

형상모수를 고려한 소프트웨어 신뢰성 비용 모형에 관한 비교 연구

김경수*, 김희철**

백석문화대학교 인터넷 정보학부*, 남서울대학교 산업경영공학과**

The Comparative Software Reliability Cost Model of Considering Shape Parameter

Kyung-Soo Kim^{*}, Hee-Cheul Kim^{**}

Dept. of Internet information, BaekSeok Culture University^{*}

Dept. of Industrial & Management Engineering, Namseoul University^{**}

요 약 본 연구에서는 소프트웨어 제품 테스트 과정에서 고장 수명분포의 형상모수를 고려한 소프트웨어 신뢰성 비용 모형에 대하여 연구 하였다. 신뢰성 분야에서 많이 사용되는 어랑 분포와 로그-로지스틱 모형을 이용한 형상모수를 반영한 문제를 제시하였다. 소프트웨어 고장모형은 유한고장 비동질적인 포아송과정을 이용하고 모수추정법은 최우추정법을 이용 하였다. 따라서 본 논문에서는 형상모수를 고려한 소프트웨어 비용모형 분석을 위하여 소프트웨어 고장 시간 자료를 적용하여 비교 분석하였다. 본 연구에서 사용된 어랑 분포와 로그-로지스틱분포에 근거한 소프트웨어 비용 모델을 비교한 결과 어랑 모형은 최적의 소프트웨어 방출 시간을 예측 할 수 있지만 로그-로지스틱 모형은 방출시간을 예측 할 수 없기 때문에 로그-로지스틱 보다 어랑 모형이 보다 효율적으로 나타나고 있다. 이 연구를 통하여 소프트웨어 개발자들은 소프트웨어 개발 비용을 파악 하는데 어느 정도 도움을 줄 수 있을 것으로 사료된다.

주제어 : 형상모수, 비동질적 포아송 과정, 비용모형, 어랑 분포, 로그-로지스틱 분포

Abstract In this study, reliability software cost model considering shape parameter based on life distribution from the process of software product testing was studied. The shape parameter using the Erlang and Log-logistic model that is widely used in the field of reliability problems presented. The software failure model was used finite failure non-homogeneous Poisson process model, the parameters estimation using maximum likelihood estimation was conducted. In comparison result of software cost model based on the Erlang distribution and the log-logistic distribution software cost model, because Erlang model is to predict the optimal release time can be software, but the log-logistic model to predict to optimal release time can not be, Erlang distribution than the log-logistic distribution appears to be effective. In this research, software developers to identify software development cost some extent be able to help is considered.

Key Words : Shape Parameter, NHPP, Cost Model, Erlang Distribution, Log-logistic Distribution

Received 31 December 2013, Revised 10 February 2014

Accepted 20 March 2014

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 신뢰성은 일정한 환경조건에서 일정기간 동안 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 비동질적 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 모형은 Yamada 와 Ohba [4]에 의해 지연된 S-형태 신뢰성장모형(Delayed S-shaped reliability growth model)과 변곡된 S-형태 신뢰성장모형(Inflexion 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 최근

에, Huang [8]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 설명할 수 있다고 하였고[9] 또한, 반-로지스틱 위험함수를 이용한 학습과정 특성을 연구하기도 하였다[10].

소프트웨어 신뢰성 분야에서 널리 사용되는 분포 중에서 감마분포(Gamma distribution)[11]와 로그-로지스틱 분포(Log-logistic distribution)[12][13]는 척도(Scale) 모수 및 형상(Shape)모수 특성에 따라 다양한 분포를 표현할 수 있어 신뢰성 데이터 분석에 가장 널리 사용된다. 척도 모수는 분포의 크기를 나타내지만 형상모수는 분포의 비대칭성(왜도, Skewness)과 첨도(Kurtosis)등 분포의 형상을 결정하는 모수로서 중요한 위치를 차지하고 있다[11].

본 논문에서는 시간이 흐름에 따라 고장률(위험함수)이 증가하는 감마분포의 특수한 경우인 2차 어랑(2-Erlang)분포와 로그-로지스틱 분포에서 형상모수가 2를 가진 분포를 수명분포로 적용한 소프트웨어 개발 비용 모형을 비교 제시하였다.

2. 관련연구

2.1 소프트웨어 신뢰성

$N(t)$ 을 시간 t 까지 검출된 소프트웨어의 누적고장수라고 하고, $m(t)$ 를 이에 대한 기대값을 나타내는 평균값 함수(MVF, Mean Value Function)로 가정하고 $\lambda(t)$ 을 강도함수(Intensity function) (즉, t 에서의 순간 결함 검출율)이면 비동질 포아송 과정(NHPP, Non-homogeneous Poisson Processes)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 평균값 함수 $m(t)$ (Mean

value function)와 강도 함수 $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[2].

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

유한 고장 NHPP 모형에서는 시간 $(0, t]$ 까지 탐색되어 질 수 있는 결함의 기대값을 θ 라고 표현하면 유한 고장 NHPP모형의 평균값 함수와 강도함수는 다음과 같이 표현할 수 있다[10][14].

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

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

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

이라고 하면 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 모형에서 θ 을 모수공간이라고 표시하면 유한고장 NHPP 모형의 우도함수는 다음과 같이 알려져 있다[10, 14].

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

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

2.2 소프트웨어 개발 비용모형

소프트웨어 비용 모형은 다음과 같이 정의 된다 [15][16].

$$E = E_1 + E_2 + E_3 + E_4 \quad (6)$$

$$= E_1 + C_2 \times t + C_3 \times m(t) + C_4 \times [m(t+t') - m(t)]$$

단, E : 소프트웨어 개발 예상 총비용

E_1 : 소프트웨어 설계 및 초기 소프트웨어 개발의 비용 (분석 데이터, 소프트웨어 개발 전문가의 수, CPU

시간 등)

E_2 : 단위 시간당 소프트웨어 테스트 비용(상수) 즉, $E_2 = C_2 \times t$ (단, C_2 는 단위 시간당 비용이고 t 는 테스트 시점)

E_3 : 기본 결함을 감지하고 결함을 제거하는 등의 활동으로 하나의 결함을 제거하는 비용 즉, $E_3 = C_3 \times m(t)$ (단, C_3 는 테스트 과정에서 하나의 결함을 제거하는 비용, $m(t)$ 는 t 시점에서 탐색되어 질 수 있는 결함의 기대 수)

E_4 : 운영 소프트웨어 시스템에서 남아있는 모든 결함을 제거하는 비용(상수) 즉,

$E_4 = C_4 \times [m(t+t') - m(t)]$ (단, C_4 는 소프트웨어 출시된 이후에 소프트웨어 운영 단계에서 사용자가 관찰되는 결함수정 비용, t' 는 소프트웨어 시스템을 출시한 후 운영 및 소프트웨어를 유지할 수 있는 시간)

현실적으로는 C_4 는 C_2 와 C_3 보다 높은 비용을 나타낸다. 그러므로 최적의 최적 소프트웨어 방출시간 (t)는 다음과 같이 유도 할 수 있다.

$$\frac{\partial E}{\partial t} = E' = (E_1 + E_2 + E_3 + E_4)' = 0 \quad (7)$$

3. 제안한 형상모수가 2차인 어랑(2-Erlang) 분포와 로그-로지스틱(Log-logistic) 분포 소프트웨어 개발 비용 모형

소프트웨어 신뢰성 분야에서 널리 사용되는 분포 중 에서 감마분포(Gamma distribution)[11]와 로그로지스틱 분포(Log-Logistic distribution)[13]는 척도(Scale)모수 및 형상(Shape)모수 특성에 따라 다양한 분포를 표현할 수 있어 신뢰성 데이터 분석에 가장 널리 사용된다.

특히 감마분포에서 형상모수(α)가 상수인 경우를 어랑분포(Erlang distribution)라고 하고 확률밀도함수와 누적 분포함수는 각각 다음과 같다.

$$f(t) = \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t} \quad (8)$$

$$F(t) = \left(1 - \exp(\beta t) \sum_{j=0}^{\alpha-1} \frac{(\beta t)^j}{j!} \right) \quad (9)$$

단, $\alpha > 0, \alpha = 1, 2, 3, \dots, t \in [0, \infty], \beta > 0$ 을 만족한다.

본 논문에서는 고장률이 증가하는 형상모수가 $\alpha = 2$ 인 경우를 고려하였다. 따라서 (3), (4)식과 관련하여 유한 고장 NHPP 모형에서의 강도함수 $\lambda(t)$ 와 평균값 함수 $m(t)$, 는 각각 다음과 같이 나타 낼 수 있다.

$$\lambda(t = \theta f(t)) = \theta \frac{\beta^2}{\Gamma(2)} t^{2-1} e^{-\beta t} = \theta \beta^2 t e^{-\beta t} \quad (10)$$

$$m(t) = \theta F(t) = \theta [1 - e^{-\beta t} (1 + \beta t)] \quad (11)$$

이경우의 우도함수는 t 을 최종고장시점 x_n 으로 대치하고 (5)식에 (10)식과 (11)식을 대입하면 다음과 같다.

$$L_{NHPP}(\Theta | \underline{x}) = \left[\prod_{i=1}^n \theta \beta^2 x_i e^{-\beta x_i} \right] \cdot \exp\left(-\theta [1 - e^{-\beta x_n} (1 + \beta x_n)]\right) \quad (12)$$

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

모수 추정방법은 최우추정법(Maximum likelihood estimation; MLE)을 사용하였고 최우추정법을 이용하기 위한 로그 우도함수는 (12)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\Theta | \underline{x}) = n \ln \theta + 2n \ln \beta + \sum_{i=1}^n \ln x_i - \beta \sum_{i=1}^n x_i - \theta [1 - e^{-\beta x_n} (1 + \beta x_n)] \quad (13)$$

(13)식에서 θ 와 β 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\theta}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 을 수치 해석적 방법으로 계산할 수 있다[11].

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

$$\text{즉, } \hat{\theta} = \frac{n}{1 - e^{-\beta x_n} (1 + \beta x_n)} \quad (14)$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \beta} = \frac{2n}{\beta} - \theta \beta x_n^2 e^{-\beta x_n} - \sum_{i=1}^n x_i = 0 \quad (16)$$

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

로그로지스틱분포(Log-Logistic distribution)[13]는 신뢰성 분야에서 많이 사용되는 분포이고 위험함수가 증가하다가 감소하는 속성을 가지고 있다. 이 분포의 확률 밀도 함수와 분포함수는 다음과 같이 표현됨이 알려져 있다.

$$f(t|\tau, k) = \frac{\tau k (\tau t)^{k-1}}{[1 + (\tau t)^k]^2} \quad (\tau, k > 0) \quad (17)$$

$$F(t|\tau, k) = \frac{(\tau t)^k}{[1 + (\tau t)^k]} \quad (18)$$

(17)식과 (18)식을 이용하면 유한 고장 NHPP 모형에서의 평균값 함수와 강도함수는 다음과 같이 유도 할 수 있다.

$$m(t|\theta, \tau, k) = \theta \frac{(\tau t)^k}{[1 + (\tau t)^k]} \quad (\theta > 0, \tau, k > 0) \quad (19)$$

$$\lambda(t|\theta, \tau, k) = \theta \frac{\tau k (\tau t)^{k-1}}{[1 + (\tau t)^k]^2} \quad (20)$$

본 연구에서는 형상모수 $k = 2$ 인 경우를 적용하고자 한다[11]. 따라서 t 을 최종고장시점 x_n 으로 대치하고 (5)식에 (19)식과 (20)식을 대입하면 우도함수는 다음과 같이 표현할 수 있다.

$$L_{NHPP}(\Theta | \underline{x}) = \left(\prod_{i=1}^n \frac{2\theta \tau^2 x_i}{[1 + (\tau x_i)^2]^2} \right) \exp\left(-\theta \frac{(\tau x_n)^2}{[1 + (\tau x_n)^2]}\right) \quad (21)$$

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

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

$$\ln L_{NHPP}(\Theta | \underline{x}) = n \ln 2 + n \ln \theta + 2n \ln \tau + \sum_{i=1}^n \ln x_i - 2 \sum_{i=1}^n \ln [1 + (\tau x_i)^2] - \theta \frac{(\tau x_n)^2}{[1 + (\tau x_n)^2]} = 0 \quad (22)$$

(22)식을 이용하여 각 모수에 대한 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{\tau}_{MLE}$ 은 다음 식을 만족한다[13].

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - \frac{(\tau x_n)^2}{1 + (\tau x_n)^2} = 0 \quad (23)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \tau} = \frac{2n}{\tau} - 2\tau \sum_{i=1}^n \frac{x_i^2}{\ln(1 + (\tau x_i)^2)} - \theta \left(\frac{2\tau x_n^2 (1 + \tau^2 x_n - \tau^2 x_n^2)}{[1 + (\tau x_n)^2]^2} \right) = 0 \quad (24)$$

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

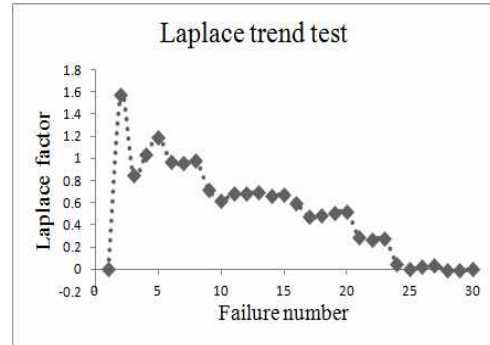
4. 소프트웨어 고장 자료 및 비용 분석

이 절에서는 소프트웨어 고장 시간 자료[12][17] (Failure interval time data)를 이용하여 본 논문에서 제시하는 소프트웨어 신뢰모형들의 개발비용을 분석하고자 한다. 이 자료의 고장 시간은 평균은 239.697이고 중앙값과 표준편차는 각각 144.015와 216.9314이고 범위는 708.66인 기초통계량을 가진 738.68시간단위에 30번의 고장이 발생한 자료이며 <표 1>에 나열 되어 있다.

<Table 1> 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 | 55.29 | 0.5529 |
| 5 | 58.72 | 0.5872 |
| 6 | 71.92 | 0.7192 |
| 7 | 77.07 | 0.7707 |
| 8 | 80.9 | 0.809 |
| 9 | 101.9 | 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 | 177.5 | 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 | 5.35 |
| 26 | 537.27 | 5.3727 |
| 27 | 552.9 | 5.529 |
| 28 | 673.68 | 6.7368 |
| 29 | 704.49 | 7.0449 |
| 30 | 738.68 | 7.3868 |



[Fig. 1] Laplace trend test

또 한 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[12][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 존재함으로써 즉, 극단값(Extreme value)이 존재하지 않으므로 이 자료를 이용하여 신뢰 성장모형을 제시하는 것이 효율적임을 시사하고 있다[14].

<Table 2> Parameter estimation of each model

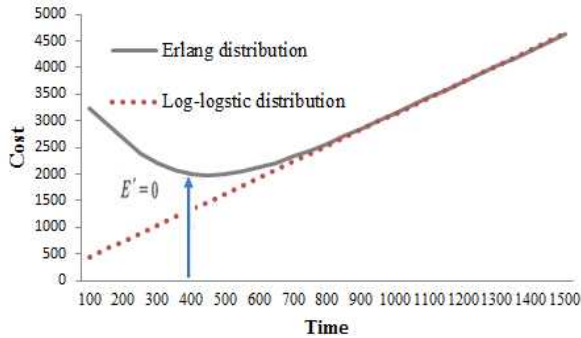
| Model | MLE |
|---------------------------|--|
| Erlang distribution | $\hat{\theta}_{MLE} = 30.6013, \hat{\beta}_{MLE} = 7.924 \times 10^{-3}$ |
| Log-logistic distribution | $\hat{\theta}_{MLE} = 32.2412, \hat{\tau}_{MLE} = 4.953 \times 10^{-1}$ |

모수 추정은 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. 그리고 이러한 계산은 초기 값을 0.0001와 3을, 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. 그 결과는 <표 2>에 요약되었다.

본 연구에서는 (6)식에서 다음과 같이 가정하여 비용 곡선을 분석하고자 한다.

(가정 1) $E_1 = 300\$, c_2 = 3\$, c_3 = 4\$, c_4 = 150\$, t' = 300$



[Fig. 2] The curve under the condition of (Assumption 1)

(그림 2)에서 보는 바와 같이 제안된 어랑 모형의 비용 성장 곡선은 처음에는 감소하다가 점차 증가하는 추세를 보이고 있다. 소프트웨어 시스템의 잔여 결함의 수는 결함의 제거하는 과정에서 점점 줄어들게 되고 즉, 남아 있는 결함이 관측될 확률은 낮아지게 된다.

따라서 테스트의 초기 단계에 있는 소프트웨어는 여전히 많은 오류가 있기 때문에 쉽게 감지 및 제거 단계에서 오류를 제거하는 비용은 운용 단계에서 오류를 제거하는 것보다 훨씬 낮기 때문에, 소프트웨어의 총비용은 감소한다. 그러나 일정시간 이후의 단계에서 소프트웨어에 남아있는 결함의 수는 적기 때문에 이 테스트 단계에서 결함을 검출 시간이 상대적으로 길고 오류를 제거하는 비용은 운용 단계에서 상대적으로 높기 때문에 비용 곡선은 시간이 흐름에 따라 지속적으로 증가하게 된다 [15].

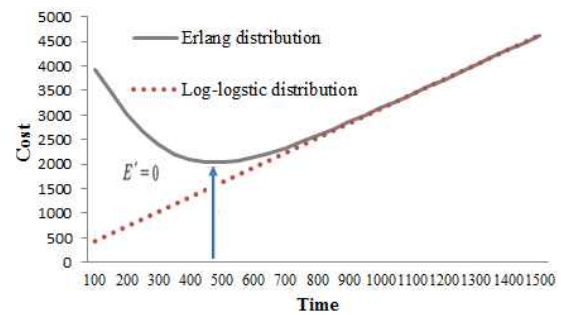
결국 비용 곡선의 추세를 이용하여 최적의 소프트웨어 방출시간을 추정 할 수 있다. 이러한 상황은 가장 현실적인 상황이며 대부분의 경우 실제 소프트웨어 개발의 과정에서 이러한 패턴을 가지게 된다[16]. 그러나 로그-로지스틱 모형의 경우에는 비용패턴이 선형추세를 보이고 있기 때문에 테스트의 초기 단계에서 소프트웨어 오

류에 대한 감지 및 제거 단계에서 오류를 제거하는 비용은 총비용에 영향을 주지 못하기 때문에 최적의 소프트웨어 방출시간을 추정 할 수 없고 단지 시간이 흐름에 따라 지속적으로 비용 상승효과 정보만 얻을 수 있다.

(가정 2) $E_1 = 300\$, c_2 = 3\$, c_3 = 4\$, c_4 = 150\$, t' = 600$

(가정 1)에서 다른 가정은 동일하고 소프트웨어 시스템을 출시 한 후 운영 및 소프트웨어를 유지할 수 있는 시간 t' 을 300에서 600으로 증가한 경우는 (그림 3)에 나타내었다. (그림 2)와 (그림 3)을 비교 하였을 때 소프트웨어 시스템을 출시 한 후 운영 및 소프트웨어를 유지할 수 있는 시간이 증가하였기 때문에 역시 소프트웨어 최적 방출시간이 연기됨을 알 수 있다. 따라서 이 경우에도 소프트웨어 방출시기 이후에 결함을 감소시킬 수 있도록 운영단계보다 테스트 단계에서 가능한 결함들을 제거해야 한다. 그러나 로그-로지스틱 모형의 경우에는 비용패턴이 선형추세에서 기울기가 증가하기 때문에 비용이 증가함을 알 수 있다.

본 연구에서 사용된 어랑 분포와 로그-로지스틱분포에 근거한 소프트웨어 비용 모델을 비교한 결과 어랑 모형은 최적의 소프트웨어 방출 시간을 예측 할 수 있지만 로그-로지스틱 모형은 방출시간을 예측 할 수 없기 때문에 로그-로지스틱 보다 어랑 모형이 보다 효율적으로 나타나고 있다



[Fig. 3] The curve under the condition of (Assumption 2)

5. 결 론

소프트웨어 신뢰도 성장 모델은 최적의 소프트웨어

방출 시간과 테스트 작업의 비용을 예상 할 수 있다.

따라서 보다 효율적인 모델은 테스트 비용을 줄이고 소프트웨어를 방출 이익을 증가 시킬 수 있도록 해야 한다. 본 연구에서 사용된 감마형(어랑 분포)과 로그형(로그-로지스틱분포)에 근거한 소프트웨어 비용 모델을 비교한 결과 감마형 모형은 최적의 소프트웨어 방출 시간을 예측 할 수 있지만 로그형 모형은 방출시간을 예측 할 수 없기 때문에 로그형 보다 감마형 모형이 보다 효율적으로 나타나고 있다. 제안된 감마형 모형의 경우에는 결합의 총수는 소프트웨어 방출이후 소프트웨어 운용기간 및 소프트웨어를 유지하는 동안 발견된 수이고 이 결합들은 소프트웨어 사용자가 전부 다 발견되지 않을 것이라는 가정을 한다. 그것은 실제 오류 수정 비용은 운용 단계에서 남아있는 모든 오류를 제거하는 비용보다 낮다는 것을 알 수 있다. 따라서 최적의 소프트웨어 방출 시간은 현실적으로 미리 예측해 볼 수 있다.

대용량 소프트웨어가 수정과 변경하는 과정에서 결합의 발생을 거의 피할 수 없는 상황이 현실이다. 신뢰성 요구를 만족하고 총비용을 최소화하는 상황이 최적방출 시간이다. 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 경제적 개발 비용을 파악 하는데 어느 정도 도움을 줄 수 있으리라 사료된다.

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 measures, *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] Shyur 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, *J. Syst. 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 Delayed S-Shaped and Extreme Value Distribution Software Reliability Model using the Perspective of Learning Effects", *International Journal of Advancements in Computing Technology(IJACT)* Vol. 5, No. 9, pp.1210-1218, 2013.
- [11] Hee-Cheul Kim, Yue-Soon Choi, Jong-Goo Park, "An Approach for the NHPP Software Reliability Model Using Erlang Distribution", *The journal of the Korea Institute of Maritime Information & Communication Sciences* v.10 no.1 , pp.7 - 14, 2006.
- [12] Kyung-Soo Kim, Hee-Cheul Kim, "The Comparative Study of Software Optimal Release Time of Finite NHPP Model Considering Property of Nonlinear Intensity Function", *The Journal of Digital Policy & Management*, 11(1): 159-166, Sep, 2013.
- [13] Kim, Hee Cheul, "The Comparative Study of Software Optimal Release Time of Finite NHPP

Model Considering Half-Logistic and Log-logistic Distribution Property", The Journal of Korea Society of Digital Industry and Information, 9(2): 1-10, June, 2013.

- [14] L. Kuo and T. Y. Yang., Bayesian Computation of Software Reliability, Journal of the American Statistical Association, Vol.91, pp. 763-773, 1996.
- [15] Ye Zhang and Kaigui Wu, Software Cost Model Considering Reliability and Time of Software in Use, Journal of Convergence Information Technology(JCIT), Volume 7, Number 13, pp. 135-142, 2012.
- [16] Kyung-Soo Kim, Hee-Cheul Kim, "The Comparative Software Cost Model of Considering Logarithmic Fault Detection Rate Based on Failure Observation Time", The Journal of Digital Policy & Management, 11(11): 335-342, Nov, 2013.
- [17] 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
- [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.

김 경 수(Kim, Kyung Soo)



- 2001년 8월 : 순천향대학교 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