# Choisir et mettre en œuvre des outils de test **Open Source**

aos

Spoc pratique
**Travaux pratiques continus**

Durée
**3 jours**

Vos Formateurs
**Olivier Charles, Philippe Brutus**

Code Kertify : **AOS**

S'inscrire

**www.kertify.com/inscription**

## *L'open source dans tous les domaines du test*

Le test est une activité importante dans le cadre de la qualité logicielle. De bonnes pratiques de tests lors des différentes étapes du développement contribuent à la qualité du produit final.

Ce stage se concentre sur les outils Open Source dédiés aux tests et à leur mise en œuvre dans les projets. Nous utiliserons plusieurs familles d'outils : Testlink, xUnit, TestNG, Hudson, Cobertura, Findbug, Selenium, Mantis,...

## Les objectifs de la formation

- Avoir une vision d'ensemble des différents types d'outils de test dans le monde Open Source.
- Comprendre l'utilisation des outils d'analyse statique et dynamique de code.
- Mettre en œuvre des tests unitaires à l'aide d'un framework dédié.
- Mettre en œuvre un générateur de build et une intégration continue.
- Identifier les rôles des outils pour les tests fonctionnels (génération de données, référentiel, gestion des anomalies).
- Comprendre et mettre en œuvre sur un exemple une automatisation de tests

## Participants

- Développeurs/testeurs,
- Maîtres d'œuvre
- Homologateur
- Maîtres d'ouvrage
- Ingénieurs qualité

## Prérequis

- Connaissances de base de l'ingénierie logicielle, du développement
- Connaissances de base des méthodes et des techniques de test

## Le programme de notre formation

### 1. Introduction au monde de l'Open Source

- Open Source versus logiciel libre.
- Le modèle économique de l'Open Source.
- Les licences (GPL, LGPL, BSD).
- Les projets communautaires.
- Les phases et axes du test.
- Le panorama des outils de test Open Source.
- La problématique d'acquisition d'un outil Open Source.

## 2. Les outils de tests unitaires

- Les besoins en test unitaire.
- Le test fonctionnel et le TDD.
- Les xUnit.
- Le test structurel et l'analyse de couverture.
- Les critères d'analyse de couverture (branch, BCCC).
- Les analyseurs dynamiques.
- Le test structurel et l'analyse statique.
- Les outils d'analyse de la qualité.
- **Travaux pratiques.** Mise en œuvre de tests unitaires à l'aide d'un framework dédié. Mesures de la couverture à l'aide d'un analyseur dynamique. Mesure de la qualité du code à l'aide d'outils dédiés.

## 3. Les outils de test d'intégration

- Les besoins en test d'intégration.
- Les outils de gestion de configuration.
- Les générateurs de builds. L'intégration continue.
- Les outils d'intégration continue.
- **Travaux pratiques.** Mise en œuvre d'un générateur de build et mise en place d'une intégration continue.

## 4. Les outils de test de performance

- Les besoins en test de performance. La typologie des outils. Les profilers.
- Les robots de test de charge. Les analyseurs de fuite mémoire.
- Les analyseurs de trafic réseau.
- **Travaux pratiques.** Mise en œuvre d'outils de profiling, d'un robot de test de charge, d'un outil de détection de fuites mémoire.

## 5. Les outils de gestion des anomalies

- Les besoins en gestion des anomalies.
- Les attributs d'une anomalie.
- Les fonctionnalités des outils de gestion des anomalies.
- Les outils de Bug Tracking.
- **Travaux pratiques.** Prise en main d'outils de gestion d'anomalies.

## 6. Les outils de test de validation

- Les besoins en test de validation.
- Les référentiels de test et leur mise en œuvre.
- Les générateurs de données de test.
- Les fonctionnalités des robots de test.
- Les robots de test d'IHM.
- Les robots de test d'applications Web.

- **Travaux pratiques.** Mise en œuvre d'un référentiel de test, d'un générateur de données de test, de robots de test d'IHM, de robots de test d'applications Web.