

Test plan

1. INTRODUCTION

1.1 Purpose

Przygotowanie procesu testowego dla funkcjonalności "Lista produktów".

1.2 Project overview

Przedmiotem testów jest aplikacja, która ma na celu sprzedaż produktów odzieżowych dla piesków. Główny akcent będzie położony na funkcjonalność "Lista produktów" służącą do przeglądania produktów w ramach kategorii.

1.3 Audience

Odbiorcą dokumentu jest Product Owner oraz zespół developerski.

2. TEST STRATEGY

2.1 Tests goal

Celem testów jest zapewnienie jakości w obszarze przedmiotu testów oraz zaspokojenie wymagań funkcjonalnych i нефункциональных określonych w dokumentacji biznesowej.

2.2 Test Assumptions

1. Dokumentacja biznesowa jest dostępna pod kątem testów statycznych.
2. Testy dynamiczne rozpoczną się w momencie dostarczenia gotowej funkcjonalności.
3. Środowisko testowe jest dostępne i sprawne.
4. Zespół QA ma dostęp do zasobów projektu.
5. Zespół QA ma kompetencje do prowadzenia procesu testowego.

2.3 Levels and types of testing

2.3.1 Unit tests

Purpose: pokrycie testami pojedynczych modułów, klas, metod aplikacji;

Scope: funkcjonalność "Lista produktów";

Testers: developerzy;

Method: test automatyczny pisany z poziomu kodu;

Timing: na koniec implementacji danej funkcjonalności;

2.3.2 Integration and system tests

Purpose: sprawdzenie integracji pomiędzy modułami;

Scope: funkcjonalność "Lista produktów";

Testers: zespół QA;

Method: manualne testy funkcjonalne oraz automatyczne testy нефункциональные;

Timing: testy jednostkowe są napisane; wersja aplikacji przekazana do testów;

2.3.3 Acceptance tests

Purpose: potwierdzenie czy wyprodukowana aplikacja spełnia założenia biznesowe;

Scope: funkcjonalność "Lista produktów";

Testers: Klient / Klient + QA / użytkownik w ramach testów beta / testy produkcyjne alfa;

Method: manualne testy funkcjonalne;

Timing: po testach systemowych;

2.3.4 Exploratory tests

Purpose: poznanie aplikacji w obszarze kluczowych funkcjonalności;

Scope: funkcjonalność "Lista produktów" i szerzej: ścieżka zakupowa;

Testers: zespół QA;

Method: manualne testy funkcjonalne;

Timing: po dołączeniu do projektu;

2.3.5 Functional tests

Purpose: przetestowanie aplikacji w aspekcie funkcjonalnym;

Scope: funkcjonalność "Lista produktów";

Testers: zespół QA;

Method: manualne testy funkcjonalne;

Timing: 1. po wdrożeniu danej funkcjonalności 2. w ramach testów regresji 3. w ramach smoke testów;

2.3.6 Non-functional tests

Purpose: zapewnienie jakości w obszarze нефункциональных характеристик системы, таких как performance, seo, bezpieczeństwo, responsywność;

Scope: funkcjonalność "Lista produktów";

Testers: zespół QA;

Method: testy z wykorzystaniem narzędzi;

Timing: w ramach testów danej funkcjonalności w trakcie sprintu;

2.3.7 Automated Regression Testing

Purpose: 1. pokrycie testów regresji w obszarze powtarzalnych scenariuszy testowych;

Scope: funkcjonalność "Lista produktów";

Testers: zespół QA;

Method: testy automatyczne;

Timing: po zakończeniu testów manualnych;

2.4 Test deliverables

1. Test plan.
2. Przypadki testowe.
3. Raporty defektów.
4. Raport podsumowujący z testów.
5. Skrypty z testów automatycznych.
6. Zapytania z API.

2.5 Test effort estimation

QA activity	Test effort (SP)
Opracowanie test planu	5 SP
Opracowanie przypadków testowych	5 SP
Przygotowanie raportu końcowego	5 SP
Opracowanie skryptów testów automatycznych	8 SP

3. EXECUTION STRATEGY

3.1 Entry and exit criteria

Entry criteria:

1. wersja aplikacji gotowa do testów dynamicznych;
2. napisane przypadki testowe
3. dostępne środowisko testowe

Exit criteria:

1. wszystkie zgłoszone defekty są zamknięte;
2. wszystkie przypadki testowe zostały wykonane

3.2 Test cycles

3.3 Validation and defect management

1. Zgłoszenie defektu przez zespół QA.
2. Analiza zgłoszenia defektu przez zespół dev/qa/PO.
3. Naprawianie przez zespół dev.
4. Retesty przez zespół QA.
5. Zamknięcie defektu lub ->
6. Idź do 3-4.

3.4. Test Metrics

Metric	Formula
procent wykonanych przypadków testowych	$\text{liczba wykonanych przypadków} / \text{liczba wszystkich przypadków} * 100$
procent zaliczonych przypadków testowych	$\text{liczba zaliczonych przypadków testowych} / \text{liczba wykonanych przypadków testowych} * 100$
procent defektów w skali wykonanych przypadków testowych	$\text{liczba defektów zgłoszonych} / \text{liczba przypadków testowych} * 100$

4. TEST MANAGEMENT PROCESS

4.1 Test management tool

Proces zarządzania procesem testowym będzie oparty o Jirę i mechanizm ticketowania zgłoszeń.

4.2 Test design process

Do pisania przypadków testowych będzie wykorzystane narzędzie TestRail.

Za pisanie przypadków testowych odpowiedzialny będzie tester z zespołu QA.

Przypadki testowe podlegać będą regularnych przeglądom w ramach test review.

Przypadki testowe będą sprawdzane pod kątem utrzymania/pielęgnowności.

4.3 Test execution process

Wykonanie testów będzie miało charakter manualny.

Testy wykonywane będą w ramach test run'ów, które grupują paczki przypadków testowych w podziale na funkcjonalności/priorytety.

4.4 Test risks and mitigation factors

Risk	Probability	Impact	Mitigation plan
------	-------------	--------	-----------------

Aplikacja jest skomplikowana oraz dojrzała			Dokładne zapoznanie się z dokumentacją Wykonanie testów eksploracyjnych Rozmowa z PO o logice biznesowej
Niedoświadczony zespół			Ścisła współpraca w ramach zespołu (junior vs senior) Systematyczne przeglądy
Napięty harmonogram			Umiejętne zarządzanie priorytetami Przewidzieć więcej czasu na testy Wzmocnić/rozszerzyć zespół Efektywny podział pracy
Braki warsztatowe/narzędziowe			Wewnętrzne/zewnętrzne szkolenia Ustalenie budżetu na rozwój/narzędzia Transfer wiedzy nt. narzędzi
Słaba jakość dokumentacji			Komunikacja z biznesem Stosować bardziej szczegółowy sposób określania wymagań biznesowych
Ciągłe zmiany w dokumentacji			Komunikacja z biznesem Większa koncentracja na analizie wymagań
Trudności w komunikacji w ramach zespołu			Integracja zespołowa Warsztaty miękkie z komunikacji
Wrażliwi developerzy 😊			Czekoladki Memy Zakaz używania "u mnie działa"
Trudny klient			Komunikacja z biznesem Integracja Zrozumienie potrzeb, oczekiwań, motywacji Uświadomienie klienta o obopólnych korzyściach z harmonijnej współpracy Przydzielenie osoby do kontaktu z biznesem Pokazanie "strat" w projekcie (np. obsunięcie w harmonogramie)

4.5 Test responsibility

4.5.1 QA team

- testy akceptacyjne
- testy integracyjne
- opracowywanie przypadków testowych
- tworzenie raportu defektu

- e) testowanie statyczne
- f) tworzenie test planu
- e) raport końcowy/podsumowujący
- f) pisanie testów automatycznych

4.5.2 Development team

- a) pisanie testów jednostkowych
- b) debuggowanie defektów
- c) wsparcia techniczne dla zespołu QA
- d) code review testów automatycznych
- e) analiza dokumentacji

5. TEST ENVIRONMENT. COMPATIBILITY CHART

Testy będą realizowane na środowisku produkcyjnym.

Aplikacja będzie testowana na systemie operacyjnym zainstalowanym na komputerze danego testera.

Tester zapewni pokrycie testami na co najmniej 3 przeglądarkach internetowych (np. Chrome, FF, Opera, Edge).

Tester zapewni pokrycie testami aplikacji w trybie responsywnym korzystając z trybu device mode w narzędziach developerskich przeglądarki.

6. TESTING TOOLS

Area	Tool
zarządzenie projektem tworzenie ticketów zgłoszenia defektów	Jira
dokumentacja testowa	Confluence
tworzenie przypadków testowych	TestRail
testowania responsywności	Devtools'y
SEO, performance, security	Lighthouse (devtools'y)
tworzenie skryptów automatycznych	Selenium intelliJ
tworzenie testów jednostkowych	intelliJ