

품질 향상을 위한 테스트 프로세스 모델 제안

임응호*, 최진영*

*고려대학교 컴퓨터과학기술대학원

Eung-ho.lim@eds.com, choi@formal.korea.ac.kr

Quality Based Test Process Model Suggestion

Eungho Lim*, Jinyoung Choi*,

*Dept. of , Computer Science Technology, Korea University

요 약

소프트웨어 품질에 대한 관심이 증가하면서 품질을 측정하고, 평가, 개선하기 위한 테스트 프로세스에 대한 필요성이 대두했다. 현재, IEEE/Std 829-1983, SEI 의 CMM, DoD 의 MIL-STD-498, ANSI/IEEE 의 SVVP 등 테스트 프로세스에 대한 여러 국제적 지침이 많이 있지만 그 수준이 너무 일반적이고 개괄적이어서 실제 소프트웨어 개발에 적용되기엔 어려움이 있다.

본 논문에서는 IEEE/Std 829-1983 과 현재 국내 A 대기업에서 실시되고 있는 테스트 프로세스를 소개하고 이들을 조사, 비교 분석해 실제 소프트웨어 개발에 적용할 수 있는 품질 관리 계획 내 포함될 수 있는 엔터프라이즈 테스트 프로세스 (Enterprise Test Process)를 제안한다.

1. 서론

오늘날 우리는 다양한 컴퓨터 시스템 환경 속에서 살고 있다. 최근 컴퓨터 시스템의 신뢰성을 좌우하는 핵심요소가 소프트웨어 부분이라는 것이 인식됨에 따라 소프트웨어의 품질에 대한 관심이 증가되고 있다. 그러나 현재 많은 국내 업계에서 소프트웨어 개발에 있어 개발범위, 납기 기한, 예산 등의 여러 이유로 품질을 높이는 것보다는 기간과 비용에 맞추어 개발완료하기에 급급한 것이 실정이다. 이런 개발 방식으로 인해 여전히 국내 IT업계가 글로벌 IT업계에 비해 질적으로 뒤쳐져있다. 글로벌 IT기업의 경우, 전사적 개발 방법론과 더불어 관리 이론을 정립하고 있으며, 그 내부 품질 관리에 대해 정형화된 프로세스를 이용 테스트에 적용하여 개발 초기 단계에서 시작하여, 이행단계에 이르는 전체 개발 일정 속에 일종의 통제 프로세스를 이용 하여 관리 하고 있다.

소프트웨어 품질을 높이기 위해서는 품질을 측정하고, 평가, 개선하기 위한 체계적인 테스트 프로세스가 필요하다.

일반적으로 개발일정에 쫓겨 테스트라고 하면 개발이 끝난 시점에서 에러 없이 잘 작동하는지 여부만을 판단하는 경우가 소규모 기업일수록 다반사이다. 품질 관리란 개발 전 과정에 걸쳐 진행되어야 하고 개발비용의 40%정도는 투자 되어야만 원하는 수준(Grade)에 계획한 품질(Quality)를 얻을 수 있다. 이는 나아가, 추후 발생 가능한 수정작업(Corrective Action)의 비용을 줄임으로서 운영 측면의 ITSM (Information Technology Service Management) 비용 감소에 기여하게 됨으로 운영이후 발생할 유지보수 비용을 줄여나간다. 전체 System Life Cycle 을 볼 때, 초기 품질 비용의 투자가 전체 Life Cycle Cost 를 줄여나갈 수 있다.

현재 SEI 의 CMM, DoD 의 MIL-STD-498, ANSI/IEEE 의 SVVP 가 소프트웨어 개발의 품질향상을 위한 많은 지침을 제공하고 IEEE/Std 829-1983 이 소프트웨어 테스트의 국제 표준 문서를 제공하고 있다. 하지만 이런 표준 및 지침들은 특정 기술이나 방법론을 반영하지 않은 일반적인 프로세스만을 제공하고 있고, 그 수준역시 너무

개괄적이며 또한 일종의 Check List 성격을 띄고 있는 하나의 Guide Line (Best Practice)로 실제 소프트웨어 개발에 적용함에 있어 한계점이 있다.

이에 본 논문에서는 IEEE/Std 892-1983 과 국내 A 대기업의 Test Process 를 소개하고 이들의 수직적 조사와 분석을 통해 실제 소프트웨어 개발 시 활용 가능한 개발전반에 걸친 테스트 프로세스 모델을 제시해 보도록 하겠다.

2. 관련 연구

2.1 테스트의 정의 및 분류

테스트란 시스템이 정해진 요구를 만족하는지, 예상과 실제 경과가 어떤 차이를 보이는지 수동 또는 자동 방법을 동원하여 검사하고 평가하는 일련의 과정이며 소프트웨어에 현존하는 조건과 요구되는 조건 사이의 차이점을 찾아내고 그 특성을 평가하기 위한 분석 과정이다[1]. 테스트는 시각, 단계, 목적, 방법 등에 따라 아래와 같은 유형으로 나누어 볼 수 있다.

[표 1] 소프트웨어 시험의 분류

시각	단계	방법 목적	블랙박스 시험			화이트 박스시험
			기능 시험	성능 시험	스트레스 시험	구조 시험
검 중 (개발자)	단위시험 통합시험 시스템시험					
확 인 (사용자)	인수시험 설치시험					

- 단위 시험 (Unit Test) : 프로그램 단위를 보다 큰 시스템으로 통합하기에 앞서 개발 프로그래머에 각각의 모듈에 대한 테스트.

- 통합시험 (Integration) : 각 모듈 간의 상호 작용이 올바르게 되는지를 검사하는 테스트

- 시스템 시험 (System Test) : 통합 테스트가 완료된 후에 완전한 시스템에 대한 사용자의 요구사항이 만족되는지를 검증하기 위해 수행되는 테스트

- 인수 시험 (User Acceptance Test) : 시스템이 사용자의 요구 사항들을 만족시키고 있는가를 밝히기 위해서 기능, 성능 그리고 강도를 시험하는 테스트

- 설치 시험 (Installation Test) : 현장에 시스템을 설치 가능함으로써 이상유무를 확인하는 테스트

- 기능 시험 (Functional Test) : 주어진 입력에 기대되는 출력을 제공하는지 검사하는 테스트

- 성능 시험 (Performance Test) : 응답시간, 처리량, 메모리 활용도, 처리율을 검사하는 테스트

- 스트레스 시험 (Stress Test) : 과부하가 걸리때, 최저조건에 미달되고 최고조건에 초과될 때, 물리적인 충격이나 변화시에 반응 정도로 신뢰성을 검사하는 테스트

- 구조시험 (Structural Test) : 소프트웨어의 내부 논리 경로의 복잡도를 평가하는 테스트

- 블랙박스 테스트 : 소프트웨어 내부를 무시하고 외부에서 본 입력과 출력만을 가지고 평가하는 방식의 테스트

- 화이트 박스 테스트 : 소프트웨어 내부의 논리적 흐름을 살펴가면서 오류를 찾아 내는 테스트 [2].

2.2 IEEE Standard 829-1983, Standard For Software Test Documentation

이 문서는 테스트와 품질보증에 대한 국제표준으로서 테스트 수행의 표준 산출물과 구성항목을 소개하고 있다. IEEE Std. 829-1983 의 테스트 산출물로는 테스트 계획서, 테스트 설계 명세서, 테스트 케이스 명세서, 테스트 절차 명세서, 테스트 항목 전달 보고서, 테스트 로그, 테스트 사건 보고서, 테스트 요약서의 총 8 종류의 문서가 있다.

2.3 국내 A 대기업의 테스트 프로세스

국내 A 대기업에서 사용되는 테스트의 정의는 “프로그램간 상호 기능 및 인터페이스의 정상 작동 여부와 성능관련 요구사항의 수해여부를 시험하는 활동”이다[3]. 그 테스트의 종류는 아래와 같다.

[표 2] 테스트 종류

테스트	시험목표 a 및 특징 b	기준	주관자
1. 단위	a. 단위소스코드 모듈의 정상작동 b. 블랙박스/화이트박스 시험기법	소스 코드	개발자
2. 통합	a. 프로그램간 상호기능/인터페이스 b. 상향/하향/혼합식 방식	설계서	QAO
3. 시스템	a. 시스템 성능관련 고객요구 합치 b. 보안/스트레스/불륨테스트	설계서	QAO
4. 인수	a. 고객인수조건 합치	분석서	고객

[표 3] 테스트 체계

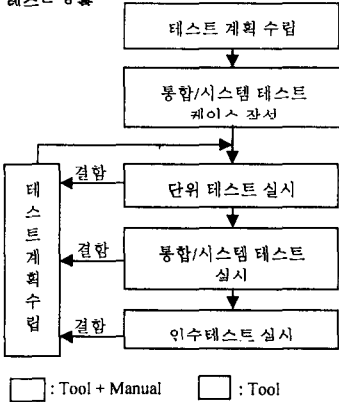
절차	준비/수행 내용	산출물
1. 테스트 계획	목적, 환경, 기준 대상, 조직, 접근방법, 일정/자원	테스트 계획서 인수테스트 기준
2. 시나리오 작성	테스트 케이스, 예상 결과, 데이터	테스트 시나리오
3. 테스트 수행	환경준비, 테스트 실시, 결합 수정, 결과정리	테스트 결과보고서 결합보고서

- 테스트 계획 : 테스트 수행을 위해 통합 테스트와 시스템 테스트를 중심으로 수립한다. 테스트의 목적, 환경, 기준대상, 조직, 접근방법, 테스트 일정/자원을 정의 하며 산출물로는 테스트계획서와 인수테스트 기준이 있다.

- 시나리오 작성 : 사용자의 요구사항이나 업무 분석자료에서 시스템 테스트 케이스를 토출하고 테스트 수행시 예상되는 결과, 테스트할 데이터를 정의한다. 산출물로는 테스트 시나리오가 있다.

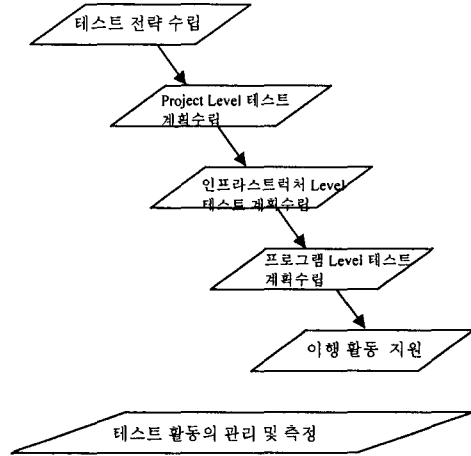
- 테스트 수행 : 프로그램간의 상호 기능 및 인터페이스의 정상 작동 여부를 통합 테스트하고, 사용자의 시스템 성능관련 요구사항의 수행, 해당 결합 내용을 보완한 후 결과보고서에 작성한다. 모든 테스트 경로가 정상인 경우 테스트를 종료한다.

테스트 종류



3. Test Process Model 제안

본 논문에서 제시하고자 하는 프로세스는 총 6 단계로 되어 있다. 지금부터 각 단계의 세부 내용 및 절차에 대해 알아보겠다.



3.1 테스트 전략 수립

테스트 프로세스의 첫번째 단계로서 테스트의 방향, 품질 요구사항 등 전반적인 계획이 수립된다. 본 단계는 소프트웨어 개발의 초기 단계로서 품질 계획 수립의 한 부분으로 포함된다. 본 단계는 프로젝트의 크기를 추산하고, 소프트웨어의 품질기준을 설정하는 테스트 전략 평가와 현재 상황 분석, 접근 분석 시행, 전략 수립, 검토와 승인 단계로 이뤄진다.

3.2 프로젝트 Level 테스트 계획수립

개발 중인 제품이 요구사항에 일치하는지(Conformance to requirements)와 사용의 적합성(Fitness for use)을 개발 초기 품질 수준의 계획아래 테스트 계획과 실행도 함께 시작된다[4]. 본 단계는 프로젝트 자체의 범위, 계획에 대한 인증이 그 목적이다. 특정한 프로그램의 컴포넌트 또는 개발 하에 있는 구성요소의 요구사항에 적합성과 타당성 검증에 초점이 맞추어진다

3.3 인프라스트럭처 Level 테스트 계획수립

이 단계에선 폭 넓은 요구사항을 만족하는지 검증하고 인증하는 것이 목적이다. 테스트를 진행하면서 개발중인 컴포넌트는 인프라 구조 내 컴포넌트들과 함께 Interface를 테스트 하게 된다. 테스트 종류에는 패키지 통합 테스트, 산출물 인증 테스트, 시스템 통합 테스트, 인프라 구조 통합 테스트가 있다.

3.4 프로그램 Level 테스트 계획수립

본 단계에선 마지막 제품의 마지막 검증과 인증을 목표로 실시된다. 요구사항에 적절한 인증을 목표로 실시된다. 요구사항에 적절한 테스트 데이터, 테스트 케이스들이 개발되며, 강화된 통합 테스트와 산출물 테스트들이 수행된다. 이때 실시되는 테스트의 종류로는 인수 테스트,

데이터 무결성 테스트, 설치테스트, 상호운용성 테스트, 보존성 테스트, 보안 테스트, 저장 테스트, 유용성 테스트들이 있다.

3.5 이행 활동 지원

위 단계의 테스트가 끝난 후 프로그램 실행 상태를 일정 기간동안 모니터링 한다. 이때 결점이 발생하면 수정하고 모니터링 보고서를 작성한다.

3.6 테스트 활동의 관리 및 측정

테스트 프로세스의 마지막 과정이 아니라 전체적인 과정을 관리하는 단계로서, 계획되고 실행되고 끝날 때 까지 테스트 계획이 품질계획의 일부로서 적절히 수행되는지를, 평가 Metrics를 이용하여 관리하는 단계이다. 전략 관리, 테스트 계획 활동 관리, 테스트 실행 관리, 테스트 실행 이후 활동 관리로 구성된다. 구현이후 결함은 결함관리와 추적관리에 의해 해결된다.본 단계에선 소프트웨어의 품질(Quality)과 수준(Grade)에 대한 명확한 구분이 있어야 한다. 이상이 없으면 테스트는 종료된다.

4. 제안된 모델의 현 IT 업계 적용

최근 소프트웨어의 수준(Grade)은 과거와 달리 무척 많은 기능을 요구함에 따라 급격히 향상되어 오고 있다. 소프트웨어의 수준이 높아짐에 따른 부작용은 품질이 이러한 변화에 대해 효과적으로 관리되지 않는 데서 일어난다. 물론 높은 품질을 보장하기 위해서는 많은 비용과 인력의 노력이 필요한 것은 사실이다. 이러한 분위기를 적절하고 합리적인 관리를 통해 보다 전체 개발일정을 통한 체계적인 소프트웨어 테스트 방법론을 통해 관리함으로써 수준향상에 따른 품질의 저하를 방지할 수 있게 될 것이다. 또한 개발이후 관리비용의 경우, 일반적으로 따르면 개발비용의 60%가 투입된다는 점에서 개발 시, 품질에 대한 강조는 간과될 수 없는 것이다.

5. 결론 및 향후 과제

소프트웨어 품질에 결정적인 역할을 담당하는 테스트는 개발에 없어서는 안될 중요한 과정이다. 테스트는 제품개발과 동시에 병행 시작해야 하고, 테스트의 수준 및 효과는 테스트 프로세스에 의해 많은 영향을 받는데, 이에 본 논문에서 효과적인 테스트 프로세스를 제안하였다. 앞으로 모든 소프트웨어 개발에 있어 체계적인 테스트 프로세스가 프로젝트의 계획단계의 중요한 영역으로 자리잡아 높은 품질의 제품을 생산하여 국제적 경쟁력을 향상시켜야 하겠다.

SI 업계는 비용과 일정에 대한 경쟁력을 위해 품질에 대해 간과함으로써 결국 소프트웨어에 또 다른 위협을 야기시키고, 이러한 위협관리는 엄청난 예비비의 지출과

더불어 일정을 연기시키는 결과를 낳게 된다. SI의 경쟁력강화를 위해서는 반드시, 계획단계에서 시작하여 개발종료 순간까지, 품질계획과 더불어 테스트에 대한 체계적인 계획수립과 이행이 필수이다.

향후 과제로는 본 논문에서 제안한 테스트 프로세스의 상세 프로세스 설계를 통해 각각의 단계별 서브 프로세스와 그 프로세스의 각 입력과 출력 그리고, 활동의 디자인을 완성할 것이다. 이와 더불어 실제 소프트웨어 개발에 적용하여, 이에 발생한 문제점을 보완해야 더욱 완벽한 테스트 프로세스를 완성시켜야 할 것이다.

[참고 문헌]

[1] IEEE Std 829-1983, Standard for Software Test Documentation

[2] 이 주현, 실용소프트웨어 공학론, 법영사 200년

[3] <http://www.sds.samsung.co.kr/support/epartner/ecpaper/data6/2-3.pdf>

[4] PMI, A Guide to the Project Management Body of Knowledge 2000 Edition