

afstand

Inhoud

Hallo, dit is een testbericht voor kogels

1. Controle 1

2. CONTROLE 2

3. Typen ad-hoc tests

- Buddy Testing: Twee teamleden (vaak een ontwikkelaar en een tester) werken samen om een specifieke module samen te testen, waardoor verschillende perspectieven worden geboden.
- Pair Testing: Twee testers werken samen op dezelfde machine om de toepassing te beoordelen en te testen, waardoor spontane ideeën worden aangewakkerd.
- Monkey Testing: Deze aanpak is zwaar ongestructureerd en omvat het invoeren van willekeurige gegevens of het uitvoeren van willekeurige acties om te zien hoe de software zich gedraagt en ervoor te zorgen dat deze niet crasht.



4. OPSOMMINGSTEKENS NA AFBEELDINGEN

Voordelen & nadelen

profs

- Tijdbesparing: omzeilt het tijdrovende proces van het schrijven van formele testcases, waardoor het ideaal is voor krappe deadlines.
- Complementeert formele testen: vindt diepere of obscure bugs die gestructureerde testsuites missen.
- Flexibiliteit: past zich perfect aan wanneer vereisten onvolledig of veranderlijk zijn.

1. Technieken gebruikt bij ad hoc testen

Ad hoc-testtechnieken zijn informele methoden die door testers worden gebruikt om snel defecten te identificeren zonder vooraf gedefinieerde testgevallen. Deze technieken zijn gebaseerd op

ervaring, intuïtie en willekeurige verkenning van de toepassing.

- Foutgissen: Gebaseerd op de ervaring van de tester om defectgevoelige gebieden te voorspellen, met de nadruk op veelvoorkomende fouten en risicovolle modules.
- Random Input Testing: Onverwachte of ongeldige gegevens gebruiken om het gedrag van het systeem te controleren en ervoor te zorgen dat de toepassing ongebruikelijke invoer correct verwerkt.
- Grenstests (informeel): minimale en maximale invoerwaarden testen om fouten aan de randen van invoerlimieten te identificeren.
- Verkennende testbenadering: de toepassing gelijktijdig leren en testen zonder vooraf gedefinieerde testcases om defecten dynamisch te ontdekken.
- Monkey Testing: willekeurige acties uitvoeren zonder logica om de stabiliteit van het systeem te controleren en crashes te identificeren.
- Sessiegebaseerd testen: uitvoeren van testen in korte, gerichte sessies om specifieke modules efficiënt te behandelen.
- Negatieve tests: gebruik van onjuiste of ongeldige invoer om te controleren of het systeem fouten en valideringen correct verwerkt.
- Testen van gebruikersscenario's: simuleren van echt gebruikersgedrag om ervoor te zorgen dat de toepassing correct werkt in situaties in de praktijk en om gebruiksproblemen te identificeren

Voorbeeld: mobiele app

Een tester voert willekeurige acties uit op een mobiele applicatie:

- Switch tussen schermen snel
- Het apparaat vaak draaien
- Gebruik de app met een lage internetverbinding
- Meerdere functies tegelijk openen

Dit kan prestatieproblemen, UI-glitches of app-crashes aan het licht brengen.

Ideale omstandigheden voor ad hoc testen

Het meest effectief wanneer de tijd beperkt is, testers hebben een sterke domeinkennis en snelle verkennende controles zijn nodig om verborgen gebreken te ontdekken.

- Wanneer de testtijd beperkt is.
- Nadat de formele test is afgerond.
- Wanneer een snelle validatie van een functie of module vereist is.
- Wanneer gedetailleerde testcases niet beschikbaar zijn.
- Wanneer de ontwikkeling bijna voltooid is.
- Bij het testen van gebieden met een hoog risico of die vatbaar zijn voor fouten.

Beperkingen van ad hoc testen

Ontbreekt documentatie en structuur, waardoor het moeilijk is om defecten te reproduceren en

volledige testdekking te garanderen.

- Geen goede documentatie, waardoor het moeilijk is om defecten later op te sporen.
- Testdekking is niet gegarandeerd vanwege het ongestructureerde karakter.
- Dit is sterk afhankelijk van de kennis en ervaring van de tester.
- Gevonden problemen kunnen moeilijk te reproduceren zijn.
- Kan formele en systematische testmethoden niet vervangen.

Voordelen van ad hoc testen

Helpt onverwachte defecten snel te identificeren zonder formele testcases, waardoor tijd wordt bespaard en de algehele testdekking wordt verbeterd.

- Identificeert bugs die mogelijk niet worden gevonden via schriftelijke testcases.
- Het kan in korte tijd worden uitgevoerd, vooral als de deadlines krap zijn.
- Stimuleert creatief denken en helpt bij het genereren van nieuwe testscenario's.
- Verbeterd de productkwaliteit door onverwachte problemen aan het licht te brengen.
- Dit kan in elke fase van het [Software Development Life Cycle Process \(SDLC\)](#).

Best practices voor het uitvoeren van ad-hoctests

- Heb sterke productkennis: begrijp de toepassing en de functionaliteit ervan duidelijk.
- Foutgevoelige gebieden identificeren: focus op modules waar defecten waarschijnlijk optreden.
- Prioriteer kritieke functies: test eerst belangrijke en risicovolle gebieden.
- Plan op een basisniveau: houd een ruw idee van wat te testen, zelfs zonder formele testcases.
- Gebruik de juiste tools: gebruik foutopsporing en testtools om de effectiviteit te verbeteren.

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/>

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.