A Brief Essay on Software Testing

컴퓨터공학부
박선율

ABSTRACT

- Testing :
  - 소프트웨어 개발 과정에서 중요한 부분이다.
  - 버그의 발견에 구애 받지 않는다.
  - 적절한 기능수행에서의 신뢰를 증가시킨다.
  - 관련 활동들은 전체적인 개발과정을 포함한다.

INTRODUCTION

- Testing은 몇몇의 높은 노력을 요하는 업무을 포함하는 challenging activity
- Test selection , ability to launch the test , deciding the test outcome , evaluating and finding , judging

TERMINOLOGY AND BASIC CONCEPS

- On the Nature of the Testing Discipline
  - Testing의 목적 : 소프트웨어가 적절한 수행을 할 것이라는 것을 입증하여 소프트웨어에 대한 신뢰성을 증가시키는 것.
  - Test techniques는 두가지로 나눌 수 있다
    - Static analysis techniques
    - Dynamic analysis techniques
    - 둘은 서로 상호보완적이다.

TERMINOLOGY AND BASIC CONCEPS

- A General Definition
  - Software testing은 명세서에 기입된 예상되는 행동에 기대어, 대게 무한한 실행 범위로부터 적당히 선택된 test case의 유한 집합에서 프로그램 실행의 동적 검증으로 구성된다.

TERMINOLOGY AND BASIC CONCEPS

- Fault Versus Failure
  - Failure : 프로그램이 요구된 기능을 수행하는 것에 대한 명백한 무능. 잘못된 output, 비정상적인 결과 등 시스템 고장의 증거가 된다.
  - Fault : failure의 원인. 예를 들어, missing of incorrect piece of code.
  - Error : 어떤 실행이 일어날 때까지, fault가 오랫동안 발견되지 않고 남아 있을 때 발생되는 불안정한 상태. Failure의 원인이 될 수 있다.

Fault -> Error -> Failure

TERMINOLOGY AND BASIC CONCEPS

- The Notion of Software Reliability
  - 몇 개, 또는 많은 fault들은 필연적으로 testing과 debugging을 피해간다.
  - 소프트웨어가 주어진 시간과 환경에서 failure 없이 실행될 수 있는 가능성을 측정하는 것이다.
  - Software reliability를 평가하는 것은 testing을 통해 제공될 수 있다.

TYPES OF TESTS

- Static Techniques
  - 오로지 software models and code등의 project documentation의 시험에 기반한다.
  - 일반적인 개발 과정에서 can be applied :
    - Language syntax, completeness 등을 확인하는 requirements stage에서.
    - 모순의 발견과 요구 충족을 평가하는 design phase에서.
    - Implemented product를 위해 채택된 form을 확인하기 위한 implementation phase 동안에.

TYPES OF TESTS

- Traditional static technique들은 포함한다 :
  - Software inspection.
  - Software reviews.
  - Code reading.
  - Algorithm analysis and tracing.
- 위의 techniques에 따라 적용되는 process들은 무겁고 시간이 오래 걸린다. 이를 극복하기 위해 Formal method 사용에 의지한 static analysis technique가 제안된다.

TYPES OF TESTS

- Dynamic Techniques
  - Obtain information of interest about a program.
  - Standard dynamic techniques는 testing 과 profiling을 포함한다.
  - Specific dynamic techniques는 simulation, sizing, timing analysis, prototyping을 포함한다.
- Objective of Testing
  - Acceptance/qualification testing : 최종 테스트
  - Installation testing : 대상환경에서 시스템 설치 확인

TYPES OF TESTS

- Alpha testing : in-house user들에게 배포
- Beta testing : 외부의 tester들에게 배포
- Reliability Achievement : 신뢰성 향상
- Conformance testing/functional testing : 의도된 기능 충족 확인
- Regressing testing : retesting.
- Performance testing : 요구된 성능을 충족하는지 확인
- Usability testing : 유용성 테스트
- Test-driven development : test case specification 사용 추진

TEST LEVELS

- Unit Test :
  - Unit은 소프트웨어의 가장 작은 조각이다.
  - Unit test의 목적은 unit이 기능적 명세서를 만족시키는지, 의도된 디자인 구조와 일치되는지 보증하는 것이다.

TEST LEVELS

- Integration Test :
  - Integration은 소프트웨어 조각 또는 구성요소가 모여서 더 큰 구성요소가 되는 과정이다.
  - 발생할 수 있는 문제를 드러내는데 초점을 맞춘다.
  - 명세서에 따라 각 요소들이 상호작용 하는지 입증하는 데 목적을 둔다.

TEST LEVELS

- Incremental or nonincremental approach
  - Incremental approach
    - Big-bang testing : 요소들이 서로 연결되어 있어서 한 번에 테스트되어 질 수 있다.
  - Nonincremental approach
    - Top-down : 메인 프로그램에서 하위의 것으로 내려가면서 한번에 하나씩 모듈을 통합시킨다.
    - Bottom-up : 모듈의 가장 낮은 부분에서 시작해서 점차적으로 위쪽으로 연결시키면서 전체적인 시스템을 구성한다.

TEST LEVELS

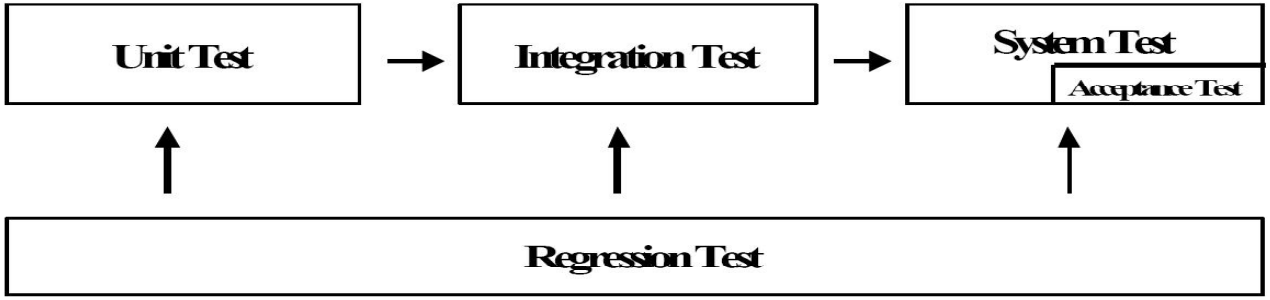
- System Test
  - 시스템이 실제 유저의 요구에 따라서 실행되는지 입증하는데 초점을 둔다.
  - System testing의 기본 목표를 요약하자면 :
    - Unit of integration testing 동안 발견되지 않은 failures를 발견한다.
    - 요구된 성능을 충족시킴으로써 개발된 제품의 신뢰를 증가시킨다.
    - 제품 공개를 결정하는데 유용한 정보를 모은다.

TEST LEVELS

- 각 시스템 function이 예상된대로 작동하고, 모든 failure들은 드러난다는 것을 보증한다.
- 실행, 보안, 신뢰도, 복구를 위한 testing을 포함한다.
- Acceptance testing
  - 새로운 level이라기 보다는 시스템 testing의 연장이다.
  - User가 시스템의 기능을 완전히 배우고 받아들이는데 요구되는 노력을 파악하는 것이 목적이다.

TEST LEVELS

- Regression Test
  - Testing level을 구분하지 않는다.
  - 코드 등 부분적 변경 후에, unit, combination of components, 또는 whole system을 retesting할 것을 지시한다.
  - 다양한 technique들은 testing 비용이 감소되고 좀 더 효율적 이도록 발전되고 있다.



STRATEGIES FOR TEST CASE SELECTION

- Selection Criteria Based on Code
  - Motivation : potential failure를 찾아내는 것.
  - 반복적으로 선택된다.
- Selection Criteria Based on Specifications
  - Equivalence classes : 각각의 input condition은 반드시 개별적으로 고려되어야 한다.
  - Boundary conditions : 직관적 사실에 기반.
  - Cause-effect graphs : 입력조건의 가능한 조합을 체계적으로 탐구하는데 사용할 수 있는 combinatorial logic networks.

STRATEGIES FOR TEST CASE SELECTION

- Other Criteria
  - Based on Tester's Intuition and Experience
    - 가장 넓게 사용되는 technique중 하나.
    - 테스터의 비슷한 프로그램에서의 경험, 직관, 기술에서 이끌어 내어진다.
  - Fault-Based
    - 있음직한 fault를 명확하게 나타내는 것을 목표함.
  - Based on Operational Usage
    - 소프트웨어의 작동환경을 가능한 가깝게 재연한다.

TEST DESIGN

- Traditional test process 는 다음의 subsequent phase들을 포함한다.
  - Planning, Design, Execution, Results evaluation
- Design substeps :
  - Test의 목적을 정한다
  - Test case 명세서를 정의한다.
  - Test 절차를 디자인 한다.
  - 통과/실패의 기준을 정한다.

TEST EXCUTION

- Launching the Tests
  - Code-based criterion :
    - Entry-exit path 제공.
  - Specification-based criterion :
    - Test cases는 명세서의 추상레벨에서 지정된 이벤트의 결과에 일치된다.

TEST EXCUTION

- Test Oracles
  - 주어진 테스트에서 프로그램이 올바르게 수행되는지 아닌지 판단한다.
  - 항상 결정 되는 것이 아니고, 이럴 경우의 출력은 결정적이지 않은 것으로 구분된다.
  - Regression testing의 경우, 시스템의 이전 버전이 될 수 있다.
  - 항상 올바른 판단을 하는 것이 아니므로, 정확성이 측정되어 소개된다.

TEST EXCUTION

- Test Tools
  - Test harness
  - Test generator
  - Capture/replay
  - Oracle/file comparators/assertion checking
  - Coverage analyzer/instrumenter
  - Tracers
  - Reliability evaluation tools

TEST DOCUMENTATION

- Test documentation은 testing phase를 제어하고 조정하는 것에 기여한다.
  - Test Plan : 테스트 항목, pass/fail 기준, 하드웨어, 소프트웨어 같은 testing에 필요한 환경등을 정의한다.
  - Test Design Specification : 테스트되기 위한 기능을 설명한다.
  - Test Case Specification : 테스트가 실행되기 위해 요구되는 input/output을 정의한다.

TEST DOCUMENTATION

- Test Procedure Specification : 테스트 단계와 테스트 케이스 세트를 실행하기 위해 필요한 특별한 요구들을 기입한다.
- Test Log : 실패가 발생했는지, 테스트가 완료되었는지 등을 포함하여 테스트 결과를 기록한다.
- Test Incident or Problem Report : input을 포함하여, 예상되었고 관찰된 결과, 날짜, 시간, 절차, 환경 등의 일어난 일을 기술한다.

TEST MANAGEMENT

- Testing phase에서 성공적인 testing을 위한 가장 중요한 요소는 협력적인 태도이다.
- Manager는 개발 기간 동안에 오류발견에 대해 호의적인 태도를 갖도록 하는데 중요한 역할을 한다.
- Management는 사람, 도구, 정책 등과 함께 구성된다.

TEST MANAGEMENT

- The main manager's activity
  - 업무의 완료 시기를 정한다.
  - 업무 수행에 필요한 자원과 노력을 추정한다.
  - 업무와 연관된 risk의 양을 정한다.
  - 노력/비용을 추정한다.
  - 품질 관리 대책을 마련한다.

TEST MEASUREMENT

- Evaluation of the Program under Test
  - Program measurement to aid in test planning and design.
    - Linguistic measures. : 프로그램 또는 명세서의 특성에 기반.
    - Structural measures : object들 사이의 구조적 관계에 기반.
    - Hybrid measures : 언어적, 구조적 특성에 기반.
  - Fault density : 프로그램의 크기와, fault의 갯수와의 비율로 측정된다.

TEST MEASUREMENT

  - Life testing, reliability evaluation : 신뢰성을 측정하거나, testing이 stop될 수 있는지 판단 한다.
- Evaluation of the Test Performed
  - Coverage/thoroughness measure : Testing 동안 실행되어지는 수많은 기능 등.
  - Effectiveness : 프로그램에서 testing의 효과를 정량화 한다.

TEST MEASUREMENT

- Measures for Monitoring the Testing Process
  - 많은 수의 fault 또는 failure를 찾아낼 수 있는 테스트 기준이 가장 유용한 것으로 간주된다.
  - 하나의 fault가 여러 개의 failure를 만들 수 있고, 또는 하나의 failure가 여러 개의 fault에서 기인될 수 있다.
  - 가장 객관적인 measure는 통계적인 것 : 테스트에 대해 통계적으로 예상한다.

CONCLUSION

- 우리는 제품의 완벽을 보장하는 test는 할 수 없을것이다. 하지만 trial-and-error방식에서 좀 더 조직적이고, 비용효율이 높고, 예측할 수 있는 engineering 분야로 변형함을 추구해야 할 것이다.
- 끝.