

Wprowadzenie do Domain-Driven Design

Cele szkolenia

Celem szkolenia jest nabycie umiejętności praktycznego wykorzystania wzorców strategicznych oraz taktycznych Domain-Driven Design.

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Wykorzystywać wzorce strategiczne DDD do podjęcia decyzji związanych z architekturą
- Wykorzystywać wzorce taktyczne DDD do utrzymania wysokiej jakości kodu
- Stosować CQRS w celu separacji Data Model od Domain Model

Profil uczestników

Szkolenie przeznaczone jest dla programistów i architektów, którzy chcą nauczyć się jak wykorzystać wzorce DDD do podejmowania lepszych decyzji związanych z architekturą oraz implementacją.

Szkolenie dla programistów, architektów, liderów technicznych.

Przygotowanie uczestników

Udział w szkoleniu nie wymaga szczególnego przygotowania.

Szczegółowy program szkolenia

1. Wprowadzenie do Domain-Driven Design

- Wady i zalety
- Rodzaje i stosowalność DDD

2. DDD, a architektura

- Domain Model, a Read Model
- Command Query Responsibility Segregation

- Przegląd architektur

3. Poznawanie domeny

- Ubiquitous Language
- Wprowadzenie do Event Storming
- Rodzaje Event Storming

4. Wzorce taktyczne DDD

- Modelowanie domeny przy pomocy Aggregate, Entity oraz Value Object
- Dbanie o cykl życia dzięki Factory i Repository
- Domain Service - utrzymywanie spójności domeny
- Event - komunikacja pomiędzy domenami
- Policy - rozszerzanie agregatów
- Specification - modelowanie złożonych reguł

5. Wzorce strategiczne DDD

- Rodzaje domen
- Bounded Context i Context Mappings

6. Testowanie, a DDD

- Wybór odpowiedniego typ testu
- Poprawne korzystanie z Mocków

Metoda realizacji szkolenia

Analiza domeny wykorzystując technikę Event Storming. Podejmowanie decyzji architektonicznych w oparciu o uzyskane informacje i wykorzystując wzorce strategiczne DDD. Implementacja Core Domain przy użyciu wzorców taktycznych DDD.

Liczba dni, liczba godzin szkoleniowych

3 dni, 24 godziny szkoleniowe

Ścieżka rozwoju po szkoleniu

1. Szkolenia:
 - *Event Storming*
2. Książki:
 - [Implementing Domain-Driven Design](#)
 - [Domain-Driven Design Distilled](#)
 - [Domain-Driven Design](#)