

지수 및 비지수족 분포 모형에 근거한 소프트웨어 최적방출시기에 관한 비교 연구

김희철*, 신현철**

The Comparative Study of Software Optimal Release Time Based on Gamma Exponential and Non-exponential Family Distribution Model

Kim Hee Cheul*, Hyun-Cheul Shin**

요 약

본 연구에서는 소프트웨어 제품을 개발하여 테스트를 거친 후 사용자에게 인도하는 시기를 결정하는 방출문제에 대하여 연구 하였다. 방출문제의 분포는 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생할 가능성이 있는 무한고장수를 가진 비동질적인 포아송 과정에 기초하였다. 수명분포는 지수 족 분포와 비지수족 분포를 이용한 최적 방출시기에 관한 문제를 비교 제시하였다. 이러한 지수 및 비지수분포에 근거하여 소프트웨어 요구 신뢰도를 만족시키고 소프트웨어 개발 및 유지 총비용을 최소화 시키는 최적 소프트웨어 방출 정책에 대하여 논의 되었다. 본 논문의 수치적인 예에서는 고장 간격 시간 자료를 적용하고 모수추정 방법은 최우추정법을 이용하고 추세분석을 통하여 자료의 효율성을 입증한 후 최적 방출시기를 추정하였다.

Abstract

Decision problem called an optimal release policies, after testing a software system in development phase and transfer it to the user, is studied. The applied model of release time exploited infinite non-homogeneous Poisson process. This infinite non-homogeneous Poisson process is a model which reflects the possibility of introducing new faults when correcting or modifying the software. The failure life-cycle distribution used exponential and non-exponential family which has various intensity. Thus, software release policies which minimize a total average software cost of development and maintenance under the constraint of satisfying a software reliability requirement becomes an optimal release policies. In a numerical example, after trend test applied and estimated the parameters using maximum likelihood estimation of inter-failure time data, estimated software optimal release time.

• 제1저자, 교신저자 : 김희철

• 투고일 : 2010. 03. 25, 심사일 : 2010. 04. 19, 게재확정일 : 2010. 04. 29.

* 남서울대학교 산업경영공학과 교수 ** 백석문화대학 인터넷정보학부 교수

▶ Keyword : 소프트웨어 방출정책(Software Release Policies), 기록 값 통계량(Record Value Statistic), 지수 및 비지수족(Exponential and Non-exponential family)

I. 서론

소프트웨어 신뢰성은 컴퓨터 시스템에 대한 적용과 이에 대한 연구 분야에서 중요한 역할을 담당해 오고 있다. 소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 유발 할 수 도 있다. 따라서 소프트웨어 신뢰도는 현대의 소프트웨어 생산품 개발에서 중요한 분야 가운데 하나이다.

소프트웨어 신뢰성 엔지니어링에서의 연구 활동은 지난 30년 전부터 행해져 오고 있고 많은 신뢰도 성장 모형들이 소프트웨어에 남아 있는 고장들의 수와 소프트웨어 신뢰도의 추정에 관한 문제들을 제안해 왔다. 일반적으로 소프트웨어 개발과정은 설계단계, 디자인, 코딩 그리고 테스트 단계를 거친다. 이러한 과정을 거친 후 소프트웨어 제품을 방출하게 되는데 방출이후에 발견되지 않은 고장들이 나타난다면 이것들에 대한 보전 비용(Maintenance cost)은 크게 증가 할 것이다. 결국, 소프트웨어 시스템 시험을 끝내고 그것을 사용자에게 넘기는 시기 결정은 매우 중요한 사항이 된다.

소프트웨어 방출시간에 대한 연구들은 대부분 유한 고장 NHPP(Non-Homogeneous Poisson Process)모형을 사용하였다[1][2][3][4]. 이러한 유한(finite)고장 NHPP모형은 소프트웨어가 유한개의 고장이 있고 고장 제거 단계에서는 새로운 고장이 발생하지 않는다는 가정을 한 모형이다. 그러나 실제 고장 제거 단계에서도 새로운 고장이 발생 할 수 있다. 따라서 본 연구에서는 새로운 고장도 고려하는 무한(infinite) 고장 NHPP 모형을 이용하여 최적 방출시기에 대한 문제를 제안하고자 한다. 이 분야에서는 Musa-Okumoto의 대수 포아송 실행시간 모형[5][6] 과 로그-파우어 모형[7][8]을 이용한 방출 문제에 대한 문제 들이 이미 연구되었고 최근에도 이와 관련된 문제에 대한 연구는 Yang 과 Xie(2000)와 Huang(2005)에 의해 연구되고 있다[9][10][11]. 본 연구에서는 소프트웨어의 결함을 제거하거나 수정 작업 중에도 새로운 결함이 발생할 가능성이 있는 무한고장수를 가진 비동질적인 포아송 과정에 기초하고 수명분포는 신뢰성 분야 및 모든 분야에 잘 적용되는 지수족 분포(어랑분포 와 랄리분포)와 효율적인 분포로 알려진 비지수족 분포(버르 분포 및 로그-로지스틱분포)를 이용한 최적 방출시기에 관한 문제를 비교 제시 하였다.

II. 관련 연구

1. 지수 족 분포

확률밀도함수가 다음과 같은 형태로 표시되면 지수 족(Exponential family) 분포라고 한다[12].

$$f(x|\theta) = h(x) c(\theta) \exp\left(\sum_{i=1}^k w_i(\theta) t_i(x)\right) \dots\dots\dots (1)$$

단, $h(x) \geq 0$ 이고 $t_1(x), \dots, t_k(x)$ 는 θ 에 의존하지 않는 관찰 값 x 을 의미한다.

이러한 지수 족 분포는 수학적 특성 때문에 모든 분야에서 데이터의 정보를 가장 잘 반영하는 분포들로 알려져 있다.

본 논문에서는 일반적으로 신뢰성 분야에서 많이 사용하는 감마분포에서 형상모수가 2인 Erlang 분포와 와이블 분포에서 형상모수가 2인 Rayleigh분포를 적용하고자 한다.

Erlang 분포와 Rayleigh분포의 확률밀도함수는 각각 다음과 같다[1].

$$f_{Erl}(t|b_1) = b_1^2 t \exp(-b_1 t) \dots\dots\dots (2)$$

$$f_{Ray}(t|b_2) = 2b_2(2t)^{b_2-1} e^{-(2t)^{b_2}} \dots\dots\dots (3)$$

2. Erlang 분포와 Rayleigh분포를 이용한 NHPP

NHPP 모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도 함수(Intensity function) $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[13][14].

$$m(t) = \int_0^t \lambda(s) ds, \quad \frac{dm(t)}{dt} = \lambda(t) \dots\dots\dots (4)$$

따라서 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률 밀도 함수(Probability density function)로 알려져 있다. 즉,

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, \quad n = 0, 1, 2, \dots, \infty \dots\dots\dots (5)$$

이처럼 시간 관련 모형(Time domain models) 들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이러한 모형들은 고장 강도 함수 $\lambda(t)$ 가 다르게 표현됨으로서 평

균값 함수 $m(t)$ 도 역시 다르게 나타나고 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다[10].

수리 시점에서도 고장이 발생할 수도 있는 상황을 반영하기 위하여 RVS(Record Value Statistics)모형(무한 고장)을 사용하는 NHPP 모형의 평균값함수는 다음과 같이 알려져 있다[13][14].

$$m(t) = -\ln(1 - F(t)) \dots\dots\dots (6)$$

따라서 (4)식 과 (6)식을 연관시키고 $f(t)$ 을 확률밀도함수, $F(t)$ 을 분포함수라고 하면 다음과 같은 관계식에 의해 NHPP의 강도함수는 $F(t)$ 의 위험함수($h(t)$)가 된다.

$$\lambda(t) = m'(t) = f(t)/(1 - F(t)) = h(t) \dots\dots\dots (7)$$

Erlang 분포와 Rayleigh분포의 $\lambda(t)$ 와 $m(t)$ 는 <표 1>에 요약되었다 [1].

표 1. 지수 족 모형에 대한 강도함수와 평균값 함수
Table 1. $\lambda(t)$ and $m(t)$ of exponential family model

Model	$\lambda(t)$	$m(t)$
Erlang	$b_1^{-2} t / (b_1 t + 1)$	$b_1 t - \ln(1 + b_1 t)$
Rayleigh distribution	$2 b_2^2 t$	$(b_2 t)^2$

한편, NHPP모형에 대한 우도함수는 다음과 같이 알려져 있다[1][2].

$$L_{NHPP}(\theta | \underline{x}) = \prod_{i=1}^n \lambda(x_i) e^{-m(t)} \dots\dots\dots (8)$$

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

따라서 Erlang 분포모형의 최우추정치를 추정하기 위한 로그 우도함수는 다음과 같다,

$$\ln L(b_1 | \underline{x}) = 2n \ln b_1 - \sum_{i=1}^n \ln(b_1 x_i + 1) + \sum_{i=1}^n \ln x_i - b_1 x_n + \ln(1 + b_1 x_n) \dots\dots\dots (9)$$

(9)식을 이용하여 b_1 에 대하여 미분하고 그 값을 0으로 하는 최우추정치 \hat{b}_{1MLE} 을 추정 할 수 있다.

유사한 방법으로 Rayleigh분포 모형의 최우추정치를 추정

하기 위한 우도함수는 다음과 같이 유도된다[1].

$$\ln L(b_2 | \underline{x}) = n \ln 2 + 2n \ln b_2 + \sum_{i=1}^n \ln x_i - (b_2 x_n)^a \dots\dots\dots (10)$$

(10)식을 이용하여 최우추정치 $\hat{\beta}_{2MLE}$ 는 다음과 같이 구할 수 있다[5, 11]

$$\hat{b}_{2MLE} = \left(\frac{n}{x_n^2} \right)^{\frac{1}{2}} \dots\dots\dots (11)$$

3. Burr분포와 Log-Logistic분포를 이용한 NHPP

Burr 분포[15] 분포는 1942년 Burr에 의해 만들어진 분포이고 이 분포의 특징은 왜도(Skewness)와 첨도(Kurtosis) 측면에서 보다 넓은 범위에 분포하기 때문에 어떠한 수명 단봉자료(Unimodal data)에서도 적용할 수 있는 장점을 가진 분포라고 알려져 있다[16]. 이 분포의 확률밀도 함수와 분포함수는 다음과 같이 표현됨이 알려져 있다.

$$f(t | r, \beta_1) = k \beta_1 t^{r-1} / (1 + t^r)^{\beta_1+1} \dots\dots\dots (12)$$

$$F(t | r, \beta_1) = 1 - (1 + t^r)^{-\beta_1} \dots\dots\dots (13)$$

단, $t > 0, r > 0, \beta_1 \geq 1$.

이러한 Burr 분포에 대한 연구는 많은 학자들에 의하여 연구되었는데 그 중에서 Austin[16]은 이 분포를 관리도에 적용하였으며 Dubey[17]는 고장 시간 모형으로서의 Burr 분포의 장점을 고찰 하였고 Tadikamalla[18]는 다른 분포와의 관계를 연구하였고 Upadhyay 와 Javed, Peshwani[19]은 베이지안적 접근 방법을 동원하여 4모수에 대한 모수 추정방법을 연구하기도 하였다. Burr 분포에 대한 NHPP의 강도함수와 평균값 함수는 <표 2>에 요약되었다.

따라서 Burr 분포 모형에 대한 우도함수는 다음과 같이 유도 할 수 있다[15].

$$L(r, \beta_1 | \underline{x}) = \left(\prod_{i=1}^n \frac{r \beta_1 x_i^{r-1}}{1 + x_i^r} \right) \cdot (1 + x_n^r)^{-\beta_1} \dots\dots\dots (14)$$

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

최우추정법(MLE)을 이용하기 위한 Burr(r, β_1) 모형 로 그 우도 함수는 (14)식과 관련하여 다음과 같이 유도된다.

$$\ln L(r, \beta_1 | \underline{x}) = n \ln k + n \ln \beta_1 + (r-1) \sum_{i=1}^n \ln x_i \dots\dots\dots (15)$$

$$- \sum_{i=1}^n \ln(1+x_i^r) - \beta_1 \ln(1+x_n^r)$$

(15)식에서 k 와 β_1 에 대하여 편미분 하여 다음과 같은 식을 만족하는 \hat{r}_{MLE} 와 $\hat{\beta}_{1MLE}$ 을 수치 해석적 방법으로 계산할 수 있다[11].

$$\frac{\partial \ln L(r, \beta_1 | \underline{x})}{\partial r} = \dots\dots\dots (16)$$

$$\frac{n}{r} + \sum_{i=1}^n \ln x_i - \sum_{i=1}^n \frac{x_i^r \ln x_i}{1+x_i^r} - \frac{\beta_1 x_n^r \ln x_n}{1+x_n^r} = 0$$

$$\frac{\partial \ln L(r, \beta_1 | \underline{x})}{\partial \beta_1} = \frac{n}{\beta_1} - \ln(1+x_n^r) = 0 \dots\dots\dots (17)$$

표 2. 비지수족 모형에 대한 강도함수와 평균값 함수
Table 2. $\lambda(t)$ and $m(t)$ of non-exponential family model

Model	$\lambda(t)$	$m(t)$
Burr	$\frac{r \beta_1 t^{r-1}}{1+t^r}$	$\beta_1 \ln(1+t^r)$
Log-Logistic distribution	$\frac{\tau^k k t^{k-1}}{1+(\tau t)^k}$	$\ln(1+(\tau t)^k)$

로그-로지스틱 분포(Log-Logistic)[3][13]모형은 단위 당 고장발생률(위험함수)이 증가하다가 감소하는 속성을 가지고 있다. 이 분포의 확률밀도 함수와 분포함수는 다음과 같이 표현됨이 알려져 있다[13].

$$f(t|\tau, k) = (\lambda k(\tau t)^{k-1}) / [1+(\tau t)^k]^2 \dots\dots\dots (18)$$

$$F(t|\tau, k) = (\tau t)^k / [1+(\tau t)^k] \dots\dots\dots (19)$$

Log-Logistic 분포에 대한 NHPP의 강도함수와 평균값 함수는 <표 2>에 요약되었다

한편, 무한 NHPP 고장 우도 함수는 다음과 같다[3].

$$L(\tau, k | \underline{x}) = \left(\prod_{i=1}^n \frac{\tau^k k x_i^{k-1}}{1+(\tau x_i)^k} \right) \cdot [1+(\tau x_n)^k]^{-1} \dots\dots\dots (20)$$

최우추정법(MLE)을 이용하기 위한 Log-Logistic모형 로그 우도 함수는 (20)식과 관련하여 다음과 같이 유도된다.

$$\ln L(\tau, k | \underline{x}) = kn \ln \tau + n \ln k + (k-1) \sum_{i=1}^n \ln x_i \dots\dots\dots (21)$$

$$- \sum_{i=1}^n \ln(1+(\tau x_i)^k) - \ln[1+(\tau x_n)^k]$$

(21)식에서 τ 와 r 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\tau}_{MLE}$ 와 \hat{k}_{MLE} 을 수치 해석적 방법으로 계산할 수 있다[11]

$$\frac{\partial \ln L(\tau, k | \underline{x})}{\partial \tau} = \frac{kn}{\tau} - \sum_{i=1}^n \frac{x_i^k k \tau^{k-1}}{1+(\tau x_i)^k} - \frac{k \tau^{k-1} x_n^k}{1+(\tau x_n)^k} = 0 \dots\dots\dots (22)$$

$$\frac{\partial \ln L(\tau, k | \underline{x})}{\partial k} = n \ln \tau + \frac{n}{k} + \sum_{i=1}^n \ln x_i \dots\dots\dots (23)$$

$$- \sum_{i=1}^n \frac{(\tau x_i)^k (\ln \tau x_i)}{1+(\tau x_i)^k} - \frac{(\tau x_n)^k (\ln \tau x_n)}{1+(\tau x_n)^k} = 0$$

III. 요구 신뢰도 및 비용최소화를 고려한 방출시간

NHPP 모형에서 테스트 시점 x_n (마지막 고장시점)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[1][14].

$$\hat{R}(x | x_n) = \exp\left(-\int_{x_n}^{x_n+x} \lambda(\tau) d\tau\right) \dots\dots\dots (24)$$

$$= \exp[-\{m(x+x_n) - m(x_n)\}]$$

(24)식에서 Erlang 분포 모형에 대한 신뢰도는 $t = x_n$ 라고 하면 다음과 같이 표현된다.

$$R(x | T_R) = \exp[-b_1 x + \ln(1+b_1(T_R+x)) - \ln(1+b_1 T_R)] \dots\dots\dots (25)$$

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

$$\ln R_0 = -\beta x + \ln(1+\beta(T_R+x)) - \ln(1+\beta T_R) \dots\dots\dots (26)$$

비용 최소화와 관련된 최적 방출시간은 신뢰도와 비용모형에 의해서 결정된다. 소프트웨어 방출시간을 T 로 표현하고 $m(T)$ 와 $m(\infty)$ 을 각각 $(0, T]$ 와 $(0, \infty)$ 의 기간에 발견된 기대 고장수라고 표현하고 $C(T)$ 을 소프트웨어 라이프사이클(life cycle) 동안에 기대되는 소프트웨어 비용이라고 하면 $C(T)$ 는 다음과 같이 표현된다[8][10].

$$C(T) = c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \dots\dots\dots (27)$$

위 식에서 c_1 는 테스트링 동안에 하나의 고장을 수리하는 비용이고 c_2 가동 중에 하나의 고장을 수리하는 비용($c_2 > c_1$), 그리고 c_3 는 단위 시간당 테스트링 비용을 나타낸다. 이와 관련하여 총비용의 최소화는 무한고장 평균값 함수를 가진 NHPP 모형에 대하여 발생 할 수 있다. 무한 수명에 대한 비용함수 $C(T)$ 식인 (27)식에서 $m(\infty)$ 은 직접 추정 할 수 없기 때문에 이 식을 사용하기 위해서는 소프트웨어 수명시간인 T_{LC} 을 지정하여 분석한다[8]. 이러한 T_{LC} 는 소프트웨어마다 서로 다른 임의의 값이기 때문에 유한 고장 NHPP 모형이라고 할 수는 없다.

따라서 비용함수를 고려하여 소프트웨어의 모든 수명에서 총비용을 최소화함으로써 최적 테스트링 시간을 결정 할 수 있고 다음과 같은 식을 만족하면 비용함수 $C(T)$ 는 유일한 최소값을 가진다[8][13].

$$\frac{\partial C(T)}{\partial T} = 0, \quad \frac{\partial^2 C(T)}{\partial^2 T} > 0 \dots\dots\dots (28)$$

결국 Erlang 분포에 의한 최소비용 관련 방출시간은 (6) 식과 (27)를 연관하면 다음과 같이 유도 된다.

$$C(T) = c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \dots\dots\dots (29)$$

$$= (c_1 - c_2) (b_1 T - \ln(1 + b_1 T))$$

$$+ c_2 (b_1 T_{LC} - \ln(1 + b_1 T_{LC})) + c_3 T$$

T 에 관해서 비용함수 $C(T)$ 을 미분하면 다음과 같은 방정식을 만족하는 최적방출시간 T_C 를 계산 할 수 있다.

$$(c_1 - c_2) \left(b_1 - \frac{b_1}{1 + b_1 T_C} \right) + c_3 = 0 \dots\dots\dots (30)$$

위 식에서 최적방출시간은 소프트웨어 지정 수명시간인 T_{LC} 와 의존하지 않는다는 것을 알 수 있다. 이러한 사실은 무한 고장 평균값 함수를 가진 NHPP모형들이 새로운 결점들이 발생함으로써 몇 개의 고장이 야기 될 수 있는 점을 고려한 모형으로 적합 시킬 수 있다[5][14]. 실제로 만족할 만한 신뢰도가 부여되고 동시에 시스템 고장과 연계된 기대 총비용을 최소화시키기 위하여 필요하다면 충분한 테스트링을 계속해야 한다. 따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다. 따라서 Erlang 분포모형

을 사용한 최적 방출시간 T_{OP} 는 T_R 과 T_C 에 대하여 다음을 만족한다[8].

$$T_{OP} = \text{Max}(T_C, T_R) \dots\dots\dots (31)$$

$$\text{단, } \ln R_0 = -b_1 x + \ln(1 + b_1(T_R + x)) - \ln(1 + b_1 T_R)$$

$$(c_1 - c_2) \left(b_1 - \frac{b_1}{1 + b_1 T_C} \right) + c_3 = 0$$

유사한 방법으로 다른 모형에 대한 최적 방출시간 T_{OP} 을 추정하기 위한 T_R 과 T_C 의 관련 식은 <표 3>에 요약 되었다.

표 3. 각 모형에 대한 T_R 와 T_C
Table 3. T_R and T_C for each model

Model	T_R 과 T_C
Erlang	$\ln R_0 = -b_1 x + \ln(1 + b_1(T_R + x)) - \ln(1 + b_1 T_R)$ $(c_1 - c_2) \left(b_1 - \frac{b_1}{1 + b_1 T_C} \right) + c_3 = 0$
Rayleigh	$\ln R_0 = -b_2^2 [(x + T_R)^2 - (T_R)^2]$ $2(c_1 - c_2) b_2^2 T_C + c_3 = 0$
Burr	$\ln R_0 = -\beta_1 \ln(1 + (x + T_R)^r) + \beta_1 \ln(1 + T_R^r)$ $\frac{(c_1 - c_2) \beta_1 r T_C^{r-1}}{1 + T_C^r} + c_3 = 0$
Log-Logistic	$\ln R_0 = -\ln(1 + \{\tau(x + T_R)\}^k) + \ln(1 + (\tau T_R)^k)$ $\frac{(c_1 - c_2) k T_C^{k-1} \tau^k}{\ln(1 + (\tau T_C)^k)} + c_3 = 0$

IV. 수치적인 예

이 장에서 S27[20]가 인용한 고장 간격 시간 자료 (Failure interval time data)를 가지고 지수 족 분포 모형과 비지수족인 분포 모형에 근거한 최적 방출시기를 분석하고자 한다. 이 자료는 1197.945 시간단위에 41번의 고장이 발생된 자료이며 <표 4>에 나열 되어 있고 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[20].

추세 분석에는 일반적으로 라플라스 추세 검정(Laplace trend test)을 시행한다. 이 검정을 실시한 결과 <그림 1>에서 보여 주듯이 라플라스 요인(Factor)이 -2와 2사이에 존재함으로써 신뢰성장(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰도와 소프트웨어 방출시기를 추정하는 것이 가능하다[13][15][16].

각 모형에 대한 모수 추정은 최우추정법을 이용하였고 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기 값을 10^{-5} 와 10 을, 허용 한계(Tolerance for width of interval)는 10^{-10} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다[13]. 따라서 모수 추정의 결과는 <표 5>에 요약되었다.

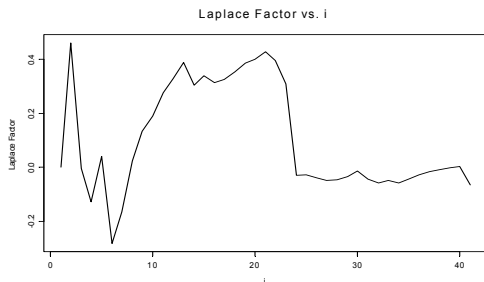


그림 1. 라플라스 추세 검정
Fig. 1. Laplace trend test

표 4. 고장 간격자료
Table 4. Failure Interval data

Failure number	Failure Time(hours)	Failure Interval (hours)
1	5.649	5.649
2	8.92	3.271
3	20.29	11.37
4	29.955	9.665
5	34.715	4.76
6	75.95	41.235
7	78.171	2.221
8	78.625	0.454
9	83.022	4.397
10	89.114	6.092
11	89.804	0.69
12	92.86	3.056
13	93.66	0.8
14	110.655	16.995
15	111.988	1.333
16	122.545	10.557
17	127.045	4.5
18	128.712	1.667
19	128.99	0.278
20	131.768	2.778
21	131.829	0.061
22	141.712	9.883

23	164.212	22.5
24	342.85	178.638
25	356.144	13.294
26	399.144	43
27	446.494	47.35
28	476.644	30.15
29	497.144	20.5
30	497.661	0.517
31	591.161	93.5
32	665.644	74.483
33	686.444	20.8
34	765.944	79.5
35	772.977	7.033
36	774.944	1.967
37	791.561	16.617
38	815.978	24.417
39	837.145	21.167
40	861.945	24.8
41	1197.945	336

표 5. 각 모형의 모수 추정값
Table 5. Estimation of each model

Model		MLE
Erlang		$\hat{b}_1 = 0.068428$
Rayleigh		$\hat{b}_2 = 0.005345$
Burr	$r = 2$	$\hat{\beta}_1 = 2.892064$
Log-Logistic	$k = 2$	$\hat{\tau} = 0.000069$

$c_1 = 5(\$)$, $c_2 = 20(\$)$ 그리고 $c_3 = 0.5(\$)$ 라고 가정하고 시스템 수명시간은 2000시간이고 임무시간 x 을 1.5이고 R_0 을 0.95(95%)를 투입하여 각 모형에 대한 추정시간의 결과와 최적방출시간은 <표 6>에 요약되었다. 이 표에서 지수 즉 모형인 Erlang 모형은 비용 및 신뢰도 측면에서 약간 차이는 있지만 다른 모형에 비해 모두 유사한 방출시간을 나타내고 있다 그러나 Rayleigh 모형은 신뢰측면에서는 아주 좋으나 비용 측면의 방출시간은 아주 늦게 나타나 따라서 Rayleigh 모형을 사용하고자 하면 비용측면에 대한 관리가 절실히 필요하다. 비지수 즉 분포이고 넓은 범위에 분포하는 장점을 가진 Burr분포모형은 비용 및 신뢰도 측면에서 비교적 방출시간을 나타내고 있지만 비교모형 중에서 방출시간이 가장 느려 비효율적인 모형으로 나타나고 있다.

표 6. 최적 방출시간 계산($R_0 = 0.95$)
 Table 6. Calculation for optimal release time ($R_0 = 0.95$)

Model		추정시간	최적방출 시간(T_{OP})
Erlang		$\hat{T}_R = 291.9701$ $\hat{T}_C = 112.7180$	291.9701
Rayleigh		$\hat{T}_R = 1.9053$ $\hat{T}_C = 583.3824$	583.3824
Burr	$r = 2$	$\hat{T}_R = 573.6543$ $\hat{T}_C = 654.3781$	654.3781
Log-Logistic	$k = 2$	$\hat{T}_R = 99.4723$ $\hat{T}_C = 60.000518$	99.4723

단위당 고장 발생률(위험함수)이 증가하다가 감소하는 속성을 가지고 있는 비지수족인 Log-Logistic 분포모형은 비용 및 신뢰도 측면에서 안정되어 있고 최적 방출시간도 다른 비교 모형보다 방출시간이 빨라서 효율적인 모형으로 나타나고 있다. 결국 지 수족 모형은 일반적인 비지수족 모형보다는 효율적 일수 있지만 로그형 분포모형 보다는 비효율적 분포로 나타나고 있다. 이 결과는 적합 시킨 자료에 의한 결과이기 때문에 다른 자료를 적합 시키면 결과는 달라질 수 있지만 로그형 및 지수 족 분포가 이 분야에서 효율적 모형으로 선택 될 수 있음을 보여주고 있다.

V. 결 론

본 연구는 형상모수를 고정시킨 지 수족 분포와 비지수족 분포 모형을 적용한 무한고장 NHPP 모형을 이용하여 최적 방출시기에 관한 문제를 알아보았다. 즉, 대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피 할 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 최적 방출 시간이다. 본 연구에서는 지수족 모형은 일반적인 비지수족 모형보다는 효율적 일수 있지만 로그형 분포모형 보다는 비효율적 분포로 나타나고 있다. 이 결과는 적합 시킨 자료에 의한 결과이기 때문에 다른 자료를 적합 시키면 결과는 달라질 수 있지만 로그형 및 지수 족 분포가 이 분야에서 효율적 모형으로 선택 될 수 있음을 보여주고 있다. 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 있으리라 사료 된다.

참고문헌

- [1] Lawless, J. F., "Statistical Models and Methods for Lifetime Data," John Wiley & Sons, New York, 1981.
- [2] 김희철, "일반화감마분포를 이용한 NHPP 소프트웨어 신뢰도 모형에 관한 연구," 한국컴퓨터정보학회 논문지, 제 10권, 제6호, 27-35쪽, 2005년 12월.
- [3] Gokhale, S. S. and Trivedi, K. S. "A time/structure based software reliability model," Annals of Software Engineering, 8, pp. 85-121, 1999.
- [4] 김희철, "지수화 지수 분포에 의존한 NHPP 소프트웨어 신뢰성장 모형에 관한 연구," 한국컴퓨터정보학회논문지, 제11권, 제5호, 9-18쪽, 2006년 11월.
- [5] Musa, J. D. and Okumoto, K., "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement," Proceeding the 7th International Conference on Software engineering, pp. 230-238, 1984.
- [6] 김희철, 신현철 "포아송 실행시간 모형에 의존한 소프트웨어 최적방출시기에 대한 베이지안 접근 방법에 대한 연구," 한국 컴퓨터정보학회 논문지, 제14권 제7호, pp. 27-35, 2009년 7월.
- [7] Almering, V. and Genuchten, M. V and Cloudf, G. and Sonnemans, P. J. M., "Using Software Reliability Growth Models in Practice," IEEE SOFTWARE, pp. 82-88, 2007.
- [8] 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.
- [9] Yang, B. and Xie, M., "A study of operational and testing reliability insoftware reliability analysis," RELIABILITY ENGINEERING & SYSTEM SAFETY, Vol. 70, pp. 323-329, 2000.
- [10] Huang, C. Y., "Cost-Reliability-optimal release

policy for software reliability models incorporating improvements in testing efficiency”, The journal of Systems and software. Vol. 77, pp. 139-155, 2005.

- [11] 김희철, “극값 분포 특성을 가진 소프트웨어 신뢰성 보증 모형에 관한 연구,” 한국 통신학회논문지, 제34권 제6호, 623-629쪽, 2009년 6월.
- [12] Casella, G and Berger, R. L. “*Statistical Inference*,” Duxbury Press, pp.1123-115,1990.
- [13] 김희철, “Log-Logistic 분포모형에 근거한 소프트웨어 취약방출시기에 관한 비교연구,” 한국컴퓨터정보학회 논문지, 제13권, 제7호, 1-9쪽, 2008년 12월.
- [14] Kuo, L. and Yang, T. Y, “Bayesian Computation of Software Reliability,” Journal of the American Statistical Association, Vol. 91, pp. 763-773, 1996.
- [15] 김희철, “Burr 분포모형에 근거한 소프트웨어 방출시기에 관한 연구,” 정보, 보안 논문지, 제 9권, 제 1호, 84-93쪽, 2009년 3월.
- [16] J. A. Austin, “Control chart constants for largest and smallest in sampling from a normal distribution using the generalized Burr estimation”, Technometrics, Vol. 15, pp. 931-933, 1971.
- [17] S. D. Dubey, “Statistical treatment of certain life testing and reliability problems,” ARL TR pp. 73-0155, AD 774537, 1973.
- [18] P. K. Tadikamalla, “A look at the Burr and related distributions,” Inter. Statist. Rev., 48, pp. 337-344, 1980
- [19] S. K. Upadhyay and I. A. Javed and M. Peshwani, “Bayesian analysis of generalized four-parameter Burr distribution via Gibbs sampler,” METRON-International Journal of statistics, Vol, LXII, n.1, pp. 115-135, 2004.
- [20] K. Kanoun and J. C. Laprie, “*Handbook of Software Reliability Engineering*,” M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY: 1996; pp.401-437.

저 자 소 개



김 희 철
 1992년 : 동국대학교 통계학과 석사
 1998년 : 동국대학교 통계학과 박사
 2000년 3월 ~ 2004년 2월 :
 송호대학 정보산업계열 교수
 2005년 3월~현재 : 남서울대학교
 산업경영공학과 교수
 관심분야 : 소프트웨어 신뢰성공학,
 웹프로그래밍, 전산통계,
 해외투자 및 쇼핑몰



신 현 철
 2002년 : 원광대학교 컴퓨터공학과
 (공학박사)
 1994년 ~ 현재 :
 백석문화대학 인터넷정보학부 교수
 2005 ~ 현재 : 한국정보처리학회 이사
 2005 ~ 현재 : 한국사이버테러정보
 전학회 부회장
 관심분야 : 통신공학, 컴퓨터과학,
 소프트웨어공학