Test Driven Development en C#

Cours Pratique de 3 jours - 21h
Réf : TDN - Prix 2024 : 1 870€ HT

Ce cours couvre à la fois les pratiques individuels, de binôme et d'équipe. Il apprend la conception émergente, le refactoring des codes existants et une bonne couverture des tests automatiques. Il explique les tests fonctionnels automatiques, réalisés en collaboration avec des experts fonctionnels et des testeurs.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Ecriture des tests automatiques.

Isolation des tests.

Développement piloté par les tests.

Développement piloté par les comportements.

Refactoring sous tests.

TRAVAUX PRATIQUES

Alternance de séquences conceptuelles et de travaux pratiques.

LE PROGRAMME

dernière mise à jour : 01/2022

1) Agilité

- Le manifeste pour le développement agile de logiciels.
- Les principes agiles.
- Les signataires du manifeste agile.
- Pas une, mais des méthodes et pratiques agiles.
- eXtreme Programming.
- Retourner la pyramide des tests.
- Cadrant des tests agiles.
- Preuve intégrée : livrer du code et de la preuve de fonctionnement.

2) Tests automatiques

- Typologie des tests. Tests automatiques.
- Utilisation de bibliothèque de test NUnit ou équivalent.
- Arrangement Action Assertion.
- Principe d'isolation des tests pour la reproductibilité.
- Séparer le code de production et les tests. Objet et contrat.
- Créer un test sur un code de production.
- Nommer un test unitaire.
- La couverture comme outil de développement.

Travaux pratiques : Live coding et exercice de développement de tests automatiques.

3) Isolation des tests

- Notions de Fake, Stub & Mock.
- Isolation d'un composant à tester.
- Isolation d'une intégration à tester.
- Outils d'isolation.
- Pratique extract & override.

PARTICIPANTS

Ce stage s'adresse à des ingénieurs ou des leaders techniques, architectes, managers et chefs de projets en développement logiciel.

PRÉREQUIS

Pratique de la programmation objet en C#.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques…
Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

• Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
• À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
• Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Utilisation de bibliothèque Moq ou équivalent.
Travaux pratiques : Live coding et exercice d'extract & override et de mock.

4) Développement piloté par les tests
- TDD test driven development.
- Le baby step.
- Trouver un chemin vers la fonctionnalité.
- Utiliser les assertions pour coder.
Travaux pratiques : Live coding et exercice de TDD.

5) Développement piloté par les comportements
- Problèmes de communication. Histoire utilisateur, critères d'acceptation.
- Tests fonctionnels.
- ATDD Acceptance Test Driven Development.
- Spécification par l'exemple. Stratégie 3 amigos pour un meilleur dialogue. Partager la compréhension du domaine métier.
- Spécifications et recette dans un même geste, en continu. BDD Behavior Driven Development.
- Features et Steps. Le cycle du BDD.
- Utilisation de la bibliothèque Specflow ou équivalent.
- Développement piloté par les comportements.
Travaux pratiques : Live coding et exercice de BDD.

6) Développement en binôme
- Le style pilote & co-pilote.
- Le style ping pong.
- Soutien mutuel.
- Négociation constante.
Travaux pratiques : Exercice de développement TDD en binôme.

7) Refactoring sous tests.
- Code historique. Dette technique. Analysis Paralysis.
- Un code qui a 2 ans devrait être meilleur qu'à 2 semaines.
- Cesser de produire encore plus de dette technique.
- Le principe du boy scout.
- Tests de fumé, de caractérisation. Tests parallèles. Notion de code smells.
- Rappel des principes SOLID de la programmation objet.
- Améliorer en priorité là où l'on travaille.
- Utilisation de refactoring automatique avec Resharper ou équivalent.
Travaux pratiques : Live coding et exercice de refactoring.

LES DATES

CLASSE À DISTANCE
2024 : 13 mai, 15 juil., 28 oct.

PARIS
2024 : 08 juil., 21 oct.