

Programme de formation Test Driven Development (TDD)

Cette formation TDD aborde les bases techniques et opérationnelles de la méthode «Test Driven Development » avec le langage Java comme support

Durée

2 jours / 14 heures

Description

La réalisation logicielle est une pratique qui se cherche encore dans la meilleure façon de produire du code de qualité, et fiable. Depuis quelques années maintenant, une pratique devient prépondérante pour atteindre cet objectif : le TDD ou développement piloté par les tests. TDD permet entre autre, d'avoir un feedback rapide sur son travail, une meilleure confiance dans le résultat produit et facilite à la fois les bonnes pratiques de développement ainsi que le travail en binôme. Cette formation vous en donnera les clefs.

Objectifs pédagogiques

- Comprendre les différents types de tests et leurs enjeux
- Écrire des tests unitaires
- Développer en suivant la méthode TDD
- Appréhender un projet existant avec des tests

Public

Développeurs, Chefs de projet technique

Pré-requis

Connaissance de Java et de l'objet

Méthodes pédagogiques

30 % théorie / 70 % pratique

Profil intervenant

L'ensemble de nos formations sont animées par des formateurs expérimentés possédant une expérience terrain éprouvée.

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers d'ateliers de mise en pratique des notions et concepts abordés pendant la formation.

Programme

Le rôle de développeur

- Objectif du métier
- Dépendance vis à vis des facteurs externes
- Confiance dans son travail

Les tests techniques

- Présentation des différents types de tests techniques
- Structure d'un test unitaire
- Outils et frameworks de tests
- Écriture des premiers tests

Test driven development

- Tester après : constat d'échec
- TDD, qu'est-ce que c'est ?
- Cycle TDD
- Pourquoi ?

Mettre en oeuvre le tdd

- Comment aborder les tests
- Les 4 règles du design simple
- Pair programming
- Mise en pratique guidée pas à pas

Qualité des tests

- De quoi est composé un bon test ?
- Ce qui est inutile de tester
- Mise en pratique façon « Ping-pong programming »
- Communiquer l'intention par les tests

Stratégies de test

- Approche globale ou en détail
- Connue ou inconnue
- Valeur ou rapidité
- Cas nominal ou cas d'erreurs

Les doublures de test

- Les différents types de doublure
- Connue ou inconnue
- Exemple d'utilisation
- Test unitaire versus test d'intégration

Legacy code (code existant généralement non testé)

- Appliquer TDD sur code legacy
- Test de caractérisation

- Identification des blocages
- Isolation
- En pratique

Retrospective