

# Test plan TKS

## 1. INTRODUCTION

### 1.1 Purpose

Dostarczenie w pełni funkcjonalnej aplikacji e-commerce dla klienta biznesowego wykorzystując do tego test plan. Modu, który na którym będziemy się skupiać to moduł koszyka.

Aplikacja znajduje się pod adresem [www.komputronik.pl](http://www.komputronik.pl)

### 1.2 Project overview

Główne zadanie, które ma spełniać aplikacja:

- możliwość wyboru produktów z bazy dostępnych produktów
- filtrowanie i sortowanie produktów
- dodawanie i kasowanie produktów z koszyka w celu wyboru właściwych produktów
- finalizacja zakupów

### 1.3 Audience

Project Manager

Scrum Master

Product Owner

Dev FE

Dev BE

Test Manager

Testerzy manualni

Testerzy automatyczni

Zespoły będą podzielone na następujące grupy:

Zespół QA: 3 MT (manual tester, 2 junior, 1 mid), 1 AT (junior automation tester), 1 TM (Test Manager)

Zespół dev: 3 FE (frontend dev), 2 BE (backend dev), 1 PO (Product Owner), 1 SM (Scrum Master), 1 PM (Project Manager).

## 1. TEST STRATEGY

### 2.1 Tests goal

Ujawnienie usterek

Pokrycie wymagań testowych

Wykrywanie defektów i zapobieganie ich występowaniu

Retesty/Testy regresji

Sprawdzanie i zbadanie funkcjonalności

Sprawdzenie wartości brzegowych

Aplikacja dostępna do trybu desktop i mobile

Dostarczenie biznesowi informacji o stanie aplikacji i możliwych różnicach w wytycznych

Czytelna i jasno opisana dokumentacja

### 2.2 Test Assumptions założenia testowe

Dostępność dokumentacji biznesowej i jej wytycznych

Dostępność środowiska testowego (wersja aplikacji dla testerów)

Dostępność zespołów zaangażowanych w projekt

Zaowocowania testowe i ich wdrożenie poprzez odpowiednie techniki

Dostępność wszystkich narzędzi do testowania

Pokrycie wszystkich decyzji biznesowych przypadkami testowymi

## 2.3 Levels and types of testing

### 2.3.1 Unit tests

**Purpose:** sprawdzenie/pokrycie danej części/linii kodu źródłowego naszej aplikacji

**Scope:** kod źródłowy/poszczególne elementy gotowe do testowania

**Testers:** zespół deweloperów

**Method:** techniki białego skrzynki, pokrycie instrukcji, testy automatyczne

**Timing:** charakter ciągły

### 2.3.2 Integration and system tests

**Purpose:** pokrycie się aplikacji z wymaganiami biznesowymi, współgranie modułów

**Scope:** funkcjonalność front i back endów

**Testers:** testerzy

**Method:** testy manualne (technikami czarnoskrzynkowymi, retesty, testy regresji) testy automatyczne

**Timing:** po testach jednostkowych

### 2.3.3 Acceptance tests

**Purpose:** akceptacja aplikacji z warunkami biznesowymi

**Scope:** całość aplikacji lub jej większe gotowe elementy

**Testers:** klient końcowy wraz z PO

**Method:** testy manualne, eksploracyjne, BDD

**Timing:** gdy testy zostaną zakończone i aplikacja jest już gotowa

### 2.3.4 Exploratory tests

**Purpose:** poznanie aplikacji i jej funkcjonalności, znalezienie defektów

**Scope:** całość aplikacji

**Testers:** testerzy

**Method:** testy manualne czarnoskrzynkowe

**Timing:** wraz ze startem projektu oraz na etapach, kiedy funkcje i produkty są wdrażane do aplikacji

### 2.3.5 Functional tests

**Purpose:** sprawdzenie poprawności działania danej funkcjonalności pod względem wymagań i funkcji

**Scope:** dana funkcjonalność

**Testers:** testerzy

**Method:** manualne testowanie, testy czarnoskrzynkowe, testy automatyczne

**Timing:** na etapie testów integracyjnych, systemowych, akceptacyjnych

### 2.3.6 Non-functional tests

**Purpose:** całościowe działanie aplikacji

**Scope:** wydajność, obciążenie, przeciążenie, użyteczność, pielęgnacja, SEO, bezpieczeństwo, pielęgnacja aktualizacji

**Testers:** testerzy manualni i automatyczni

**Method:** testy manualne użyteczności, pokrycie przypadków, testy automatyczne

**Timing:** na każdym etapie testowania zgodnie z zasadami wczesnego testowania

### 2.3.7 Automated Regression Testing

**Purpose:** sprawdzenie testów regresji, przpieszenie pracy testerów manualnych

**Scope:** integracja modułów, systemów, funkcjonalności

**Testers:** testerzy automatyczni

**Method:** testy automatyczne z poziomu kodu, z użyciem narzędzi

**Timing:** po zmianach w kodzie źródłowym, po integracji modułów

### 2.4 Test deliverables

- plan testów
- przypadki testowe
- raporty defektów
- raporty testów, raporty podsumowujące

### 2.5 Test effort estimation

QA activity	Test effort (MD)
<ul style="list-style-type: none"><li>• Stworzenie test planu</li><li>• Opracowanie wymagań</li><li>• Scenariusze testowe, przypadki testowe</li><li>• Raportowanie defektów</li><li>• Wykonanie testów regresji</li><li>• Raportowanie z testów plus raport podsumowujący</li></ul>	<ul style="list-style-type: none"><li>• 2MD</li><li>• 2MD</li><li>• 8/10MD</li><li>• 2/3MD</li><li>• 2MD</li><li>• 3MD</li></ul>

## 1. EXECUTION STRATEGY

### 3.1 Entry and exit criteria

Entry criteria:

- kompletna dokumentacja
- narzędzia pracy są dostępne
- gotowe przypadki testowe
- gotowa do testowania aplikacja
- działające środowisko testowe

Exit criteria:

- pokrycie instrukcji
- wykonane wszystkie przypadki testowe
- zgłoszone defekty o wysokiej i średniej wadliwosci zostały naprawione
- testy regresji zostały wykonane i naprawione
- brak błędów funkcjonalnych
- brak błędów krytycznych
- testy akceptacyjne wykonane i zaakceptowane
- raport końcowy
- protokół odbioru

### 3.2 Test cycles

### 3.3 Validation and defect management

(co się dzieje z defektem jak go znajdujemy zarządzanie cyklem defektu)

Wykrycie defektu podczas testowania.

Określenie defektu skali: niska/średnia/wysoka/bardzo wysoka

W przypadku defektu niskiej rangi defekt odrzucony, nie do naprawy.

W przypadku defektów innej skali ocenienie ich czy nadają się do testowania.

liczba wykonanych przypadków testowych / liczba wszystkich przypadków testowych \* 100. Jeśli tak to naprawa, ponowne testowanie i jeśli test ma wynik pozytywny to zamknięcie testu.

Jeeli tak to naprawa, ponowne testowanie i jeeli test ma wynik negatywny to ponowne dziaania naprawcze i cykl. Jeeli ponowny cykl przyjdzie z wynikiem negatywny to dyspozycja odroczenia.

Wedug standardu IEEE 1044.

3.4. Test Metrics

Metric	Formula
procent wszystkich przypadków testowych	liczba wykonanych przypadków testowych / liczba wszystkich przypadków testowych * 100
procent przypadków testowych zaliczonych	liczba zaliczonych przypadków testowych / liczba wykonanych przypadków testowych * 100
procent defektów w caociowej liczbie przypadków testowych	liczba zgoszonych defektów / liczba przypadków testowych * 100

1. TEST MANAGEMENT PROCESS

4.1 Test management tool

Jira/Confluence narzdzie do zarzdzania projektem i defektami.

TestRail do zarzdzania testami i raportem kocowym.

4.2 Test design process

Przypadki testowe na podstawie dostarczonych wymaga bd realizowane w TestRail.

Przeglady koleeskie.

Testy eksploracyjne.

4.3 Test execution process

Praca na testrunach w metodologi Scrumowej.

4.4 Test risks and mitigation factors

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

Sprztowe	niskie	wysokie	Stworzenie planu niezbędnego hardware i software z wyprzedzeniem czasowym. Zakup niezbędnych licencji .
Czasowe	rednie/wysokie		
Personnel		wysoki	Przesunicie odbioru produktu lub zwikszenie ilosci osób nad nim pracujcych w celu przypieszenia czasu pracy.
Requirements	rednie	wysoki	Praca w niedowiadczoneym zespole. Moliwe zmiany personalne lub odpyw pracowników. Dziaania agodcze z wikszenie zespou lub kolejnych testerów.
		wysoki	

#### 4.5 Test responsibility

Test Manager

##### 4.5.1 QA team

Test Manager

Testerzy manualni

Testerzy automatyczni

##### 4.5.2 Development team

Project Manager

Scrum Master

Product Owner

Dev FE

Dev BE

5.1 TEST ENVIRONMENT

- systemy Windows, Linux oraz MacOS
- urządzenia mobilne z oprogramowaniem Android oraz iOS w najnowszej aktualizacji i 5 wersji wstecz
- najnowsze wersje najpopularniejszych przeglądarek plus 3 wersje wstecz

6.1 TESTING TOOLS

Area	Tool
<ul style="list-style-type: none"><li>• Zarządzanie przypadkami testowymi</li><li>• Zarządzanie defektami</li><li>• Zarządzanie dokumentacją</li><li>• Testowanie API</li><li>• Testy automatyczne</li><li>• Komunikacja wew</li></ul>	<ul style="list-style-type: none"><li>• TestRail, Jira</li><li>• Jira Confluence</li><li>• Confluence</li><li>• Postman</li><li>• Selenium IDE</li><li>• Slack</li></ul>