

Qualité des applications

Référence : **DEQUAL001**

Durée : **3 jours (21 heures)**

Certification : **Aucune**

Connaissances préalables

- Disposer d'une première expérience de développement et maîtriser un langage de programmation (C#, Java ou C++)

Profil des stagiaires

- Architectes
- Développeurs
- Analystes
- Chefs de projets

Objectifs

- Connaître les bonnes pratiques d'écriture d'un code incluant la maintenance de l'application
- Connaître les outils nécessaires à la fabrique logicielle pour produire des livrables de qualité
- Appréhender l'offre des outils de tests de performance et de charge
- Appréhender les outils et phases de mise en œuvre d'une intégration continue
- Savoir utiliser Git pour gérer les codes sources

Certification préparée

- Aucune

Méthodes pédagogiques

- 6 à 12 personnes maximum par cours, 1 poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions

Formateur·rice

- Consultant-Formateur expert Développement

Méthodes d'évaluation des acquis

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation des compétences acquises envoyée au stagiaire
- Attestation de fin de stage adressée avec la facture

Contenu du cours

1. Généralités

- Qu'est-ce que la qualité
- Le coût de la non-qualité
- La qualité du code
- La qualité des livrables
- La qualité de la documentation

2. Génie logiciel

- Couplage fort / couplage faible
- Structuration modulaire des programmes
- Les cycles de vie (en V, en spirale)
- Notion de méthode agile
- Développement piloté par les tests (TDD)
- Développement piloté par le comportement (BDD)

3. Les tests

- Qu'est-ce que le test ?
- Plan de test, jeux de test
- Tests fonctionnels et non fonctionnels
- Test sen boîte blanche vs boîte noire
- Tests automatisés : avantages et limites
- Tests unitaires, d'intégration, de recette
- Tests d'IHM : client lourd, client léger
- Test de charge, de performances, de mise à l'échelle
- Exemple avec JUnit et Selenium

4. Git

- Principe de Git
- Commit
- Branch, merge et gestion des conflits
- Travailler en local ou à distance
- Gestion des versions du code et de la BdD
- Règles pour travailler ensemble
- Les différents outils : GitHub, GitLab, ...

5. Jenkins

- Intégration continue
- Déploiement continu
- Utilisation de Jenkins
- Intégration de Maven
- Présentation des alternatives

Notre référent handicap se tient à votre disposition au [01.71.19.70.30](tel:01.71.19.70.30) ou par mail à referent.handicap@edugroupe.com pour recueillir vos éventuels besoins d'aménagements, afin de vous offrir la meilleure expérience possible.