

INF352
2023-2024



INF352 : TESTS LOGICIELS : WHITE BOX TESTING

Présentation

Mai 2024

NGOGANG TCHATCHOUANG ANDY BRAYAN
NGOUOT LAILA
DJOUATSA RENELLE
JIEUKENG MARIELLE CLOWER

21Y116
21Y129
20Y114
20U2623

Sous l'encadrement de : Dr KIMBI Xaviera

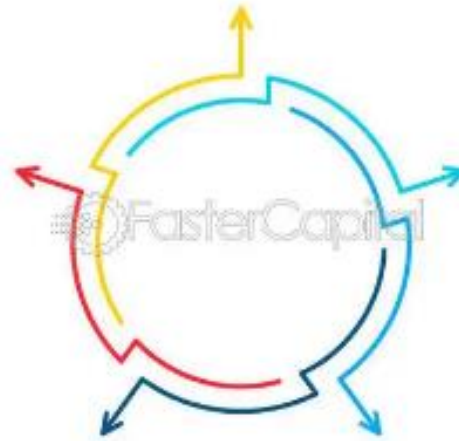
Sommaire



0 1 Introduction aux tests en boîte blanche

0 2 Le but des tests en boîte blanche

0 3 Techniques et outils pour les tests en boîte blanche



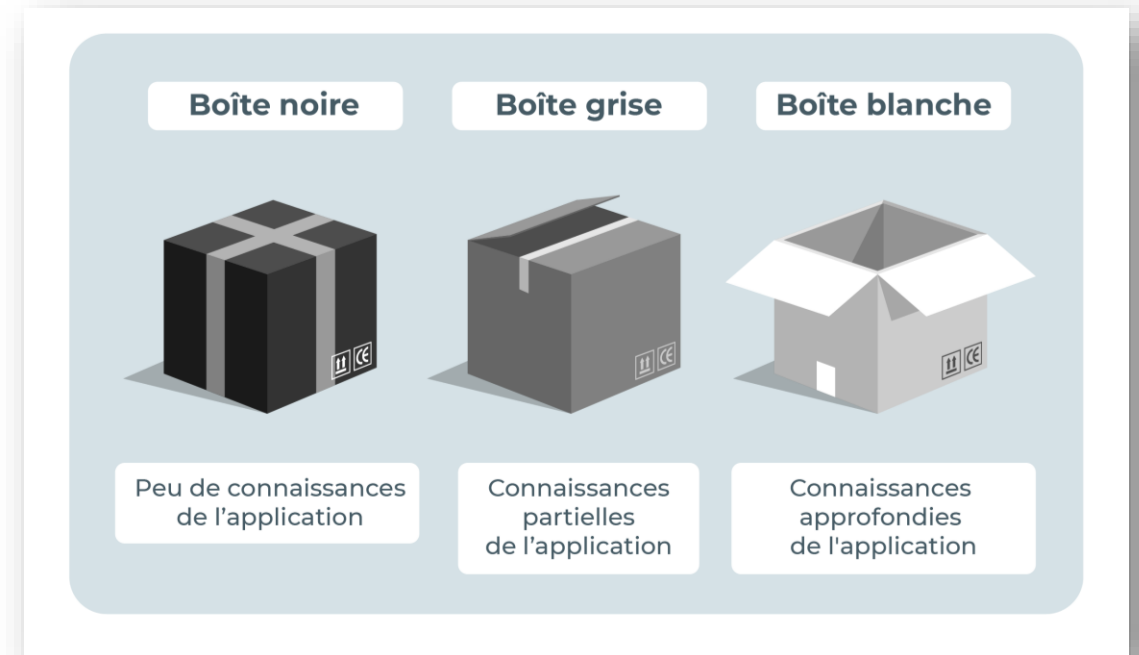
0 4 Avantages et limites des tests en boîte blanche

0 5 Étude de cas

Introduction



- Les tests par boîte blanche, également appelés tests de structure ou de code, sont une méthode de test qui examine la structure interne d'un logiciel pour garantir qu'il fonctionne comme prévu.
- Contrairement aux tests par boîte noire, qui traitent le logiciel comme une boîte noire sans connaître son implémentation interne, les tests par boîte blanche utilisent la connaissance du code source pour identifier les chemins et les conditions qui doivent être testés.



Objectifs des tests par boîte blanche



Les objectifs que nous retrouvons sont :

- **Identifier les erreurs logiques et de structure dans le code.**
- **Assurer une couverture de code suffisante.**
- **Améliorer la maintenabilité et la lisibilité du code.**
- **Détecter les vulnérabilités de sécurité potentielles.**

Techniques de tests par boîte blanche



Parmi les techniques de tests par boîte blanche, on retrouve :

- **Couverture de chemin**
- **Couverture de branche**
- **Couverture de condition**
- **Analyse du flux de données**
- **Tests de mutation**

Couverture de chemin

- La couverture de chemin consiste à tester tous les chemins possibles à travers le code source. Il existe différents types de couverture de chemin, tels que la couverture de chemin linéaire, cyclomatique et maximale. Par exemple, pour un scénario de test, on pourrait avoir comme objectif de valider tous les chemins d'exécution possibles à travers une fonction.

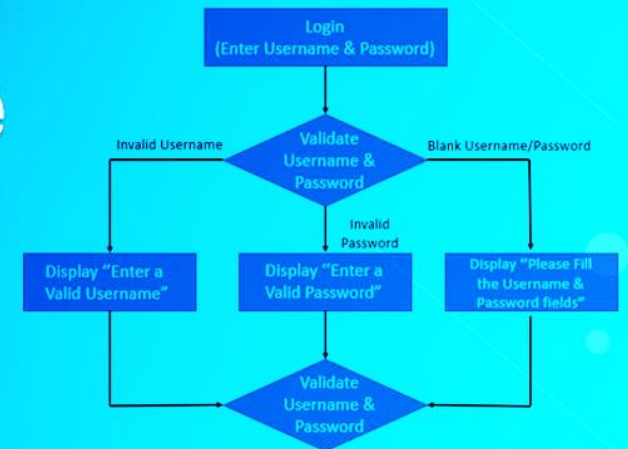


Couverture de branche



- La couverture de branche vise à tester toutes les branches de décision et conditionnelles du code source. On distingue la couverture de branche de décision (détermination de toutes les branches de code contrôlées par des instructions de branchement) et la couverture de branche conditionnelle modifiée (test de toutes les combinaisons possibles des conditions logiques).

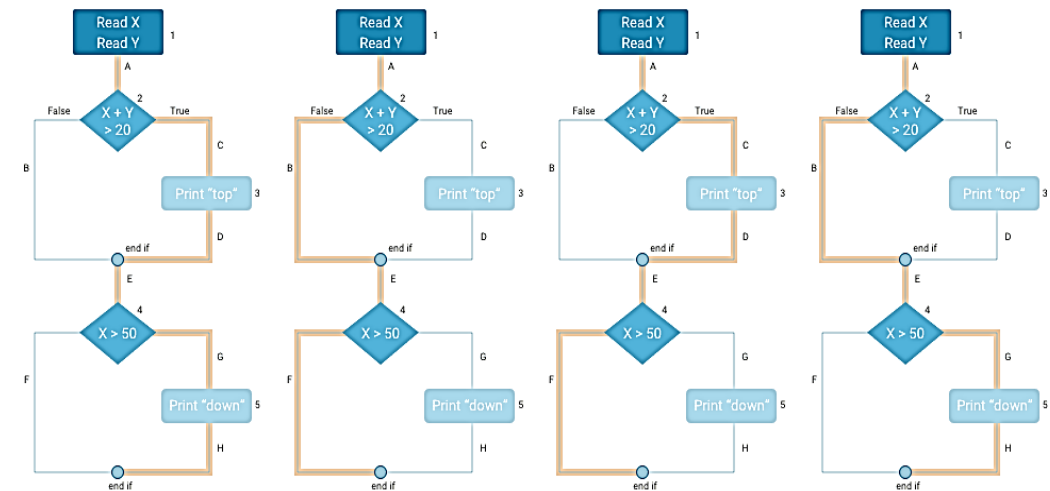
Branch Coverage



www.educba.com

Couverture de condition

- La couverture de condition consiste à tester toutes les conditions logiques à l'intérieur du code source. Il existe différents types de couverture de condition, tels que la couverture de condition conjonctive (AND), disjonctive (OR) ou modifiée. Les stratégies varient en fonction des conditions logiques impliquées.



Couverture de flèche

- La couverture de flèche se base sur **le graphe de flux de contrôle du logiciel** pour générer des tests permettant de couvrir tous les chemins possibles à travers le code. Cette technique permet de s'assurer que chaque instruction est exécutée au moins une fois.

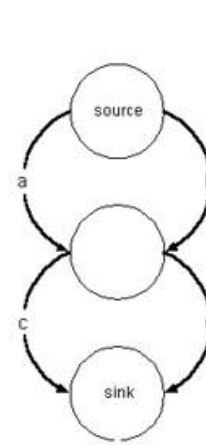


Fig 1. Simplified Control Flowgraph

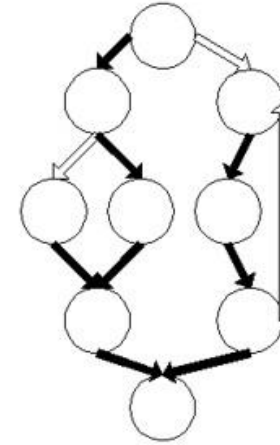


Fig 2. Flowgraph with marked edges

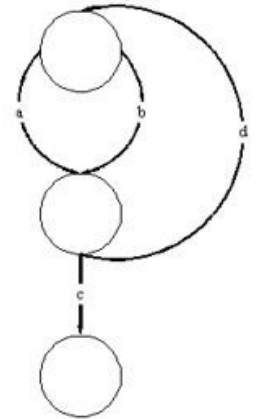


Fig 3. Simplified Flowgraph with Looping

Analyse du flux de données

- L'analyse du flux de données consiste à identifier les différents chemins de données possibles dans le code source. Cela permet de tester le traitement des données et de vérifier que les données sont correctement manipulées par le programme.

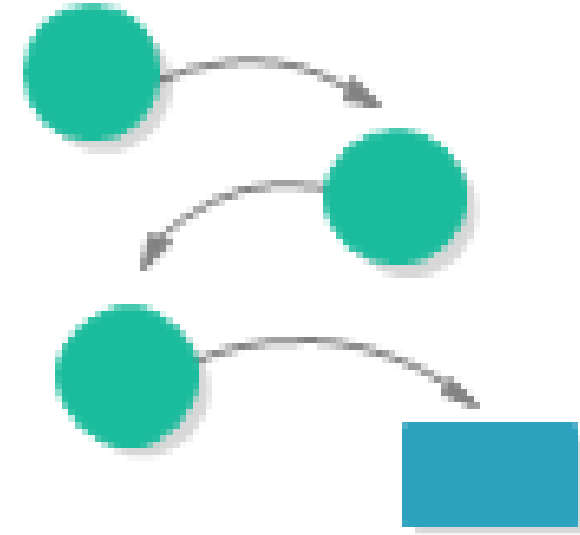


Diagramme de flux de données

Tests de mutations



- Les tests de mutation consistent à introduire des défauts intentionnels dans le code source pour évaluer la qualité des tests. Cette technique permet de tester la robustesse des tests existants en simulant des erreurs potentielles dans le code.



Avantages & inconvénients des tests par boîte blanche



Parmi les avantages, on retrouve :

- **Détection des erreurs internes** : Les tests par boîte blanche permettent de détecter des erreurs internes telles que des boucles infinies, des erreurs de syntaxe ou de logique dans le code source. En analysant la structure interne du logiciel, ces tests peuvent identifier les erreurs dès les premières étapes du processus de développement, ce qui facilite leur résolution rapide.
- **Couverture complète du code** : Les tests par boîte blanche garantissent une couverture exhaustive du code. Les techniques telles que la couverture de chemin, la couverture de branche et la couverture de condition permettent de s'assurer que toutes les parties du code sont testées. Cela réduit le risque de laisser des parties non testées qui pourraient contenir des erreurs.

Avantages et inconvénients des tests par boîte blanche



- **Identification des zones critiques:** En analysant la structure interne du code, les tests par boîte blanche permettent d'identifier les zones critiques du logiciel. Ces zones correspondent souvent aux parties les plus complexes ou les plus sensibles du code. En se concentrant sur ces zones, les tests peuvent fournir une assurance supplémentaire quant à la qualité et à la robustesse des fonctionnalités essentielles.
- **Aide à la maintenance:** Les tests par boîte blanche facilitent la maintenance du logiciel. En comprenant la structure interne du code, les développeurs peuvent localiser plus facilement les parties spécifiques du code qui nécessitent une modification lors de l'introduction de nouvelles fonctionnalités ou de la correction de bugs. Cela permet un processus de maintenance plus efficace et réduit les risques de régressions.

Avantages et inconvénients des tests par boîte blanche



Parmi les inconvénients, nous avons :

- **Connaissance approfondie du code requise** : Les tests par boîte blanche exigent une compréhension approfondie du code source et de ses structures internes. Cela signifie que les testeurs doivent avoir une expertise technique solide pour concevoir et exécuter des tests efficaces. L'accès à des personnes ayant une connaissance avancée de la programmation peut être nécessaire pour tirer pleinement parti de cette approche de test.
- **Limitations de la couverture** : Bien que les tests par boîte blanche visent à obtenir une couverture complète du code, il peut être difficile de tester toutes les combinaisons possibles, en particulier dans les systèmes logiciels complexes. Certains chemins d'exécution peuvent être difficiles à atteindre ou à tester, ce qui peut entraîner une couverture limitée.

Avantages et inconvénients des tests par boîte blanche



- **Sensibilité aux changements de l'implémentation** : Les tests par boîte blanche sont étroitement liés à l'implémentation interne du code. Lorsque le code évolue, les tests doivent être ajustés en conséquence pour refléter les modifications. Cela peut entraîner une maintenance continue des tests pour assurer leur pertinence et leur validité.

ETUDE DE CAS : Tests d'une application de commerce électronique



- Dans cette étude de cas, nous allons examiner l'application des techniques de tests par boîte blanche pour tester une application de commerce électronique. Notre objectif est de garantir que toutes les fonctionnalités clés de l'application sont correctement testées en examinant le code source et la structure interne de l'application.

ETUDE DE CAS : Tests d'une application de commerce électronique



- **Couverture de chemin :**

Pour notre application de commerce électronique, nous utilisons la technique de la couverture de chemin pour tester les différentes fonctionnalités. **Par exemple**, supposons que nous ayons une fonctionnalité permettant aux utilisateurs de passer une commande. Nous identifions les différents chemins d'exécution possibles dans cette fonctionnalité, tels que le cas où le panier est vide, le cas où les informations de paiement sont incorrectes, le cas où les articles sont en rupture de stock, etc. Nous créons des scénarios de test pour couvrir tous ces chemins et nous nous assurons que chaque partie du code est correctement testée.

ETUDE DE CAS : Tests d'une application de commerce électronique



- **Couverture de branche**

La technique de la couverture de branche est utilisée pour tester les différentes branches de décision et de condition dans notre application de commerce électronique. **Par exemple**, supposons que nous ayons une fonctionnalité permettant aux utilisateurs de filtrer les produits par catégorie. Nous identifions les différentes branches de décision, telles que la sélection d'une seule catégorie, la sélection de plusieurs catégories, ou aucune sélection de catégorie. Nous créons ensuite des tests pour couvrir chaque branche de décision et nous nous assurons que toutes les conditions sont correctement évaluées.

ETUDE DE CAS : Tests d'une application de commerce électronique



- **Couverture de condition**

Pour tester les conditions logiques dans notre application de commerce électronique, nous utilisons la technique de la couverture de condition. **Par exemple**, supposons que nous ayons une fonctionnalité permettant aux utilisateurs de bénéficier d'une réduction sur leur commande en saisissant un code promotionnel. Nous identifions les différentes conditions logiques, telles que la validité du code promotionnel, les restrictions d'utilisation, etc. Nous concevons des scénarios de test pour couvrir toutes les conditions possibles et nous nous assurons que chaque condition est correctement vérifiée.

ETUDE DE CAS : Tests d'une application de commerce électronique



- **Couverture de flèche**

La technique de la couverture de flèche est utilisée pour tester les différents chemins possibles dans le flux de contrôle de notre application de commerce électronique. **Par exemple**, supposons que nous ayons une fonctionnalité permettant aux utilisateurs de suivre leur commande. Nous construisons un graphe de flux de contrôle qui représente toutes les actions et les différentes transitions possibles dans cette fonctionnalité. Ensuite, nous générons des tests pour couvrir tous les chemins possibles du graphe de flux de contrôle, en nous assurant que chaque itération, chaque boucle et chaque condition sont testées.

ETUDE DE CAS : Tests d'une application de commerce électronique



- **Couverture de flèche**

La technique de la couverture de flèche est utilisée pour tester les différents chemins possibles dans le flux de contrôle de notre application de commerce électronique. **Par exemple**, supposons que nous ayons une fonctionnalité permettant aux utilisateurs de suivre leur commande. Nous construisons un graphe de flux de contrôle qui représente toutes les actions et les différentes transitions possibles dans cette fonctionnalité. Ensuite, nous générons des tests pour couvrir tous les chemins possibles du graphe de flux de contrôle, en nous assurant que chaque itération, chaque boucle et chaque condition sont testées.

Défis rencontrés et leçons apprises



- **Compréhension du code source** : Pour appliquer efficacement les techniques de tests par boîte blanche, il est essentiel de bien comprendre le code source de l'application. Une connaissance approfondie du code facilite l'identification des chemins, des branches et des conditions à tester.
- **Gestion des données de test** : Les tests par boîte blanche nécessitent souvent la création de données de test spécifiques pour couvrir différentes conditions. La gestion de ces données de test peut être un défi en termes de création, de maintenance et de gestion des ensembles de données de test.
- **Maintenance des tests** : Lorsque le code évolue, les tests par boîte blanche doivent également être mis à jour en conséquence. Une bonne gestion des tests et une documentation claire sont nécessaires pour faciliter la maintenance à long terme.
- **Couverture exhaustive** : Il peut être difficile de garantir une couverture exhaustive de toutes les branches, conditions et chemins possibles, en particulier dans les applications complexes. Il est important de concevoir des stratégies de test efficaces pour maximiser la couverture tout en optimisant les ressources disponibles.



Conclusion



Merci pour votre aimable attention
