

# Tytuł szkolenia: DDD i Mikroserwisy

## Kod szkolenia: DDD-mikroserwisy

### Wprowadzenie

Wiedza na temat DDD jest niesamowicie pomocna przy tworzeniu systemów opartych o architekturę mikroserwisów. Pozwala stworzyć rozwiązanie skalowalne, autonomiczne i odporne na zmiany. Mikroserwisy powinny dostarczać konkretne możliwości biznesowe. Jak robić to dobrze? Tego uczy nas Domain Driven Design. **Naucz się tworzyć wydajne mikroserwisy!**

### Adresaci szkolenia

Szkolenie skierowane jest głównie do: Architektów, Developerów, DevOpsów.

#### UWAGA!

Dla grup zamkniętych istnieje możliwość dostosowania programu oraz ilości dni szkolenia według potrzeb uczestników.

### Cel szkolenia

Dzięki szkoleniu dowiesz się, w jaki sposób wykorzystać wiedzę związaną z DDD, by umiejętnie świadczyć swoje usługi. Dowiesz się, jak wzorce strategiczne pozwolą na stworzenie systemu, który działa autonomicznie, z jasno określonymi przypadkami użycia i właścicielstwem danych. Ponadto, zrozumiesz zyski i koszty wdrażania danej architektury, zdobędziesz umiejętność poprawnego zamodelowania mikroserwisu, dowiesz się, jak integrować / testować / wizualizować / monitorować mikroserwisy oraz uzyskasz kompetencje z zarządzania transakcjami w świecie mikroserwisów.

### Czas i forma szkolenia

- 28 godzin (4 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

### Plan szkolenia

#### 1. Podstawy mikroserwisów

- Jakie są zasady architektury mikroserwisowej
- Kiedy stosować tę architekturę
- Kiedy nie stosować / wystarczy zwykły modułowy monolit

#### 2. Modelowanie serwisów

- Podstawy Domain Driven Design
- Bounded Context a mikroserwis
- Podstawy modelowania
- Strategie dekompozycji mikroserwisów
- Zasada pojedynczej odpowiedzialności mikroserwisów
- Reużywalność

#### 3. Loosely-Coupled Architecture

- Czym jest architektura luźno ze sobą związana i jakie przynosi zyski
- Miary spójności
- Miary złączenia
- Miara częstotliwości zmian

- Trade-offy
- Mikroserwisy a Loosely-Coupled Architecture

#### **4. Komunikacja międzyserwisowa**

- Metody komunikacji
- Komunikacja synchroniczna
- Komunikacja asynchroniczna
- Dobre i złe praktyki

#### **5. Integracja międzyserwisowa**

- Strategie integracji
- Określanie odpowiedzialności
- Izolacja mikroserwisów
- Choreografia a orkiestracja

#### **6. Testy w świecie mikroserwisów**

- Rodzaje testów
- Anty wzorzec – testy E2E
- Testy kontraktów
- Testy na produkcji
- Metryki testów

#### **7. Transakcje a mikroserwisy**

- Rozproszone transakcje
- Jak modelować by nie mieć transakcji
- Wzorzec sagi
- Wzorzec process managera

#### **8. Techniczne wzorce**

- REST
- CQRS
- Event Driven Architecture
- Event Sourcing
- API Gateway
- Service Discovery

#### **9. Socjotechniczne wzorce**

- Zespoły programistyczne a mikroserwisy
- Prawo Conway'a
- Zespoły nastawione na aktywności i rezultaty
- Wzorce pracy zespołowej

#### **10. Wizualizacja architektury – Model C4**

- Czym jest model C4
- Context
- Container
- Component
- Code

#### **11. Event Storming – metoda do szybkiego odkrywania i modelowania logiki biznesowej**

- Podstawy techniki
- Modelowanie procesów biznesowych
- Projektowanie aplikacji
- Określanie granic modułów

#### **12. Legacy – transformacja do mikroserwisów**

- Od czego zacząć

- Rozdział obszarów bezstanowych
- Rozdział obszarów stanowych
- Wzorce podziału bazy danych
- Wzorce podmiany aplikacji