

## Program szkolenia:

# Domain Driven Design - zaawansowane techniki modelowania i archetypy (część 3)

## Informacje:

<b>Nazwa:</b>	<b>Domain Driven Design - zaawansowane techniki modelowania i archetypy (część 3)</b>
<b>Kod:</b>	<b>DDD-DDD</b>
<b>Kategoria:</b>	Domain Driven Design i Event Storming
<b>Odbiorcy:</b>	developerzy, architekci, analitycy
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	60% wykłady / 40% warsztaty

Kiedy Agregaty i polityki to zbyt mało aby sprostać poziomowi złożoności. Warsztat jest sumą doświadczeń z dziesiątek projektów doradczych. Znajdziesz w nim to, co nurtuje najlepszych.

Warsztat jest przeznaczony dla uczestników, którzy już praktykują DDD i szukają kolejnych narzędzi oraz wzorców.

Zakładamy biegłość w technikach podstawowych i wprowadzamy rozszerzenia mające na celu stworzenie ekspresywnych i rozszerzalnych modeli dla złożonych problemów biznesowych.

## Zalety szkolenia:

- Skupienie na technikach rozwikłania bounded contextów
- Archetypy modeli biznesowych
- Modelowanie dziedzin abstrakcyjnych, takich jak konfigurowalne silniki
- Gotowe rozwiązania i receptury

## Szczegółowy program:

### 1. Archetypy modeli biznesowych

#### 1.1. Jak wydestylować generyczne pod domeny

1.1.1. Rozplatanie logiki z zastosowaniem zasady Single Source of Truth

1.1.2. Znajdowanie zależności logicznych wiele do wielu

1.1.3. Being vs Behaving vs Becoming

#### 1.2. Archetypy modeli

1.2.1. Party, Party Role i Party Relationship - model organizacji

1.2.2. Definicja produktu, instancja produktu, kanał dystrybucji - model produktów, usług i umów

1.2.3. Obieg dokumentu vs redagowanie dokumentu

1.2.4. Model dostępność - uniwersalny otwieracz do problemów klasy concurrency

1.2.5. Import danych - poczekalnia Inventory

1.2.6. PIM i Inventory

1.2.7. Dynamiczne drzewa reguł

1.2.8. Scoring

1.2.9. Reguły zależne od sekwencji wcześniejszych zdarzeń - technicznie event sourcing

1.2.10. Reguły zmienne w czasie

#### 1.3. Typowe błędy

1.3.1. Oferta to nie zamówienie

1.3.2. Zakup to nie zamówienie

1.3.3. Status to nie otwarcie na rozbudowę

### 2. Wzorce Taktyczne - Building Blocks

2.1. Jak rozdzielać złożoną logikę domenową od CRUD

2.2. Jak modelować dynamicznie definiowane reguły

2.2.1. Modelowanie funkcyjne

2.2.2. Fabryki polityk składających się z dynamicznych funktorów

2.2.2.1. Polityki jako łańcuchy i drzewa

2.3. Modelowanie logiki zmiennej w czasie

2.3.1. Kopiowanie kodu jako wzorzec

2.4. Dodatkowe wzorce - rozszerzenia Building Blocks

2.4.1. Role Object

2.4.2. Extension Object

2.4.3. Sagi - modelowanie złożonych procesów biznesowych

2.4.4. Dekoratory Polityk - Supple Design

2.4.5. Agregat jako maszyna stanów

2.4.6. Łańcuch odpowiedzialności

2.5. Praktyczne przykłady modelowania biznesowego z wykorzystaniem Building Blocks

### 3. Wzorce Strategiczne

3.1. Podejście archetypowe - dwa poziomy modelu

3.1.1. Knowledge Level

3.1.2. Operations Level

3.2. Cztery poziomy modelu dużej skali

3.2.1. Capability

3.2.2. Operations

3.2.3. Policy

3.2.4. Decision Support

### 4. Architektura aplikacji

4.1. Model logiki aplikacyjnej jako dokumentów RESTful

4.1.1. Czwarty poziom dojrzałości REST

4.1.2. Modelowanie dokumentów REST jako komend

4.2. Multi-tenancy - wersjonowanie API dla różnych dżerżawców

4.2.1. Dobieranie command-hander dla tenantID

## 5. Case study

5.1. Wybrane przypadki po uzgodnieniu z grupą