

PROGRAM KURSU Tester manualny

■ BLOK 0 - PREWORK

WSTĘP DO TESTOWANIA

■ WSTĘP DO APLIKACJI INT.

- Wstęp
- Struktura aplikacji
- Technologie
- Przeglądarki internetowe

■ PROJEKT IT

- Wstęp
- Metodyki prowadzenia projektu IT
- Role w projekcie IT

■ WSTĘP DO CERTYFIKACJI ISTQB

- Wstęp
- Podstawy testowania
- Typy i poziomy testowania
- Statyczne techniki testowania
- Techniki projektowania testów
- Zarządzanie testowaniem

■ SŁOWNIK ISTQB

- Słownik

■ BAZY DANYCH WPROWADZENIE

- Wprowadzenie do baz danych
- Dane konfiguracyjne
- Pozyskiwanie informacji o bazie danych
- Pobieranie danych z tabeli

■ BLOK 1

ISTQB FOUNDATION LEVEL

■ PODSTAWY TESTOWANIA

1. Co to jest testowanie?

- Typowe cele testowania,
- testowanie a debugowanie.

2. Dlaczego testowanie jest niezbędne?

- Znaczenie testowania dla powodzenia projektu,
- zapewnienie jakości a testowanie,
- pomyłki, defekty, awarie,
- defekty, podstawowe przyczyny oraz skutki.

3. Siedem zasad testowania

4. Proces testowy

- Proces testowy w kontekście,
- czynności i zadania testowe,

- produkty pracy związane z testowaniem,
- śledzenie powiązań między podstawą testów a produktami pracy związanymi z testowaniem.

5. Psychologia testowania

- Psychologia człowieka a testowanie,
- różnice w sposobie myślenia testerów i programistów.

■ TESTOWANIE W CYKLU ŻYCIA OPROGRAMOWANIA

1. Modele cyklu życia oprogramowania

- Wytwarzanie oprogramowania a testowanie oprogramowania,
- modele cyklu życia oprogramowania w kontekście.

2. Poziomy testów

- Testy modułowe,
- testy integracyjne,
- testy systemowe,
- testy akceptacyjne.

3. Typy testów

- Testowanie funkcjonalne,
- testowanie niefunkcjonalne,
- testowanie białoskrzynkowe,
- testowanie związane ze zmianami.

4. Testowanie pielęgnacyjne

- Zdarzenia wywołujące pielęgnację,
- analiza wpływu związana z pielęgnacją.

■ TESTOWANIE STATYSTYCZNE

1. 1. Podstawy testowania statycznego

- Produkty pracy badane metodą testowania statycznego,
- zalety testowania statycznego,
- różnice między testowaniem statycznym a dynamicznym.

2. Proces przeglądu

- Proces przeglądu produktów pracy,
- role i obowiązki w przeglądzie formalnym,
- typy przeglądów,
- stosowane techniki przeglądu,
- czynniki powodzenia związane z przeglądami.

■ TECHNIKI TESTOWANIA

1. Kategorie technik testowania

- Wybór technik testowania,
- kategorie technik testowania i ich cechy charakterystyczne.

2. Czarnoskrzynkowe techniki testowania

- Podział na klasy równoważności,
- analiza wartości brzegowych,
- testowanie w oparciu o tablicę decyzyjną,
- testowanie przejść pomiędzy stanami,
- testowanie oparte na przypadkach użycia.

3. Białoskrzynkowe techniki testowania

- Testowanie i pokrycie instrukcji kodu,
- testowanie i pokrycie decyzji,
- korzyści wynikające z testowania instrukcji i testowania decyzji.

4. Techniki testowania oparte na doświadczeniu

- Zgadywanie błędów,
- testowanie eksploracyjne,
- testowanie w oparciu o listę kontrolną.

5. Wybór technik testowania

ZARZĄDZANIE TESTAMI

1. Organizacja testów

- Niezależne testowanie,
- zadania kierownika testów oraz testera.

2. Planowanie i szacowanie testów

- Cel i treść planu testów,
- strategię testów i podejście do testowania,
- kryteria wejścia i wyjścia (definicja gotowości i definicja ukończenia),
- harmonogram wykonywania testów,
- czynniki wpływające na pracochłonność testowania,
- techniki szacowania testów.

3. Monitorowanie testów i nadzór nad testami

- Miary stosowane w odniesieniu do testowania,
- cel, treść i odbiorcy raportów z testów.

4. Zarządzanie konfiguracją

5. Czynniki ryzyka a testowanie

- Definicja ryzyka,
- czynniki ryzyka produktowego i projektowego,
- testowanie oparte na ryzyku a jakość produktu.

6. Zarządzanie defektami

NARZĘDZIA WSPOMAGAJĄCE TESTOWANIE

1. Uwarunkowania związane z narzędziami testowymi

- Klasyfikacja narzędzi testowych,
- korzyści i czynniki ryzyka związane z automatyzacją testowania,
- szczególnie uwarunkowania związane z narzędziami do wykonywania testów i zarządzania testami.

2. Skuteczne korzystanie z narzędzi

- Główne zasady wyboru narzędzi,
- projekty pilotażowe związane z wprowadzaniem narzędzia w organizacji,
- czynniki sukcesu związane z narzędziami.

BLOK 2 WARSZTATY

WARSZTAT 1 – AGILE I PRZYPADEK TESTOWY

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

WARSZTAT 2 – PRZYPADEK TESTOWY WARSZTAT

1. Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś.
2. 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.
3. Tworzenie przypadków testowych na podstawie dokumentacji oraz dostępnej aplikacji z naszego Test Labu.

WARSZTAT 3 – BŁĄD

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

WARSZTAT 4 – BAZY DANYCH

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

WARSZTAT 5 – API

1. Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś.
2. Web Services
 - definicja,
 - zastosowanie – przykłady,
 - zalety i wady,
3. XML i JSON
 - o języku XML,
 - znaczniki,
 - struktura,
 - poprawność składniowa,
 - [CDATA],
 - o języku JSON,
 - JSON – struktura,
 - poprawność składniowa.
4. SOAP
 - o SOAP,
 - struktura SOAP,
 - WSDL,
 - Soap UI – obsługa narzędzia,

5. HTTP

- protokół internetowy,
- HTTP request,
- metody HTTP,
- HTTP response,
- response code.

6. REST API

- API,
- REST – zasady,
- metody REST (HTTP),
- response code REST,
- obsługa narzędzia POSTMAN.

WARSZTAT 6 – RETESTY

1. Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś.
2. Retest a regresja
 - Kiedy retest a kiedy regresja?
 - Jak wykonywać retesty?
 - Jak przygotować się do regresji?
3. Wykonujemy retesty błędów znalezionych podczas modułu praktycznego.

WARSZTAT 7 – TESTOWANIE MOBILNE I RAPORTOWANIE

1. Daily scrum (stand-up) – omówienie, co udało się zrobić wczoraj i co planujemy zrobić dziś.
2. Aplikacje mobilne
 - co to jest aplikacja mobilna,
 - typy aplikacji mobilnych,
 - mobilne systemy operacyjne.
3. Testy aplikacji mobilnych
 - Jak zacząć,
 - fragmentacja rynku – jak wybrać urządzenie.
 - sieć,
 - pakiety,
 - użyteczność,
 - Emulatory iOS – Android.
4. 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.
5. Tworzenie raportów – przygotowanie raportów z prac z wykorzystaniem danych zgromadzonych w narzędziu Jira.
6. Retrospektywa Agile – podsumowanie kursu i omówienie uwag, które mogą prowadzić do udoskonalenia produktu.

TEST LAB LINUX HTML I CSS

LINUX

1. Podstawy Linuxa, najpopularniejsze wersje Linuxa
2. Obsługa emulatora terminala oraz podstawy Vi
3. Podręczniki systemowe
4. 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
5. Podstawowe komendy: grep, find, pipe (|). przekierowanie
6. Użytkownicy: typy, tworzenie i zarządzanie
7. Manager pakietów (Ubuntu + macOS)
8. Zaawansowane tematy: procesy, zdalny terminal (screen), harmonogram zadań, symlinki, hardlinki, zmienne systemowe

HTML I CSS

1. Znaczniki, struktura dokumentu
2. Formatowanie tekstu
3. Hiperłącza, obrazy, tabele
4. CSS – kilka podstawowych stylów

Obszerny spis stron i narzędzi potrzebnych w trakcie kursu

TEST LAB

1. Nauka UML i BPMN
2. Testy aplikacji FinTech
3. Testy aplikacji CRM
4. Testy aplikacji HR
5. Automatyzacja testów Selenium IDE (record&play)
6. Praca z dokumentacją techniczną