Software Craftsmanship

Clean Code, TDD, BDD et principes SOLID

Présentiel ou en classe à distance

3 jours (21 h) Réf.: OB218

Un logiciel peut être parfaitement fonctionnel mais poser de nombreux problèmes de fiabilité et de maintenabilité en raison de la piètre qualité de son code. La conséquence ? Si les coûts initiaux de développement ont été maitrisés, ceux liés à sa maintenance ou à son évolution peuvent vite devenir exponentiels. Le Software Craftsmanship ou “l’artisanat du logiciel” propose un ensemble de méthodes et d’approches de haute qualité pour concevoir et construire des bases de code de tous volumes maintenables à coût constant. Ces méthodes s’appuient sur les principes SOLID, les principes du Clean Code et de l’eXtreme Programming.

A qui s'adresse cette formation ?

- Développeurs

- Avoir une bonne connaissance du langage Java et de ses API fondamentales
- Une première confrontation aux problèmes de gestion de bases de code anciennes, volumineuses et non (ou mal) testées est un avantage

Programme

Pendant

1 - Introduction à la pratique du développement dirigé par les tests, le cycle de développement TDD

- Exemple du Kata FizzBuzz
- Développement du Kata
- Bilan : les étapes de la pratique du TDD

2 - Introduction des principes du Clean Code et des principes SOLID

3 - Pratiques de l’eXtreme Programming

- Pratiques des Katas
- Pratiques des Coding Dojo
- Pratiques du Pair Programming

4 - Le principe Open / Closed : application au pattern Strategy

- Exemple du Kata RPN Calculator
- Développement du Kata
- Bilan : le pattern Strategy et son implémentation

5 - Travail sur le code legacy, Single Responsibility Principl

- Spécificités du travail sur code legacy
- Exemple du Kata Rental Movie (code legacy)
- Bilan : détecter les manquements au SRP

6 - Utilisation de Gherkin / Cucumber pour l'écriture de tests

- Écriture de tests en Gherkin : méthodes, organisation, syntaxe
- Écriture de classes Cucumber pour l'exécution de ces tests
- Fonctionnalités avancées : tests paramétrés, tables de données, tags
- Intégration avec Maven, génération de rapports de tests

7 - Mise en oeuvre sur un kata complexe : Mars Rover

- Écriture des tests avec Cucumber
- Résolution du kata



Les objectifs de la formation

- Comprendre la notion de développement dirigé par la valeur
- Maîtriser les différentes étapes du développement TDD
- Maîtriser les principes SOLID et les patterns du "clean code" en programmation objet
- Comprendre les pratiques de l'eXtreme programming : pair programming, coding dojos
- Savoir appliquer ces principes et patterns au développement de nouveau code (Greenfield development) et au code existant (Brownfield development



Evaluation

Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.



Les points forts de la formation

- Une formation essentiellement axée sur la pratique : 70% du temps de la formation est consacré au codage. La partie codage porte principalement sur le codage de "Katas", un exercice essentiel pour s'exercer à la pratique du TDD / BDD.
- Le partage de bonnes pratiques et les retours d'expériences d'un spécialiste du Software Craftsmanship

Evaluation

Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.