

# UML 2, modélisation avec Enterprise Architect

3 j (21 heures)

Ref : EN-AR

## Public

Architectes logiciel, concepteurs, développeurs

## Pré-requis

Connaissances de base d'UML et d'un langage orienté Objet (Java, C#, C++...)

## Moyens pédagogiques

Formation réalisée en présentiel ou à distance selon la formule retenue  
Exposés, cas pratiques, synthèse, assistance post-formation pendant trois mois  
Un poste par stagiaire, vidéoprojecteur, support de cours fourni à chaque stagiaire

## Modalités de suivi et d'évaluation

Feuille de présence émargée par demi-journée par les stagiaires et le formateur  
Exercices de mise en pratique ou quiz de connaissances tout au long de la formation permettant de mesurer la progression des stagiaires  
Questionnaire d'évaluation de la satisfaction en fin de stage  
Auto-évaluation des acquis de la formation par les stagiaires  
Attestation de fin de formation

## Objectifs

- Maitriser UML dans les différentes activités d'ingénierie logicielle
- Formaliser les exigences sous forme de use cases
- Mettre en œuvre UML avec Enterprise Architect
- Elaborer une conception de qualité par l'utilisation des patterns d'architecture
- Réaliser des modélisation statiques et dynamiques

## Programme détaillé

### INTRODUCTION A LA MODELISATION METIER

---

- Démarche projet et activités d'ingénierie logicielle
- Importance de l'activité de conception
- UML au sein de la démarche projet
- Les différents diagrammes

La notion de stéréotype et de profil

Modélisation du métier (processus métier, diagramme d'activité et diagramme de classes du domaine)

## **SPECIFIER LES EXIGENCES**

---

Exigences fonctionnelle et non fonctionnelle. Approche FURPS

Définition des acteurs

Définition des cas d'utilisation, diagramme de cas d'utilisation

Illustration de scénarios de cas d'utilisation avec le diagramme de séquence

## **MODELISATION STATIQUE**

---

Architecture du code

Patterns en couches/layers

Pattern MVC

Structuration en packages

Identification des classes

Attributs, Opérations, Diagramme de classes

Relations d'association entre classes

Relations de généralisation

## **MODELISATION DYNAMIQUE**

---

Définir les opérations

Scénarios de cas d'utilisation de niveau conception

Description des interactions avec le diagramme de séquence

Gestion des états

Diagramme d'états

## **MODELISATION DU DEPLOIEMENT**

---

La définition des composants déployables et de leurs interfaces

Diagramme de composants

Le déploiement des composants sur l'architecture matérielle

Diagramme de déploiement

## **FONCTIONS AVANCEES DE L'OUTIL**

---

Génération de code et "reverse engineering"

Génération de la documentation

Réalisation de grands projets et utilisation collaborative