AGL - Les tests logiciels

Amélie Cordier, Marie Lefevre, François Fouquet

Introduction

Les différents types de tests

Comment mettre en œuvre les tests

Les jeux de tests

Quelques réflexions

Lectures d'actualité

Plan

Introduction

Les différents types de tests

Comment mettre en œuvre les tests

Les jeux de tests

Quelques réflexions

Lectures d'actualité

Qu'est-ce qu'un logiciel ?

- Un exécutable...
- ... et son code source
- Des documents de conception et de spécification
- Un cahier des charges ?

L'ensemble de ces éléments permet de définir des tests de diverses natures.

Qu'entend-on par "tester un logiciel" ?

- Vérifier que logiciel fait ce qu'il doit faire
  - Conformité aux cahier des charges
  - Conformité aux documents de spécification
- Vérifier que le logiciel "fonctionne bien"
  - Pas de "plantages" inattendus ou de consommation excessive de ressources
  - Correction du résultat obtenu (1+1 doit être égal à 2).

≪Tester, c'est exécuter le programme dans l'intention d'y trouver des anomalies ou des défauts ≫ - G. Myers, The Art of Software testing

Deux grands principes : Validation et Vérification

- Validation : est-ce que le logiciel réalise les fonctions attendues ?
- Vérification : est-ce que le logiciel fonctionne correctement ?
- Les méthodes de V&V
  - Test statique : lire le code
  - Test dynamique : exécuter le code pour s'assurer de son fonctionnement
  - Vérification symbolique : vérification runtime
  - Vérification formelle : preuve, model-checking, etc.

Plan

Introduction

Les différents types de tests

Comment mettre en œuvre les tests

Les jeux de tests

Quelques réflexions

Lectures d'actualité

Deux grandes catégories de tests

- Les tests fonctionnels :
  - Objectif : vérifier le respect des spécifications
  - Exemples : vérification de la correction, tests de qualité, tests de performances, etc.
- Les tests structurels :
  - Objectif : détecter les erreurs d'implémentation
  - Exemples : débordement de pile, échec d'initialisation, résultat erroné, etc.

Quelques tests...

- Tests unitaires : fonction, module, composant
- Tests d'intégration : test de composition entre plusieurs composants
- Tests de validation (ou de conformité) : adéquation aux spécifications
- Tests de non-régression : vérification que l'évolution du code ne crée pas de nouvelles anomalies
- Test de bon fonctionnement (*test-to-pass*) : les cas de test contiennent des données d'entrée valides
- Test de robustesse (*test-to-fail*) : les cas de test correspondent à des données d'entrée invalides
- Tests de performances : test de charge ou de stress, définissant à la capacité à répondre à une demande de ressources anormale

Plan

Introduction

Les différents types de tests

Comment mettre en œuvre les tests

Les jeux de tests

Quelques réflexions

Lectures d'actualité

Comment concevoir les tests ?

- Absence de tests (à ne pas faire)
- Tests manuels (printf)
- Tests formels (vérification de la conformité à un modèle)
- Preuves
- Utilisation d'outils dédiés (JUnit, CUnit, etc.)

Quelles compétences sont requises pour créer des tests ?

- Être créatif et imaginer des scénarios pour mettre un logiciel en défaut. Exemple : je ne mets pas de "@" dans mon adresse email.
- Il faut imaginer des jeux de tests pour vérifier l'ensemble des fonctionnalités et des contraintes.
- Il est important que les personnes qui codent et les personnes qui testent soient différentes.

Quand appliquer les tests ?

- Les tests peuvent (doivent) s'appliquer à chaque étape du cycle de vie du logiciel :
  - Spécification
  - Conception
  - Implémentation
- Pour chaque phase de test, le testeur doit élaborer des rapports de tests :
  - Exécuter les tests spécifiés
  - Analyser les résultats obtenus (dans une phase distincte)
  - Émettre des fiches de non conformité, si nécessaire.

Plan

Introduction

Les différents types de tests

Comment mettre en œuvre les tests

Les jeux de tests

Quelques réflexions

Lectures d'actualité

Pourquoi définir un jeu de tests ?

- Pour tester les différents cas d'échecs possibles
- Pour factoriser les tests
- Pour définir des séquences réutilisables
- Pour systématiser les tests

Comment définir un jeu de tests ?

- Imaginer les scénarios de tests possibles
- Définir un ou des tests pour chacun des scénarios
- Fabriquer un "oracle". L'oracle connaît la bonne réponse
- Tester le programme : le résultat obtenu doit correspondre au résultat prédit par l'oracle

Exemple de jeu de test : 14 tests pour traiter la classe triangle

Cet exemple est extrait de G.J. Myers, "The Art of Software Testing".

- Cas scalène valide (1,2,3 et 2,5,10 ne sont pas valides)
- Cas équilatéral valide
- Cas isocèle valide (2,2,4 n'est pas valide)
- Cas isocèle valide avec les trois permutations (e.g. 3,3,4 ; 3,4,3 ; 4,3,3)
- Cas avec une valeur à 0
- Cas avec une valeur négative

Exemple de jeu de test : 14 tests pour traiter la classe triangle

suite...

- Cas ou la somme de deux entrées est égale à la troisième entrée
- 3 cas pour le test 7 avec les trois permutations
- Cas où la somme de deux entrées est inférieure à la troisième entrée
- 3 cas pour le test 9 avec les trois permutations
- Cas avec les trois entrées à 0
- Cas avec une entrée non entière
- Cas avec un nombre erroné de valeurs (e.g. 2 entrées, ou 4)
- Pour chaque cas de test, avez-vous défini le résultat attendu ?

Plan

Introduction

Les différents types de tests

Comment mettre en œuvre les tests

Les jeux de tests

Quelques réflexions

Lectures d'actualité

Quel est le coût du test ?

- Le test représente 30 à 40% des coûts de développement d'un logiciel
- Le test représente en moyenne 1/3 du temps de développement.

Les tests sont-ils vraiment importants ?

- Le test représente une phase importante du développement logiciel
- Mettre en place une procédure rigoureuse de tests (vérification et validation) est essentiel dans le cadre d'une démarche qualité
- Le test a mauvaise réputation car il est long, coûteux, et qu'il met souvent en retard les projets... mais... il permet souvent d'éviter des anomalies qui seraient encore plus coûteuses.

Les tests sont-ils difficiles à mettre en oeuvre ?

Mettre en oeuvre des tests logiciels rigoureux est une démarche compliquée...

- Il est impossible de réaliser des jeux de tests exhaustifs
- Il est très difficile de faire des tests structurels complets
- Comme l'exhaustivité n'est pas possible, le choix des tests à effectuer doit être pertinent
- Le processus de test est souvent frustrant : un bon test est un test qui soulève une erreur
- Les erreurs peuvent être dues à une incompréhension des spécifications ou à un mauvais choix d'implémentation
- Mettre en place un processus de tests est compliqué, en particulier dans les grosses applications

Plan

Introduction

Les différents types de tests

Comment mettre en œuvre les tests

Les jeux de tests

Quelques réflexions

Lectures d'actualité

Vous avez dit TDD ? Et Agile programming ? Késako ?

- http ://agileprogramming.org/
- http ://agilemanifesto.org/
- http ://www.agiledata.org/essays/tdd.html

Références

- sebastien.bardin.free.fr/cours-BL.pdf
- http ://www-spiral.lip6.fr/ vmm/fr/Enseignement/DESS/-Test/Cours/C1.pdf