



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.

Formation animée en présentiel

La formation en présentiel se déroule sur des jours consécutifs

Formation disponible en mode "formation à distance"

La formation à distance se déroule de préférence sur des jours consécutifs (contactez nous si besoin de décomposer en demies journées) En inter-entreprises, l'outil de visio-conférence privilégié est Microsoft Teams

En intra-entreprises, on privilégie Zoom mais Microsoft Teams est également proposé

Objectifs

- Identifier et décrire les différents concepts du Domain Driven Design.
- Reconnaître et expliquer l'opportunité de la mise en oeuvre de l'approche DDD.
- Déterminer et développer un Event Storming.
- Distinguer et interpréter le vocabulaire lié au DDD.
- Reconnaître une architecture hexagonale pour en construire le schéma.
- Pratiquer cette architecture hexagonale pour développer un projet.

Public

- Profils ayant une expérience en programmation orientée objet
- Développeurs
- Architecte
- Pour les profils Chef de projet, Analyste ou Expert métier nous vous proposons la formation [DDD pour expert métier](#)

Prérequis

Aucun

Répartition

50% Théorie, 50% Pratique

Evaluations des acquis

L'évaluation des acquis de la formation se fera en séance au travers d'ateliers, d'exercices et/ou de travaux pratiques. Dans le cas d'une formation officielle éditeur, veuillez nous consulter afin que nous vous fassions part des modalités d'évaluation.

A l'issue de la formation, vous sera transmis une évaluation à chaud de l'action de formation qui vous permettra de nous faire part de vos retours quant à votre expérience apprenant avec Zenika.

Ressources pédagogiques

Les ressources pédagogiques proviennent de productions des équipes Zenika et/ou de la documentation éditeur dans le cas d'une formation "Officielle". Les documents sont en français ou en anglais.

RQTH et ma formation Zenika

Si vous êtes sujet à un handicap, prenez contact avec nos équipes pour que nous puissions définir ensemble comment nous pourrions aménager la session afin que vous puissiez vivre une expérience en formation inchangée.

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

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
- 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