DDD : Domain-Driven Design

Acquérir les pratiques d'une conception logicielle orientée métier

DESCRIPTION

Dans de nombreux projets, on se précipite souvent sur le choix des solutions (frameworks, stacks techniques, data stores) avant même de comprendre pleinement les besoins. Pourtant, le logiciel est avant tout un outil pour résoudre efficacement des problèmes métier. Ceux qui ont expérimenté des techniques de découverte, comme l'Event Storming, se demandent vite comment passer des post-its au code.

La conception pilotée par le domaine (Domain-Driven Design, ou DDD) est une approche méthodologique de conception logicielle qui privilégie la modélisation des systèmes en fonction de la complexité des domaines métiers. Elle est particulièrement adaptée aux projets de développement logiciel complexes, car elle permet d'aligner étroitement la structure du code avec les besoins réels de l'organisation.

Lors de cette formation de deux jours, nous explorerons l'essentiel du Domain-Driven Design, des concepts fondamentaux au cycle complet de développement logiciel.

OBJECTIFS PEDAGOGIQUES

- Découvrir l'Event Storming Big Picture
- Appréhender le découpage en sous-domaines métier
- Expliquer la projection des sous-domaines
- Clarifier le système de valeurs des Bounded Contexts
- Exposer le Context Mapping
- Découvrir l'Event Storming Software Design
- Expérimenter la modélisation de l'agrégat
- Préciser le spectre et les limites du Model-Driven Design
- Identifier l'intérêt du Supple Design
- Comprendre le concept de Deep Modeling et sa relation avec le Supple Design

PUBLIC CIBLE

- Développeur
- Architecte
- Chef de projet
- Tech lead
- Scrum master

Stage pratique
Domain-Driven Design

Code :
FXDDD

Durée :
2 jour(s) (14,00 heures)

Exposés : **20 %**
Cas pratiques : **70 %**
Echanges d'expérience : **10 %**

Inter-entreprises :
Prochaines sessions disponibles sur notre site web.
Tarif : 1 750,00 € HT / participant

Intra-entreprise :
Tarifs et dates sur demande.

PRE-REQUIS

- Connaissance en Java (JDK 25 ou +) et C# (.NET 9 ou +) pour les profils développeurs.
- Savoir coder en Java ou en C# est un impératif.

METHODE PEDAGOGIQUE

Formation pratique, visant à l'acquisition d'un savoir-faire, basée sur un cas pratique ainsi que des échanges et retours d'expérience pratique du formateur. Une session d'Event Storming sera proposée aux participants pour s'approprier les problématiques métier, et suivre une implémentation (en binômes ou tous ensemble).

PROFIL DES INTERVENANTS

Cette formation est dispensée par un·e ou plusieurs consultant·es d'OCTO Technology ou de son réseau de partenaires, expert·es reconnus des sujets traités.

Le processus de sélection de nos formateurs et formatrices est exigeant et repose sur une évaluation rigoureuse leurs capacités techniques, de leur expérience professionnelle et de leurs compétences pédagogiques.

MODALITÉS D'ÉVALUATION ET FORMALISATION À L'ISSUE DE LA FORMATION

L'évaluation des acquis se fait tout au long de la session au travers des ateliers et des mises en pratique.

Afin de valider les compétences acquises lors de la formation, un formulaire d'auto-positionnement est envoyé en amont et en aval de celle-ci.

En l'absence de réponse d'un ou plusieurs participants, un temps sera consacré en ouverture de session pour prendre connaissance du positionnement de chaque stagiaire sur les objectifs pédagogiques évalués.

Une évaluation à chaud est également effectuée en fin de session pour mesurer la satisfaction des stagiaires et un certificat de réalisation leur est adressé individuellement.

PROGRAMME PEDAGOGIQUE DETAILLE

Jour 1 : Explorer l’espace du problème

OUVERTURE DE SESSION

Accueil des stagiaires et tour de table des attentes

Présentation du déroulé des deux journées de formation

INTRODUCTION

Définition d'un modèle métier

Introduction au contexte métier

DÉCOUVERTE DU MÉTIER

EventStorming Big Picture

- Motivations métier pour notre cas pratique

Core Domain Chart

- Priorisation des domaines visant à soutenir les objectifs stratégiques

Context Mapping

- Modélisation et cartographie des modèles et de leurs relations

EventStorming Software Design

- Modélisation au sein de Bounded Context Coeur

Jour 2 : Model-Driven Design

Modélisation de l'agrégat de métier

- Example Mapping
- Responsibility-Driven Design

INTRODUCTION AU MODEL-DRIVEN DESIGN

- Présentation et motivation
- Building blocks
- Supple Design
- Deep Modeling

COLLABORER ET PRODUIRE LE CODE COEUR

- Mise en pratique : “Building Blocks”
- Mise en pratique : “Supple Design”

CLÔTURE DE SESSION

Accessibilité

L'inclusion est sujet important pour OCTO Academy.
Nos référent·es sont à votre disposition pour faciliter l'adaptation de votre formation à vos besoins spécifiques.
Pour les contacter : academy.accessibilite@octo.com