IT분야에 있어서 소프트웨어 테스팅 및 품질보증의 역할

신성우*°, 이남용**

*바산네트워크(주), **숭실대학교 컴퓨터학부

The Role of Software Testing and Quality Assurance in the IT Industry

Shin, Sung Woo°, Lee, Nam Yong

A division of Test Consulting, BashanNetwork Co. Ltd.

Dept. of Computer Science, Soonsil University

E-mail : swshin@bashannet.com, nylee@computing.ssu.ac.kr

요 약

오늘날 우리가 개발하는 소프트웨어 시스템은 사람의 실수로 인하여 문제가 생기는 경우가 너무나도 많다. 소프트웨어 테스팅은 품질보증에서 가장 중요한 부분이며, 특히 소프트웨어의 명세, 디자인, 코드 구현 부분에 대한 객관적인 평가를 가능하게 한다. 국내에서는 아직까지는 소프트웨어 테스팅 및 품질보증 활용의 경우가 미흡한 실정이지만 외국의 경우 개발비용의 40% 이상이 테스팅에 소요되는 경우도 있다. 본 고에서는 해외 IT 기업의 테스팅 수행 현황을 조사함으로써 IT 분야에 있어서 소프트웨어 테스팅 및 품질보증의 역할이 무엇인가를 살펴보도록 하겠다.

1. 서론

오늘날 우리는 컴퓨터의 역할이 더욱더 증대하고 있는 세상에 살고 있다. 컴퓨터를 구성하는 부분에서 가장 중요한 요소 중 하나는 소프트웨어이다. 다양한 산업이 발달함에 따라 소프트웨어의 역할은 더욱더 중요해지고 있다. 특히 최근에는 인터넷 세상이라 불리울 정도로 사회 문화 전반에 걸친 다양한 활동에 소프트웨어를 이용하고 있다. 산업계, 금융계, 학술적 연구 등 어떠한 분야간에 적절한 소프트웨어 없이는 효율적인 업무 수행을 기대하기 힘든 것이 현실이다. 모든 사람이 더욱 컴퓨터와 소프트웨어에 의존하여 작업을 하고 있다. 하지만 지금까지 국내 업체들은 여러 가지 문제 때문에 소프트웨어를 품질 높게 작성하는 것보다는 완성하는 것에 급급해왔다. 이러한 개발방식 덕택에 국내 소프트웨어 업계가 양적으로는 크게 성장하였으나 질적 성장은 아직도 세계의 유명 업체에 비하여 뒤처지고 있는 현실이다. 소프트웨어 품질에 영향을 주는 요소로는 시간, 비용, 제품품질이 있다.[1] 이 중 제품품질요소를 향상시키는데 결정적인 역할을 수행하는 것이 소프트웨어 테스팅이다.

소프트웨어 테스팅은 품질보증에서 가장 중요한 부분이며, 특히 소프트웨어의 명세, 디자인, 코드 구현 부분에 대한 객관적인 평가를 가능하게 한다. 국내에서는 아직까지는 소프트웨어 테스팅 및 품질보증 활용의 경우가 미흡한 실정이지만 외국의 경우 개발비용의 40% 이상이 테스팅에 소요되는 경우도 있다. 본 고에서는 해외 IT 기업의 테스팅 수행 현황을 조사함으로써 IT 분야에 있어서 소프트웨어 테스팅 및 품질보증의 역할이 무엇인가를 살펴보도록 하겠다.

2. 소프트웨어 테스팅과 제품의 품질

미국의 소프트웨어 업체인 STI가 2001년도에 수행한 조사결과에 따르면 소프트웨어 테스팅 산업분야는 다음과 같은 특징을 가지고 있다.

- 소프트웨어 산업분야 기업은 다른 산업분야에 비하여 테스트 전문가 확보(43%) 비율이 높다.
- 금융(13%), 서비스(9%), 시스템(8%) 등 일반적인 업계에서도 소프트웨어 테스팅 전문가를 확보하고 있다.
- 소프트웨어 테스팅은 개발팀의 한 부속 팀으로 진행되는 경우와(44%) 독립적인 테스팅 전문팀을 운영(39%)하는 경우로 구분된다.
- 지난 2년동안 소프트웨어 테스팅에 대한 책임의식이 바뀌었다. 과거에는 전문적인 테스팅 담당이 주 책임(58%)을 가지고 있다고 인식하였으나 최근에는 전문 테스팅 담당과 개발팀 공동의 책임(39%)이라는 형태로 바뀌고 있다.
- 향후 2년간 전망을 조사한 결과, 오직 전체 응답자 중 4%만이 테스팅 직원 및 부서를 고려하지 않다고 답했다.
- 대부분의 기업에서 소프트웨어 테스팅 부서는 품질보증부서(58%), 소프트웨어 테스팅 부서(21%), 소프트웨어 엔지니어링 부서(10%) 기타 등으로 불리우고 있다.
- 소프트웨어 테스팅의 결과 보고는 IT, MIS 담당 임원(29%), 개발팀장(27%), 제품 관리자(18%), 순으로 활용되고 있다.

소프트웨어 테스팅은 소프트웨어에 악영향을 미치거나 시스템에 문제를 발생시킬 수 있는 여러 가지 잠재적 위험을 인식하고 평가함으로써 소프트웨어의 안정성 및 높은 품질을 제공한다. 고객 만족은 궁극적으로 사업의 성공에 영향을 준다. 다시 말해서 고객이 소프트웨어를 구매한 후 문제점을 발견하여 수정을 요구하는 상황이 발생하면 안됨을 말한다. 고품질의 소프트웨어는 단순하게 가장 우수한 프로그래머가 버그가 없음을 확신하고 제품을 만든다고 이루어지는 것은 아니다. 이렇게 잘 만들어진 소프트웨어라 할지라도 항상 몇몇 에러를 가지고 있다. 평균적으로 경험많은 개발자의 경우 100라인당 6개정도의 에러를 발생시킨다. 만약 코드가 구조적이지 않고 문서화가 잘 진행되지 않은 경우에는 100라인당 12~20의 에러가 평균적으로 발생한다. 여기서 말하고자 하는 바는 모든 소프트웨어는 항상 버그 없이는 개발할 수 없다는 것을 강조하는 것이 아니다. 오히려 모든 제품 개발과정 중에 소프트웨어 테스팅을 수행할 필요가 있음을 강조하는 것이다. 소프트웨어 개발 중 테스트 단계를 거치게되면 상당히 광범위한 분야에 결점을 제거할 수있게 되며 그 결과 고객 만족을 높이는데 결정적이 기여를 할 수 있게 된다. 소프트웨어 제품의 품질이라는 것은 소프트웨어의 결함률로 생각할 수 있다. 다음은 소프트웨어 제품에 영향을 주는 품질 요소[2]이다.

- 정확성 : 프로그램이 스펙과 일치하는점 및 고객의 목적에 어느정도 부합하는 가를 확인하는 특성
- 신뢰성 : 의도하는 기능이 필요로 하는 정확성에 부합되게 수행되는가를 확인하는 특성
- 효율성 : 요구되는 작업을 수행하기 위해 필요한 컴퓨팅 자원 소모율 및 개발 코드의 양
- 무결성 : 인가되지 않은 사람이 데이터에 접근 할 수 있는 여부를 확인하는 특성
- 사용성 : 프로그램 활용을 위해 필요한 학습 및 사용준비 노력의 정도를 확인하는 특성
- 유지보수성 : 유지보수에 필요한 노력의 정도를 확인하는 특성
- 테스트성 : 의도대로 적절하게 프로그램이 작동하는 가를 테스트 할 수 있는 정도를 확인하는 특성
- 이식성 : 다양한 하드웨어 및 운영체제에 이식할 수 있는 정도를 나타내는 특성
- 재사용성 : 소프트웨어가 다시 활용될 수 있는 정도를 나타내는 특성

현재 판매되고 있는 제품이 아무리 시장지배력이 강하다고 할지라도, 제품의 품질이 나쁘거나 위에 나열한 품질 기준을 일정 수준이상 준수하지 못할 경우 고객과의 거리가 멀어질 수 밖에 없고 결국 시장으로부터 퇴출 당하게 될 것이다. 제품의 품질은 경쟁력에 직접적인 영향을 미친다. 소프트웨어 테스팅은 개발단계 동안 다양한 형태로 진행되기 때문에, 대부분의 개발자들은 소프트웨어 테스팅을 수행할 시간이 거의 없다고 생각한다. 또한 소프트웨어 테스팅이란 엄청난 시간낭비이며 자신들을 공격하는 행위라고 간주한다. 그렇기 때문에 소프트웨어 테스팅을 개발자에게 이해시키는 점도

매우 중요한 요소이다. 지금처럼 점점 국제적 경쟁이 더욱 심해지는 시대에서 소프트웨어 품질 향상은 고객 만족을 이끌어낼 수 있는 원동력이 된다. 낮은 품질의 소프트웨어만 개발한다면 고객이 외면할 것이고 결국 사업 실패를 낳게 된다.

3. 소프트웨어 테스팅 절차

미국 IT업계의 경우 사내에 테스팅 전문 인력 또는 부서를 형성하고 있는 경우가 43%이상며 일부는 외부 전문 기관에 테스팅을 의뢰하는 경우도 많다. 하지만 국내의 경우 아직까지 IT분야는 테스팅 및 품질보증 부서의 역할이 그렇게 충분하게 성숙하지 않은 단계이다. 2000년 전후로 국내에서는 소프트웨어 제품의 품질에 대한 논의가 활발히 이루어졌으며, 최근에는 소프트웨어 개발 조직의 능력 성숙도에 대한 관심이 부쩍 높아진 현실이다. CMM이나 SPICE 같은 경우 전체 개발과정에서 발생하는 여러 행위들에 대해 상위 개념 수준으로 정의를 하고 있기 때문에 이 활동을 통해 직접적인 소프트웨어 품질 향상이 되는 것은아니다. 반면에 테스팅의 경우 전체적인 프로세스 측면은 CMM이나 SPICE보다 다소 약하지만 세부적인 수행 행위를 중심으로 하기 때문에 보다 직접적인 품질보증 활동이라 할 수 있다.

지금까지 국내 IT업계의 관리자들은 소프트웨어 테스팅에 대한 마인드가 부족한 현실이다. 아직까지도 상당수의 관리자들은 개발과정중 소스코드와 관계된 것이라면 모든 것이다 개발자의 책임이라고 인식하고 있다. 하지만 개발자의 관점에서는 모든 문제점을 인식하고 수정하기가 힘들다. 설령 테스트의 필요성을 인식하고 있는 관리자라 할지라도, 테스트를 제품이 완성되가는 시점에서 수행하는 베타 테스트나 인수 테스트 정도로 인식하고 있는 경우가 많다. 하지만 테스트는 개발 마지막 단계에서만 수행되는 것이 아니고 개발 전과정에 걸처 진행된다는 것을 이해해야만 한다. 테스터에 대한 마인드 확립도 중요하다. 과거에 테스터는 제품의 코드 등 내부정보와는 관계없이 수행하는 블랙박스 테스팅이 중심이었다. 그렇기 때문에 이들은 개발 언어에 대한 이해가 부족해도 괜찮았으며 필요시 사용할 스크립트 언어 1~2개 정도만 사용할 수 있으면 되었었다. 하지만 IT 산업계의 변화는 테스터에게 더욱 많은 것을 요구하게 되었다. 예컨대, 인터넷 관련 솔루션을 개발할 경우 테스터는 개발자 수준으로 인터넷에 활용되는 각종 기술에 대한 충분한 이해를 하고 있어야만 한다. 여기에는 Java, ASP, COM등 최근에 각광받고 활용되는 모든 기술이 포함된다.

특히 인터넷 기반의 솔루션일 경우 과거에 비하여 시장의 주기가 매우 짧아졌다. 그렇기 때문에 가능한 빨리 개발을 끝내야지만 보다 원활하게 시장의 요구를 따를 수가 있다. 제품을 빠르게 개발하여 출시하기 위해서는 품질보증 활동이 매우 효과적으로 수행되어야만 한다. 특히 품질보증의 여러 행위 중 테스팅의 역할이 더욱 중요시 된다. 소프트웨어 테스팅은 크게 6가지 단계로 진행된다 [2]. 6가지 단계는 테스트 전략 수립 단계, 테스트 계획 단계, 테스트 디자인 단계, 테스트 구현 단계, 테스트 수행 단계, 테스트 평가 단계로 구분 할 수 있다. 각각의 테스트 단계는 소프트웨어 개발 단계와 연관을 갖는다. 예컨대 고객으로부터 소프트웨어 요구사항을 전달받은 경우, 개발팀, 테스팅 팀의 관련자가 모여 요구사항에 대한 검토를 수행할 수 있다. 이 단계를 거치게 되면 고객이 필요로 하는 것을 초기에 보다 정확하게 파악함으로써 고객의 만족도를 높여 프로젝트 성공 가능성을 높이게 된다.

테스트 전략 단계에서는 테스트 수행 시 필요한 각종 행위, 평가 기준, 전략 등을 간략하게 정의한다. 다시말하자면 테스트를 수행하기 위해 필요한 아이디어를 정리한 것이다. 이 단계에서 테스트 가능성에 대한 검사를 병행하게 되는데, 세부적인 항목은 다음과 같다.

테스트 대상에 대한 요구사항 학습 및 인식, 테스트의 목표 및 목적의 구체화, 예상 발생 위험요소 파악. 테스트 계획 단계에서는 전략단계에서 파악한 정보를 바탕으로 보다 구체적으로 테스트 요구사항에 대한 분석 작업을 수행한다. 여기에는 테스트 계획 작성, 자동화 테스트 필요 부분 파악, 테스트 자원 할당 등이 해당된다. 테스트 디자인 단계는 실질적으로 테스트가 어떻게 진행되는 가를 세부적으로 나타낸다. 원칙적으로는 테스트 디자인 단계에서 나오는 문서만 가지고 수동 테스트를 끝낼 수 있다. 이 단계에서는 테스트 케이스 및 프로

시듀어를 작성하는 것이 가장 큰 목적이다. 테스트 구현단계에서는 테스트 진행시 필요한 여러 가지 디라이버나 스텁 등을 작성하는 것이 가장 큰 목적이다. 특히 이 단계는 컴퓨터가 이해할 수 있는 여러 스크립트 등을 작성하게 된다. 테스트 수행단계는 지금까지 준비한 계획 및 절차에 따라 실제로 테스팅을 수행하는 단계이다. 마지막 단계인 테스트 평가 단계는 수행단계의 결과를 바탕으로 적절한 형태의 보고서를 작성하는 단계이다.

4. IT업계의 도전 사항

최근 소프트웨어는 과거에 비해 더욱 복잡하며, 짧은 기간내에 완성해야 한다. 특히 소프트웨어 개발 실패시 오는 피해 비용은 감당하기 힘든 수준이 되었다. 그렇기 때문에 과거 어느때 보다도 효율적이고 효과적인 소프트웨어 품질관리 기술이 필요하다. 특히 깊이 있게 검증된 각종 실무지침이나 유용한 툴의 활용은 성공적인 소프트웨어 품질관리에 있어서 절대적인 요소이다.

테스팅이란 소프트웨어 품질을 측정하는 수단이다. 이 것은 특정한 하나의 행위도 아니고 개인적으로 수행하는 테스트도 아니다. 경험 많은 관리자라면 테스팅이 각종 결점을 찾아내고 제거하고, 제품의 품질을 개발 초기부터 측정하여 지속적인 개발과정의 품질향상을 유도하는 역할을 수행하는 등 다양한 분야를 포함하고 있다는 것을 알고 있다. 위험을 줄이는 가장 효과적인 방법은 개발 초기부터 지속적이고 반복적으로 테스팅을 수행하는 것이다. 이러한 방식을 택하게 되면, 각종 결점은 새로운 기능이 구현될때마다 바로 제거된다. 그렇기 때문에 개발이 끝나가는 시점에서 개발된 어플리케이션을 보다 빠르고 정확하게 테스팅을 하게 되고 궁극적으로는 제품을 조기에 출시하는대 도움을 주게 된다. 특히 제품 개발과정에서 지속적으로 각종 기능을 테스트하기 때문에 제품에 대한 충분한 이해 및 확신을 할 수 있게 되며, 이 것은 각종 비즈니스 홍보에 보다 효과적으로 대응할 수 있다. 물론 테스팅을 개발기간동안 지속적으로 수행하기 위해서는 테서트 및 개발자 모두에게 보다 많은 노력이 필요한 것은 사실이다. 이런 분위기를 조성하기 위해서 관리자는 테스터와 개발자의 화합을 유도할 책임이 있으며 적절한 보상체계도 도입을 해야한다. 최근에는 반복적 개발방법론이 도입되고 있다. 반복적 개발방법론의 경우 각 단계가 끝날 때 마다 종료 조건을 만족해야만 다음단계로 넘어갈 수 있다. 이러한 경우 테스팅을 종료조건을 확인하는 방법으로 사용할 수 있다. 일단 어떠한 방법론을 선택하고 개발 팀에 맞게 수정하여 확립했다면, 그 것을 수행할 각종 자원을 적절하게 할당하는 일이 매우 중요하다. 품질 엔지니어링도 다른 소프트웨어 엔지니어링과 매우 유사한 특징을 가진다. 만약 어떤 개발 프로젝트 수행시 품질에 대한 격차(Quality gap)가 너무 크게 발생한다면 그 조직이 아직 적절하게 개발 프로세스를 확보하지 못한 상태를 의미한다. 이러한 품질 격차에 대한 분석은 SEI-CMM, CMMI, 6 시그마, ISO 등에서 밝히고 있는 다양한 기법을 적용하여 활용할 수 있다.

위에서 언급한 어떤 모델을 적용하여 소프트웨어 품질 관리를 정의하고 구현하여 활용하던 관계없이 기본적인 시작점은 비슷하다. 소프트웨어 품질관리는 소프트웨어 엔지니어링 중 하나이다. 일반적인 엔지니어링이 항상 적절한 방법, 도구, 평가 기준을 가지고 있듯이 소프트웨어 품질관리 모델도 이러한 특징을 모두 포함하고 있다. 그렇기 때문에 이러한 모델을 적용하기 위해서는 모델에 대한 세부적인 학습 및 분석이 효과적인 적용을 위한 가장 기본이라 할 수 있다. 국내 SI 업체가 이러한 품질관리 기법을 조기에 도입하여 활용하면 보다 높은 품질의 소프트웨어 생산이 가능해지므로 고객도 지속적으로 제품을 이용하게 될 것이다.

5. 결론

소프트웨어 테스팅은 개발할 제품의 요구사항을 도출하는 단계서부터 적용해야만 한다. 소프트웨어 테스팅은 제품의 완성단계에만 적용하는 것이 아니며, 제품 개발과 병행하여 수행해야만 한다. 또한 테스팅은 단순한 디버깅 작업만을 의미하는 것이 아니며 소프트웨어 품질을 향상시키기 위해 사용하는 여러 기법 및 행위를 모두 포함한다. 이러한 소프트웨어 테스팅의 수준 및 효과는 사용하는 테스팅 절차에따라 많은 영향을 받는다. 그렇기 때문에 성공적인 소프트웨어 테스팅 수행을 위해서

는 적절한 소프트웨어 공학적인 실무를 충분히 이해하고 있어야 한다. 또한 테스팅을 담당하는 팀은 조직의 개발 성숙도(CMM, SPICE)에 관계없이 지속적으로 테스트 절차를 향상시키는 노력을 기울여야만 한다. 개발초기 단계부터 각종 결점을 줄이는 적절한 예방활동을 지속적으로 적용하게되면, 개발 중후기에 발생하는 여러 위험관리 부담을 줄일 수 있게 되며, 개발 후기 단계에서 발생하는 많은 에러 및 비용 문제를 줄일 수가 있다. 그리고 독립적인 테스팅 수행 팀을 구성하는 것에 대한 중요성을 IT 산업계의 관리자가 꼭 이해를 해야만 한다. 뿐만아니라 테스팅 담당 직원에게 적절한 교육, 권한, 책임을 부여하는 것도 매우 중요하다. 서론에서 언급했듯이 높은 품질의 소프트웨어를 만들기 위해서는 개발자 만으로는 부족한 시대이다. 그렇기 때문에 개발자와 테스터가 모두 필요한 존재임을 인식해야 하며, 테스터에게도 제품의 품질 향상에 대한 적절한 보상체계를 갖추어서 좀더 능동적이고 생산있게 업무를 수행할 수 있도록 해야만 한다. 이제는 단순하게 제품만 만들어 납품하던 시대는 지났다. 무한 경쟁속에 제품의 품질이 매우 중요시되는 시대이다. 그렇기 때문에 SI 업계가 보다 경쟁력 있는 회사가 되기 위해서는 위에서 언급한 여러 요소들을 최대한 빠른 시일내에 반영하고 지속적인 품질향상 노력을 기울여야만 할 것이다.

[참고문헌]

[1] Software Quality Assurance - A practical approach: Earnest Wallmuller Prentice Hall, 1994

[2] ISO 9126 - Software Quality Model

[3] Ratioanl Approach for Software Testing, Rational Coporation 2001