

## UML 2 - Analyse et conception

Utiliser le formalisme UML 2 dans une démarche de développement logiciel

 Présentiel ou en classe à distance



4 jours (28 h)

Forfait intra : 6.550,00 € HT

Réf.: OB210

Le monde du développement logiciel est aujourd'hui en quête d'agilité. Il s'agit souvent de mettre en oeuvre une méthode de gestion de projet agile, favorisant le développement rapide d'un logiciel répondant aux besoins des utilisateurs. Mais qu'en est-il de l'agilité du logiciel lui-même, c'est-à-dire de sa capacité à s'adapter aux nouveaux besoins ? Cette agilité là n'est pas adressée par l'agilité de la méthode, mais par l'application de bonnes pratiques de conception logicielle au niveau du code lui-même. Que le projet adopte une méthode agile ou non, les choix structurants de conception du code doivent être partagés par tous les développeurs. Ces choix doivent donc être définis et formalisés de façon claire et rigoureuse.

### A qui s'adresse cette formation ?



#### Pour qui

- Développeurs
- Architectes logiciels
- Analystes
- Chefs de projets...



#### Prérequis

- Connaissance d'un langage de programmation orienté objet

### Programme

#### 1 - Introduction et prise en main du sujet

- Processus de développement : gestion de projet en cascade vs agile - la place des activités de spécification et de conception
- Conception Orientée Objet : séparation des préoccupations et limitation des dépendances - introduction à UML

#### 2 - Définir le système à réaliser

- S'approprier le métier : au travers de l'étude de cas, comprendre les processus métier (diagramme d'activité) et les informations du domaine (diagramme de classe)
- Spécifier les exigences - au travers de l'étude de cas, définir et formaliser dans un document lisible et rigoureux - Les exigences fonctionnelles : identifier les acteurs et les cas d'utilisation du système - Les exigences non fonctionnelles à l'aide de l'approche FURPS - Le détail des interfaces du système
- Prendre en main le plan de développement projet : objectifs et planification des différentes itérations/sprints de l'étude de cas

#### 3 - Concevoir le système logiciel - Les fondamentaux

- Définir la plate-forme technique : définir l'architecture matérielle (diagramme de déploiement), choisir le framework logiciel
- Concevoir un code source répondant aux exigences, maintenable et évolutif
- Définir une architecture du code : le pattern en couches MVC, étendu au système entier
- Concevoir les attributs : attributs identifiants et dérivés - association entre classes (diagramme de classe)
- Concevoir les traitements et la communication entre classes (diagramme de séquence) : utiliser les scénarios de cas d'utilisation - répondre aux exigences fonctionnelles, séparer les préoccupations selon MVC
- Affiner la structuration du code source : affiner la structuration en packages (diagramme de packages), factoriser du code avec la généralisation - du bon usage de l'héritage, faire communiquer les classes en limitant les dépendances : utilisation des interfaces et des singletons - pattern de communication requête/notification, gérer les états (diagramme d'états)
- Concevoir les composants déployables : définir les composants et leurs interfaces (diagramme de composant), définir le déploiement des composants

(diagramme de déploiement)

#### 4 - Pour aller plus loin

- Compléments sur les processus itératifs/agiles
- Le processus TUP (Two Tracks Unified Process) et l'approche MDA (Model Driven Architecture)
- Mettre en place UML dans l'établissement



#### Les objectifs de la formation

- Découvrir les différentes phases de la modélisation objet en UML
- Savoir traduire un besoin fonctionnel en s'appuyant sur les diagrammes UML
- Appréhender UML de façon opérationnelle sur toutes les étapes du projet de développement
- Maîtriser les bonnes pratiques de conception logicielle fondamentale



#### Evaluation

- Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.



#### Les points forts de la formation

- Une présentation des meilleures pratiques de conception d'architecture logicielle (architecture en couches MVC, patterns de communication, la structuration en packages, ...) avec UML 2.
- Étude de cas complète depuis la modélisation du métier jusqu'à la conception, avec le modèleur Enterprise Architect (autre modèleur possible sur demande).
- Une progression pédagogique très opérationnelle : Les diagrammes sont présentés pour chacune des étapes d'un projet de développement géré de façon agile ou non.
- 85% des participants à cette formation se sont déclarés satisfaits ou très satisfaits au cours des 12 derniers mois.