Programme détaillé
formation 'Conception UML'- 3j

Objectifs : fournir connaissances théoriques et pratiques permettant de concevoir un logiciel avec UML

Audience : Développeurs, chefs de projets .

Prérequis : aucun .

Moyens pédagogiques : 1 ordinateur/stagiaire. Supports cours. Travaux pratiques. Vidéoprojecteur. Tests

Durée : 3 jours (total 21h)

Vous allez apprendre à

- Comprendre le rôle d'UML dans un projet informatique
- Identifier la valeur ajoutée d'UML pour MOE et MOA
- Comprendre les spécification de l'organisation OMG
- Documenter vos projets avec UML
- Capturer les besoins utilisateurs
- Lire les 13 diagrammes UML 2.4
- Créer des diagrammes statiques
- Créer des diagrammes dynamiques
- Associer les bons diagrammes aux phases d'un projet
- Identifier les diagrammes clés : use cases, classes, activité

Cours détaillé formation UML

Introduction UML

- Se libérer d'un langage de programmation
- Approche MDA
- Consortium OMG
- Role d'UML dans un projet
- Différence Modèle / Processus
- Historique UML : unifier 3 méthodes
- UML pour un client
- UML mour un chef de projet
- UML pour un développeur (MOE)
- UML pour maîtrise d'ouvrage (MOA)
- UML : un langage graphique pour documenter
- Extensions UML : stéréotype, profils, contraintes, etc.
- Les 13 diagrammes d'UML 2

UML et langages

- consortium OMG
- Approche MDA
- PIM, PSM
- Générateurs de code

UML et processus agiles
- RUP (Ratinal Unified Process)
- XP (eXtrem Programming)
- SCRUM

Approche Objet
- Objets : identité, état et comportement.
- Popularité, avantages de l'Orienté Objet.
- Abstraction, encapsulation, classification.
- Classes et instances. Classes abstraites.
- Héritage. Interfaces Polymorphisme.
- Surcharge et redéfinition.

Notations communes aux 13 diagrammes
- Commentaire
- Contraintes
- Stéréotype
- Libellé

Diagramme de cas d'utilisation
- Quand l'utiliser ?
- Use case et analyse détaillée
- Acteurs, Interactions
- frontières du système
- Granularité
- Inclusions, Extensions
- Fiche détaillée
- Préconditions, postconditions
- Contraintes
- Chemin nominal
- Alternatives
- Exceptions

Diagramme d'activité
- Quand l'utiliser ?
- Lien avec use cases
- Etat initial
- Etat final
- transitions
- Actions
- flux d'activité

Diagramme de classe
- Quand l'utiliser ?
- Association, multiplicités,
- Rôles,
- Généralisation,
- Agrégation, composition
- Visibilité
- Interfaces
- Packages

Diagramme de séquence
- Quand l'utiliser ?
- Notion de séquence
- Ligne de vie
- Acteurs
- Message synchrone
- Message asynchrone

Diagramme de package
- Quand l'utiliser ?
- Organisation logique
- Organisation physique
- Organisation projet

Diagramme d'Objet
- Quand l'utiliser ?
- Photo des instances d'objet
- Performances
- valeur ajoutée

Diagramme de composants
- Quand l'utiliser ?
- Qu'est ce qu'un composant ?
- Notion d'artéfact
- Notation
- DLL, JAR
- Composant distribué

Diagramme de déploiement
- Quand l'utiliser ?
- Notation serveur
- Notation Firewall
- Lien avec Composants
- valeur ajoutée

Diagramme de structure composite
- Quand l'utiliser ?
- Lien avec diagrammes composant
- frontières internes et externes
- Interfaces

Outils
- Fonctionalités clés
- Génération code / orm
- Reverse ingeniering
- Documentation
- Outil : Visual Paradigm
- Outil : Enterprise Architect
- Outil : ArgoUML