1. 애플리케이션 테스트 케이스 설계

* 애플리케이션 테스트 케이스 작성

1. 소프트웨어 테스트의 이해

* 개념

개발된 응용 애플리케이션이나 시스템이 사용자가 요구하는 기능과 성능, 사용성, 안정성 등을 만족하는지 확인하고, 노출되지 않은 숨어있는 소프트웨어의 결함을 찾아내는 활동

* 필요성

오류 발견 관점

오류 예방 관점

품질 향상 관점

* 기본 원칙

결함 존재 증명 : 테스트는 결함이 존재함을 밝히는 활동

완벽 테스팅은 불가능 : 무한경로, 무한 입력값으로 인한 완벽한 테스트가 어렵다

초기 집중 : 개발 초기에 체계적인 분석 및 설계가 수행되면 테스팅 기간 단축

결함 집중 : 적은 수의 모듈 에서 대다수 결함이 발견된다는 원칙, 파레토 법칙

살충제 패러독스 : 동일한 테스트 케이스에 의한 반복적 테스트는 새로운 버그를 찾지 못한다

정황 의존성 : 소프트웨어의 성격에 맞게 테스트를 수행

오류-부재의 궤변 : 요구사항을 충족시켜주지 못한다면 결함이 없다고 해도 품질이 높다고 볼 수 없다

* 소프트웨어 테스트 산출물

테스트 계획서 : 목적, 범위 정의, 대상 시스템 구조 파악, 테스트 수행 절차

테스트 베이시스 : 분석 설계 단계의 논리적인 케이스

테스트 케이스 : 테스트를 위한 설계 산출물, 응용 소프트웨어가 사용자의 요구사항을 준수하는지 확인하기 위해 설계된 입력값, 실행 조건, 기대결과로 구성

테스트 슈트 : 테스트 케이스를 실행환경에 따라 구분해 놓은 테스트 케이스 집합

테스트 시나리오 : 애플리케이션의 테스트 되어야 할 기능 및 특징, 테스트가 필요한 상황을 작성

테스트 스크립트 : 테스트 케이스의 실행 순서를 작성

테스트 결과서 : 테스트 결과를 정리한 문서

1. 소프트웨어 테스트 유형

* 프로그램 실행 여부

정적 테스트 : 리뷰, 정적 분석

동적 테스트 : 화이트박스, 블랙박스, 경험 기반 테스트

* 테스트 상세 기법

화이트 박스 테스트

각 응용 프로그램의 내부 구조와 동작을 검사

코드 분석과 프로그램 구조에 대한 지식을 바탕으로 문제가 발생할 가능성이 있는 모듈 내부를 테스트 하는 방법

소스 코드의 모든 문장을 한번 이상 수행함으로써 진행

구조 기반 테스트, 코드 기반 테스트, 로직 기반 테스트, 글래스 박스 테스트

화이트 박스 테스트 유형

구문 커버리지 : 모든 명령문 적어도 한번 수행

결정 커버리지 : 결정 포인트 내의 전체 조건식이 적어도 한번은 참과 거짓의 결과를 수행

조건 커버리지 : 각 개별 조건식이 적어도 한번은 결과가 되도록 수행

조건-결정 커버리지 : 전체 조건식 뿐만 아니라 개별 조건식도 수행

변경 조건-결정 커버리지 : 개별 조건식이 다른 개별 조건식에 영향을 받지 않고 전체 조건식에 독립적으로 영향을 주도록 함으로 조건/결정 커버리지를 향상

다중 조건 커버리지 : 결정 조건 내 모든 개별 조건식의 모든 가능한 조합을 100% 보장

기본 경로 커버리지 : 수행 가능한 모든 경로 테스트

제어 흐름 테스트 : 프로그램 제어 구조를 그래프 형태로 나타내 내부 로직 테스트

데이터 흐름 테스트 : 제어 흐름 그래프에 데이터 사용 현황을 추가

루프 테스트 : 반복 구조에 초점을 맞춰 실시

블랙박스 테스트 : 프로그램 외부 사용자의 요구사항 명세를 보면서 수행

동등 분할 / 동치 분한 / 균등 분할 / 동치 클래스 분해 : 입력 데이터의 영역을 유사한 도메인 별로 그룹핑 하여 대푯값 테스트 케이스 도출

경곗값 분석 / 한곗값 : 최솟값 바로 위, 최대치 바로 아래 등 입력값의 극한 한계를 테스트

결정 테이블 테스트 : 요구사항의 논리와 발생 조건을 테이블 형태로 나열하여 조건과 행위를 모두 조합

상태 전의 테스트 : 테스트 대상 시스템이나 객체의 상태를 구분하고 이벤트에 의해 어느 한 상태에서 다른 상태로 전이되는 경우의 수를 테스트

유스 케이스 테스트 : 유스케이스로 모델링 되어 있을 때 프로세스 흐름을 기반으로 테스트 케이스를 명세화

분류 트리 케스트 : SW의 일부 또는 젠체를 트리 구조로 분석

페어와이즈 테스트 : 테스트 데이터값들 간에 최소한 한 번씩을 조합하는 방식

원인-결과 그래프 테스트 : 그래프를 활용하여 입력 데이터 간의 관계 및 출력에 미치는 영향을 분석

비교 테스트 : 여러 버전의 프로그램에 같은 입력값을 넣어서 동일 한 결과 데이터가 나오는지 비교

오류 추정 테스트 : 개발자가 범할 수 있는 실수를 추정하고 이에 따른 결함이 검출되도록 테스트 케이스를 설계

* 테스트에 대한 시각

검증, 확인

* 테스트의 목적

회복 테스트 : 정상적 복귀 여부

안전 테스트 : 불법적인 소프트웨어가 접근하여 시스템을 파괴하지 못하도록 소스 코드의 보안적인 결함을 미리 점검

성능 테스트 : 사용자의 요구에 시스템이 반응하는 속도 측정

구조 테스트 : 시스템의 내부 논리 경로 복잡도 측정

회귀 테스트 : 오류를 제거하거나 수정한 시스템에서 오류 제거와 수정에 의해 새로이 유입된 오류가 없는지 확인하는 일종의 반복 테스트

병행 테스트 : 변경된 시스템과 기존 시스템에 동일한 데이터를 입력 후 결과 비교

* 테스트의 종류