

Objectifs / Durée de la formation

Durée: 4 jours, soit 28 heures

- Acquérir les meilleures pratiques de modélisation en analyse et en conception
- Apprendre à mettre en oeuvre les modèles UML au sein d'un processus de développement

Participants / Pré-requis

- Chefs de projets, ingénieurs concepteurs et ingénieurs de développement
- Formation initiale ou culture générale en informatique requise

Moyens pédagogiques

- Formateur expert dans le domaine
- Mise à disposition d'un ordinateur, support de cours remis à chaque participant, vidéo projecteur, tableau blanc et paperboard
- Feuille de présence signée en demi-journée, évaluation des acquis tout au long de la formation, questionnaire de satisfaction, attestation de stage

Programme

1. Principe de la modélisation objet

- Rappels et principes de la conception objets
- Évolution des techniques de conception de logiciels

2. Concept de base

- Origine d'UML
- UML standard reconnu
- Les principaux modèles

3. Processus de développement

- UML et le processus de développement
- Description d'un processus générique
- Principes du développement itératif
- Les différentes phases d'un projet

4. Étude préliminaire

- Élaboration du cahier des charges
- Identifier les acteurs, les messages
- Modéliser le contexte

5. Capture des besoins fonctionnels

- Identifier les cas d'utilisation
- Décrire les cas d'utilisation : description textuelle
- Organiser les cas d'utilisation
- Relations entre cas d'utilisation : inclusion - extension
- Identifier les classes candidates
- Valider et consolider

6. Développement du modèle statique

- Affiner les classes et les associations
- Agrégation et composition, multiplicité
- Ajouter les attributs, ajouter les opérations
- Optimiser avec la généralisation

7. Développement du modèle dynamique

- Identifier et formaliser les scénarios
- Construire les diagrammes d'états
- Valider les diagrammes d'états avec les diagrammes d'interactions : diagrammes de séquence, de collaboration
- Confronter les modèles statiques et dynamiques

8. Compléments aux diagrammes d'états

- Transition automatique
- Actions en entrée ou en sortie d'états
- Transition interne, transition propre
- Automates parallèles et hiérarchiques
- Complémentarité entre diagrammes

9. Découpage en package

- Notion de package
- Découpage en packages
- Dépendances entre packages

10. Les diagrammes de déploiements

- Les diagrammes de composants
- Les diagrammes de déploiements
- Notion de package
- Découpage en packages
- Dépendances entre packages