Méthodes Agiles, ingénierie et test logiciel

Cours Pratique de 2 jours
Réf : MAD - Prix 2022 : 1 580€ HT
Le prix pour les dates de sessions 2023 pourra être révisé

Les méthodes de développement agiles recommandent de livrer fréquemment un logiciel à tester, tout en encourageant l'excellence technique. Vous apprendrez à mettre en place une méthodologie de développement pilotée par les tests, à augmenter la qualité de votre code et à utiliser les tests comme spécification.

OBJECTIFS PÉDAGOGIQUES

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

Comprendre la philosophie et les règles d'ingénierie logicielle en agile

Mettre en œuvre des tests unitaires avec un framework de test

Implémenter une solution en Test Driven Development en Java

Appliquer des techniques pour refactorer un code à risque

Utiliser des doubles de tests pour simuler un comportement

Automatiser un scénario de tests fonctionnels

MÉTHODES PÉDAGOGIQUES

Alternance cours et ateliers. Echanges d'expériences. Partage et formalisation des bonnes pratiques.

LE PROGRAMME

1) Introduction

- Pratiques d'ingénierie logicielle et méthodes Agiles.
- Le développement incrémental et itératif.
- L'équipe Agile. Scrum et XP.

Réflexion collective : Partage d'expérience et échanges autour des pratiques d'ingénierie et de test logiciel.

2) Les tests agiles

- Définition et périmètre des tests agiles.
- Cycle de développement : origine du TDD (Test Driven Development), ATDD, TDR, les types de tests...

3) Les tests développeurs

- Définition et objectifs : les patterns basiques XUnit.
- Principe des tests unitaires automatisés.
- Règles de simplicité : règle des "3 A" (Arrange, Act, Assert).
- Mise en œuvre de tests unitaires avec JUnit, le framework de test en Java.
- Lanceur de tests (TestRunner).
- Les méthodes d'Assertions.

4) Le TDD, développement guidé par les tests

- Le cycle de développement.
- Le principe du TDD : "test first", "tester, coder, refactorer".
- TDD et pratiques agiles (XP) : l'intégration continue, le Pair Programming.

PARTICIPANTS

Développeurs, architectes, testeurs, futurs managers Agiles et Scrum Masters, responsables qualité/méthodes.

PRÉREQUIS

Connaissances de base des méthodes Agiles.
Connaissances en programmation en Java.

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.

- Démonstration du TDD sur un cas concret.
Etude de cas : Implémentation d'une solution en TDD, écriture des tests Junit.

5) "Refactoring", le remaniement de code

- Principes du refactoring.
- Réduire l'apparition de la dette technique, rendre le code compréhensible.
- Comment identifier le code à risque ? La notion de "Code Smells", signes de danger potentiel.
- Les principales opérations de refactoring.
- Rappel sur les Design Patterns.

Travaux pratiques : Refactoring de code à risque.

6) Isolation des tests

- Les doubles de test, leur utilisation.
- Le "Mock Object" pour vérifier certaines hypothèses.
- Le "Fake", pour la simulation.
- Le "Stub" : fournir une réponse prédéfinie à un appel.

Travaux pratiques : Utilisation de double de tests.

7) Le test comme cahier des charges, la notion d'ATDD

- Les principes et avantages de l'ATDD.
- Du scénario au test de recette.
- Combiner ATDD, BDD et TDD.
- Les outils (Fitnesse, Cucumber...).

Etude de cas : Rédaction et automatisation de scénarios de tests.

8) Conclusions

- Les bénéfices du TDD, le coût des tests.
- Les autres types de tests (interface graphique, Web..).
- Quelques outils.

LES DATES

PARIS LA DÉFENSE
2022 : 05 déc.

CLASSE A DISTANCE
2022 : 05 déc.

2023 : 02 févr., 06 avr., 20 juil., 05 oct.