

# 소프트웨어의 일정테스트노력과 웨이블 테스트 노력의 비교 연구

## A Study on the Reliability Comparison of S/W between Uniform Testing and Weibull Testing

최규식, 김용경  
건양대학교

Che Gyu-Shik, Kim Yong-Kyung  
Konyang Univ.

### 요약

본 논문에서는 소프트웨어 테스트 단계중에 발생하는 테스트노력 소요량을 고려한 소프트웨어 신뢰도 성장 모델을 제시하여 시간중속적인 테스트 노력소요량 동태를 일정 테스트 노력일 때와 웨이블 테스트 노력일 때를 비교하여 연구한다. 소프트웨어 신뢰도 척도에 대한 데이터 분석기법을 개발하도록 한다. 테스트 시간의 경과와 신뢰도와의 관계도 심도 있게 연구한다. 목표신뢰도를 만족시키는 최적발행시각을 정한다. 개발 후 테스트를 시작하기 전의 신뢰도가 어떠한 조건에 있는가를 검토하여 각 조건에 따른 최적 발행시각을 결정한다.

### Abstract

We propose software reliability growth model, considering testing effort resource during testing stage of S/W, and compare the time dependent testing effort resource behavior in this paper. We develop the data technology method for the S/W reliability measure. We study in detail between the time elapse and reliability. Also, we determine the optimum release time which meets the target reliability. We decide optimum release time for each condition how the reliability is good before testing after development.

## I. 서론

소프트웨어 개발에는 많은 개발자들이 소요된다. 소프트웨어 테스트 단계 기간동안에는 소프트웨어의 신뢰도가 내재 결함을 검출 및 수정하는데 소요되는 개발자원의 양에 크게 의존한다. Musa 등은 기존 소프트웨어 신뢰도 성장모델을 분류하는 하나의 안을 개발하였다. Yamada 등은 역일 테스트 시간, 테스트 노력량, 테스트 노력에 의해서 검출되는 소프트웨어 결함의 수 사이의 관계를 명시적으로 설명할 수 있는 간단하고도 새로운 모델을 제시하였다.

본 논문에서는 일정한 값으로서의 테스트노력과 웨이블곡선으로서 테스트노력을 시간중속 거동적으로 설명한다.

## II. 소프트웨어의 신뢰도

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

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

그림 1에 이 관계를 표시하였다.

### 1. 일정테스트 노력곡선

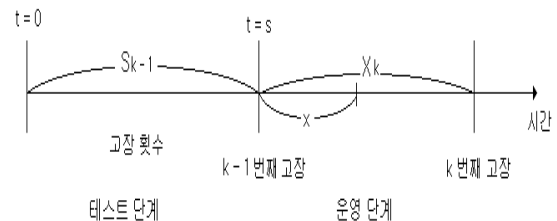
NHPP의 표준 이론으로부터 평균치 함수를

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

로 정의할 때 임의의  $t \geq 0$  과  $x > 0$  에서

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

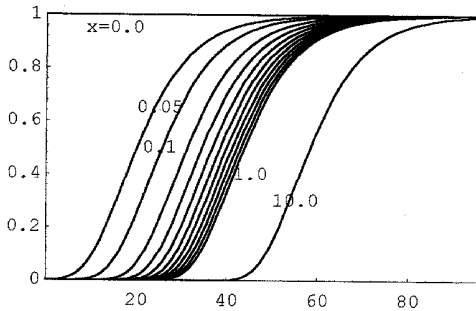
이므로, 식(1)의 신뢰도는 다음과 같이 표현할 수 있다.



▶▶ 그림 1. 고장발생 표현

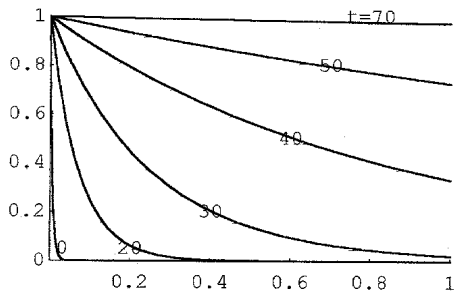
$$R(x|t) = \exp[-m(x)e^{-bt}] \quad (4)$$

식(4)로 표시된 신뢰도의 특성을 이해하기 위해 테스트시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도와 관계는 그림 2, 그림 3과 같다. 그림-2의 경우, 최종 결함 수정 후 경과되는 각각의 시간에 대해서 테스트 시간과 신뢰도 성장과의 관계를 보여주고 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. 발행시간과 신뢰도와의 관계

이와는 대조적으로 그림 3의 경우, 주어진 각각의 테스트 시간  $t$ 에 대해서 최종 결함 수정 후 경과시간  $x$ 와 신뢰도 성장과의 관계를 보여주고 있다. 일정한 테스트시간  $t$ 에 대해서 경과시간  $x$ 가 작으면 작을수록 신뢰도가 높다.



▶▶ 그림 3. 결함 수정후 경과시간과 신뢰도와의 관계

우리가 원하는 목표신뢰도를  $R_0$ 라 하면

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

가 된다.

## 2. 웨이블 테스트노력 곡선

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

$$w(t) = \alpha \cdot \beta \cdot m \cdot t^{m-1} \cdot \exp[-\beta t^m] \tag{6}$$

(6)의 적분형태

$$W(t) = \alpha(1 - \exp[-\beta t^m]) \tag{7}$$

는 시간  $(0,t)$ 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.

$$m(t) = \alpha(1 - \exp[-rW(t)]) \tag{8}$$

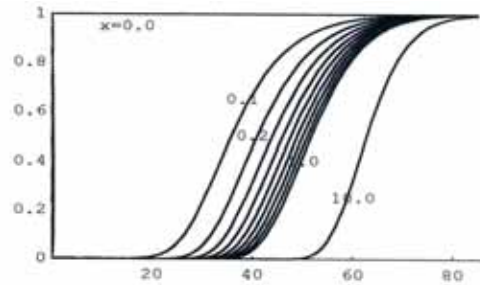
NHPP 고장강도함수는 평균치 함수의 미분 형태이다.

$$\lambda(t) = dm(t)/dt = \alpha \cdot r \cdot w(t) \cdot \exp[-rW(t)] \tag{9}$$

식(9)의  $m(t)$ 를 가진 NHPP 모델에 의해서 소프트웨어 신뢰도 평가에 대한 두 가지 정량적 평가 척도를 얻을 수 있다. 시간  $t$ 에서의 신뢰도는 다음과 같다.

$$R(x|t) = \exp\{-\alpha(\exp[-r \cdot W(t)] - \exp[-r \cdot W(t+x)])\} \tag{10}$$

식(10)로 표시된 신뢰도의 특성을 이해하기 위해 테스트 시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도의 관계는 그림 4와 같다.



▶▶ 그림 4. 발행시간과 신뢰도와의 관계

이 그림에서는 최종 결함 수정 후 경과되는 각각의 시간에 대해서 발행시간과 신뢰도 성장과의 관계를 보여주고 있다. 일정한 경과시간  $x$ 에 대해서 테스트시간 및 발행시기를 늦추면 늦출수록 신뢰도가 성장함을 알 수 있다.

$$R(x|0) = \exp\{-\alpha[1 - e^{-\alpha x(1 - e^{-\beta x^m})}]\} \tag{11}$$

이다.

따라서, 시간  $T$ 에서 발행된 소프트웨어의 신뢰도는 경과시간  $x$ 에 대해서 지수함수적으로 감소한다. 감소비를 줄이기 위해서는 발행시간을 늦추거나 결함검출비를 높여야 한다.

### III. 발행시각 결정

#### 1. 일정테스트 노력

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

1)의 경우

$$\exp(-m(x)) = R(x|0) \tag{12}$$

이므로,

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

으로 되어 식(4)로부터  $T_1$ 을 다음과 같이 구할 수 있다.

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

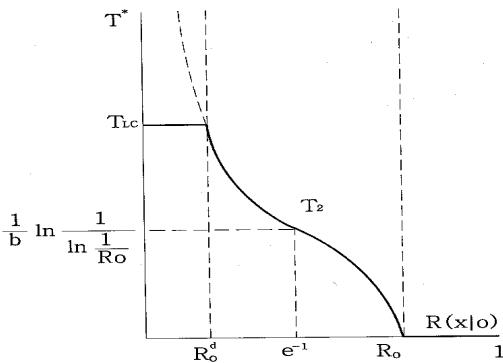
이 때

$$T_1 = \frac{1}{b} \left\{ \ln \frac{1}{R_o} \right\} \tag{15}$$

이다. 이러한 관계를 그림 5에 표시하였다.

#### 2. 웨이블테스트

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



▶▶ 그림 5. 목표신뢰도와 발행시각과의 관계

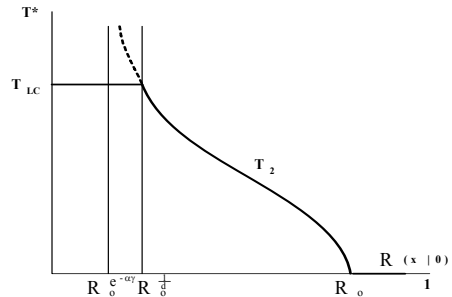
목표신뢰도를  $R_o$ 라 하면

$$\exp \{ -ae^{-\alpha x(1-e^{-\beta x^*})} [1 - e^{-\alpha x(1-e^{-\beta x^*})}] \} = R_o \tag{16}$$

이다.

- 1)  $R_o > R(x|0) > R_o^d$ 에서 양의 유일 해  $T^* = T_2$ 이 존재한다. 이 값은 식(23)으로 표현된다.
- 2)  $R(x|0) \geq R_o$ 이면  $T_1 \leq 0$ 이므로  $T^* = T_2 = 0$ 이다.
- 3)  $R(x|0) < R_o^d$ 이면  $T_1 \geq T_{LC}$ 이므로  $T^* = T_2 = T_{LC}$ 이다. 이와 같은 내용을 그림 6에 표시하였다.

$R_o > R(x|0) > R_o^d$ 는 소프트웨어를 개발한 후 테스트 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다.



▶▶ 그림 6. 목표신뢰도와 발행시각 곡선

$R(x|0) < R_o^d$ 는 소프트웨어의 전 수명기간에 걸쳐서 테스트를 해도 목표신뢰도를 만족시키지 못하는 경우이다.

### IV. 최적 발행 시각

#### 1. 일정 테스트 노력

- 1)  $R(x|0) > R_o$ 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.
- 2)  $R_o > R(x|0) > R_o^d$ 는 소프트웨어를 개발한 후 테스트 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시각과 총 비용을 최저로 하는 발행시각 중 큰 값을 취하는 것이 이상적이다.
- 3)  $R(x|0) < R_o^d$ 는 소프트웨어의 전 수명기간에 걸쳐서 테스트를 해도 목표신뢰도를 만족시키지 못하는 경우로서 소프트웨어 개발에 실패한 경우로 볼 수 있다. 따라서, 이

중에서 가장 이상적인 범위는  $R_o > R(x|0) > R_o^d$ 인 경우이다.

## 2. 웨이블 테스트 노력

- 1)  $R(x|0) \geq R_o$ 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.
- 2)  $R_o > R(x|0) > R_o^g$ 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)  $R(x|0) < R_o^g$ 는 소프트웨어의 전 수명기간에 걸쳐서 테스트를 해도 목표신뢰도를 만족시키지 못하는 경우로서 소프트웨어 개발에 실패한 경우로 볼 수 있다. 따라서, 이 중에서 가장 이상적인 경우는  $R_o > R(x|0) > R_o^g$ 인 경우이다.

## 3. 실제 예

$\alpha=6759.6$ ,  $\beta=4.5343 \times 10^{-3}$ ,  $\gamma=1.5791 \times 10^{-3}$ ,  $m=0.9032$ ,  $R_o=0.9$ ,  $a=200$ ,  $b=0.0248$ ,  $x=0.1$ ,  $T_{LC}=100$ 이라 하자.

### 1) 일정 테스트 노력

$d = e^{bT_{LC}} = 11.9417$ ,  $R(x|T) = 0.3280$ ,  $R_o^d = 0.9^{11.9417} = 0.2842$ 이므로 조건 1)에 해당된다. 이 때  $T_1 = 95.12$ 가 된다.

### 2) 웨이블 테스트 노력

$$g = e^{-\alpha \gamma (1 - e^{-\beta T_{LC}})} = 0.0679, \quad R_o^g = 0.2119,$$

$R(x|0) = e^{-a(1 - e^{-\alpha \gamma (1 - e^{-\beta x^m})})} = 0.2995$ 이므로, 조건 1)에 해당된다. 이 때  $T_2 = 49.55$ 이다.

## V. 결론

일정 테스트 노력 곡선인 경우, 그 조건은  $R(x|0) > R_o$ ,  $R_o > R(x|0) > R_o^d$ ,  $R(x|0) < R_o^d$ 이다. 이 중에서 이상적인 경우는  $R_o > R(x|0) > R_o^d$ 인 경우이다. 이 범위의 조건일 때에 최적 발행시각을 결정하고자 하는 본 연구의 목적에 부합된다.

웨이블곡선인 경우를 고찰해보면  $R(x|0) \geq R_o$ ,  $R_o > R(x|0) > R_o^g$ ,  $R(x|0) < R_o^g$ 이다. 이 중에서 이상적인 경우는  $R_o > R(x|0) > R_o^g$ 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.

## ■ 참고 문헌 ■

- [1] C. V. Ramamoorthy, F. B. Bastani, "Software reliability - Status and perspectives", IEEE Trans. on Software Eng., Vol.SE-8, pp.354-371, 1982 August
- [2] J. D. Musa, A. Iannino, K. Okumoto, "Software Reliability : Measurement, Prediction, Application", pp.230-238, 1987 March
- [3] S. Yamada, H. Ohtera, H. Narihisa, "Software reliability growth models with testing- efforts", IEEE Trans. on Reliability, Vol.R-35, pp.19-23, 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, pp.8-16, 1985 April