

## Fiche de la formation

### Durée :

2 jours, possibilité d'ajouter des modules complémentaires.

### Audience :

Développeurs PHP confirmés.

### Pré-requis :

Maîtrise de la programmation orientée objet (OOP).

### Objectif pédagogiques :

Acquérir la maîtrise de l'outil et la méthode associée pour assurer la qualité du code.

### Méthodologie :

La méthodologie mise en oeuvre dans cette formation a pour but premier une prise en main réelle des outils et technologies abordés, orientée vers une maîtrise propre à assurer une productivité en accord avec les besoins de l'entreprise.

## En bref

Le savoir est la composante essentielle du savoir-faire !

Pour que vos équipes soient performantes sur une technologie ou un outil, elles doivent en connaître les fondamentaux. L'expérience démontre que les meilleurs outils utilisés sans formation initiale sont très largement sous-exploités, voire contre-productif.

Afin d'éviter cette situation, génératrice de frustration et obstacle à la qualité, Vesperia vous propose des prestations de transfert de compétence à forte valeur ajoutée, reposant soit sur le matériel de formation des technologies concernées, soit sur des programmes établis sur-mesure (suite à un audit de compétences).

## Description

Tester une application est capital pour garantir sa qualité.

Parmi les différentes méthodes de tests possibles, le test unitaire, appliqué aux couches les plus basses de l'application, est la plus efficace.

Intégrer la pratique du test unitaire, et mieux, du développement piloté par le test, est le meilleur moyen d'améliorer significativement la qualité du code produit tout en réduisant considérablement les bugs.

## Programme de cours de la formation

### Tests Unitaires avec PHPUnit

- I . What is unit testing ? [Core I-V]
- II. Wording
- III. Starting working with PHPUnit
- IV. Assertions in details
- V. Advanced PHPUnit usage

## Plan de formation détaillé

### I. What is unit testing? [Core I-V]

1. Introduction
  1. About Quality Assurance
2. Manual testing
  1. Definition
  2. Benefits
  3. Drawbacks
3. Automated Testing
  1. Definition
  2. Benefits
  3. Drawbacks
4. Unit Testing
  1. Definition
  2. Benefits
  3. Drawbacks

### II. Wording

1. Definitions
  1. Fonctionnal test
  2. Technical test
  3. Test case
  4. Test suite
  5. Test group
  6. Fixture
  7. Unit
  8. Assertion
  9. Annotation
  10. Code coverage
  11. Mock
  12. Stub
  13. TDD
  14. BDD
2. Summary

### III. Starting working with PHPUnit

1. History
2. Competitors
3. Installation
4. Configuration
  1. Bootstrap
  2. phpunit.xml
5. Generating test skeleton
6. Running test
7. Test result interpretation

### IV. Assertions in details

### V. Advanced PHPUnit usage