

# Formation - Améliorer ses tests automatisés

Cette formation a pour objectif de rendre le code des tests automatisés plus maintenable et plus fiable. Vous apprendrez les idées et techniques qui permettent d'écrire des ensembles de tests plus rapides, plus faciles à comprendre et qui n'échouent plus pour de mauvaises raisons. Vos tests vous permettront enfin de réellement gagner du temps, d'avoir une meilleure qualité logicielle et surtout un meilleur retour sur investissement.

## Objectifs :

### Cette formation a pour objectif de :

- Développer une compréhension approfondie des tests logiciels en identifiant leurs qualités et défauts, équilibrer une suite de tests avec des méthodologies variées, et communiquer efficacement les fonctionnalités du système testé
- Acquérir les compétences nécessaires pour rédiger des tests clairs et maintenables, identifier et éviter les pièges courants qui nuisent à l'efficacité des tests
- Acquérir les compétences pour créer et gérer des doublures de test adaptées à différents besoins sans dépendre de frameworks, assurant des tests maintenables et évolutifs pour le code et les dépendances externes
- Apprendre à rédiger des tests qui fonctionnent de manière indépendante et à gérer efficacement les données dans une base de données partagée pour maintenir l'intégrité des tests

### À l'issue de cette formation chaque apprenant sera capable de :

- Sélectionner des types de tests adaptés aux risques associés, créer de la documentation utile à partir des tests, et évaluer les avantages ainsi que les inconvénients des différentes approches de tests
- Appliquer le modèle Arrange-Act-Assert, optimiser les tests avec des assertions uniques et des patterns de conception, et réduire la redondance grâce à l'utilisation de data providers, tout en rendant les tests accessibles à tous les profils professionnels
- Développer et appliquer des doublures efficaces, y compris les fakes et mocks, pour garantir des tests robustes qui respectent l'évolution et l'intégrité du système tout en permettant le refactorisation du code
- Implémenter des tests autonomes, utiliser des constructeurs pour la préparation des données et appliquer diverses stratégies pour réinitialiser la base de données après les tests

## Description :

### **Cette formation vous permettra de :**

Ecrire des tests facilement maintenables, de pouvoir vous servir de vos tests comme documentation, d'écrire des tests plus rapides, de ne pas tomber dans les pièges de mocks et de bien gérer les fixtures de vos tests en intégration avec une base de données

### **Méthode et moyens pédagogiques utilisés :**

- ✓ Cours vidéos
- ✓ QCM
- ✓ Travaux pratiques
- ✓ Connexion internet
- ✓ Adresse email
- ✓ PC permettant de travailler à distance / domicile
- ✓ Casque audio ou écouteurs
- ✓ Son

## Prérequis :

Cette formation ne nécessite pas de prérequis.

## Evaluation:

En amont : Questionnaire d'évaluation à l'entrée en formation

En cours : Quizz d'activation des connaissances et cas pratiques

En fin de formation : Questionnaire d'évaluation de fin de formation

## Certification :

Remise de :

- Une attestation de suivi de formation
- Certificat de réussite UpSkill4IT (si les conditions sont remplies)

## Délais d'accès :

Video on demand, accès immédiat après la signature du contrat

## Durée de formation :

Cette formation en e-learning dure 7h sous format VOD (e-learning) et peut être réalisée en 1 journée.

## Accès et format :

Cette formation est réalisable à votre rythme durant 2 mois en format VOD (e-learning)

## Public concerné :

- Développeurs
- Testeurs / Ingénieurs QA

## Actualisation :

Date 05/03/2024 – V:001

## Présentation du formateur :

**Charles DESNEUF,**  
Software Architect & Craft Coach

Développeur depuis plus d'une dizaine d'année, il s'intéresse à tout ce qui permet de concevoir des applications qui rendent véritablement service à nos utilisateurs et utilisatrices et porte donc un intérêt particulier aux tests. Au quotidien il intervient en tant qu'Architecte et Coach Craft auprès d'équipes pour les aider à améliorer leurs pratiques afin de délivrer de nouvelles fonctionnalités à leurs utilisateurs de manière continue et sur des cycles de temps courts.

## Accessibilité :

Si vous avez des difficultés ou êtes en situation de handicap veuillez nous contacter à l'adresse mail suivante : [contact@upskill4it.com](mailto:contact@upskill4it.com)

# Sommaire 1/1

## Section 1 - Théorie

Développer une compréhension approfondie des tests logiciels en identifiant leurs qualités et défauts, équilibrer une suite de tests avec des méthodologies variées, et communiquer efficacement les fonctionnalités du système testé

1. Leçon 1 - Pourquoi tester
2. Leçon 2 - Caractéristiques des bons tests - Qu'est-ce qui fait un bon test
3. Leçon 3 - Unitaire, Intégration, ... - Une définition parmi tant d'autres
4. Leçon 4 - Économie des tests - Des stratégies à géométrie variable
5. Leçon 5 - Nommer les tests - Ce que l'on conçoit bien s'énonce clairement

## Section 2 - Des tests davantage lisibles et maintenables

Acquérir les compétences nécessaires pour rédiger des tests clairs et maintenables, identifier et éviter les pièges courants qui nuisent à l'efficacité des tests

1. Leçon 1 - Arrange Act Assert - Une structure de test reconnaissable au premier coup d'œil
2. Leçon 2 - Une assertion par test - Comprendre facilement pourquoi un test échoue
3. Leçon 3 - Assertions sur mesure - Faciliter la compréhension de ce qui est vérifié
4. Leçon 4 - Méthodes de création - Encapsuler la logique de création des objets
5. Leçon 5 - Attention aux structures de contrôle - Ne pas se perdre dans les tests
6. Leçon 6 - Pas de calcul dans les tests - Garder la logique dans le code de prod
7. Leçon 7 - Expected object - Vérifier l'intégralité de l'état d'un objet d'un seul coup
8. Leçon 8 - Convention des variables - Connaître leur rôle à la vitesse de l'éclair
9. Leçon 9 - Object mother - Créer facilement les types courants
10. Leçon 10 - Builders - Écrire les tests comme une histoire
11. Leçon 11 - Data driven tests - Améliorer la documentation en supprimant la duplication
12. Leçon 12 - Organisation des classes de tests - Sortir du classique mapping 1-1

## Section 3 - Mocks, Stub, Fake, Spy, Doubles ?

Acquérir les compétences pour créer et gérer des doublures de test adaptées à différents besoins sans dépendre de frameworks, assurant des tests maintenables et évolutifs pour le code et les dépendances externes

1. Leçon 1 - Différents types de doublures - S'en sortir dans les batailles de vocabulaire
2. Leçon 2 - Problème de mocks - Figurer une implémentation à jamais
3. Leçon 3 - Don't mock what you don't own - Mettre de la distance avec les affaires des autres
4. Leçon 4 - Fake en mémoire - Remplacer une implémentation pénible
5. Leçon 5 - Test de contrat - Ne pas découvrir le pot aux roses en production
6. Leçon 6 - Stubbing du temps - Prendre le contrôle du système

## Section 4 - Tester avec une base de données

Apprendre à rédiger des tests qui fonctionnent de manière indépendante et à gérer efficacement les données dans une base de données partagée pour maintenir l'intégrité des tests

1. Leçon 1 - Attention aux données partagées - Ne pas coupler les tests discrètement
2. Leçon 2 - Nettoyer les fixtures - Merci de laisser la base de données dans l'état où vous voudriez la trouver

## Evaluation finale

Quiz