

게임 QA - ISTQB

NEXON CQA TEAM
meircell@nexon.co.kr
2011. 10. 19

- 정적 기법

CONTENTS

1. 정적 기법과 테스트 프로세스

2. 리뷰 프로세스

3. 도구에 의한 정적 분석

게임 QA - ISTQB

1. 정적 기법과 테스트 프로세스

1. 정적 기법

0. Dynamic & Static

동적 테스트(Dynamic testing)

- 실제 구현된 시스템을 실행해서 테스트
- 실제적인 사용/운용에 대한 테스트
- 구현 → 테스트 → 수정(구현) → 테스트 → 완료

정적 테스트(Static testing)

- 시스템에 대한 요구사항 정의서, 설계서, 사양서, 소스코드 등
- 리뷰, 인스펙션, 정적 분석
- 개발 프로세스 초기에 결함 예방 및 이를 통한 비용 감소

1. 정적 기법

0. Static testing Issue

개발자가 진행해야 하는 테스트로 인식

- 테스트 전문가 조차도 개발자가 해야한다고 인식

테스트 전문가가 테스트 관점에서 수행/참여 해야 함

- 전문적이고 효율적인 테스트 및 결과

1. 정적 기법과 테스트 프로세스

1. 리뷰에 대해서

정적 테스트 - 실행 없이 테스트 하는 기법

- 리뷰 등의 수동적 기법
- 정적 분석 자동화 도구를 이용한 자동적 방법

동적 테스트를 실행하기 전에 적절하게 수행 가능

결함 수정 비용

- 동적 테스트 : 높음
- 정적 테스트 : 낮음

수동적으로 진행되지만, 자동화 툴도 있음

- 개발 산출물을 검토하고 이에 대한 의견 제시
- 구조적인 부분들은 툴을 이용해서 자동으로 테스트 목록 구성 가능

1. 정적 기법과 테스트 프로세스

2. 리뷰의 이점과 목적

리뷰의 이점

- 조기 결함 발견 및 수정
- 개발 생산성 향상 및 기간 단축
- 테스트 비용 감소 및 시간 단축
- 개발 수명주기 전체에 걸친 비용 감소
- 결함 감소 (∴ 품질 향상)
- 개발 및 타부서간 커뮤니케이션 향상 : 커뮤니케이션에 대한 준비

리뷰의 목적

- 결함 발견
- 동적 테스트와 상호 보완 : 서로 다른 종류의 결함들을 효과적으로 발견
- 동적 테스트 : 장애 / 리뷰 : 결함(장애의 원인)

1. 정적 기법과 테스트 프로세스

3. 리뷰를 통해 찾기 쉬운 결함 종류

표준 위반

요구사항 결함

개발 설계 결함

불충분한 유지보수성

부정확한 인터페이스 명세

1. 정적 기법과 테스트 프로세스

4. 리뷰와 테스트

리뷰 등의 정적 기법들은 그 관심에 비해 적극적으로 다루어지지 않고 있음

최근의 주요 관심사

- 결함 예방 차원의 활동
- 결함을 빠르게 발견해서 비용 감소
- 개발 단계에서의 하위레벨 테스트(유닛, 통합 테스트) 강조
- 조기 테스트 설계

1. 정적 기법과 테스트 프로세스

5. 조기 테스트 설계

조기 테스트 설계

- 코딩 전 초기 산출물을 통해서 테스트 케이스 생성
- 개발 문서의 결함 발견
- 향후 유입 될 수 있는 결함 예방

초기 산출물은 변경될 가능성이 높음

- 조기 테스트 설계를 통해 생성된 테스트 케이스가 무의미 해 질 수 있음
- 테스트 케이스의 유지보수 비용 부담
- 테스트 케이스가 쓸모 없어지더라도 결함 예방 활동으로써 중요
- 그러므로 리스크가 높거나 중요 기능을 우선적으로 작업

주요 관심사

- 결함 예방 차원의 활동
- 결함을 빠르게 발견해서 비용 감소
- 개발 단계에서의 하위레벨 테스트(유닛, 통합 테스트) 강조

게임 QA - ISTQB

2. 리뷰 프로세스

2. 리뷰 프로세스

0. 리뷰 프로세스

리뷰는 공식적인 것 부터 비공식적인 것 까지 다양한 형태

리뷰 형태에 대한 선택 기준

- 개발 프로세스의 성숙도
- 법적 또는 규제적 요구사항(NASA의 납품 기준)
- 감사에 대한 필요성
- 달성하고자 하는 합의된 목적

단순한 검토로 인식되어 효과에 대한 의구성이 있음

2. 리뷰 프로세스

1. 리뷰 효과 저하

리뷰 효과 저하 원인

- 중재자 및 검토자의 낮은 능력
- 절차 미준수 및 역할 미분담
- 적절하지 못한 리뷰 형식 및 인원 구성

리뷰 효과를 높이기 위한 사항

- 저하 원인을 반대로 진행

* 왜 문제점에 대해서 연구하는가?

∴ 반대로 하면 성공하지 못하더라도 실패하지는 않음

2. 리뷰 프로세스

2. 공식 리뷰 단계

계획 활동

- 참가인원 선정 및 역할 분배, 시작/종료 기준 정의, 대상 정의

시작

- 문서 배포 및 리뷰에 대한 설명 진행, 시작 기준 점검

개별 준비

- 각자 사전 리뷰를 통해 결함 발견

리뷰 미팅

- 개별 준비 결과 토의, 회의록 작성, 결함 처리 방안 제안 및 결함 여부 결정

재작업

후속 처리 확인

2. 리뷰 프로세스

3. 역할과 책임

관리자

- 리뷰 실행여부 결정 및 프로젝트 일정에 리뷰시간 할당, 리뷰 결과 확인 및 승인

중재자

- 리뷰 리딩, 리뷰 계획 및 미팅 진행, 후속 조치 처리 여부 등의 추적/관리
- 참석자들의 다양한 의견을 중재(리뷰 성패를 좌우함)

저자 : 리뷰 대상 문서 작성자/책임자

검토자

- 실제 리뷰 진행자, 결함 발견 및 정리, 다양한 관점과 역할을 대표

기록자

- 모든 이슈 , 문제점 등을 기록/정리하고 문서화

2. 리뷰 프로세스

4. 테스트 전문가의 역할

검토자로서 리뷰에 참여

테스팅 관점에서 발견한 결함과 검토 의견에 중점

테스트 케이스를 생성하면서 발견된 [다른 종류]의 결함으로 리뷰에 기여

리뷰 진행을 통해 이후 진행될 테스트 수행을 위한 준비

2. 리뷰 프로세스

5. 리뷰 참석으로 테스터가 얻는 것

조기 테스트 설계로 테스트 준비시간 단축

테스트 대상에 대한 상세한 지식

개발 히스토리 파악 : 효율적 커뮤니케이션의 기반

2. 리뷰 프로세스

6. 비공식적 리뷰

공식적 절차 없음

개발팀 자체 리뷰 또는 선임자의 설계/코드 리뷰

선택적 문서화

리뷰어에 따라 성과가 좌우됨

주요 목적 : 저렴한 방법으로 일정 수준의 성과 달성

2. 리뷰 프로세스

7. 기술적 리뷰

동료/기술 전문가가 참여하는 결함 발견을 위한 문서화되고 정의된 프로세스 존재

관리자 개입 없이 동료 검토 형태로 수행 가능

중재자가 미팅을 주도하는 것이 이상적

미팅전 사전 준비 필요

체크리스트, 리뷰 리포트, 발견한 인시던트 리스트, 관리자 참여 활용을 선택적 적용

공식적이거나 비공식적으로 진행

성공할 경우 일관되고 정량적인 효과 도출

주요 목적 : 기술적 문제 해결, 토론, 의사결정, 대안평가, 결함 발견,
명세서 또는 표준과의 적합성 검토

2. 리뷰 프로세스

8. 워크쓰루

저자에 의한 진행 및 제어

시나리오 사용, 여행 연습, 동료 그룹 검토

시간 및 인원수 등에 제한이 없고, 상황에 따라 변경(Open-ended) 세션

준비과정, 검토자 지정, 리뷰 리포트 준비, 발견한 인시던스 리스트, 기록자 지정(선택적 적용)

공식적이거나 비공식적으로 진행

주요 목적 : 학습, 시스템에 대한 이해 향상, 결함 발견

2. 리뷰 프로세스

9. 인스펙션

훈련된 중재자에 의한 진행/제어

주로 동료 검사

역할이 정의되어 있음

메트릭을 수집하고 활용함

체크리스트와 규칙을 기반으로 시작과 종료 조건이 있는 정식 프로세스 존재

미팅 전 준비 과정 필요

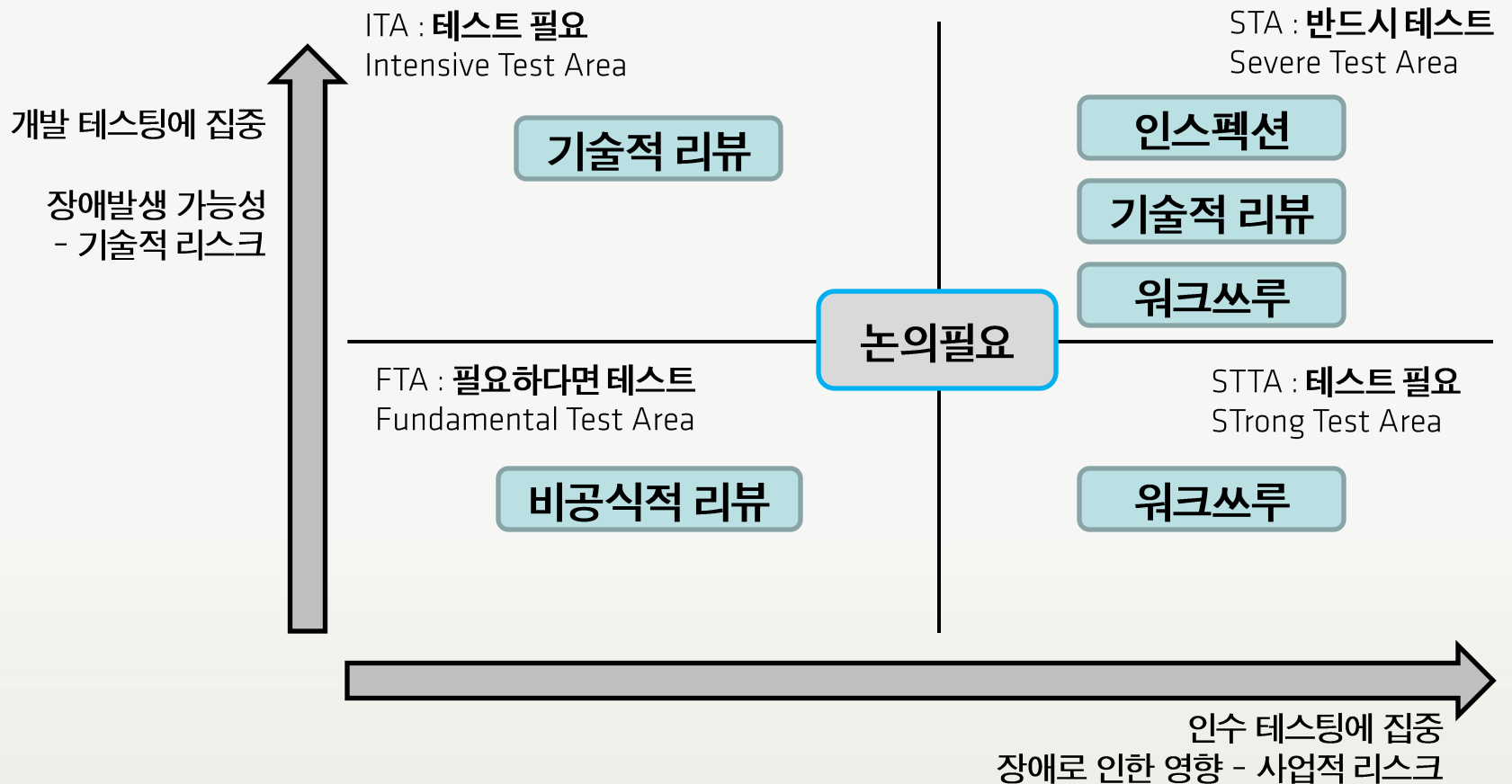
인스펙션 리포트와 발견사항(인시던트) 리스트 산출

정식적인 후속 처리 확인 프로세스 및 선택적인 리뷰 프로세스 개선 활동

주요 목적 : 결함 발견

2. 리뷰 프로세스

10. 리뷰와 리스크 분석 연계



2. 리뷰 프로세스

11. 리뷰의 성공요소

명확하게 정의된 목적과 적합한 인력

결함 발견을 환영하는 분위기와 결함에 대한 객관적인 표현

사람관련 이슈와 심리적인 측면 고려(제품을 리뷰하고 저자는 리뷰하지 말라)

상황에 따른 적절한 리뷰 기법 적용

체크리스트와 역할 분담 활용

관리자의 적극적인 리뷰 프로세스 지원

학습과 프로세스 개선에 대한 강조

리뷰 경험과 효과를 내재화 하고 지속적으로 적용

게임 QA - ISTQB

3. 도구에 의한 정적 분석

3. 도구에 의한 정적 분석

1. 특징

동적 테스트로 찾기 힘든 결함 발견

장애보다는 그 원인인 결함을 발견

프로그램 코드 뿐만 아니라 HTML, XML 같은 리소스도 분석

3. 도구에 의한 정적 분석

2. 분석 가치

테스트 실행 전에 조기 결함 발견

계산을 통한 코드와 설계의 의심스러운 부분에 대한 조기 경보

동적 테스팅으로는 발견되지 않는 결함

소프트웨어 모델상의 의존도와 불일치성 발견

코드 설계와 유지보수성 향상

3. 도구에 의한 정적 분석

3. 발견되는 주요 결함

정의되지 않은 값으로 변수 참조

모듈과 컴포넌트 간에 일관되지 않은 인터페이스

사용되지 않는 변수 및 코드

코딩 표준 위반

보안 취약성

코드와 소프트웨어 모델의 구문 규칙 위반