

Mise en œuvre des tests unitaires avec JUNIT

Ces dernières années, des études ont été faites afin de rendre les processus de développement de logiciel plus simples, plus faciles à mettre en place et plus sensible aux besoins du client. Différents **processus dits « agiles »** ont vu le jour (XP par exemple) et proposent de nouvelles pratiques visant à améliorer l'organisation des équipes de développement.

A sein de ces processus, la mise en place d'une véritable **stratégie de test** est essentielle et doit permettre de répondre aux différentes contraintes des nouvelles pratiques - approche incrémentale des développements et nombreuses livraisons, fiabilité des systèmes ...). On parle désormais de **programmation pilotée par les tests**.

Dans ce cadre le **test unitaire** est devenu une phase essentielle du cycle de développement logiciel. L'idée est d'aboutir à une granularité fine des tests en développant pour chaque incrément logiciel une classe chargée de le tester et de vérifier le résultat sans intervention du développeur. Les erreurs sont ainsi détectées facilement à un stade où leur correction est peu coûteuse.

<p>Objectifs</p> <ul style="list-style-type: none"> ■ Appréhender les tests unitaires ■ Développer des tests unitaires avec Junit et ses dérivés ■ Avoir une vision globale des tests unitaires (évolutions et outils) <p>Pré-requis Connaissance et pratique du langage JAVA.</p> <p>Profil participant Développeur Concepteur/Architecte Chef de Projet</p> <p>Durée 2 jours</p> <p>Tarif 640,00 € HT</p>	<p>Plan du cours</p> <ul style="list-style-type: none"> ▼ Présentation et Objectifs <ul style="list-style-type: none"> ■ Pourquoi des tests unitaires (enjeux et positionnement)? ■ Différents types de tests unitaires (présentation méthodes) ■ Tests unitaires en J2EE ▼ Ecrire un test unitaire - Utilisation du framework Junit <ul style="list-style-type: none"> ■ Présentation générale ■ Développement de classes de test ■ Exemples - Travaux pratiques ▼ Test unitaire par composant <ul style="list-style-type: none"> ■ Présentation de CACTUS, Mock Object et DBUnit ■ Test unitaire sur servlet ■ Test unitaire sur JSP ■ Test unitaire Database ■ Exemples – Travaux pratiques ▼ Utilisation dans Eclipse <ul style="list-style-type: none"> ■ Présentation générale d'Eclipse ■ Installation du Plugin ■ Exemple d'utilisation – Travaux pratiques ▼ Automatisation de Junit <ul style="list-style-type: none"> ■ Suites de tests (TestSuite) ■ Automatisation des tests avec ANT ▼ Présentation d'autres framework <ul style="list-style-type: none"> ■ HttpUnit (Interface WEB) ■ StrutsTestCase (Struts) ■ XMLUnit ■ Exemples – Travaux pratiques ▼ Evolution des tests unitaires <ul style="list-style-type: none"> ■ Synthèse et évolution (méthodes) ■ Outils du marché (présentation de l'offre Parasoft JTEST) 	<p>Travaux pratiques</p> <ul style="list-style-type: none"> ▼ Premiers exemples avec Junit ▼ Exemple sur différents composants (JSP, Servlet, Database) ▼ Intégration et utilisation dans Eclipse ▼ Automatisation avec ANT ▼ Utilisation de Cactus ▼ Utilisation de Mock Object ▼ Utilisation de DBUnit, HttpUnit ▼ Présentation de JTEST de Parasoft
---	---	---