

■ 애플리케이션 테스트 수행하기

■ 가. 프로젝트 수행 단계에 따른 테스트의 분류

1) 단위 테스트

작은 소프트웨어 단위(컴포넌트 또는 모듈)를 테스트하는 것으로서, 일반적으로 개발자 자신에 의해 행해진다.

2) 통합 테스트

모듈 사이의 인터페이스, 통합된 컴포넌트 간의 상호 작용을 테스트하는 것으로, 하나의 프로세스가 완성된 경우 부분적으로 통합 테스트를 수행하는 경우도 있다.

3) 시스템 테스트

통합된 단위 시스템의 기능이 컴퓨터 시스템에서 정상적으로 수행되는지를 테스트하는 것으로 성능 및 장애 테스트가 여기에 포함된다.

4) 인수 테스트

일반적으로 최종 사용자와 업무에 따른 이해관계자 등이 테스트를 수행함으로써 개발된 제품에 대해 운영 여부를 결정하는 테스트로, 실제 업무 적용 전에 수행한다.

■ 나. 테스트 단계별 도구

1) 테스트 계획 : 요구사항 관리

2) 테스트 분석/설계 : 테스트케이스 생성, 커버리지 분석

3) 테스트 수행 : 테스트 자동화, 정적분석, 동적분석, 성능 테스트, 모니터링

4) 테스트 통제 : 형상관리, 테스트 관리, 결함 추적/관리

■ 다. 결함관리

1) 결함관리 프로세스

가) 결함관리 계획

결함관리 계획은 전체 프로세스에서 결함관리에 대한 일정, 인력, 업무 프로세스를 확보하여 계획을 수립하는 것을 말한다.

나) 결함 기록

테스터는 발견된 결함에 대한 정보를 결함관리 DB에 기록한다.

다) 결함 검토

등록된 결함에 있어서 주요 내용을 검토하고, 결함을 수정할 개발자에게 전달한다.

라) 결함 수정

개발자는 할당된 결함의 프로그램을 수정한다.

마) 결함 재확인

테스터는 개발자가 수정한 내용을 확인하고 다시 테스트를 수행한다.

바) 결함 상태 추적 및 모니터링 활동

결함관리 팀장은 결함관리 데이터베이스를 이용하여 게시판 형태의 서비스를 제공한다.

사) 최종 결함 분석 및 보고서 작성

발견된 결함에 대한 내용과 이해관계자들의 의견이 반영된 보고서를 작성하고 결함관리를 종료한다.

2) 결함의 상태 및 추적

결함 등록(Open)

테스터와 품질 관리(QA) 담당자에 의해 결함이 처음 발견되어 등록되지만, 아직 분석이 되지 않은 상태
결함 검토(Reviewed)

등록된 결함을 담당 모듈 개발자, 테스터, 품질 관리(QA) 담당자와 검토하는 상태

결함 할당(Assigned)

결함의 영향 분석 및 수정을 위해 개발자와 문제 해결 담당자에게 할당된 상태

결함 수정(Resolved/Fixed)

개발자에 의해 결함의 수정이 완료된 상태 (Resolved : 해결하다.)

결함 조치 보류(Deferred)

수정이 필요한 결함이지만 현재 수정이 불가능해서 연기된 상태로서 우선순위, 일정 등을 고려하여 재오픈을 준비하는 상태

결함 종료(Closed)

발견된 결함이 해결되고 테스터와 품질 관리(QA) 담당자에 의해 종료 승인을 한 상태

결함 해제(Clarified)

테스터, 품질 관리(QA) 담당자가 결함을 검토한 결과, 결함이 아니라고 판명된 상태

3) 결함 분류

가) 시스템 결함

비정상적인 종료/중단, 응답 시간 지연, 데이터베이스 에러 등 주로 애플리케이션 환경과 데이터베이스 처리에서 발생하는 결함을 말한다.

(1) 비정상적인 종료/중단

특정 기능 실행 시 응용 프로그램의 작동 정지, 종료, 시스템 다운이 되는 경우이다.

(2) 응답 시간 지연

응용 프로그램 작동 후 조회 또는 보고서 출력 시 지연되는 경우와 메모리 부족, 하드웨어와 소프트웨어의 비일관성으로 발생하는 경우이다.

(3) 데이터베이스 에러

응용 프로그램 작동 후 사용자 데이터의 등록, 수정, 삭제, 조회가 정상적으로 작동하지 않는 경우

나) 기능 결함

사용자의 요구사항 미반영/불일치, 부정확한 비즈니스 프로세스, 스크립트 에러, 타 시스템 연동 시 오류 등 기획, 설계, 업무 시나리오 단계에서 발생된 결함을 말한다.

(1) 요구사항 미반영/불일치

요구사항에 명시된 기능이 응용 프로그램에 구현되지 않은 경우와 다르게 구현되어 작동하는 경우

(2) 부정확한 비즈니스 프로세스

기능 자체는 수행되나 내부 프로세스 로직의 문제로 부정확한 결과를 내는 경우이다.

(3) 스크립트 에러

특정 기능 실행 시 웹 브라우저에서 스크립트 오류가 발생하는 경우이다.

(4) 타 시스템 연동 시 오류

기존 시스템과의 연동을 통해 데이터를 주고받는 과정에서 오류가 발생하는 경우이다.

다) GUI 결함

GUI 결함은 응용 프로그램의 UI 비일관성, 부정확한 커서/메시지, 데이터 타입의 표시 오류 등으로 사용자 화면 설계에서 발생된 결함을 말한다.

(1) 응용 프로그램 UI 비일관성

프로젝트에서 정의한 UI 표준과 상이하게 구현된 경우이다.

(2) 부정확한 커서/메시지

커서의 위치가 입력 대상의 첫 번째 필드에 위치해 있지 않거나, 탭 시퀀스가 순차적으로 동작하지 않는 경우, 각 기능에서 제공하는 메시지 내용이 부정확한 내용을 보여주는 경우이다.

(3) 데이터 타입의 표시 오류

입력 필드에 지정된 형식과 다르게 입력해도 저장이 되는 경우와 입력 필드에 유효하지 않은 데이터(Invalid Data)를 입력했을 때 오류가 나는 경우이다.

라) 문서 결함

기획자, 사용자, 개발자 간의 의사소통과 기록이 원활하지 않은 경우에 발생하는 결함으로 사용자의 온라인 / 오프라인 매뉴얼의 불일치, 요구사항 분석서와 기능 요구사항의 불일치로 인한 불완전한 상태의 문서의 경우를 말한다

마) 결함 관련 용어

(1) 에러(Error)

소프트웨어 개발 또는 유지 보수 수행 중에 발생한 부정확한 결과로, 개발자의 실수로 발생한 오타, 개발 명세서의 잘못된 이해, 서버루틴의 기능 오해 등이 있다.

(2) 오류(Fault)

프로그램 코드 상에 존재하는 것으로 비정상적인 프로그램과 정상적인 프로그램 버전 간의 차이로 인하여 발생되며, 잘못된 연산자가 사용된 경우에 프로그램이 서버루틴으로부터의 에러 리턴을 점검하는 코드가 누락된것을 말한다.

(3) 실패(Failure)

정상적인 프로그램과 비정상적인 프로그램의 실행 결과의 차이를 의미하며, 프로그램 실행 중에 프로그램의 실제 실행 결과를 개발 명세서에 정의된 예상 결과와 비교함으로써 발견한다.

(4) 결함(Defect)

버그, 에러, 오류, 실패, 프로그램 실행에 대한 문제점, 프로그램 개선 사항 등의 전체를 포괄하는 용어이다.

■ 애플리케이션 결함 조치하기

■ 가. 부하 및 성능 테스트

- 1) 동시 이용자 수 : $TPS(Throughput) * \text{호출간격}(\text{응답시간}(\text{Sec}) + \text{대기시간}(\text{Sec}))$
- 2) 동시 단말 이용자(Concurrent User) : PC 앞에 앉아서 시스템을 이용하는 이용자로서, Active User와 In-active User(가상적 유저)의 합으로 정의한다.
- 3) 액티브 이용자(Active User) : 동시에 같은 서비스나 업무를 실행하고 나서 응답을 기다리고 있는 이용자
- 4) 처리량(Throughput) : 단위 시간당 처리하는 건수로써 단위 시간당 요청 건수(Arrival rate)와 단위 시간당 처리 건수(Service Rate)로 구분되어 표현한다.
- 5) 대기 시간(Thinktime) : 응답을 받은 직후부터 다음 명령 또는 호출할 때까지 사용자가 대기하는 시간

■ 나. 장애 복구 테스트

- 1) 웹 서버 장애 테스트 : 동작 중인 웹 서버 10대 중 1대의 웹 서버 프로세스를 강제로 Kill한다.
- 2) 웹 애플리케이션 장애 테스트 : 동작 중인 WAS서버의 인스턴스 8개 중 실행 중인 인스턴스 한 개를 멈춘다.
- 3) 데이터베이스 장애 테스트 : Active한 Primary 서버의 DB서비스를 강제로 Kill한다.

■ 다. 코드 인스펙션(inspection, 점검)

- 1) 인스펙션
인스펙션은 개발 가이드에 따른 표준(체크리스트)을 준수하였는지를 파악하기 위함에 있으므로 기능적으로 이상이 없는 소스 코드를 대상으로 검증한다.
- 2) 워크스루(walk-through)
팀이나 관리자의 필요에 따라 집단 토의 통하여 프로그램 산출물 평가와 개선에 대하여 검증한다.
- 3) 인스펙션 필요성
 - 가) 결함을 빨리 찾을수록 수정(fix) 비용이 적게 든다.
 - 나) 인스펙션의 데이터를 통해 업무에 집중할 수 있다.
 - 다) 인스펙션을 함으로써 교차 교육(Cross-training)을 돕는다.
 - 라) 제품의 "re-engineering"이 가능한 영역을 식별하도록 돕는다.
 - 마) 소프트웨어를 개발하고 유지하는 데 적은 비용이 든다.
 - 바) 스케줄에 긍정적인 효과를 준다.
 - 사) 품질을 향상시킨다.

■ 형상관리

▪ 가. 정의

형상관리(configuration management)는 형상항목을 식별하여 그 기능적, 물리적 특성을 문서화하고, 그러한 특성의 변경, 제어, 처리 상태를 기록 및 보고하면서 명시된 요구사항에 부합하는지 확인 및 감독하는 활동을 의미한다.

▪ 나. 특징

- 1) 형상은 크게 하드웨어와 소프트웨어로 구분하여 관리한다.
- 2) 형상은 소프트웨어 개발 산출물(문서, 코드 등)이 배치되어 있는 형태를 의미한다.
- 3) 일반적으로 소프트웨어가 하드웨어보다는 변화가 더 빠르게 바뀐다.
- 4) 소프트웨어는 하루에도 몇번씩 바뀔 수 있으며, 어느 단계에서나 변경이 가능하다.
- 5) 언제 어떻게 변경될지 모르기 때문에 소프트웨어 형상에 대한 관리가 매우 어렵다.

■ 형상관리 분야별 역할

▪ 가. 경영층

프로젝트의 형상관리 활동에 대한 방침, 자원, 자금을 제공하며 정기적으로 검토한다.

▪ 나. 프로젝트 관리자

프로젝트의 형상관리 조직(형상관리자, 형상통제위원회)을 구성하며, 내부/외부의 변경요청에 대한 최종 승인권한을 가진다. 또한 형상관리 활동의 주요 사안을 경영층에 보고하는 역할을 한다.

▪ 다. 개발자

등록된 형상항목을 사용하여 엔지니어링 활동을 수행하고, 변경된 형상항목에 대하여 조치를 취하는 역할을 한다.

▪ 라. 형상관리자

형상관리 계획서에 따라 형상관리 활동을 수행하는 사람으로서 절차에 따라 문서화 및 변경관리를 담당한다.

▪ 마. 형상통제 위원회

형상항목의 변경에 대하여 수락 및 거절하는 역할을 담당한다. 일반적으로 프로젝트 관리자, 형상관리자, 품질 담당자, 기술 담당자, 고객관리 담당자로 구성된다.

■ 형상관리 절차 4단계

▪ 1단계 : 형상식별

형상관리의 시작으로 시스템을 구성하는 형상관리의 대상들을 구분하고, 관리 목록의 번호를 부여한다.

▪ 2단계 : 형상통제

소프트웨어 형상변경 제안을 검토/승인하여 현재의 소프트웨어 기준선에 반영될 수 있도록 통제하는 것을 의미한다.

▪ 3단계 : 형상감사

형상항목의 변경이 제대로 이루어졌는지 무결성을 검토/승인하는 것을 의미한다.

▪ 4단계 : 형상기록/보고

프로젝트에 대한 모든 상황을 효율적으로 관리하기 위하여 정보를 기록하고, 보고한다.

■ 형상관리 유지보수

■ 가. 정의

소프트웨어의 유지보수는 사용자에게 배달되어 사용되는 과정에서 발생하는 활동들을 의미하며, 개발과정의 순환이라 할 수 있다. 소프트웨어는 계속적으로 변경되기 때문에 유지보수 과정을 체계적으로 문서화하여 불일치와 혼돈을 최소화 하여야 한다.

■ 나. 4가지 유형

1) 수정 유지보수(하자보수)

프로그램이 사용자에게 배달되어 사용되는 동안에 오류가 발견되어 수정하는 것을 의미한다.

2) 적응 유지보수

변경된 환경(하드웨어/소프트웨어)에 따라 적절하게 조화를 이루도록 소프트웨어를 변경시키는 활동을 의미한다.

3) 완전 유지보수

시스템의 완벽성을 추구해가는 과정으로 사용자가 소프트웨어를 사용하면서 필요한 새로운 기능을 추가하거나, 변경하는 것을 의미한다.

4) 예방 유지보수

미래의 유지보수를 위해 준비하는 활동을 의미한다.

■ 형상항목의 정합성

■ 가. 정의

정합성이란 가지런히 들어맞으며, 이론의 내부에 모순이 없는 것, 공리계(公理界)에서 어떤 논리식에 대해 그것과 그것의 부정이 동시에 성립되지 않는 것이란 의미이다.

■ 나. 정합성 검증 절차

1) 형상관리 계획 수립

형상관리를 체계적으로 수행하기 위해 형상관리 계획을 수립해야 한다.

2) 산출물 평가 시행

소프트웨어의 형상에 대하여 유지보수 및 변경사항에 대하여 주기적으로 평가하고 확인해야 한다.