

Domain Driven Design

Audience Course Domain Driven Design

The course Domain Driven Design is intended for software developers and software architects who want to use DDD in application development.

Prerequisites Course Domain Driven Design

Knowledge of Object Oriented Analysis and Design and design principles such as SOLID and DRY.

Realization Training Domain Driven Design

Presentations by the trainer using slides and demos that are alternated with practical exercises.

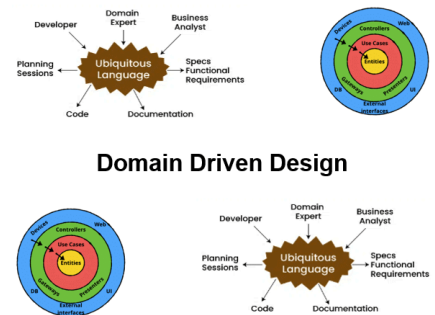
Certificate Domain Driven Design

After successfully completing the course, participants will receive a certificate of participation in Domain Driven Design.

Duration: 2 days

Price: € 1699

[Open Schedule](#)



Content Course Domain Driven Design

In the course Domain Driven Design, participants learn to apply the principles of DDD to the design of applications. The components of the Domain-driven Design model are discussed and knowledge is gained of the DDD methodologies in the design of application architectures.

Intro DDD

This module introduces Domain-Driven Design (DDD) and its key principles. Topics include domains, contexts, benefits and challenges of DDD, interviews with domain experts, and differentiating strategic versus tactical DDD. The differences from traditional design are also discussed.

Domain Understanding

Participants learn to identify domains such as core, supporting, and generic ones. Subdomains and the role of domain experts are explored through event storming, business rules, workflows, and constraints discovery.

Bounded Contexts

This module focuses on understanding bounded contexts and their role in separating different subdomains. Learners explore context maps, service interactions, and relationships such as partnership, shared kernel, customer-supplier, open host service, and anti-corruption layers.

Ubiquitous Language

Participants develop a shared language between developers and domain experts. Topics include message flowing, assigning responsibilities, collaborative modeling techniques like example mapping, and using documents and diagrams in DDD.

Building Domain Models

This module covers the construction of domain models, including entities, value objects, aggregates, and their roots. Learners work with repositories, factories, domain events, and services to capture business behavior and structure the domain.

DDD Application Architecture

This module explores architecture styles used in DDD such as layered and hexagonal architecture. Learners apply CQRS, integrate DDD into microservices and event-driven architectures, and examine connections with BDD. Common pitfalls and anti-patterns are also discussed.

Modules Course Domain Driven Design

Module 1: Intro DDD	Module 2: Domain Understanding	Module 3: Bounded Contexts
What is Domain-Driven Design? Basic concepts of DDD Domains and Contexts Components of DDD Benefits of DDD Challenges in adopting DDD Domain Expert Interviews Knowledge Crunching Strategic and Tactical DDD DDD vs. Traditional Design	Identifying Domains Core Domain Supporting Domain Generic Domains Recognizing Subdomains Role of Domain Experts Event Storming Workshop Exploring Business Rules Exploring Workflows Exploring Constraints	Understanding Bounded Contexts Domain Relationships Purpose of Subdomains Context maps Services Interaction Context Relationships Partnership and Shared Kernel Customer-Supplier Open Host Service Anti-Corruption Layer
Module 4: Ubiquitous Language	Module 5: Building Domain Models	Module 6: DDD Application Architecture
Avoid Miscommunication Developers and Domain Experts Bridging the Gap Defining Ubiquitous Language Message Flowing Assign Responsibilities Collaborative Modeling Example Mapping Documents in DDD Diagrams in DDD	Entity Modeling Value Objects Unique Identity Attribute Role of Aggregates Aggregate Roots Repositories and Factories Domain Events Capturing Business Changes Defining Domain Services Modules in DDD	Layered Architecture in DDD Hexagonal Architecture Ports and Adapters Command Query Segregation CQRS Pattern DDD in Microservices Architecture Event-Driven Architecture Linkage with BDD Common Pitfalls Anti Patterns