

# Tests unitaires

---

Gilles QUERRET • Riverside Software

# Test-driven development

- Les frameworks de tests viennent du principe de développement piloté par les tests
    - Commencer par tester
    - Vérifier qu'ils échouent
    - Effectuer les développements
    - Vérifier que les tests passent
    - Refactoriser le code
  - 1994 : premiers concepts Xtreme Programming
  - 2000 : début des xUnit
-

# Ecrire le test

- La spécification :
    - Une méthode à un paramètre INTEGER devant false si la valeur est 42
  - Ecriture du test :
    - **IF** MaClasse:MaMethode(42) VAUT « False » **ALORS** OK **SINON** Echec
  - Première exécution :
    - Méthode inexistante, cela sera reporté en échec
-

# Ecrire le code

- Ecrire le code le plus court possible répondant à la spécification

```
method public logical myMethod(a as int):  
    if a eq 42 then return false.  
    else return true.  
  
end method.
```

```
method public logical myMethod(a as int):  
    return false.  
  
end method.
```

- La spécification donne le résultat pour 42, pas pour tout autre valeur...
  - Pas de spécification, pas de test
-

# Nouveaux tests

- Spécification additionnelle :
    - Tout autre valeur que 42 doit renvoyer TRUE
  - Ecriture des tests complémentaires :
    - **IF** MaClasse:MaMethode(0) VAUT « True » **ALORS** OK **SINON** Echec
    - **IF** MaClasse:MaMethode(10000) VAUT « True » **ALORS** OK **SINON** Echec
  - Première exécution :
    - Les deux nouveaux tests sont en échec
  - Une fois le code corrigé, les tests passent
-

# En bref

- S'assurer de la qualité du code dès le début du projet, et tout au long du projet
  - Confiance dans le résultat du code
    - Si le code est modifié ou réécrit, les tests doivent toujours réussir
  - Les tests peuvent servir de spécifications
-

# Structure des tests

- Tous les frameworks de tests unitaires se basent sur le même principe :
  - Initialisation
  - Exécution de code
  - Assert / Expect

```
PROCEDURE test1:  
  RUN assertTrue (DYN-FUNCT('maFonction')).  
END PROCEDURE.  
PROCEDURE test2:  
  RUN expectStop(TRUE).  
  RUN foo/bar/failure.p.  
END PROCEDURE.
```

---

# En Progress

- 2004 : ProUnit
  - 2010 : OEUnit
  - 2014 : ABLUnit
-



# ABLUnit

- Ajouté en 2014 dans la version 11.4 d'OpenEdge
  - Ecriture des tests sous forme de procédures ou de classes, pouvant être regroupés par suite
  - Exécution directement dans Developer Studio
  - Ou en ligne de commandes (par Ant) dans un outil d'intégration continue
    - Rappel : les tests unitaires doivent être exécutés à chaque commit !
-

# ABLUnit

- Ajouté en 2014 dans la version 11.4 d'OpenEdge
  - Ecriture des tests sous forme de procédures ou de classes, pouvant être regroupés par suite
  - Exécution directement dans Developer Studio
  - Ou en ligne de commandes (par Ant) dans un outil d'intégration continue
    - Rappel : les tests unitaires doivent être exécutés à chaque commit !
-

# Précisions

- Les tests doivent être indépendants
    - L'ordre d'exécution n'est pas garanti
    - Chaque test doit configurer son environnement et le nettoyer
  - Une pratique courante dans les tests est l'utilisation de mock objects (object factices)
    - Lecture dans des fichiers texte / XML / JSON au lieu de base de données
    - Mock SMTP pour intercepter l'envoi de mails
-

# Automatisation

## ■ Tests summary

Showing Last 25 builds ▾

### Top 10 most broken tests

| Test   | Times broken  | Job        | Most recent builds                            |
|--|---------------|------------|---|
| ■ btdebtor_tester <a href="#">Convert_ConvertIntoANewDebtorWithSuppliedIdInUse</a>           | 1 <div></div> | Unit Tests | <a href="#">#1466</a> , <a href="#">#1109</a> |
| ■ btdebtor_tester <a href="#">btdebtor_JustInstantiate</a>                                   | 1 <div></div> | Unit Tests | <a href="#">#1466</a>                         |
| btdebtor_tester <a href="#">Convert_ConvertIntoANewDebtorWithSuppliedId</a>                  | 1 <div></div> | Unit Tests | <a href="#">#1466</a> , <a href="#">#1109</a> |
| btdebtor_tester <a href="#">Convert_ConvertIntoANewDebtor</a>                                | 1 <div></div> | Unit Tests | <a href="#">#1466</a> , <a href="#">#1109</a> |
| profession_tester <a href="#">GetCTBEmployeeProfessions_OneEmployment2Profession</a>         | 1 <div></div> | Unit Tests | <a href="#">#1466</a>                         |
| profession_tester <a href="#">GetCTBEmployeeProfessions_2Employment2Profession</a>           | 1 <div></div> | Unit Tests | <a href="#">#1466</a>                         |
| profession_tester <a href="#">GetCTBEmployeeProfessions_OneEmploymentnoProfession</a>        | 1 <div></div> | Unit Tests | <a href="#">#1466</a>                         |
| profession_tester <a href="#">GetCTBEmployeeProfessions_OneEmployment3Changes2Profession</a> | 1 <div></div> | Unit Tests | <a href="#">#1466</a>                         |
| profession_tester <a href="#">GetCTBEmployeeProfessions_OneEmployment3Profession</a>         | 1 <div></div> | Unit Tests | <a href="#">#1466</a>                         |
| profession_tester <a href="#">GetCTBEmployeeProfessions_OneEmploymentOneProfession</a>       | 1 <div></div> | Unit Tests | <a href="#">#1466</a>                         |

# Tests d'intégration

- **Si** votre application met à disposition des services sur AppServer
  - Alors effectuez un déploiement après chaque build sur un environnement de test dédié
  - Et effectuez les tests uniquement en utilisant les services à disposition
-

# Example

```
class eu.rssw.invoice_tester inherits UnitTest:
  def var h as handle no-undo.

  @Before.
  method override public void Instantiate ( ):
    create server h.
    h:connect("-AppService ...").
  end method.

  @After.
  method override public void DeInstantiate ( ):
    h:disconnect().
    delete object h.
  end method.

  @Test.
  method public test1( ):
    run createCustomer(1, 'CustName', 'Address', output ret).
    assertTrue(ret).
  end method.

  @Test.
  method public test2( ):
    run createCustomer(1, '', 'Address', output ret).
    assertFalse(ret).
  end method.

end class.
```

---