

Burr분포 학습 효과 특성을 적용한 소프트웨어 신뢰도 모형에 관한 연구

김대성¹, 김희철^{2*}

¹경기과학기술대학교 e-비즈니스과, ²남서울대학교 산업경영공학과

The Study of Software Reliability Model from the Perspective of Learning Effects for Burr Distribution

Dae-Soung Kim¹ and Hee-Cheul Kim^{2*}

¹Division of e-Business, Gyeonggi College of Science and Technology

²Division of Industrial and Management Engineering, Namseoul University

요 약 본 연구에서는 소프트웨어 제품을 개발하여 테스트를 하는 과정에서 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 효율적인 학습기법을 이용한 NHPP 소프트웨어 모형에 대하여 연구 하였다. 적용분포는 버르 분포를 적용한 유한고장 NHPP에 기초하였다. 소프트웨어 오류 탐색 기법은 사전에 알지 못하지만 자동적으로 발견되는 에러를 고려한 영향요인과 사전 경험에 의하여 세밀하게 에러를 발견하기 위하여 테스트 관리자가 설정해놓은 요인인 학습효과와 특성에 대한 문제를 비교 제시 하였다. 그 결과 학습요인이 자동 에러 탐색요인보다 큰 경우가 대체적으로 효율적인 모형임을 확인 할 수 있었다. 본 논문의 수치적인 예에서는 고장 간격 시간 자료를 적용하고 모수추정 방법은 최우추정법을 이용하여 추세분석을 통하여 자료의 효율성을 입증한 후 평균자승오차와 R^2 (결정계수)를 이용하여 효율적인 모형을 선택 비교하였다.

Abstract In this study, software products developed in the course of testing, software managers in the process of testing software test and test tools for effective learning effects perspective has been studied using the NHPP software. The Burr distribution applied to distribution was based on finite failure NHPP. Software error detection techniques known in advance, but influencing factors for considering the errors found automatically and learning factors, by prior experience, to find precisely the error factor setting up the testing manager are presented comparing the problem. As a result, the learning factor is greater than automatic error that is generally efficient model could be confirmed. This paper, a numerical example of applying using time between failures and parameter estimation using maximum likelihood estimation method, after the efficiency of the data through trend analysis model selection were efficient using the mean square error and R^2 .

Key Words : Learning Effects, Burr Distribution, Time Truncated Model,

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 이나 디버깅

(Debugging) 측면에서 비용을 줄이기 위해서는 소프트웨어의 신뢰성의 변동과 테스트 비용을 사전에 알고 있어야 효율적이다. 따라서 신뢰도, 비용 및 방출 시간의 고려 사항을 가진 소프트웨어 개발 과정은 필수 불가결 한다. 결국 소프트웨어 제품의 결합내용을 예측하기 위한 모형 개발이 필요하다. 지금까지 많은 소프트웨어 신뢰성 모형이 제안 되었다. 이 중에서 비동질적 포아송 과정

*교신저자 : 김희철(kim1458@nsu.ac.kr)

접수일 11년 08월 30일

수정일 11년 09월 30일

게재확정일 11년 10월 06일

((Non-homogeneous Poisson process; NHPP)에 의존한 모형은 여러 탐색 과정측면에서는 우수한 모형이고 이 모형은 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

Gokhale과Trivedi[1]은 고양된 비동질적인 포아송 과정 모형(Enhanced NHPP)을 제시하였고 Goel과 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 제안 하였다. 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 예측되는 기술을 제시하기도 하였다. 그리고 최근에는 S-형태 모형은 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 익숙해지는 학습 과정을 설명할 수 있다고 하였다[8].

따라서 본 연구는 Burr분포를 수명분포로 적용한 NHPP 소프트웨어 모형에 대하여 자동적으로 발견되는 에러를 고려한 영향요인(Influential factor)과 사전에 설정되는 학습요인(Learning factor)으로 구성된 학습효과와 특성에 대한 문제를 비교 제시 하였다.

2. 관련 연구

2.1 Burr 분포

버르분포는 1942년 Burr에 의해서 만들어진 분포이고 왜도(Skewness)와 첨도(Kurtosis) 측면에서 보다 넓은 범위에 분포하기 때문에 어떠한 수명 단봉자료(Unimodal data)에서도 적용할 수 있는 장점을 가진 이 분포의 확률 밀도 함수와 분포함수는 다음과 같이 알려져 있다[9].

$$f(t|a,b) = abt^{a-1} / (1+t^a)^{b+1} \tag{2.1}$$

$$F(t|a,b) = 1 - (1+t^a)^{-b} \tag{2.2}$$

그리고 위험함수는 다음과 같이 표현 가능하다.

$$h(t|a,b) = \frac{f(t|a,b)}{1-F(t|a,b)} = \frac{abt^{a-1}}{1+t^a} \tag{2.3}$$

단, $t > 0, a > 0, b \geq 1$.

이러한 Burr분포에 대한 연구는 많이 수행 되었다. Austin[10]은 관리도에 적용할 때 효율적 모형이 된다고 하였고 Dubey[11]는 고장시간 모형 측면에서의 장점을 고찰 하였고 Tadikamalla[12]으로 다른 분포와의 관계를 연구하였고 Upadhyay와 Javed, Peshwani[13]은 베이 지안적 접근방법을 동원하여 모수추정 방법을 연구하였다.

2.2 유한 고장 NHPP모형과 학습효과 특성

신뢰도에서 관측시간 $(0, t]$ 사이에 발견된 고장 수 $N(t)$ 을 모형 화 하는데 비동질적 포아송 과정(NHPP)이 널리 사용하여 왔다. 이 과정(Process)에서 강도함수(Intensity function) $\lambda(t) = dE[N(t)]/dt$ 은 t 에 대한 단조(Monotonic)함수로 흔히 가정 한다[14].

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

$$m(t) = \int_0^t \lambda(s)ds, \quad \frac{dm(t)}{dt} = \lambda(t) \tag{2.4}$$

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

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, \quad n = 0, 1, 2, \dots \tag{2.5}$$

이처럼 시간관련 모형(Time domain models)들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이 모형들은 유한 고장 모형과 무한 고장 범주로 분류한 다[15].

유한 고장(Finite failure) NHPP 모형들은 충분한 테스트 시간이 주어지면 결함들(Faults)의 기대 값이 유한 값 $(\lim_{t \rightarrow \infty} m(t) = \theta < \infty)$ 을 가진 고장 NHPP모형의 평균값 함수는 다음과 같이 표현 할 수 있다[9].

$$m(t) = \theta F(t) \tag{2.6}$$

(2.6)식으로 부터 강도함수(Intensity) $\lambda(t)$ 는 다음과 같이 유도된다.

$$\lambda(t) = \theta F'(t) \tag{2.7}$$

소프트웨어 에러들을 발견하기 위하여 자동 에러 탐색요인(Autonomous errors-detected factor) γ 과 학습요인(learning factor) η 이 포함된 영향요인들(Influential factors)이 고려될 수 있다. 따라서 $f(t)$ 을 t 시점에서 에러를 발견된 확률을 나타내는 강도함수이고 $F(t)$ 을 $(0, t]$ 시점까지의 누적함수라고 가정하면 영향요인들을 고려한 모형은 다음과 같이 표현 된다[8].

$$f(t) = (\gamma + \eta F(t)) (1 - F(t)) \tag{2.8}$$

단, $\gamma > 0, \eta > 0$.

(2.8)식에서 자동 에러 탐색요인 γ 은 사전에 알지 못하지만 테스트링 과정에서 테스트링 관리자가 자동적으로 에러를 발견하는 요인이지만 학습 요인 η 은 과거에 발견된 에러 패턴을 바탕으로 세밀하게 에러를 발견하기 위하여 테스트링 관리자가 설정해놓은 요인을 의미한다.

한편 (2.8)을 다음과 같이 위험함수 형태로 변경 할 수 있다.

$$h(t) = (\gamma + \eta F(t)) \tag{2.9}$$

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

(2.9)식에서 누적함수는 다음과 같이 수정 가능하다.

$$F(t) = \frac{h(t) - \gamma}{\eta} \tag{2.10}$$

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

$$x_k = \sum_{i=1}^k t_k \quad (k=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \tag{2.11}$$

이라고 하면 데이터 집합 D_i 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 구

성되며 이 시간 절단 모형에서의 θ 를 모수공간이라고 표시하면 우도함수는 다음과 같이 알려져 있다[9, 10].

$$L_{NHPP}(\theta | D_{x_n}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp(-m(x_n)) \tag{2.12}$$

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[2, 9].

$$\begin{aligned} \hat{R}(t | x_n) &= e^{-\int_{x_n}^{x_n+t} \lambda(\tau) d\tau} \\ &= \exp[-\{m(t+x_n) - m(x_n)\}] \end{aligned} \tag{2.13}$$

2.3 관측 자료에 대한 모형 비교

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

2.3.1 평균자승오차(Mean square error)

평균자승오차는 실제 관찰 값과 예측 값에 대한 차이를 측정하는 도구로서 다음과 같이 정의 된다.

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n-k} \tag{2.14}$$

단, $m(x_i)$ 은 시간 $(0, x_i]$ 까지 나타난 에러들의 누적합수를 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.

2.3.2 R^2

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

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left(m(x_i) - \sum_{j=1}^n m(x_j)/n \right)^2} \tag{2.15}$$

3. 제안한 학습효과를 이용한 Burr NHPP 모형

버르 분포[9, 10]의 위험함수는 특성을 유지하면서도 분포함수를 보다 용이하게 전개하기 위하여 본 연구에서는 (2.3)식에서 형상모수가 $a=1$ 인 경우인 다음과 같은 형태를 이용 하고자 한다.

$$h(t|1,b) = \frac{b}{1+t} \tag{3.1}$$

따라서 (2.10)식을 이용하면 다음과 같이 누적분포함수는 수정 가능하다.

$$F(t) = \frac{b-\gamma-\gamma t}{(1+t)\eta} \tag{3.2}$$

확률 밀도함수는 (2.8)식과 (3.2)식을 이용하면 다음과 같이 표현된다.

$$f(t) = \frac{b\{(1+t)(\eta+\gamma)-b\}}{\eta(1+t)^2} \tag{3.3}$$

따라서 유한 고장 NHPP모형의 평균값 함수와 강도함수는 (2.3)식과 (2.4)식을 이용하면 각각 다음과 같이 표현 할 수 있다.

$$m(t) = \theta F(t) = \theta \left(\frac{b-\gamma-\gamma t}{(1+t)\eta} \right) \tag{3.4}$$

$$\lambda(t) = \theta f(t) = \theta \left(\frac{b\{(1+t)(\eta+\gamma)-b\}}{\eta(1+t)^2} \right) \tag{3.5}$$

이경우의 우도함수는 (2.7)식에 (3.4)식과 (3.5)식을 대입하면 다음과 같다.

$$L_{NHPP}(\theta|D_{x_n}) = \prod_{i=1}^n \left[\theta \left(\frac{b\{(1+x_i)(\eta+\gamma)-b\}}{\eta(1+x_i)^2} \right) \right] \cdot \exp \left[-\theta \left(\frac{b-\gamma-\gamma x_n}{\eta(1+x_n)^2} \right) \right] \tag{3.6}$$

모수 추정방법은 최우추정법(MLE)을 사용하였고 최우추정법을 이용하기 위한 로그 우도 함수는 (3.6)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta|D_{x_n}) = n \ln \theta + n \ln b + \sum_{i=1}^n \ln \left(\frac{b\{(1+x_i)(\eta+\gamma)-b\}}{\eta(1+x_i)^2} \right) - \sum_{i=1}^n \ln \left(\eta(1+x_i)^2 \right) - \theta \left(\frac{b-\gamma-\gamma x_n}{\eta(1+x_n)^2} \right) \tag{3.7}$$

(3.7)식에서 θ 와 b 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 을 수치 해석적 방법으로 계산할 수 있다[10, 15].

$$\frac{\partial \ln L_{NHPP}(\theta|D_{x_n})}{\partial \theta} = \frac{n}{\theta} - \left(\frac{b-\gamma-\gamma x_n}{\eta(1+x_n)} \right) = 0 \tag{3.8}$$

$$\frac{\partial \ln L_{NHPP}(\theta|D_{x_n})}{\partial b} = \tag{3.9}$$

$$\frac{n}{b} - \sum_{i=1}^n \frac{1}{((1+x_i)(\eta+\gamma)-b)} - \frac{\theta}{\eta(1+x_n)} = 0$$

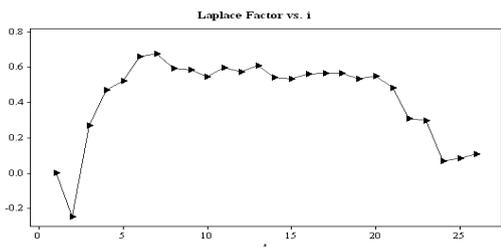
4. 수치적인 예

이 장에서는 NTDS(Naval Tactical Data System)에 의해 발생된 소프트웨어 고장간격자료[2, 16]를 적용하여 학습효과 특성을 분석하고자 한다. 이 자료는 250 시간단 위에 26번의 고장이 발생한 자료이며 표 1에 나열 되어 있다. 한편, 제시 하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세검정이 선행되어야 한다[17].

[표 1] 고장간격자료
[Table 1] Failure interval time

Failure number	Failure Interval Time	Failure Time	Failure Time $\times 10^{-4}$
1	9	9	0.0009
2	12	21	0.0021
3	11	32	0.0032
4	4	36	0.0036
5	7	43	0.0043
6	2	45	0.0045
7	5	50	0.0050
8	8	58	0.0058
9	5	63	0.0063
10	7	70	0.0070
11	1	71	0.0071
12	6	77	0.0077
13	1	78	0.0078
14	9	87	0.0087
15	4	91	0.0091
16	1	92	0.0092
17	3	95	0.0095
18	3	98	0.0098
19	6	104	0.0104
20	1	105	0.0105
21	11	116	0.0116
22	33	149	0.0149
23	7	156	0.0156
24	91	247	0.0247
25	2	249	0.0249
26	1	250	0.0250

추세분석에는 일반적으로 라플라스 추세검정(Laplace trend test)을 이용한다. 이 검정을 실시한 결과 그림 1에 나타내었다. 이 그림에서 라플라스 요인(Factor)이 -2와 2 사이에 존재함으로써 신뢰성장(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰도를 추정하는 것이 가능하다[17].

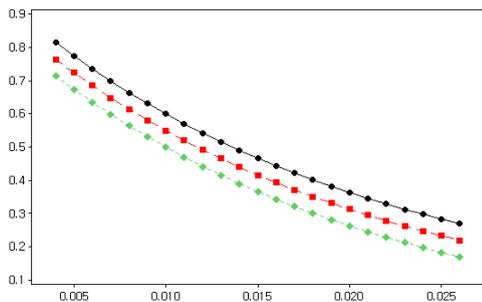


[그림 1] 라플라스 추세검정
[Fig. 1] Laplace trend test

또한 표 2에서는 이 자료에 대한 기초통계량이 요약되어 있다. 이 표에서 왜도와 첨도가 각각 4.21과 19.07로 나타나 비대칭 분포임을 보여주고 있다.

[표 2] 고장 간격자료의 기초통계량
[Table 2] Basic statistics of failure interval time

평균	표준편차	최소값	Q1	증위수
9.62	17.79	1	2	5.5
Q3	최대값	왜도	첨도	관측 수
9	91	4.21	19.07	26



● (η=0.7, γ=0.3) ■ (η=0.5, γ=0.5) ◆ (η=0.4, γ=0.6)

[그림 2] 임무시간에 대한 신뢰도
[Fig. 2] Reliability for mission time

신뢰성 모형의 모수 추정은 모수추정을 용이하게 하기 위하여 원 데이터를 변수변환(failure time × 10⁻⁴)하여

최우추정법을 이용하였다. 그리고 학습요인 η과 자동 에러 탐색요인 γ은 0.1부터 0.9까지 고정 시킨 비선형 방정식의 계산방법은 수치 해석적 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 1과 3을, 허용 한계(Tolerance for width of interval)는 10⁻¹⁰ 을 주고 수렴성을 모니터링하면서 충분한 반복 횟수인 100번을 C언어를 이용하여 모수 추정을 수행하였다. 각 모형에 대한 모수의 추정 값들의 결과와 MSE(평균자승오차) 값들이 표 3에 요약되었다.

[표 3] 각 모형의 모수 추정값
[Table 3] Parameter estimation of each model

Model		MLE		모형 비교	
η	γ	$\hat{\theta}_{MLE}$	\hat{b}_{MLE}	MSE	R ²
0.1	0.9	31.99049	1.005806	419.7419	0.81336
0.2	0.8	28.91977	1.004303	291.0853	0.81335
0.3	0.7	27.54029	1.007802	278.6586	0.81334
0.4	0.6	27.07625	1.008703	260.8151	0.81829
0.5	0.5	26.94113	1.007097	258.0348	0.81681
0.6	0.4	26.62777	1.010501	253.0840	0.81333
0.7	0.3	26.49463	1.011605	237.9513	0.81334
0.8	0.2	26.32404	1.014906	247.6003	0.81335
0.9	0.1	26.00840	1.024702	245.1847	0.81336

이 표에서 모든 경우의 결정계수 R²값이 0.8 보다 높기 때문에 실제 관찰된 고장수와 추정된 고장 수는 모두 잘 적용되어 설명력이 높다고 할 수 있다 즉 상관관계가 높다. 또한 η이 증가하고 γ이 감소하는 경우에 대체적으로 MSE 값들이 작게 나타나 점점 효율적인 모형으로 변해가고 있으나 η=0.7과 γ=0.3일 경우가 MSE 값들이 작게 나타나 가장 효율적인 모형으로 나타남을 확인할 수 있었다. 결국 학습요인 η 이 자동 에러 탐색요인 γ보다 클수록 효율적인 모형으로 나타나고 있다.

그림 2에서는 임무시간(Mission time)에 의존한 신뢰도($\hat{R}(t | x_{26}=0.025)$)을 η > γ(η=0.7, γ=0.3), η = γ(η=0.5, γ=0.5)와 η < γ(η=0.4, γ=0.6) 인 경우에 대하여 그려본 결과 단조 감소형태를 나타내지만 η이 증가할수록 신뢰도가 상승으로 나타나고 있다.

따라서 학습요인 η이 자동 에러 탐색요인 γ 보다 큰 경우가 대체적으로 효율적인 모형을 확인 할 수 있어서 학습요인을 고려하면 이 분야에서 효율적 모형으로 선택 될 수 있음을 보여주고 있다.

4. 결론

대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 수 없는 상황이 현실이다. 따라서 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 효율적인 학습 과정을 이용한 NHPP 소프트웨어 모형에 대하여 연구하였다. 사전에 알지 못하지만 자동적으로 발견되는 에러를 고려한 영향요인과 사전경험에 의하여 세밀하게 에러를 발견하기 위하여 테스트 관리자가 설정해놓은 요인인 학습효과의 특성에 대한 문제를 비교 제시하였다.

본 연구에서는 왜도와 첨도 측면에서 보다 넓은 범위에 분포하기 때문에 어떠한 수명 단분자료에서도 적용할 수 있는 장점을 가진 버르 분포를 유한 고장 NHPP 소프트웨어 수명분포로 적용하여 분석한 결과 학습요인이 자동 에러 탐색요인보다 큰 경우가 대체적으로 효율적인 모형임을 확인할 수 있어서 이 분야에서 효율적 모형으로 선택될 수 있음을 보여주고 있다. 경우에 따라서는 왜도와 첨도 측면에서 효율적인 카파분포, 지수화지수분포 등 업데이트된 분포에 대한 적용 문제를 비교 분석하는 연구도 가치 있는 일이라 판단되고 이 연구를 통하여 소프트웨어 개발자들은 수명분포에 의존한 학습요인을 파악하는데 어느 정도 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 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] Shyr 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] Hee-Cheul Kim and Jong-Goo Park "The study for NHPP Software Growth Model based on Burr Distribution", *THE KOREAN INSTITUTE OF MARITIME INFORMATION & COMMUNICATION SCIENCES*, Vol 11(3), pp. 514-522, 2007.

[10] I. W. Burr, "Cumulative frequency function", *Ann. Math. Statistic*. Vol 13, pp. 215-232, 1942.

[11] J, A. Austin, "Control chart constants for largest and smallest in sampling from a normal distribution using the generalized Burr estimation," *Technometrics*, Vol. 15, 1971, pp. 931-933.

[12] S. D. Dubey, "Statistical treatment of certain life testing and reliability problems", *ARL TR*, pp. 73-0155, AD 774537, 1973.

[13] P. K. Tadikamalla, "A look at the Burr and related distributions," *Inter. Statist. Rev.* 48, 1980, pp.337-344.

[14] 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, 2004, pp.115-135.

[15] J. F. Lawless. *Statistical Models and Methods for Lifetime Data*. John Wiley & Sons, New York, 1981.

[16] L. Kuo and T. Y. Yang. "Bayesian Computation of Software Reliability". *Journal of the American Statistical Association*, Vol.91, pp. 763-773, 1996.

[17] K. Kanoun and J. C. Laprie, "Handbook of Software Reliability Engineering", M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY: 1996; p.401-437.

김대성(Dae-Soung Kim)

[정회원]



- 1985년 2월 : 강원대학교 물리학과 졸업(이학석사)
- 1995년 8월 : 미국 Georgia State University 졸업(Ph. D)

<관심분야>

e-비즈니스, Atomic and molecular physics

김 희 철(Hee-Cheul Kim)

[정회원]



- 1992년 2월 : 동국대학교 통계학과 졸업(이학석사)
- 1998년 8월 : 동국대학교 통계학과 졸업(이학박사)

<관심분야>

전산통계, 소프트웨어 신뢰성 공학