

## Programme de formation

### Productivité et qualité des développements PHP

Les clefs de l'amélioration de la productivité et de la qualité des développements PHP, basées sur les pratiques agiles et les principes Clean Code/Software Craftmanship

#### Durée

2 jours / 14 heures

#### Description

Cette formation forme à la pensée agile coté programmation PHP. En partant de la présentation des principes fondamentaux du génie logiciels et des pratiques agiles, l'auditeur est amené à comprendre par la pratique pourquoi et comment les pratiques permettent de mieux implémenter les principes. Le voyage dans les best practices de codage couvre tous les niveaux du code, depuis les règles de nommage des variables jusqu'à l'utilisation des design patterns

#### Objectifs pédagogiques

- Réaliser un développement orienté objet
- Utiliser les best practices de la programmation PHP
- Mettre en pratique le refactoring
- Améliorer la productivité et la qualité du code

#### Public

Développeurs PHP confirmés, Experts Techniques, Architectes Logiciels

#### Pré-requis

Connaissance de PHP et notions de programmation orienté objet

#### Méthodes pédagogiques

40 % théorie / 60 % pratique

#### Profil intervenant

L'ensemble de nos formations sont animées par des formateurs expérimentés possédant une expérience terrain éprouvée.

#### Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers d'ateliers de mise en pratique des notions et concepts abordés pendant la formation.

## Programme

### Introduction

- Clean Code
- Software Craftsmanship
- Refactoring
- Pattern
- Agile ?

### Principes du génie logiciel

- Lisibilité
- Simplicité
- Evolutivité
- Testabilité
- Separation of Concerns !

### Pratiques agiles

- Pair Programming
- Propriété Collective
- Test Driven Development
- Refactoring
- Intégration Continue

### Outils d'analyse de qualité

- PHP\_CodeSniffer
- PHPDepend
- Sonar pour PHP
- L'intégration continue avec Jenkins/PHPUnit
- Les tests unitaires avec PHPUnit et Atoum

### Dans les Objets

- Nommage
- Commentaires
- Formatage
- Structure de contrôle
- Méthodes
- Exceptions

### Design Orienté Objet

- Concepts de base
- Spécificités PHP
- UML

### **General Responsibility Assignment Software Patterns (GRASP)**

- Creator
- Information Expert
- Faible Couplage
- Cohésion Forte
- Contrôleur
- Polymorphisme
- Fabrique Pure
- Indirection
- Protected Variation

### **SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion)**

- Single Responsibility
- Open/Closed
- Liskow
- Interface Segregation
- Dependency Inversion (Ioc, Dependency Injection)

### **Conclusion et recommandations**