

Course code: **CDDD**

Course title: **Designing of complex domain models using the Domain Driven Design**

Days: 3

## Description:

### Course intended for:

The training is offered to programmers, designers and architects, who would like to get familiar with the practical aspects and to exercise designing of complex domain models using the Domain Driven Design.

### Course objective:

The training objective includes getting familiar with and mastering of the following issues associated with DDD:

- DDD modeling techniques and team work
- DDD tactical techniques and patterns
- DDD strategic techniques and patterns
- Techniques of refactoring and use of design patterns
- Architecture in DDD and architectural patterns

### Requirements:

The training participant should have experience in object-oriented programming, as well as basic knowledge of issues associated with analysis and design, including familiarity with UML.

### Course parameters:

3\*8 hours (3\*7 net hours) of lectures and workshops (with a visible emphasis on workshops-80%). During the workshops, the participants design a system and its selected module, getting familiar with the concepts associated with DDD, which are also discussed with the trainer.

## 1. The main assumptions of DDD

- Applicability of DDD
- The process and roles in DDD
- Ubiquitous Language
- Architecture and DDD

## 2. Modeling techniques

- Effective modeling
  - Knowledge crunching
  - Continuous Learning
  - Deep models
  - Model-Driven Design
- Communication and language
  - Ubiquitous Language
  - Modeling session handling
  - Team work techniques

## 3. DDD tactical patterns (Building Blocks)

- Domain isolation in architecture
- Object-oriented model paradigms
- Associations
- entities
- Value objects



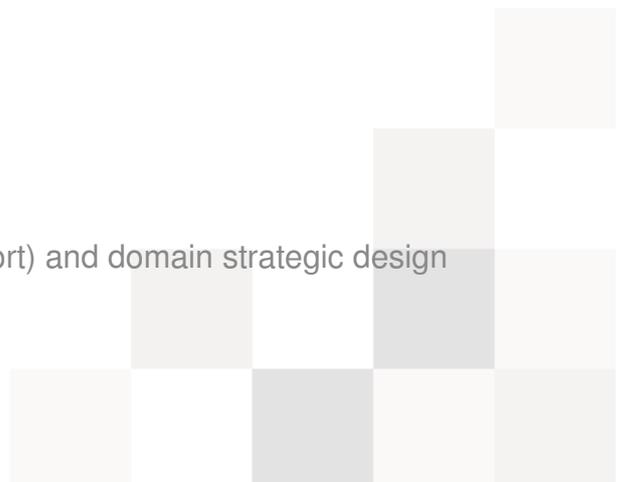
- Services
- Modules
- Aggregates
- Factories
- Repositories and modeling from the perspective of relational databases

## 4. Enhancement and refactoring techniques

- GoF design patterns in the context of DDD
  - Saga as a mediator and facade
  - Specification as a composite
  - Policy/strategy decorator
  - State for aggregates
  - Chain of Responsibility
  - Builder
- Supple Design patterns
  - Intention-Revealing Interfaces
  - Side-Effect-Free Functions
  - Assertions
  - Conceptual Contours
  - Standalone Classes
  - Closure of Operations
  - Declarative programming

## 5. DDD strategic patterns

- Domain types (Core, Generic, Support) and domain strategic design techniques



- Specification of boundaries and integration: Bounded Context and Context Map, Shared Kernel, Anticorruption Layer, Open Host Service
- Inter-team work: Customer/Supplier, Conformist, Separate Ways

## 6. Architecture

- Application architecture and the system architecture
- Strategic design
- Large scale model: Capability, Operations, Policy, Decision Support
- Architectural designs
  - Multilayered Architecture with presentation, application, domain and infrastructure layers
  - CqRS architecture
  - Event-Driven Architecture
  - SOA Architecture

