

소프트웨어 테스트 노력의 비교 연구

A Comparison Study on Software Testing Efforts

최규식

건양대학교

(che)@konyang.ac.kr

abstract -

We propose a software-reliability growth model incorporating the amount of uniform and Weibull testing efforts during the software testing phase in this paper. The time-dependent behavior of testing effort is described by uniform and Weibull curves. Assuming that the error detection rate to the amount of testing effort spent during the testing phase is proportional to the current error content, the model is formulated by a nonhomogeneous Poisson process. Using this model the method of data analysis for software reliability measurement is developed.

The optimum release time is determined by considering how the initial reliability $R(x_0)$ would be. The conditions are $R(x_0) > R_o$, $R_o > R(x_0) > R_o^d$ and $R(x_0) < R_o^d$ for uniform testing efforts. Ideal case is $R_o > R(x_0) > R_o^d$.

Likewise, it is $R(x_0) \geq R_o$, $R_o > R(x_0) > R_o^{\frac{1}{\beta}}$ and $R(x_0) < R_o^{\frac{1}{\beta}}$ for Weibull testing efforts. Ideal case is $R_o > R(x_0) > R_o^{\frac{1}{\beta}}$.

1. 서론

최근 운영시스템, 제어프로그램, 적용프로그램과 같은 여러 가지 소프트웨어 시스템이 더욱더 복잡화 및 대형화되고 있기 때문에 신뢰성이 높은 소프트웨어 시스템을 개발하는 일이 매우 중요하며, 따라서 소프트웨어 제품 개발에 있어서 소프트웨어의 신뢰도가 핵심사항이라고 할 수 있다. 소프트웨어 개발에서 중요한 문제는 개발된 소프트웨어의 테스트를 언제 중단하여 발행할 것이냐를 결정하는 것이다.

그동안 테스트 기간중의 소프트웨어 결함 검출현상을 설명하기 위한 여러 가지 소프트웨어 신뢰도 모델이 개발되었다. 소프트웨어 테스트에 의해서 발견되는 누적결함(또는 소프트웨어 고장간의 시간간격)과 소프트웨어 테스트 시간간격 사이의 관계를 짓는 모델들을 소프트웨어 신뢰도 성장모델(SRGM ; software reliability growth model)[1]이라 한다. 이러한 모델들을 이용하여 평균 초기결함의 수, 평균 고장간 시간 간격, 임의의 테스트 시간에 소프트웨어 내의 평균 잔여결함수, 소프트웨어 신뢰도 함수와 같은 소프트웨어 신뢰도 척도를 추정할 수 있다.

소프트웨어 개발에는 많은 개발자들이 소요된다. 소프트웨어 테스트 단계 기간동안에는 소프트웨어의 신뢰

도가 내재 결함을 검출 및 수정하는데 소요되는 개발자원의 양에 크게 의존한다. Musa 등[2]은 기존 소프트웨어 신뢰도 성장모델을 분류하는 하나의 안을 개발하였다. Yamada 등[3]은 역일 테스트 시간, 테스트 노력량, 테스트 노력에 의해서 검출되는 소프트웨어 결함의 수 사이의 관계를 명시적으로 설명할 수 있는 간단하고도 새로운 모델을 제시하였다. 테스트 노력은 테스트 단계에서 소요되는 인력, CPU시간, 실행테스트 케이스 등등에 의해서 측정된다.

본 논문에서는 일정한 값으로서의 테스트노력과 웨이블곡선으로서 테스트노력을 시간종속 거동적으로 설명한다. 소프트웨어 테스트에서 결함 검출비가 현재의 내재 결함수에 비례하고, 임의의 테스트 시간에서 그 비례가 현재의 테스트 노력 여하에 달려 있다는 것을 가정하여 비동차 포아송 과정(NHPP)[4]에 바탕을 둔 신뢰도성장 모델을 개발한다.

2장에서는 NHPP를 기준으로 한 SRGM에 대해서 고장확률 및 신뢰도를 연구하고, 3장에서는 테스트 시간중 두 가지 안에 대해서 목표신뢰도를 맞추어서 발행하는 발행 시각 결정 방법에 대해서 연구한다. 4장에서는 여러 조건을 고려하여 각 조건에 맞는 최적발행정책을 연구한다.

2. 소프트웨어의 신뢰도

소프트웨어 신뢰도는 규정한 환경 하에서 주어진 시간에 소프트웨어를 결함 없이 운영할 수 있는 확률인 것으로 정의하며, 다음과 같이 조건확률로 표현할 수 있다.[2]

$$R(x|s) = \Pr\{X_k > x | S_{k-1} = s\} \quad (1)$$

이는 소프트웨어를 개발하여 결함검출 테스트를 시작하여 계속 s 유닛 시간 동안 $(k-1)$ 번째 결함을 발견하여 수정한 후, k 번째 결함이 발견되기 전까지 운영될 x 시간동안에 고장 없이 소프트웨어가 동작할 확률인 것이다.

2.1 일정테스트 노력곡선

NHPP의 표준 이론으로부터 평균치 함수를

$$m(t) = a(1 - e^{-bt}) \quad (2)$$

로 정의할 때 임의의 $t \geq 0$ 과 $x > 0$ 에서

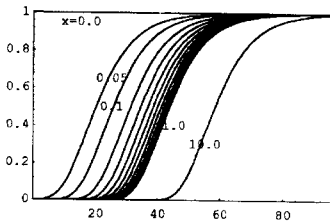
$$\Pr\{N(t+x) - N(t) = k\} = \frac{[m(t+x) - m(t)]^k}{k!} \exp\{-[m(t+x) - m(t)]\} \quad (3)$$

이므로, 식(1)의 신뢰도는 다음과 같이 표현할 수 있다.

$$R(x|t) = \Pr\{N(t+x) - N(t) = 0\} = \exp\{-m(x)e^{-bt}\} \quad (4)$$

즉, 어느 시각 t 부터 $(t+x)$ 시각까지 새로운 결함이 발견되지 않을 확률을 신뢰도로 정의하며, 이는 평균치 함수의 차이를 지수함수의 지수로 취한 형태를 하고 있다.

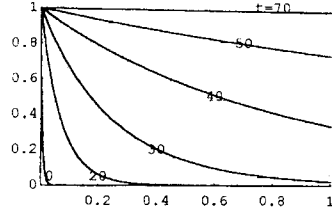
식(4)로 표시된 신뢰도의 특성을 이해하기 위해 테스트시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도와 관계는 [그림-1], [그림-2]와 같다.



[그림-1] 일정형 발행시각과 신뢰도와와의 관계

[그림-1]의 경우, 최종 결함 수정 후 경과되는 각각의 시간에 대해서 테스트 시간과 신뢰도 성장과의 관계를 보여주고 있다. 일정한 경과시간 x 에 대해서 테스트시간 및 발행시기를 늦추면 늦출수록 신뢰도가 s 형 곡선으로 성장함을 알 수 있다. 그림에서 $x = 0.0$ 일 때는 테스트

시간에 관계없이 신뢰도가 1이나, x 의 값이 커지면 커질수록 곡선이 횡축의 우측으로 이동하여 신뢰도가 저하됨을 알 수 있다.



[그림-2] 결함 수정후 경과시간과 신뢰도와와의 관계

이와는 대조적으로 [그림-2]의 경우, 주어진 각각의 테스트 시간 t 에 대해서 최종 결함 수정 후 경과시간 x 와 신뢰도 성장과의 관계를 보여주고 있다. 일정한 테스트 시간 t 에 대해서 경과시간 x 가 작으면 작을수록 신뢰도가 높으며, 경과시간 x 가 증가함에 따라 신뢰도가 급격히 감소된다.

목표신뢰도를 R_0 라 하면 식(4)로부터

$$\exp\{-m(x)e^{-bt}\} = R_0 \quad (5)$$

가 된다.

2.2 웨이블 테스트노력 곡선

본 논문에서는 시각 t 에서의 테스트노력 형상을 기술하기 위해 테스트 노력함수로서 웨이블곡선을 이용한다.

$$w(t) = a \cdot \beta \cdot m \cdot t^{m-1} \cdot \exp\{-\beta t^m\} \quad (6)$$

(6)의 적분형태

$$W(t) = a(1 - \exp\{-\beta t^m\}) \quad (7)$$

는 시각 $(0, t]$ 에서의 누적 테스트 노력량을 나타낸다.

소프트웨어 고장은 시스템 내에 잔존하고 있는 소프트웨어의 결함에 의해서 프로그램 동작이 제대로 되지 않는 것을 말한다. 평균치 함수를 다음과 같이 정의한다.

$$m(t) = a(1 - \exp\{-rW(t)\}) \quad (8)$$

소프트웨어 결함검출 현상을 통계적으로 모델링할 때에 식(8)의 $m(t)$ 에 의한 NHPP에 근거하여 $N(t)$ 의 평균치함수를 정의하면 웨이블 테스트 노력함수를 고려한 소프트웨어 신뢰도성장모델을 만들 수 있다.

$$\Pr\{N(t) = n\} = \text{poim}(n; m(t)) \quad (9)$$

NHPP 고장강도함수는 평균치 함수의 미분 형태이다.

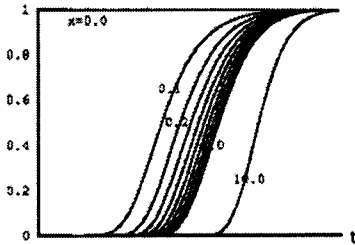
$$\lambda(t) = dm(t)/dt = a \cdot r \cdot w(t) \cdot \exp\{-rW(t)\} \quad (10)$$

식(8)의 $m(t)$ 를 가진 NHPP 모델에 의해서 소프트웨어 신뢰도 평가에 대한 두 가지 정량적 평가 척도를 얻을

수 있다. 시각 t 에서의 신뢰도는 다음과 같다.

$$R(x|t) = \exp\{-a(\exp[-r \cdot W(t)] - \exp[-r \cdot W(t+x)])\} \quad (11)$$

식(11)로 표시된 신뢰도의 특성을 이해하기 위해 테스트시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도의 관계는 [그림-3]과 같다.



[그림-3] 웨이블형 발행시각과 신뢰도와 의 관계

이 그림에서는 최종 결함 수정 후 경과되는 각각의 시간에 대해서 발행시각과 신뢰도 성장과의 관계를 보여주고 있다. 일정한 경과시간 x 에 대해서 테스트시간 및 발행시기를 늦추면 늦출수록 신뢰도가 성장함을 알 수 있다. 또한, 비록 신뢰도가 성장하여 목표신뢰도 이상이 될 수 있으나, 결함 수정 후 경과시간이 길어지면 길어질수록 결함 발견 확률이 높아 신뢰도가 저하된다는 것도 알 수 있다. 이것은 일정 테스트 노력일 경우와 유사하다.

$R(x|t)$ 는 시각 t 에서 최종적으로 결함을 발견하여 수정한 후, x 유니트의 경과시간동안 새로운 결함이 발견되지 않을 확률이다. 소프트웨어를 개발하여 결함테스트를 하면 할수록 결함을 발견하여 수정하는 빈도가 작아지므로 신뢰도가 성장되며, 결함 수정 후 경과시간이 길어지면 길수록 결함 발견 확률이 높아지기 때문에 소프트웨어의 신뢰도는 낮아진다.

3. 발행시각 결정

3.1 일정테스트 노력

- 1) $R_0 > R(x|0) > R_0^g$ 에서 양의 유일 해 $T^* = T_1$ 가 존재한다.
- 2) $R(x|0) > R_0$ 이면 $T_1 < 0$ 이므로 $T^* = T_1 = 0$ 이다.
- 3) $R(x|0) < R_0^g$ 이면 $T_1 > T_{LC}$ 이므로 $T^* = T_1 = T_{LC}$ 이다

1)의 경우

$$\exp(-m(x)) = R(x|0) \quad (12)$$

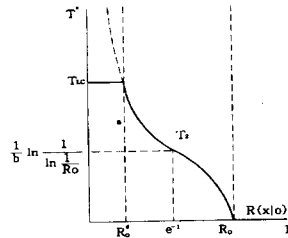
이므로,

$$m(x) = -\ln R(x|0) = \ln \frac{1}{R(x|0)} \quad (13)$$

으로 되어 T_1 을 다음과 같이 구할 수 있다.

$$T^* = T_1 = \frac{1}{b} \ln \left\{ \frac{\ln \frac{1}{R(x|0)}}{\ln \frac{1}{R_0}} \right\} \quad (14)$$

이러한 관계를 [그림-4]에 표시하였다.



[그림-4] 목표신뢰도와 발행시각과의 관계

3.2 웨이블테스트

마찬가지로, 신뢰도를 요건에 최근접시키는 유일한 시각이 존재한다. 최적 소프트웨어 발행시각은 미리 규정된 소프트웨어 목표신뢰도를 만족시키는 최근접이 되는 시각이다.

목표신뢰도를 R_0 라 하면

$$\exp\{-ae^{-\sigma(1-e^{-rT})}[1-e^{-\sigma(1-e^{-rT})}]\} = R_0 \quad (15)$$

이다. 식(15)를 만족하는 발행시각을 T_2 라 하면

$$T_2 = \left\{ -\frac{1}{\beta} \ln \left[1 + \frac{1}{a\gamma} \ln \frac{\ln R_0}{\ln R(x|0)} \right] \right\}^{\frac{1}{m}} \quad (16)$$

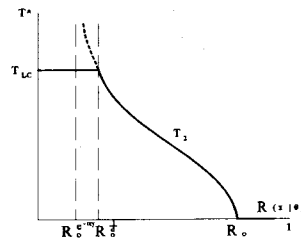
이므로

1) $R_0 > R(x|0) > R_0^g$ 에서 양의 유일 해 $T^* = T_2$ 이 존재한다.

2) $R(x|0) \geq R_0$ 이면 $T_2 \leq 0$ 이므로 $T^* = T_2 = 0$ 이다.

3) $R(x|0) < R_0^g$ 이면 $T_2 \geq T_{LC}$ 이므로 $T^* = T_2 = T_{LC}$ 이다.

이와 같은 내용을 [그림-5]에 표시하였다.



[그림-5] 목표신뢰도와 발행시각 곡선

4. 최적 발행 시각

4.1 일정 테스트 노력

$R(x|0)$ 은 소프트웨어를 개발하여 테스트를 거치지 않은 상태에서 x 시간까지 시간이 경과할 때의 소프트웨어 신뢰도를 나타내는 것이다.

1) $R(x|0) > R_0$

는 소프트웨어 개발 즉시 신뢰도가 목표신뢰도를 만족하기 때문에 더 이상 결함 발견을 위한 테스트를 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다. 그러나, 이러한 경우는 현실적으로 고려하기 어려운 경우이다.

2) $R_0 > R(x|0) > R_0^{\frac{1}{k}}$

는 소프트웨어를 개발한 후 테스트 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시기와 총 비용을 최저로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

3) $R(x|0) < R_0^{\frac{1}{k}}$

는 소프트웨어의 전 수명기간에 걸쳐서 테스트를 해도 목표신뢰도를 만족시키지 못하는 경우로서 소프트웨어 개발에 실패한 경우로 볼 수 있다.

따라서, 이 중에서 가장 이상적인 범위는

$R_0 > R(x|0) > R_0^{\frac{1}{k}}$

인 경우이다.

4.2 웨이블 테스트 노력

$R(x|0)$ 은 소프트웨어를 개발하여 테스트를 거치지 않은 상태에서 x 시간까지 시간이 경과할 때의 소프트웨어 신뢰도를 나타내는 것이다.

1) $R(x|0) \geq R_0$

는 소프트웨어 개발 즉시 신뢰도가 목표신뢰도를 만족하기 때문에 더 이상 결함 발견을 위한 테스트를 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다. 그러나, 이러한 경우는 현실적으로 고려하기 어려운 경우이다.

2) $R_0 > R(x|0) > R_0^{\frac{1}{k}}$

는 소프트웨어를 개발한 후 테스트 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시기와 총 비용을 최저로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

3) $R(x|0) < R_0^{\frac{1}{k}}$

는 소프트웨어의 전 수명기간에 걸쳐서 테스트를 해도 목표신뢰도를 만족시키지 못하는 경우로서 소프트웨어 개발에 실패한 경우로 볼 수 있다.

따라서, 이 중에서 가장 이상적인 경우는

$R_0 > R(x|0) > R_0^{\frac{1}{k}}$

인 경우이다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한 쪽 또는 양 쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

5. 결론

소프트웨어의 신뢰도에 대한 정의를 하고, 테스트 시간의 경과와 신뢰도와의 관계, 결함 수정 후 경과되는 시간과 신뢰도와의 관계를 연구하였다. 발행시각을 결정함에 있어서 개발 후 테스트를 시작하기 전의 신뢰도 $R(x|0)$ 가 어떠한 조건에 있는가를 검토하여 각 조건에 따른 최적 발행시각을 결정하였다. 이 중에서 이상적인 경우는 $R_0 > R(x|0) > R_0^{\frac{1}{k}}$ 인 경우이다. 이 범위의 조건일 때에 최적 발행시각을 결정하고자 하는 본 연구의 목적에 부합된다. 이 경우에 대해서 최적발행시각이 목표신뢰도 면에서 어떠한 경향을 보이는지 그림으로 표시하였다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한 쪽 또는 양 쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적 발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

웨이블곡선인 경우를 고찰해보면 이상적인 경우는

$R_0 > R(x|0) > R_0^{\frac{1}{k}}$

인 경우이다. 이 범위의 조건일 때에 최적 발행시각을 결정하고자 하는 본 연구의 목적에 부합된다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한 쪽 또는 양 쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적 발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

감사의 글		
본 연구는	한국과학재단	목적기초연구
(R01-2000-000-00273-0) 지원으로 수행되었음		

참고문헌

- [1] C. V. Ramamoorthy, F. B. Bastani, "Software reliability - Status and perspectives", IEEE Trans. on Software Eng., vol. SE-8, pp354-371, 1982 August
- [2] J. D. Musa, A. Iannino, K. Okumoto, "Software Reliability : Measurement, Prediction, Application", pp230-238, 1987 March
- [3] S. Yamada, H. Ohtera, H. Narihisa, "Software reliability growth models with testing-efforts", IEEE Trans. on Reliability, vol. R-35, pp19-23, 1986 April
- [4] H. Ascher, H. Feigold, "Repairable Systems Reliability : Modeling, Inference, Misconceptions, and Their Causes", 1984, March
- [5] Martin Trachtenberg, "The Linear Software Reliability Model and Uniform Testing", IEEE Trans. on Reliability, vol. R-34, pp8-16, 1985 April