

# Program kursu Tester Manualny



## MODUŁ 0: PREWORK

### Projekt IT

- Zarządzanie projektami IT
- Role w projekcie IT.

### Bazy danych - wprowadzenie

- Wprowadzenie do baz danych
- dane konfiguracyjne,
- pozyskiwanie informacji o bazie danych,
- Pobieranie danych z tabeli.

### Aplikacje internetowe - wprowadzenie

- Struktura aplikacji internetowych,
- technologie web,
- przeglądarki internetowe.

### Słownik testera

- Słownik testera.

## MODUŁ 1: MODUŁ PRAKTYCZNY

### DZIEŃ 1 – Wprowadzenie do testów oprogramowania

- Co to jest testowanie?
- 7 zasad testowania,
- czynności i zadania testowe,
- poziomy testów,
- typy testów,
- techniki projektowania testów.

### DZIEŃ 2 – Agile i przypadek testowy

- Agile – historia, manifest programowania zwinnego, Scrum,
- planowanie tygodnia – omówienie i spisanie zadań, nadanie im priorytetów, Sprint Planning,
- praktyczne podejście do przypadku testowego – czym jest, z jakich podstawowych elementów się składa oraz jak powinien wyglądać w praktyce.

### DZIEŃ 3 – przypadek testowy warsztat

- Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś,
- Jira + TestFlo – omówienie funkcjonalności narzędzia: tworzenie podstawowych zgłoszeń typu: zadanie, historia, epika, tworzenie sprintu i dodawanie zadań do sprintu, komunikacja w Jira i komentowanie zgłoszeń, tworzenie prostych,
- tworzenie przypadków testowych na podstawie dokumentacji oraz dostępnej aplikacji.

### DZIEŃ 4 – bazy danych

- Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś,
- SQL: powtórka prework, pobieranie danych (Select), łączenie tabel (JOIN), tworzenie baz danych, modyfikacja baz danych.

### DZIEŃ 5 – zgłaszanie błędów

- Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś,
- jak zgłaszać błędy? – praktyczne wskazówki, jak zgłaszać błędy, aby były one czytelne dla deweloperów,
- wyszukiwanie błędów w dostępnej aplikacji,
- DevTools – omówienie działania narzędzi programistycznych w przeglądarkach.

## DZIEŃ 6 – Web Services i API

- Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś,
- Web Services: definicja, przykłady zastosowań, zalety i wady,
- XML i JSON: znaczniki, struktura, poprawność składniowa,
- SOAP: struktura SOAP, WSDL, obsługa narzędzia Soap UI,
- HTTP: protokół internetowy, HTTP request, metody HTTP, HTTP response, response code,
- REST API: zasady, metody REST (HTTP), response code REST, obsługa narzędzia POSTMAN.

## DZIEŃ 7 – retesty

- Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś,
- retest a regresja: jak wykonywać retesty, jak przygotować się do regresji,
- retesty błędów znalezionych podczas zajęć ze zgłaszania błędów.

## DZIEŃ 8 – testowanie mobilne i raportowanie

- Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś,
- aplikacje mobilne: co to jest aplikacja mobilna, typy aplikacji mobilnych, mobilne systemy operacyjne,
- testy aplikacji mobilnych: jak zacząć, fragmentacja rynku, sieć, pakiety, użyteczność, Emulatory iOS i Android,
- raportowanie – po co raportujemy, czym są raporty w testach i do czego je wykorzystujemy w projekcie, przedstawienie najważniejszych elementów raportu, omówienie przykładowego raportu z testów,
- tworzenie raportów – przygotowanie raportów z prac z wykorzystaniem danych zgromadzonych w narzędziu Jira,
- retrospektywa Agile – podsumowanie kursu i omówienie uwag, które mogą prowadzić do udoskonalenia produktu.

## EGZAMIN

- Egzamin końcowy z kursu Tester Manualny.

## MODUŁ 2: AI DLA TESTERÓW<sup>1</sup>

### AI dla testerów

- Wstęp do ChatGPT,
- Prompt Engineering,
- Testowanie oprogramowania przy użyciu ChatGPT,
- ChatGPT i PostgreSQL.

## MODUŁ 3: ISTQB FOUNDATION LEVEL<sup>2</sup>

### Podstawy testowania

- Co to jest testowanie,
- dlaczego testowanie jest niezbędne,
- zasady testowania,
- czynności testowe,
- testalia i role związane z testami,
- niezbędne umiejętności i dobre praktyki w dziedzinie testowania.

### Testowanie w cyklu życia oprogramowania

- testowanie w kontekście cyklu wytwarzania oprogramowania,
- testowanie jako czynnik określający sposób wytwarzania oprogramowania,
- poziomy testów,
- typy testów,
- testowanie związane ze zmianami i pielęgnacyjne.

### Testowanie statyczne

- podstawy testowania statycznego,
- informacje zwrotne i proces przeglądu.

<sup>1</sup>self-learning, materiały dodatkowe, nieobowiązkowe, dostępne we wszystkich pakietach

<sup>2</sup>self-learning, materiały dodatkowe, nieobowiązkowe, dostępne w pakietach: Profesjonalny, Premium i Pewna Praca

## Techniki projektowania testów

- Ogólna charakterystyka technik testowania,
- podział na klasy równoważności,
- analiza wartości brzegowych,
- testowanie w oparciu o tablicę decyzyjną,
- testowanie przejść pomiędzy stanami,
- białoskrzynkowe techniki testowania,
- techniki testowania oparte na doświadczeniu,
- podejścia do testowania oparte na współpracy.

## Zarządzanie testowaniem

- Planowanie testów,
- techniki szacowania,
- ustalanie priorytetów przypadków testowych,
- piramida testów,
- kwadranty testowe,
- zarządzanie ryzykiem,
- monitorowanie testów,
- nadzór nad testami i ukończenie testów,
- zarządzanie konfiguracją,
- zarządzanie defektami.

## Narzędzia testowe

- narzędzia wspomagające testowanie,
- korzyści i ryzyka związane z automatyzacją testów.

## MODUŁ 4: TEST LAB<sup>3</sup>

### Linux

- Podstawy Linuxa, najpopularniejsze wersje Linuxa,
- obsługa emulatora terminala oraz podstawy Vi,
- podręczniki systemowe,
- system plików: struktura katalogów, poruszanie się po katalogach, prawa dostępu do plików, komendy: ls, mkdir, cd, pwd, cp, mv, rm, cat, less, head, tail, wc, touch,
- podstawowe komendy: grep, find, pipe (|). przekierowanie,
- użytkownicy: typy, tworzenie i zarządzanie,
- manager pakietów (Ubuntu + macOS),
- zaawansowane tematy: procesy, zdalny terminal (screen), harmonogram zadań, symlinki, hardlinki, zmienne systemowe.

### HTML i CSS

- Znaczniki, struktura dokumentu,
- formatowanie tekstu,
- hipertącza, obrazki, tabele,
- CSS – kilka podstawowych stylów.

### Narzędzia testowe

- Nauka UML i BPMN,
- testy aplikacji FinTech,
- testy aplikacji CRM,
- testy aplikacji HR,
- automatyzacja testów Selenium IDE (record&play),
- praca z dokumentacją techniczną.