

## Program szkolenia:

# Jenkins as code - automatyzacja tworzenia jobów i pipeline'ów w kodzie z wykorzystaniem Job DSL

## Informacje:

<b>Nazwa:</b>	<b>Jenkins as code - automatyzacja tworzenia jobów i pipeline'ów w kodzie z wykorzystaniem Job DSL</b>
<b>Kod:</b>	<b>job-dsl</b>
<b>Kategoria:</b>	Narzędzia
<b>Odbiorcy:</b>	developerzy, DevOps, admini
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	40% wykłady / 60% warsztaty

Dowiedz się, jak zbudować zarządzane w kodzie skalowalne środowisko CI/CD z wykorzystaniem Jenkinsa i Job DSL.

Celem szkolenia jest:

- nabycie umiejętności umożliwiających automatyzację tworzenia jobów i pipeline'ów w Jenkins (Continuous Integration i Continuous Delivery)
- zwiększenie niezawodności i skrócenie czasu wdrażania aplikacji (krótszy time to market)
- zdobycie wiedzy z zakresu migracji istniejących jobów do Jenkins Job DSL
- solidne przećwiczenie w praktyce zarządzania jobami Jenkins w kodzie

**Uwaga.** Szkolenie zakłada podstawową wiedzę grupy szkolonej w zakresie Continuous Integration oraz Jenkinsa. W przeciwnym razie zalecane jest skorzystanie z 3-dniowego szkolenia zawierającego dodatkowo wprowadzenie do Jenkinsa oraz całego ekosystemu związanego z CI: [Jenkins - Continuous Integration](#)

## Zalety szkolenia:

- przekrojowe ćwiczenia praktyczne
- zalecane wzorce i praktyki

## Szczegółowy program:

### 1. Ciągła integracja i zarządzanie w kodzie

1.1. założenia

1.2. potrzeba

1.3. korzyści

### 2. Jenkins jako serwer ciągłej integracji

2.1. funkcje i zadania

2.2. mechanizmy konfiguracji

2.3. rozszerzalność przez pluginy

### 3. Wprowadzenie do Groovy - języka pisania definicji zadań w Jenkinisie

3.1. podstawowa składania

3.2. tworzenie i zarządzanie kolekcjami

3.3. konfiguracja z domknięciami (Closures)

3.4. wsparcie dla XML i JSON

3.5. dobre praktyki

### 4. Jenkins Job DSL - wprowadzenie

4.1. architektura

4.2. DSL playground

4.3. API viewer

4.4. seed jobs

### 5. Jenkins Job DSL - zagadnienia praktyczne

5.1. tworzenie jobów

5.2. wykorzystywanie pluginów Jenkinsa

5.3. tworzenie widoków

5.4. rozszerzanie mo'zliwo'sci Job DSL z configuration closure

5.5. migracja istniej'acych job'ów do Job DSL

## 6. Delivery Pipeline

6.1. wyj'scie poza Continuous Integration

6.2. orkiestracja job'ów

6.3. wizualizacja pipeline w Jenkinsie

6.4. przyk'ladowe podej'scie do Delivery Pipeline dla microserwis'ów

6.5. tworzenie w'lasnego frameworka na bazie Job DSL

## 7. Automatyczne testowanie wygenerowanych job'ów

7.1. potrzeba automatycznego testowania

7.2. sanity testing

7.3. testowanie jednostkowe z mockami

7.4. testy regresji

7.5. testowanie integracyjne ze stawianym na z'ądanie serwerem Jenkinsa

7.6. dobre praktyki

## 8. Jenkins Pipeline - alternatywa dla Job DSL

8.1. podstawowe za'oz'enia

8.2. mocne strony

8.3. rekomendowane zastosowania

8.4. ograniczenia w stosunku do Job DSL

## 9. Oskryptowywanie zada'ń administracyjnych

9.1. administracyjne czynno'sci niewygodne do r'ęcznego wykonywania

9.2. Groovy console

9.3. osadzanie cyklicznych zada'ń administracyjnych w jobach

9.4. bezpiecze'ństwo