

# 비선형 강도함수 특성을 이용한 유한고장 NHPP모형에 근거한 소프트웨어 최적방출시기 비교 연구

김경수\*, 김희철\*\*

백석문화대학교 인터넷 정보학부\*, 남서울대학교 산업경영공학과\*\*

## The Comparative Study of Software Optimal Release Time of Finite NHPP Model Considering Property of Nonlinear Intensity Function

Kyung-Soo Kim\*, Hee-Cheul Kim\*\*

Dept. of Internet information, BaekSeok Culture University\*

Dept. of Industrial & Management Engineering, Namseoul University\*\*

**요약** 본 연구에서는 소프트웨어 제품을 개발하여 테스트를 거친 후 사용자에게 인도하는 시기를 결정하는 방출문제에 대하여 연구 하였다. 소프트웨어의 결함을 제거하거나 수정 작업과정에서 유한고장수를 가진 비동질적인 포아송 과정에 기초하였다. 수명강도는 다양한 형상모수와 척도모수에 이용 할 수 있기 때문에 신뢰성 분야에서 많이 사용되는 비선형 특성을 가진 반-로지스틱 분포 모형을 이용한 방출시기에 관한 문제를 제시하였다. 소프트웨어 요구 신뢰도를 만족시키고 소프트웨어 개발 및 유지 총비용을 최소화 시키는 최적 소프트웨어 방출 정책에 대하여 논의 되었다. 본 논문의 수치적인 예에서는 고장 시간 자료를 적용하였으며 모수추정 방법은 최우추정법을 이용하고 최적 방출시기를 추정하였다. 따라서 소프트웨어 방출시기를 사전 정보로 활용하면 잠재적 보안피해액을 줄 일 수 있다고 판단된다.

**주제어** : 소프트웨어 방출정책, 비동질적 포아송 과정, 반-로지스틱 분포, 임무시간, 강도함수.

**Abstract** In this paper, make a study decision problem called an optimal release policies after testing a software system in development phase and transfer it to the user. When correcting or modifying the software, finite failure non-homogeneous Poisson process model, presented and propose release policies of the life distribution, half-logistic property model which used to an area of reliability because of various shape and scale parameter. In this paper, discuss optimal software release policies which minimize a total average software cost of development and maintenance under the constraint of satisfying a software reliability requirement. In a numerical example, the parameters estimation using maximum likelihood estimation of failure time data, make out estimating software optimal release time. Software release time is used as prior information, potential security damages should be reduced.

**Key Words** : Software Release Policies, NHPP, Half-Logistic Distribution, Mission Time, Intensity Function.

Received 11 July 2013, Revised 13 August 2013

Accepted 20 September 2013

Corresponding Author: Hee-Cheul Kim(Namseoul University)

Email: kim1458@nsu.ac.kr

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1738-1916

## 1. 서론

소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 유발 할 수 도 있다. 따라서 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. 결국 소프트웨어 제품의 결함내용을 예측하기 위한 모형 개발이 필요하다. 지금까지 많은 소프트웨어 신뢰성 모형이 제안 되었다. 이 중에서 비동질적 포아송 과정(non-homogeneous Poisson process; NHPP)에 의존한 모형은 여러 탐색 과정측면에서는 우수한 모형이고 이 모형은 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생 되지 않는다는 가정을 하고 있다.

이 분야에서 Gokhale과Trivedi [1]은 고양된 비동질적인 포아송 과정 모형(enhanced NHPP) 모형을 제시하였고 Goel 과 Okumoto [2]은 지수적 소프트웨어 신뢰성 모형(exponential software reliability growth model)을 제안 하였다. 이 모형은 결함의 누적수가 S 형태나 지수적 형태(S-shaped or exponential-shaped)를 가진 평균값 함수(mean value function)를 이용하였다. 이러한 모형에 의존한 일반화 모형은 Yamada 와 Ohba [3]에 의해 지연된 S 형태 신뢰 성장모형(delayed S-shaped reliability growth model)과 변곡된 S-형태 신뢰성장모형(inflexion S-shaped reliability growth model)이 제안되었다. Zhao [4]는 소프트웨어 신뢰도에서 변환점 문제를 제시하였고 Shyur [5]는 변환점을 이용한 일반화한 신뢰도 성장 모형을 제안하였다. Pham와 Zhang[6]는 테스트 커버리지(coverage)를 측정하여 소프트웨어 안정도를 평가 할 수 있는 소프트웨어 안정도 모형을 제시했다. 비교적 최근에, Huang [7]은 일반화 로지스틱 테스트 노력 함수(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 설

명할 수 있다고 하였고[8] 또한, 대수 선형 위험함수를 이용한 학습과정 특성을 연구하기도 하였다[9].

본 연구에서는 소프트웨어 제품을 개발하여 테스트를 거친 후 사용자에게 인도하는 시기를 결정하는 방출문제에 대하여 연구하였다. 따라서 유한고장수를 가진 비동질적인 포아송 과정에 기초하고 강도함수는 비선형 모형인 반-로지스틱 특성을 이용한 최적 방출시기에 관한 문제를 제시하여 소프트웨어 요구 신뢰도를 만족시키고 소프트웨어 개발 및 유지 총비용을 최소화 시키는 최적 소프트웨어 방출 정책에 대하여 논의되었다. 따라서 소프트웨어 방출시기를 사전 정보로 활용하면 잠재적 보안피해를 줄 일 수 있다고 판단된다.

## 2. 관련연구

### 2.1 유한 고장 NHPP 모형

신뢰도에서 관측시간  $(0, t]$  사이에 발견된 고장 수  $N(t)$ 을 모형 화 하는데 비동질적 포아송 과정이 널리 사용되어 왔다. 이 과정(process)에서 강도함수(intensity function) 혹은 고장 발생률(rate of occurrence of failure; ROCOF)  $\lambda(t) = dE[N(t)]/dt$  은  $t$ 에 대한 단조(monotonic)함수로 흔히 가정한다[1]. 이 범주에서 지금까지 알려진 모형들은 Goel-Okumoto 모형, Weibull 모형 그리고 Cox-Lewis 모형 등이 있는데 이모형들에 대한 강도함수는 각 각 시간에 의존한 함수, 멱(power) 함수, 대수 선형(log-linear) 함수를 가정하였다 [10].

NHPP 모형에서 평균값 함수  $m(t)$  (mean value function)와 강도 함수  $\lambda(t)$ 는 다음과 같은 관계로 표현 할 수 있다.

$$m(t) = \int_0^t \lambda(s) ds, \quad \frac{dm(t)}{dt} = \lambda(t) \quad (1)$$

$N(t)$ 는 모수  $m(t)$ 을 가진 포아송 확률밀도함수(probability density function)로 알려져 있다. 즉,

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

이처럼 시간관련 모형(time domain models)들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다[11]. 유한 고장(finite failure) NHPP 모형들은 충분한 테스트 시간이 주어지면 결함들(faults)의 기대값이 유한 값( $\lim_{t \rightarrow \infty} m(t) = \theta < \infty$ )을 가지고 반면에 무한 고장(infinite failure) NHPP 모형들은 무한 값을 가진다고 가정된다. 유한 고장 NHPP 모형에서 충분한 테스트 시간이 주어졌을 때 탐색되어 질 수 있는 결함의 기대값을  $\theta$ 라고 표현하고  $F(t)$ 를 분포함수라고 표현하면 유한 고장 NHPP모형의 평균값 함수는 다음과 같이 표현할 수 있다[1][10][11].

$$m(t) = \theta F(t) \quad (3)$$

(3)식으로 부터 강도함수(failure intensity) $\lambda(t)$ 는 다음과 같이 유도된다.

$$\lambda(t) = \theta F'(t) \quad (4)$$

또한, 시간  $(0, t]$ 까지 조사하기 위한 시간 절단(time truncated)모형은  $n$ 번째 까지 고장시점 자료를

$$x_k = \sum_{i=1}^k t_i \quad (k=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (5)$$

이라고 하면 데이터 집합  $D_n$ 는  $\{n, x_1, x_2, \dots, x_n; t\}$ 와 같이 구성된다.  $n$ 번째까지 고장시점이 관찰된 고장 절단 모형일 경우에 데이터 집합  $D_{x_n}$ 은  $\{x_1, x_2, \dots, x_n\}$ 으로 구성되며 이 시간 절단 모형에서의  $\theta$ 를 모수공간이라고 표시하면 우도함수는 다음과 같이 알려져 있다[10][11].

$$L_{NHPP}(\theta | \underline{x}) = \left( \prod_{i=1}^n \lambda(x_i) \right) \exp(-m(x_n)) \quad (6)$$

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

## 2.2 효율적인 모형 선택 기준

최근에 모형에 대한 효율성을 조사하기 위한 기준으로서 평균제곱오차(MSE)와 결정계수( $R^2$ )를 사용한다

[9][13].

### 2.2.1 평균 제곱오차

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 누적 분포함수를 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.

### 2.2.2 $R^2$

결정계수( $R^2$ )는 관찰 값의 차이에 대한 설명력을 나타내는 도구로서 다음과 같이 정의 된다.

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

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

## 2.3 신뢰도와 비용최소화를 고려한 방출시간

NHPP 모형에서 테스트 시점  $x_n$ 에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간  $(x_n, x_n + x]$  (단,  $x$ 는 임무시간(mission time))동안 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도  $\hat{R}(x | x_n)$ 는 다음과 같이 됨이 알려져 있다[13] [14].

$$\begin{aligned} \hat{R}(x | x_n) &= \exp\left(-\int_{x_n}^{x_n+x} \lambda(\tau) d\tau\right) \\ &= \exp[-\{m(x+x_n) - m(x_n)\}] \end{aligned} \quad (7)$$

결국 소프트웨어 방출시간  $T_R$ 이 신뢰도

$R_0 = \hat{R}(x | T_R)$  을 확보해야 한다면 다음 방정식을 만족해야한다.

$$\ln \hat{R}_0 = -m(x + T_R) + m(T_R) \tag{8}$$

비용 최소화와 관련된 최적 방출시간은 신뢰도와 함께 비용모형에 의해서 결정된다. 소프트웨어 방출시간을  $T$ 로 표현하고  $m(T)$ 와  $m(T_{LC})$ 을 각각  $(0, T]$ 와  $(0, T_{LC}]$ 의 기간에 발견된 기대 고장수라고 표현하고  $C(T)$ 을 소프트웨어 생명주기(life cycle) 동안에 기대되는 소프트웨어 비용이라고 하면  $C(T)$ 는 다음과 같이 표현 된다 [13][14].

$$C(T) = c_1 m(T) + c_2 [m(T_{LC}) - m(T)] + c_3 T \tag{9}$$

위 식에서  $T_{LC}$ 는 지정된 소프트웨어 수명시간,  $c_1$ 는 테스트 동안에 하나의 고장을 수리하는 비용이고  $c_2$ 가동 중에 하나의 고장을 수리하는 비용( $c_2 > c_1$ ), 그리고  $c_3$ 는 단위 시간당 테스트 비용을 나타낸다[13].

따라서 비용함수를 고려하여 소프트웨어의 모든 수명에서 총비용을 최소화함으로써 최적 테스트 시간을 결정할 수 있고 다음과 같은 식을 만족하면 비용함수  $C(T)$ 는 유일한 값을 가진다[12].

$$\frac{\partial C(T)}{\partial T} = 0, \quad \frac{\partial^2 C(T)}{\partial^2 T} > 0 \tag{10}$$

결국  $T$ 에 관해서 비용함수  $C(T)$ 을 미분하면 최적방출시간  $T_C$ 을 계산 할 수 있다[13][14].

따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적방출시간이다. 그러므로 최적 방출시간  $T_{OP}$ 는  $T_R$ 과  $T_C$ 에 대하여 다음을 만족한다[14].

$$T_{OP} = \text{Max}(T_C, T_R) \tag{11}$$

### 3. 반-로지스틱 분포특성을 이용한 NHPP 소프트웨어 모형

#### 3.1 기본 모형인 Goel-Okumoto 모형

이 분야에서 가장 기본적인 모형은 Goel-Okumoto 모

형[8]이다. 이모형은 결함 당 고장발생 시간의 분포(수명 분포)를 지수분포를 가정하였다[2].

따라서 소프트웨어 결함(fault)당 고장 발생률이 일정한 형태를 가지며 평균값 함수는

$$m(t|\theta, \beta) = \theta(1 - e^{-\beta t}) \quad (\theta > 0, \beta > 0) \tag{12}$$

이라고 알려져 있고 강도함수는  $\lambda(t|\theta, \beta) = \theta e^{-\beta t}$  이라고 알려져 있고  $\theta$ 는 결함들(faults)의 기대 값이고  $\beta$ 는 고장 발생률을 의미한다.

$t$ 을 최종고장시점  $x_n$ 으로 대치하고 (12)식에 (6)식과  $\lambda(t|\theta, \beta) = \theta e^{-\beta t}$  대입하면 우도함수는 다음과 같이 표현 할 수 있다.

$$L_{NHPP}(\theta | D_{x_n}) = \left( \prod_{i=1}^n \theta \beta e^{-\beta x_i} \right) \exp(-\theta(1 - e^{-\beta x_n})) \tag{13}$$

(14)식을 이용한 최우추정법(MLE)을 이용하기 위한 로그우도함수를 구하면 다음과 같다.

$$\ln L_{GO}(\theta | D_{x_n}) = n \ln \theta + n \ln \beta - \beta \sum_{i=1}^n x_i - \theta(1 - e^{-\beta x_n}) \tag{14}$$

따라서 고장절단모형에서의 모수  $\theta$ 와  $\beta$ 에 관한 편미분식은 다음과 같이 유도 할 수 있다[11].

$$\frac{\partial \ln L_{GO}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-\beta x_n} = 0 \tag{15}$$

$$\frac{\partial \ln L_{GO}(\theta | \underline{x})}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^n x_i - \theta x_n e^{-\beta x_n} = 0 \tag{16}$$

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

각 모수에 대한 최우추정량  $\hat{\theta}_{MLE}$ 와  $\hat{\beta}_{MLE}$ 은 다음 식을 만족한다[12].

$$\frac{n}{\hat{\theta}} = 1 - e^{-\hat{\beta} x_n} \tag{17}$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^n x_i + \hat{\theta} x_n e^{-\hat{\beta} x_n} \quad (18)$$

### 3.2 비선형 반-로지스틱 강도함수

반-로지스틱 분포(half-Logistic distribution)[15]는 공정관리 부분에서 많이 사용되는 분포이고 평균값 함수와 강도함수는 다음과 같이 알려져 있다[15][16].

$$m(t|\theta, \beta) = \theta \frac{(1 - e^{-\beta t})}{(1 + e^{-\beta t})} \quad (\theta > 0, \beta > 0) \quad (19)$$

$$\lambda(t|\theta, \beta) = \theta \frac{2\beta e^{-\beta t}}{(1 + e^{-\beta t})^2} \quad (20)$$

$t$  을 최종고장시점  $x_n$  으로 대치하고 (6)식에 (20)식과 (21) 대입하면 우도함수는 다음과 같이 표현 할 수 있다.

$$L_{HL}(\theta | D_{x_n}) = \left( \prod_{i=1}^n \theta \frac{2\beta e^{-\beta x_i}}{(1 + e^{-\beta x_i})^2} \right) \exp\left(-\theta \frac{(1 - e^{-\beta x_n})}{(1 + e^{-\beta x_n})}\right) \quad (21)$$

(21)식을 이용하여 각 모수에 대한 최우추정량  $\hat{\theta}_{MLE}$  와  $\hat{\beta}_{MLE}$  은 다음 식을 만족한다.

$$\frac{\partial \ln L_{HL}(\theta | \underline{x})}{\partial \theta} = 0, \quad \frac{\partial \ln L_{HL}(\theta | \underline{x})}{\partial \beta} = 0, \quad \frac{\partial^2 \ln L_{HL}(\theta | \underline{x})}{\partial^2 \beta} = 0 \quad (22)$$

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

(22)식은 다음과 같이 요약 할 수 있다[16].

$$\hat{\theta} = \frac{(1 + e^{-\beta x_n})}{(1 - e^{-\beta x_n})} \quad (23)$$

$$g(\beta) = \sum_{i=1}^n x_i - \frac{n}{\beta} - 2 \sum_{i=1}^{n-1} \frac{x_i e^{-\beta x_i}}{(1 + e^{-\beta x_i})} - \frac{2x_n e^{-\beta x_n}}{(1 + e^{-\beta x_n})} \left[ 1 - \frac{n}{(1 - e^{-\beta x_n})} \right] = 0 \quad (24)$$

(24)식을 뉴우턴 랩슨법(Newton-Raphson method)을 이용하면  $\hat{\beta}_{MLE}$  을 추정 할 수 있다고 하였고 이 값을

(24)식에 대입하면  $\hat{\theta}_{MLE}$  의 값을 추정 할 수 있다고 하였다[16].

## 4. 소프트웨어 고장 자료 분석 및 방출 시기 분석

이 장에서 소프트웨어 고장 간격 시간 자료[17] (failure interval time data)를 가지고 제시하는 신뢰모형들을 분석하고자 한다. 이 자료의 고장 시간은 738.68 시간단위에 30번의 고장이 발생된 자료이며 <표 1>에 나열 되어 있다. 또 한 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다 [18].

추세 분석에는 일반적으로 라플라스 추세 검정(Laplace trend test)을 사용한다. 이 검정을 실시한 결과(그림 1)에서 라플라스 추세 검정의 결과는 라플라스 요인(factor)이 -2와 2사이에 존재함으로서 신뢰성장(reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰 성장모형을 제시하는 것이 효율적임을 시사하고 있다[15].

Goel-Okumoto 모형에 대한 모수 추정은 최우추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(bisection method)을 사용하였다. 이러한 계산은 초기값을 0.001와 1을, 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. 그리고 반-로지스틱 분포에 대한 모수 추정값은 R. Satya Prasad, K. R. H. Rao and R.R. L Kantha,[16]들이 추정한 값을 본 논문에서는 사용하였다. 그 결과는 <표 2>에 요약되었고 또한 이 표에서 평균제곱오차와 결정계수도 요약 되었다. 이 표에서도

평균제곱오차의 값이 반-로지스틱 분포 모형이 Goel-Okumoto 모형보다 작게 나타나 반-로지스틱 분포 모형이 보다 효율적임을 보이고 있다. 그리고 추정치에 대한 차이를 설명하는 결정계수 측면에서는 결정계수가 0.7 이상으로 나타나 추정치의 효율성이 높게 나타나고 있다.

<Table 1> Failure time data

Failure number	Failure time (hours)	Failure interval (hours)
1	30.02	30.02
2	31.46	1.44
3	53.93	22.47
4	55.29	1.36
5	58.72	3.43
6	71.92	13.2
7	77.07	5.15
8	80.9	3.83
9	101.9	21
10	114.87	12.97
11	115.34	0.47
12	121.57	6.23
13	124.97	3.4
14	134.07	9.1
15	136.25	2.18
16	151.78	15.53
17	177.5	25.72
18	180.29	2.79
19	182.21	1.92
20	186.34	4.13
21	256.81	70.47
22	273.88	17.07
23	277.87	3.99
24	453.93	176.06
25	535	81.07
26	537.27	2.27
27	552.9	15.63
28	673.68	120.78
29	704.49	30.81
30	738.68	34.19

<Table 2> Parameter estimation and  $MSE, R^2$

Model	MLE	Model Comparison	
		MSE	$R^2$
Goel-Okumoto	$\hat{\theta}_{MLE} = 31.85962,$ $\hat{\beta}_{MLE} = 0.003096$	59.46841	0.749
Half-Logistic distribution	$\hat{\theta}_{MLE} = 31.27686,$ $\hat{\beta}_{MLE} = 0.00433$	16.41152	0.796

한편, Goel-Okumoto 모형의 신뢰도는 (8)식을 이용하면 다음과 같이 유도 된다.

$$\hat{R}_{GO}(t | x_n) = \exp(\theta e^{-\beta x_n} (e^{-\beta t} - 1)) \quad (25)$$

따라서 소프트웨어 방출시간  $T_R$ 이 신뢰도  $R_0 = \hat{R}(x | T_R)$ 을 확보해야 한다면 다음 방정식을 만족해야 한다.

$$\ln R_0 = (\theta e^{-\beta T_R} (e^{-\beta x} - 1)) \quad (26)$$

그리고 Goel-Okumoto 모형의 비용 함수  $C(T)$ 는 (9)를 연관하면 다음과 같이 유도 된다.

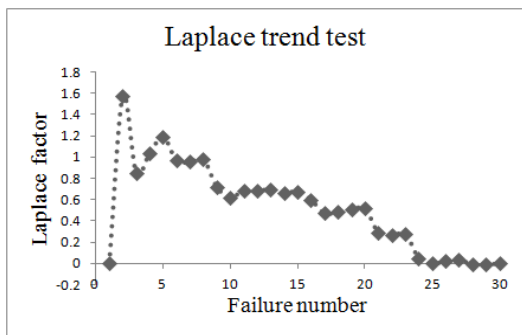
$$T_C = \left[ \ln \left( \frac{(c_1 - c_2)\theta \beta}{-c_3} \right) \right] / \beta \quad (27)$$

<Table 3> Optimal software release time estimation ( $R_0 = 0.95$ )

Model	Estimation time	Optimal software release time ( $T_{OP}$ )
Goel-Okumoto	$\hat{T}_R = 1649.952$ $\hat{T}_C = 563.3521$	1649.952
Half-Logistic distribution	$\hat{T}_R = 1638.472$ $\hat{T}_C = 603.4241$	1638.472

유사하게 반-로지스틱 분포에 대한 신뢰도와 비용함수는 다음 식을 만족해야 한다.

$$\frac{\ln R_0}{\theta} = \frac{1 - e^{-\beta T_R}}{1 + e^{-\beta T_R}} - \frac{1 - e^{-\beta(x + T_R)}}{1 + e^{-\beta(x + T_R)}} \quad (28)$$



[Fig. 1] Laplace trend test

$$\frac{2(c_1 - c_2)\theta\beta}{-c_3} - \frac{(1 + e^{-\beta T_c})^2}{e^{-\beta T_c}} = 0 \quad (29)$$

<표 3>에서는  $c_1=20(\$)$ ,  $c_2=25(\$)$  그리고  $c_3=30(\$)$  라고 가정하고 시스템 수명시간은 2000시간이고 임무시간  $x$ 는 100이고  $R_0$ 을 0.95(95%)를 투입하여 각 모형에 대한 추정시간의 결과와 최적방출시간은 <표 3>에 요약되었다. 이 표에서 Goel-Okumoto 모형과 반-로지스틱 분포 모형 모두 비용 측면 보다는 신뢰도 측면에서 방출시간이 길기 때문에 신뢰도에 대한 사전 노력이 필요하다고 할 수 있고 비용측면에서는 반-로지스틱 분포 모형보다 Goel-Okumoto 모형이 길지 않기 때문에 Goel-Okumoto 모형이 보다 더 효율적으로 나타나고 있지만 신뢰도 측면에서는 반대로 나타나고 있다. 최적시기에 관해서는 기존모형인 Goel-Okumoto 모형보다 반-로지스틱 분포 모형이 더 효율적이라고 할 수 있다.

결국 제안된 반-로지스틱 분포 모형도 비용 측면보다는 신뢰도 측면이 방출시간에 많은 영향을 미친다고 할 수 있다.

한편, 보안을 강화하기위해 비용을 들이면 들일수록 보안사고로 인한 잠재적 피해액은 줄어든다. 그러나 일정수준이 지나면 잠재적 피해액 보다 보안을 위한 비용이 더욱 많이 발생할 수 있다. 보안책임자는 최소한의 비용으로 최대한 잠재적 보안피해액을 줄이도록 노력해야 한다[19].

따라서 이러한 방출시기를 사전 정보로 활용하면 잠재적 보안피해액을 줄 일 수 있다고 판단된다.

## 5. 결론

본 연구는 Half-Logistic 분포 모형을 적용한 유한고장 NHPP 모형을 이용하여 최적 방출시기에 관한 문제를 알아보았다. 즉, 대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 수 없는 상황이 현실이다. 실제로 만족할 만한 신뢰도가 부여되고 동시에 시스템 고장과 연계된 기대 총비용을 최소화시키기 위하여 필요하다면 충분한 테스트를 계속해야 한다. 따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적방출시간이다. 본 연구에서는 반-로지스틱 분포를 이

용한 방출시기 모형을 시도한 결과 효율성을 입증 하였다. 따라서 반-로지스틱 분포를 이용한 모형도 이 분야에서 적용 가능한 모형이 될 수 있음을 확인하였다. 경우에 따라서는 왜도와 첨도 측면에서 효율적인 카파분포, 지수화지수분포 등 업데이트된 분포에 대한 방출 시기 문제를 비교 분석하는 연구도 가치 있는 일이라 판단되고 이 연구를 통하여 소프트웨어 개발자들은 방출최적시기를 파악 하는데 어느 정도 도움을 줄 수 있을 뿐 아니라 잠재적 보안피해액을 줄 일 수 있다고 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 AL, 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 Trans Reliab*, 32, pp.475-484, 1983.
- [4] Zhao M, Change-point problems in software and hardware reliability", *Commun. Stat. Theory Methods*,22(3), pp.757-768, 1993.
- [5] Shyur H-J., A stochastic software reliability model with imperfect debugging and change-point, *J Syst. Software* 66, pp.135-141, 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* 76, pp.181-194, 2005.
- [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] Kim Hee Cheul and Shin Hyun Cheul, The Comparative Study for NHPP Software Reliability Model based on the Property of Learning Effect of Log Linear Shaped Hazard Function, Convergence Security Journal, Vol. 12, No. 3, pp. 19-26, 2012.
- [10] J. F. Lawless, Statistical Models and Methods for Lifetime Data. John Wiley & Sons, New York, 1981.
- [11] L. Kuo and T. Y. Yang., Bayesian Computation of Software Reliability, Journal of the American Statistical Association, Vol.91, pp. 763-773, 1996.
- [12] Kim Hee Cheul, The Comparative Study for NHPP Software Reliability Growth Model Based on Non-linear Intensity Function, Convergence Security Journal, Vol. 7, No. 7, pp. 2-8, 2007.
- [13] Xie, M. and Homg, G. Y, Software release time determination based on unbound NHPP model, Proceeding of the 24th International Conference on Computers and Industrial Engineering. pp. 165-168, 1999.
- [14] Hee-Cheul KIM and Hyoung-Keun Park, The Comparative Study of Software Optimal Release Time Based on Burr Distribution, International Journal of Advancements in Computing Technology, Volume 2, Number 3, pp. 119-128, 2010.
- [15] Khaled M.S. Faqih, Proceedings of the International Multi Conference of Engineers and Computer Scientists, Vol I IMECS, Hongkong, pp.18-20, March, 2009.
- [16] R. Satya Prasad, K. R. H. Rao and R.R. L. Kantha, 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
- [17] M.Xie, T.N. Goh, P. Rajan, Some effective control chart procedures for reliability monitoring; Elsevier science Ltd, Reliability Engineering and system safety 77, pp. 143-150, 2002.
- [18] 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.
- [19] Kim Hee Cheul and Kyung-Soo Kim, The Comparative Study for Property of Learning Effect based on Software Reliability Model using Doubly Bounded Power Law Distribution, Convergence Security Journal, Vol. 13, No. 1, pp.71-77, 2013.

**김 경 수(Kim, Kyung Soo)**



- 2001년 8월 순천대학교 전산학과 (공학 박사)
- 2005년 3월 ~ 2007년 2월 : VCU DBLab Visiting Scholar
- 1998년3월~현재: 백석문화대학교 인터넷정보학부 교수
- 관심분야 : 소프트웨어신뢰성 공학, 정보보안, 웹 프로그래밍

· E-Mail : kkskim@bscu.ac.kr

**김 희 철(Kim, Hee Cheul)**



- 1992년 2월 : 동국대학교 통계학과 (이학 석사)
- 1998년 8월 : 동국대학교 통계학과 (이학박사)
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수
- 관심분야 : 소프트웨어신뢰성 공학, 전산 통계

· E-Mail : kim1458@nsu.ac.kr