



## TESTS UNITAIRES AVEC JUNIT ET ÉCLIPSE

Rédigé par Yannick Le Nir

Ref : *CO-UNIT-TEST*

Revu par : l'équipe pédagogique

Crée le 21/11/2012

|   |   |
|---|---|
| 1 Test unitaire de non régression:..... | 2 |
| 2 Assertions :.....                     | 2 |

### 1 Test unitaire de non régression:

Un test unitaire de régression s'applique à chaque fois que le logiciel est modifié, afin de s'assurer que ce qui fonctionnait avant les modifications fonctionne toujours

L'intérêt de modifier un code déjà testé, permet soit de corriger des défauts, soit d'ajouter des fonctionnalités. Ces ajouts ont lieu en phase de maintenance (TMA) ou en évolution de versions. De manière générale, les tests de non régressions s'appliquent à tous les types de tests (unitaires, intégration, systèmes), mais nous ne nous intéressons pour l'instant qu'aux tests unitaires.

On distingue deux problèmes spécifiques :

1. Evolution/maintenance : les tests (DT/Oracles) sont-ils toujours valides
2. Sélection/minimisation : faut-il tout rejouer, ce qui peut être coûteux, et sinon comment sélectionner un sous ensemble pertinent.

Grâce à des plugins comme celui présent dans Eclipse, il est possible de facilement rejouer des tests et ou détecter les couples (DT/Oracles) qui ne sont plus syntaxiquement corrects.

### 2 Assertions :

***asserEquals(a,b)*** : teste si a,b sont des valeurs de type primitif égales ou des objets possédant une méthodes equals

***assertTrue(a)***:teste si a (expression booléenne) est vrai

***assertFalse(a)*** :teste si a (expression booléenne) est vrai

***assertSame(a,b)*** :teste si a et b réfèrent au même objet

***assertNotSame(a,b)*** :teste si a et b ne réfèrent pas au même objet

***assertNull(a)*** : teste si un objet a est null

***assertNotNull(a)***: teste si un objet a n'est pas null

***fail(message)*** : teste si le test doit échouer (levée d'exception)