BO·TT·EGA
IT minds

Program szkolenia:

Testowanie automatyczne

Informacje:

Nazwa:	**Testowanie automatyczne**
Kod:	**NET-.NET test**
Kategoria:	.NET
Grupa docelowa:	developerzy
Czas trwania:	3 dni
Forma:	50% wykłady / 50% warsztaty

Zaczynamy od omówienia i przećwiczenia w praktyce zasad SOLID oraz innych strategii, dzięki którym kod staje się czytelniejszy i łatwiejszy w utrzymaniu. Podczas zadań praktycznych zobaczymy jakie kroki pozwalają na refaktoryzację istniejącego kodu do "lepszej" postaci… bo przecież większość pracy programisty polega na utrzymywaniu i modyfikacji już zaimplementowanych rozwiązań. Kod "lepszy" prawie zawsze implikuje również charakterystykę: kod "łatwo testowalny".

Po zapoznaniu się/utrwaleniu najistotniejszych reguł programowania obiektowego intensywnie zagłębiamy się w temat testów jednostkowych. Uczestnicy poznają zarówno najlepsze praktyki dotyczące tego zagadnienia, jak również różne korzyści płynące z pokrycia kodu testami automatycznymi. Wbrew pozorom zredukowana liczba błędów w finalnym oprogramowaniu jest tylko jednym z wielu celów. Świadome – pragmatyczne, a nie dogmatyczne – zastosowanie testów pomaga pisać przyjaźniejszy kod oraz, co istotne – zwiększyć satysfakcję programisty z wykonywanej pracy. Podczas dyskusji oraz wykonywanych zadań trener dzieli się swoim wieloletnim doświadczeniem, spostrzeżeniami oraz wypracowanymi sposobami na możliwe bezbolesne rozpoczęcie bądź rozwinięcie przygody uczestników z testami. Pierwszego dnia poruszamy również temat Test Driven Development (TDD) – jednej z popularnych metod pisania testów jednostkowych.

Drugiego dnia skupiamy się głównie na praktycznych aspektach testowania. Poznajemy pojęcie "mocków" i podczas wielu zadań badamy jak testy sprawdzają się w sytuacjach, które każdy programista spotyka na co dzień. Omawiamy również bardzo szeroki wachlarz dostępnych na rynku bibliotek i narzędzi. Druga część dnia to dogłębne i szczegółowe przećwiczenie przysparzających trudności scenariuszy: testowanie baz danych, komponentów działających w środowisku wielowątkowym oraz integracji tworzonych systemów z usługami zewnętrznymi.

Dzień trzeci to głównie poszerzanie horyzontów, mające na celu wskazanie dodatkowych zastosowań testów ("convention tests") oraz sposobów ich tworzenia ("Behavior Driven Development – BDD"). Wisienką na testowym torcie jest przybliżenie uczestnikom szkolenia koncepcji "testowania mutacyjnego ", o której niewiele osób słyszało. Końcówka materiału szkoleniowego to kilka dodatkowych ćwiczeń utrwalających zdobytą podczas trzech dni wiedzę. Następne poświęcimy kilkadziesiąt minut na testowanie kodu JavaScript, co często jest najbardziej zaniedbanym obszarem podczas pokrywania oprogramowania testami.

Niniejsze szkolenie to w ok 80% ćwiczenia praktyczne, dzięki którym już następnego dnia uczestnicy są w stanie wykorzystać zdobytą wiedzę w codziennej pracy. Części teoretyczne są uzupełnieniem ćwiczeń oraz punktem wyjściowym do aktywnej dyskusji z uczestnikami.

Każdy z trzech dni podzielony jest na sześć sekcji trwających około godziny. Pozostały czas przeznaczony jest na dyskusje trenera z zespołem, wymianę doświadczeń oraz wspólne poszukiwanie najlepszych rozwiązań.

Szkolenie kierowane jest do programistów .NET, gdyż ćwiczenia oraz prezentowane narzędzia powiązane są z tym właśnie środowiskiem.

W trakcie szkolenia uczestnicy będą wykorzystywać system kontroli wersji Git, co w przypadku nieznajomości tego narzędzia może być dodatkowym walorem edukacyjnym.

Materiały wstępne

Przed szkoleniem możesz zapoznać się z serią naszych artykułów: Testowanie automatyczne.

Zalety szkolenia:

- Narzędzia automatyzacji
- Zagadnienia architektury aplikacji wspierającej testowalność kodu
- Najlepsze wzorce i praktyki
- Aspekty Behavior Driven Development i Domain Driven Design

BOTTEGA IT minds

Szczegółowy program:

1. Dzień 1

- 1.1. SOLID
  - 1.1.1. Co to jest SOLID?
  - 1.1.2. Zastosowanie SOLID w praktyce
  - 1.1.3. Dodatkowe dobre praktyki
- 1.2. Testy jednostkowe
  - 1.2.1. Co to są i po co są testy?
  - 1.2.2. Jak pisać testy
  - 1.2.3. Test Driven Development (TDD)

2. Dzień 2

- 2.1. Mocki
- 2.2. Praktyczne scenariusze wykorzystania testów
- 2.3. Biblioteki i narzędzia – przegląd i rekomendacje
- 2.4. Testy bazy danych
- 2.5. Testowanie scenariuszy wielowątkowych
- 2.6. Testowanie integracji z usługami zewnętrznymi

3. Dzień 3

- 3.1. Convention tests
- 3.2. Behavior Driven Development (BDD)
- 3.3. Testowanie mutacyjne
- 3.4. Dodatkowe ćwiczenia praktyczne
- 3.5. Testowanie JavaScript