

# 로그선형 학습요인을 이용한 유한고장 NHPP모형에 근거한 소프트웨어 최적방출시기 비교 연구

김희철\* · 신현철\*\*

## 요 약

본 연구에서는 소프트웨어 제품을 개발하여 테스트를 거친 후 사용자에게 인도하는 시기를 결정하는 방출문제에 대하여 연구 하였다. 소프트웨어의 결함을 제거하거나 수정 작업과정에서 학습요인을 고려한 유한고장수를 가진 비동질적인 포아송 과정에 기초하였다. 수명강도는 다양한 형상모수와 척도모수에 이용 할 수 있기 때문에 신뢰성 분야에서 많이 사용되는 로그 선형 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.

## The Comparative Study of Software Optimal Release Time of Finite NHPP Model Considering Log Linear Learning Factor

Kim Hee Cheul\* · Shin Hyun Cheul\*\*

## 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, considering learning factor, presented and propose release policies of the life distribution, log linear type 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.

**Key words :** Software Release Policies, NHPP, Log Linear Type, Mission Time.

---

접수일(2012년 11월 15일), 수정일(1차: 2012년 12월 17일),  
게재확정일(2012년 12월 20일)

\* 남서울대학교 산업경영공학과

\*\* 백석문화대학교 인터넷정보학부

## 1. 서 론

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 불가결하다. 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 에러 탐색 과정측면에서는 우수한 모형이고 이 모형은 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지 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-shape d)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 하

였다. 그리고 최근에는 S-형태 모형은 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 익숙해지는 학습 과정을 설명할 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 유한 고장수를 가진 비동질적인 포아송 과정에 기초하고 수명분포는 로그선형 학습요인을 이용한 최적 방출시기에 관한 문제를 제시하여 소프트웨어 요구 신뢰도를 만족시키고 소프트웨어 개발 및 유지 총비용을 최소화 시키는 최적 소프트웨어 방출 정책에 대하여 논의되었다.

## 2. 관련 연구

### 2.1 유한고장 NHPP 모형

신뢰도에서 관측시간  $(0, t]$  사이에 발견된 고장 수  $N(t)$ 을 모형화 하는데 비동질적 포아송 과정

이 널리 사용하여 왔다. 이 과정(process)에서 강도 함수(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 이 모형들에 대한 강도함수는 각각 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 표현할 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  $\left(\lim_{t \rightarrow \infty} m(t) = \theta < \infty\right)$  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 NHPP 모형의 평균값 함수는 다음과 같이 표현 할 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

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

이라고 하면 데이터 집합  $D_t$  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,  $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$

NHPP 모형에서 테스트 시점  $x_n$  에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간  $(x_n, x_n + t]$  (단,  $t$  는 임무시간(mission time)사이에서 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도(reliability)  $\hat{R}(t | x_n)$  )는 다음과 같이 됨이 알려져 있다[1][7].

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

### 2.2 학습효과를 고려한 강도함수와 누적함수

소프트웨어 테스트링 작업에 있어서 학습효과는 테스트링 관리자에 의해 동일한 혹은 조작 가능한 작업이 될 수 있으므로 이러한 효과들을 어떠한 방법으로 반영하는가는 소프트웨어 신뢰성에 중요한 과정이 된다.

소프트웨어 에러들을 발견하기 위하여 자동 에러 탐색요인(autonomous errors-detected factor)  $\gamma$  과 학습요인(learning factor)  $\eta$  이 포함된 영향요인들(influential factors)이 고려될 수 있다. 따라서  $f(t)$  을  $t$  시점에서 에러를 발견된 확률을 나타내는 확률밀도 함수이고  $F(t)$  을  $(0, t]$  시점까지의 누적분포함수라고 가정하면 영향요인들을 고려한 모형은 다음과 같이 표현 된다[8][9].

$$f(t) = (\gamma + \eta F(t)) (1 - F(t)) \quad (8)$$

단,  $\gamma, \eta > 0, t > 0$ .

(8)식에서 자동 에러 탐색요인  $\gamma$  는 사전에 알지 못하지만 테스트링 과정에서 테스트링 관리자가 자동적으로 에러를 발견하는 요인이지만 학습 요인  $\eta$  은 과거에 발견된 에러패턴을 바탕으로 세밀하게 에러를 발견하기 위하여 테스트링 관리자가 설정해 놓은 요인을 의미한다.

한편 (8)을 다음과 같이 위험함수 형태로 변경 할 수 있다.

$$h(t) = (\gamma + \eta F(t)) \quad (9)$$

$$\text{단, } h(t) = \frac{F'(t)}{1-F(t)} = \frac{f(t)}{1-F(t)}$$

(9)식에서 누적분포함수와 확률밀도함수는 다음과 같이 수정 가능하다.

$$F(t) = \frac{h(t) - \gamma}{\eta}, \quad f(t) = F'(t) = \frac{h'(t)}{\eta} \quad (10)$$

### 2.3 효율적인 모형 선택 기준

최근에 모형에 대한 효율성을 조사하기 위한 기준으로서 평균제곱오차(MSE)와 결정계수( $R^2$ )를 사용한다[8][9].

#### 2.3.1 평균제곱오차

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.

#### 2.3.2 $R^2$

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 모

형으로 간주된다.

### 2.4 신뢰도와 비용 최소화를 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[8][9].

$$\begin{aligned} \hat{R}(x | x_n) &= \exp\left(-\int_{x_n}^{x_n+x} \lambda(\tau) d\tau\right) \\ &= \exp[-\{m(x+x_n) - m(x_n)\}] \end{aligned} \quad (11)$$

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

$$\ln \hat{R}_0 = -m(x + T_R) + m(T_R) \quad (12)$$

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[12][13].

$$C(T) = c_1 m(T) + c_2 [m(T_{LC}) - m(T)] + c_3 T \quad (13)$$

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.[4].

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

$$\frac{\partial C(T)}{\partial T} = 0, \quad \frac{\partial^2 C(T)}{\partial^2 T} > 0 \quad (14)$$

$$F(t) = \frac{h(t) - \gamma}{\eta} = \frac{e^{a+bt}}{\eta} - \frac{\gamma}{\eta} \quad (17)$$

결국  $T$  에 관해서 비용함수  $C(T)$  을 미분하면 최적방출시간  $T_C$  를 계산 할 수 있다[5, 6].

따라서 신뢰성 요구를 만족하고 총 비용을 최소화 하는 상황이 최적방출시간이다. 그러므로 최적 방출시간  $T_{OP}$  는  $T_R$  과  $T_C$  에 대하여 다음을 만족한다[13].

$$T_{OP} = \text{Max}(T_C, T_R) \quad (15)$$

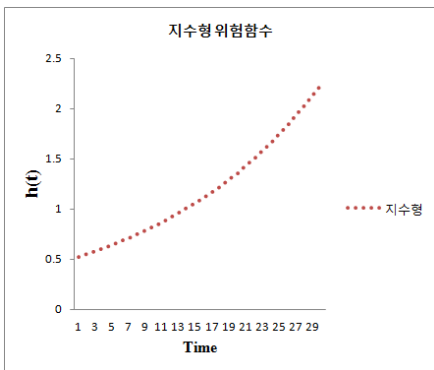
### 3. 로그선형 위험함수를 이용한 NHPP 소프트웨어 모형

로그선형위험함수 다음과 같이 정의 된다[10].

$$h(t) = \exp(a+bt) \quad (16)$$

단,  $a, b > 0$

위 위험함수의 패턴은 (그림 1)에서 모여 주듯이 위험함수가 시간에 따라 상승하는 패턴(bathtub-shap ed)을 가지고 있다.



(그림 1) 위험함수

따라서 학습효과를 고려한 분포함수는 (10)식을 이용하면 지수형 분포함수는 다음과 같이 유도 된다.

결과적으로 학습효과를 고려한 확률밀도함수는 각각 다음과 같이 유도 된다.

$$f(t) = F'(t) = \frac{h'(t)}{\eta} = \frac{b}{\eta} e^{a+bt} \quad (18)$$

따라서 (3)식과 (17)식을 이용하면 학습효과를 고려한 유한 고장 NHPP 모형의 지수형 평균값 함수와 강도함수는 다음과 같이 표현할 수 있다.

$$m(t) = \theta F(t) = \theta \left( \frac{e^{a+bt}}{\eta} - \frac{\gamma}{\eta} \right) \quad (19)$$

$$\lambda(t) = \theta F'(t) = \theta \frac{b}{\eta} e^{a+bt} \quad (20)$$

이 경우의 우도함수는 (6)식에 (19)식과 (20)식을 대입하면 다음과 같다.

$$L_{NHPP}(\theta | \underline{x}) = \quad (21)$$

$$\prod_{i=1}^n \left( \frac{\theta b e^{a+bx_i}}{\eta} \right) \exp \left[ -\theta \left( \frac{e^{a+bx_n} - \gamma}{\eta} \right) \right]$$

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

모수 추정방법은 최우추정법(MLE)을 사용하였고 최우추정법을 이용하기 위한 로그우도함수는 (21)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta | D_{x_n}) = \quad (22)$$

$$n \ln \theta + n \ln b - n \ln \eta + n a + b \sum_{i=1}^n x_i - \theta \left( \frac{e^{a+bx_n} - \gamma}{\eta} \right)$$

본 논문에서는 지수모형의 특성을 유지하면서도 보다 간결하게 하기 위하여 (22)식에서 형상모수  $a = 0.5$  으로 가정한 모형을 사용하고자 한다. 즉,  $\theta$  와  $b$  에 대하여 편미분하여 다음과 같은 식을 만족하는  $\hat{\theta}_{MLE}$  와  $\hat{b}_{MLE}$  을 수치 해석적 방법으로 추정할 수 있다 [13].

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - \left( \frac{e^{a+b x_n} - \gamma}{\eta} \right) = 0 \quad (23)$$

$$\text{즉, } \hat{\theta} = \frac{n\eta}{e^{a+b x_n} - \gamma}$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial b} = \frac{n}{b} + \sum_{i=1}^n x_i - \frac{\theta x_n e^{a+b x_n}}{\eta} = 0 \quad (24)$$

#### 4. 소프트웨어 고장 자료 분석 및 방출 시기 분석

이 장에서 소프트웨어 고장 간격 시간 자료[14] (failure interval time data)를 가지고 제시하는 신뢰 모형들을 분석하고자 한다. 이 자료의 고장 시간은 18.735 시간단위에 30번의 고장이 발생한 자료이며 <표 1>에 나열 되어 있다. 모수추정은 최우추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(bisection method)을 사용하였다. 그리고 모수추정을 용이하게 하기 위하여 원 자료를 변수변환(failure time  $\times 10^{-2}$ )하여 사용하였다. 이러한 계산은 초기값을 0과 10을, 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 MSE 값들이 <표 2>에 요약되었다. 그 결과는 <표 3>에 요약되었다. 그리고 이 표에 모형에 대한 효율성을 조사하기 위한 기준으로서 MSE와  $R^2$  값도 요약되었다.

<표 3>에서 학습요인  $\eta$ 이 증가하고 자동 에러 탐색요인  $\gamma$ 이 감소하는 경우에 평균값 추정치가 점점 실제 평균값 함수(고장 수)에 근접하게 추정되어 MSE 값들이 작게 나타나고 있어 점점 효율적인 모형으로 변해가고 있음을 확인 할 수 있다. 특히  $\eta=0.6$  보다 크고  $\gamma=0.4$ 보다 작을 경우 급격하게 MSE 값이 작게 나타나고 있다. 즉 학습요인  $\eta$ 이 자동 에러 탐색요인  $\gamma$ 보다 클수록 효율적인 모형으로 나타나고 있다. 그리고 추정치에 대한 차이를 설명하는 결정계수 측면에서는 결정계수가 0.7 이상으로 나타나 추정치의 효율성이 높게 나타나고 있다.

<표 1> 고장 자료

Failure Number	Failure Interval (second)	Failure Time(second)
1	0.479	0.479
2	0.266	0.745
3	0.277	1.022
4	0.554	1.576
5	1.034	2.610
6	0.949	3.559
7	0.693	4.252
8	0.597	4.849
9	0.117	4.966
10	0.170	5.136
11	0.117	5.253
12	1.274	6.527
13	0.469	6.996
14	1.174	8.170
15	0.693	8.863
16	1.908	10.771
17	0.135	10.906
18	0.277	11.183
19	0.596	11.779
20	0.757	12.536
21	0.437	12.973
22	2.230	15.203
23	0.437	15.640
24	0.340	15.980
25	0.405	16.385
26	0.575	16.96
27	0.277	17.237
28	0.363	17.600
29	0.522	18.122
30	0.613	18.735

<표 3> 각 모형의 모수 추정값 및 MSE와  $R^2$

영향 요인	지수형( $a = 0.5$ )				
	$\eta$	$\gamma$			
			$MLE$	$MSE$	$R^2$
0.1	0.9	$\hat{b}_{MLE} = 0.1074, \hat{\theta}_{MLE} = 0.9621$	16203.4	0.836	
0.2	0.8	$\hat{b}_{MLE} = 0.1076, \hat{\theta}_{MLE} = 1.9119$	12957.0	0.847	
0.3	0.7	$\hat{b}_{MLE} = 0.1078, \hat{\theta}_{MLE} = 2.8498$	10653.8	0.857	
0.4	0.6	$\hat{b}_{MLE} = 0.1080, \hat{\theta}_{MLE} = 3.7758$	8957.19	0.865	
0.5	0.5	$\hat{b}_{MLE} = 0.1082, \hat{\theta}_{MLE} = 4.6902$	7668.75	0.873	
0.6	0.4	$\hat{b}_{MLE} = 0.1084, \hat{\theta}_{MLE} = 5.5933$	523.256	0.793	
0.7	0.3	$\hat{b}_{MLE} = 0.1085, \hat{\theta}_{MLE} = 6.4853$	484.933	0.820	
0.8	0.2	$\hat{b}_{MLE} = 0.1087, \hat{\theta}_{MLE} = 7.3662$	447.29	0.840	
0.9	0.1	$\hat{b}_{MLE} = 0.1089, \hat{\theta}_{MLE} = 8.2364$	<b>412.09</b>	0.856	

단, Influential factors :  $\eta, \gamma$  ;  $MLE$  : Maximum likelihood estimation;  $MSE$ : Mean square error;  $R^2$  : Coefficient of determination; Smallest  $MLE$  and highest  $R^2$  are indicated in boldface.

한편, 로그선형 학습요인을 이용한 신뢰도는 (11)식을 이용하면 다음과 같이 유도 된다.

$$\hat{R}_{Exp}(t | x_n) = \exp \left[ - \left( \frac{\theta}{\eta} (e^{a+b(x_n+t)} - \gamma) \right) + \left( \frac{\theta}{\eta} (e^{a+bx_n} - \gamma) \right) \right] \quad (25)$$

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

$$\ln R_0 = - \left( \frac{\theta}{\eta} (e^{a+b(T_R+x)} - \gamma) \right) + \left( \frac{\theta}{\eta} (e^{a+bT_R} - \gamma) \right) \quad (26)$$

그리고 지수형 학습요인을 이용한 비용 함수  $C(T)$ 는 (13)를 연관하면 다음과 같이 유도 된다.

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

<표 4> 최적 방출시간 계산( $R_0 = 0.95$ )

영향요인	추정시간	최적방출시간( $T_{OP}$ )
$\eta = 0.3, \gamma = 0.7$	$\hat{T}_R = 18.00082$ $\hat{T}_C = 18.86836$	18.86836
$\eta = 0.5, \gamma = 0.5$	$\hat{T}_R = 14.00485$ $\hat{T}_C = 12.53580$	14.00485
$\eta = 0.7, \gamma = 0.3$	$\hat{T}_R = 14.06487$ $\hat{T}_C = 12.68066$	14.06487

<표 4>에서는  $c_1=20(\$)$ ,  $c_2=25(\$)$  그리고  $c_3=30(\$)$  라고 가정하고 시스템 수명시간은 200시간이고 임무시간  $x$ 는 0.01이고  $R_0$ 을 0.95(95%)를 투입하여 각 모형에 대한 추정시간의 결과와 최적방출시간은 <표 4>에 요약 되었다. 이 표에서 학습요인  $\eta$ 이 클수록 비용 측면에서는 개선되고 있고 학습요인과 자동 탐색요인이 같은 경우가 최적방출시간에 대해서는 빨라지고 있음을 확인 할 수 있었다. 결국 학습요인에 대한 사전 노력은 신뢰도 측면이든 비용측면이든 방

출시간에 많은 영향을 미친다고 할 수 있다.

## 5. 결 론

본 연구는 지수형 학습요인을 이용한 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 로그선형 학습요인을 이용한 방출시기 모형을 시도한 결과 효율성을 입증 하였다. 따라서 로그선형 학습요인을 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 방출 최적시기를 파악 하는데 어느 정도 도움을 줄 수 있으리라 사료 된다.

## 참고문헌

- [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 an

d 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] 김희철, 신현철 "대수 선형 위험함수 학습효과에 근거한 NHPP 신뢰성장 소프트웨어 모형에 관한 비교 연구", 한국융합보안학회, 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] 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.

[13] 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.

[14] Y. HAYAKAWA and G. TELFAR "Mixed Poisson-Type Processes with Application in Software

Reliability", Mathematical and Computer Modelling, 31, pp. 151-156, 2000.

---

[ 저 자 소 개 ]

---



**김희철 (Hee-cheul Kim)**

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

email : kim1458@nsu.ac.kr



**신현철 (Hyun-cheul Shin)**

1990년 2월 광운대학교 전자계산학과 졸업(공학석사)  
 2002년 2월 원광대학교 컴퓨터공학과 졸업(공학박사)  
 19945-현재 백석문화대학교 인터넷 정보학부 교수

email : hcshin@bcc.ac.kr