

## Program szkolenia:

# Pomiary i monitorowanie JVM

### Informacje:

<b>Nazwa:</b>	<b>Pomiary i monitorowanie JVM</b>
<b>Kod:</b>	<b>Java-EE-jvm</b>
<b>Kategoria:</b>	Java EE
<b>Grupa docelowa:</b>	architekci
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	30% wykłady / 70% warsztaty

---

Czy jesteś w stanie powiedzieć co w danym momencie dzieje się w Twojej aplikacji? Jak wyglądają czasy odpowiedzi API, ile czasu trwa proces biznesowy? Czy potrafisz uzasadnić dobór krytycznych parametrów technicznych aplikacji typu rozmiar pul wątków przez pokazanie ich użycia?

Dzięki temu warsztatowi dowiesz się jak wykorzystać metryki do znalezienia odpowiedzi na wszystkie powyższe pytania. Na przykładzie narzędzi dostępnych w OpenSource pokażę jak zacząć zbierać metryki oraz zbudujemy załączek architektury systemu wizualizacji metryk, monitoringu i powiadamiania o anomaliach.

Warsztat kierowany jest zarówno do programistów którzy chcą lepiej rozumieć co dzieje się w ich aplikacjach jak i architektów, którzy chcą dowiedzieć się jak zbudować i wykorzystać w praktyce ekosystem do agregacji i monitoringu metryk.

### Zalety szkolenia:

- Kompleksowe podejście do narzędzi
- Najlepsze praktyki i wzorce
- Dostęp do ekspertów posiadających doświadczenie w projektach dużej skali

## Szczegółowy program:

### 1. Czym są metryki?

1.1. Co warto mierzyć i dlaczego

1.2. Konkretne przykłady z realnych systemów

1.2.1. Kiedy pomogy zapobiec awarii

1.2.2. Kiedy poinformowały o awarii lub pomogły rozwikłać czemu doszło do awarii.

### 2. Metryki w ekosystemie IT

2.1. Strategiczne spojrzenie na umiejscowienie narzędzi powiązanych z metrykami w ekosystemie IT

2.2. Jakie komponenty są konieczne a jakie dają dodatkową wartość

2.3. Krótki opis każdego z komponentów wraz z przykładowymi implementacjami w OpenSource.

### 3. Metryki w JVM - `io.dropwizard.metrics`

3.1. Przedstawienie biblioteki: obiekt `MetricRegistry`

3.2. Typy metryk wraz z konkretnymi przykładami zastosowań - kiedy wybrać: `Timer`, `Counter`, `Gauge`, `Meter`

### 4. Pomiary kodu biznesowego

4.1. Integracja `MetricRegistry` ze Spring

4.2. Mierzenie poprzez adnotacje

4.3. Mierzenie czasów odpowiedzi w API

### 5. Metryki w ekosystemie: agregacja i wizualizacja

5.1. Wprowadzenie do zagadnienia agregacji i przechowywania metryk

5.2. Przedstawienie najważniejszych istniejących narzędzi

5.3. Uruchomienie Graphite + Graphana na Dockerze do dalszej pracy

5.4. Wpięcie kodu odpowiedzialnego za wysyłanie metryk do Graphite

### 6. Pomiary kodu technicznego

6.1. Co mierzyć przy komunikacji z innymi systemami?

6.1.1. Mierzenie klientów HTTP

6.1.2. Mierzenie interakcji z bazami danych

6.2. Pomiary pól połączeń

6.3. Pomiary pól wątków

## 7. Pomiary JVM

7.1. Metryki JVM

7.2. Wpicie biblioteki do ich zbierania

7.3. Wyjaśnienie znaczenia każdej z metryk

## 8. Metryki w ekosystemie: wizualizacja metryk 2

8.1. Funkcje w Graphite - prezentacja najpopularniejszych

8.2. Wykorzystanie funkcji Graphite do lepszej wizualizacji danych

8.3. Zaawansowane funkcje Graphany - templates

## 9. Metryki w ekosystemie: alertowanie

9.1. Po co nam alertowanie?

9.2. Narzędzie: Cabot

## 10. Pomiary kodu legacy

10.1. Pobieranie podstawowych metryk bez modyfikacji kodu

10.2. Narzędzie: jmxtrans

## 11. Metryki w ekosystemie: metryki systemowe

11.1. Narzędzie: Diamond

## 12. Metryki - pitfalls

12.1. Najczęstsze błędy przy interpretacji metryk

12.2. Kiedy metryki kłamią?

12.3. Technicznie - wpływ metryk na aplikacje