



Durée : 3 jours

Nombre de participants : 6 à 12

TARIFS

Inter-Entreprises

Tarif Public : 1 417,50 € H.T.

Tarif Intra-

Entreprise : *Nous consulter***PUBLIC CONCERNÉ****Développeurs, Testeurs, Architectes, Futurs Managers Agiles (Futurs Scrum masters), Responsables qualité/méthodes.****OBJECTIFS**

- Analyser, concevoir, développer et tester des projets logiciels Agiles.
- Gérer des projets logiciels en appliquant des méthodes Agiles et itératives.
- Savoir introduire des tests.
- Intégrer les tests dans le pilotage de projet.

MOYENS PÉDAGOGIQUES

- Diffusion de Powerpoint.

PRÉ-REQUIS

- Vous travaillez en équipe dans un contexte de projet ou de produit.
- Compétences techniques relatives au rôle du candidat.

PROGRAMME**1. Introduction aux spécificités des méthodes agiles :**

- Les spécificités des méthodes agiles :
Le développement incrémental et itératif.
L'incidence sur la politique de tests et son coût.
- Le développement guidé par les tests :
Le cycle de base.
Le principe.
Les trois bénéfices majeurs.
Les règles de simplicité.

2. Architecture et Conception :

- Principes d'architecture dans un environnement Agile.
- Pratique de Conception dans une équipe Agile.
- Principes qui permettent d'amplifier facilement la testabilité et le Refactoring.

3. Test Driven Development - Étude du développement à base de "Test-First" incluant les concepts suivants :

- Définition et Principes.
- Théorie et xUnit.
- Les 3A.
- Gérer les exceptions.
- L'identification des faiblesses du code.
- Les mécanismes d'amélioration.
- L'articulation avec le développement par les tests.
- Rappel sur les patrons de conception (« design patterns »).

4. Refactoring :

- Conception émergente.
- Test Driven Design.
- Quand faut-il refactorer ?
- Comment éviter les grandes dettes techniques.
- Refactoring pour la maintenance.

5. Le cycle de vie des « scénarios clients » :

- Les attributs d'un bon scénario.
- L'articulation entre plan de livraison et d'itération.
- Du scénario au test de recette.
- Le codeur de Maxwell.
- Les techniques de test.
- Les outils de base (JUnit, XML, etc.) et avancés (Fit, etc.).

6. Les meilleures pratiques de Développement Agile :

- Conditions Limites.
- TDD et Gestion de base de données SGBDR.
- TDD et Gestion des données liées aux fichiers et aux repository (SVN, Git, Sourcesafe).
- TDD et Gestion des IHM.

7. Techniques avancées avec le TDD :

- Corriger des anomalies.
- Gérer la montée en charge et la sécurité des produits NTIC.
- Gestion de la sécurité logicielle.
- Gestion de la performance.
- Stress tests.

8. Les objets Mock :

- Mock, Stub et Fake.
- Application de la théorie sans utiliser de bibliothèque.
- Découverte des bibliothèques du marché.

9. Self-Test et Outils collaboratifs :

- Intégration Continue (SVN, CVS, Sourcesafe).
- Intégrateur Continu (Hudson, Cruisecontrol).
- Couverture de test.
- Test Driven Requirement avec Fitness (outil de communication).
- Behaviour Driven Development.