

spaziatura

Sommario

Salve, questo è il post di prova per i proiettili

1. Controllo 1

2. CONTROLLO 2

3. Tipi di test ad-hoc

- Test amico: Due membri del team (spesso uno sviluppatore e un tester) collaborano per testare insieme un modulo specifico, offrendo prospettive diverse.
- Prova in coppia: Due tester lavorano insieme sulla stessa macchina per rivedere e testare l'applicazione, scatenando idee spontanee.
- Test delle scimmie: Fortemente non strutturato, questo approccio comporta l'immissione di dati casuali o l'esecuzione di azioni casuali per verificare il comportamento del software e garantire che non si blocchi.



4. PUNTO ELENCO DOPO LE IMMAGINI

Vantaggi e svantaggi

Pro

- Risparmio di tempo: Ignora il lungo processo di scrittura dei test case formali, rendendolo ideale per scadenze ravvicinate.
- Completa i test formali: Trova bug più profondi o oscuri non rilevati nelle suite di test strutturati.
- Flessibilità: Si adatta perfettamente quando i requisiti sono incompleti o in fase di modifica.

1. Tecniche utilizzate nei test ad hoc

Le tecniche di test ad hoc sono metodi informali utilizzati dai tester per identificare rapidamente i difetti senza test case predefiniti. Queste tecniche si basano sull'esperienza, l'intuizione e l'esplorazione casuale dell'applicazione.

- Individuazione degli errori: basata sull'esperienza del tester per prevedere le aree soggette a difetti, con particolare attenzione agli errori comuni e ai moduli rischiosi.
- Test con input casuale: utilizzo di dati inattesi o non validi per controllare il comportamento del sistema e garantire che l'applicazione gestisca correttamente gli input insoliti.
- Test limite (informale): verifica dei valori di input minimo e massimo per identificare gli errori ai bordi dei limiti di input.
- Approccio al testing esplorativo: apprendimento e test simultanei dell'applicazione senza test case predefiniti per individuare i difetti in modo dinamico.
- Test delle scimmie: esecuzione di azioni casuali senza logica per controllare la stabilità del sistema e identificare gli arresti anomali.
- Test basato su sessioni: esecuzione di test in brevi sessioni mirate per coprire in modo efficiente moduli specifici.
- Test negativi: utilizzo di input non corretti o non validi per verificare che il sistema gestisca correttamente gli errori e le convalide.
- Test dello scenario utente: simulazione del comportamento reale dell'utente per garantire il corretto funzionamento dell'applicazione in situazioni reali e per identificare i problemi di utilizzabilità

Esempio: App mobile

Un tester esegue azioni casuali su un'applicazione mobile:

- Passare rapidamente da uno schermo all'altro
- Ruotare frequentemente il dispositivo
- Usa l'app con una connessione Internet insufficiente
- Apertura di più feature contemporaneamente

Ciò potrebbe causare problemi di prestazioni, glitch dell'interfaccia utente o arresti anomali dell'app.

Condizioni ideali per i test ad hoc

La soluzione più efficace quando il tempo è limitato, i tester hanno una solida conoscenza del dominio e sono necessari rapidi controlli esplorativi per individuare i difetti nascosti.

- Quando il tempo di esecuzione del test è limitato.
- Dopo il completamento dei test formali.
- Quando è necessaria la convalida rapida di una feature o di un modulo.
- Quando non sono disponibili test case dettagliati.
- Quando lo sviluppo è quasi completato.
- Quando si testano aree ad alto rischio o soggette ad errori.

Limitazioni dei test ad hoc

Mancanza di documentazione e struttura, rendendo difficile la riproduzione dei difetti e garantire la copertura completa dei test.

- Mancanza di una documentazione adeguata, che rende difficile il monitoraggio dei difetti in un secondo momento.
- La copertura dei test non è garantita a causa della sua natura non strutturata.
- Dipende in larga misura dall'esperienza e dalle conoscenze del tester.
- I problemi rilevati possono essere difficili da riprodurre.
- Impossibile sostituire i metodi di test formali e sistematici.

Vantaggi dei test ad hoc

Consente di identificare rapidamente i difetti imprevisti senza test case formali, risparmiando tempo e migliorando la copertura complessiva dei test.

- Identifica i bug che potrebbero non essere trovati tramite test case scritti.
- Può essere eseguita in tempi brevi, soprattutto quando le scadenze sono ravvicinate.
- Incoraggia il pensiero creativo e contribuisce a generare nuovi scenari di test.
- Migliora la qualità dei prodotti rilevando problemi imprevisti.
- Può essere eseguita in qualsiasi fase del [Software Development Life Cycle Process \(SDLC\)](#).

Procedure ottimali per l'esecuzione di test ad hoc

- Una conoscenza approfondita del prodotto: Comprendere chiaramente l'applicazione e le relative funzionalità.
- Identificazione delle aree soggette a errori: Concentrarsi sui moduli in cui è probabile che si verifichino difetti.
- Assegnazione di priorità alle caratteristiche critiche: Testare prima le aree importanti e ad alto rischio.
- Pianificazione a livello di base: Tieni un'idea approssimativa di cosa verificare, anche senza test case formali.
- Utilizzare Gli Strumenti Appropriati: Utilizzare gli strumenti di debug e test per migliorare l'efficacia.

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/>

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).