Tests et qualité du code
Tests fonctionnels (Boite noire)

Youness LAGHOUAOUTA

Institut National des postes et télécommunications
laghouaouta@inpt.ac.ma

Fevrier 2022

Typologie des tests

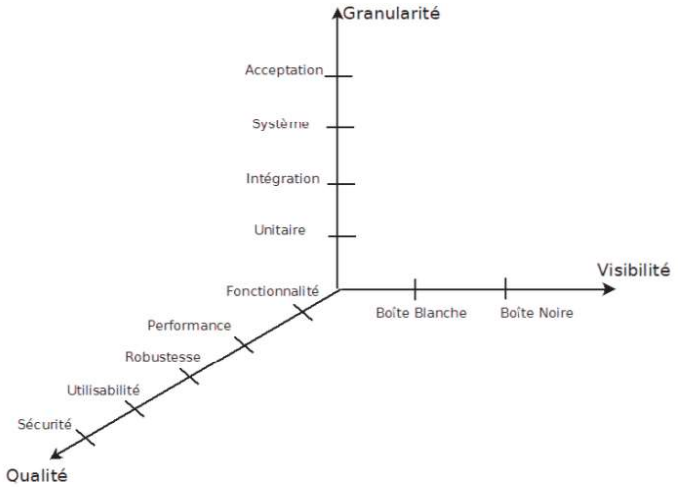

Figure: Typologie des tests [1]

Test fonctionnel (Boite noire)

- Un test qui ignore les mécanismes internes d'un système et qui vérifie si les sorties obtenues sont bien celles prévues pour des entrées et des conditions d'exécution données [2]
- Sélection des tests à partir d'une spécification du système (formelle ou informelle), sans connaissance de l'implantation
- Possibilité de construire les tests avant l'implémentation
- Permet d'assurer l'adéquation spécification – code, mais aveugle aux défauts fins de programmation

Test fonctionnel (Boite noire)

- Pas trop de problème d'oracle pour le CT
- Mais problème de la concrétisation
- Quand l'utiliser ?
  - test système, test d'intégration
  - test unitaire
  - leur planification et conception peut commencer tôt dans le processus de développement du logiciel, i.e., dès que les spécifications sont disponibles

Sélection des tests

- L'efficacité du test dépend crucialement de la qualité des CT/DT
- Ne pas manquer un comportement fautif
- MAIS trop de CT/DT est coûteux (design, exécution, stockage, ...)
- L'exhaustivité est inenvisageable
- Méthodes de sélection des tests boite noire : comment trouver des DT pertinentes et pas trop nombreuses ?
  - Analyse partitionnelle
  - Test aux limites
  - Test pairwise
  - Graphe cause-effet
  - Etats/transitions
  - ...

Analyse partitionnelle

- Diviser le domaine des entrées en un nombre fini de classes (d'équivalence) tel que le programme réagisse de la même façon pour toutes les valeurs d'une classe
- Pour chaque classe d'équivalence, toutes les entrées génèrent le même comportement (détection du même défaut)
- Comment s'y prendre ?
  - Identifier les classes d'équivalence
    - Sur la base des conditions sur les entrées/sorties
    - En prenant des classes d'entrées valides et invalides
  - Définir un ou quelques CTs pour chaque classe

Exemple 1

- Supposons que la donnée à l'entrée est un réel supérieur ou égale à 0
- Classes d'équivalence :
  1. entrée $>= 0$
  2. entrée < 0
  3. entrée n'est pas un réel
- Données de tests représentatives :
  1. 9.34, 100, 0
  2. -1, -7.14, -300
  3. "a", "9.3e", ""

Exemple 2

- Spécification : en fonction de l'âge d'une personne, le programme décide si la personne est majeure ou non.
- Incertitude :
  - A quel âge on est adulte?
  - Que se passe-t-il pour une entrée invalide?
  - Y a-t-il un âge limite?

Exemple 2

- Entrée :
  - CE1 : dans [0,18[
  - CE2 : dans [18,130]
  - CE3 : n'est pas dans l'intervalle [0,130] ou n'est pas un entier
- Sortie :
  - CE4 : "Majeure"
  - CE5 : "Mineure"
  - CE6 : Vide
- Erreur :
  - CE7 : "Entrée invalide"
  - CE8 : Vide

Exemple 2

- Cas de test :
  1. Entrée : 6; Sortie : "Mineure"; Erreur : Vide
  2. Entrée : 33; Sortie : "Majeure"; Erreur : Vide
  3. Entrée : "abc"; Sortie : Vide; Erreur : "Entrée invalide"
- Les cas de test couvrent l'ensemble des classes d'équivalence

Sélection des classes d'équivalence

- Entrée appartenant à un intervalle (ou liste ordonnées de valeurs) :
  - Une classe pour les valeurs inférieures
  - Une classe pour les valeurs supérieures
  - Une classe pour les valeurs valides
- Entrée est un ensemble de valeurs
  - Une classe avec l'ensemble vide
  - Une classe avec des valeurs en dehors de l'ensemble
  - Une classe avec des valeurs valides
- Entrée est une contrainte
  - Une classe avec la contrainte respectée
  - Une classe avec la contrainte non respectée

Classes d'équivalence et multi-variables

Test aux limites

- Il ne s'agit pas d'une méthode de sélection des tests, mais d'une tactique pour améliorer l'efficacité des DT produites par d'autres méthodes
- Intuition : de nombreuses erreurs se produisent dans les cas limites
  - Pour chaque donnée en entrée : déterminer les bornes du domaine
  - Prendre des valeurs sur les bornes et juste un peu autour
- Exemple :
  - Accéder à la n+1ème case d'un tableau de n cases

Test aux limites

- Pour un intervalle : les 2 limites, et 4 valeurs correspondant aux valeurs des limites $\pm$ le plus petit delta possible
- Variable appartient à un ensemble ordonné de valeurs : le premier, le second, l'avant dernier et le dernier
- Une condition d'entrée spécifie un nombre de valeurs : cas de test à partir du nombre minimum et maximum de valeurs, et des tests pour des nombres de valeurs hors limites invalides
- Chaine de caractères (max n caractères) : chaîne vide, chaînes avec un seul caractère, n-1, n, n+1 caractères
- Tableau : tableau vide, avec un seul élément, avec n-1 ou n éléments

Analyse partitionnelle et test aux limites

- Analyse partitionnelle :
  - Réduction du nombre de cas de test
  - Choix des classes délicat
- Test aux limites :
  - Heuristique solide pour le choix des données au sein des classes
  - Le test aux limites produit à la fois des cas de test nominaux (dans l'intervalle) et de robustesse (hors intervalle)
  - Nécessite la présence d'une relation d'ordre entre les données considérées
  - Explosion combinatoire des données de test

Méthode combinatoire : approche pairwise

Test exhaustif souvent impraticable

Tester un sous-ensemble des combinaisons de valeurs tel que chaque combinaison de n variables est testée

- Approche pairwise (n=2) : sélectionner les DT pour couvrir toutes les paires de valeurs :
  - Nombre paires beaucoup plus petit que le nombre de combinaisons
  - Une majorité de défauts sont souvent détectables par des combinaisons de 2 valeurs de variables
  - Un seul test peut couvrir plusieurs paires
  - Grande réduction si beaucoup de variables à petits domaines

Inconvénients

- Aucune chance de détecter des bugs demandant des combinaisons de plus de 2 valeurs
- Le résultat attendu de chaque test doit être fournis manuellement

Exemple 1

- 3 variables : a,b,c
  - $a \in \{1, 2\}$
  - $b \in \{X, Y, Z\}$
  - $c \in \{true, false\}$
- Pour couvrir toutes les combinaisons : 12 tests
- Pour couvrir toutes les paires de valeurs : 6 tests

a	b	c	paires couvertes
1	X	True	(1,X) (1,True) (X,true)
1	Y	False	(1,Y) (1,False) (Y,False)
1	Z	True	(1,Z) (1,True) (Z,True)
2	X	False	(2,X) (2,False) (X,False)
2	Y	True	(2,Y) (2,True) (Y,True)
2	Z	False	(2,Z) (2,False) (Z,False)

Exemple 2

- Confuguration d'une plateforme : 5 paramètres
  - SE: Windows XP, Apple OS X, Red Hat Linux
  - Navigateur: Internet Explorer, Firefox
  - Protocole: IPv4, IPv6
  - CPU: Intel, AMD
  - SGBD: MySQL, Sybase, Oracle

Exemple 2

- Confuguration d'une plateforme : 5 paramètres
  - SE: Windows XP, Apple OS X, Red Hat Linux
  - Navigateur: Internet Explorer, Firefox
  - Protocole: IPv4, IPv6
  - CPU: Intel, AMD
  - SGBD: MySQL, Sybase, Oracle
- Nombre de combinaisons : 72
- Nombre de combinaison de 2 éléments : $\frac{2}{5}\mathrm{C} = \frac{5!}{2!(5-2)!} = 10$
- Nombre de paires N : $\frac{2}{5}\mathrm{C} * 2^2 < \mathrm{N} < \frac{2}{5}\mathrm{C} * 3^2 \rightarrow 57$
- Nombre minimum de tests ?

Exemple 2

SE	Navigateur	Protocole	CPU	SGBD
Windows XP	Internet Explorer	IPv4	Intel	MySQL
Red Hat Linux	Firefox	IPv6	AMD	MySQL
Apple OS X	Internet Explorer	IPv4	AMD	Oracle
Apple OS X	Firefox	IPv6	Intel	MySQL
Apple OS X	Internet Explorer	IPv6	Intel	Sybase
Windows XP	Firefox	IPv6	AMD	Oracle
Red Hat Linux	Firefox	IPv4	Intel	Sybase
Red Hat Linux	Internet Explorer	IPv6	Intel	Oracle
Windows XP	Internet Explorer	IPv4	AMD	Sybase

Exemple 3

- On veut tester l'impression de fichiers depuis plusieurs applications sur des OS et via des réseaux différents.

SE	Réseau	Imprimante	Application
Windows10	IP	HP	Word
Linux	Wifi	Canon	Excel
Mac OS	Bluetooth		PowerPoint
			PDF Reader

Exemple 3

- On veut tester l'impression de fichiers depuis plusieurs applications sur des OS et via des réseaux différents.

SE	Réseau	Imprimante	Application
Windows10	IP	HP	Word
Linux	Wifi	Canon	Excel
Mac OS	Bluetooth		PowerPoint
			PDF Reader

- Nombre de combinaisons : 72
- Nombre de combinaison de 2 éléments : ${}^{2}_{4}\mathrm{C} = \frac{4!}{2!(4-2)!} = 6$
- Nombre de paires N : $6 * 2^2 < N < 6 * 4^2 \rightarrow 53$
- Nombre minimum de tests : 12

Exemple 3

OS	Reseau	Imprimante	Application
Windows10	Wifi	HP	PDF Reader
Mac OS	IP	Canon	Word
Linux	IP	HP	Excel
Linux	Wifi	Canon	PowerPoint
Windows10	Bluetooth	Canon	Excel
Windows10	Bluetooth	HP	PowerPoint
Linux	Bluetooth	Canon	PDF Reader
Mac OS	Wifi	HP	Excel
Mac OS	IP	HP	PowerPoint
Mac OS	IP	Canon	PDF Reader
Mac OS	Bluetooth	HP	Word
Windows10	IP	Canon	Word
Linux	Wifi	Canon	Word

Approche pairwise

- Peut être utilisé avec l'analyse partitionnelle de plusieurs paramètres
- Génération outillée :
  - Liste des outils : http://www.pairwise.org
  - Outil en ligne : https://pairwise.yuuniworks.com/
- Combinaison aléatoire des valeurs
- Résultat attendu à fournir manuellement
- Extension : combinaisons de trois valeurs, quatre valeurs... $\rightarrow$ Explosion du nombre de tests

Réferences

1 Blasquez, Isabelle, Hervé Leblanc, and Christian Percebois. "Les tests dans le développement logiciel, du cycle en V aux méthodes agiles." Revue des Sciences et Technologies de l'Information-Série TSI: Technique et Science Informatiques 36.1-2 (2017): 7-50.

2 IEEE standard glossary of software engineering terminology. IEEE Std 610.12-1990.