

## Program szkolenia:

# Zaawansowane techniki implementacji DDD - Java, .NET, TypeScript

## Informacje:

<b>Nazwa:</b>	<b>Zaawansowane techniki implementacji DDD - Java, .NET, TypeScript</b>
<b>Kod:</b>	<b>ddd-workshop-impl</b>
<b>Kategoria:</b>	Warsztaty eksperckie DDD
<b>Odbiorcy:</b>	architekci, developerzy
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	20% wykłady/80% warsztaty

---

Intensywne, praktyczne szkolenie prowadzone przez uznanego eksperta w dziedzinie Domain-driven Design i Event Sourcing.

Szkolenie skierowane jest do programistów, architektów i liderów technicznych, którzy chcą pogłębić swoje umiejętności w przejściu od modelowania biznesowego do implementacji solidnej, skalowalnej architektury aplikacji.

Szkolenie prowadzone jest w jednym z trzech języków programowania do wyboru: Java, C# lub TypeScript, zależnie od preferencji grupy uczestników.

## Zalety szkolenia:

- Zrozumienie najlepszych praktyk i najczęstszych błędów w implementacji Agregatów,
- Praktyczne zastosowanie zarówno podstawowych (ORM) jak i zaawansowanych techniki spójności danych oraz wzorce integracyjne (outbox, inbox, itd.)
- Zrozumienie kiedy nie, a kiedy używać architektur typu Hexagonal/Ports and Adapters, CQRS /Vertical Slices, Event Driven, Event Sourcing

## Szczegółowy program:

### 1. Od modelu domenowego do architektury aplikacji

1.1. Techniki przejścia od modelu biznesowego/domenowego do architektury oraz struktur w kodzie.

1.2. Zastosowanie technik typu EventStorming Process/Design level, model C4, Mapy Kontekstów do opracowania projektu architektury.

1.3. Dobór stylów architektonicznych do skali problemu, czyli architektura wielowarstwowa kontra CRUD, podział pionowy.

1.4. Odzwierciedlenie podziału biznesowego w podziale architektury.

### 2. Logika biznesowa i spójność danych

2.1. Implementacja wzorców typu Agregat, Decider, Value Object, Entity do zapewnienia spójności procesu biznesowego.

2.2. Techniczna implementacja tranzakcyjności zapisu danych. Omówienie różnic między podejściami typu Repozytoria, ORM, Event Sourcing, bazy dokumentowe, itd.

2.3. Serwisy domenowe, Specyfikacje i inne techniki w zapewnieniu spójności danych między wieloma operacjami biznesowymi.

2.4. Implementacja pełnego przepływu zarówno w architekturze wielowarstwowej, CQRS i innych. Zrozumienie różnic między nimi, kiedy warto, a kiedy nie.

2.5. Testowanie jednostkowe i integracyjne procesów biznesowych.

### 3. Orkiestracja procesów biznesowych

3.1. Praktyczna implementacja wielokrokowych procesów biznesowych. Zrozumienie różnic między Saga, Process Managerem i Choreografią.

3.2. API wewnętrzne, kontra API zewnętrzne. Dlaczego ten podział jest kluczowy do zarządzania coupling.

3.3. Różnice między podziałem logicznym, architektonicznym i technicznym, czyli Architektury Oparte na Zdarzeniach, Messaging, w praktyce.

3.4. Zrozumienie różnic między komendą, zdarzeniem i stanem.

3.5. Zapewnienie gwarancji spójności, kolejność zdarzeń, idempotentność i inne wyzwania przy obsłudze procesów biznesowych.

3.6. Użycie i rodzaje implementacji wzorców integracyjnych: Outbox, Inbox, Request /Reply.

3.7. Zrozumienie driverów architektonicznych przy doborze technologii obsługi wiadomości. Zrozumienie różnic między technologiami typu Kafka, RabbitMQ, kolejki Cloud Native.

3.8. Testowanie procesów biznesowych.

#### 4. DDD na produkcji

4.1. Wersjonowanie, ewolucja kontraktów.

4.2. Dokumentacja implementacji i jej ewolucja w czasie.

4.3. Zapewnienie obserwowalności i śledzenie procesów biznesowych do wykrywania i analizy błędów. Implementacja i analizowanie przy pomocy Open Telemetry.

4.4. Raporty, modele odczytu, w praktyce. Zrozumienie czy Eventual Consistency jest takie straszne, techniki na jego obejście.

4.5. Dobór narzędzi: framework kontra "zrób to sam". Jak dobrać narzędzia pod projekt i zespół.

4.6. Zaawansowane techniki testowania automatycznego. Kiedy, gdzie i w jaki sposób.