

espacement

Table des matières

Bonjour, ceci est un test post pour les puces

1. Vérification 1

2. VÉRIFICATION 2

3. Types de tests ad hoc

- Tests de copains : Deux membres de l'équipe (souvent un développeur et un testeur) collaborent pour tester ensemble un module spécifique, offrant des perspectives différentes.
- Tests de paires : Deux testeurs travaillent ensemble sur la même machine pour examiner et tester l'application, éveillant des idées spontanées.
- Test du singe : Fortement non structurée, cette approche consiste à entrer des données aléatoires ou à effectuer des actions aléatoires pour voir comment le logiciel se comporte et s'assurer qu'il ne s'arrête pas.



4. PUCE APRÈS IMAGES

Avantages et inconvénients

Avantages

- Gain de temps : Contourne le processus fastidieux de rédaction de scénarios de test formels, ce qui le rend idéal pour les délais serrés.
- Complète les tests formels : Détecte les bogues plus profonds ou obscurs que les suites de tests structurées ne détectent pas.
- Flexibilité : S'adapte parfaitement lorsque les besoins sont incomplets ou en évolution.

1. Techniques utilisées dans les tests ad hoc

Les techniques de test ad hoc sont des méthodes informelles utilisées par les testeurs pour identifier rapidement les défauts sans cas de test prédéfinis. Ces techniques reposent sur

l'expérience, l'intuition et l'exploration aléatoire de l'application.

- Estimation des erreurs : sur la base de l'expérience du testeur en matière de prévision des zones sujettes aux défauts, en se concentrant sur les erreurs courantes et les modules à risque.
- Test d'entrée aléatoire : utilisation de données inattendues ou non valides pour vérifier le comportement du système et s'assurer que l'application gère correctement les entrées inhabituelles.
- Test des limites (informel) : test des valeurs d'entrée minimale et maximale pour identifier les erreurs aux bords des limites d'entrée.
- Approche de test exploratoire : apprentissage et test simultanés de l'application sans cas de test prédéfinis afin de détecter les défauts de manière dynamique.
- Monkey Testing : exécution d'actions aléatoires sans logique pour vérifier la stabilité du système et identifier les pannes.
- Tests par session : réalisation de tests au cours de sessions courtes et ciblées afin de couvrir efficacement des modules spécifiques.
- Test négatif : utilisation d'entrées incorrectes ou non valides pour vérifier que le système gère correctement les erreurs et les validations.
- Test du scénario utilisateur : simulation du comportement réel de l'utilisateur pour s'assurer que l'application fonctionne correctement dans des situations réelles et pour identifier les problèmes de convivialité

Exemple : Application mobile

Un testeur effectue des actions aléatoires sur une application mobile :

- Basculement rapide entre les écrans
- Faire pivoter le périphérique fréquemment
- Utiliser l'application avec une connexion Internet faible
- Ouvrir plusieurs fonctions à la fois

Cela peut révéler des problèmes de performances, des problèmes d'interface utilisateur ou des pannes d'applications.

Conditions idéales pour les tests ad hoc

Plus efficaces lorsque le temps est limité, les testeurs ont une connaissance approfondie du domaine et des vérifications exploratoires rapides sont nécessaires pour découvrir les défauts cachés.

- Lorsque le temps de test est limité.
- Une fois le test formel terminé.
- Lorsque la validation rapide d'une fonctionnalité ou d'un module est requise.
- Lorsque des scénarios de test détaillés ne sont pas disponibles.
- Lorsque le développement est presque terminé.
- Lors du test des zones à haut risque ou sujettes à erreur.

Limites des tests ad hoc

Manque de documentation et de structure, ce qui rend difficile la reproduction des défauts et garantit une couverture complète des tests.

- Pas de documentation appropriée, ce qui rend difficile le suivi des défauts plus tard.
- La couverture de test n'est pas garantie en raison de sa nature non structurée.
- Dépend fortement de l'expérience et des connaissances du testeur.
- Les problèmes détectés peuvent être difficiles à reproduire.
- Impossible de remplacer les méthodes de test formelles et systématiques.

Avantages des tests ad hoc

Permet d'identifier rapidement les défauts inattendus sans cas de test formel, ce qui permet de gagner du temps et d'améliorer la couverture globale des tests.

- Identifie les bogues qui peuvent ne pas être détectés par des cas de test écrits.
- Peut être effectué en peu de temps, surtout lorsque les délais sont serrés.
- Encourage la pensée créative et aide à générer de nouveaux scénarios de test.
- Améliore la qualité des produits en détectant les problèmes inattendus.
- Peut être réalisé à n'importe quelle étape du [processus de développement de logiciel \(SDLC\)](#).

Meilleures pratiques pour la réalisation de tests ad hoc

- Posséder de solides connaissances sur les produits : Comprendre clairement l'application et ses fonctionnalités.
- Identifier les zones sujettes à erreur : Concentrez-vous sur les modules susceptibles de présenter des défauts.
- Hiérarchiser les fonctionnalités critiques : Testez d'abord les zones importantes et à haut risque.
- Planification à un niveau de base : Gardez une idée approximative de ce qu'il faut tester, même sans cas de test formel.
- Utiliser les outils appropriés : Utiliser des outils de débogage et de test pour améliorer l'efficacité.

1. <https://www.youtube.com/>
2. <https://fdk-stage.cisco.com/c/en/us/support/docs/licensing/ask-licensing/cda/device-management/lic219871-this-is-the-test-article-to-check-the.html>
3. <https://www.geeksforgeeks.org/software-engineering/adhoc-testing-in-software/>

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.