

게임 QA - ISTQB

- 소프트웨어 수명주기와 테스트

NEXON CQA TEAM
meircell@nexon.co.kr
2011. 10. 05

CONTENTS

1. 소프트웨어 개발모델

2. 테스트 레벨

3. 테스트 유형

4. 유지보수 테스트

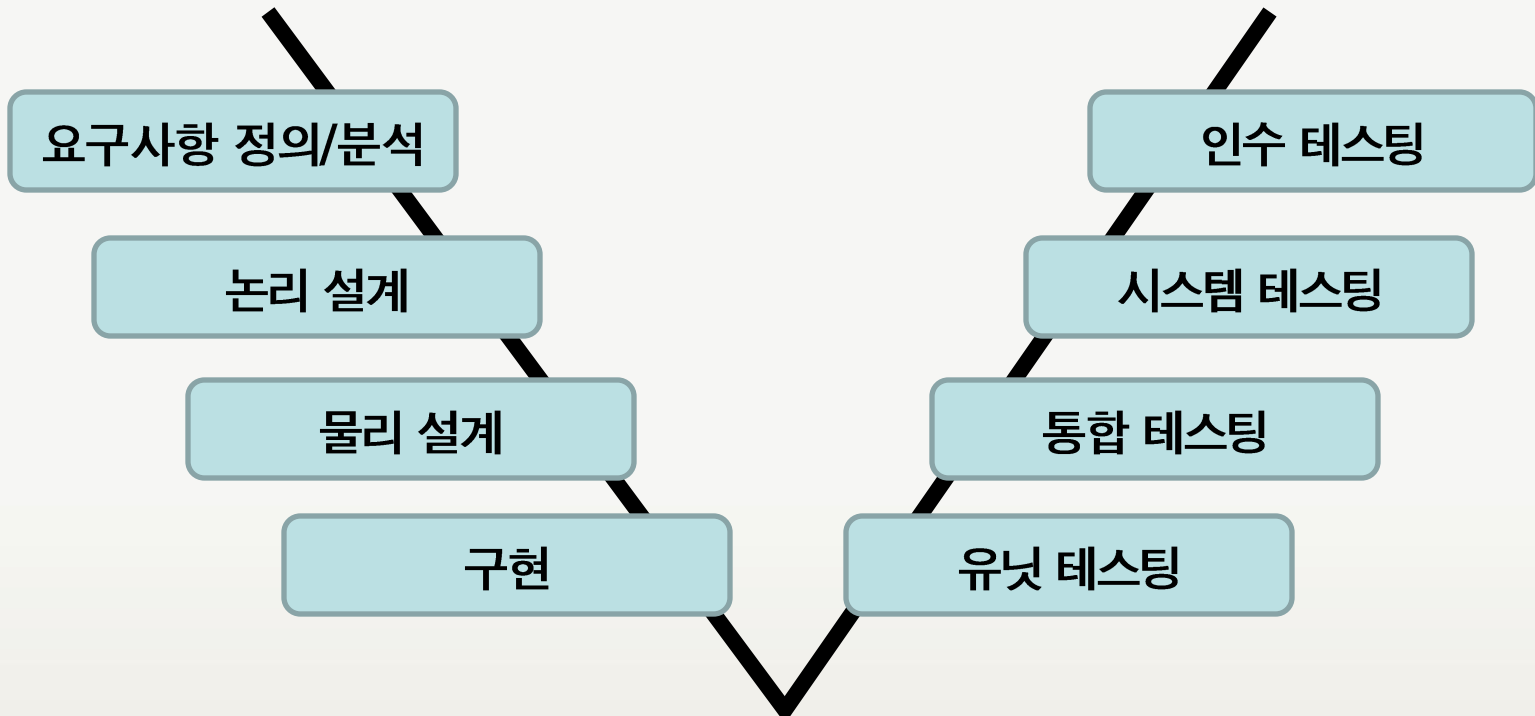
게임 QA - ISTQB

1. 소프트웨어 개발 모델

1. 소프트웨어 개발 모델

1. V-모델(순차적 개발 모델)

요구사항 정의/분석 - 시스템 설계 - 구현 - 테스트 : 폭포수 모델이 근간



1. 소프트웨어 개발 모델

1. V-모델(순차적 개발 모델)

각 개발 단계에 따라 테스트도 진행되며, 각각 짝을 이뤄 V 형태를 이룸

4단계 V 모델을 기본으로 다양한 변형이 있음

- 기본 형태 : 요구사항 분석 - 논리 설계 - 물리 설계 - 코딩

1. 소프트웨어 개발 모델

2. V-모델을 통해 보는 테스트 개념

테스트 레벨

개발 초기 단계 테스트 수행 의미

결함 예방 차원에서의 테스트 의미

Verification과 Validation의 의미

1. 소프트웨어 개발 모델

2-1. 테스트 레벨

각 개발 단계에 대응되는 테스트의 단위

- 컴포넌트 테스트, 통합 테스트, 시스템 테스트, 인수 테스트

테스트 레벨에 따라서 테스트 전략, 기법, 수행 주체, 완료 기준이 바뀜

- 모든 레벨에 사용되는 계획 : 마스터 테스트 계획

각각의 테스트 레벨은 독립적이며 별도의 보고를 진행

각 테스트 레벨은 종속성을 지니므로, 다른 테스트 레벨로 이전하기 위한 종료 및 시작조건을 정의하는 것이 필요

1. 소프트웨어 개발 모델

2-2. 개발 초기 단계 테스트 수행 의미

개발 산출물을 리뷰 형태로 검토하면서 결함을 발견하는 정적 테스트

정적 테스트를 진행하면서 테스트 케이스 생성 및 결함 발견을 할 수 있음

이 리뷰 과정에서는 다른 테스트를 통해서 발견되지 않는 결함이 발견됨

1. 소프트웨어 개발 모델

2-3. 결함 예방 차원에서의 테스트 의미

개발 전 결함 확인으로 결함을 예방함

- 많은 결함이 기획 단계에서 오류를 내포하고 있음

개발 단계의 후반으로 갈수록 결함에 대한 비용이 증가됨

- 결함의 초기 발견은 비용을 획기적으로 감소 시킬 수 있음

1. 소프트웨어 개발 모델

2-4. Verification과 Validation의 의미

Verification

- 개발 단계의 산출물이 해당 단계에서 설정한 조건을 만족하는지 검증
- 결과물이 기획에 부합하는지 여부

Validation

- 개발 단계의 산출물이 사용자의 요구를 만족하는지 검증
- 결과물이 기획에 부합할 뿐만 아니라, 사용자의 요구를 만족하는지 여부

1. 소프트웨어 개발 모델

3. 반복적 - 점증적 모델

[요구사항 분석 - 설계 - 구현/테스팅]의 개발주기를 짧게 연속적으로 반복하는 활동
주로 초기에 리스크가 높은 모듈/아키텍처에 집중해서 리스크를 초반으로 분산

주요 관련 모델

- 애자일 모델
- RUP
- RAD
- 이해관계자 중심의 소프트웨어 개발
- 프로토타이핑

개발주기를 거치면서 개발된 모듈이 증가하면서 부분 시스템을 구성

최초 작업 이후 증분 모듈에 대해서 리그레션 테스팅을 하는 것이 중요해짐

1. 소프트웨어 개발 모델

4. 개발 수명주기 모델에서의 테스트

개발 모델과 무관한 성공적인 테스트를 위한 요건

- 개발 활동에 상응하는 테스트 활동
- 각 테스트 레벨에는 레벨에 맞는 목적
- 각 단계에 맞는 테스트 분석과 설계는 개발 활동 진행 중 시작되어야 함
- 단계 초기 개발 산출물 초안이 나오면 테스터는 리뷰에 참여해야 함

테스트 레벨은 프로젝트 등에 따라 재조정되거나 합쳐짐

- 반드시 4개 레벨이 딱 떨어지는 것은 아님

게임 QA - ISTQB

2. 테스트 레벨

2. 테스트 레벨

1. 컴포넌트 테스트

유닛 테스트와 동의어

테스트가 가능한 최소 단위로 나누어진 소프트웨어에서 결함을 찾고 검증하는 작업

시스템의 다른 부분에서 격리해서 독립적으로 수행

상황에 따라 스텝, 드라이버, 시뮬레이터 등이 필요할 수 있음

구조적인 테스트, 기능성 테스트, 비기능성 테스트를 포함

주로 소스코드를 이용해서 코드 단위로 진행

2. 테스트 레벨

2. 통합 테스트

컴포넌트간 인터페이스 테스트 및 OS 등과 상호 연동을 테스트

- 모듈간 또는 OS, 파일 시스템 등과 정상적으로 연동되는지 테스트

테스터는 아키텍처에 대해서 충분히 이해하고 테스트를 계획해야 함

2. 테스트 레벨

3. 시스템 테스트

프로젝트 전체 차원에서의 환경하에서 진행하는 테스트

이전 테스트에서 확인하지 못한 환경특성 장애를 확인하기 위해 실제 환경을 조성
- 서버 스트레스 테스트, 보안모듈 테스트 등

일반적으로 독립적인 테스트 팀이 수행하며, 품질 평가 형태로 진행됨

테스팅을 통해서 출시 여부에 대한 정보를 산출

2. 테스트 레벨

4. 인수 테스트

해당 프로젝트를 배포하거나 사용할 수 있는지 평가

실제 사용자 및 고객이 전담하는 경우가 대부분

버그 발견은 큰 관심사가 아니며, 기능 및 비기능적인 특성에 대한 확신을 얻는 것이 목적
- CBT/OBT에서 왜 버그리포팅에 그다지 신경을 쓰지 않는지에 대한 대답...

인수 테스트 이후 다른 테스트가 추가될 수 있으므로, 언제나 최종 테스트인 것은 아님

게임 QA - ISTQB

3. 테스트 유형

3. 테스트 유형

1. 기능 테스트

프로그램이 해야 하는 [무엇]을 테스트

기능성 품질 특성

- 적합성
- 정확성
- 준수성
- 상호운영성
- 보안성

3. 테스트 유형

2. 비기능 테스트

프로그램이 [어떻게] 동작하는지를 테스트

비기능성 품질 특성

- 신뢰성
- 사용성
- 효율성
- 유지보수성
- 이식성

3. 테스트 유형

3. 구조적 테스트

특정 유형의 구조에 대한 커버리지를 평가해서 테스트의 보장성/충분함을 측정하는 테스트

커버리지 : TS에 의해 테스트 된 정도

화이트 박스 테스트 기법을 통해서 코드를 테스트
- 각 구문, 선택문 등을 모두 테스트

자동화 툴을 사용해서 테스트를 진행할 수 있음

3. 테스트 유형

4. 확인/리그레션 테스트

확인 테스트

- 결함이 발견되고 수정된 뒤, 수정되었는지 확인하는 테스트
- 제대로 수정이 되었는지 확인

리그레션 테스트

- 결함 수정으로 인해 바뀌지 말았어야 하는 부분이 바뀐 것은 없는지 확인하는 테스트
- 바뀌지 말았어야 하는 것들이 그대로 있는지 확인
- 개발 후반으로 갈수록 대상이 늘어남 : 자동화를 고려할 수 있음

3. 테스트 유형

5. 유지보수 테스트

프로그램이 배포되면, 수년이상 사용되기 때문에 시스템/환경 변화에 맞춰 수정 필요

변경

- 개선활동, 요구사항 변경 적용, 긴급 변경, 환경 변경 등으로 진행
- OS/DB 업그레이드, 새로운 취약점 등이 대상

마이그레이션

- 하나의 환경에서 다른 환경으로 바꾸는 것
- 프로그램이 아예 바뀌기 때문에 관련 데이터 등의 변환 작업이 필요
- 시스템 단종 등이 원인