



PLAN DE FORMATION

Hybrid Technologies Solutions

370 route de Puechabon
34380 VIOLS LE FORT

SIRET : 878 640 184 00015

TVA : FR78 878 640 184

Votre contact :

MOMMER Christophe

Tel. 06 35 52 09 61

Email hybrid-technologies-solutions@outlook.com

Formation : Test Driven Development

(avec C# et Visual Studio)

Durée : 2 jours

Du besoin au code

- Définition du besoin : ce qui est important
- Le lien entre le besoin et le code : de la user-story aux tasks
- Découpage des stories : INVEST
- Critères d'acceptation : SMART

Travaux pratiques

Projet factice à planifier

Test unitaires

- Définition
- Structure et organisation d'un test
- Structure d'une classe de test : initialisation, nettoyage, ...
- Avantages des TUs
- Différence entre TU et TI (test intégré)
- Mise en pratique avec Visual Studio

Travaux pratiques

Création de tests unitaires avec Visual Studio sur du code existant

TDD

- Principes et concepts
- Exemple : FizzBuzz
- Tests et qualité
- Dette technique
- Ce que résout la logique TDD

Travaux pratiques

Implémentation de FizzBuzz

ATDD (Acceptance TDD) & BDD (Behaviour DD)

- Comment améliorer le travail en équipe
- Savoir piloter un projet grâce à ces méthodes
- Rendre les intentions claires

Travaux pratiques

Définition d'un projet en ATDD et BDD

Concepts avancés

- Mock, Stub, Fake, Spy
- Styles de TDD : Classic, Chicago, London
- Comment approcher un projet ? Middle-out vs Outside-in

Travaux pratiques

Manipuler un exemple concret pour chaque point ci-dessus

Clean Code

- Pourquoi et comment garder une base code clean ?
- Quelques pratiques : SOLID, YAGNI, DRY, KISS, CQS
- Détecter les mauvaises pratiques communes : STUPID
- L'art du nommage
- Les abstractions et l'injection de dépendance
- Taille du code
- Gestion des erreurs

Travaux pratiques

Petit atelier permettant d'implémenter les principes ci-dessus

Legacy et TDD

- Définition de « Legacy »
- Quelle stratégie adopter ?
- Comment itérer pour sécuriser le fonctionnel ?
- Les meilleurs patterns pour le refactoring

Travaux pratiques

Mise en place de la stratégie sur un code legacy donné

Aller plus loin

- Mettre en place l'intégration continue avec Azure DevOps
- Bien faire du Pair Programming
- Ping-Pong Programming
- TDD : limites, progression
- Les outils et bonnes pratiques

Travaux pratiques

Exercice de pair-programming sur un exemple concret

Compétences attendues :

A la fin de cette formation, les participants seront en mesure de :

- Savoir quel est un besoin correctement identifié et un besoin à affiner

- Transformer un besoin en éléments de travail
- Expliquer ce qu'est un test unitaire et un test d'intégration
- Mettre en pratique le TDD sur un nouveau besoin
- Utiliser des techniques avancées pour tester des algorithmes complexes
- Organiser et structurer son code proprement
- Adopter une stratégie de migration de code legacy vers du TDD
- Travailler en équipe grâce aux concepts du TDD