

간격

목차

안녕하세요! 이걸 총알을 넣는 기둥이에요

1. 확인 1

2. 확인 2

3. 임시 테스트 유형

- Buddy 테스트: 두 팀 구성원(대개 개발자와 테스터)이 서로 협력하여 특정 모듈을 함께 테스트함으로써 서로 다른 관점을 제공합니다.
- 쌍 테스트: 두 테스터가 같은 기계에서 함께 작업해 애플리케이션을 검토하고 테스트하면서 즉흥적인 아이디어를 촉발한다.
- 원숭이 테스트: 구조화되지 않은 이 접근 방식에서는 무작위 데이터를 입력하거나 무작위 작업을 수행하여 소프트웨어의 작동 방식을 확인하고 소프트웨어가 충돌하지 않도록 합니다.



4. 이미지 뒤에 글머리 기호

장점 및 단점

장점

- 시간 절약: 시간이 많이 소요되는 정식 테스트 사례 작성 프로세스를 건너뛰므로 마감일이 촉박한 경우에 적합합니다.
- 공식 테스트를 보완합니다. 구조화된 테스트 도구 모음에서 놓치는 더 깊거나 모호한 버그를 찾습니다.
- 유연성: 요구 사항이 불완전하거나 변경될 경우 완벽하게 적응합니다.

1. Adhoc 테스트에 사용되는 기술

애드 혹 테스트 기법은 미리 정의된 테스트 사례 없이 결함을 신속하게 식별하기 위해 테스터가 사용하는 비공식적인 방법입니다. 이러한 기법은 경험, 직관, 그리고 애플리케이션에 대한 무작위 탐색에 의존한다.

- 오류 추측: 일반적인 실수 및 위험 모듈을 중심으로 결함이 발생하기 쉬운 영역을 예측한 테스터의 경험을 기반으로 합니다.
- 임의 입력 테스트: 예상치 못한 데이터 또는 유효하지 않은 데이터를 사용하여 시스템 동작을 확인하고 애플리케이션이 비정상적인 입력을 올바르게 처리하는지 확인합니다.
- 경계 테스트(비공식적): 최소 및 최대 입력 값을 테스트하여 입력 제한의 가장자리에서 오류를 식별합니다.
- 탐색적 테스트 접근 방식: 결함을 동적으로 발견하기 위해 사전 정의된 테스트 사례 없이 애플리케이션을 동시에 학습하고 테스트합니다.
- 원숭이 테스트: 시스템 안정성을 확인하고 충돌을 식별하기 위해 논리 없이 임의의 작업을 수행합니다.
- 세션 기반 테스트: 특정 모듈을 효율적으로 다루기 위해 짧고 집중적인 세션에서 테스트를 수행합니다.
- 부정 테스트: 잘못된 입력 또는 잘못된 입력을 사용하여 시스템이 오류 및 검증을 올바르게 처리하는지 확인합니다.
- 사용자 시나리오 테스트: 실제 사용자 행동을 시뮬레이션하여 실제 상황에서 응용 프로그램이 올바르게 작동하는지 확인하고 사용성 문제를 식별합니다.

예: 모바일 앱

테스터는 모바일 애플리케이션에 대해 무작위 작업을 수행합니다.

- 화면 간 신속한 전환
- 자주 장치 회전
- 인터넷 연결이 낮은 앱 사용
- 여러 기능을 한 번에 열기

이로 인해 성능 문제, UI 결함 또는 앱 충돌이 발견될 수 있습니다.

어드 혹 테스트를 위한 이상적인 조건

시간이 한정되어 있을 때 가장 효과적이고, 테스터들은 도메인 지식이 강하며, 숨겨진 결함을 발견하기 위해 빠른 탐색적 점검이 필요하다.

- 테스트 시간이 제한된 경우.
- 공식적인 테스트가 완료된 후
- 기능 또는 모듈의 빠른 검증이 필요한 경우.
- 자세한 테스트 사례를 사용할 수 없는 경우
- 개발이 거의 완료되면.
- 고위험 또는 오류가 발생하기 쉬운 영역을 테스트할 경우.

임시 테스트의 제한 사항

문서화 및 구조가 부족하여 결함을 재현하고 완전한 테스트 범위를 확보하기가 어렵습니다.

- 적절한 문서가 없으므로 나중에 결함을 추적하기가 어렵습니다.
- 비정형 특성으로 인해 테스트 커버리지가 보장되지 않습니다.
- 테스터의 경험과 지식에 크게 좌우됩니다.

- 발견된 문제는 재현하기 어려울 수 있습니다.
- 형식적이고 체계적인 테스트 방법을 대체할 수 없습니다.

애드혹 테스트의 장점

공식 테스트 사례 없이 예기치 않은 결함을 신속하게 식별하여 시간을 절약하고 전반적인 테스트 범위를 개선할 수 있습니다.

- 서면 테스트 사례를 통해 찾을 수 없는 버그를 식별합니다.
- 특히 마감일이 촉박한 경우 단시간에 수행할 수 있습니다.
- 창의적인 사고를 장려하고 새로운 시험 시나리오를 만들 수 있도록 도와줍니다.
- 예기치 않은 문제를 발견하여 제품 품질을 개선합니다.
- SDLC([Software Development Life Cycle Process](#))의 모든 단계에서 수행할 수 있습니다.

임시 테스트를 수행하기 위한 모범 사례

- 강력한 제품 지식 보유: 애플리케이션 및 그 기능을 명확하게 파악합니다.
- 오류가 발생하기 쉬운 영역 파악: 결함이 발생할 가능성이 높은 모듈에 초점을 맞춥니다.
- 주요 기능의 우선 순위 지정: 중요한 영역과 위험도가 높은 영역을 먼저 테스트합니다.
- 기본 레벨에서 계획: 공식적인 시험 사례가 없더라도 어떤 시험을 치를지에 대한 대략적인 생각을 유지하라.
- 적절한 툴 사용: 디버깅 및 테스트 툴을 활용하여 효율성을 높입니다.

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/>

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.