



Domain Driven Design

Durée 3 jour(s) (DDD-03)

Conception Orientée Domaine Métier

Description

La conception de logiciels portant sur un domaine métier complexe se heurte très souvent aux mêmes problèmes récurrents : - Code fragile et rigide, qui vieillit mal, coûteux en maintenance, et difficile à faire évoluer - Transmission difficile ou impossible des connaissances, surtout dans un contexte de turn-over régulier - Absence de capitalisation sur la connaissance du métier - Perte de crédibilité et de confiance dans l'application L'approche DDD propose de résoudre ces problèmes en attaquant de front cette complexité : le modèle du domaine est le noyau du logiciel, que ce soit du point de vue de l'architecture, du nommage des composants, ou de l'effort apporté. Cette formation expose les concepts essentiels du DDD. Son fil rouge est l'amélioration d'une conception existante, au fur et à mesure qu'on introduit des briques DDD de plus en plus avancées.

Objectifs

- Eviter les problèmes récurrents provoqués par des conceptions centrées sur la donnée ou la technique
- Mettre sous contrôle la dette technique IT
- Concevoir des logiciels maintenables, évolutifs, et capitalisables
- Utiliser une approche de conception pragmatique et éprouvée
- Homogénéiser la conception avec le vocabulaire des patterns du DDD
- Valider une conception ou une réalisation selon des critères simples
- Parler un langage unifié entre exigences, conception, et implémentation : l'Ubiquitous Language
- Marier l'approche DDD et l'Agilité
- Envisager le DDD dans le contexte des architectures Microservices, Event sourcing/CQRS, NoSQL, REST

Public

- Développeurs
- Architecte
- Analyste
- Chef de projet

Prérequis

- Expérience en programmation orientée objet

Répartition

50% Théorie, 50% Pratique

Programme

De la conception centrée sur la technique à la conception centrée sur le métier

- Domaine métier et modélisation
- Problèmes de conception: les « usual suspects », causes et conséquences
- Les propositions essentielles du DDD
- Présentation du fil rouge : l'application de réservation de salle

- Equipes amont et aval : les patterns de Context Mapping
- Implémenter : les Styles d'Intégration

Tour d'horizon : DDD et architecture

- CQRS et Event Sourcing
- DDD et NoSQL
- DDD et Microservices
- DDD et REST

Les concepts métier au cœur de la conception : les briques du DDD tactique

- Mettre sous contrôle la dette technique en utilisant les briques simples du DDD : Value Objects et Entités
- Démarquer nettement le code métier des frameworks d'interaction (web, ..) avec les Application Services
- Eviter la duplication et la dispersion de la logique métier avec les Domain Services
- Eviter la pollution du code métier par les problématiques de persistance et d'infrastructure avec les Repositories et les Infrastructure Services

Maîtriser la complexité : le DDD tactique avancé

- Connaître la définition et les implications de Transactional Consistency et Eventual Consistency
- Maîtriser la complexité induite par les graphes complexes d'associations entre objets, avec les Agrégats
- Découper le logiciel en Modules métier
- Séparer les préoccupations avec les Domain Events
- Les Analysis Patterns: des modèles de domaine réutilisables

Intégrer des logiciels et des équipes : le DDD Stratégique

- L'Architecture Hexagonale: le Domain Model comme noyau du logiciel