Tytuł szkolenia: Testy jednostkowe w .NET Core - wersja rozszerzona

Kod szkolenia: testy-jednostkowe-NET-Core-wersja-rozszerzona

Wprowadzenie

Testy jednostkowe umożliwiają podniesienie niezawodności aplikacji oraz jakości kodu. Podczas szkolenia, uczestnicy poznają narzędzia oraz zasady tworzenia testów jednostkowych w metodyce TDD. Przeprowadzimy również refaktoryzację „starego" kodu aplikacji, w celu umożliwienia testów jednostkowych. Szkolenie prowadzone jest w formie warsztatów połączonych z dyskusją.

Adresaci szkolenia

Programiści .NET

Cel szkolenia

Celem szkolenia jest nabycie wiedzy tworzenia skutecznych testów jednostkowych w celu podniesienia niezawodności i jakości kodu.

Czas i forma szkolenia

- 28 godzin (4 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

Plan szkolenia

1. Test-Driven Development

· Wprowadzenie do TDD
· Cykl Red-Green-Refactor
· Zasady FIRST
· Utworzenie pierwszego testu jednostkowego
· Uruchamianie i debugowanie testów jednostkowych
· Korzyści z testów jednostkowych

2. Biblioteka nUnit / xUnit

· Instalacja biblioteki nUnit/xUnit
· Utworzenie testu
· Prawidłowe nazewnictwo testów jednostkowych
· Weryfikacja wyniku
· Parametryzacja przypadków testowych
· Weryfikacja wartości liczbowych, tekstów oraz dat
· Weryfikacja kolekcji
· Weryfikacja zwracanego typu
· Weryfikacja metod void
· Weryfikacja wyjątkó
· Weryfikacja zdarzeń
· Weryfikacja czasu wykonania
· Weryfikacja metody asynchronicznej

3. **Biblioteka Fluent Assertions**

- Instalacja biblioteki FluentAssertions
- Weryfikacja poprzez przykład
- Weryfikacja pustych wartości
- Weryfikacja tekstów
- Weryfikacja zakresów liczbowych, daty i czasu
- Weryfikacja kolekcji i słowników
- Weryfikacja wyjątków
- Weryfikacja zdarzeń
- Weryfikacja czasu wykonania

4. **Tworzenie atrap (Mock)**

- Instalacja biblioteki Moq
- Utworzenie atrapy
- Definicja zachowania metody
- Definicja zachowania właściwości
- Definicja zachowania zdarzenia
- Weryfikacja wywołania metody
- Weryfikacja wywołania właściwości
- Linq To Mocks

5. **Refaktoryzacja**

- Dostosowanie kodu w celu przeprowadzenia testów jednostkowych
- Przydatne wzorce projektowe

6. **Testy integracyjne (+1 dzień)**

- Testowanie usług sieciowych REST API
- Testowanie bazy danych w pamięci (Entity Framework)

7. **Testy mutacyjne**

- Użycie Stryker.net