

소프트웨어 시험단계의 테스트를 위한 시험 측정 항목의 개발

○
이하용*, 황석형** 양해솔***

*한국소프트웨어품질연구소

**선문대학교 자연과학대학 정보과학부

***호서대학교 벤처전문대학원

e-mail:insq@unitel.co.kr, shwang@omega.sunmoon.ac.kr
hsyang@office.hoseo.ac.kr

Development of Test Measurement Items for Test of Software Testing Phase

Ha-Yong Lee*, Suk-Hyung Hwang** Hae-Sool Yang**

*Institute of Software Quality(INSQ)

**Sunmoon Univ. Div of Information & Computer Science

***Graduate School of Venture, Hoseo University

요 약

소프트웨어의 개발 과정에서 생명주기 단계별로 시험 측정 항목을 개발하여 적용함으로써 각 단계별로 고품질의 개발 산출물을 생산할 수 있으며 결과적으로 고품질의 소프트웨어 개발에 기여하게 된다. 이러한 과정을 거쳐 왔다면 소프트웨어의 오류는 최소화되었겠지만 시험 단계를 통해 잔존하는 오류를 발견하여 개선할 필요가 있다. 본 연구에서는 생명주기 단계 중 소프트웨어 시험 단계의 테스트를 위한 시험 측정 항목을 개발하여 시험 단계에서 최종적으로 소프트웨어의 품질을 향상시킬 수 있는 여건을 마련하였다.

1. 서 론

소프트웨어의 개발 과정에서 생명주기 단계별로 시험 측정 항목을 개발하여 적용함으로써 각 단계별로 고품질의 개발 산출물을 생산할 수 있으며 결과적으로 고품질의 소프트웨어 개발에 기여하게 된다. 이러한 과정을 거쳐 왔다면 소프트웨어의 오류는 최소화 되었겠지만 시험 단계를 통해 잔존하는 오류를 발견하여 개선할 필요가 있다. 본 연구에서는 생명주기 단계 중 소프트웨어 시험 단계의 테스트를 위한 시험 측정 항목을 개발하여 시험 단계에서 최종적으로 소프트웨어의 품질을 향상시킬 수 있는 여건을 마련하였다.

2. 관련 연구 현황

2.1 국외의 동향

국외 소프트웨어 선진국에서는 품질평가를 위한 표준 제정을 지속적으로 진행해 나가고 있다. 제품 품질평가 특성에 관한 표준으로서 ISO/IEC 9126과

품질평가 프로세스에 관한 표준으로서 ISO/IEC 14598이 표준화를 진행 중에 있다.

2.2 국내의 동향

현재 품질평가 및 시험 기술에 관한 국내의 동향은 전반적으로 기반이 취약하다고 할 수 있다. 품질평가를 위한 관련 규격이 제정되어 있지 않으며 소프트웨어 품질시스템 인증을 외국에 의존하는 등 기반 연구가 매우 취약한 실정이다. 국내 소프트웨어 산업체에서는 품질 개선 기술 및 제품 평가 기술 개발을 시급한 과제로 생각하고 있으나 자체적으로 기술 개발을 추진하기에는 많은 어려움이 따르고 있다.

3. 소프트웨어의 결함

소프트웨어의 결함이란 원하는 제품의 속성과 불일치를 보이는 것이다. 결함에는 제품 명세의 결함과 고객이나 사용자의 기대와 불일치하는 경우와 같

이 두 가지 종류가 있다.

3.1 제품 명세의 결함(defect)

제품 명세의 결함이란 만들어진 제품이 제품에 대한 명세와 다른 경우를 말한다. 즉, 제품을 만드는 알고리즘이 명세와 다르다면 결함이 있는 것으로 생각할 수 있다.

3.2 고객이나 사용자의 기대와 불일치

이러한 불일치는 만들어진 제품에 사용자가 원하는 것이 없거나 만들어진 제품에 포함될 수 있도록 명세되지 않았다는 것이다. 이러한 잘못은 명세서를 작성하는 과정에서 또는 요구 과정에서 발생하거나 구현된 요구가 만족스럽지 못한 경우에 발생한다.

결함은 일반적으로 다음과 같은 세 가지 부류 중 하나의 형태로 나타난다.

① 잘못(wrong)

명세가 부정확하게 구현된 경우에 해당한다. 이 결함은 고객이나 사용자의 명세와 불일치하는 것이다.

② 누락(missing)

명세되거나 요구된 것이 만들어진 제품에 구현되어 있지 않은 경우에 해당한다. 이것은 명세대로 구현되지 않은 경우이거나 고객의 요구사항이 제품이 만들어지는 동안 또는 만들어진 이후에 확인된 경우이다.

③ 여분(extra)

명세되지 않았던 요구가 제품에 포함된 경우이다. 이것은 항상 명세와 불일치하지만 제품의 사용자가 원하는 속성일 수도 있다. 그러나 이것은 결함으로 간주된다.

3.3 결함과 실패

결함은 소프트웨어 시스템에 포함된다. 결함은 잘못(wrong), 누락(missing), 여분(extra)으로 분류된다. 결함은 소프트웨어 자체에서 발견되거나 매뉴얼과 문서를 지원할 때 발견될 수 있다. 결함이 소프트웨어 시스템에 포함된 문제점이긴 해도, 그것이 사용자, 고객, 운영 시스템에 영향을 미치지 전까지는 충격이 되지 않는다. 동작 중에 발생하는 오류나 사용자, 구매자에게 부정적인 영향을 미치는 결함을 실패(failure)라 한다. 결함이 갖는 주된 관심사는 결함이 실패로 바뀔 수 있다는 것이다. 조직에 손해를 끼치는 것은 실패(failure)이다.

어떤 결함은 전혀 실패로 나타나지 않을 수도 있다. 다시 말해서, 하나의 결함이 많은 실패를 야기할 수도 있다는 것이다.

3.4 공정과 결함 비율

대부분의 결함은 작업을 적절히 수행하지 않은 공정에서 발생된다. 만일 요구 공정에 결함이 있다면 적절한 정보를 모을 수 없을 것이다. W. Edwards Deming은 적어도 90%의 결함은 공정 문제에 의해 발생한다고 했으며 많은 연구들이 증거를 제공했다. 테스트하는 사람은 이점을 중시해야 한다.

4. 테스트의 효과

시스템 개발 생명주기 동안에 어플리케이션 시스템을 시연한 IBM의 연구에서 60개의 오류가 나왔다. 이 연구에서 코딩하기 전에 테스트하는 것이 에러를 발견하는데 있어 50% 더 효과적이고 코드 작성 후에는 80% 효과적이라는 것을 보여 주었다.

다른 연구에서는 코드 작성 후에 오류를 고치는 것이 비용 면에서 코드 작성 전보다 적어도 10배 이상 차이가 났음을 보여 주었다. 케미컬 은행의 사례에서는 에서는 오류의 2/3가 코드 작성 이전에 발생한다는 것을 보여 주었다.

(그림 1)은 1000라인을 가진 시스템으로서 일반 생명주기에 의한 결과는 왼쪽에 나타내었고 테스트는 코드 작성 후에 실시한다. 일반 생명주기에서는 60개의 오류가 코딩 후에 남아 있으며 테스트를 거쳐 48개의 오류를 검출하여 480 단위의 비용을 발생 시켜서 총 테스트 비용은 1,680 단위가 되었다. 생명주기 테스트를 거친 경우는 일반 생명주기 테스트 비용의 1/3 정도만 소요되었다.

일반 생명주기		비율	생명주기 테스트	
누적 테스트 비용	누적 에러/1000 줄 코드		누적 에러/1000 줄 코드	누적 테스트 비용
0	20	요구단계 20개 오류 ↓ 비용 검출 = 1	10	10
0	40	설계단계 20개 오류 ↓ 비용 검출 = 1	15	25
0	60	프로그래밍단계 20개 오류 ↓ 비용 검출 = 1	18	42
480	12	시작 단계 80% 오류 감소 ↓ 비용 검출 = 10	4	182
1680	0	제품 "0" 결함 ↓ 비용 검출 = 100	0	582

(그림 1) 생명주기 테스트의 경제적 의미

5. 시험단계의 시험 측정 항목 개발

테스트는 시스템의 실행 프로그램을 평가하기 위한 기회를 제공한다. 시험 단계에서 효과적인 테스트를 수행하는 것은 실행 프로그램이 명세된대로 기능한다는 높은 수준의 보증을 제공한다.

그러나 실행 프로그램이 적절하게 동작한다는 것을 완전하게 보증한다고 하기에는 미흡한 점이 있다. 이러한 이유 때문에 완전한 시스템을 테스트하는 것이 중요하다. 테스트 트레이드 오프는 생명주기의 다양한 단계 사이에서 일어난다. 요구, 설계, 프로그래밍 단계에서 더 많은 테스트를 수행하면 할수록 시험 단계에서 테스트를 수행할 필요성이 적어진다.

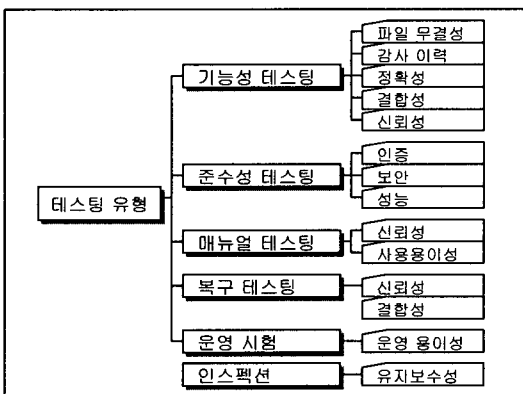
시험 단계 동안 산출되는 항목들은 다음과 같이 정리할 수 있다.

- 시험단계 테스트 계획
- 테스트 데이터
- 이전 단계 테스트 결과
- 컴퓨터 운영자와 같은 제3자의 테스트
- 형식을 갖춘 테스트 결과 문서

5.1 테스트의 유형

어플리케이션 시스템을 테스트하는 방법은 여러 가지가 있다. 이러한 모든 테스트 형태는 조적이 새로운 어플리케이션 시스템을 설치할 때에 최고의 성공 가능성을 갖도록 하기 위해 필요하다.

시험 단계에서 수행할 테스트 유형에 대해 (그림 2)와 같은 체계를 구축하였다.



(그림 2) 테스트 유형의 체계

5.2 시험 측정 항목 사례

다음에 개발된 시험 측정 항목 중의 일부를 기술하였다.

1) 매뉴얼, 복구, 기능 테스트(신뢰성)

매뉴얼 테스트는 자동화 시스템과 상호 작용하는 사람들이 그들의 기능을 올바르게 수행할 수 있다는 것을 보증한다.

복구 테스트는 설치 중인 어플리케이션이 이미 설치된 어플리케이션 또는 새로운 어플리케이션에 의해 인터페이스되는 어플리케이션의 어떤 부분에도 영향을 미치지 않는다는 것을 확인하는 것이다.

기능 테스트는 환경이 다양하고 트랜잭션이 반복될 때, 시스템 요구사항이 정확하게 수행될 수 있다는 것을 확인한다.

<표 1>은 매뉴얼, 복구, 기능 테스트에 관한 시험을 위한 시험 측정 항목의 예를 나타내었다.

<표 1> 매뉴얼, 복구, 기능 테스트에 관한 시험 측정 항목

#	테스트기준	평가				제안된 테스트
		매우 적합	적합	부 적합	N/A	
1	각 데이터 요소 명세에 따르지 않는 데이터가 테스트되었는가?					데이터 확인 프로그램이 데이터 요소 명세를 따르지 않는 데이터를 reject하는가를 검증하라.
2	시스템 명세에 따르지 않는 것들을 reject 하기 위해 데이터 관계를 테스트하였는가?					시스템이 시스템 명세에 따르지 않는 데이터 관계를 reject 한다는 것을 검증하라.
3	무효 식별자가 테스트되었는가?					프로그램이 무효 식별자를 reject 하는 것을 검증하라.
4	테스트가 누락된 일련번호가 검출되는 것을 검증하는가?					시스템이 누락된 일련번호를 검출하는지 확인하라.
5	복구 프로세스로부터 얻어진 결과가 정확한가?					복구 프로세스로부터 얻어진 결과의 정확성을 검증하라.

2) 준수성 테스트(인증)

테스팅은 인증 규칙이 적절하게 구현되었다는 것을 확인해야 한다. 테스트 조건은 인증된 트랜잭션이 가결되었다는 것 뿐만 아니라 그들이 거부되었다는 것을 확인하기 위해 인증되지 않은 트랜잭션이나 프로세스를 포함해야 한다.

<표 2>는 인증에 관한 준수성 테스트를 위한 시험 측정 항목의 예를 나타내었다.

<표 2> 준수성 테스트(인증)

#	테스트기준	평가				제안된 테스트
		매우 적합	적합	부 적합	N/A	
1	수동 절차가 적절한 인증이 받아들여진다는 것을 보증하는가?					인증 절차가 수반된다는 것을 확인하기 위해 수동 절차를 테스트하라.
2	자동화된 인증 규칙이 테스트되었는가?					프로그램이 자동화된 인증 규칙을 시행한다는 것을 증명하라.
3	현 인증명과 식별자가 테스트의 부분으로 포함되었는가?					인증을 위한 실 식별자가 프로그램에 포함되어 있는지 확인하라.
4	다중 인증이 요구된다면 프로시저 기능이 적절하게 수행되는가?					다중 인증 프로시저가 적절하게 수행되는가를 확인하라.
5	관리하기 위해 인증 위반을 기록하는 프로시저가 테스트되었는가?					인증 기록이 적절하게 준비되고 인도되는가를 확인하라.

3) 기능 테스트(파일 응집성)

파일 응집성 제어는 파일 응집성 감소의 가능성을 최소화하기 위한 수단으로 구현되어야 한다. 그리고 파일 응집성 제어는 응집성의 감소를 막고 발견해야 한다.

<표 3>은 구현단계의 시험을 위한 시험 측정 항목의 예를 나타내었다.

<표 3> 기능 테스트에 관한 시험 측정 항목

#	테스트기준	평가				제안된 테스트
		매우 적합	적합	부 적합	N/A	
1	파일 균형 제어(balancing control)가 테스트되었는가?					파일 기능을 적절하게 균형잡기 위한 절차를 확인하라.
2	독립적으로 유지되는 제어가 테스트되었는가?					독립적으로 유지되는 제어가 자동화된 파일 제어를 확실하게 할 수 있는가를 확인하라.
3	업데이트가 적절하게 기록되었는가를 보증하기 위해서 무결성 절차가 테스트되었는가?					새로운 제어가 적절하게 업데이트된 트랜잭션에 영향을 주는가를 확인하라.
4	프로그램 고장 후에 무결성이 계속 유지될 수 있는가를 보증하기 위해서 테스트가 수행되는가?					고장난 프로그램이 파일 무결성에 영향을 미치는지를 결정하라.
5	비었거나 자료가 하나뿐인 레코드 파일 상태가 테스트되었는가?					각 파일에 대해 하나 또는 빈 레코드로 시스템을 가동하라.

6. 결론

소프트웨어에 대한 테스트는 개발 중인 시스템의 실행 프로그램을 평가할 수 있는 기회를 제공함으로써 실행 프로그램의 오류를 발견하여 문제점을 개선할 수 있도록 한다.

소프트웨어를 개발하는 과정에서 오류가 없는 완전 무결한 소프트웨어를 개발하는 것은 불가능하다. 다만, 오류를 최소화하기 위해 생명주기 단계마다 테스트를 실시함으로써 품질 향상을 도모하고 개발된 프로그램에 대해 시험단계에서 테스트를 실시함으로써 간존해 있는 오류를 발견하여 개선하게 된다.

본 연구에서는 고품질의 소프트웨어 제품을 개발하기 위한 방안으로 시험 단계에서 적용할 수 있는 시험 측정 항목을 개발하였다.

본 연구를 통해 개발된 시험 단계의 테스트를 위한 시험 측정 항목을 활용하여 효과적인 시험 공정이 이루어질 수 있을 것으로 기대한다.

향후, 테스트 항목을 추가하고 구체적인 테스트 방법에 대한 보완이 이루어질 수 있도록 지속적인 연구가 진행되어야 할 것이다.

참 고 문 헌

[1] Moller, K. H. and Paulish, D. J., "Software Metrics", Chapman & Hall(IEEE Press), 1993.
 [2] Wallmuller, E., "Software Quality Assurance A practical approach", Prentice Hall, 1994.
 [3] ISO/IEC 12119, "Information Technology - Software Package - Quality requirement and testing".
 [4] 吉澤. 東. 片山, "소프트웨어의 品質管理と生産技術", 日本規格協會, 1990. 5.
 [5] 水野幸男, "소프트웨어의 綜合的品質管理", 日科技連出版, 1993.
 [6] 양해술, 이하용, "설계단계에서의 품질평가 툴킷(ESCORT-D)의 설계 및 구현", 한국정보과학회 논문지(C), Vol. 3, No. 3, 1997. 6.
 [7] 양해술, "한진해운 신정보(영업 및 물류)시스템의 품질보증과 품질평가", 한진해운(주) 구현단계 확인평가, 1998. 9. 7.
 [8] 양해술, "소프트웨어 제품 평가 지원도구의 개발", ETRI 컴퓨터·소프트웨어 기술연구소 용역과제, 3차년도최종보고서, 1999. 10.