

유한 및 무한고장 지수파우어 NHPP 소프트웨어 신뢰성모형에 대한 비교 연구

김희철*, 신현철**

The Comparative Study for Software Reliability Model Based on Finite and Infinite Failure Exponential Power NHPP

Hee-Cheul Kim*, Shin-Hyun Cheul**

요약 소프트웨어 고장분석을 위한 비동질적인 포아송과정에서 결함당 고장발생률이 상수이거나, 단조 증가 또는 단조 감소하는 패턴을 가질 수 있다. 본 논문에서는 결함의 기대값을 가정하는 유한고장소프트웨어 NHPP모형과 수리시점에서도 고장이 발생할 상황을 반영하는 무한고장 NHPP모형들을 비교 제시하였다. 소프트웨어 신뢰성분야에서 많이 사용되는 지수파우어분포에 근거한 유한고장과 무한고장 소프트웨어 신뢰성모형에 대한 비교문제를 제시하였다. 그 결과 유한고장모형이 무한고장모형보다 효율적으로 나타났다. 그리고 모수추정법은 최우추정법을 이용하였다. 이 연구를 통하여 소프트웨어 개발자들은 소프트웨어 고장현상을 파악하는데 어느 정도 도움을 줄 수 있을 것으로 사료된다.

Abstract NHPP software reliability models for failure analysis can have, in the literature, exhibit either constant, monotonic increasing or monotonic decreasing failure occurrence rates per fault. In this paper, finite failure NHPP models that assuming the expected value of the defect and infinite failures NHPP models that repairing software failure point in time reflects the situation, were presented for comparing property. Commonly used in the field of software reliability based on exponential power distribution software reliability model finite failures and infinite failures were presented for comparison problem. As a result, finite fault model is effectively infinite fault models, respectively. The parameters estimation using maximum likelihood estimation was conducted. In this research, software developers to identify software failure property some extent be able to help is considered.

Key Words : Exponential Power Distribution, Finite and Infinite Model, Mission Time, NHPP, exponential distribution

1. 서론

소프트웨어 신뢰성은 일정한 환경조건에서 일정기간동안 고장이 나지 않고 운영 할 수 있는 확률이다.

따라서 이러한 소프트웨어 신뢰성은 시스템 신뢰도에 영향을 주는 중요한 요소가 되고 디

자인 속성 측면에서는 하드웨어 신뢰성과는 다른 면을 가지고 있다.

소프트웨어의 다양한 기능은 소프트웨어 신뢰성문제들에 관한 주요한 요인이 된다. 소프트웨어의 신뢰도의 일반적인 정의는 일정한 기간 동안 주어진 환경 하에서 컴퓨터 프로그램을 고장 없이 사용할 수 있는 확률을 의미한다. 결국 소

*Corresponding Author : Division of Industrial & Management Engineering, Namseoul University, Seoul, Korea(kim1458@nsu.ac.kr)

**Division of Internet information, BaekSeok Culture University, Seoul, Korea

Received June 2, 2015

Revised June 07, 2015

Accepted June 10, 2015

소프트웨어 개발과정에서 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 방출시간의 고려사항을 가진 소프트웨어 개발 과정은 필수 불가결 하다.

지금까지 많은 소프트웨어 신뢰성모형이 제안되었다. 이 중에서 비동질적 포아송과정(Non-homogeneous Poisson process; NHPP)에 의존한 모형[1]은 에러 탐색 과정측면에서는 우수한 모형이고 이러한 모형은 결함이 발생하면 즉시 제거되고 디버깅과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

이 분야에서 Gokhale과Trivedi [2]은 고양된 비동질적인 포아송 과정 모형(Enhanced NHPP)을 제시하였고 Goel 과 Okumoto [3]은 결함의 누적수가 S 형태나 지수적 형태(S-shaped or exponential-shaped)를 가진 평균값함수(Mean value function)를 이용한 지수적 소프트웨어 신뢰성모형(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(Inflection S-shaped reliability growth model)이 제안되었다. Zhao [5]는 소프트웨어 신뢰도에서 변환점 문제를 제시하였고 Shyur [6]는 변환점을 이용한 일반화한 신뢰도 성장모형을 제안하였다. Pham와 Zhang[7]는 테스트 커버리지(Coverage)를 측정하여 소프트웨어 안정도를 평가 할 수 있는 소프트웨어 안정도모형을 제시했다. 비교적 최근에, Huang [8]은 일반화 로지스틱 테스트 노력함수(Generalized logistic testing-effort function)와 변환점 모수(Change-point parameter)를 통합하여 효율적인 소프트웨어 신뢰성 예측기술을 제시하기도 하였다. 그리고 최근에는 S-형태모형은 소프트웨어

관리자들이 소프트웨어 및 검사도구에 익숙해지는 학습과정을 설명할 수 있다고 하였다[9]. 또한 kim[10]은 반로지스틱분포를 이용한 학습과정에 대한 연구도 이루어졌다.

본 논문에서는 지수과우어 분포를 수명분포로 적용하여 결함의 기대값을 가정하는 유한고장소프트웨어 NHPP모형과 수리시점에서 고장이 발생할 상황을 반영하는 무한고장 NHPP모형들을 비교 제시하였다.

2. 관련연구

2.1 NHPP 소프트웨어 신뢰성

$N(t)$ 을 시간 t 까지 검출된 소프트웨어의 누적고장수라고 하고, $m(t)$ 를 이에 대한 기대값을 나타내는 평균값함수(Mean Value Function)로 가정하고 $\lambda(t)$ 을 강도함수(Intensity function)(즉, t 에서의 순간 결함 검출율)이면 비동질 포아송 과정(NHPP)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 NHPP모형에서 평균값 함수 $m(t)$ 와 강도함수 $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[1][10].

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

이처럼 시간관련모형(Time domain models)들은 NHPP에 의해서 확률고장과정으로 설명이 가능하다. 이러한 NHPP모형들은 유한고장모형과 무한고장 범주로 분류한다[1]. 유한고장 NHPP모형에서는 시간 $(0, t]$ 까지 탐색되어 질 수 있는 결함의 기대값을 θ 라고 표현하면 유한고장 NHPP모형의 평균값함수와 강도함수는 다

음과 같이 표현할 수 있다[1][11].

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

반면에 무한고장 NHPP모형들은 수리시점에 서도 고장이 발생할 상황을 반영하기 위하여 기록 멈춤 통계량(Record breaking statistics)을 사용하는 RVS(Record Value Statistics)모형을 사용할 수 있다고 하였고 이 RVS모형과 NHPP 모형에 관해서 평균값함수는 다음과 같이 된다고 하였다[1][11].

$$m(t) = -\ln(1-F(t)) \quad (4)$$

따라서 (2)식과 (4)식을 연관시키고 $f(t)$ 을 확률밀도함수, $F(t)$ 을 분포함수라고 하면 NHPP의 강도함수는 다음과 같이 위험함수($h(t)$)가 된다.

$$\lambda(t) = m'(t) = f(t)/(1-F(t)) = h(t) \quad (5)$$

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

$$x_n = \sum_{i=1}^n t_i \quad (i=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (6)$$

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 시간절단 모형에서 θ 을 모수공간이라고 표시하면 NHPP모형의 우도함수는 다음과 같이 알려져 있다[10].

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp[-m(x_n)] \quad (7)$$

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

2.2 효율적 모형을 위한 모형의 비교

최근 모형에 대한 효율성을 조사하기 위한 기준으로서 평균제곱오차(Mean square error; MSE)와 결정계수(Coefficient of determination; R^2)를 사용한다[12].

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

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n-k} \quad (8)$$

단, $m(x_i)$ 은 시간 $(0, x_i]$ 까지 나타난 고장들의 누적개수를 의미하고 $\hat{m}(x_i)$ 는 x_i 시점까지 평균 값 함수로부터 추정된 고장의 누적개수를 의미한다. 그리고 n 은 관찰 값의 수이고 k 는 모수의 수를 의미한다. 즉, 작은 평균제곱오차 값을 가진 모형은 효율적인 모형이 된다.

결정 계수는 예측 값의 차이에 대한 설명력을 나타내는 도구로서 다음과 같이 정의 된다. 따라서 보다 큰 결정계수를 가진 모형은 효율적인 모형으로 간주된다.

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left(m(x_i) - \sum_{j=1}^n m(x_j)/n \right)^2} \quad (9)$$

3. 제안한 지수파우어분포를 이용한 유한고장과 무한고장 NHPP 소프트웨어 신뢰성 모형

지수파우어분포(Exponential power distribution)는 다양한 형상모수와 척도모수로 이루어진 분포로서 다양한 현상을 나타낼 수 있는 분포로서 척도모수(δ)와 형상모수(κ)에 따른 확률밀도함수의 형태는 [그림 1]에 요약되었다[13].

$$f(t) = (e^{1-e^{\delta t}}) e^{\delta t^{\kappa}} \delta^{\kappa} t^{\kappa-1} \quad (10)$$

$$F(t) = 1 - e^{1-e^{\delta t}} \quad (11)$$

단, $\delta, \kappa > 0, t \in [0, \infty]$

[그림 1]에서 보여주듯이 형상모수 $\kappa=2$ 인 경우에는 거의 대칭을 이루는 형태로서 수명분포로 적합하지 못하지만 형상모수 $\kappa=1$ 인 경우에는 고장시간이 지남에 따라 감소형태(신뢰성장)를 나타내기 때문에 수명분포로 적합하다[14].

따라서 본 연구에서는 형상모수 $\kappa=1$ 인 경우를 적용하고자 한다.

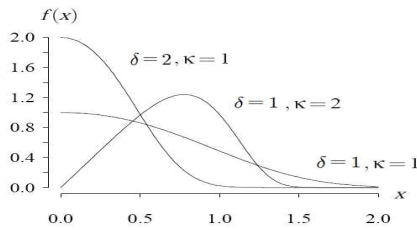


그림 1. 척도모수 δ 와 형상모수 κ 를 고려한 확률밀도함수
Fig. 1. Probability density function considering scale parameter δ and shape parameter κ

따라서 $\kappa=1$ 인 경우 (3)식과 (8), (9)식을 이용하면 유한고장 NHPP 강도함수와 평균값함수는 다음과 같이 표현 할 수 있다.

$$\lambda(t) = \theta F'(t) = \theta (e^{1-e^{\delta t}}) e^{\delta t}, \quad (\theta, \delta > 0) \quad (12)$$

$$m(t) = \theta F(t) = \theta [1 - e^{1-e^{\delta t}}] \quad (13)$$

이경우의 우도함수는 (7)식에 (12)식과 (13)식을 대입하면 다음과 같다.

$$L_{NHPP}(\theta | \underline{x}) = \left[\prod_{i=1}^n \theta (e^{1-e^{\delta x_i}}) e^{\delta x_i} \right] \exp[-\theta (1 - e^{1-e^{\delta x_n}})] \quad (14)$$

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

모수 추정방법은 최우추정법(Maximum likelihood estimation; MLE)을 사용하였고 최우

추정법을 이용하기 위한 로그우도함수는 (14)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln \theta + \sum_{i=1}^n (1 - e^{\delta x_i}) + \delta \sum_{i=1}^n x_i - \theta (1 - e^{1-e^{\delta x_n}}) \quad (15)$$

(15)식에서 θ 와 δ 에 대하여 편미분 하여 다음과 같은 식을 만족하는 최우추정값 $\hat{\theta}_{MLE}$ 와 $\hat{\delta}_{MLE}$ 을 수치 해석적 방법으로 계산할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - (1 - e^{1-e^{\delta x_n}}) = 0 \quad (16)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \delta} = - \sum_{i=1}^n x_i e^{\delta x_i} + \sum_{i=1}^n x_i - \theta x_n e^{\delta x_n} e^{1-e^{\delta x_n}} = 0 \quad (17)$$

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

또한, 임무시간에 따른 신뢰도는 다음과 같이 유도 된다[12].

$$\hat{R}(\tau | x_n) = \exp[-m(\tau + x_n) + m(x_n)] \quad (18)$$

$$\text{단, } m(\tau + x_n) = \theta [1 - e^{1-e^{\delta(\tau+x_n)}}], \\ m(x_n) = \theta [1 - e^{1-e^{\delta x_n}}]$$

이고 τ 는 임무시간(Mission time)을 의미한다.

반면에 무한고장 NHPP 강도함수와 평균값함수는 (4)식과 (5)식을 이용하면 다음과 같이 표현 할 수 있다.

$$\lambda(t) = \delta e^{\delta t}, \quad m(t) = e^{\delta t} - 1 \quad (19)$$

이경우의 우도함수는 (7)식에 (19)식을 대입하면 다음과 같다.

$$L_{NHPP}(\theta | \underline{x}) = \left[\prod_{i=1}^n \delta e^{\delta x_i} \right] \exp[-(e^{\delta x_n} - 1)] \quad (20)$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$
 모수 추정방법은 최우추정법을 사용하였고 최우추정법을 이용하기 위한 로그우도함수는 (20) 식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln \delta + \delta \sum_{i=1}^n x_i - (e^{\delta x_n} - 1) \quad (21)$$

(22)식에서 δ 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\delta}_{MLE}$ 을 수치 해석적 방법으로 계산할 수 있다[9].

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \delta} = \frac{n}{\delta} + \sum_{i=1}^n x_i - x_n e^{\delta x_n} = 0 \quad (22)$$

단, $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$
 또한, 임무시간에 따른 신뢰도는 다음과 같이 유도 된다[12].

$$\hat{R}(\tau | x_n) = \exp \left[- (e^{\delta(\tau+x_n)} - 1) + (e^{\delta x_n} - 1) \right] \quad (23)$$

4. 소프트웨어 고장시간 자료 분석

이 절에서는 소프트웨어 고장 시간자료[15] (Failure time data)를 이용하여 본 논문에서 제시하는 소프트웨어 신뢰모형들을 이용한 소프트웨어 고장시간을 분석하고자 한다. 이 자료의 고장시간은 18.735 시간당위에 30번의 고장이 발생된 자료이며 [표 1]에 나열 되어 있다.

표 1. 소프트웨어의 고장시간자료

Table 1. Failure time data of software

Failure number	Failure time (hours)	Failure number	Failure time (hours)
1	0.479	16	10.771
2	0.745	17	10.906
3	1.022	18	11.183
4	1.576	19	11.779
5	2.61	20	12.536
6	3.559	21	12.973
7	4.252	22	15.203
8	4.849	23	15.64
9	4.966	24	15.98
10	5.136	25	16.385
11	5.253	26	16.96
12	6.527	27	17.237
13	6.996	28	17.6
14	8.17	29	18.122
15	8.863	30	18.735

Laplace trend test

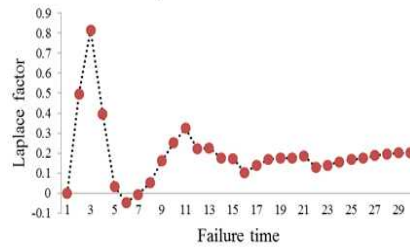


그림 2. 라플라스 추세검정
 Fig. 2. Laplace trend test

또 한 제시하는 신뢰모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[16][17]. 추세분석에는 일반적으로 라플라스 추세검정(Laplace trend test)을 사용한다. 이 검정을 실시한 결과 [Fig.2]에서 라플라스 추세 검정의 결과는 라플라스요인(Factor)이 -2와 2사이에 존재함으로써 즉, 극단값(Extreme value)이 존재하지 않으므로 이 자료를 이용하여 신뢰성장모형을 제시하는 것이 효율적임을 시사하고 있다[17].

모수추정은 최우추정법을 이용하고 모수추정을 용이하게 하기 위하여 원래의 고장시간 데이터를 변수변환(Failure time $\times 10^{-1}$)하여 적용하였

다. 비선형 방정식의 계산방법은 수치 해석적 기본방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기 값을 0.01과 3을, 허용 한계(Tolerance for width of interval)는 10^{-5} 을 주고 수렴성을 확인하면서 충분한 반복횟수인 100번을 C-언어를 이용하여 모수추정을 수행하였다. 최우추정법의 결과와 모형에 대한 효율성을 조사하기 위한 기준으로서 MSE (평균제곱오차)와 R^2 (결정계수)가 [표 2]에 요약되었다. 이 표에서 유한고장 지수파우어 모형이 무한고장 지수파우어 모형보다 실제 값과 예측 값에 대한 차이를 측정하는 평균제곱오차가 낮고 예측 값의 차이에 대한 설명력을 의미하는 설명력도 높게 나타나기 때문에 유한고장 지수파우어 모형이 무한고장 지수파우어 모형보다 효율적인 모형으로 간주할 수 있다.

표 2. 각 모형에 대한 모수추정값 및 MSE , R^2
 Table 2. Parameter estimation of each model MSE , R^2

Model	MLE	Model Comparison	
		MSE	R^2
Finite Exponential	$\hat{\theta}_{MLE} = 54.9038$	1.6151	0.989
	$\hat{\delta}_{MLE} = 0.3109$		
Infinite Exponential	$\hat{\delta}_{MLE} = 1.7112$	82.9704	0.889

Note. MLE : Maximum likelihood estimation;
 MSE : Mean square error;
 R^2 : Coefficient of determination.

[그림 3]에서는 평균값함수의 패턴을 요약 하였다. 이 그림에서 유한고장 지수파우어 모형이 무한고장 지수파우어 모형이 보다 참값에 더 가깝게 추정되었다. 그리고 [그림 4]에서는 강도함수의 패턴을 요약하였다. 이 그림에서도 유한고장 지수파우어 모형의 강도함수의 패턴은 비감소 속성을 보여주고 있다. 반면에 무한고장 레일리모형은 증가 속성을 나타내고 있다.

[그림 5]에서 보여 주듯이 임무시간에 대한 신뢰도 그림에서도 유한고장 지수파우어 모형과 무한고장 지수파우어 모형을 비교 했을 때

임무시간이 증가 할수록 무한고장 지수파우어 모형보다는 유한고장 지수파우어 모형이 각 시점에서 신뢰도의 상승으로 나타나고 있다.

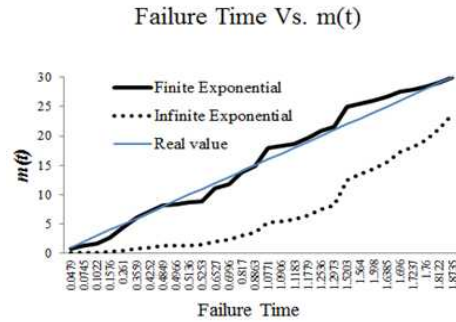


그림 3. 평균값함수
 Fig. 3. Mean value function

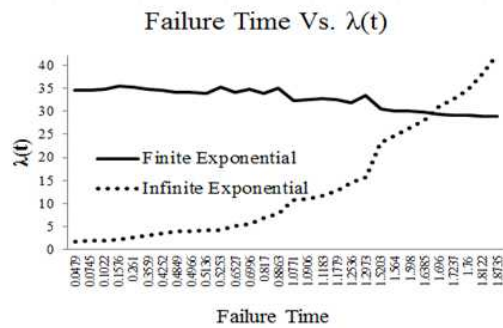


그림 4. 강도함수
 Fig. 4. Intensity function

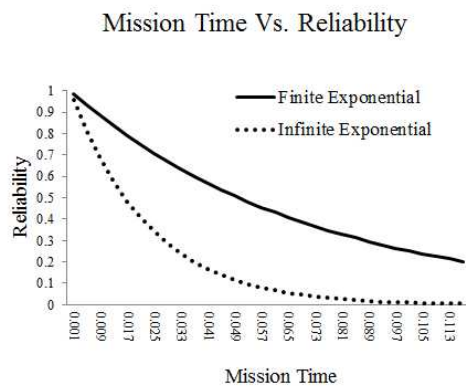


그림 5. 신뢰도
 Fig. 5. Reliability

5. 결론

대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 수 없는 상황이 현실이다.

소프트웨어 신뢰성은 개발의 최종단계에 있는 테스트 공정이나 실제 사용단계에 있어서 소프트웨어 내에 존재하는 고장 수나 고장 발생시간에 의해서 효과적으로 평가할 수 있는 상황으로 그 평가기술이 중요하게 된다. 따라서 소프트웨어개발의 테스트공정이나 실제사용단계에 있어서 고장 발생환경이나 고장 발생현상을 수리적으로 모형화가 가능하면 평가를 할 수 있다. 테스트시간이나 혹은 실행시간, 발생된 고장 수와 고장 발생시간과의 관계를 효율적으로 관리함으로써 소프트웨어 신뢰도를 성장 시킬 수 있다. 이러한 과정을 소프트웨어 성장과정이라고 볼 수 있다.

본 연구에서는 결함의 기대값을 가정하는 유한고장소프트웨어 NHPP모형과 수리시점에서도 고장이 발생할 상황을 반영하는 무한고장 NHPP모형들을 비교 제시하였다.

따라서 본 연구에 제안된 지수파우어 분포에 근거한 유한고장소프트웨어 NHPP모형과 무한고장 NHPP모형들을 비교 제시하였다. 그 결과 유한고장모형이 무한고장모형이 효율적으로 나타났다. 이 연구를 통하여 소프트웨어 개발자들은 다양한 소프트웨어 고장형태를 고려함으로써 소프트웨어 고장형태에 대한 사전지식을 파악하는데 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 Half-Logistic Distribution Software Reliability Model using the Perspective of Learning Effects", *Journal of Next Generation Information Technology*, Vol. 4, No. 8, pp. 132-139, 2013.

- [11] 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, Vol. 5, No.9, pp. 1210 -1218, 2013.
- [12] 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.
- [13] <http://www.math.wm.edu/~leemis/chart/UDR/UDR.html>.
- [14] Lawless, J. F, "Statistical Models and Methods for Lifetime Data", John Wiley & Sons, New York, 1981.
- [15] Y. HAYAKAWA and G. TELFAR, "Mixed Poisson-Type Processes with Application in Software Reliability", Mathematical and Computer Modelling, 31, pp. 151-156, 2000.
- [16] 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.
- [17] 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.

저자약력

김희철 (Hee-Cheul Kim)

[정회원]



- 1992년 2월 : 동국대학교 통계학과 (이학석사)
- 1998년 8월 : 동국대학교 통계학과 (이학박사)
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수

<관심분야>

소프트웨어신뢰성 공학, 전산 통계, 웹프로그래밍

신현철 (Shin-Hyun Cheul)

[정회원]



- 2002년 2월 : 원광대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 : 한국정보처리학회 이사
- 1994년 3월 ~ 현재 : 백석문화대학교 인터넷정보학부 교수

<관심분야>

통신공학, 컴퓨터과학, 소프트웨어공학