

## Program szkolenia:

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

## Informacje:

<b>Nazwa:</b>	<b>Efektywna praca z istniejącymi systemami (Techniki wprowadzania zmian, Refaktoryzacja, Automatyzacja)</b>
<b>Kod:</b>	<b>legacy-legacy</b>
<b>Kategoria:</b>	Refaktoryzacja kodu legacy
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	40% wykłady / 60% 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.

## Zalety szkolenia:

- Rzeczywiste zastosowania
- Najlepsze wzorce i praktyki
- Strategiczne podejście do testowania automatycznego

## Szczegółowy program:

### 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. Automatyzacja

- 2.1. Automatyzacja budowania projektu przy użyciu Maven / Ant / Gradle (do wyboru)
- 2.2. Wykorzystanie serwera Continuous Integration
- 2.3. Optymalizacja procesu budowania pod kątem częstszych releasów

### 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. Testowanie regresyjne

- 5.1. Sposoby testowania systemu
- 5.2. Rodzaje testów i przykłady ich wykorzystania

5.3. Automatyzacja procesu testowania

5.4. Wybór strategii testowania w projekcie

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