi bdzie Zephyr, TestRial, Browerstack

4.3 Proces realizacji testów
Wedug przygotowanych przypadków testowych przeprowadzane bd test run'y

4.4 Ryzyko zwizane z testami i czynniki agodzce

Ryzyko	Prawdopodobiestwo	Wpyw	Plan agodzenia skutków
Róna kultura obyczajowa zespou	wysokie	niski	impreza integracyjna przed projektem
Róny jzyk komunikacji midzyludzkiej	wysokie		ustalenie wspólnego jzyka komunikacyjnego w projekcie wraz z komunikatorem
Duy nacisk na procedury strona DE		niski	
Rónice pomidzy dev			Wypracowanie pola dziaania zespou
front a BE	wysokie		Wspólne wprowadzenie do wymaga biznesu
Innowacyjna architektura tworzenia oprogramowania		redni	
Brak znajomoci wymaga	wysokie		Ustalenie architektury przed rozpoczciem projektu, consulting, badanie kodu ródowego ju istniejcego
		redni	Zaznajomienie z wymaganiami, doprecyzowanie braków
	wysokie		
		redni	
	rednie		
		wysokie	

4.5 Odpowiedzialno zespoów

4.5.1 Zespó QA

opracowanie Plan Testu

- analiza wymaga biznesowych
- przygotowanie przypadków testowych
- wykonanie testów
- podsumowanie
- przygotowanie raportu
- retesty

4.5.2 Zespó developerski

- pisanie kodu
- naprawa bdów
- wykonanie testów jednostkowych
- przegld kodu

5. RODOWISKO TESTOWE

Testy bd wykonywane z dostpem do urzdze testowych + **licencja Browserstack**

Mobilne Android i iOS,

Przegldarki: Chrome, Mozilla, Opera wersja aktualna + 1 wersja wczeniejsza

6. NARZDZIA TESTOWE

Obszar	Narzdzia
Zarzdzanie dokumentacj	JIRA, Confluence
Zarzdzanie projektem	JIRA
Zarzdzanie przypadkami testowymi	TestRial, xRial,
Testowanie API	Postman
Wykonywanie testów	Selenium IDE, Browsertack