

Program szkolenia:

Efektywna praca z istniejącymi systemami (Techniki wprowadzania zmian, Refaktoryzacja, Automatyzacja)

Informacje ogólne

Nazwa:	Efektywna praca z istniejącymi systemami (Techniki wprowadzania zmian, Refaktoryzacja, Automatyzacja)
Kod:	Testowanie-legacy
Kategoria:	Testowanie
Grupa docelowa:	Testerzy, programiści, projektanci
Czas trwania:	2 dni
Forma:	50% wykłady / 50% warsztaty

Podczas szkolenia uczestnicy poznają techniki ułatwiające utrzymanie istniejących systemów. Głównym tematem będzie poznawanie zachowania i struktury w celu naprawiania błędów i wprowadzania nowych funkcjonalności oraz refaktoryzacja designu.

Podczas warsztatów praktycznych uczestnicy będą refaktoryzować istniejący kod i pisać testy regresyjne. Uczestnicy zrefaktoryzują istniejący system pod kątem wprowadzenia nowych funkcjonalności.

Wsparcie trenera wspomagającego

Podczas tego szkolenia stawiamy szczególny nacisk na ćwiczenia praktyczne rozwiązywane podczas warsztatów.

Dlatego szkolenie jest prowadzone przez **dwóch trenerów**: głównego i wspomagającego, który asystuje podczas warsztatów. Dzięki temu każdy uczestnik ma nieograniczony dostęp do pomocy i wiedzy eksperckiej.

Zalety szkolenia:

- » Rzeczywiste zastosowania
- » Najlepsze wzorce i praktyki
- » Dwóch trenerów prowadzących szkolenie

Program szkolenia:

1. Radzenie sobie ze zmianą

1.1. 4 powody zmian w oprogramowaniu

1.2. Zarządzanie długiem technicznym

1.3. Testowanie eksploracyjne jako sposób poznawania zachowania systemu

2. Testowanie regresyjne

2.1. Sposoby testowania systemu

2.2. Rodzaje testów i przykłady ich wykorzystania

2.3. Automatyzacja procesu testowania

2.4. Wybór strategii testowania w projekcie

2.5. Pisanie testów automatycznych w projekcie, który ich nie posiada

3. Czytanie kodu

3.1. Zbieranie i interpretowanie metryk

3.2. Wykrywanie punktów krytycznych systemu

3.3. Przegląd kodu i identyfikowanie "zapachów"

4. Refaktoryzacja

4.1. Podstawowe techniki refaktoryzacji

4.2. Wsparcie ze strony IDE

4.3. Identyfikowanie szwów i rozdzielanie zależności

4.4. Zaawansowane refaktoringi w wielu krokach

4.5. Tworzenie planu refaktoringu

4.6. Refaktoryzacja designu

4.7. Refaktoryzacja do wzorca

5. Automatyzacja

5.1. Automatyzacja budowania projektu przy użyciu Maven / Ant / Gradle (do wyboru)

5.2. Wykorzystanie serwera Continuous Integration

5.3. Optymalizacja procesu budowania pod kątem częstszych releasów