

Module 3 : Ingénierie logicielle agile

Donner un éclairage sur les bonnes pratiques de développement logiciel, plutôt que d'apporter une hyperspécialisation grâce à

Présentiel - Synchrone

- Une sensibilisation au fait d'être acteur et responsable du code que l'équipe produit en estimant qu'il s'agit de ce que l'équipe pense être le meilleur,
- Un développement d'équipe auto-organisée et responsable de ce qu'elle fait

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



Pré Requis

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

Objectifs pédagogiques et d'évaluation

- 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

Méthodes pédagogiques

- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions

Méthodes d'évaluation des Acquis

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressé avec la facture ou remise par le

Parcours pédagogique

Journée 1 Matin : Test en Agile et xUnit

1. Introduction, présentation des participants
 2. Questionnaire d'évaluation des acquis des modules 1 & 2
 3. 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
 4. Les bibliothèques 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

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

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

1. Refactoring & code testable :

- Méthodologie SOLID
- Inversion de dépendances et découplage
- Polymorphisme, nommage
- Elimination de la duplication
- Code smells et refactoring

Atelier : Kata Bowling Game

2. TDD et Code legacy :

- Test et code instable
- Stratégies de refactoring du code legacy

Atelier : Code legacy à refactorer

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

1. Les parties connexes

- L'intégration continue

2. Le Pair Programming

Atelier : Randori Nombres romains

3. Aller au-delà

- Pratiquer TDD

- Progresser en TDD

- Faire adopter TDD

4. Evaluations des acquis et plan d'action individuel et/ou collectif



Méthodes et modalités d'évaluation

Test des connaissances acquises et délivrance d'une attestation de fin



Durée

14.00 Heures **2** Jours