

# 오류항목 수정을 위한 선행조건 정의

이은서  
안동대학교 컴퓨터공학과  
e-mail: eslee@andong.ac.kr

## Definition of Precondition for the modify of error items

Eun-Ser Lee  
Computer Engineering, Andong National University

### 요 약

본 논문은 프로젝트와 소프트웨어 개발 시, 발생하는 오류를 관리하기 위하여 필요한 선행조건이 무엇인지를 연구하기 위함이다. 본 내용에서는 이를 정의하기 위하여 개괄적인 구조와 내용을 결정하여 오류가 발생하더라도 다른 곳으로 전파되는 영향을 최소화하기 위하여 이와 같은 연구를 진행하게 되었다. 오류에 의하여 단위 소프트웨어나 전체 시스템에 영향을 주어서 전체적인 품질을 떨어뜨리기 때문에 오류항목을 올바르게 수정하여 영향을 최소화해야 한다. 따라서 현재 연구에서는 개괄적인 구조와 내용을 기반으로 제시를 하고 자세한 사항의 연구는 향후 논문지를 통하여 심도있는 연구를 수행하고자 한다.

### 1. 서론

소프트웨어 시험은 체계적으로 계획되고 설명될 수 있는 과정이다. 시험 사례 설계가 수행될 수 있고, 전략이 정의될 수 있으며, 결과가 평가될 수 있다.

오류를 확인하고 검출하기 위해서는 시험을 사용하기도 한다. 소프트웨어 시험 전략은 소프트웨어의 성공적 구조의 결과인 잘 계획된 일련의 단계들에 소프트웨어 시험 사례 설계 방법을 통합하는 것이다. 그 전략은 이들 단계가 계획되고 그것이 착수되어 많은 노력과 시간, 자원이 요구될 때, 시험의 부분으로 수행하기 위한 단계를 설명하는 길을 제공한다. 그러므로 모든 시험 전략에 시험 계획, 시험 사례 설계, 시험 시행 그리고 결과 데이터 수집 및 평가를 반영해야 한다[3].

많은 방법에 있어 시험은 개별적인 과정이며, 시험의 형태는 개발 접근법이 다양하듯이 매우 다양하다. 수년간 프로그래밍 오류에 대한 우리의 유일한 방어는 신중한 설계와 프로그래머의 타고난 지적능력이다. 현재 최신식 설계 기술이 코드에 고유한 초기 에러들의 수를 감소시키기 위하여 도움을 주는 시대에 있다. 다른 시험 방법은 몇몇 명확한 접근법과 원리들로 이루어져 시작된다[1][4].

본 논문은 프로젝트와 소프트웨어 개발 시, 발생하는 오류를 관리하기 위하여 필요한 선행조건이 무엇인지를 연구하기 위함이다. 본 내용에서는 이를 정의하기 위하여 개괄적인 구조와 내용을 결정하여 오류가 발생하더라도 다른 곳으로 전파되는 영향을 최소화하기 위하여 이와 같은 연구를 진행하게 되었다.

### 2. 기반연구

#### 1) 오류수정

오류 수정은 성공적인 시험의 결과로서 나타난다. 시험 사례가 오류들을 발견할 때, 오류 수정은 오류 제거 결과의 과정인 것이다. 오류 수정이 규칙적인 과정이어야만 하고 그럴 수 있을지라도, 그것은 여전히 매우 많은 기술이다[1]. 시험의 결과를 평가한 소프트웨어 엔지니어는 소프트웨어 오류가 발생하기 전에 증후가 나타나기 위한 내부 원인이 명백히 존재한다. 이와 같은 원인에 증상을 연결시킨 불충분하게 이해된 지적 과정이 오류수정이다[2].

#### 2) 오류수정 접근법

오류 수정은 소프트웨어 오류의 원인을 찾아내고 수정하기 위한 중요한 목적을 가지고 있다. 그 목적은 계획적인 평가, 직과 그리고 운이 결합하여 실현된다.

일반적으로 오류 수정 접근법은 맹목적 강요, 역추적, 원인제거로 제시할 수 있다. 이와 같은 접근법을 활용하기 위하여 여러 가지 도구를 접목하고 있는 실정이다[5].

오류가 발견되면 그것은 수정되어야 한다. 그러나 우리가 이미 말했듯이 오류의 수정은 다른 오류들을 만들 수 있고, 그러므로 좋은 점 보다 해로운 점이 더 많을 수 있다.

### 3. 오류항목 정의를 위한 선행조건들

프로젝트나 소프트웨어 개발에서 오류는 많은 것을 내포하고 있다. 발생한 오류가 어떤 문제를 발생시키는지 정확히 예측하기 어렵다.

오류는 단순내용의 것도 있지만, 복잡한 해결 방안을 요구

하는 것이 대부분이다. 이와 같은 오류를 수정하는 것은 전체 시스템 및 구성요소인 소프트웨어의 완성도를 높여서 전체 품질을 향상 시킬 수 있게 된다.

본 절에서는 오류를 올바르게 수정하기 위하여 필요한 구조와 활동들이 무엇이며 선행되어야 할 사항은 무엇인지를 개념적으로 접근하고자 한다.

#### 1) 오류의 정의

오류의 올바르게 수정하기 위해서는 어떤 오류가 내포되어 있는지와 어떤 오류가 발생할 수 있는지 예측이 가능해야 한다.

발생되는 오류는 형태와 내용을 미리 알고 있지 않으면 이에 대한 대안을 세울 수 없기 때문에 큰 위험요소가 된다.

오류를 올바르게 정의하기 위해서는 전제 조건이 있다. 우선 현재 개발을 하고 있는 영역을 숙지하고 있어야 한다. 영역에 대한 지식이 부족하면 오류를 예측할 수 없게 되고, 예측을 하더라도 대안을 찾기 어렵게 된다. 따라서 오류를 정의하기 위하여 영역 전문가에게 도움을 얻어서 이를 기반으로 시스템 상에서 발생할 수 있는 오류를 정의해야 한다.

오류 정의에서 가장 큰 도움이 되는 것은 요구사항에 정의를 참조하는 것이다.

개발되는 모든 기능들은 영역지식을 기반으로 요구사항이 정의되기 때문이다. 이 과정에서 내포되는 오류가 상당히 많으며 이후 단계의 개발과정에서 나타나기 때문이다.

결론적으로 오류 정의를 위해서는 영역지식과 요구사항을 참조하여 수행해야 하는 단계이다.

#### 2) 오류 검출을 위한 활동

오류 정의가 완료되면 개발 단계에서 발생할 수 있는 오류를 예측할 수 있게 된다.

어떤 오류가 발생하는지 알 수 있기 때문에 오류를 찾을 수 있는 활동도 할 수 있게 된다.

막연히 오류를 정의한 것에 의하여 찾는 것도 한계가 있다. 발생한 오류는 이후 단계에서 원인을 찾아서 올바르게 수정을 해야 하는 목적이 있으므로 오류 검출 활동도 이와 부합되게 수행되어야 한다.

오류 검출 활동 시에 하드웨어 오류인지 소프트웨어 오류인지 환경오류인지를 구분하여야 한다. 이와 같은 형태를 검출하지 않으면 오류의 원인을 식별하게 어렵게 되고 대안은 마련할 수 가 없게 된다.

#### 3) 오류의 원인 식별

오류의 원인 식별을 위해서 오류정의와 오류 검출을 위한 활동이 선행되어야 한다.

오류를 단순히 찾아서 현상을 수정하게 되면 이미 오류의 내용을 다른 곳에서 참조하였거나 사용 중인 경우라면 향후에 많은 문제를 발생시키게 될 것이다. 오류의 원인을 원할히 찾고 향후 같은 문제가 발생했을 경우에도 이를

대비하기 위하여 원인식별 활동을 통한 오류의 원인도를 나타내는 산출물이 필요하게 된다. 이와 같은 산출물의 형태는 추적도, 표 또는 체크리스트의 형태가 된다.

향후 문제가 발생한 경우에 오류의 원인을 나타내는 산출물을 활용하여 오류에 대한 수정활동을 동시에 수행해야 한다.

#### 4) 오류의 영향도 측정

발생된 오류가 얼마나 심각한지를 측정해야 전체 시스템과 다른 소프트웨어에 영향을 얼마나 주는지 확인할 수 있다.

오류의 영향도를 측정하기 위해서는 오류의 심각도를 원인과 연관 지어서 분석해야 한다. 또한 영향도는 반드시 정량화하여 측정할 수 있는 체계를 마련해야 한다. 이와 같은 오류측정을 체계화하지 않으면 측정하는 담당자에 의하여 오류측정 결과치가 달라지기 때문이다. 그 과정과 방법을 정형화 및 판단기준을 정량화하여 프로세스화 할 필요가 있다.

오류의 영향도 측정은 기능의 우선순위하고는 다른 것이다. 기능의 우선순위는 개발하려는 기능을 핵심기능과 부가기능으로 나누어서 우선순위를 부여한 것이다. 오류의 영향도 측정에서의 정량화는 오류를 정성적으로 심각, 보통, 경미로 나눈다면 이를 기반으로 전체 시스템과 다른 소프트웨어 어느 정도의 영향을 주는지 정량적으로 접근하는 것이기 때문에 많은 고려사항과 복잡도가 증가하게 된다.

#### 5) 오류와 연관된 사항들의 수정 활동

오류에 대한 분석이 끝나고 영향도 측정이 완료되게 되면, 오류에 대하여 수정을 하여 잘못된 사항을 완전히 고쳐야 한다. 이때 추적성에 의하여 분석된 기능의 연관성을 같이 수정하여 오류가 전파되지 않도록 한다.

#### 6) 수정된 사항의 확인

수정된 사항은 체크리스트를 두어서 확인을 하도록 한다.

### 4. 결 론

본 연구에서 제시된 6단계는 오류항목 수정을 위한 선행 조건의 내용이다. 6단계에 대한 세부사항은 추가적으로 연구할 계획에 있다. 이와 같은 과정은 품질향상을 얻을 수 있으며, 세부적인 방법과 내용을 결정하여 실제 내용에 적용하고자 한다. 현재 단계에서는 개념을 가지고 접근을 하였으며, 향후 논문지에 게재할 예정이다.

### 참고문헌

- [1] 우치수 공역 “소프트웨어 공학 실무적 접근” Mc Grow Hill
- [2] 최은만 “객체지향 소프트웨어 공학” 사이텍미디어
- [3] 윤 청 “공적인 소프트웨어 개발 방법론” 생능출판사
- [4] 권기태 역 “소프트웨어공학” 홍릉과학 출판사
- [5] 최은만 “소프트웨어공학” 정의사