

소프트웨어 개발과정의 기술 리뷰 평가 방법

전희배¹, 양해술^{2*}

Evaluation Method of Technical Review in Software Development Process

Heui-Bae Jeon¹ and Hae-Sool Yang^{2*}

요 약 소프트웨어의 개발에서 테스트 비용을 적게 하는 유효한 기법중의 하나로 기술 리뷰의 실행이 있다. 본 연구에서는 기술 리뷰에 의한 테스트 비용의 감소율에 주목하여 새로운 리뷰 평가 척도 My 를 제안한다. 그리고, 실제의 소프트웨어 개발 과정에서 수집한 데이터를 사용하여 종래의 척도와 비교, 평가를 하였다. 그 결과 종래의 평가 척도에 대한 My 의 우위성과 유효성이 실험적으로 확인되었다. 또한, 리뷰 공정에서 수집한 데이터와 테스트 공정에서 수집한 데이터와 관계를 조사해서, 리뷰 공정에서 가능한 데이터만을 이용하여 My 의 값을 추정하는 방법에 대해서도 기술하였다.

Abstract Cost effectiveness is greatly related with the degree of reducing the testing cost by the technical reviews. In this paper, we present a new metric My for evaluating the cost effectiveness of technical reviews during software development. First, we estimate and compare My with conventional measure using data collected during practical software development procedure, then we show the validity and usefulness of the proposed measure My . Also by formulating the relationship between the data collected during the reviews and the test, we present a method to estimate the value of the metric My using only the data collected during review phase.

Key Words : 기술 리뷰(Technical Review), 관리 리뷰(Management Review)

1. 서론

최근, 컴퓨터 시스템에 있어서 소프트웨어 역할은 점점 중요하게 되었고, 소프트웨어 중의 폴트(fault)가 중대한 시스템 고장을 일으키는 경우가 많다. 따라서 소프트웨어의 테스트에 필요한 비용(테스트 비용)이 증대된, 그 비율 또한 소프트웨어 개발 비용 전체의 50%를 넘어 80%에 육박하고 있는 실정이다[5]. 따라서 테스트 비용을 적게 하는 기술의 개발은 소프트웨어 개발에서 생산성 향상을 목표로 대단히 중요하게 취급하고 있는 분야이다.

소프트웨어 리뷰는 테스트 비용을 적게 하는 대표적

인 기술들 중 하나이다. 특히, 소프트웨어 개발의 각 공정에서 작성된 산출물(product)을 평가하는 기술 리뷰(technical review)는 실제로 매우 중요한 작업으로 수행되고 있다. 일반적으로 설계공정에서 수행되는 설계 리뷰와 코딩 공정에서 수행되는 코드 리뷰가 가장 대표적이라고 할 수 있다.

프로그램의 성격에 따라 약간의 차이는 있지만 기술 리뷰의 실행에 의해 산출물중 대부분의 폴트를 테스트 공정 이전에 제거할 수 있다. 또한 기술 리뷰에서 너무 많은 폴트를 찾기 위한 노력이 비용 효과 면에서 더 나빠질 수 있으나 일반적으로 기술 리뷰의 실시에 의해 테스트 비용은 적어진다. 따라서 평균적으로 기술

본 연구는 지식경제부와 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (HTA-2008-(C1090-0801-0032))

¹서울벤처정보대학원대학교 정보경영학과(박사과정)

²호서대학교 벤처전문대학원 IT응용기술학과 교수

*교신저자: 양해술(hsyang@office.hoseo.ac.kr)

접수일 08년 07월 07일

수정일 08년 09월 10일

제재확정일 08년 10월 16일

리뷰에 의한 플트 제거 비용이 테스트에 의한 플트 제거 비용에 비해 적기 때문에 전체적인 소프트웨어 개발 비용도 줄어들게 된다.

그러나, 기술 리뷰는 지식 집약형 작업으로 동일한 정보를 사용해서 동일한 순서에 따라 작업을 수행해도 리뷰 작업자의 숙련도와 대상이 되는 생산물(소프트웨어 구성요소)에 대한 이해 용이성 등에 차이가 있으면, 기술 리뷰에서 제거된 플트수와 기술 리뷰에 필요한 비용은 큰 차이가 있다[3].

따라서, 테스트 비용을 적게 하고 소프트웨어 개발의 생산성을 향상시키기 위해서는 기술 리뷰의 실시 기술과 함께 기술 리뷰의 평가 기술을 확립하는 것이 중요하다.

지금까지 기술 리뷰의 유효성을 평가하기 위한 척도(이해, 리뷰 평가 척도)가 몇 가지 제안되어 있다[2][4]. 예를 들면, Myers는 발견된 플트수를 사용해서 기술 리뷰의 유효성을 평가하고 있다[4]. 또한 Fagan은 Error Detection Efficiency라는 리뷰 평가척도를 제안하고 있다[2]. 이와 같은 척도에서는 기술 리뷰에서 발견된 플트수와 리뷰 실시 전에 리뷰 대상이 되는 생산물 중에 포함되었던 플트수에 의해 평가를 수행한다.

최근에는, 플트수만이 아닌 소프트웨어 개발 비용에 착안한 리뷰 평가 척도도 제안되고 있다. 예를 들면 Collofello는 기술 리뷰에 필요한 비용과 기술 리뷰에 의해 절약된 비용에 기준하여 Cost Effectiveness라는 평가 척도를 제안하고 있다[1]. 그러나, 개발 비용이 크게 다른 두 종류의 프로젝트 간에 Cost Effectiveness를 사용해서 비교하는 것은 바람직하지 않다.

본 연구에서는 우선 기술 리뷰에 의해 테스트 비용의 감소율에 의거해서 정의되는 새로운 리뷰 평가 척도 My를 제안한다. 다음에 종래의 척도와와의 비교는 실제 소프트웨어 개발과정에서 수집한 데이터를 사용해서 수행한다. 제안된 정의로부터 My는 Fagan의 평가 척도와 Collofello의 평가 척도를 조합한 것으로 볼 수 있다. 즉, Fagan과 Collofello의 평가 척도가 적용될 수 없는 경우에도 My는 적용할 수 있는 가능성을 보인다.

이하, 제2장에서는 본 연구에서 대상으로 하는 소프트웨어 개발과정과 리뷰에 대해 기술한다. 제3장에서는 종래의 세 가지 평가 척도에 대해 간단히 소개하고 제4장에서는 기술 리뷰의 유효성을 평가하는 새로운 척도 My를 제안한다. 그리고 제5장에서는 실제 적용상의 문제점과 유효성을 실험적으로 확인하기 위해 A회사의 프로그래머 신입연구교육에서 수집한 데이터를 이용하여 종래의 척도와 제안 척도를 비교하여 고찰한다. 끝으로, 결론과 앞으로의 연구과제에 대해 기술하였다.

2. 소프트웨어의 리뷰와 개발 과정

2.1 소프트웨어의 리뷰

IEEE 표준에 의하면, 소프트웨어 리뷰는 「개발 계획에서 상정된 상황으로부터 오류나 모순을 파악하여 그 해소를 위한 개선을 촉구하는 것을 목적으로 소프트웨어의 구성요소와 프로젝트의 현상을 개발자 전원이 참가하여 평가하는 활동」이라고 정의하고 있다. 여기서 소프트웨어의 구성요소란 프로젝트 계획서, 요구명세서, 설계명세서, 원시 코드 등 개발 도중에 작성될 수 있는 모든 프로젝트 산출물이 포함된다.

소프트웨어 리뷰는 두 가지 타입으로 분류되며, 하나는 관리 리뷰(management review)이고 다른 하나는 기술 리뷰(technical review)이다. 관리 리뷰에서는 프로젝트 레벨에서의 개발 계획 평가와 프로젝트 현상 계획에 대한 평가를 수행하고, 기술 리뷰에서는 소프트웨어 구성요소의 평가를 수행한다[11]. 본 연구에서는 기술 리뷰(특히, 설계 리뷰와 코드 리뷰)에 착안하여 연구하였다. 이하 기술 리뷰를 간단히 리뷰라고 한다.

리뷰에 있어서 다음 항목에 대한 판정 결과에 의해, 구체적인 개선 방법을 검토한다.

- ① 소프트웨어 구성요소가 요구명세서를 만족하고 있는가?
- ② 프로젝트에서 사용하고 있는 계획, 표준, 가이드 라인에 의해 소프트웨어 구성요소의 개발이 진행되고 있는가?
- ③ 소프트웨어 구성요소에 대한 변경은 올바르게 진행되고, 그 영향은 변경명세서(change specification)에 의해 명확히 된 부분에서만 나타나고 있는가?

2.2 소프트웨어 개발 과정

본 연구에서는 그림 1에 나타난 소프트웨어 개발 과정을 생각한다. 개발 과정은 설계공정, 코딩 공정, 테스트 공정의 세 가지 공정으로 분류한다. 또 개발은 팀에 의해 수행되고, 최초로 주어진 요구명세서에는 플트가 포함되지 않고, 또 개발 도중에 요구명세서의 변경은 없다고 가정한다.

2.3 설계 공정

설계공정에서는 요구명세서를 기초로 하여 개발해야 할 시스템의 구조, 소프트웨어의 구성요소, 인터페이스 등이 결정된다. 다음에 이것들이 문서화되고 요구명세서를 만족하고 있는지의 여부를 검증한다[10]. 일반적

으로 설계 공정의 최후에 설계 리뷰가 수행된다. 설계 리뷰에서는 팀의 전 멤버에 의해 설계 문서가 요구 명세서를 만족하고 있는지 확인한다.

2.4 코딩 공정

코딩 공정에서는 팀의 각 멤버가 자신에게 할당된 설계 문서에 기초해서 코드를 작성하고, 디버깅을 수행한다. 위의 설계 공정에서 기술한 바와 같이 설계 문서에는(ed-Edr)개의 폴트가 포함되어 있다. 이 경우, 설계 문서에 포함되어 있는 폴트로 인하여 그것과 동수의 폴트가 코드에 만들어져 작성되는 것으로 한다. 또, 코딩 작업에 있어 코드 작성자의 실수가 원인이 되어 다시 ec개의 폴트가 코드에 만들어져 작성되는 것으로 한다.

코딩 공정의 최후에 작성된 코드에 대해 팀의 전 멤버에 의한 코드 리뷰가 행해진다. 코드 리뷰에서는 작성된 코드가 설계문서와 요구명세서를 만족하고 있는지 확인된다. 코드 리뷰에 의해 코드 중의 ecr개의 폴트가 제거되는 것으로 한다. 따라서, 코딩 공정 종료시에 코드에 포함되는 폴트의 수는 {(ed-Edr+ec)-Edr}이다. 또한, 코드 리뷰에 필요한 시간을 Tcr로 한다.

로 그 기능이 동작하는지를 검사하는 단위 테스트와 이들 코드들을 결합하는 과정에서 문제점이 없는지를 검사하는 결합 테스트로 나누어진다. 단위 테스트에서는 작성된 코드마다 그 동작 확인을 코드 작성자 자신이 수행하고 폴트를 제거한다. 결합 테스트에서는 작성된 모든 코드를 결합하고 그 동작 확인을 개발자 전원이 수행하고 폴트를 제거한다.

단위 테스트와 결합 테스트에 의해 코드에 포함된 {(ed-Edr+ec)-Edr}개의 폴트가 제거되는 것으로 가정한다. 단, 각 폴트의 제거가 단위 테스트와 결합 테스트 어느 것에 의해 수행되었는지는 구별하지 않는 것으로 한다.

또한, 테스트 시간 Tt는 각 코드 작성자에 의한 단위 테스트 시간과 개발자 전원에 의한 결합 테스트 시간의 합으로 계산한다.

3. 관련 연구

중래의 리뷰 평가 척도 중에서 대표적인 세 가지 척도라고 할 수 있는 Fagan의 Mf 척도, Myers의 Mm 척도 및 Collofello의 Mc 척도에 대해 간단히 기술하였다 [7, 8, 9].

3.1 Fagan의 Mf 척도

Fagan은 상세 설계 리뷰, 단위 리뷰의 유효성을 평가하였다[2]. 평가에 있어서 Error Detection Efficiency라 부르는 평가 척도 Mf를 도입하였다. Mf는 총폴트수에 대한 리뷰에서 제거 폴트수의 비율로 정의 된다.

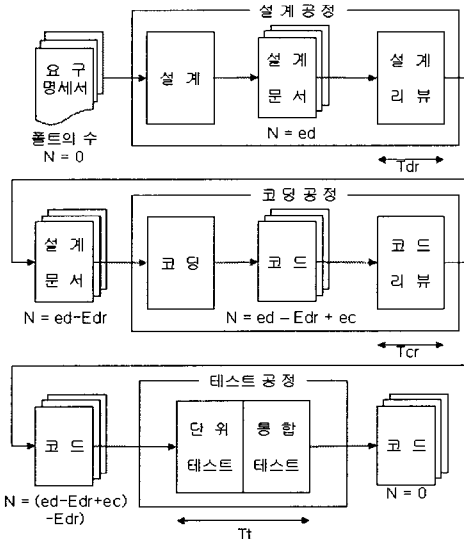
Fagan의 접근 방식을 그림 1에 적용하면, 설계 리뷰와 코드 리뷰의 평가 척도는 다음 식으로 표현하다.

$$Mf = \frac{Edr + Ecr}{ed + ec} \tag{1}$$

평가 척도 Mf는 (1)의 식과 같이 'E'와 'e'의 상대 비율에 의한 척도임을 알 수 있다. 그리고 Mf 평가 척도는 리뷰 대상의 소프트웨어 구성요소에 동일한 수의 폴트가 포함되지 않은 경우에도 적용할 수 있다는 점에서 Myers의 평가 척도 Mm보다 유용하다고 볼 수 있다. 그러나, Mf 평가 척도도 Mm 평가 척도와 동일하게 리뷰에 필요한 비용은 평가에 적용하지 않고 있다.

3.2 Myers의 Mm 척도

Myers는 외부 명세서에 주목하여 그 기능과 데이터



[그림 1] 소프트웨어의 개발 과정

2.5 테스트 공정

테스트 공정에서는 작성된 모든 코드가 요구명세서를 만족하고 있는지의 여부가 평가된다. 특히, 최종 단계에서는 결합된 코드의 평가가 수행된다[6]. 일반적으로, 테스트 공정은 완성된 코드가 초기의 설계 의도대

와의 관계로부터 테스트하는 블랙 박스 테스트, 코드 구조와 제어흐름을 참조하면서 테스트 데이터와 테스트 방법을 생각하는 화이트 박스 테스트 및 코드 리뷰와 이들을 조합한 방법의 유효성을 평가하였다[4]. 평가는 발견된 폴트수에 따라 수행한다.

Myers의 접근 방식을 그림 1에 적용하면 설계 리뷰와 코드 리뷰의 유효성 평가 척도는 다음 식으로 표현한다.

$$Mm = Edr + Ecr \quad (2)$$

그러나, 리뷰 대상의 소프트웨어 구성요소에 동일한 수의 폴트가 포함되고 동시에 리뷰와 테스트에 필요한 비용이 중요하지 않는 경우에만 Mm을 리뷰 평가 척도로서 사용할 수 있다.

3.3 Collofello의 Mc 척도

Collofello는 리뷰와 테스트 등의 폴트 제거 기술의 비용의 관점에서 평가하였다[1]. 평가에 있어서 Cost Effectiveness라고 부르는 평가 척도 Mc를 제안하였다. Mc는 폴트 제거 기술의 실시에 의해서 절약된 비용과 리뷰에 소요된 비용의 비율로 정의된다.

Collofello의 접근 방식을 그림 1에 적용하면, 설계리뷰와 코드 리뷰에 대한 유효성의 평가 척도는 다음 식으로 표현된다.

$$Mc = \frac{\Delta Ct}{Cr} \quad (3)$$

단, ΔCt 는 리뷰에 의해 감소된 테스트 비용의 절대값을, Cr 은 리뷰 비용을 나타낸다.

리뷰에 의한 폴트 1개당 평균 제거 비용을 cr 로 하면, 리뷰 비용 Cr 은 다음과 같은 식으로 표현한다.

$$Cr = (Edr + Ecr) \cdot cr \quad (4)$$

리뷰에 의한 테스트 비용의 감소량 ΔCt 에 대해 설명하기로 한다. 리뷰에 의해 제거된 $(Edr+Ecr)$ 개의 폴트가 리뷰가 아닌 테스트에 의해서 제거되었다고 가정할 경우, 제거에 필요한 테스트 비용이 ΔCt 임을 생각할 수 있다. 리뷰에 의해 폴트가 제거되지 않은 경우에도 테스트에 의한 폴트 1개당 평균 제거 비용은 변화하지 않는다고 가정한다. 이 경우 ΔCt 는 다음 식으로 표현한다.

$$\Delta Ct = (Edr + Ecr) \cdot ct \quad (5)$$

여기서, ct 는 테스트에 의한 폴트 1개당 평균 제거 비용이다. 식(3), (4), (5)로부터 Mc 는 다음 식으로 표현할 수 있다.

$$Mc = \frac{(Edr + Ecr) \cdot ct}{(Edr + Ecr) \cdot cr} = \frac{ct}{cr} \quad (6)$$

즉, Mc 는 리뷰에 의한 폴트 1개당 평균 제거 비용과 테스트에 의한 폴트 1개당 평균 제거 비용의 비율이다.

그리고, 리뷰 비용 Cr 은 리뷰 시간($Tdr+Tcr$)에 의해 표현할 수 있다고 가정한다. 이 때 cr , ct 는 각각 다음 식으로 표현된다.

$$cr = \frac{Tdr + Tcr}{Edr + Ecr} \quad (7)$$

$$ct = \frac{Tt}{(ed + ec) - (Edr + Ecr)} \quad (8)$$

4. 기술 리뷰 평가척도의 제안

4.1 리뷰 평가척도 My의 제안

종래의 평가 척도 Mf , Mm , Mc 중에서 리뷰에 의한 테스트 비용의 감소량을 고려하고 있는 Mc 가 가장 실용적이라고 생각된다.

그러나, 다음과 같은 2종류의 소프트웨어 개발 프로젝트 간에 리뷰의 유효성을 비교하는 것을 생각할 경우, Mc 는 리뷰 평가 척도로서 적당하지 않음을 알 수 있다.

- ① 프로젝트 A: 리뷰에 의해 절약된 테스트 비용을 100, 리뷰에 필요한 비용을 10으로 한다.
- ② 프로젝트 B: 리뷰에 의해 절약된 테스트 비용을 1000, 리뷰에 필요한 비용을 100으로 한다.

비용의 단위는 man-months이다. 또한 프로젝트 A, B 공히 리뷰에 의해 폴트가 제거되지 않았던 경우에 필요한 테스트 비용을 1000으로 한다.

이 경우 프로젝트 A, B 공히 Mc 에 의한 평가값은 10이 되고 리뷰의 유효성은 동일하다고 생각할 수 있다. 그러나, 프로젝트 B에 있어서의 리뷰가 프로젝트 A에 있어서의 리뷰보다 더 유효하다는 것은 명확하다. 왜냐하면 프로젝트 A에서는 리뷰를 수행해도 다시 테스트 비용으로서 900이 필요하다. 이것은 리뷰에 의해

절약된 비용이 9배의 값이 된다. 한편, 프로젝트 B에서 전체 테스트 비용이 리뷰에 의해서 절약되어 있으므로 테스트에 의해 폴트를 제거할 필요는 없다.

이와 같은 사실에 기준하여 다음과 같은 새로운 평가 척도, M_y 를 제안한다. 리뷰에 의해서 절약된 비용과 리뷰에 의해 폴트가 제거되지 않은 경우에 필요한 비용의 비로서 M_y 를 정의한다. 이와 같은 접근 방식은 그림 1에 적용하면 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned}
 M_y &= \frac{\Delta C_t - C_r}{C_t + \Delta C_t} \\
 &= \frac{(E_{dr} + E_{cr}) \cdot ct - (E_{dr} + E_{cr}) \cdot cr}{(ed + ec) \cdot ct} \\
 &= \frac{E_{dr} + E_{cr}}{ed + ec} \cdot \frac{ct - cr}{ct} \quad (9)
 \end{aligned}$$

4.2 기존 척도와의 비교

여기에서는 제안 척도 M_y 와 종래의 척도 M_m , M_f , M_c 간의 관계를 비교하기로 한다. 정의식 (2), (6), (9)로부터 M_y , M_f , M_c 간에는 다음과 같은 관계가 성립한다.

$$\begin{aligned}
 M_y &= \frac{E_{dr} + E_{cr}}{ed + ec} \cdot \frac{ct - cr}{ct} \\
 &= \frac{E_{dr} + E_{cr}}{ed + ec} \left(1 - \frac{cr}{ct}\right) \\
 &= M_f \left(1 - \frac{1}{M_c}\right) \quad (10)
 \end{aligned}$$

이와 같이 제안 척도 M_y 는 종래의 척도 M_f 와 M_c 를 조합한 것이 된다. 여기서 M_f 의 값이 상수라고 가정하면 위의 식으로부터 M_y 는 M_c 의 역수에 비례한다.

$$M_y = a_1 \left(1 - \frac{1}{M_c}\right) \quad (11)$$

단, a_1 은 상수이다.

다음에 C_r 와 C_t 의 비율이 일정하다고 가정하면, M_y 는 M_f 에 비례한다. 특히, C_t 가 C_r 에 비해 대단히 큰 경우(M_c 의 값이 대단히 큰 경우) 다음 식과 같이 M_y 와 M_f 는 근사적으로 동등하게 된다.

$$\begin{aligned}
 M_y &= M_f \left(1 - \frac{1}{M_c}\right) \\
 &\approx M_f (1 - 0) \\
 &= M_f \quad (12)
 \end{aligned}$$

또한 C_t 가 C_r 에 비해 대단히 큰 경우 즉, $C_r \ll C_t$ 이며 동시에 리뷰 전에 소프트웨어 구성요소에 포함되는 폴트수가 일정할 경우 다음 식과 같이 M_y 는 M_m 의 정수 배가 된다.

$$\begin{aligned}
 M_y &= \frac{E_{dr} + E_{cr}}{ed + ec} \\
 &= a_2 (E_{dr} + E_{cr}) \\
 &= a_2 \cdot M_m \quad (13)
 \end{aligned}$$

단, a_2 는 상수이다.

5. 평가 사례

5.1 평가의 개요

종래의 리뷰 평가 척도와 제안 평가 척도를 실제의 소프트웨어 개발과정에서 수집한 데이터를 사용해서 비교한다. 본 연구의 평가 사례는 동일한 문제에 대한 제안 척도와 종래의 척도와의 상대적인 비교와 평가 분석을 위한 실험으로서 데이터는 어느 프로그래밍 연구교육에서 수집한 것으로 주요 특징은 다음과 같다.

- ① 연구자의 학력은 대학 전산학과 졸업하고 연구교육의 실무 프로그래밍 기법의 교육을 이수한 자로 한다.
- ② 7팀이 각각 동일한 소프트웨어 시스템(사무처리용 파일처리시스템)을 C++언어를 사용하여 작성한다.
- ③ 각 팀은 4~5인으로 구성하고 팀의 구성은 연구중의 성적 등에 의해서 가급적 팀 간의 능력 차이가 없도록 배려하였다.
- ④ 개발은 각 팀이 작성한 테스트 데이터가 모두 정확히 실행된 경우에 종료한다. 그리고 테스트 데이터 작성 방침은 전체 팀이 동일하다.

5.2 평가 데이터

실험 데이터는 표 1과 같으며, 이 중에서 전체 폴트

수 (ed+ec)와 리뷰에 의해 제거된 폴트수(Edr+Ecr) 및 리뷰 비용 Cr은 각 개발자가 제출한 데이터에 의해서 산출하였다.

한편, 테스트 비용 Ct는 자동 수집 가능한 단말기 사용시간에 의해 산출하였으며, 그 외의 값 cr, ct, ΔCt 및 Ct+ΔCt는 정의에 기준하여 다른 평가 데이터로부터 산출하였다.

[표 1] 평가 데이터

항목 \ Team	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇
전체 폴트수 ed+ec	21	30	22	18	28	38	47
리뷰로 제거한 폴트수 Edr+Ecr(=Mm)	8	15	7	6	13	18	15
리뷰 비용 Cr(=Tdr+Tcr)(mn.)	500	600	510	570	570	560	420
리뷰에 의한 폴트의 평균 제거 비용 cr(=Tt/Mm)(mn.)	62.5	41.3	72.9	95.0	42.8	31.1	28.0
테스트 비용 Ct(Tt)(mn.)	5947	5032	7186	4367	5274	4866	7521
테스트에 의한 폴트의 평균 제거 비용 ct(mn)	457.5	335.5	479.1	363.9	351.6	243.3	235.0
리뷰에 의한 테스트 비용의 감소량 ΔCt(mn.)	3632	5000	3333	2222	4642	4390	3488
리뷰를 하지 않는 경우의 테스트 비용 Ct+ΔCt(mn.)	9579	10032	10519	6589	9916	9256	11009

6. 비교 분석

6.1 평가값의 비교

7개 팀(T₁ ~ T₇)의 리뷰에 대한 4가지 척도에 의한 평가값을 표 2에 정리하였다. 표 2로부터 팀 T₂에 대한 평가값은 어느 팀보다 높고, 역으로 팀 T₄에 대한 평가값은 어느 팀보다도 낮음을 알 수 있다. 여기서 4가지 척도에 의한 평가값 간의 상관관계를 조사해본 결과를 표 3에 정리하였다.

[표 2] 리뷰 평가값

Team	Mm	Mf	Mc	My
T ₁	8	0.38	7.27	0.327
T ₂	15	0.50	8.06	0.437
T ₃	7	0.32	6.56	0.268
T ₄	6	0.33	3.90	0.251
T ₅	13	0.46	8.14	0.411
T ₆	18	0.47	7.84	0.414
T ₇	15	0.32	8.30	0.279

표 3을 보면 척도 간에는 어느 정도 높은 상관관계가 있음을 알 수 있다. My, Mf 간의 상관계수 값은 0.99로 매우 높음을 알 수 있다. 이것은 Mc의 값이 이번 실험에서 얻어진 정도의 크기(3.90~8.30)라도 식 (12)에 나타난 바와 같은 관계가 My와 Mf간에 성립함을 알 수 있다.

[표 3] 평가값 간의 상관관계

Mf	0.65	-	-
Mc	0.76	0.52	-
My	0.72	0.99	0.66
	Mm	Mf	Mc

팀 T₂와 팀 T₄에 대한 평가 값은 모든 척도가 거의 동일하지만, 팀 T₇에 대한 평가값은 척도에 따라 크게 차이를 알 수 있다. Mf 및 My에 의한 평가값은 대단히 낮다. 특히, Mf에 의한 평가값은 7개 팀 중에서 가장 낮음을 알 수 있다. 그러나, Mc에 의한 평가값은 역으로 높고, 7개 팀 중에서 가장 높다.

팀 T₇의 경우, 리뷰에 의해 제거된 폴트수가 15개로 비교적 많다. 그러나, 리뷰 시간(리뷰 비용 Cr)은 420 분으로 다른 팀에 비해서 약 16%~33% 정도 짧다. 이와 같은 결과는 리뷰에 의한 폴트의 평균 제거 비용 cr은 28.0이 되고, 7개 팀 중에서 최소이므로 Mc의 값을 크게 하는 원인 중의 하나가 된다. 한편, 팀 T₇의 전체 폴트수는 47개로 대단히 많고 테스트 비용 Ct도 7521로 7개 팀 중에서 최대이다. 따라서 팀 T₇은 비교적 제거가 용이한 폴트만을 단시간의 리뷰 중에 제거한 것으로 생각할 수 있다.

이와 같은 현상은 비교적 용이하게 제거할 수 있는 많은 폴트가 다수 제거되므로 리뷰 비용 Cr이 적게 된다는 것을 보여준다. 한편, 테스트에서는 제거가 곤란

한 소수의 폴트 때문에 Ct는 커진다. 리뷰에 의해서 제거된 폴트는 제거가 용이하므로 그것들을 만약 테스트에 의해 제거한 것이라 하더라도 그 평균 제거 비용은 Ct에 기준해서 식(5)에 의해 구해진 테스트 비용의 감소량 ΔCt는 실제값보다도 크게 된다. Cr이 적어짐과 동시에 ΔCt가 크게 평가되므로 Mc의 값이 대단히 커짐을 알 수 있다.

My도 Mc와 같이 ΔCt에 기준해서 산출된다. 그러나, 정의식(9)로부터 알 수 있듯이 ΔCt가 실제값보다 크게 평가되고 있어도 My의 값이 커지는 것은 아니다. 오히려 그렇게 한 경우에는 테스트 비용 Ct가 커지므로 My의 값은 적어진다.

이와 같이 Mc에 의한 평가가 타당하지 않다고 생각되는 경우에도 My에 의한 평가는 가능하다고 볼 수 있다.

6.2 평가값 My의 예측

제안 척도 My에서는 전체 폴트수(ed+ec)와 테스트에 의한 폴트의 평균 제거 비용 Ct를 평가에 사용하고 있다. 이들 값은 테스트 공정이 종료할 때까지 알 수가 없다. 따라서 My에 의한 리뷰의 평가는 테스트 종료 후에 수행하는 것이 된다. 이것은 My의 척도로서의 유용성을 낮추는 요인이 될 수 있다.

여기서 리뷰 공정에서의 평가가 가능하기 때문에 리뷰 공정에서 수집 가능한 몇 가지 데이터(ed+ec)와 Ct와의 관계를 조사하여 그 결과를 표 4에 정리하였다.

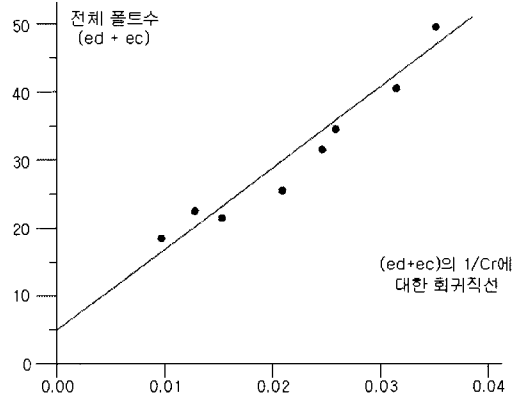
[표 4] 데이터 간의 상관관계

구분	Edr+Ecr	Tdr+Tcr	Cr	1/Cr
ed+ec	0.84	-0.43	0.88	0.98
1/Ct	0.82	-0.267	0.73	0.92

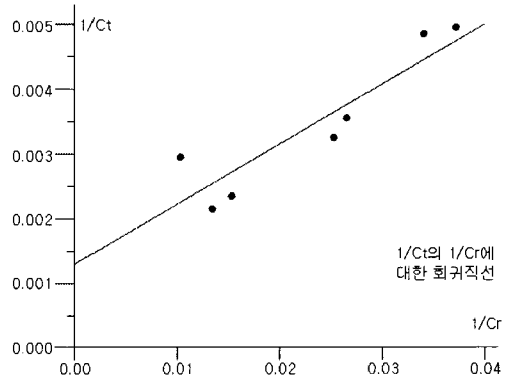
표 4로부터 (ed+ec)와 Cr의 역수는 모두 리뷰에 의한 폴트의 평균 제거 비용 Cr의 역수와 상관관계가 높다는 것을 알 수 있다. (ed+ec)와 Cr의 관계를 그림 2에, Ct와 Cr의 관계를 그림 3에 각각 표현하였다. 또, (ed+ec)와 Cr, Ct와 Cr사이에는 다음과 같은 관계가 성립함을 알 수 있다.

$$(ed+ec) = 4.88 + 1090\left(\frac{1}{Cr}\right) \quad (14)$$

$$\frac{1}{Ct} = 0.00114 + 0.0848\left(\frac{1}{Cr}\right) \quad (15)$$



[그림 2] 전체 폴트수와 1/Cr



[그림 3] 1/Cr과 1/Ct의 관계

식(14), (15)를 사용해서 산출한 7개 팀에 대한 예측값(리뷰 공정에서의 데이터만을 이용해 구한 추정값) My와 각각의 추정 오차를 표 5에 정리하였다.

[표 5] My의 추정값과 추정 오차

Team	\hat{M}_j	$\left \frac{\hat{M}_j - M_j}{M_j} \right \times 100(\%)$
T ₁	0.303	7
T ₂	0.420	4
T ₃	0.289	8
T ₄	0.286	14
T ₅	0.376	9
T ₆	0.400	4
T ₇	0.300	8

표 5로부터 추정 오차의 평균은 약 7%이고 리뷰 공정의 데이터만을 이용한 정도 높은 예측이 가능함을 알 수 있다.

7. 결론

본 연구에서는 기술 리뷰에 의한 테스트 비용의 감소에 주목해서 새로운 리뷰 평가 척도 My를 제안하고 실제의 소프트웨어 개발 과정에서 수집한 데이터를 사용해서 그 평가를 수행하였다. 그 결과, 종래의 리뷰 평가 척도에서 고려하지 못한 테스트 비용에 대한 제안 척도 My의 우수성과 유효성을 실험적으로 고찰하였다. 그리고 리뷰 공정에서 수집된 데이터와 테스트 공정에서 수집된 데이터 간의 관계를 조사하고, 리뷰 공정에서 수집 가능한 데이터만을 이용해서 My의 값을 추정하는 방법에 대해서도 기술하였다.

앞으로의 연구과제는 제안 척도 My의 타당성을 입증하기 위한 방법으로 좀 더 일반화하여 시스템적이며 수학적인 검증 방법이 필요하며, 제안 척도에 의한 리뷰의 평가 결과를 소프트웨어 생산성 향상에 기여하기 위해서는 리뷰의 평가에서 필요한 데이터의 수집과 개발자에 대한 평가 결과의 피드백을 자동적으로 수행하는 시스템의 개발이 필요하다고 생각한다.

참고문헌

[1] J. S. Collofello, S. N.: "Evaluating the effectiveness of reliability assurance techniques," J. Syst. & Software, Vol. 9, No. 3, pp.191~195, 1999.

[2] M. E. Fagan: "Design and code inspection to reduce errors in program development," IBM Syst. J., 15, No. 3, pp. 182~211, 1996.

[3] S. Kusumoto, K. Matsumoto, T. Kikuno, K. Torii: "Experimental evaluation of metrics for review activities," Proceedings of 10th Software Symposium, pp. 236~241, 2000.

[4] G. J. Myers: "A controlled experiment in program testing and code walk throughs / inspections," Commun. ACM, Vol. 21, No. 9, pp. 760~768, 1998.

[5] G. J. Myers: "The Art of Software Testing," Second Edition John Willey & Sons, Inc., 2004.

[6] A. R. Brown, W. A. Sampson: "Program Debugging," London, Macdonald, 1993.

[7] S. Kusumoto, K. Matsumoto, T. Kikuno and K. Torii: GINER: data collection and analysis system," IEICE Technical Report, SS90-5, 2000.

[8] V. R. Basili and R. W. Reiter: "An investigation of human factors in software development," Computer,

Vol. 12, pp. 21~38, 1999.

[9] D. R. Wallace, R. U. Fujii: "Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standard," NVS Special Publication 500-180, 2002.

[10] IEEE Standard Glossary of Software Engineering Terminology, IEEE, ANSI/IEEE Std. 729~783, 1990.

[11] R. Pressman, "Software Engineering: A Practitioner's Approach", McGraw-Hill, 2003.

전 희 배(Heui-Bae Jeon)

[정회원]



- 1992년 : 서울산업대학 전산학과 졸업(학사)
- 1995년 : 한국외국어대학 정보관리학과 졸업(석사)
- 2006년-현재 : 서울벤처정보대학원대학교 박사과정
- 2000년 12월 : 교보정보통신(주) 일본지사장

- 2001년 - 현재 : 일본키스코(주) 대표이사
- 2006년 - 현재 : 한국키스코(주) 대표이사

<관심분야>

S/W공학, S/W품질보증과 평가, SW개발방법론

양 해 술(Hae-Sool Yang)

[정회원]



- 1975년 : 홍익대학교 전기공학과 졸업(학사)
- 1878년 : 성균관대학교 정보처리학과 졸업(석사)
- 1991년 : 日本 오사카대학 정보공학과 S/W공학 전공(공학박사)
- 1975년~79년 : 육군중앙경리단 전자계산실 시스템분석장교

- 1980년~95년 : 강원대학교 전자계산학과 교수
- 1986년~87년 : 日本 오사카대학 객원연구원
- 1995년~2002년 : 한국S/W품질연구소 소장
- 1999년~현재 : 호서대학교 벤처전문대학원 교수

<관심분야>

S/W공학(특히, S/W 품질보증과 품질평가, 품질감리 및 컨설팅, OOA/OOD/OOP, SI), S/W 프로젝트관리, 컴포넌트 기반 개발방법론과 품질평가