

## Program szkolenia:

# Narzędzia profesjonalnego zespołu developerskiego

### Informacje:

<b>Nazwa:</b>	<b>Narzędzia profesjonalnego zespołu developerskiego</b>
<b>Kod:</b>	<b>Java-Java Tools</b>
<b>Kategoria:</b>	Java i JVM
<b>Grupa docelowa:</b>	developerzy
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

---

Szkolenie przedstawia spójny zestaw narzędzi dla całego zespołu.

Posługiwanie się jednolitym i sprawdzonym przez profesjonalistów zestawem narzędzie skutkuje drastycznym zwiększeniem produktywności widocznym niemal natychmiast.

Prezentowany zestaw narzędzi pokrywa wszystkie aspekty procesu developerskiego: zarządzania zadaniami, monitorowanie jakości, zarządzanie kodem

### Zalety szkolenia:

- Pragmatyczne podejście
- Realne zastosowania
- Kompleksowe podejście

## Szczegółowy program:

### 1. Maven (produktywność, standardy, jakości)

#### 1.1. Standardowa struktura projektu

##### 1.1.1. Zalety

##### 1.1.2. Wielomodułowe projekty Java EE

##### 1.1.3. Dobór strategii zależności pomiędzy artefaktami

##### 1.1.4. Hierarchia artefaktów

#### 1.2. Cykl budowania

##### 1.2.1. Najlepsze praktyki

##### 1.2.2. SNAPSHOT i Release wersji

#### 1.3. Zależności (Zarządzanie i rozwiązywanie konfliktów)

#### 1.4. Repozytoria - Repozytorium na poziomie organizacji

#### 1.5. Integracja z Eclipse - M2 plugin

### 2. Zarządzanie kodem - Git

#### 2.1. Eclipse/InteliJ, konsola

#### 2.2. Codzienna praca i flow

### 3. Jakość

#### 3.1. Checkstyle - automatyczna weryfikacja zgodności ze standardami

#### 3.2. Metryki

#### 3.3. Narzędzia wspierające Test Driven Development

##### 3.3.1. Zagadnienia podstawowe (scenariusze wynikające z Use Case, teoria)

##### 3.3.2. Zagadnienie otwartości kodu na testy (Wstrzykiwanie zależności, wzorce projektowe, architektura sprzyjająca testom)

##### 3.3.3. Testy jednostkowe i integracyjne

##### 3.3.4. Techniki testowania (Mock Objects, Stubs, Fakes)

3.3.5. Praktyczne wykorzystanie narzędzi: Junit, TestNg, EclEmma (kontrola pokrycia kodu), Selenium – różne podejścia do testowania GUI

#### **4. Śledzenie zdarzeń w systemie**

4.1. Logowanie przy pomocy Log4j i fasady SLF4J

4.2. Optymalizacja logów

4.3. Efektywne przeglądanie logów (chainsaw)

#### **5. Integracja całości na platformie Continuous Inegration**

#### **6. Praca zadaniowa**