

Journal of The Korea Knowledge Information Technology Society

ISSN 1975-7700

<http://www.kkits.or.kr>

The Comparative Analysis of Software Failure Time Based on Software Reliability Model and Nonlinear Regression Model

Tae-Jin Yang*

Industry-Academic Cooperation Foundation, Namseoul University

ABSTRACT

Software reliability in the software development process is an important issue. Software process improvement helps in finishing with reliable software product. Infinite failure NHPP software reliability models presented in the literature exhibit either constant, monotonic increasing or monotonic decreasing failure occurrence rates per fault. Software reliability in the final stage of the development process and the actual use test the software present in the phase failure fault number and effective time can be evaluated by the condition of the evaluation technique is important. This process can be seen as a growth process software. In this study, software managers and software failures cause a scan tool that can be utilized in the traditional software model of the Rayleigh distribution model and nonlinear regression models to compare weight model and the log-linear model was studied. As a result, the weighted regression model showed relatively efficient. Algorithm to estimate the parameters used to maximum likelihood estimator and bisection method, model selection based on mean square error (MSE) and coefficient of determination(R^2). In this study, the proposed non-linear regression model is also efficient in terms of reliability because it (the coefficient of determination is 80% or more) in the field of the conventional model can be used as an alternative could be confirmed. Through this study, software developers are considered by many to mean value of the function software failure mode identifying prior knowledge of how much fodder shall be able to help.

© 2014 KKITS All rights reserved

KEYWORDS : Nonlinear regression model, Weight Model, NHPP, Rayleigh Distributions

ARTICLE INFO: Received 5 November 2014, Revised 12 December 2014, Accepted 12 December 2014.

*Corresponding author is with the Department of Industrial & Management Engineering, Namseoul University, 91 Daehak-ro Seonghwan-eup Sebuk-gu

Cheonan-si Chungcheongnam-do, 331-707,
E-mail address: kim1458@nsu.ac.kr

1. 서론

소프트웨어 신뢰성은 일정한 환경조건에서 일정 기간동안 고장이 나지 않고 운영 할 수 있는 확률이다. 따라서 이러한 소프트웨어 신뢰성은 시스템 신뢰도에 영향을 주는 중요한 요소가 되고 디자인 속성 측면에서는 하드웨어 신뢰성과는 다른 면을 가지고 있다.

소프트웨어의 다양한 기능은 소프트웨어 신뢰성 문제들에 관한 주요한 요인이 된다. 소프트웨어의 신뢰도의 일반적인 정의는 일정한 기간 동안 주어진 환경 하에서 컴퓨터 프로그램을 고장 없이 사용할 수 있는 확률을 의미한다. 결국 소프트웨어 개발 과정에서 소프트웨어 신뢰성은 중요한 문제이다. 이 문제는 사용자의 요구조건과 테스트 비용을 만족시켜야 한다. 소프트웨어 테스트(디버깅)면에서 비용을 줄이기 위해서는 소프트웨어의 신뢰성의 변동과 테스트 비용을 사전에 알고 있어야 효율적이다. 따라서 신뢰도, 비용 및 방출 시간의 고려사항을 가진 소프트웨어 개발 과정은 필수 불가결 하다. 지금까지 많은 소프트웨어 신뢰성 모형이 제안 되었다. 이 중에서 비동질적 포아송 과정(Non-homogeneous Poisson process; NHPP)에 의존한 모형[1]은 여러 탐색 과정측면에서는 우수한 모형이고 이러한 모형은 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

이 분야에서 Gokhale과Trivedi [2]은 고양된 비동질적인 포아송 과정 모형(Enhanced NHPP) 모형을 제시하였고 Goel 과 Okumoto [3]은 결함의 누적수가 S 형태나 지수적 형태(S-shaped or exponential-shaped)를 가진 평균값 함수(Mean value function)를 이용한 지수적 소프트웨어 신뢰성 모형(Exponential software reliability growth model)을 제안 하였다. 이모형에 의존한 일반화 모

형은 Yamada 와 Ohba [4]에 의해 지연된 S-형태 신뢰 성장모형(Delayed S-shaped reliability growth model)과 변곡된 S-형태 신뢰성장모형(Inflection S-shaped reliability growth model)이 제안되었다. Zhao [5]는 소프트웨어 신뢰도에서 변환점 문제를 제시하였고 Shyur [6]는 변환점을 이용한 일반화한 신뢰도 성장 모형을 제안하였다.

Pham와 Zhang[7]는 테스트 커버리지(Coverage)를 측정하여 소프트웨어 안정도를 평가 할 수 있는 소프트웨어 안정도 모형을 제시했다. 비교적 최근에, Huang [8]은 일반화 로지스틱 테스트 노력함수(Generalized logistic testing-effort function)와 변환점 모수(Change-point parameter)를 통합하여 효율적인 소프트웨어 신뢰성 예측 기술을 제시하기도 하였다. 그리고 최근에는 S-형태 모형은 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 익숙해지는 학습 과정을 설명할 수 있다고 하였다[9]

일반적으로 고장시간을 측정하다가 시간 절단이 될 경우(시간 절단모형) 대부분의 연구가 소프트웨어 신뢰성모형에만 국한되었는데 본 논문에서는 소프트웨어 고장수와 고장시간을 독립변수와 종속변수로 가정하여 회귀모형을 적용하고자 한다.

따라서 본 연구에서는 소프트웨어 관리자들이 소프트웨어 고장원인 및 검사도구에 활용 할 수 있는 기존의 소프트웨어 NHPP 신뢰성 모형인 레일리분포 모형과 독립변수와 종속변수를 이용한 비선형회귀모형인 가중치모형과 로그선형모형을 비교분석 하고자 한다.

2. 관련 연구

2.1 유한고장 NHPP 신뢰성모형

$N(t)$ 을 시간 t 까지 검출된 소프트웨어의 누적고장수라고 하고, $m(t)$ 를 이에 대한 기대값을 나타내

는 평균값 함수(Mean Value Function)로 가정하고 $\lambda(t)$ 을 강도함수(Intensity function) (즉, t 에서의 순간 결함 검출율)이면 비동질 포아송과정(NHPP)은 누적 고장수인 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률밀도함수(Probability density function)로 알려져 있다. 즉,

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, \quad n = 0, 1, 2, \dots, \infty \quad (1)$$

따라서 NHPP 모형에서 평균값함수 $m(t)$ 와 강도함수 $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다 [2].

$$m(t) = \int_0^t \lambda(s) ds, \quad \frac{dm(t)}{dt} = \lambda(t) \quad (2)$$

유한고장 NHPP모형에서는 시간 $(0, t]$ 까지 탐색되어 질 수 있는 결함의 기대 값을 θ 라고 표현하면 유한고장 NHPP모형의 평균값 함수와 강도함수는 다음과 같이 표현할 수 있다[10].

$$m(t) = \theta F(t), \quad \lambda(t) = \theta F'(t) \quad (3)$$

시간 $(0, t]$ 까지 조사하기 위한 시간절단(Time truncated)모형은 n 번째까지 고장시점 자료를 이라고 하면 n 번째까지 고장시점이 관찰된 고장절단 모형일 경우에 데이터 집합 D_{x_n} 은 $\{x_1, x_2, \dots, x_n\}$ 으로 구성되며, 이 시간절단 모형에서 θ 을 모수공간이라고 표시하면 유한고장 NHPP모형의 우도함수는 다음과 같이 알려져 있다[1,10].

$$x_n = \sum_{i=1}^n t_i \quad (i = 1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (4)$$

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp[-m(x_n)] \quad (5)$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n]$

2.2 곡선 회귀모형

일반적으로 두 개의 변수 사이의 관계를 방정식으로 나타내는 경우, 이 방정식을 회귀 방정식이라 한다. 설명변수 x 와 종속변수 y 와의 간에 직선 관계가 가능한 경우에 그 관계를 나타내는 선형모형을 회귀식이라 부른다.

$$Y_i = b_0 + b_1 x_i + \varepsilon_i, \quad (i = 1, 2, \dots, n) \quad (6)$$

b_0, b_1 는 회귀계수라고 하고 오차 항 ε_i 는 다음과 같은 가정을 만족해야 한다[11]

모든 i 에 관해서 독립 즉, 어떤 $Y_i (i = 1, 2, \dots, n)$ 값을 관찰하는데 발생하는 오차는 다른 Y_i 값을 관찰 할 때의 오차와 어떤 상관관계도 가지지 않는다. 수식으로 표현하면 공분산이 된다.

$$Cov(\varepsilon_i, \varepsilon_j) = 0, \quad i \neq j, \quad i, j = 1, 2, \dots, n \quad (7)$$

그리고 동시에 같은 분포에 따르는 확률변수이고 기대 값이 0이고 분산은 σ^2 인 정규분포를 가진다. 즉, 오차 항 ε_i 은 확률변수로서 모든 i 에 대하여 $E(\varepsilon_i) = 0$ 이고 분산은 x_i 의 값에 상관없이 항상 일정한 상수인 σ^2 을 가진다.

$$\varepsilon_i \sim N(0, \sigma^2) \quad (i = 1, 2, \dots, n) \quad (8)$$

그러나 선형이 아닌 비선형인 경우는 로그를 취하여 선형으로 변환하여 사용하거나 예를 들면 로그 선형모형은 다음과 같은 모형을 취할 수 있다

[12].

$$\hat{Y}_i = \beta_0 + \beta_1 \ln x_i \quad (i=1, 2, \dots, n) \quad (9)$$

즉, 비선형회귀모형은 오차항의 분산이 일정한 상수를 가지지 않고 $Var(\varepsilon_i) = \sigma_i^2$ 을 따르게 된다. 이 경우에 또 하나의 방법으로 다음과 같이 가중치를 이용할 수도 있다.

$$Var(\varepsilon_i) = \sigma_i^2 = W_i \sigma^2 \quad (10)$$

여기서 W_i 는 가중치를 의미한다.

이러한 경우에 회귀추정선은 최소자승법을 이용하면 다음과 같이 가중회귀계수가 유도된다. 즉, 기울기와 절편은 다음과 같이 알려져 있다[11, 12].

$$\beta_1 = \frac{\sum_{i=1}^n W_i (x_i - \bar{x}) (y_i - \bar{y})}{\sum_{i=1}^n W_i (x_i - \bar{x})^2} \quad (11)$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad (12)$$

단, β_0 와 β_1 는 각각 b_0 와 b_1 의 추정치이고 \bar{x} 와 \bar{y} 는 다음과 같다.

$$\bar{x} = \frac{\sum_{i=1}^n W_i x_i}{\sum_{i=1}^n W_i}, \quad \bar{y} = \frac{\sum_{i=1}^n W_i y_i}{\sum_{i=1}^n W_i} \quad (13)$$

이러한 이유로 본 연구에서는 가중치를 $1/x_i$ 이 용한 모형을 제시하여 다른 모형과 비교분석하고자 한다[13]. 즉, $Var(\varepsilon_i) = \sigma_i^2 = \frac{1}{x_i} \sigma^2$. 본 논문에서 적용할 소프트웨어 신뢰모형과 비선형회귀모형에 대한 평균값함수는 <표 1>에 요약되었다.

표 1. 평균값 함수

Table 1. Mean value function		
Type	Model	Mean value function
Software reliability	Rayleigh	$\hat{m}(x_i) = \theta [1 - \exp(-\beta x_i^2)]$
	Log-linear	$\hat{Y}_i = \beta_0 + \beta_1 \ln x_i$
		$\hat{Y}_i = \beta_0 + \beta_1 x_i$
	Nonlinear regression	Note. $W_i = \frac{1}{x_i}, \beta_0 = \bar{y} - \beta_1 \bar{x}$
	Weight	$\beta_1 = \frac{\sum_{i=1}^n W_i (x_i - \bar{x}) (y_i - \bar{y})}{\sum_{i=1}^n W_i (x_i - \bar{x})^2}$

Note. $x_n = \sum_{i=1}^n t_i \quad (i=1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n)$

2.3 Rayleigh 분포를 이용한 유한고장 NHPP 모형

레이리분포(Rayleigh distribution)는 와이분포(Weibull distribution)에서 형상모수가 2를 가진 특수한 경우이다. 레이리분포는 수명시험 및 신뢰성 이론에 적합한 모델로도 알려져 있다. 이러한 레이리분포의 확률밀도함수와 분포함수는 다음과 같이 알려져 있다[13].

$$f(t) = 2\beta t \exp(-\beta t^2) \quad (14)$$

$$F(t) = 1 - \exp(-\beta t^2) \quad (15)$$

단, $\beta > 0, t \in [0, \infty)$

따라서 (3)식을 이용하면 유한고장 NHPP 강도함수와 평균값함수는 다음과 같이 표현할 수 있다.

$$\lambda(t) = \theta F'(t) = 2\theta \beta t \exp(-\beta t^2) \quad (\theta, \beta > 0) \quad (16)$$

$$m(t) = \theta F(t) = \theta [1 - \exp(-\beta t^2)] \quad (17)$$

이경우의 우도함수는 (5)식에 (16)식과 (17)식을 대입하면 다음과 같다.

$$L_{NHPP}(\theta | \underline{x}) = \left[\prod_{i=1}^n 2\theta b x_i e^{-\beta x_i^2} \right] \exp[-\theta(1 - e^{-\beta x_n^2})] \quad (18)$$

따라서 다음과 같은 식을 만족하는 최우추정치(Maximum Likelihood Estimator) $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 을 수치 해석적 방법으로 계산할 수 있다[10].

$$\hat{\theta} = \frac{n}{1 - e^{-bx_n^2}} \quad (18)$$

$$\frac{n}{b} = \sum_{i=1}^n x_i^2 + \theta x_n^2 e^{-bx_n^2} \quad (19)$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$

2.4 최적모형의 결정기준

자료의 예측 값이 실제 값과 얼마나 오차가 있는지를 판단하는 척도는 우수한 예측기법에 대한 판단기준이 된다[9, 10]. 이러한 기준은 일반적으로 결정계수(Coefficient of determination) R^2 , MSE (Mean squared error)등이 가능하다.

결정계수는 회귀직선에 의한 제공 합을 SSR , 오차제공 합을 SSE , 총 제공 합을 SST 라고 하면 다음과 같이 정의 된다.

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \quad (20)$$

$$\text{단, } SST = \sum_{i=1}^n (y_i - \bar{y})^2, \quad SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2, \quad \hat{y}_i \text{ 는 } y \text{ 의 추정치}$$

따라서 R^2 가 크면 설명력이 상대적으로 높기 때문에 효율적 모형이 된다.

효율적 모형을 판단하기 위한 또 하나의 기준으로 평균 제곱 합 MSE 는 다음과 같이 정의 된다.

$$MSE = \frac{\sum_{i=1}^n [\hat{y}_i - m(x_i)]^2}{n - k} \quad (21)$$

단, $m(x_i) = y_i$ 은 시간 $(0, x_i]$ 까지 나타난 고장들의 누적함수를 의미하고 \hat{y}_i 는 x_i 시점까지 평균값 함수로부터 추정된 고장의 누적개수를 의미한다. 그리고 n 은 관찰 값의 수이고 k 는 모수의 수를 의미한다[10].

이 기준은 자료의 예측 값이 실제 값과 얼마나 오차가 있는지를 판단하는 척도가 되기 때문에 MSE 의 값을 최소로 하는 모형이 최적이다.

3. 소프트웨어 고장자료 분석

이 절에서는 소프트웨어 고장시간 자료[14] (Failure interval time data)를 이용하여 본 논문에서 제시하는 소프트웨어 신뢰모형과 비선형 회귀모형을 이용하여 소프트웨어 고장자료를 분석하고자 한다. 이 자료의 고장시간은 18.735 시간단위에 30번의 고장이 발생된 자료이며 <표 1>에 나열되어 있다.

또한 제시하는 모형들을 분석하기 위하여 우선 자료에 대한 추세검정이 선행되어야 한다 [10,15].

표 2. 소프트웨어 고장 시간 자료
Table 2. software failure time data

Failure number	Failure time (hours)	Failure time × 10 (hours)
1	0.479	4.79
2	0.745	7.45
3	1.022	10.22
4	1.576	15.76
5	2.610	26.10
6	3.559	35.59
7	4.252	42.52
8	4.849	48.49
9	4.966	49.66
10	5.136	51.36
11	5.253	52.53
12	6.527	65.27
13	6.996	69.96
14	8.170	81.70
15	8.863	88.63
16	10.771	107.71
17	10.906	109.06
18	11.183	111.83
19	11.779	117.79
20	12.536	125.36
21	12.973	129.73
22	15.203	152.03
23	15.640	156.40
24	15.980	159.80
25	16.385	163.85
26	16.960	169.60
27	17.237	172.37
28	17.600	176.00
29	18.122	181.22
30	18.735	187.35

추세분석에는 일반적으로 라플라스 추세검정(Laplace trend test)을 사용한다[10]. 이 검정을 실시한 결과 <그림 1>에서 라플라스 추세검정의 결과는 라플라스 요인(Factor)이 -2와 2사이에 존재함으로써 즉, 극단값(Extreme value)이 존재하지 않으므로 이 자료를 이용하여 신뢰성 보장을 위한 신뢰성장모형을 제시하는 것이 효율적임을 시사하고 있다[16]. 소프트웨어 신뢰모형에 대한 모수추정은 최우추정법을 이용하고 모수추정을 용이하게 하기 위하여 원래의 고장시간 데이터를 변수변환(Failure time × 10)하여 적용하였다. 비선형 방정식의 계산방법은 수치 해석적 기본방법인 이분법(Bisection method)을 사용하였다. 그리고 이러한 계산은 초기 값을 10^{-6} 과 3을, 허용한계(Tolerance for width of interval)는 10^{-5} 을 주고 수렴성을 확인하면서 충분한 수렴반복횟수인 100번을 C-언어를 이용하여 모수추정을 수행하였다.

표 3. 모형에 대한 모수 추정값
Table 3. Estimation of each model

Type	Model	MLE
Software Reliability	Rayleigh	$\hat{\theta} = 35.305$ $\hat{\beta} = 5.4 \times 10^{-5}$
Nonlinear regression	Log linear	$\hat{\beta}_0 = -18.712$ $\hat{\beta}_1 = 8.086$
	Weight	$\hat{\beta}_0 = 0.929$ $\hat{\beta}_1 = 0.152$

Note. MLE : Maximum Likelihood Estimator

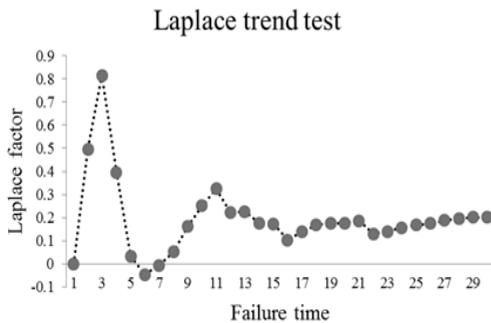


그림 1. 라플라스 추세검정
Figure 1. Laplace trend test

비선형회귀모형에 대한 회귀계수 추정은 통계 패키지 SPSS 18버전을 이용하였다. 그 결과는 <표 3>에 나열 되어 있다. 그리고 MSE(평균제곱오차)와 R²(결정계수)가 <표 4>에 요약되었다. 이 표에서 가중치회귀모형이 소프트웨어 NHPP 신뢰모형인 레일리모형과 비선형 회귀모형인 로그선형 회귀모형보다 실제 값과 예측 값에 대한 차이를 측정하는 평균제곱오차가 제일 작고 예측 값의 차이에

대한 설명력을 의미하는 설명력(결정계수)도 제일 높게 나타나기 때문에 상대적으로 가중치 회귀모형이 효율적인 모형으로 간주할 수 있다.

표 4. 모형에 대한 R^2 와 MSE 추정값

Table 4. R^2 and MSE of each model

Model	R^2	MSE
Rayleigh	0.88	9.4979
Log linear	0.84	10.5051
Weight	0.99	0.8753

Note. R^2 : Coefficient of determination
 MSE : Mean squared error

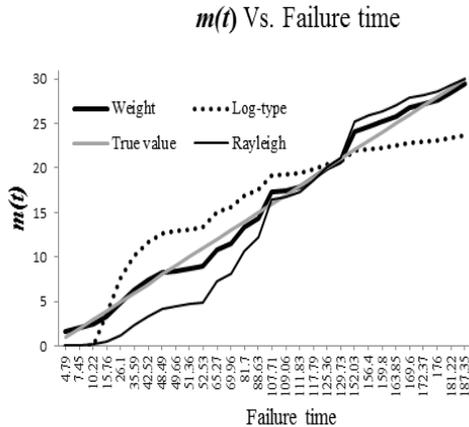


그림 2. 각 모형에 대한 평균값함수 추정

Figure 2. Estimation of mean value function for each model

<그림 2>는 각 모형에 대한 평균값함수 패턴에 대한 정보가 요약되었다. 이 그림에서 로그선형 회귀모형은 참값과의 차이에서 과대평가 추정이 이루어졌고 레일리모형은 과소평가 추정되었다. 그러나 가중치회귀모형은 다른 모형에 비해 다소 차이는 있지만 비교적 합리적으로 추정되었다. 따라서 소프트웨어 개발 시 소프트웨어 고장현상을 사전정보로 활용하면 소프트웨어 고장품질을 향상시킬 수 있다.

4. 결 론

대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 수 없는 상황이 현실이다.

소프트웨어 신뢰성은 개발의 최종단계에 있는 테스트 공정이나 실제 사용단계에 있어서 소프트웨어 내에 존재하는 고장 수나 고장발생시간에 의해서 효과적으로 평가할 수 있는 상황으로 그 평가기술이 중요하게 된다. 따라서 소프트웨어 개발의 테스트공정이나 실제사용단계에 있어서 고장발생환경이나 고장발생현상을 수리적으로 모형화가 가능하면 평가를 할 수 있다. 테스트시간이나 혹은 실행시간, 발생된 고장 수와 고장발생시간과의 관계를 효율적으로 관리함으로써 소프트웨어 신뢰도를 성장 시킬 수 있다.

따라서 본 연구에서는 소프트웨어 관리자들이 소프트웨어 고장원인 및 검사도구에 활용할 수 있는 기존의 소프트웨어 NHPP 신뢰성 모형인 레일리분포 모형과 독립변수와 종속변수를 이용한 비선형회귀모형인 가중치모형과 로그선형모형을 비교연구 하였다. 지금까지 이 분야에서는 수명분포를 이용한 소프트웨어 신뢰성 모형이 주된 대상이었지만 본 연구에서는 비선형회귀모형을 제시 비교하였다. 그 결과 소프트웨어 NHPP 신뢰성 레일리모형보다 비선형회귀모형인 가중치회귀모형이 상대적으로 효율적 모형으로 나타났다.

따라서 본 연구에 제안된 비선형회귀모형도 신뢰성 측면에서 효율적이기 때문에(결정계수가 80% 이상) 이 분야에서 기존모형의 하나의 대안으로 사용할 수 있음을 확인 할 수 있었다. 따라서 소프트웨어 개발 공정에서 개발되는 제품의 품질변동에 이상이 있음에도 불구하고 어떠한 조치나 개선 없이 그 공정을 계속해서 가동시킨다면 그 제품의 품질은 크게 떨어질 것이다. 따라서 본 논문은 소

소프트웨어 시스템 설계와 관련 업종에 종사자들로 하여금 시간 절단된 자료를 미리 예측하여 분석할 수 있으므로 시간 절단된 자료에 대하여 회귀모형을 이용한 예측방법도 사전 정보로 활용하면 소프트웨어 품질의 개선에 도움이 될 수 있으리라 사료된다. 그 결과 최적방출시간은 운용단계에서 결합비용을 줄이고 유지기간을 증가시키는 시점을 만족해야 한다. 이 연구를 통하여 소프트웨어 개발자들은 최적방출시간과 경제적 개발 비용을 파악 하는데 도움을 줄 수 있으리라 사료 된다. 향후 이러한 사전 정보 고장 환경을 이용한 베이저안적 접근 방법이나 수리적인 추정과 검정 부분에 대한 수리적인 접근이 기대 된다.

References

- [1] L. Kuo, and T. Y. Yang, *Bayesian computation of software reliability*, Journal of the American Statistical Association, Vol. 91, pp. 763-773, 1996.
- [2] Gokhale, S. S, and Trivedi, K. S, *A time/structure based software reliability model*, Annals of Software Engineering. 8, pp. 85-121. 1999.
- [3] Goel A L, Okumoto K, *Time-dependent fault detection rate model for software and other performance measure*, IEEE Trans. Reliab. 28, pp. 206-11, 1978.
- [4] Yamada S, Ohba H., *S-shaped software reliability modeling for software error detection*, IEEE Trans. Reliab, 32, pp. 475-484, 1983.
- [5] Zhao M., *Change-point problems in software and hardware reliability*, Commun. Stat. Theory Methods, 22(3), pp.757-768, 1993.
- [6] Shyr H-J., *A stochastic software reliability model with imperfect debugging and change-point*, J. Syst. Software 66, pp. 135-141, 2003.
- [7] Pham H, Zhang X., *NHPP software reliability and cost models with testing coverage*, Eur. J. Oper. Res, 145, pp. 445-454, 2003.
- [8] Huang C-Y, *Performance analysis of software reliability growth models with testing-effort and change-point*, Journal of Systems and Software 76, pp. 181-194, 2005.
- [9] Kuei-Chen. C, Yeu-Shiang. H, and Tzai-Zang. L, *A study of software reliability growth from the perspective of learning effects*, Reliability Engineering and System Safety 93, pp. 1410-1421, 2008.
- [10] Hee-Cheul KIM, *The comparative study of NHPP half-logistic distribution software reliability model using the perspective of learning effects*, Journal of Next Generation Information Technology, Vol. 4, No. 8, pp. 132-139, 2013.
- [11] Hee-Cheul KIM, and Hyun-Cheul Shin, *The study for software future forecasting failure time using curve regression analysis*, Jouranal of Information and Security, Vol. 12, No. 3, pp. 115-121, 2011.
- [12] Kwang-Yoon Song, and In-Hong Chang, *Parameter estimation and prediction for NHPP software reliability model and time series regression in software failure data*, J. Chosun Natural Sci. Vol. 7, No. 1, pp. 67-73, 2014.
- [13] SIMOS M., and GEORGE I., *Tests of fit for the rayleigh distribution based on the empirical laplace transform*, Ann. Inst. Statist. Math, Vol. 55, No. 1, pp. 137-151, 2003.
- [14] Y. HAYAKAWA, and G. TELFAR, *Mixed poisson-type processes with application in software reliability*, Mathematical and Computer Modelling, 31, pp. 151-156, 2000.
- [15] Tae-Jin Yang, and Hee-Cheul Kim, *A comparative study on infinite NHPP cost model based on*

intensity function of log power form, Journal of The Korea Knowledge Information Technology Society, Vol. 9, No. 2, pp. 295-302, 2014.

- [16] K. Kanoun, and J. C. Laprie, *Handbook of software reliability engineering*, M.R.Lyu, editor, chapter trend analysis, McGraw-Hill New York, NY, pp. 401-437, 1996.

소프트웨어 신뢰모형과 비선형 회귀모형에 근거한 소프트웨어 고장시간에 대한 비교 분석

양태진

남서울대학교 산학협력단

요 약

소프트웨어 신뢰성은 개발의 최종단계에 있는 테스트공정이나 실제 사용단계에 있어서 소프트웨어 내에 존재하는 고장 수와 고장 발생시간에 의해서 효과적으로 평가할 수 있는 상황으로 그 평가기술이 중요하게 된다. 따라서 소프트웨어 개발의 테스트공정이나 실제사용단계에 있어서 고장 발생환경이나 고장 발생현상을 수리적으로 모형화가 가능하면 신뢰성 평가를 할 수 있다. 테스트시간이나 혹은 실행시간, 발생된 고장 수와 고장 발생시간 과의 관계를 효율적으로 관리함으로써 소프트웨어 신뢰도를 성장 시킬 수 있다. 이러한 과정을 소프트웨어 성장과정이라고 볼 수 있다. 따라서 본 연구에서는 소프트웨어 관리자등이 소프트웨어 고장원인 및 검사 도구에 활용 할 수 있는 기존의 소프트웨어 모형인 레일리분포 모형과 비선형 회귀모형인 가중치모형과 로그선형모형에 비교 연구하였다. 그 결과 비선형회귀모형인 가중치 회귀모형이 상대적으로 효율적 모형으로 나타났다. 따라서 본 연구에 제안된 비선형회귀모형도 신뢰성측면에서 효율적이기 때문에(결정계수가 80% 이상) 이 분야에서 기존의 모형의 하나의 대안으로 사용할 수 있음을 확

인 할 수 있었다. 이 연구를 통하여 소프트웨어 개발자들은 다양한 평균값함수를 고려함으로써 소프트웨어 고장형태에 대한 사전지식을 파악하는데 도움을 줄 수 있으리라 사료 된다.

감사의 글

이 논문은 2014년도 남서울대학교 학술연구비 지원에 의하여 연구되었음.



Tae Jin Yang received the M.S. degree, and finished the coursework in the Department of Electronic engineering from Hanyang University in 1992 and 1995, respectively. He has

been a professor in the Industry-Academic Cooperation Foundation at Namseoul University since 2014. His current research interests include Fuzzy & Neural-Network application and software reliability. He is a membership of the KKITS.

E-mail address: solomon645@nsu.ac.kr