

Pratiquer le TDD (Test Driven Development) en C++

Détails

- **Code** : MT-CPPT
- **Durée** : 3 jours (21 heures)

Public

- Chefs de projets
- Développeurs
- Responsables tests
- Responsables qualité

Pré-requis

- Pratique professionnelle du développement C++

Objectifs

- Acquisition des principes fondamentaux et des bonnes pratiques du TDD
- Prise en main approfondie du Framework GoogleTest (GT) / GoogleMock (GM)
- Rédaction de tests élaborés et simulation avec GT / GM (Mocking / Facking)
- Pratique du refactoring d'un code développé avec l'approche TDD
- Principes de conception

Programme

La place des tests dans le processus de développement

- Processus, qualité, tests
- Typologie des tests
 - Approche TDR (Test Driven Requirements)
 - from user stories to test cases
 - Tests unitaires
 - Tests d'intégration
 - Tests de (non) régression, ...
- Mise en place du TDD au sein d'une démarche agile
 - Scrum, eXtreme Programming, ...
 - Pair Programming

Frameworks de tests automatisés

- Familles de frameworks
- Frameworks C++

Principes fondamentaux du TDD

- Calibrage des tests, principe YAGNI
- Le cycle de développement du TDD
 - Le triple A « Arrange, Act, Assert »
- Stratégies pour faire passer les tests
 - Approche Outside-In vs. Inside-Out

Prise en main de GoogleTest

- Création d'un projet
- Concepts de base
- Assertions
- Tests élémentaires
- Tests multiples : suites de tests et Test Cases
- Préparation et mutualisation des tests avec les Test fixtures
- Pilotage des tests

Mocks et doublures

- Qu'est-ce qu'une doublure ?
- Types de doublures

- Mocks et Fakes

Prise en main de GoogleMock

- Création d'une classe Mock
 - Méthodes privées, publiques, surchargées, ...
- Mocking et classes génériques
- Mocking de méthodes scellées
- Mocking de code C « legacy »
- Mocking par délégation
- Collaboration des Mocks et des objets réels

Utilisation des Matchers Hamcrest

- Matchers élémentaires et combinaisons
- Casting de Matchers
- Matchers multiples et dispatching
- Prédicats
- Arguments non copiables
- Validation de membres et pointeurs
- Matching et conteneurs

Tests comportementaux

- Simulation du cycle de vie d'un objet
- Attentes (« expectations »)
- Changement de comportement du mock selon l'état de l' « objet réel »
- Cas particuliers
 - Retour de références, pointeurs, exceptions
 - Classes et méthodes génériques

Refactoring et principes de conception

- Composition versus spécialisation
- Polymorphisme versus généricité
- Inversion des dépendances
- Quelques "mauvaises odeurs" à combattre
- Techniques de refactoring en TDD
- Présentation de quelques patterns simples mis en oeuvre lors du refactoring / TDD

Modalité

- Stage pratique en présentiel
- Stage pratique en distanciel
- Nombre de stagiaires minimum : 4
- Nombre de stagiaires maximum : 10

Méthodes pédagogiques

- Exposés
- Cas pratiques
- Echanges d'expérience

Profils des intervenants

- Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.

Modalités d'évaluation

- Evaluation des acquis de la formation par le biais de cas pratiques et/ou mises en situation.
- Attestation de formation remise à chaque participant.

Démarche qualité

- Questionnaire d'évaluation de satisfaction à chaud complété par chaque participant à l'issue de la formation.

Moyens pédagogiques

- Salle équipée de PC (1 poste par stagiaire), vidéo-projecteur.
- Espace de pause.

Dernière mise à jour le 10/05/2022