66. 애플리케이션 테스트 프로세스

테스트 계획 -> 테스트 케이스 -> 테스트 시나리오 -> 테스트 수행-> 테스트 결과 평가 및 리포팅 -> 결함 추적 및 관리

테스트 계획서 : 목적 범위 수행 절차 대상 시스템 구조 조직의 역할 및 책임

테스트 케이스 : 사용자 요구사항을 얼마나 준수하는지 확인하기 위한 입력 값, 실행 조건, 기대 결과 등으로 만들어진 명세서

테스트 시나리오 : 테스트를 수행할 여러 개의 테스트 케이스의 동작 순서를 기술한 문서

테스트 결과서 : 테스트 결과를 비교 분석

1. 테스트 기획

테스트 정의 , 대상 범위 결정

테스트 대상 시스템의 구조 파악, 조직 및 비용 산정, 시작 종료조건 정의

시작조건 : 테스트 계획, 일정, 환경 구축, 사용자 요구사항에 대한 명세서, 투입 조직 및 참여 인력의 역할과 책임 등이 완료되면 테스트가 시작되도록 조건을 정의

종료 조건 : 정상적 완료 후 , 일정이 만료 후, 비용이 소진 후 중요도에 따라 조건을 다르게 지정 가능

-테스트 기획서를 작성

2. 테스트 분석 및 디자인

목적과 원인 검토 요구사항 분석 리스크 분석 및 우선 순위 결정 테스트 데이터 테스트 환경 테스트 도구 준비

@ 테스트 데이터 : 시스템의 기능이나 적합성 등을 테스트 하기 위해 만든 데이터 집합 기능을 차례대로 테스트 할 수 있도록 만든 데이터

실제 데이터/ 가상 데이터

3. 테스트 케이스 및 시나리오 작성

설계 기법에 따라 케이스를 작성하고 검토 및 확인 한 후 테스트 시나리오 작성

4. 테스트 수행

테스트 환경을 구축 후 테스트 수행 결과를 측정하여 기록

5. 테스트 결과 평가 및 리포팅

결함을 중점 실행 절차 최적화하여 다음에 적용

6. 결함 추적 및 관리

동일한 결함 발견 시 처리시간 단축, 결함의 재발 방지

결함 관리 프로세스

-에러 발견, 에러 등록, 에러 분석, 결함 확정, 결함 할당, 결함 조치, 조치 검토 및 승인

67. 테스트 케이스/ 시나리오/ 오라클

@ 테스트 케이스 : 구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지 확인하기 위해 설계된 입력 값, 실행 조건, 기대 결과 등으로 구성된 테스트 항목에 대한 명세서. 명세 기반 테스트의 설계 산출물에 해당, 미리 설계하면 테스트 오류 방지, 인력 시간 등의 낭비 줄임. 시스템 설계 시 작성

순서 : 1. 테스트 계획 및 검토 및 자료확보 2. 위험 평가 및 우선 순위 결정 3. 테스트 요구사항 정리 4. 테스트 구조 설계 및 테스트 방법 결정 5. 테스트 케이스 정의 6. 테스트 케이스 타당성 확인 및 유지보수

@ 테스트 시나리오 : 적용하는 순서에 따라 여러 개의 테스트 케이스들을 묶은 집합, 구체적인 절차를 명세한 문서. 테스트 시나리오에는 테스트 순서에 대한 구체적인 절차, 사전 조건, 입력 데이터 등이 설정되어 있다. 순서를 미리 정함으로써 테스트 항목을 빠짐없이 수행할 수 있다.

유의사항 : 시스템별, 모듈별, 항목별 등과 같이 분리해야함. 요구사항과 설계문서 등을 토대로 작성, 식별자 번호, 순서 번호, 테스트 데이터, 테스트 케이스, 예상 결과 , 확인 등을 포함해서 작성해야 한다., 유스 케이스 간 업무 흐름이 정상적인지를 테스트 할 수 있도록 작성해야 한다. 모듈, 프로그램 간의 연계가 정상적으로 동작하는지를 테스트 할 수 있어야 한다.

@ 테스트 오라클

결과가 올바른지 판단하기 위해 사전에 정의된 참 값을 대입하여 비교하는 기법 및 활동, 테스트 케이스의 예상 결과

특징 : 제한된 검증, 수학적 기법, 자동화 가능(실행, 결과 비교, 커버리지 측정)

1. 참 오라클 : 모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공

2. 샘플링 오라클 : 특정한 몇몇 케이스

3. 추정 오라클 : 샘플링을 개선 특정 케이스의 결과 제공 나머지는 추정

4. 일관성 검사 오라클 : 변경되었을 때 전과 후의 값이 동일한지 확인

68. 테스트 자동화 도구

반복적으로 수행하던 테스트 절차를 스크립트 형태로 구현하는 자동화 도구를 적용함으로써 쉽고 효울적 테스트를 수행

휴먼 에러를 줄이고 정확성을 유지하며 품질 향상

장점 : 테스트 데이터의 재입력, 재구성 과 같은 반복적 작업을 자동화 함 -> 인력 시간 줄임 / 향상된 테스트 품질 보장 / 테스트 결과에 대한 객관적인 평가 기준 제공 / 그래프 등 다양한 표시 형태로 제공 / UI가 없는 서비스도 정밀 테스트 가능

단점 : 자동화 도구의 교육이 필요/ 단계 별로 적용하기 위한 시간 비용 노력이 필요 / 비공개 상용 도구의 경우 고가의 추가비용

고려사항 : 재사용 및 측정이 불가능한 프로그램은 제외, 용도에 맞게 적절한 도구 사용, 환경 설정, 습득 시간 고려, 프로젝트 초기에 테스트 엔지니어의 투입 시기 계획

테스트 자동화 도구 유형

1. 정적 분석 도구 - 실행하지 않고 분석. 코딩 표준, 스타일, 복잡도 및 남은 결함 등을 발견하기 위해 사용 수행하는 사람이 작성된 코드를 이해하고 있어야 함

2. 테스트 실행 도구 - 스크립트 언어를 사용하여 테스트를 실행 스크립트를 작성한 후 실행

- 데이터 주도 접근 방식 : 스프레드시트에 테스트 데이터를 저장하고 이를 읽어 실행하는 방식 다양한 테스트 데이터를 동일한 테스트 케이스로 반복하여 실행

- 키워드 주도 접근 방식 : 스프레드 시트에 동작을 나타내는 키워드와 테스트 데이터를 저장 실행 키워드로 테스트 정의

3. 성능 테스트 도구 - 애플리케이션의 처리량, 응답 시간, 경과시간, 자원 사용률 등을 인위적으로 적용한 가상의 사용자를 만들어 테스트 수행

4. 테스트 통제 도구 - 테스트 계획, 관리, 수행, 결함 관리/ 형상관리 도구, 결함 추적/ 관리 도구

5. 테스트 하네스 도구

컴포넌트 및 모듈을 테스트 하는 환경의 일부분, 테스트를 지원하기 위해 생성된 코드와 데이터, 실행될 환경을 시물레이션 하여 컴포넌트 및 모듈이 정상적으로 테스트 되도록 한다.

- 테스트 드라이버 : 하위모듈을 호출하고 파라미터 전달하고 결과 도출

- 테스트 스텁 : 제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행, 일시적으로 필요한 조건만을 가지고 있는 테스트용 모듈

- 테스트 슈트 : 테스트 대상 컴포넌트, 모듈, 시스템에 사용되는 테스트 케이스의 집합

- 테스트 케이스 : 요구사항을 정확하게 준수했는지 확인하기 위한 입력 값, 실행 조건, 기대 결과 등으로 만들어진 테스트 항목의 명세서

- 테스트 스크립트 : 자동화된 테스트 실행 절차에 대한 명세서

- 목 오브젝트 : 행위를 조건부로 입력해 두면, 그 상황에 맞는 예정된 행위를 수행

하네스 : 드 스슈케 스목

69. 결함 관리

: 오류 발생, 작동 실패 등과 같이 소프트웨어가 개발자가 설계한 것과 다르게 동작하거나 다른 결과가 발생, 예상과의 차이, 업무내용과의 불일치

결함 관리프로세스

결함 관리 계획 -> 결함 기록 -> 결함 검토 -> 결함 수정 -> 결함 재확인 -> 결함 상태 추적 및 모니터링 활동 -> 최종 결함 분석 및 보고서 작성

결함 상태 추적

지속적으로 상태 변화를 추적하고 관리해야함, 발견된 결함에 대해 결함 관리 측정 지표의 속성 값들을 분석하여 향후 결함이 발견될 모듈 또는 컴포넌트를 추정할 수 있다.

결함 분포 : 모듈 또는 컴포넌트의 특정 속성에 해당하는 결함 수 측정

결함 추세 : 테스트 진행 시간에 따른 결함 수의 추이 분석

결함 에이징 : 특정 결함 상태로 지속되는 시간 특정

결함 추적 순서

결함 등록 -> 결함 검토 -> 결함 할당 -> 결함 수정 -> 결함 조치 보류 -> 결함 종료 -> 결함 해제