

소프트웨어믿음성성장모형을 리용한 새로운 결합분석방법

김철성, 량기선

위대한 령도자 김정일동지께서는 다음과 같이 교시하시였다.

《모든 생산단위에서는 사회적으로동을 절약하고 생산의 효과성을 높이며 인민생활을 향상시키고 대외무역을 발전시키는데서 제품의 질을 높이는것이 가지는 중요성을 뚝게 인식하고 과학적인 품질관리질서를 철저히 세워 제품의 질을 결정적으로 높여야 합니다.》
(《김정일선집》 증보판 제15권 68페이지)

최근에 많은 소프트웨어개발단위들에서 소프트웨어믿음성에 대한 분석평가에 기초하여 소프트웨어개발의 최종단계에서 제품을 출하하겠는가 말겠는가를 판단하고있다. 소프트웨어믿음성성장모형(SRGM)을 리용하면 여러가지 믿음성분석결과를 쉽게 얻을수 있는 우점이 있다. 하지만 소프트웨어제품의 믿음성과 품질에 대한 분석결과를 리해하기 위해서는 일정한 전문지식이 필요하며 더우기 실지 개발형편을 장악하지 못하고 담당자들로 부터 문서만 받아본다면 더욱 어려운 일이다.

우리는 논문에서 소프트웨어제품의 믿음성과 품질을 평가하기 위한 새로운 분석방법을 제안하고 이것을 오유추적도구에 구현하여 그 효과성을 평가하였다.

1. 시계열분석을 리용한 소프트웨어믿음성평가방법

1) 소프트웨어믿음성성장모형

제한된 시험기간에 소프트웨어제품에 남아있는 모든 결함들을 찾아내어 제거하는것은 거의 불가능하므로 시험단계에서 믿음성을 평가하고 분석하기 위한 정량적방법이 필요하다.

선행연구[1, 2]에서는 이때까지 제안된 시험환경인자와 운영환경인자를 포함한 많은 소프트웨어믿음성성장모형을 소개하였다. 특히 프로젝트관리자들이 SRGM에 의해 얻어진 믿음성평가결과를 리용하여 언제 시험을 중지하고 제품을 출하하겠는가를 결정하도록 하고있다.

소프트웨어믿음성성장모형에서는 시험에서 얻어진 결함자료에 기초하여 높은 정확도를 가진 믿음성평가결과를 제공해준다.

$N(t)$ 는 시험시간($t \geq 0$)에 검출된 결함들의 총수를 표시하는 우연수, $H(t)$ 는 $N(t)$ 의 기대값이라고 하자. 이때 비동차뿔쫑분포에 기초한 SRGM[1, 2]은 평균값함수에 의해 다음과 같이 표시된다.

$$\Pr(A|N(t)=n) = \frac{H(t)^n}{n!} \exp(-H(t)) (n=0, 1, 2, \dots), \quad H(t) = \int_0^t h(x)dx$$

여기서 $\Pr(A|N(t)=n)$ 은 시간 t 에서 검출된 결함수가 n 일 확률이고 $h(x)$ 는 비동차뽕송분포의 밀도함수로서 시험시간 x 에서의 결함검출률을 표시한다.

소프트웨어의 믿음성은 시험시간과 시험에 의해 발견된 결함의 총수와의 관계를 의미한다.

그러므로 믿음성성장곡선은 시험시간이 지나감에 따라 발견된 결함의 총수의 시간에 따르는 거동을 보여준다.

그러나 이러한 방법들은 개발의 어느 한 시점만을 취하여 입력된 자료들에 기초하여 믿음성분석결과만을 제공하므로 현재 진행하고있는 시험에 의해 믿음성이 커지고있는지 판단하기 힘들며 또한 리용하고있는 도구들 역시 소프트웨어개발에 대해 불충분한 정보를 보장하고있다.

우리는 이러한 문제점들을 해결하기 위한 한가지 방법으로서 과거에 기록된 정보들을 분석함으로써 소프트웨어개발공정의 경향성을 파악하기 위한 새로운 분석방법을 제안하였다. 또한 오류추적도구에 이러한 경향성을 보여주는 그래프현시를 통한 분석기능을 새로 추가하였다.

2) 시계열분석을 리용한 새로운 분석방법

시험기간에 진행한 믿음성에 대한 과거의 정량적인 분석결과로서 결함의 총수에 대한 예측값, 소프트웨어믿음성의 예측값, 검출할수 있는 결함수의 예측값, 2개의 기준에 기초한 최적출하날자를 리용하면 경향성을 파악할수 있다.

(1) 결함의 총수의 시간에 따르는 거동을 통한 분석

매 분석시점에서 얻어진 결함들의 총수의 시간에 따르는 거동은 그림 1과 같다.

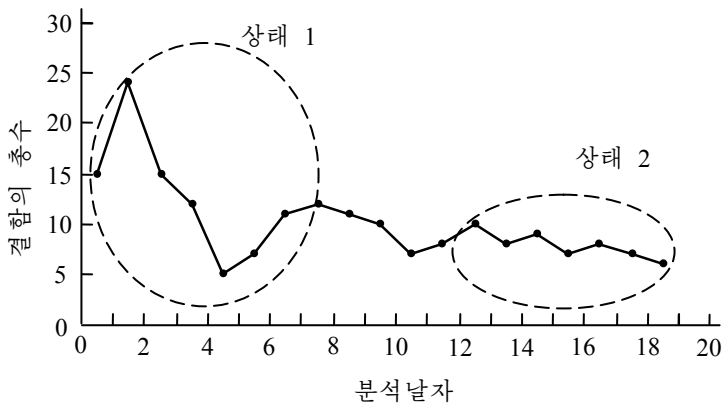


그림 1. 매 분석시점에서 얻어진 결함들의 총수의 시간에 따르는 거동

검출된 결함의 총수의 예측값이 오르내리는 거동을 보고 개발된 소프트웨어의 믿음성이 충분한 준위에 도달하였는가를 판정하여 시험을 계속할것인가 말것인가를 결정한다.

그림 1의 상태 1에서는 시험이 여전히 필요하다고 결정해야 한다. 왜냐하면 모든 분석시점에서 예측된 결함의 총수가 너무 크게 변동(증가, 감소되는 폭이 크다.)되기 때문이다. 한편 상태 2에서는 시험을 중지하고 소프트웨어를 출하할수 있다고 판단한다. 왜냐하

면 분석결과에 기초한 예측된 결함들의 총수의 거동이 일정한 범위에서 안정되어있으며 시험활동에 의한 영향을 받지 않기때문이다.

결론적으로 보면 지나간 기록들의 분석결과가 그래프상의 어떤 값으로 수렴하면 지금 진행하는 시험활동이 만족하다(다시말하면 민음성 혹은 품질이 증가하고있다.)고 볼수 있으며 시험이 적절하게 진행되었다고 볼수 있다.

(2) 소프트웨어민음성의 시간에 따르는 거동을 통한 분석

소프트웨어가 특정한 날에 시험에서 그 어떤 결함도 발생하지 않고 수요자의 요구대로 운영될 확률인 소프트웨어민음성추정의 시간에 따르는 거동은 그림 2와 같다.

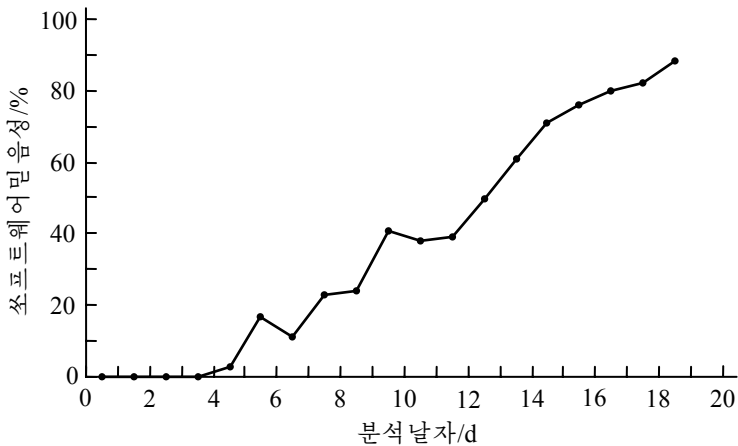


그림 2. 소프트웨어민음성추정의 시간에 따르는 거동

프로젝트성원들과 관리성원들이 프로젝트의 세부를 알지 못한다고 해도 소프트웨어민음성그래프를 보면서 소프트웨어제품이 안정하게 동작하고있는지를 알수 있다.

(3) 검출할수 있는 결함수의 시간에 따르는 거동을 통한 분석

소프트웨어시험을 통해 하루동안에 검출할수 있는 결함수의 증가 및 감소정도가 시간에 따라 어떻게 달라지는가를 그림 3에 보여주었다.

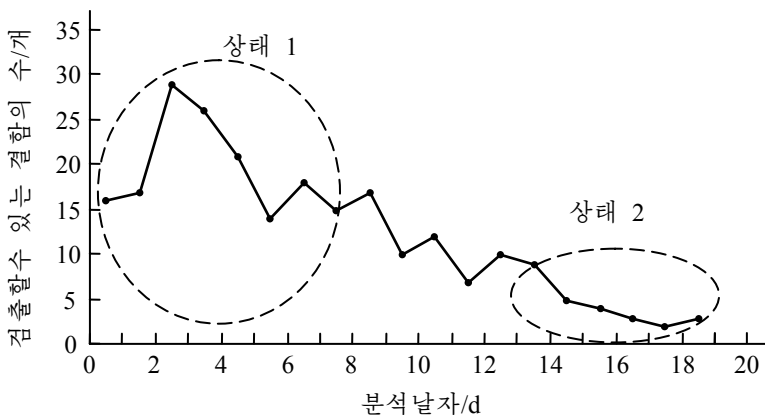


그림 3. 검출할수 있는 결함수의 시간에 따르는 거동

시험단계에서는 수요자의 요구에 따라 구현된 기능을 확인하기 위해 여러가지 시험

경우들을 많이 실행한다. 그러면 시험경우를 실행하여 영향을 받는 소프트웨어의 시험경로의 모임이 존재하며 소프트웨어제품에서 이 시험경로들의 성능은 시험경로포괄범위로 표시된다.

이 시험경로들에 의해 포괄된 잠재적인 소프트웨어결함들은 《검출할수 있는 결함》이다. 하루동안 시험할 때 새로 포함된 시험경로마다 1개의 결함이 숨어있을수 있다고 보면 검출할수 있는 결함의 수는 증가하는데 이 값은 믿음성평가측정의 한가지인 그 시점에서의 시험령역성장률에 의해 영향을 받는다.

한편 시험활동에서 하루에 여러개의 소프트웨어결함들을 검출할 때 결과는 믿음성평가측정에 포함된 그 시점에서의 결함검출률로 나타날수 있다. 그러면 검출할수 있는 결함의 수는 감소한다. 왜냐하면 검출할수 있는 결함의 일부는 실지로 검출되어 제거되기때문이다.

따라서 그 시점에서의 결함검출률과 그 시점에서의 시험령역성장률사이의 차이를 고찰해보면 소프트웨어믿음성/품질이 시험을 끝낼수 있는 준위에 이르렀는지를 판단할수 있다.

그림 3에서 상태 1에 있을 때에는 아직도 보충적인 시험을 더 해야 한다는 결정을 내려야 한다. 그것은 검출할수 있는 결함의 수가 분석할 때마다 크게 변동하기때문이다.

한편 상태 2에 있다면 시험을 중지해도 된다. 그것은 분석결과에 기초한 검출할수 있는 결함의 수가 0에 가까우며 안정한 거동을 보여주기때문이다.

이와 같이 그래프에서 현재까지의 시험기록들에 대한 분석결과가 령으로 수렴하면 시험이 성과적으로 진행되고있다는것을 말해주는것으로 된다.

(4) 출하예정날자의 시간에 따르는 거동을 통한 분석

두가지 형태로 예측한 출하예정날자의 시간에 따르는 거동 즉 남아있는 결함률을 목표로 했을 때의 수행가능한 날자(선)와 소프트웨어의 총비용을 최소로 하는 최적날자(점선)의 시간에 따르는 거동은 그림 4와 같다.

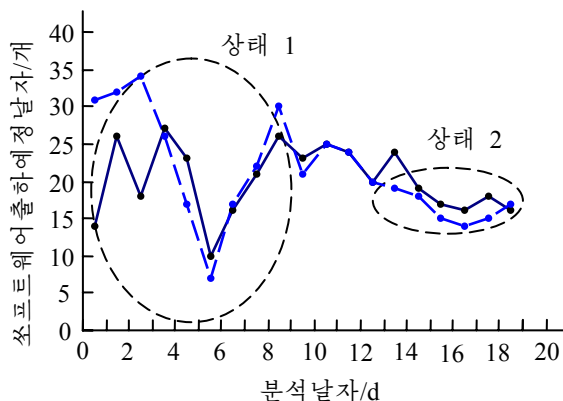


그림 4. 출하예정날자의 시간에 따르는 거동

남아있는 결함률에 대해 보기로 하자.

소프트웨어시험공정을 거친다고 하여 소프트웨어에 있는 잠재적인 모든 결함들을 다 없앨수 없다는것은 누구나 다 알고있다.

그러므로 수요자요구조건과 소프트웨어제품의 특성을 고찰하면서 그 소프트웨어에 대해 결함검출률의 허용값을 정의함으로써 검출되어야 하는 결함수의 목표값을 설정할수 있다. 즉 결함검출률이 목표값에 도달하는 날자는 분석에 의해 추정된 결함의 총수가 남아있는 결함률의 특정한 허용값아래에 이르게 되는 날자로 된다. 전체 소프트웨어비용을 최소화하자면 오유수정과 유지보수비용을 포함한 전체 생명주기비용의 예측값을 최소화해야 한다. 때문에 우리는 두가지 평가지표에 대하여 시험을 중지하고 소프트웨어제품을 출하하는 날자를 추정할수 있다.

그림 4의 상태 1을 보면 시험을 아직 계속하여야 한다고 판단해야 한다. 왜냐하면 출하할수 있는 날자는 매 분석에서 소프트웨어제품의 총결함수의 변동에 따라 큰 요동을 가지고있기때문이다.

한편 그림 4의 상태 2는 시험을 중지하고 공급자로부터 소프트웨어제품을 받을수 있는 때라는것을 알수 있다. 그것은 출하할수 있는 날자가 그 어떤 큰 요동이 없이 일정한 날자에서 안정된 거동을 보여주기때문이다.

따라서 현재까지의 시험기록에 대한 연속적인 분석이 안정된 거동을 보여주면 현재까지의 시험이 성과적으로 진행되었다고 볼수 있다.

2. 오유추적도구에 구현된 민음성분석기능과 그것의 효과성평가

오유추적도구는 시험이 진행되는 전기간 나타난 오유들을 보고하고 그 대책정형을 감시, 추적하는 기능과 함께 각종 통계기능을 포함하고있다.

그러므로 우리는 이미전부터 리용하여오던 오유추적도구에 우에서 언급한 시계렬분석방법을 적용한 그래프현시기능을 새로 추가하여 사용자들이 소프트웨어시험을 언제까지 해야 하는가를 합리적으로 결정하도록 하였다.

12개 프로젝트의 시험에 이와 같은 시계렬분석방법을 적용한 결과 시험기간이 단축된 반면에 소프트웨어의 민음성이 더욱 높아졌다.

다음의 표에 대표적인 2개 프로젝트의 시험을 통해 논문에서 제안한 시계렬분석방법의 효과성을 보여주었다.

표. 시계렬분석방법의 효과성

| No. | 평가지표 | 프로젝트 1의 시험 | | 프로젝트 2의 시험 | |
|-----|--------------|------------|---------|------------|---------|
| | | 선행방법 | 시계렬분석방법 | 선행방법 | 시계렬분석방법 |
| 1 | 시험기간/d | 59 | 47 | 128 | 116 |
| 2 | 소프트웨어의 민음성/% | 81 | 93 | 87 | 92 |

맺 는 말

소프트웨어민음성성장모형에 기초한 시계렬분석방법을 제안하고 이것을 12개의 프로젝트들에 적용하여 그 효과성을 평가하였다.

참 고 문 헌

- [1] H. Pham; Software Reliability. Springer-Verlag, 11~80, 2000.
- [2] M. Kimura, T. Fujiwara; Software Reliability. JUSE Press, 35~70, 2011.

주체107(2018)년 8월 5일 원고접수

A New Fault Analysis Method Using Software Reliability Growth Model

Kim Chol Song, Ryang Ki Son

In this paper we propose a new analysis concept for assessing the software product's reliability and illustrate the output results obtained by a tool.

Key words : software reliability growth model, time series analysis