

Rédigé par Yannick Le Nir	Ref : INTO-TEST
Revu par : l'équipe pédagogique	Crée le 11/11/2012

1	Présentation.....	2
2	Historique.....	2
3	Citations.....	2
4	Quelques chiffres.....	2
5	Types de tests.....	2
6	Déroulement classique d'un test .....	3
7	Sélection des tests.....	3
7.1	Sélection des tests en BB :.....	3
7.2	Sélection des tests en BN.....	4

# 1 Présentation

Le test logiciel est une composante essentielle du génie logiciel dont l'importance n'a cessé de se vérifier tout au long de l'histoire de l'informatique. Il peut être décomposé en deux grands domaines: la validation et la vérification. La validation vérifie l'exactitude du logiciel avec les exigences du client alors que la vérification consiste à déterminer si le logiciel fonctionne correctement. A l'origine la vérification se devait d'être formelle et garantir la correction d'un logiciel quelque soit son utilisation (variation des entrées). Devant la difficulté d'une telle tâche, le test consiste lorsqu'il est mis en œuvre, la plupart du temps à simplement vérifier le comportement du logiciel dans un certain nombre de cas d'utilisation, plus ou moins exhaustifs, visant plus à limiter le nombre de bugs qu'à complètement les éliminer.

# 2 Historique

Les erreurs logiciels (bugs) jalonnent l'histoire de l'informatique et sont autant de d'arguments en faveur du test logiciel. On en trouve dans tous les domaines :

- calculateurs des voitures
- logiciels bancaires (2006, 2010)
- logiciels critiques (Ariane 5, Mars Climate Orbiter,...)
- logiciels très critiques (Therac-25, missiles Patriot,...)

# 3 Citations

E.W. Dijkstra (Notes on Structured Programming, 1972)

« Testing can only reveal the presence of errors but never their absence. »

G.J. Myers (The Art of Software Testing, 1979)

« Tester, c'est exécuter le programme dans l'intention d'y trouver des anomalies ou des défauts. »

Donald Knuth (1977)

« Beware of bugs in the above code ; I have only proved it correct, not tried it. »

Thomas A. Henzinger (2001)

« It has been an exciting twenty years, which has seen the research focus evolve [...] from a dream of automatic program verification to a reality of computer-aided [design] debugging. »

# 4 Quelques chiffres

Coût des bugs :

64 milliards \$/an aux US  
impacts humains, environnementaux,...

Coût du test logiciel :

10 milliards \$/an aux US  
plus de 50 % du développement d'un logiciel critique  
en général 30 % du développement d'un logiciel standard

# 5 Types de tests

Il y a plusieurs type de tests qui interviennent tout au long du développement logiciel :

- Tests unitaires : test de chaque module du logiciel indépendamment des autres. La granularité d'un module peut varier selon les cas (fonction, package, classe, ...)
- Tests d'intégration : test des différents modules les uns avec les autres

## INTRODUCTION AU TEST LOGICIEL

- Tests systèmes ou de validation : valider le logiciel par rapport aux spécifications du client en évaluant également des composantes spécifiques telles que la sécurité ou les performances
- Tests de régression : vérifier que les mise à jours du logiciel n'ont pas introduits de bugs.

## 6 Déroulement classique d'un test

Quelque soit le type de test, on retrouve majoritairement la procédure suivant :

1. Choisir un cas de test (CT)
2. Instancier une donnée de test (DT) à partir du CT
3. Prévoir le résultat attendu à partir de la DT (Oracle)
4. Exécuter le programme sur la DT
5. Comparer le résultat obtenu avec l'oracle

On distingue néanmoins les tests faits à partir de spécifications: Boîte Noir (BN) et les test faits à partir du code source : Boîte Blanche (BB). Les tests BN ne nécessites pas de disposer du code source mais restent approprié au tests systèmes ainsi qu'à certains tests unitaires. Les tests BB, permettent de découvrir des défauts fins de programmation mais pas les fonctionnalités absentes. Une bonne suite de tests mélange donc habilement BB et BN. Il existe enfin une dernière manière d'effectuer des tests, dite aléatoire ou statistique mais nous ne la détaillerons pas dans ce cours.

## 7 Sélection des tests

Selon le type de tests (BB ou BN), la sélection des tests à réaliser va dépendre de critères différents.

### 7.1 Sélection des tests en BB :

Dans le cadre des BB, le code source est disponible et nous allons nous en servir afin d'explorer les différents comportements possible du programme. L'objectif est de déterminer les chemins minimaux entre l'entrée et la sortie du programme afin d'assurer que :

- toutes les conditions d'arrêts des instructions itératives (boucles) ont été testées.
- Toutes les branches des instructions conditionnelles (if) ont été testées.
- Toutes les structures de données internes (variables locales) ont été testées.

Pour ce faire, une méthode consiste à déterminer la mesure de complexité de Mac Cabr. Cette mesure se calcule à partir du graphe de flot du programme par la formule suivante :

Nombre cyclomatique = Nb Arcs - Nb Noeud + 2

Cette valeur détermine le nombre de chemins minimums (égal au nombre de régions du graphe de flot) qu'il va falloir tester (un test par chemin pour tester toutes les possibilités du programme)

### 7.2 Sélection des tests en BN

Dans le cadre des BN, le programme est vu sous un aspect fonctionnel au lieu de structurel car on a pas accès au code source. Il faut alors partitionner le domaine de définition des entrées en plusieurs classes et générer des cas de test aux limites des classes