

Formation Software Craftmanship

Maîtriser les principes, méthodes et outils du Craft pour développer des logiciels de qualité et durables

Grâce à cette formation, les participants comprendront l'attitude et l'état d'esprit du software craftsmanship, appliqueront les principes de la POO et les principes SOLID, et maîtriseront les techniques de tests automatisés, dont TDD et BDD. Ils utiliseront les acronymes YAGNI, KISS, DRY et POLA pour guider la conception, et apprendront à identifier et corriger les code smells dans un code legacy. La formation s'adresse aux développeurs ayant une expérience en programmation et une connaissance de base en POO.

Durée

3 jours

Objectifs pédagogiques

- Comprendre l'attitude et l'état d'esprit du software craftsmanship
- Appliquer les principes de la programmation orientée objet (POO) et les principes SOLID
- Maîtriser les techniques de tests automatisés, dont TDD et BDD
- Utiliser les acronymes pour guider la conception : YAGNI, KISS, DRY, POLA
- Identifier et corriger les code smells dans un code legacy

Public

Développeurs, chefs de projet

Prérequis

Connaissances en programmation orienté objet (POO) et expérience en développement logiciel.

Programme de formation

Introduction à la formation Software Craftmanship

Présentation des objectifs et du déroulement de la formation

Discussion sur l'importance du software craftsmanship

Comparaison avec les autres approches (Waterfall, Agile, etc.)

Exemple de cas pratiques : Analyse et discussion sur les approches de développement de logiciels

Le Craft

Attitude et état d'esprit du software craftsmanship

Les valeurs et principes clés du craft

Importance du continuous learning

Exemple de cas pratiques : Analyse de situations réelles pour identifier des opportunités d'apprentissage continu

Les tests automatisés

Les 3A : Arrange, Act, Assert

Test-Driven Development (TDD)

Chicago school vs London school

Behavior-Driven Development (BDD)

Overview de outils de tests automatisés : JUnit, Selenium...

Exemple de cas pratiques : Mise en place de tests automatisés pour un projet existant

Principes généraux et meilleures pratiques de développement

Principes fondateurs de la POO

Les principes SOLID

Importance du nommage

Le bon sens par les acronymes:

YAGNI/KISS/DRY/POLA

Refactoring legacy code

Les smells de Martin Fowler (Duplicated method, Duplicated class, Long method, Long class, Primitive obsession)

Bref aperçu d'autres smells

Approval test/Golden master

Double appel

Exemple de cas pratiques : Identification et correction des code smells dans un projet hérité

Moyens et méthodes pédagogiques

- La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques.
- Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- Un support de cours numérique est fourni aux stagiaires

Modalités d'évaluation

- **En amont de la session de formation**, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation.
- **En cours de formation**, l'évaluation se fait sous forme de quiz, travaux pratiques, analyse de cas, présentation de projets et évaluation par les pairs
- **En fin de session**, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.