

Module 3 : Ingénierie Logicielle Agile : 2 jours

Total horaire : 14 heures

Objectifs

- Appréhender le développement agile basé sur des techniques modernes : TDD, intégration continue, automatisation des tests
- Comprendre l'importance des tests dans la production de logiciels et le développement
- Repenser ses applications existantes par la mise en place de tests pour amélioration continue
- Sensibiliser à son rôle de membre d'une équipe agile

Pré-requis

- Modules 1 & 2 ou posséder une culture agile et savoir travailler en équipe

Public

- Développeurs, architectes, testeurs, futurs Managers Agile (Scrum Masters), responsables qualité/méthodes
- 4 à 12 personnes maximum

Modalités pédagogiques

- Avant : Supports mis en ligne pour appréhensions des sujets traités
- Pendant : 2 jours en présentiel
 - 40% théorique
 - 50% ateliers
 - 10 % échanges
- Après : Accès aux ressources en ligne, discussions en ligne
- Méthode pédagogique : Mises en pratique par coding katas privilégiant interactions et échanges entre les participants, questionnaires et plans d'actions

Profil des formateurs

- Expert en Méthodes Agiles et particulièrement Scrum
- Certifié Scrum Master

Déroulé

Avant et Après :

- Accès aux ressources partagées (supports)
- Communauté en Ligne
- Plan d'action individualisé

Journée 1 Matin : Test en Agile et xUnit

- Introduction, présentation des participants
- Questionnaire d'évaluation des acquis des modules 1 & 2
- Le test en Agile :
 - Qualité et agilité
 - Approche globale et pilotage par les tests
 - TDD, BDD, ATDD
 - Automatisation des tests
 - Outils, reproductibilité
 - Types de tests
- Les librairies xUnit :
 - Déclarer un test
 - Les assertions
 - Préparation et nettoyage du test
 - Organisation des tests

Atelier: Ecriture de tests unitaires sur un code existant

Journée 1 Après-midi : Principes et styles du TDD

- Principes du TDD :
 - Objectifs et avantages
 - Principe du Test First
 - Émergence du design à partir des tests
- Ateliers : FizzBuzz*
- Style de TDD :
 - Tests basés sur l'état ou le comportement
 - Utilisation de doublures (mocks, stubs, spy, fakes, etc.)
- Atelier : Fizzbuzz*
- Evaluations des acquis et plan d'action individuel et/ou collectif

Journée 2 Matin : Refactoring & code testable, TDD & Code legacy

- Refactoring & code testable :
 - Méthodologie SOLID
 - Inversion de dépendances et découplage
 - Polymorphisme, nommage
 - Elimination de la duplication
 - Code smells et refactoring
- Atelier : Kata Bowling Game*
- TDD et Code legacy :
 - Test et code intestable
 - Stratégies de refactoring du code legacy

Atelier : Code legacy à refactorer

Journée 2 Après-midi : Parties connexes & Aller au-delà

- Les parties connexes
 - L'intégration continue
- Le Pair Programming
Atelier : Randori Nombres romains
- Aller au-delà
 - Pratiquer TDD
 - Progresser en TDD
 - Faire adopter TDD
- Evaluations des acquis et plan d'action individuel et/ou collectif

Description des outils pédagogiques

Journée et Séquence	Nom du Kata	Description
Jour 1 matin	Movie finder	Ecriture de tests unitaires sur un code existant
Jour 1 après-midi	Fizzbuzz	S'exercer sur des codes selon l'approche TDD
Jour 2 matin	Kata bowling Code legacy	Mettre en œuvre des codes sur TDD & de design patterns S'exercer sur les principes de refactoring
Jour 2 après-midi	Randori Nombres romains	Convertir des nombres entiers en leur équivalent en chiffres romains, à travers un randori (tous les participants se succèdent pour écrire un morceau de la solution).

Remarques : Un kata est un exercice que l'on fait avec pour objectif de maîtriser une technique précise. Le kata est toujours axé sur une problématique bien identifiée, comme la pratique du TDD, la mise en œuvre de design patterns, de principes de refactoring, ou autre.

Lors de la formation, plusieurs katas se pratiquent seul ou à plusieurs. Cela devient alors un dojo. Le dojo a la particularité d'être régulier pour un groupe de personnes (1 fois par semaine, par mois...) La formation sensibilise à l'intérêt de mettre en œuvre des dojos de code dans l'entreprise.