

테스트 설계

테스트 설계 목적

- 소프트웨어의 결함을 발견하고 수정하여 품질을 보장 한다.
- 소프트웨어가 요구사항을 만족하고, 사용자에게 신뢰성을 제공하는지 확인.

테스트의 종류

- **단위 테스트 (Unit Testing):** 개별 모듈이나 함수가 올바르게 작동하는지 확인.
- **통합 테스트 (Integration Testing):** 두개 이상의 모듈이나 소프트웨어 구성 요소들을 결합해 함께 예상대로 작동하는지 확인하는 테스트.
- **시스템 테스트 (System Testing):** 완전하고 통합된 소프트웨어 시스템이 전체적으로 올바르게 작동하는지 확인.
- **인수 테스트 (Acceptance Testing):** 최종 사용자의 요구사항을 만족하는지 확인.

3. 테스트 기법

블랙박스 테스트

- 소프트웨어의 내부 구조나 작동원리를 모르는 상태에서 소프트웨어의 동작을 테스트하는 방법.
- 개발자 입장이 아닌 사용자 입장에서 소프트웨어에 대한 요구사항과 결과물이 일치하는지 테스트하는 기법.
- 사용자의 요구사항과 결과물이 일치하는지 확인.
- 내부 오류를 확인하기 어렵다.

화이트박스 테스트

- 소프트웨어 혹은 제품의 내부 구조, 동작을 세밀하게 검사하는 테스트 방식
- 개발자가 소프트웨어 또는 컴포넌트 등의 로직에 대한 테스트를 수행하기 위해 설계 단계에서 요구된 사항을 확인하는 개발자 관점의 단위테스팅 기법

4. 효과적인 테스트의 특징

- **완전성:** 모든 요구사항을 커버하도록 설계된 테스트.
- **반복 가능성:** 동일한 테스트 케이스를 반복 실행해도 동일한 결과를 얻을 수 있어야 한다.
- **독립성:** 테스트는 독립적으로 실행 가능해야 하며, 다른 테스트에 의존하지 않아야 한다.
- **재사용성:** 잘 설계된 테스트 케이스는 재사용이 가능해야 한다.

테스트 케이스 설계

- 목적: 개별 테스트 케이스를 설계하여 각 요구사항이 올바르게 구현되었는지 검증.
- 포함 항목:
 - ID: 각 테스트 케이스에 고유한 식별자 부여.
 - 테스트 케이스 이름: 테스트 케이스 이름
 - 입력 데이터 (Input Data): 테스트 실행 시 사용할 입력 값.
 - 전제 조건 (Preconditions): 테스트 실행 전에 만족해야 할 조건.
 - 기대 결과 (Expected Result): 입력 데이터에 따른 예상 출력 값.
 - 실제 결과 (Actual Result): 테스트 실행 후 기대되는 상태.
 - Pass 여부: 테스트 후 성공/실패 여부.

테스트 케이스 예시

계산기 앱 테스트 케이스 예시

ID	이름	입력 데이터	전제 조건	기대 결과	실제 결과	Pass 여부
TC01	덧셈 테스트	2 + 3	앱 정상 실행	5	5	Pass
TC02	뺄셈 테스트	7 - 4	앱 정상 실행	3	3	Pass
TC03	곱셈 테스트	6 * 5	앱 정상 실행	30	30	Pass

사용자 인증 기능 테스트 예시

ID	이름	입력 데이터	전제 조건	기대 결과	실제 결과	Pass 여부
TC01	올바른 로그인 테스트	ID: user1, PW: pass1234	계정이 활성화 상태	로그인 성공 메시지 출력	로그인 성공	Pass
TC02	잘못된 비밀번호 테스트	ID: user1, PW: wrongpass	계정이 활성화 상태	"비밀번호가 틀렸습니다" 출력	메시지 출력	Pass
TC03	존재하지 않는 사용자 테스트	ID: unknown, PW: pass1234	사용자 계정 미존재	"계정을 찾을 수 없습니다"	메시지 출력	Pass
	비밀번호는	ID: user1, PW: (비		"비밀번호를	메시	