

# 지수 및 역지수 분포를 이용한 NHPP 소프트웨어 무한고장 신뢰도 모형에 관한 비교연구

김희철\* , 신현철\*\*

## The Comparative Study of NHPP Software Reliability Model Based on Exponential and Inverse Exponential Distribution

Hee-Cheul Kim\* , Hyun-Cheul Shin\*\*

**요 약** 소프트웨어 개발과정에서 소프트웨어 신뢰성은 매우 중요한 이슈이다. 소프트웨어 고장분석을 위한 무한 고장 비동질적인 포아송과정에서 고장발생률이 상수이거나, 단조 증가 또는 단조 감소하는 패턴을 가질 수 있다. 본 논문에서는 소프트웨어 신뢰성에 대한 적용 효율을 나타내는 지수 및 역지수분포를 이용한 신뢰성 모형을 비교 제안한다. 효율적인 모형을 위해 평균제곱오차(MSE), 결정계수( $R^2$ )에 근거한 모델선택, 최우추정법, 이분법에 사용된 파라미터를 평가하기 위한 알고리즘이 적용되었다. 제안하는 지수 및 역지수분포를 이용한 신뢰성 모형을 위해 실제 데이터를 사용한 고장분석이 적용되었다. 고장데이터 분석은 지수 및 역지수분포를 이용한 강도함수와 비교하였다. 데이터 신뢰성을 보장하기 위하여 라플라스 추세검정(Laplace trend test)을 사용하였다. 본 연구에 제안된 역지수분포 신뢰성모형도 신뢰성 측면에서 효율적이기 때문에 (결정계수가 80% 이상) 이 분야에서 기존 모형의 하나의 대안으로 사용할 수 있음을 확인 할 수 있었다. 이 연구를 통하여 소프트웨어 개발자들은 다양한 수명분포를 고려함으로써 소프트웨어 고장형태에 대한 사전지식을 파악하는데 도움을 줄 수 있으리라 사료 된다.

**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. In this paper, we were proposed the reliability model with the exponential and inverse exponential distribution, which made out efficiency application for software reliability. 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( $R^2$ ), for the sake of efficient model, were employed. Analysis of failure, using real data set for the sake of proposing the exponential and inverse exponential distribution, was employed. This analysis of failure data compared with the exponential and inverse exponential distribution property. In order to insurance for the reliability of data, Laplace trend test was employed. In this study, the inverse exponential distribution 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. From this paper, the software developers have to consider life distribution by prior knowledge of the software to identify failure modes which can be able to help.

**Key Words** : Software Reliability, Non-Homogeneous Poisson Process, Inverse Exponential Distribution, Maximum likelihood estimator, Bisection method

\*Corresponding Author : Division of Industrial & Management Engineering, Namseoul University(kim1458@nsu.ac.kr)  
 Received April 10, 2016

Revised April 22, 2016

Accepted April 23, 2016

## 1. 서론

소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리사회에 엄청난 손실을 유발할 수도 있다. 따라서 소프트웨어 개발과정에서 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 방출시간의 고려사항을 가진 소프트웨어 개발과정은 필수 불가결하다. 결국 소프트웨어 제품의 결함내용을 예측하기 위한 모형 개발이 필요하다. 지금까지 많은 소프트웨어 신뢰성모형이 제안되었다. 이 중에서 비동질적 포아송 과정(Non-Homogeneous Poisson Process ; NHPP)에 의존한 모형[1]은 에러 탐색 과정 측면에서는 우수한 모형이고, 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

Gokhale과 Trivedi[1]은 고양된 비동질적인 포아송 과정 모형(Enhanced NHPP) 모형을 제시하였고 Goel과 Okumoto[2]은 지수적 소프트웨어 신뢰성모형(Exponential software reliability growth model)을 제안 하였다. 이 모형은 결함의 누적수가 S-형태나 지수적 형태(S-shaped or exponential-shaped)를 가진 평균값함수(Mean value function)를 이용하였다.

이러한 모형에 의존한 일반화모형은 Yamada와 Ohba[3]에 의해 지연된 S-형태 신뢰성장모형(Delayed S-shaped reliability growth model)과 변곡된 S-형태 신뢰 성장모형(Inflection S-shaped reliability growth model)이 제안되었다. Zhao[4]는 소프트웨어 신뢰도에서 변환점 문제를 제시하였고 Shyur[5]는 변환점을 이용한 일반화한 신뢰도 성장모형을 제안하였다. Pham와 Zhang[6]는 테스트 커버리지(Coverage)를 측정하여 소프트웨어 안정도를 평가 할 수 있는 소프트웨어 안정도 모형을 제시했다.

또한, Huang[7]은 일반화 로지스틱 테스트 노력 함수(Generalized logistic testing-effort function)와 변환점 모수(Change-point parameter)를 통합하여 효율적인 소프트웨어 신뢰성을 예측하는 기술을 제시하기도 하였다. 그리고 최근에는 S-형태 모형은 소프트웨어 관리자들이 소프트웨어 및 검사도구에 익숙해지는 학습과정을 설명할 수 있다고 하였다[8]. 그리고 김희철[9]은 Rayleigh형과 Burr형 NHPP 소프트웨어 신뢰모형에 관한 통계적 공정 관리 접근방법 비교연구 결과 레일리분포모형 보다는 버르분포모형이 상대적으로 효율적 모형임을 확인하였다.

따라서 본 연구는 NHPP 모형에서 수명분포가 지수분포를 따르는 모형과 역지수분포를 따르는 소프트웨어 신뢰도 모형을 비교 분석 하고자 한다.

## 2. 관련연구

### 2.1 NHPP 소프트웨어 신뢰성

$N(t)$ 을 시간  $t$ 까지 검출된 소프트웨어의 누적 고장수라고 하고,  $m(t)$ 를 이에 대한 기대값을 나타내는 평균값 함수(Mean Value Function)로 가정하고  $\lambda(t)$ 을 강도함수(Intensity function) (즉,  $t$ 에서의 순간 결함 검출율)이면 비동질 포아송 과정(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, NHPP모형에서 평균값함수  $m(t)$ 와 강도함수  $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[1][9].

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

이처럼 시간관련모형(Time domain models)들은 NHPP에 의해서 확률고장과정으로 설명이 가

능하다[1]. 이러한 NHPP모형들은 유한고장모형과 무한고장 범주로 분류한다[1]. 유한고장 NHPP모형에서는 시간  $(0, t]$  까지 탐색되어 질 수 있는 결함의 기대값을  $\theta$  라고 표현하면 유한고장 NHPP 모형의 평균값 함수와 강도함수는 다음과 같이 표현할 수 있다[1, 9].

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

반면에 무한고장 NHPP모형들은 수리시점에서 고장이 발생할 상황을 반영하기 위하여 기록 멈춤 통계량(Record breaking statistics)을 사용하는 RVS(Record Value Statistics)모형을 사용할 수 있다고 하였고 이 RVS모형과 NHPP모형에 관해서 평균값함수는 다음과 같이 된다고 하였다 [10].

$$m(t) = -\ln(1 - F(t)) \quad (4)$$

따라서 (2)식과 (4)식을 연관시키고  $f(t)$ 을 확률밀도함수,  $F(t)$ 을 분포함수라고 하면 NHPP의 강도함수는 다음과 같이 위험함수( $h(t)$ )가 된다.

$$\lambda(t) = m'(t) = f(t)/(1 - F(t)) = h(t) \quad (5)$$

시간  $(0, t]$  까지 조사하기 위한 시간절단(Time truncated)모형은  $n$  번째까지 고장시점 자료를

$$x_n = \sum_{i=1}^n t_i \quad (i = 1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (6)$$

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 고장절단모형에서  $\theta$  을 모수공간이라고 표시하면 NHPP모형의 우도함수는 다음과 같이 알려져 있다[1, 9].

$$L_{NHPP}(\theta | \underline{x}) = \left( \prod_{i=1}^n \lambda(x_i) \right) \exp[-m(x_n)] \quad (7)$$

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

### 2.2 기존의 지수분포 NHPP 모형

소프트웨어 신뢰성 분야에서 널리 사용되는 모형 중에서 가장 많이 연구된 모형이 수명분포가 지수분포를 따르는 Goel-Okumoto 모형이다. 이 모형에 대한 평균값함수와 강도함수는 다음과 같이 알려져 있다 [2, 3].

$$m(t|\theta, \beta) = \theta F(t) = \theta(1 - e^{-\beta t}) \quad (\theta > 0, \beta > 0) \quad (8)$$

$$\lambda(t|\theta, \beta) = \theta f(t) = \theta \beta e^{-\beta t} \quad (9)$$

여기서,  $\theta$ 는 결함들(Faults)의 기대값이고  $\beta$ 는 고장발생률을 의미한다.

본 연구에서는 (8)식과 (9)식에서  $t$  을 최종고장시점  $x_n$  으로 대치하고 (7)식에 (8)식과 (9)식을 대입하면 최우추정법(MLE)을 이용하기 위한 로그우도함수를 구하면 다음과 같다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln \theta + n \ln \beta - \beta \sum_{i=1}^n x_i - \theta(1 - e^{-\beta x_n}) \quad (10)$$

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

따라서 고장절단모형에서의 모수  $\theta$  와  $\beta$  에 관한 편미분식은 다음과 같이 유도 할 수 있다 [2].

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-\beta x_n} = 0 \quad (11)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^n x_i - \theta x_n e^{-\beta x_n} = 0 \quad (12)$$

따라서 각 모수에 대한 최우추정량  $\hat{\theta}_{MLE}$  와  $\hat{\beta}_{MLE}$  은 다음 식을 만족한다.

$$\frac{n}{\theta} = 1 - e^{-\beta x_n} \quad (13)$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^n x_i + \hat{\theta} \hat{\beta} x_n e^{-\hat{\beta} x_n} \quad (14)$$

NHPP 모형에서 테스트 시점  $x_n$  에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간  $(x_n, x_n + \tau]$  (단,  $\tau$  는 임무시간(Mission time)) 사이에서 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도(Reliability)  $\hat{R}(\tau | x_n)$  는 다음과 같이 됨이 알려져 있다[1, 11].

$$\begin{aligned} \hat{R}(\tau | x_n) &= e^{-\int_{x_n}^{x_n + \tau} \lambda(x) dx} \\ &= \exp[-\{m(\tau + x_n) - m(x_n)\}] \end{aligned} \quad (15)$$

따라서 Goel-Okumoto 모형의 신뢰도는 다음과 같이 유도 할 수 있다.

$$\hat{R}(\tau | x_n) = \exp[-\theta(1 - e^{-\beta(\tau + t)}) + \theta(1 - e^{-\beta t})] \quad (16)$$

### 2.3 역지수분포

의학 및 신뢰성과 생태학 분야에서 폭넓게 적용 할 수 있는 분포가 역와이블 분포(inverse Weibull distribution)[13]이다. 이러한 분포의 분포함수는 다음과 같이 알려져 있다.

$$F(t) = \exp[-(\beta t)^{-\gamma}] \quad (17)$$

여기서 형상모수  $\gamma=1$  인 경우가 역지수분포(inverse exponential distribution)가 된다고 알려져 있다.

따라서 역지수분포의 분포함수와 확률밀도함수는 다음과 같다.

$$F(t) = \exp[-(\beta t)^{-1}] \quad (18)$$

$$f(t) = F'(t) = \beta^{-1} t^{-2} \exp[-(\beta t)^{-1}] \quad (19)$$

### 2.4 효율적인 모형 선택기준

최근에 모형에 대한 효율성을 조사하기 위한 기준으로서 평균제곱오차(MSE)와 결정계수( $R^2$ )를 사용한다[10, 11]. 평균제곱오차(mean square error)는 실제 관찰 값과 예측 값에 대한 차이를 측정하는 도구로서 다음과 같이 정의 된다.

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n - k}$$

단,  $m(x_i)$ 은 시간(0,  $x_i$ ]까지 나타난 에러들의 누적분포함수를 의미하고  $\hat{m}(x_i)$ 는  $x_i$  시점까지 평균값 함수로부터 추정된 에러의 누적개수를 의미한다. 그리고  $n$  은 관찰 값의 수이고  $k$ 는 모수의 수를 의미한다. 즉, MSE의 값이 작으면 효율적 모형이 된다. 그리고 결정계수( $R^2$ )는 관찰 값의 차이에 대한 설명력을 나타내는 도구로서 다음과 같이 정의 된다.

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left( m(x_i) - \frac{\sum_{j=1}^n m(x_j)}{n} \right)^2}$$

따라서  $R^2$ 의 값이 클수록 설명력이 커져서 효율적 모형으로 간주된다.

## 3. 제안된 역지수분포 NHPP

### 유한고장모형

역지수분포의 강도함수와 평균값함수는 다음과 같이 유도할 수 있다.

$$m(t) = \theta F(t) = \theta e^{-(\beta t)^{-1}} \quad (20)$$

$$\lambda(t) = \theta f(t) = \theta \beta^{-1} t^{-2} \exp[-(\beta t)^{-1}] \quad (21)$$

이경우의 유도함수는 (7)식에 (20)식과 (21)식

을 대입하면 다음과 같이 유도 할 수 있다.

$$L_{NHPP}(\theta | \underline{x}) = \left[ \prod_{i=1}^n \theta \beta^{-1} x_i^{-2} \exp[-(\beta x_i)^{-1}] \right] \exp[-\theta e^{-(\beta x_n)^{-1}}] \quad (22)$$

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

모수 추정방법은 최우추정법(Maximum likelihood estimation; MLE)을 사용하였고 최우추정법을 이용하기 위한 로그우도함수는 (22)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln \theta - n \ln \beta - 2 \sum_{i=1}^n x_i - \sum_{i=1}^n (\beta x_i)^{-1} - \theta e^{-(\beta x_n)^{-1}} \quad (23)$$

따라서 고장절단모형에서의 모수  $\hat{\theta}_{MLE}$  와  $\hat{\beta}$  에 관한 편미분식은 다음과 같이 유도 할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - e^{-(\beta x_n)^{-1}} = 0 \quad (24)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \beta} = -\frac{n}{\beta} + \frac{1}{\beta^2} \sum_{i=1}^n x_i - \theta \frac{1}{\beta^2 x_n} e^{-(\beta x_n)^{-1}} = 0 \quad (25)$$

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

또한, 임무시간(Mission time)  $\delta$ 에 따른 신뢰도는 다음과 같이 유도 된다[1, 11].

$$\hat{R}(\delta | x_n) = \exp[-m_1(x_n) + m_2(x_n)] \quad (26)$$

단,  $m_1(x_n) = \theta e^{-[\beta(x_n + \delta)]^{-1}}$ ,  $m_2(x_n) = \theta e^{-(\beta t)^{-1}}$

#### 4. 소프트웨어 고장시간 분석

표 1. 소프트웨어의 고장시간자료

Table 1. software failure time data

Failure number	Failure time (hours)	Failure time $\times 10^{-2}$
1	30.02	0.3002
2	31.46	0.3146
3	53.93	0.5393
4	55.290	0.5529
5	58.720	0.5872
6	71.920	0.7192
7	77.070	0.7707
8	80.900	0.809
9	101.90	1.019
10	114.87	1.1487
11	115.34	1.1534
12	121.57	1.2157
13	124.97	1.2497
14	134.07	1.3407
15	136.25	1.3625
16	151.78	1.5178
17	177.50	1.775
18	180.29	1.8029
19	182.21	1.8221
20	186.34	1.8634
21	256.81	2.5681
22	273.88	2.7388
23	277.87	2.7787
24	453.93	4.5393
25	535.00	5.3500
26	537.27	5.3727
27	552.9	5.5290
28	673.68	6.7368
29	704.49	7.0449
30	738.68	7.3868

이 절에서는 소프트웨어 고장 시간자료 [12] (Failure time data)를 이용하여 본 논문에서 제시하는 소프트웨어 신뢰모형들을 이용한 소프트웨어 고장시간을 분석하고자 한다. 이 자료의 고장시간은 738.68 시간단위에 30번의 고장이 발생한 자료이며 [표 1]에 나열 되어 있다.

또한 제시하는 신뢰모형들을 분석하기 위하여 우선 자료에 대한 추세검정이 선행 되어야 한다[11]. 추세분석에는 일반적으로 라플라스 추세검정(Laplace trend test)을 사용한다. 이 검정을 실

시한 결과 [그림 1]에서 라플라스요인(Factor)이 -2와 2사이에 존재함으로써 즉, 극단값(Extreme value)이 존재하지 않으므로 이 자료를 이용하여 신뢰성장모형을 제시하는 것이 효율적임을 시사하고 있다[14].

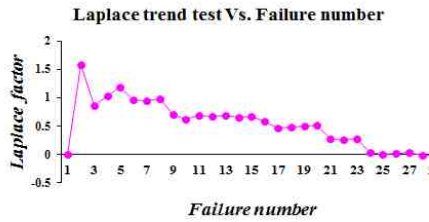


그림 1. 라플라스 추세검정  
Fig. 1. Laplace trend test

표 2. 모수 추정값 및 MSE, R<sup>2</sup>

Table 2. Parameter estimation of the each model and MSE, R<sup>2</sup>

Model	MLE	Model Comparison	
		MSE	R <sup>2</sup>
Goel-Okumoto	$\hat{\theta}_{MLE} = 33.4092$ $\hat{\beta}_{MLE} = 3.089 \times 10^{-1}$	5.8501	0.9370
Inverse exponential distribution	$\hat{\theta}_{MLE} = 31.851$ $\hat{\beta}_{MLE} = 2.2616$	66.0673	0.9061

Note. MLE : Maximum likelihood estimation;  
MSE : Mean square error;  
R<sup>2</sup> : Coefficient of determination

모수추정은 최우추정법을 이용하고 모수 추정을 용이하게 하기 위하여 원래의 고장시간 데이터를 변수변환(Failure time  $\times 10^{-2}$ )하여 적용하였다. 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기 값을 0.001과 5을, 허용 한계(Tolerance for width of interval)는  $10^{-5}$ 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다. 최우추정법의 결과와 모형에 대한 효율성을 조사하기 위한 기준으로서 MSE(평균제곱오차)와 R<sup>2</sup>

(결정계수)가 [표 2]에 요약되었다.

이 표에서 Geol-Okumoto모형이 역지수 모형보다 실제 값과 예측 값에 대한 차이를 측정하는 평균제곱오차가 낮고 예측 값의 차이에 대한 설명력을 의미하는 결정계수도 높게 나타나기 때문에 Geol-Okumoto모형이 역지수 모형보다 효율적인 모형으로 간주할 수 있다. 그러나 둘 모형이 결정계수 값이 80%이상 나타나 이 분야에서 효율적 모형임을 확인 할 수 있다[9].

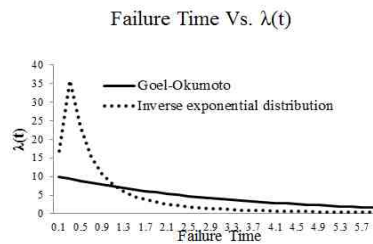


그림 2. 강도함수의 형태

Fig. 2. Pattern of intensity function

[그림 2]은 강도함수에 대한 패턴이 요약 되었다. 이 그림에서 Geol-Okumoto모형은 고장시간이 지남에 따라 완만한 감소형태를 보이고 있지만 역지수 모형은 증가하다가 감소하는 형태를 보이고 있다.

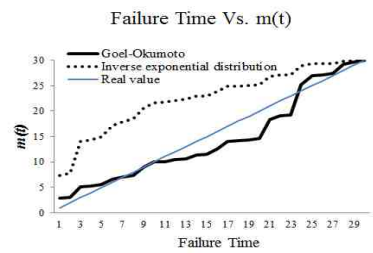


그림 3. 평균값 함수의 형태

Fig. 3. Pattern of mean value function

[그림 3]은 평균값 함수에 대한 패턴이 요약 되었다. 이 그림에서 역지수 모형은 참값과의 차이에 있어서 과대평가 추정이 이루어졌고 Geol-Okumoto모형은 과소평가 추정 되었다.

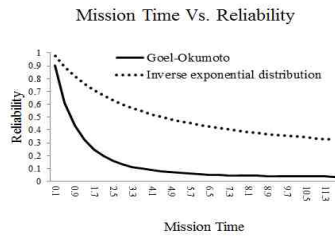


그림 4. 평균값 함수의 형태  
Fig. 4. Pattern of reliability

[그림 4]에서 보여 주듯이 임무시간에 대한 신뢰도 그림에서는 Geol-Okumoto모형과 역지수모형이 비교 했을 때 임무시간이 증가 할수록 Geol-Okumoto모형보다는 역지수모형이 각 시점에서 신뢰도의 상승으로 나타나고 있다.

### 5. 결론

소프트웨어 신뢰성은 개발의 최종단계에 있는 테스트 공정이나 실제 사용단계에 있어서 소프트웨어 내에 존재하는 고장 수나 고장 발생시간에 의해서 효과적으로 평가할 수 있는 상황으로 그 평가 기술이 중요하게 된다. 따라서 소프트웨어 개발의 테스트공정이나 실제사용단계에 있어서 고장 발생 환경이나 고장 발생현상을 수리적으로 모형화가 가능하면 평가를 할 수 있다.

따라서 본 연구에서는 소프트웨어 관리자들이 소프트웨어 고장원인 및 검사 도구에 활용 할 수 있는 수명분포가 다양한 특성을 가지는 지수분포와 역지수분포 모형에 대하여 비교 연구 하였다.

그 결과 지수분포모형이 역지수분포에 비해 실제 값과 예측 값에 대한 차이를 측정하는 평균제곱오차가 낮고 예측 값의 차이에 대한 설명력을 의미하는 결정계수도 제일 높게 나타나기 때문에 역지수분포 모형보다는 지수분포모형이 효율적인 모형으로 간주할 수 있다. 그러나 미래 임무시간에 대한 신뢰도 그림에서는 역지수모형이 지수분포모형보다 임무시간이 증가할수록 신뢰도의 상승으로 나타나고 있다.

따라서 본 연구에 제안된 역지수분포 모형도

신뢰성 측면에서 효율적이기 때문에 (결정계수가 80% 이상) 이 분야에서 기존의 모형의 하나의 대안으로 사용할 수 있음을 확인 할 수 있었다. 이 연구를 통하여 소프트웨어 개발자들은 다양한 수명분포를 고려함으로써 소프트웨어 고장형태에 대한 사전지식을 파악하는데 도움을 줄 수 있으리라 사료 된다.

### REFERENCES

- [1] Gokhale, S. S. and Trivedi, K. S. A, "time/structure based software reliability model", *Annals of Software Engineering*, 8, pp. 85-121, 1999.
- [2] Goel A L, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", *IEEE Trans. Reliab.* 28, pp. 206-11, 1978.
- [3] Yamada S, Ohba H. S-shaped software reliability modeling for software error detection. *IEEE Transaction on Reliability* 3, pp. 475-84, 1983.
- [4] Zhao M. Change-point problems in software and hardware reliability. *Communication Stat Theory Methods* 22(3), pp. 757-68, 1993.
- [5] Shyur H-J. A stochastic software reliability model with imperfect debugging and change-point. *J Syst Software* 66(2), pp. 135-41, 2003.
- [6] Pham H, Zhang X., "NHPP software reliability and cost models with testing coverage", *Eur. J. Oper. Res.*, 145, pp.445-454, 2003.
- [7] Huang C-Y. Performance analysis of software reliability growth models with testing-effort and change-point. *J Syst Software*. 2005; 76:181-94.
- [8] 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.

- [9]Hee-Cheul KIM, "The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Rayleigh and Burr Type", Journal of Korea Society of Digital Industry and Information Management, Volume 10, No.2, pp. 1-11, 2014.
- [10]Kim H-C. The Property of Learning effect based on Delayed Software S-Shaped Reliability Model using Finite NHPP Software Cost Model, Indian Journal of Science and Technology 8(34), pp.1-7, 2015.
- [11] Tae-Hyun Yoo, "The Infinite NHPP Software Reliability Model based on Monotonic Intensity Function", Indian Journal of Science and Technology, Volume 8, No. 14, pp. 1-7, 2015.
- [12] K.H Rao, R. S, Prasad and. R.L.Kantham "Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC", International Journal of Computer Applications (0975 -8887), Volume 21, No.7, pp. 1-5, May 2011.
- [13]Kersey, Jing Xiong, "Weighted Inverse Weibull and Beta-Inverse Weibull Distribution", Electronic Theses & Dissertations Paper 661, pp. 1-53, 2010.
- [14]K. Kanoun and J. C. Laprie, "Handbook of Software Reliability Engineering", MR.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY, pp. 401-437, 1996.

---

저자약력

---

김희철 (Hee-Cheul Kim) [정회원]



- 1992년 2월 : 동국대학교 통계학과 (이학석사)
- 1998년 8월 : 동국대학교 통계학과 (이학박사)
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수

<관심분야>

소프트웨어인쇄성 공학, 전산 통계, 웹프로그래밍

신현철(Hyun-Cheul Shin) [정회원]



- 2002년 2월 : 원광대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 : 한국정보처리학회 이사
- 1994년 3월 ~ 현재 : 백석문화대학교 컴퓨터공학부 교수

<관심분야>

통신공학, 컴퓨터과학, 소프트웨어공학, 소프트웨어공학