SARL ONEO
1025 rue Henri Becquerel
Parc Club du Millénaire, Bât. 27
34000 MONTPELLIER
N° organisme : 91 34 0570434

Nous contacter :
+33 (0)4 67 13 45 45
www.ait.fr
contact@ait.fr

Référence	**4-IT-QBAS**
Durée	**4 jours (28 heures)**
Éligible CPF	**NON**
Mise à jour	**21/12/2021**

Qualité logicielle et tests - Les fondamentaux

Objectifs pedagogiques

- Comprendre la problématique de la complexité des développements logiciels
- Comprendre le bénéfice de l'intégration continue pour améliorer la qualité des développements
- Adopter les bonnes pratiques de rédaction des tests logiciels
- Mettre en place une stratégie de campagne de tests automatisés avec les Framework et outils du marché
- Travailler avec un gestionnaire de code source tel que Git, et adopter les réflexes du travail collaboratif
- Comprendre le rôle des différents outils d'une chaîne d'intégration continue
- Utiliser un outil de construction logiciel pour automatiser les tâches de développement
- Mettre en place une plateforme d'intégration continue autour de Jenkins

Public concerne

- Développeurs, chefs de projets, architectes logiciels

Prerequis

- Posséder les connaissances et compétences équivalentes aux fondamentaux du développement Java ou aux fondamentaux du développement .NET ou aux fondamentaux de la programmation orientée objet en C++

Moyens pedagogiques

- Réflexion de groupe et apports théoriques du formateur
- Travail d'échange avec les participants sous forme de réunion-discussion
- Utilisation de cas concrets issus de l'expérience professionnelle
- Validation des acquis par des questionnaires, des tests d'évaluation, des mises en situation et des jeux pédagogiques.
- Alternance entre apports théoriques et exercices pratiques (en moyenne 30 à 50%)
- Remise d'un support de cours.

Modalites d'evaluation

- Feuille de présence signée en demi-journée,
- Evaluation des acquis tout au long de la formation,
- Questionnaire de satisfaction,
- Attestation de stage à chaque apprenant,
- Positionnement préalable oral ou écrit,
- Evaluation formative tout au long de la formation,
- Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles

Moyens techniques en presentiel

- Accueil des stagiaires dans une salle dédiée à la formation équipée à minima d'un vidéo projecteur et d'un tableau blanc et/ou paperboard.
- Pour les formations nécessitant un ordinateur, un PC est mis à disposition de chaque participant.

Moyens techniques en distanciel

- A l'aide d'un logiciel (Teams, Zoom…), d'un micro et éventuellement d'une caméra les apprenants interagissent et communiquent entre eux et avec le formateur.
- Sessions organisées en inter comme en intra entreprise.
- L'accès à l'environnement d'apprentissage ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.
- Pour toute question avant et pendant le parcours, assistance technique à disposition au 04 67 13 45 45.

Organisation

- Délai d'accès : 5 jours ouvrés
  (délai variable en fonction du financeur)
- Les cours ont lieu de 9h à 12h30 et de 13h30 à 17h

Accessibilite

- Les personnes en situation d'handicap sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.
- Pour tout renseignement, notre référent handicap reste à votre disposition : mteyssedou@ait.fr

Profil formateur

- Nos formateur sont des experts
- Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.

Certification possible

- Aucune

Qualité logicielle et tests - Les fondamentaux

Introduction

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

Les tests agiles

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

Les tests developpeurs

- ☐ 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.

Le TDD, developpement guide 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.

"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.

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.

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...).

Conclusions

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