**WHITE-BOX TESTING TECHNIQUES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **RÉPUBLIQUE DU CAMEROUN**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** | **C:\Users\ddd\Desktop\photo\Opera Instantané_2022-02-08_190740_www.cameroondesks.com.pngREPUBLIC OF CAMEROON**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |
| **PAIX – TRAVAIL – PATRIE**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** | **PEACE – WORK – FATHERLAND**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |
| **MINISTÈRE DES ENSEIGNEMENTS SUPERIEURE**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** | **MINISTRY OF HIGHER**  **EDUCATION**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |
|  |  |
| **UNIVERSITE DE YAOUNDE**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** | **UNIVERSITY OF YAOUNDE**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |

**Rédigé et présenté par**:

**Sous l’encadrement**

**Mme REGIS ATEMENGUE**

Année académique 2023-2024

* JIEUKENG MARIELLE CLOWER 20U2623
* NGOGANG TCHATCHOUANG ANDY BRAYAN 21Y116
* DJOUASTA RENELLE 21Y114
* NGOUOT LAÏLA 21Y129

**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.

1. **OBJECTIFS DES TESTS PAR BOITE BLANCHE**

• 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.

1. **TECHNIQUES DE TESTS PAR BOITE BLANCHE**

Les tests par boîte blanche sont des tests qui sont conçus en se basant sur une connaissance interne du code source d'un logiciel. Ces tests visent à vérifier la logique interne du programme et à s'assurer que toutes les branches et conditions du code sont correctement gérées. Parmi les techniques de tests par boîte blanche, on retrouve :

1. **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.

1. **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).

1. **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.

1. **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.

1. **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.

1. **Tests de Mutation :**

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.

Les techniques de tests par boîte blanche offrent une approche approfondie pour tester la logique interne d'un logiciel. En combinant différentes stratégies de couverture de chemin, de branche, de condition, de flèche, d'analyse du flux de données et de tests de mutation, les testeurs peuvent s'assurer de la fiabilité et de la robustesse d'une application. Ces techniques permettent de détecter efficacement les erreurs et les défauts potentiels dans le code source, ce qui contribue à améliorer la qualité du logiciel. Il est donc essentiel d'intégrer ces techniques dans une stratégie globale de test pour garantir la performance et la stabilité des applications logicielles.

1. **AVANTAGES ET INCONVENIENTS DES TESTS PAR BOITE BLANCHE**

Les tests par boîte blanche sont une approche de test logiciel qui offre plusieurs avantages et présente quelques inconvénients. Dans cette partie, nous examinerons ces aspects en détail.

* Avantages des Tests par Boîte Blanche :

**1. 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.

**2. 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.

**3. 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.

**4. 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.

* Inconvénients des Tests par Boîte Blanche :

**1. 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.

**2. 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.

**3. 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.

Les tests par boîte blanche offrent des avantages significatifs en termes de détection précoce des erreurs, de couverture exhaustive du code, d'identification des zones critiques et d'aide à la maintenance. Cependant, ils nécessitent une connaissance approfondie du code, peuvent présenter des limitations en termes de couverture et sont sensibles aux changements de l'implémentation. Malgré ces inconvénients, les tests par boîte blanche restent une pratique essentielle pour garantir la qualité des logiciels et sont complémentaires aux tests par boîte noire. Ils sont largement utilisés dans l'industrie du développement logiciel pour assurer la fiabilité, la sécurité et la maintenabilité des systèmes informatiques.

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

**1. EXAMEN DES EXIGENCES DE L'APPLICATION DE COMMERCE ELECTRONIQUE**

- L'application doit permettre aux utilisateurs de rechercher des produits par mot-clé, catégorie et caractéristiques.

- Les utilisateurs doivent pouvoir ajouter des produits au panier et passer à la caisse pour effectuer des achats.

- Le processus de paiement doit prendre en charge différentes méthodes de paiement, telles que les cartes de crédit et les portefeuilles électroniques.

- Après l'achat, les utilisateurs doivent recevoir une confirmation de commande par e-mail.

- Les utilisateurs doivent pouvoir gérer leur compte, y compris leurs informations personnelles et leurs commandes précédentes.

**2. CONCEPTION DES CAS DE TEST**

a) Cas de test : Recherche de produits par mot-clé

- Action : Ouvrir l'application et accéder à la fonction de recherche.

- Action : Entrer un mot-clé valide dans le champ de recherche.

- Résultat attendu : Les produits correspondants au mot-clé sont affichés.

- Résultat réel : Les produits correspondants sont affichés correctement.

b) Cas de test : Ajout d'un produit au panier

- Action : Sélectionner un produit à partir des résultats de recherche.

- Action : Cliquer sur le bouton "Ajouter au panier".

- Résultat attendu : Le produit est ajouté au panier avec succès.

- Résultat réel : Le produit est ajouté au panier sans problème.

c) Cas de test : Processus de paiement

- Action : Accéder au panier et cliquer sur le bouton "Passer à la caisse".

- Action : Sélectionner une méthode de paiement valide.

- Action : Entrer les informations de paiement requises.

- Résultat attendu : La transaction est effectuée avec succès.

- Résultat réel : La transaction est traitée correctement.

d) Cas de test : Confirmation de commande par e-mail

- Action : Vérifier la boîte de réception de l'e-mail associée au compte utilisateur.

- Résultat attendu : Un e-mail de confirmation de commande est reçu.

- Résultat réel : L'e-mail de confirmation de commande est reçu et contient les détails appropriés.

e) Cas de test : Gestion du compte utilisateur

- Action : Accéder à la section "Mon compte" de l'application.

- Action : Mettre à jour les informations personnelles.

- Résultat attendu : Les informations personnelles sont mises à jour avec succès.

- Résultat réel : Les informations personnelles sont correctement mises à jour.

**3. EXECUTION DES TESTS**

J'exécuterais les cas de test que j'ai conçus, en enregistrant les résultats réels et en comparant avec les résultats attendus pour chaque cas.

**4. RAPPORTS DE BUGS**

Si des problèmes sont identifiés lors des tests, je documenterais les problèmes rencontrés avec des détails tels que la description, les étapes de reproduction et les captures d'écran, afin de les signaler aux développeurs pour résolution.

**5. AMELIORATIONS POSSIBLES**

- Ajouter des tests de charge pour évaluer la capacité de l'application à gérer un grand nombre d'utilisateurs simultanés.

- Effectuer des tests de sécurité pour identifier les vulnérabilités potentielles de l'application.

- Intégrer des tests de performance pour évaluer la réactivité et la vitesse de l'application dans des conditions normales et de charge.

- Mettre en place des tests de compatibilité pour vérifier le bon fonctionnement de l'application sur différents navigateurs et appareils.

- Automatiser les tests pour garantir une exécution rapide et efficace des cas de test à chaque modification du code.

En suivant ces étapes et en intégrant ces améliorations, vous pourrez assurer la qualité et la fiabilité de votre application de commerce électronique tout au long de son cycle de vie.

**CONCLUSION**

Les tests par boîte blanche sont un outil puissant pour améliorer la qualité et la fiabilité des logiciels. En utilisant diverses techniques, les testeurs peuvent identifier les erreurs logiques, assurer une couverture de code suffisante et améliorer la maintenabilité du code. Cependant, il est important de prendre en compte les avantages et les inconvénients de ces techniques pour concevoir une stratégie de test efficace et équilibrée.