

Domain Driven Design

Doelgroep Cursus Domain Driven Design

De cursus Domain Driven Design is bestemd voor software developers en software architecten die DDD willen gebruiken bij applicatie ontwikkeling.

Voorkennis Cursus Domain Driven Design

Kennis van Object Oriented Analysis and Design en design principes zoals SOLID en DRY.

Uitvoering Training Domain Driven Design

Presentaties door de trainer aan de hand van slides en demos die worden afgewisseld met praktijk oefeningen.

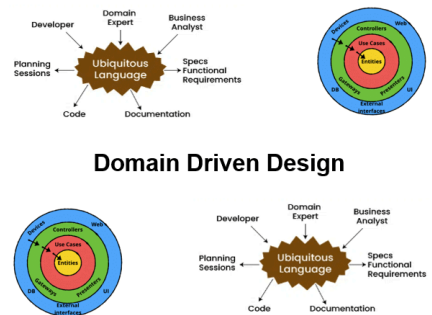
Certificaat Domain Driven Design

De deelnemers krijgen na het goed doorlopen van de cursus een certificaat van deelname aan Domain Driven Design.

Duur: 2 dagen

Prijs: € 1699

Open Rooster



Inhoud Cursus Domain Driven Design

In de cursus Domain Driven Design leren de deelnemers de principes van DDD toe te passen bij het ontwerp van applicaties. De componenten van het Domain-driven Design model worden behandeld en kennis wordt opgedaan van de DDD methodieken bij het ontwerp van applicatie architecturen.

Intro DDD

Deze module introduceert Domain-Driven Design (DDD) en de kernconcepten. Onderwerpen zijn onder meer domeinen, contexten, voordelen en uitdagingen bij implementatie. Ook worden strategische versus tactische DDD benaderingen behandeld, evenals het verschil met traditionele ontwerpmodellen.

Domeinbegrip

Deelnemers leren hoe domeinen worden geïdentificeerd, inclusief kern-, ondersteunende en generieke domeinen. Subdomeinen en de rol van domeinexperts komen aan bod via event storming, het ontdekken van bedrijfsregels, workflows en beperkingen.

Bounded Contexts

Deze module behandelt het concept van bounded contexts en hoe ze de structuur van domeinen bepalen. Er wordt gewerkt met context maps, relaties tussen services en patronen zoals partnership, shared kernel, open host services en anti-corruption layers.

Ubiquitous Language

Er wordt gefocust op het ontwikkelen van een gemeenschappelijke taal tussen ontwikkelaars en domeinexperts. Methoden zoals message flowing, voorbeeldmapping, en het gebruik van documenten en diagrammen binnen DDD worden besproken.

Opbouw van Domeinmodellen

In deze module leren deelnemers entiteiten, value objects, en aggregates te modelleren. Er wordt gewerkt met repositories, factories, domeinevents en services. De nadruk ligt op het vastleggen van zakelijke veranderingen en modulair ontwerp.

DDD Applicatiearchitectuur

De architectuurprincipes van DDD worden hier behandeld, inclusief layered en hexagonal architecture. CQRS, event-driven design, en de relatie met microservices en BDD worden besproken. Deelnemers leren ook veelvoorkomende valkuilen en anti-patterns vermijden.

Modules Cursus 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