

## PROJEKT DLA PRZYSZŁYCH TESTERÓW OPROGRAMOWANIA

Projekt szkoleniowo-stażowy  
Junior tester oprogramowania  
**manualno-automatyzujący**



**Quality Island Sp. z o.o.**

---

Concept Tower, Grzybowska 87, Warszawa  
Telefon: +48 531 119 616  
E-mail: kontakt@qualityisland.pl




## Quality Island to MY!

Testowanie oprogramowania to nasza pasja. **Jesteśmy niewielką firmą z wielkimi możliwościami!** Stawiamy na jakość oraz indywidualne podejście do klientów. Realizujemy projekty testowe, udostępniamy kompetencje i zasoby, budujemy zespoły testerskie, usprawniamy procesy testowania, prowadzimy szkolenia i kursy oraz świadczymy usługi audytowe i konsultingowe, opierając się na doświadczeniu ze współpracy z największymi firmami.

Nasz zespół składa się z testerów automatyzujących, testerów manualnych, architektów IT, analityków testowych i kierowników testów. **Obsługujemy cały proces testowy**, od dokumentacji po testy manualne, automatyzację i niezależne testy akceptacyjne.

**Oferujemy szkolenia z zakresu testowania oprogramowania dla osób indywidualnych, firm oraz instytucji.**

Quality Island to grupa doświadczonych profesjonalistów różnych specjalności w dziedzinie zapewniania jakości oprogramowania, realizujących projekty zadaniowo dla różnych klientów. **Jesteśmy zweryfikowaną, wpisaną do Rejestru Instytucji Szkoleniowych firmą szkoleniową.** Nasze usługi są dostosowane do specyficznych potrzeb każdego klienta, co pozwala nam zapewnić najwyższą jakość. Dzięki naszemu doświadczeniu możemy skutecznie wspierać firmy w osiągnięciu ich celów biznesowych.



## Projekt szkoleniowo-stażowy Junior tester manualno-automatyzujący

Quality Island przedstawia innowacyjny program szkoleniowo-stażowy dla przyszłych junior **testerów manualnych i automatyzujących**. Program ten został stworzony z myślą o osobach, które pragną rozpocząć swoją karierę w testowaniu oprogramowania, łącząc intensywne szkolenie teoretyczne z praktycznym doświadczeniem, które umożliwia zdobycie kluczowych umiejętności w jednym z najszybciej rozwijających się obszarów IT.

Program obejmuje około 200 godzin zajęć i pracy prowadzonych zdalnie w trybie wieczorowo-weekendowym, co pozwala uczestnikom pogodzić pracę i naukę z dotychczasowymi obowiązkami. Szkolenie przygotowuje do pracy zarówno jako tester manualny, jak i automatyzujący, dzięki czemu uczestnicy zdobywają wszechstronną wiedzę potrzebną na współczesnym rynku pracy. Program oferuje również unikalną możliwość pracy nad **komercyjną aplikacją**, co pozwala uczestnikom zdobyć **realne doświadczenie projektowe**.

Co wyróżnia ten program, to wsparcie nie tylko w zdobywaniu umiejętności technicznych, ale także w procesie rekrutacji. Uczestnicy mogą liczyć na **profesjonalne doradztwo w tworzeniu CV** oraz **przygotowanie do rozmów kwalifikacyjnych**, co zwiększa ich szanse na znalezienie pracy w branży IT.

Dodatkowo, uczestnicy mają możliwość zdobycia **certyfikatów** powszechnie uznawanych przez pracodawców, co jest cennym elementem ich portfolio zawodowego.

Nad całym procesem czuwa osobiście doświadczony mentor, CEO firmy Quality Island **Tomasz Stelmach**, który dba o to, aby każdy uczestnik zdobył solidne podstawy zarówno w testowaniu manualnym, jak i automatyzującym. Tomasz zapewnia indywidualne wsparcie oraz doradztwo, pomagając uczestnikom rozwijać swoje umiejętności w praktycznym kontekście. Dodatkowo, uczestnicy mają dostęp do grupy wsparcia na platformie Discord, gdzie mogą wymieniać się doświadczeniami i zasięgać pomocy od innych kursantów.

# Ogólny plan projektu

## ZAKRES TEORETYCZNY I PRAKTYCZNY ISTQB

---

### Wprowadzenie do testowania oprogramowania

- Wprowadzenie do testowania oprogramowania: Czym jest testowanie oprogramowania?
- Testować czy nie? Znaczenie i konieczność testowania.
- Podstawowe pojęcia techniczne związane z testowaniem oprogramowania.
- Siedem zasad testowania: Kluczowe zasady, które każdy tester powinien znać.
- Zapewnianie jakości a testowanie: Różnice między QA a testowaniem.

### Proces i psychologia testowania

- Czynności i zadania testowe: Kluczowe kroki w procesie testowym.
- Różnice w sposobie myślenia testerów i programistów: Jakie są kluczowe różnice?
- Psychologia testowania: Jak podejście psychologiczne wpływa na skuteczność testów?
- Podstawowy proces testowy: Omówienie etapów procesu testowego.

### Modele wytwarzania oprogramowania i poziomy testów

- Modele wytwarzania oprogramowania: Przegląd popularnych modeli (np. Waterfall, Agile).
- Poziomy testów: Testy jednostkowe, integracyjne, systemowe i akceptacyjne.
- Typy testów: Przegląd różnych typów testów (funkcjonalne, нефункционалне, regresyjne, etc.).
- Testowanie funkcjonalne vs. нефункционалне: Cele i metody.

### Cykl życia oprogramowania i testy

- Testowanie pielęgnacyjne: Czym jest i kiedy jest stosowane?
- Cykl życia oprogramowania vs. testy: Jak testy wpasowują się w cykl życia projektu.
- Testowanie statyczne: Techniki i korzyści.
- Techniki statyczne vs. dynamiczne: Porównanie podejść.

### Proces przeglądu i techniki testowania

- Proces przeglądu: Etapy i znaczenie przeglądów.
- Typy przeglądów: Różne rodzaje przeglądów (np. formalne, nieformalne).
- Statyczne techniki testowania: Kiedy i jak stosować.
- Wprowadzenie do technik testowania: Podstawy testów białoskrzynkowych i czarnoskrzynkowych.

# Ogólny plan projektu

## Techniki testowania: Białoskrzynkowe i czarnoskrzynkowe

- Testy białoskrzynkowe: Przegląd technik i narzędzi.
- Testy czarnoskrzynkowe: Kluczowe metody i podejścia.
- Podział na klasy równoważności: Jak grupować dane testowe?
- Analiza wartości brzegowych: Jak testować wartości graniczne?

## Zaawansowane techniki testowania

- Testowanie w oparciu o tablicę decyzyjną: Kiedy i jak stosować?
- Testowanie przejść pomiędzy stanami: Testy w oparciu o model stanów.
- Testowanie oparte na przypadkach użycia: Jak testować na podstawie scenariuszy użycia?
- Testowanie i pokrycie instrukcji kodu oraz decyzji: Techniki zapewniające pełne pokrycie kodu.
- Eksploracja oprogramowania: Dynamiczne podejście do testowania.

## Zarządzanie testami i procesem testowym

- Organizacja testów: Struktura zespołu testowego.
- Zadania kierownika testów oraz testerów: Rola i odpowiedzialności.
- Planowanie i szacowanie testów: Jak efektywnie planować testy?
- Harmonogram testów: Tworzenie i zarządzanie harmonogramem testów.
- Zarządzanie defektami: Proces zgłaszania i śledzenia błędów.
- Miary testowe: Jak mierzyć skuteczność i efektywność testów?

## Strategie testów i zarządzanie ryzykiem

- Strategie testów i podejście do testowania: Jak opracować strategię testową?
- Techniki szacowania testów: Metody szacowania czasu i zasobów.
- Monitorowanie testów: Jak śledzić postępy testów?
- Raportowanie: Jak efektywnie raportować wyniki testów?
- Ryzyko w testowaniu: Identyfikacja i zarządzanie ryzykiem w procesie testowym.

## Narzędzia wspierające proces testowy

- Klasyfikacja narzędzi testowych: Przegląd dostępnych narzędzi.
- Mocne i słabsze strony automatyzacji testów: Kiedy warto automatyzować testy?
- Skuteczne korzystanie z narzędzi: Najlepsze praktyki.
- Czym kierować się przy wyborze narzędzi?: Kryteria wyboru odpowiednich narzędzi.

## WPROWADZENIE DO TESTOWANIA API POSTMAN

---

### Wprowadzenie do API

- Co to jest API?: Definicja i podstawowe pojęcia.
- Jak działa API i do czego służy?: Przykłady zastosowań API w praktyce.
- Rodzaje API: Przegląd różnych typów API (np. publiczne, prywatne, partnerskie).
- SOAP vs REST: Porównanie dwóch głównych architektur API.

### Rest API – podstawy

- Rest API – omówienie: Charakterystyka i zasady działania RESTful API.
- Metody Rest API: Przegląd podstawowych metod HTTP (GET, POST, PUT, DELETE, etc.).
- Endpoint – co to takiego?: Wyjaśnienie pojęcia endpointu.
- Żądania i odpowiedzi – omówienie: Struktura i elementy żądania oraz odpowiedzi HTTP.
- Kody odpowiedzi HTTP: Znaczenie i interpretacja kodów statusu HTTP.

### Dokumentacja API i testowanie

- Dokumentacja API: Jak czytać i tworzyć dokumentację API.
- Testy API – podstawy: Wprowadzenie do testowania API, cele i korzyści.
- Narzędzia do testów API: Przegląd popularnych narzędzi (Rest Assured, Postman, JMeter, Fiddler).
- W jaki sposób testować API: Metodyka i proces testowania API.
- Jakie błędy powinny wychwycić testy API: Typowe błędy i problemy wykrywane przez testy API.
- Korzyści ze stosowania testów API: Dlaczego warto testować API?
- Najlepsze praktyki testowania API: Kluczowe zasady i techniki.

### Wprowadzenie do Postman

- Postman – wprowadzenie do narzędzia: Podstawowe funkcje i interfejs.
- Omówienie najważniejszych elementów Postmana: Panele, opcje i konfiguracje.
- Kolekcje: Tworzenie i zarządzanie kolekcjami żądań.
- Tworzenie żądań: Jak konfigurować i wysłać żądania API.
- Odczytywanie odpowiedzi: Analiza odpowiedzi API w Postmanie.
- Autoryzacja / Tokeny: Jak obsługiwać autoryzację w API.
- Parametry żądań: Konfiguracja parametrów w zapytaniach API.
- Zmienne: Użycie zmiennych w Postmanie dla dynamicznych testów.

## Zaawansowane funkcje Postmana

- Postman i Newman: Wprowadzenie do narzędzia Newman do uruchamiania testów w CLI.
- Pierwszy test w Postman: Tworzenie prostego testu.
- Pierwszy zestaw testów: Organizacja i uruchamianie zestawów testów.
- Parametry zaawansowane: Zaawansowane opcje konfiguracji żądań i odpowiedzi.
- Automatyzacja testów – skrypty testowe: Tworzenie i uruchamianie skryptów testowych w Postmanie.
- Refactoring testów: Jak utrzymywać i optymalizować zestawy testów.
- Debugowanie testów: Techniki rozwiązywania problemów z testami API.
- Asercje i weryfikacje: Tworzenie asercji do weryfikacji odpowiedzi API.
- Uruchamianie skryptów testowych: Wykonywanie skryptów automatyzujących testy.

## Testy API w CI/CD

- Test API w CI: Integracja testów API w procesach Continuous Integration/Continuous Deployment.
- Uruchamianie poszczególnych testów: Zarządzanie i wykonywanie pojedynczych testów.
- Uruchamianie kolekcji testów: Automatyzacja całych zestawów testowych za pomocą Postmana i Newmana.

## AI SZTUCZNA INTELIGENCJA W TESTOWANIU

---

### Wprowadzenie do AI i testowania oprogramowania

- Znaczenie AI w innowacjach technologicznych i jej wpływ na branżę IT.
- Ewolucja testowania oprogramowania i miejsce AI w tym procesie.
- Przykłady zastosowania AI w testowaniu – od prostych do zaawansowanych przypadków.
- Korzyści z integracji AI w procesach testowych.
- Wyzwania i rozwiązania w implementacji AI w testowaniu oprogramowania.
- Przegląd głównych algorytmów i modeli AI używanych w IT
- Bezpieczeństwo danych
- Prompty

### Narzędzia AI w testowaniu oprogramowania

- Omówienie narzędzi AI w kontekście ciągłej integracji i ciągłego dostarczania (CI/CD).
- Porównanie narzędzi AI do tradycyjnych metod testowania.
- Przykłady udanej integracji narzędzi AI w istniejące środowiska testowe.
- Omówienie przyszłościowych narzędzi AI w testowaniu i ich potencjału.
- Jak wybierać odpowiednie narzędzia AI do konkretnych potrzeb testowych.
- ChatGPT 4o
- Bard (Gemini)
- CoPilot

### AI w testach manualnych

- Jak AI przekształca podejście do testów funkcjonalnych.
- AI w analizie wymagań i identyfikacji kluczowych obszarów testowych.
- Generowanie danych
- Tworzenie przypadków testowych
- Tworzenie przypadków testowych – ćwiczenie
- Wykorzystanie AI to stworzenia US
- Dokumentacja
- Test strategie / Test Polityki
- Test plany
- Opisy, dokumentacja



# Ogólny plan projektu

## Automatyzacja testów z użyciem AI

- Integracja AI z istniejącymi frameworkami do automatyzacji testów.
- Przypadki użycia AI w automatyzacji – od prostych skryptów do złożonych scenariuszy.
- Optymalizacja procesów testowych za pomocą AI.
- Mierzenie skuteczności automatyzacji testów z wykorzystaniem AI.
- Trendy i przyszłość automatyzacji testów z AI.
- Testy automatyczne API
- Postman & AI w Chat GPT
- Testy automatyczne – frontend
- Testy automatyczne generowane w Cypressie
- Tabnine
- Github Copilot
- Automatyzacja testów wydajnościowych
- Testy wydajnościowe z K6
- Tworzenie skryptów z wykorzystaniem Chat GPT
- Wsparcie w procesach i na środowiskach CI/CD

## Własne rozwiązania wspierające

- DeadLinkChecker
- Skrypty do weryfikowania statusu odpowiedzi

## Przyszłość testowania oprogramowania z AI

- Rozwój kompetencji z wykorzystaniem AI
- Nauka programowania z AI
- Nauka języka angielskiego z wersją Chat GPT v. 4.0
- Walidacja pomysłów czy rozwiązań
- Przegląd najnowszych innowacji i trendów w AI dla testowania.
- Rola AI w kształtowaniu przyszłości testowania oprogramowania.
- Strategiczne planowanie wdrożenia AI w testowaniu w organizacjach.
- Jak AI wpływa na rolę i umiejętności testera w przyszłości.
- Przyszłość pracy w testowaniu: jak AI zmienia branżę.

## Panel Q&A

- Panel pytań i odpowiedzi
- Podsumowanie

### Aplikacje webowe – wprowadzenie

- Aplikacje webowe – specyfika budowy i działania
- Front-end i Back-end
- Konsola programistyczna przeglądarki internetowej
- Aplikacje internetowe vs aplikacje mobilne i desktopowe
- Najczęstsze problemy spotykane podczas pracy z aplikacjami internetowymi oraz sposoby ich rozwiązywania

### Automatyzacja testów – wprowadzenie

- Co to jest automatyzacja i dlaczego szybko zyskuje na popularności?
- Co należy, a czego nie należy automatyzować?
- Zalety i wady automatyzacji
- Czy automatyzacja testów jest “lekiem na całe zło” w zapewnianiu jakości oprogramowania?
- Automatyzacja testów – dlaczego nie dla wszystkich?

### Wprowadzenie do HTML i konsola programistyczna DevTools

- Co to jest HTML?
- Struktura dokumentu
- Rodzaje znaczników
- WebElementy
- DevTools
- Requesty

### Selenium – wprowadzenie do narzędzi

- Selenium WebDriver
- Dlaczego Selenium od wielu lat jest liderem na rynku automatyzacji aplikacji internetowych?
- Możliwości i ograniczenia Selenium WebDriver

### Narzędzia niezbędne do rozpoczęcia pracy z Selenium WebDriver

- Java Development Kit – instalacja niezbędnego środowiska do programowania w języku JAVA
- IntelliJ IDEA – komercyjne, zintegrowane środowisko programistyczne (IDE)
- Różne przeglądarki internetowe – konfiguracja sterowników/driverów
- MAVEN – profesjonalna organizacja projektów
- Dodatkowo i opcjonalnie Chocolatey

# Ogólny plan projektu

## Selenium WebDriver – instalacja i konfiguracja

- Selenium – wybór wersji, pobranie, instalacja i pierwsza konfiguracja
- Dokumentacja Selenium

## Pierwszy projekt w Selenium

- Konfiguracja pierwszego projektu w Selenium
- Stworzenie prostego przypadku testowego do zautomatyzowania
- Planowanie i omówienie podejścia do realizacji zadania

## Selektory

- Co to są selektory i do czego są używane?
- Typy selektorów – omówienie możliwości i ograniczeń, wad oraz zalet każdego z typów selektorów
- ID
- NAME, CLASS, TAG
- LINKTEXT, PARTIALLINKTEXT
- CSS Selectors
- XPATH

## Obsługa głównych elementów stron www – praktyczne przykłady

- Pola tekstowe, obszary tekstowe – metody clear(),
- Przyciski – metody submit() i click()
- Listy rozwijane, menu – klasa Select
- Elementy wyboru opcji – elementy RadioButton oraz Checkbox
- ZADANIA WYBRANE: Sprawdzanie tekstu elementu, sprawdzanie atrybutów elementu, zaznaczanie opcji w elementach Select, zaznaczanie wybranych opcji na rozwijanych menu i listach, automatyzowanie pól wyboru

## Selenium API – Najczęściej wykorzystywane metody z biblioteki

- Sprawdzenie istnienia elementu na stronie – metoda isElementPresent()
- Sprawdzanie stanu elementu – metody isEnabled(), isSelected(), isDisplayed()
- Nawigacja w przeglądarce – metody back(), forward(), refresh(),
- Maksymalizowanie okna przeglądarki
- Obsługa plików cookies

## TestNG / JUnit5

- Podstawowe adnotacje
  - Test
  - BeforeAll
  - AfterAll
  - BeforeEach
  - AfterEach
- Asercje
  - Parametryzacja testów
  - Napisanie prostego testu z wykorzystaniem JUnit5 i asercji

## Synchronizacja testów - oczekiwanie na zdarzenia w przeglądarce

- Synchronizacja testów z niejawnym oczekiwaniem – ImplicitWait
- Synchronizacja testów z jawnym oczekiwaniem – ExplicitWait
- Synchronizacja testów z niestandardowymi warunkami oczekiwanymi
- Synchronizacja testów z wykorzystaniem klasy FluentWait

## Wzorce projektowe Page Pattern i Page Factory

- Czym jest Page Object Pattern
- Klasa PageFactory
- Napisanie prostego testu z użyciem POP i PageFactory
- Plik konfiguracyjny .properties
- Obsługa różnych przeglądarek
- Zastosowanie dziedziczenia czyli klasa BaseTest
- Dodanie logowania w testach
- Raportowanie za pomocą biblioteki Allure
- Praktyka: Stworzone dotychczas testy przepisujemy z wykorzystaniem poznanych elementów

## Najlepsze praktyki oraz podejście do testów automatycznych

- Omówienie najlepszych praktyk
- Zestaw wskazówek pomocnych w codziennej pracy Testera Automatyzującego

# Ogólny plan projektu

## BDD – Cucumber i Gherkin

- Wprowadzenie do BDD
  - wady / zalety
  - język Gherkin
  - słów kilka o Cucumber Framework
- Cucumber: Instalacja i konfiguracja
- Cucumber: Podstawowe aspekty
  - plugin w IntelliJ
  - słowa kluczowe (feature, scenario, given, when, then, and)
  - plik .features
  - parametryzacja testów
  - Cucumber expressions
- Prosty test z wykorzystaniem Cucumbra
- Połączenie Cucumbra z Page Object Pattern
- Praktyka: Dostosowanie stworzonego projektu pod Cucumber
- Praktyka: Tworzymy testy z wykorzystaniem poznanych elementów

## Opcjonalnie Git, GitLab, Selenium, Grid

- Omówienie najlepszych praktyk
- Zestaw wskazówek pomocnych w codziennej pracy Testera Automatyzującego

## Panel Q&A

- Panel pytań i odpowiedzi
- Podsumowanie

## JĘZYK JAVA DLA TESTERÓW - PODSTAWY

---

### WPROWADZENIE DO PROGRAMOWANIA

- Wprowadzenie do pojęć programowania
- Co to jest programowanie?
- Rola programowania w testowaniu oprogramowania
- Zrozumienie algorytmów i struktur danych
- Podstawowe pojęcia: zmienne, operatory, typy danych
- Wprowadzenie do struktur danych: tablice, listy

### WPROWADZENIE DO JAVY

- Historia i znaczenie Javy w branży IT
- Dlaczego Java jest popularna?
- Przegląd zastosowań Javy w testowaniu oprogramowania
- Środowisko pracy z Javą
- Instalacja i konfiguracja środowiska programistycznego (IDE)
- Omówienie narzędzi: IntelliJ IDEA, Eclipse
- Pierwszy program w Javie
- Struktura programu w Javie
- Tworzenie i uruchamianie prostego programu „Hello, World!”

### PODSTAWOWE KONSTRUKCJE PROGRAMISTYCZNE

- Zmienne i typy danych w Javie
- Przegląd typów danych: int, double, char, boolean
- Operacje na zmiennych
- Operatory i wyrażenia
- Operatory arytmetyczne, logiczne i relacyjne
- Przykłady zastosowania w kodzie
- Struktury sterujące
- Instrukcje warunkowe: if, else, switch
- Pętle: for, while, do-while

## WPROWADZENIE DO PROGRAMOWANIA OBIEKTOWEGO (OOP)

- Podstawy Programowania Obiektowego
- Koncepcje OOP: objekty, klasy, dziedziczenie
- Zasady enkapsulacji, polimorfizmu, abstrakcji
- Tworzenie klas i obiektów w Javie
- Deklaracja klas i obiektów
- Konstruktorzy i destruktory
- Przykłady zastosowań OOP w Javie
- Implementacja prostych klas
- Dziedziczenie i interfejsy

## AUTOMATYZACJA TESTÓW Z WYKORZYSTANIEM JAVY

- Wprowadzenie do testowania automatycznego
- Rola automatyzacji w procesie testowania
- Przegląd narzędzi do automatyzacji testów (np. Selenium)
- Pisanie skryptów testowych w Javie
- Struktura skryptu testowego
- Tworzenie testów jednostkowych z użyciem JUnit
- Przykłady zastosowania Javy w testach automatycznych
- Automatyzacja testów dla aplikacji webowych
- Tworzenie i uruchamianie testów funkcjonalnych

## MINI PROJEKT I WARSZTATY

- Metodyki testowania klawiatury.
- Zaawansowane techniki używania czytników ekranu.
- Metody oceny kontrastu i kolorów.
- Scenariusze testów dla różnych interfejsów użytkownika.
- Uwzględnienie różnych środowisk testowych.

## PANEL Q&A

# Ogólny plan projektu

## PRAKTYCZNE UMIEJĘTNOŚCI TESTERA OPROGRAMOWANIA

---

### Wprowadzenie do testowania oprogramowania

- Co to jest testowanie oprogramowania i dlaczego jest ważne?
- Rola testera w procesie tworzenia oprogramowania.
- Przegląd typów testów (manualne vs automatyczne, testy funkcjonalne, regresyjne, etc.).

### Podstawy tworzenia przypadków testowych

- Co to jest przypadek testowy?
- Struktura przypadku testowego: krok po kroku.
- Definiowanie warunków początkowych, danych wejściowych, działań oraz oczekiwanych wyników.
- Przykłady dobrze napisanych przypadków testowych.

### Tworzenie skutecznych przypadków testowych

- Techniki pisania przypadków testowych: ekwiwalentne klasy, analiza wartości brzegowych, pokrycie decyzji.
- Jak identyfikować scenariusze testowe na podstawie wymagań?
- Rola priorytetyzacji przypadków testowych.
- Praktyczne ćwiczenia: Pisanie przypadków testowych dla prostego scenariusza.

### Zgłaszanie błędów

- Co to jest błąd (bug) i jak go definiować?
- Rola narzędzi do zarządzania błędami (np. JIRA, Bugzilla).
- Kluczowe elementy zgłoszenia błędu: tytuł, opis, kroki do odtworzenia, oczekiwany vs rzeczywisty wynik, zrzuty ekranu/logi.
- Przykłady dobrze i źle zgłoszonych błędów.

### Praktyka zgłaszania błędów

- Jak unikać typowych błędów przy zgłaszaniu defektów?
- Praca z narzędziami do zarządzania defektami: Tworzenie, aktualizowanie i śledzenie zgłoszeń.
- Praktyczne ćwiczenia: Zgłaszanie błędów na podstawie symulowanych przypadków.

### Testowanie oparte na wymaganiach

- Analiza wymagań i ich rola w procesie testowania.
- Jak przekształcać wymagania w przypadki testowe?
- Testowanie przypadków granicznych i nietypowych scenariuszy.



# Ogólny plan projektu

## Weryfikacja i walidacja

- Różnice między weryfikacją a walidacją.
- Jak zapewnić, że przypadki testowe pokrywają wszystkie wymagania?
- Techniki zapewniania jakości w procesie testowania.

## Komunikacja w zespole testerskim

- Rola komunikacji w procesie testowania.
- Jak skutecznie współpracować z deweloperami i innymi testerami?
- Jak raportować wyniki testów i zgłaszać postępy w pracy?

## Wprowadzenie do JIRA

- Co to jest JIRA?: Wprowadzenie do narzędzia do zarządzania projektami zwinnego.
- Rola JIRA w Scrum i Agile PM: Jak JIRA wspiera zarządzanie projektami Agile?
- Podstawy nawigacji w JIRA: Interfejs użytkownika, kluczowe funkcje i narzędzia.
- Tworzenie i zarządzanie backlogiem w JIRA: Jak efektywnie zarządzać wymaganiami i zadaniami?
- Tworzenie i śledzenie zadań: Epiki, user stories, zadania i bugi w JIRA.
- Użycie tablicy Scrum w JIRA: Jak monitorować postęp pracy zespołu?
- Raportowanie i analiza w JIRA: Wykresy spalania, raporty i inne narzędzia analityczne.
- Automatyzacja procesów w JIRA: Tworzenie automatycznych reguł i workflow.
- Integracja JIRA z innymi narzędziami: Łączenie JIRA z narzędziami do CI/CD, testowania, komunikacji itp.

## SZKOLENIA NARZĘDZIOWE

---

To szkolenie zapewnia kompleksowe wprowadzenie do narzędzi niezbędnych w pracy każdego testera oprogramowania, umożliwiając skuteczne zarządzanie projektami, organizację testów oraz analizę wyników. Program obejmuje cztery kluczowe obszary, które stanowią fundament codziennej pracy testerskiej.

- **Wprowadzenie do DevTools:** Poznasz narzędzie przeglądarkowe DevTools, które jest nieocenione w analizie i debugowaniu aplikacji webowych. W trakcie tego modułu nauczysz się, jak śledzić ruch sieciowy, analizować elementy HTML, CSS i JavaScript, a także diagnozować problemy z wydajnością aplikacji.
- **Wprowadzenie do JIRA:** JIRA to jedno z najczęściej używanych narzędzi do zarządzania projektami w branży IT. W tym module dowiesz się, jak tworzyć zgłoszenia, monitorować postępy w projekcie oraz efektywnie współpracować z zespołem deweloperskim. Nauczysz się również, jak integrować JIRA z innymi narzędziami testerskimi.
- **Wprowadzenie do Zephyr Scale:** Zephyr Scale to rozbudowany system zarządzania testami, który doskonale współpracuje z JIRA. W tym module poznasz podstawy tworzenia i zarządzania planami testów, raportowania wyników oraz śledzenia pokrycia testów. To narzędzie znacząco ułatwia organizację pracy zespołu testerskiego.
- **Wprowadzenie do niezbędnych pluginów dla testerów oprogramowania:** Poznasz kluczowe wtyczki i rozszerzenia, które ułatwiają pracę testera, takie jak dodatki do przeglądarek wspomagające analizę aplikacji webowych, narzędzia do automatyzacji procesów testowych oraz rozszerzenia do zarządzania testami i zgłoszeniami.
- Szkolenie jest skoncentrowane na praktycznym wykorzystaniu narzędzi, dzięki czemu uczestnicy od razu mogą wdrożyć zdobytą wiedzę w swoich projektach testerskich. To doskonały sposób na zdobycie umiejętności, które zwiększą efektywność pracy w środowisku testerskim.

# Ogólny plan projektu

## INNE MNIEJSZE MINI SZKOLENIA TEMATYCZNE

Nasze mini szkolenia to zbiór specjalistycznych kursów, które wzbogacą Twoje kompetencje testerskie o kluczowe umiejętności, niezbędne w dzisiejszym dynamicznie rozwijającym się świecie IT. Zróżnicowany program szkoleniowy został zaprojektowany tak, aby dostarczyć praktyczną wiedzę z różnych obszarów testowania, przygotowując Cię na różnorodne wyzwania w projektach IT. Oto, czego możesz się spodziewać:

**Testowanie dostępności z WCAG 2.2:** W trakcie tego modułu nauczysz się testować aplikacje pod kątem ich dostępności dla osób z niepełnosprawnościami, zgodnie z najnowszymi standardami WCAG 2.2. Poznasz najlepsze praktyki weryfikacji dostępności na różnych urządzeniach i nauczysz się identyfikować bariery, które utrudniają użytkownikom korzystanie z aplikacji. To umiejętność, która coraz częściej jest wymagana przez pracodawców, ponieważ dostępność cyfrowa staje się standardem.

**Testy wydajnościowe:** Dowiesz się, jak przeprowadzać testy wydajnościowe, które pozwalają ocenić, jak systemy radzą sobie pod obciążeniem. Poznasz narzędzia i techniki, które pomogą Ci monitorować stabilność aplikacji w warunkach rzeczywistego użytkownika oraz wykrywać wąskie gardła w działaniu systemów. To niezbędna umiejętność, która pozwoli Ci zadbać o niezawodność aplikacji w warunkach dużej liczby użytkowników.

**Testowanie aplikacji mobilnych na Android i iOS:** Przygotujemy Cię do testowania aplikacji mobilnych na dwóch najpopularniejszych platformach – Android i iOS. Dowiesz się, jak analizować wydajność, funkcjonalność i doświadczenia użytkownika na urządzeniach mobilnych. Zrozumiesz różnice między systemami i jak dostosować techniki testowania, aby zapewnić jak najwyższą jakość aplikacji na obu platformach.

**Podstawy SQL do testowania baz danych:** Umiejętność pracy z bazami danych to fundament w testowaniu aplikacji, które przetwarzają duże ilości danych. W tym module nauczysz się podstaw SQL, które pozwolą Ci na weryfikowanie poprawności danych, przeprowadzanie zapytań oraz testowanie wydajności i integralności baz danych. Wiedza ta umożliwi Ci efektywną współpracę z deweloperami i administratorami baz danych.

# Ogólny plan projektu

## **Automatyzacja procesów z RPA (Robotic Process Automation):**

Wprowadzimy Cię w świat automatyzacji procesów biznesowych za pomocą RPA, co pozwala na znaczące usprawnienie powtarzalnych czynności. Nauczysz się, jak wykorzystać narzędzia RPA do automatyzacji testów i procesów, co zredukuje błędy ludzkie i przyspieszy realizację zadań. To przyszłościowa umiejętność, szczególnie ceniona w nowoczesnych organizacjach.

**System kontroli wersji GIT:** Kontrola wersji to podstawa w pracy zespołowej nad projektami IT. W ramach tego szkolenia poznasz GIT, popularny system kontroli wersji, który umożliwia zarządzanie zmianami w kodzie, śledzenie postępów i współpracę z zespołem w sposób efektywny. Nauczysz się, jak tworzyć i zarządzać repozytoriami, co jest kluczowe w pracy nad złożonymi projektami programistycznymi i testerskimi.

**Inne kluczowe narzędzia i techniki testerskie:** Oprócz powyższych tematów, mini szkolenia obejmą wiele mniejszych, ale równie istotnych zagadnień, które wzbogacą Twój zestaw umiejętności testerskich. Dowiesz się, jak korzystać z różnych narzędzi do automatyzacji testów, weryfikacji jakości kodu, a także jak tworzyć testy end-to-end, które symulują działania użytkowników na różnych poziomach aplikacji.

Każde z tych szkoleń jest skoncentrowane na praktycznych umiejętnościach, które można od razu wykorzystać w codziennej pracy testera. Udział w tych kursach nie tylko zwiększy Twoje kompetencje techniczne, ale również znacząco podniesie Twoją atrakcyjność na rynku pracy, przygotowując Cię na współpracę z różnorodnymi zespołami w dynamicznym środowisku IT.

# Ogólny plan projektu

## SZKOLENIE HR DLA TESTERÓW OPROGRAMOWANIA

---

### **Tworzenie profesjonalnego CV**

- Wprowadzenie do tworzenia CV: Jakie elementy powinno zawierać CV dla testera oprogramowania?
- Formatowanie i układ: Jak zaprezentować swoje doświadczenie i umiejętności w sposób czytelny i atrakcyjny?
- Kluczowe sekcje w CV: Edukacja, doświadczenie, umiejętności techniczne, projekty, certyfikaty.
- Dostosowanie CV do ofert pracy: Jak modyfikować CV w zależności od stanowiska, na które aplikujesz?
- Przykłady dobrych praktyk: Analiza przykładowych CV testerów oprogramowania.

### **Budowanie profilu na LinkedIn i GitHub**

- Tworzenie profilu na LinkedIn: Jak zoptymalizować swój profil, aby przyciągnąć rekruterów i pracodawców?
- Sekcje profilu LinkedIn: Nagłówek, podsumowanie, doświadczenie, umiejętności, rekomendacje.
- Budowanie sieci kontaktów: Jak efektywnie nawiązywać i utrzymywać kontakty na LinkedIn?
- Profil na GitHub: Dlaczego warto mieć GitHub i jak go wykorzystać do prezentacji swoich umiejętności?
- Prezentacja projektów na GitHubie: Jak publikować i opisywać swoje projekty, aby były atrakcyjne dla rekruterów?

### **Przygotowanie do rozmowy kwalifikacyjnej**

- Rodzaje rozmów kwalifikacyjnych: Czego możesz się spodziewać na rozmowie na stanowisko testera oprogramowania?
- Typowe pytania rekrutacyjne: Przegląd najczęściej zadawanych pytań technicznych i behawioralnych.
- Przykładowe odpowiedzi: Jak odpowiedzieć na pytania dotyczące doświadczenia, umiejętności i motywacji?
- Praktyczne ćwiczenia: Symulacje rozmów kwalifikacyjnych z omówieniem odpowiedzi i sugestiami poprawy.
- Negocjacja warunków zatrudnienia: Jak rozmawiać o wynagrodzeniu i innych benefitach?

# Ogólny plan projektu

## **Efektywne szukanie pracy**

- Strategie szukania pracy: Jakie są najlepsze metody na znalezienie pracy jako tester oprogramowania?
- Wykorzystanie portali z ofertami pracy: Jak efektywnie korzystać z platform takich jak LinkedIn, Pracuj.pl, Indeed.
- Networking: Jakie znaczenie ma networking i jak go efektywnie wykorzystywać w poszukiwaniu pracy?
- Przygotowanie aplikacji: Jak tworzyć spersonalizowane listy motywacyjne i dopasowane CV?
- Monitorowanie aplikacji: Jak śledzić swoje aplikacje i utrzymywać kontakt z rekruterami?

## **Dalszy rozwój i poszerzanie kompetencji**

- Planowanie kariery: Jak ustalić swoje cele zawodowe i jak je osiągać?
- Samodzielne doskonalenie: Kursy online, certyfikaty, samodzielne projekty – jak rozwijać swoje umiejętności?
- Śledzenie trendów w branży: Jak być na bieżąco z nowościami w testowaniu oprogramowania?
- Uczestnictwo w społecznościach testerskich: Meetupy, konferencje, fora – jak angażować się w środowisko testerskie?
- Mentoring i wsparcie: Jak znaleźć mentora i korzystać z jego doświadczenia?

## **PRACA: TESTOWANIE KOMERCYJNEJ APLIKACJI WEBOWEJ**

- Tworzenie przypadków i scenariuszy testowych
- Testy manualne
- Testy automatyczne (regresja)
- Praca z dokumentacją testową
- Testy akceptacyjne i systemowe
- Zgłaszanie błędów
- Praca samodzielna i w zespole
- Raportowanie

## Kompleksowy program szkoleniowy

Nasz program rozpoczyna się od intensywnych spotkań szkoleniowych, które obejmują wszystkie kluczowe aspekty testowania oprogramowania. Zajęcia są prowadzone przez doświadczonych praktyków, którzy nie tylko posiadają bogatą wiedzę teoretyczną, ale również szerokie doświadczenie zawodowe. W trakcie szkolenia poznasz różnorodne techniki testowania, narzędzia, które są standardem w branży, a także najlepsze praktyki i metodyki, takie jak Agile czy Scrum. Zdobędziesz solidną wiedzę z zakresu manualnego i automatycznego testowania, zarządzania przypadkami testowymi, a także raportowania i dokumentowania błędów. Opowiemy między innymi o takich tematach jak: **testowanie aplikacji webowych, testowanie aplikacji mobilnych, automatyzacji testów, testowaniu wydajnościowym, testowaniu integracyjnym, testowaniu baz danych, testach bezpieczeństwa (cybersecurity), rozwiązaniach raportowych, architekturze testów i automatyzacji testów, robotyzacji procesów biznesowych, testach dostępności, AI w testowaniu i wiele innych.**

Dzięki temu będziesz doskonale przygotowany do pracy w rzeczywistych projektach IT.

## Praktyka na realnym, komercyjnym projekcie

To, co nas wyróżnia, to możliwość uczestniczenia w testach komercyjnych aplikacji, które są rzeczywiście wykorzystywane przez użytkowników. Będziesz pracować nad prawdziwymi problemami i wyzwaniem, z jakimi zmagają się firmy na co dzień. W praktyce oznacza to, że nie tylko utrwalisz zdobytą wiedzę, ale także zdobędziesz cenne doświadczenie rynkowe. Będziesz mógł zobaczyć, jak wygląda cykl życia produktu od strony testów i jak Twoja praca przyczynia się do jego jakości. Takie doświadczenie jest niezwykle cenne przez pracodawców, ponieważ pokazuje, że potrafisz efektywnie działać w warunkach zbliżonych do rzeczywistych, a Twoja praca ma bezpośredni wpływ na jakość produktu.

## Staż, który stworzy Twoje CV

Po zakończeniu naszego programu otrzymasz nie tylko certyfikat, który potwierdzi zdobyte przez Ciebie umiejętności, ale również fantastyczny wpis w CV, który będzie dowodem Twojego stażu w Quality Island. Staż w naszej firmie to coś więcej niż tylko formalność – to intensywny okres pracy nad realnymi projektami, gdzie zdobędziesz doświadczenie, które jest niezwykle atrakcyjne na rynku pracy. Taki wpis znacząco zwiększa Twoje szanse na rynku pracy, dając Ci przewagę nad kandydatami, którzy posiadają jedynie teoretyczną wiedzę. **Pracodawcy widzą, że masz już za sobą realne doświadczenie, dodatkowo w firmie, która profesjonalnie zajmuje się testowaniem oprogramowania, co czyni Cię bardzo wartościowym kandydatem.**

## Indywidualne wsparcie

Każdy uczestnik naszego projektu jest prowadzony przez doświadczonych specjalistów od początku do końca. Nasz zespół mentorów i trenerów nie tylko przekazuje wiedzę, ale również wspiera Cię na każdym etapie nauki i praktyki. Otrzymasz indywidualne wsparcie, które pomoże Ci zrozumieć trudniejsze zagadnienia, rozwijać swoje umiejętności i radzić sobie z wyzwaniami. Dzięki temu masz pewność, że nie zostaniesz sam z problemami, a Twoje wątpliwości zostaną szybko rozwiązane. Nasze podejście zapewnia, że każdy uczestnik programu wyjdzie z niego w pełni przygotowany do podjęcia pracy jako junior tester oprogramowania.



Zapraszamy do kontaktu:



**+48 531 119 616**



**[kontakt@qualityisland.pl](mailto:kontakt@qualityisland.pl)**

