한국정보전자통신기술학회논문지(jkiiect)'18-08, Vol.11 No.4
논문 18-11-04-460 http://dx.doi.org/10.17661/jkiiect.2018.11.4.460

Type-2 Gumbel과 Erlang 분포의 형상모수를 따르는 수명분포에 근거한 소프트웨어 개발 비용모형에 관한 특성 연구

양태진*

A characteristic study on the software development cost model based on the lifetime distribution following the shape parameter of Type-2 Gumbel and Erlang distribution

Tae-Jin Yang*

요 약 정보기술의 발달로 컴퓨터 소프트웨어 시스템의 규모는 끊임없이 확장되고 있다. 소프트웨어 개발에 대한 신뢰성 및 비용은 소프트웨어 품질에 큰 영향을 미치고 있다. 본 연구에서는 소프트웨어 고장 간격시간 자료를 바탕으로 NHPP 모형에서 Type-2 Gumbel과 Erlang 분포의 형상모수를 따르는 수명분포에 근거한 소프트웨어 개발 비용모형에 관한 특성을 비교하고, 분석하였다. 그 결과, Go-Okumoto 모형 및 제시한 모형인 Erlang 모형과 Type-2 Gumble 모형에 대한 비용곡선의 추세는 모두 초기단계에서 감소하다가, 고장시간이 지나는 후반부에 가서는 점차 증가하는 결과를 보였다. 또한, Erlang 모형과 Type-2 Gumble 모형을 비교한 결과, Erlang 모형이 소프트웨어 출시시기가 빠르고, 출시시점의 비용도 경제적임을 알 수 있었다. 본 연구를 통하여, 소프트웨어 운용자들은 소프트웨어 출시시기 이후에 결함이 감소되도록 운영단계보다 테스팅 단계에서 결함들을 제거해야 하며, 소프트웨어 개발비용에 관한 특성을 파악하는데 필요한 사전정보을 연구할 수 있을 것으로 기대된다.

Abstract With the development of information technology, the scale of computer software system is constantly expanding. Reliability and cost of software development have a great impact on software quality. In this study, based on the software failure interval time data, a comparative analysis was performed on the characteristics of the software development cost model based on the lifetime distribution following the Type-2 Gumbel and Erlang distribution in the NHPP model. As a result, the trends of the cost curves for the Go-Okumoto model and the proposed Erlang model and the Type-2 Gumble model both decreased in the initial stage and gradually increased in the latter half of the failure time. Also, Comparing the Erlang model with the Type-2 Gumble model, we found that the Erlang model is faster and more cost-effective at launch. Through this study, Software operators should remove possible defects from the testing phase rather than the operational phase to reduce defects after the software release date, it is expected to be able to study the prior information needed to understand the characteristic of software development cost.

Key Words : Erlang distribution, Laplace trend test, NHPP, Software cost model, Type-2 Gumbel

1. 서론

정보 기술의 발달로 컴퓨터 소프트웨어 시스템의 규모는 끊임없이 확장하고 성장하고 있다. 소프트웨어 개발에 대한 신뢰성 및 비용이 소프트웨어 품질에 큰 영향을 미치고 있다. 또한, 소프트웨어 개발 비용은 소프트웨어의 신뢰성과 관련된 소프트웨어 유지 보수 비용에도 크게 영

Funding for this paper was provided by Namseoul University year 2018
*Corresponding Author : Department of Electronic Engineering, Namseoul University(solomon645@nsu.ac.kr)
Received June 29, 2018 Revised July 09, 2018 Accepted August 26, 2018

향을 미칠 수 있다[1].

특정 환경에서 지정된 시간 동안 고장없이 소프트웨어가 정상적으로 작동할 확률로 정의되는 소프트웨어 신뢰성은 가장 중요한 품질 표준이 된다[2].

지난 수십 년 동안, 소프트웨어의 신뢰성을 예측하기 위해 소프트웨어 고장 프로세스에 대해 NHPP에 기반한 많은 소프트웨어 신뢰성 성장 모형들이 제안되었다. 즉, 잔존 고장의 수, 고장율 등의 신뢰성 특성을 추정하기 위해 제어된 시험환경 내에서 결함 강도함수 및 평균값 함수를 이용하여 비동질 포아송 과정(Non-homogeneous Poisson process ; NHPP)에 근거한 소프트웨어 신뢰도 모형이 개발되고, 제안되었다[1][6].

Yamada와 Osaki [2]는 최우추정법을 이용하여 평균값 함수의 결과를 추정 할 수 있다고 하였고, Huang [3]은 평균값 함수의 신뢰구간을 나타내는 그래프를 제시하여 설명하였다. Chatterjee와 Singh [4]는 불완전한 디버깅을 통한 물류 지수 테스트 커버리지 함수를 통합하여 NHPP에 기반한 소프트웨어 신뢰성 모델을 제안하였다.

본 연구에서는 소프트웨어 고장시간의 간격 자료를 바탕으로 NHPP 모형에서 Type-2 Gumbel과 Erlang 분포의 형상모수를 따르는 수명분포에 근거하여 소프트웨어 개발 비용모형에 관한 특성을 비교한 후, 결과에 대한 평가분석을 하고자 한다.

2. 배경연구

2.1 기본모형인 Goel-Okumoto 모형

Goel-Okumoto 모형은 이 분야에서 잘 알려진 기본적인 모형이다[9]. Goel-Okumoto 모형[5]에 대한 $f(t)$와 $F(t)$를 각각 확률밀도함수와 누적밀도함수라고 하면 유한고장 강도함수와 평균값 함수는 다음과 같다[12][13].

$$\lambda(t|\theta, b) = \theta f(t) = \theta b e^{-bt}, \quad (\theta > 0,\ b > 0) > 0) \tag{1}$$

$$m(t|\theta, b) = \theta F(t) = \theta (1 - e^{-bt}) \tag{2}$$

n번째 까지 고장시점 자료와 (1)식과 (2)식을 고려하면 유한고장 NHPP모형의 우도함수는 다음과 같다[1].

$$L(\theta, b|\underline{x}) = \left(\prod_{i=1}^{n} \theta b e^{-b x_i}\right) \exp\left[-\theta(1 - e^{-b x_n})\right] \tag{3}$$

단, $\underline{x} = (0 \le x_1 \le x_2 \le \cdots \le x_n]$.

(3)식을 이용한 로그우도함수는 다음과 같은 로그 조건식으로 간략화된다.

$$\ln L_{NHPP}(\Theta|\underline{x}) = n\ln\theta + n\ln b - b\sum_{k=1}^{n} x_k - \theta\left(1 - e^{-b x_n}\right) \tag{4}$$

따라서, 각 모수에 대한 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{b}_{MLE}$ 은 다음과 같은 조건식을 만족한다[5].

$$\frac{n}{\hat{\theta}} = 1 - \exp\left(-\hat{b}\, x_n\right), \tag{5}$$

$$\frac{n}{\hat{b}} = \sum_{I=1}^{n} x_n + \hat{\theta} x_n \exp\left(-\hat{b}\, x_n\right) \tag{6}$$

2.2 Erlang 분포 NHPP 모형

어랑분포(Erlang distribution)는 여러 가지 확률적 현상을 해석 할 수 있는 분포이다. 형상모수(a)와 척도모수(b)에 고려한 확률밀도함수와 누적분포는 다음과 같다[5][12].

$$f(t) = \frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt} \tag{7}$$

$$F(t) = \left(1 - e^{-bt} \sum_{i=0}^{a-1} \frac{(bt)^i}{i!}\right) \tag{8}$$

단, $a, b > 0,\ a = 1, 2, 3\cdots,\ t \in [0, \infty]$

본 논문에서는 형상모수$a = 2$인 경우를 고려하고자 한다. 따라서 이 경우 (7)식과 (8)식을 이용하면 유한고장 NHPP모형의 로그우도함수는 다음과 같이 유도 할 수 있다[10][13].

$$\ln L_{NHPP}(\Theta|\underline{x}) = \tag{9}$$
$$n\ln\theta - n\ln\Gamma(a) + na\ln b + (a-1)\sum_{i=1}^{n} \ln x_i$$
$$-b\sum_{I=1}^{n} x_i - \theta + \theta e^{-b x_n}\left(\sum_{i=0}^{a-1} \frac{(b\, x_n)^i}{i!}\right)$$

다음과 같은 조건식을 만족시키는 최우추정량 $\hat{\theta}_{MLE}$와 $\hat{b}_{MLE}$을 수치 해석적 방법으로 추정 할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\Theta \mid \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-b x_n}\left(\sum_{i=0}^{a-1}\frac{(b x_n)^i}{i!}\right) = 0 \quad (10)$$

$$\frac{\partial \ln L_{NHPP}(\Theta \mid \underline{x})}{\partial b} = \frac{a n}{b} - \sum_{i=1}^{n} x_i + \frac{\partial\left[\theta e^{-b x_n}\left(\sum_{i=0}^{a-1}\frac{(b x_n)^i}{i!}\right)\right]}{\partial b} = 0 \quad (11)$$

2.3 Type-2 Gumbel 분포 NHPP 모형

Type-2 Gumbel 분포 모형의 NHPP의 강도함수와 평균값 함수는 다음과 같이 표현이 가능하다[7].

$$\lambda(t|a,b) = \theta f(t|a,b) = \theta\left(a b t^{-a-1} e^{-b t^{-a}}\right) \quad (12)$$

$$m(t|a,b) = \theta e^{-b t^{-a}} \quad (13)$$

단, $a, b > 0,\ t \in [0, \infty]$

따라서 (12)식과 (13)식을 이용하면 로그우도함수는 다음과 같이 유도된다[10].

$$ln L_{NHPP}(\Theta \mid \underline{x}) = n ln\theta + n ln a + n ln b - (a+1)\sum_{i=1}^{n} ln x_i - b\sum_{i=1}^{n} x_i^{-a} - \theta e^{-b x_n^{-a}} \quad (14)$$

본 연구에서는 (14)식에서 형상모수가 $a=2$ 를 고려하고자한다. 따라서 다음과 같은 식을 만족하는 $\hat{\theta}_{MLE}$ 와 $\hat{b}_{MLE}$ 추정값을 수치 해석적 방법으로 추정 할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\Theta \mid \underline{x})}{\partial \theta} = \frac{n}{\theta} - e^{-b x_n^{-2}} = 0 \quad (15)$$

즉, $\hat{\theta} = \frac{n}{e^{-b x_n^{-2}}}$

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

2.4 소프트웨어 비용 모형

소프트웨어 비용 모형의 일반적인 형태는 다음과 같이 구성된다[8][12].

$$E = E_1 + E_2 + E_3 + E_4 = E_1 + C_2 \times t + C_3 \times m(t) + C_4 \times [m(t+t') - m(t)] \quad (17)$$

단, E: 소프트웨어 개발 예상 총비용

E_1은 데이터 분석, 인력, 중앙 처리 장치(CPU) 처리시간 등과 같은 소프트웨어 설계 및 초기 소프트웨어 개발 비용을 의미하며 상수로 간주된다. 또한 E_2는 단위 시간당 일정한 소프트웨어 테스팅 비용을 의미하며 다음과 같이 구체화 된다.

$$E_2 = C_2 \times t \quad (18)$$

단, C_2는 단위 시간당 비용이고 t는 테스팅 시점.

그리고 E_3는 기본 결함을 감지하고 하나의 결함을 제거하는 비용을 의미하며 다음과 같이 구체화 된다.

$$E_3 = C_3 \times m(t) \quad (19)$$

단, C_3는 테스팅 과정에서 발견된 하나의 결함을 제거하는 비용, $m(t)$는 t시점에서 탐색되어 질 수 있는 결함의 기대수를 나타낸다. 또한 E_4는 운영 소프트웨어 시스템에서 남아있는 모든 결함들을 제거하는 비용(상수)으로 다음과 같이 구체화 된다.

$$E_4 = C_4 \times [m(t+t') - m(t)] \quad (20)$$

C_4는 소프트웨어 출시 된 이후에 소프트웨어 운영 단계에서 운용자가 발견된 결함수정 비용, t' 는 소프트웨어 시스템을 출시 한 이후 운영 및 소프트웨어를 유지할 수 있는 시간을 의미한다. 현실적으로는 C_4 는 C_2 와 C_3 보다 높은 비용을 나타낸다. 그러므로 최적의 소프트웨어 출시(방출) 시간 (t)는 다음과 같이 유도 할 수 있다[8][12].

$$\frac{\partial E}{\partial t} = E' = (E_1 + E_2 + E_{3+} E_4)' = 0 \quad (21)$$

3. 소프트웨어 고장시간을 이용한 비용 특성분석

표 1. 소프트웨어 고장시간 자료
Table 1. Software failure time data

Failure number	Failure time (hours)	Failure numbe	Failure time(hours)*
1	30.02	16	151.78
2	31.46	17	177.50
3	53.93	18	180.29
4	55.290	19	182.21
5	58.720	20	186.34
6	71.920	21	256.81
7	77.070	22	273.88
8	80.900	23	277.87
9	101.90	24	453.93
10	114.87	25	535.00
11	115.34	26	537.27
12	121.57	27	552.9
13	124.97	28	673.68
14	134.07	29	704.49
15	136.25	30	738.68

이 장에서는 소프트웨어 고장시간 정보자료[9](Failure time information data)를 적용하여 신뢰성 모형들의 특징을 상호 비교, 분석하고자 한다.

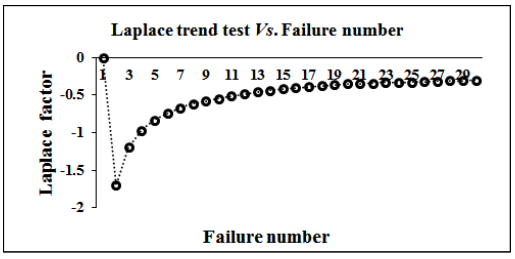

그림 1. 라플라스 검정의 추정결과
Fig. 1. Estimation results of Laplace test

자료에 대한 신뢰성을 확보하기 위하여 추세검정은 라플라스 추세검정(Laplace trend test)을 사용하였다. [그림 1]에서 라플라스 추세검정의 결과는 라플라스요인(Factor)의 추정값이 -2와 2사이에 분포(극단값(Extreme value)이 존재하지 않으므로)함으로서 이 자료를 적용하여 신뢰성 비용 모형을 제시하는 것이 유용하다고 할 수 있다[11]. 모수추정은 최우추정법을 이용하고 모수추정을 용이하게 하기 위하여 원래의 고장시간 데이터를 수치변환($Failure\ time \times 10^{-2}$)하여 적용하였다. 비선형 방정식의 추정은 이분법(Bisection method)을 사용하였다. 이러한 추정은 초기값을 0.001과 5.0으로 사용하고, 허용한계(Tolerance for width of interval) 값은 10^{-5} 으로 지정한 후, 수렴성을 위하여 충분한 반복 횟수인 50번을 적용하여 최우추정법(Maximum Likelihood Estimation ; MLE)으로 모수 추정을 실시하였다. 그 결과는 [표 2]에 나열되었다.

표 2. 각 모형에 대한 모수추정
Table 2. Parameter estimation of each model

Model	MLE	
Goel - Okumoto	$\hat{\theta}=33.4092$	$\hat{b}=3.09\times10^{-1}$
Type - 2 Gumbel	$\hat{\theta}=30.3852$	$\hat{b}=6.96\times10^{-1}$
Erlang	$\hat{\theta}=30.5978$	$\hat{b}=7.92\times10^{-1}$

본 연구에서는 (17)식을 이용하여 효율적인 값을 가정하여 [그림 2]와 [그림 3]과 같이 비용 곡선의 특징을 비교 및 분석하고자 한다.

[가정1] (22)

$E_1=5\$,\ c_2=5\$,\ c_3=1.5\$,\ c_4=5\$,\ t'=10$

(22)식의 가정을 적용한 비용곡선의 결과는 [그림 2]에 요약되었다. 이 그림에서의 패턴은 초기단계의 추세는 감소하고 일정한 추세를 보이다가 고장시간이 지나는 후반부에 가서는 점차 증가하는 추세를 보이고 있다. 그 이유는 소프트웨어 시스템의 남아있는 결함의 수는 결함을 제거하는 과정에서 점점 줄어들기 때문에 되고 남아 있는 잔존 결함이 관측될 확률은 점점 낮아지게 되기 때문이다. 따라서, 후반부의 단계에서 소프트웨어에 남아있는 잔존 결함의 수는 적기 때문에 이 테스트 단계에서 오류를 검출할 시간이 상대적으로 길고 오류를 제거하는 비용은 상대적으로 높기 때문에 비용곡선의 추세는 시간이 흐름에 따라 지속적으로 증가하게 된다. 그러므로, 이러한 사전정보를 이용한 비용곡선의 추세를 이용하여 최적의 소프트웨어 출시(방출) 시간을 예측 할 수 있다. 이러한 작업

은 가장 현실적인 내용이며 대부분의 경우, 실제 소프트웨어 개발 과정에서도 이러한 추세를 가지게 된다고 알려져 있다[8].

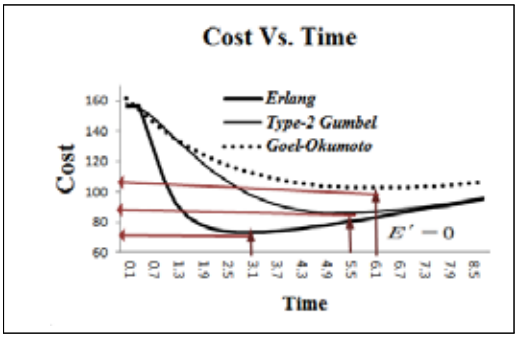

그림 2. [가정1]을 적용한 비용곡선
Fig. 2. The cost curve applied to the condition of [*Supposition* 1]

[그림 2]에서 기본 모형인 Go-Okumoto 모형에 대한 비용곡선의 추세도 초기단계의 추세는 감소한 후, 일정한 추세를 보이다가 고장시간이 지나는 후반부에 가서는 점차 증가하는 추세를 보이고 있다. Erlang 모형과 Type-2 Gumble 모형도 유사한 비용 패턴을 보이고 있으나, 고장시간의 전체 구간에서 기본 모형인 Go-Okumoto 모형보다 비용이 감소하고 있어 상대적으로 더욱 효율적인 모형으로 간주할 수 있다. 결론적으로, Erlang 모형과 Type-2 Gumble 모형을 비교한 결과, Erlang 모형이 소프트웨어 출시(방출) 비용이 가장 적고, 출시 시점도 제일 빨라서 효율적인 모형임을 알 수 있다.

[가정2] (23)

$E_1 = 5\$,\ c_2 = 5\$,\ c_3 = 1.5\$,\ c_4 = 15\$,\ t' = 10$

[가정1]과 비교하여, 다른 가정은 모두 동일하지만, 소프트웨어 출시 이후에 소프트웨어 운영 단계에서 운용자가 발견한 결함수정 비용인 C_4를 15$로 증가시킨 경우로서, 시뮬레이션 결과는 [그림 3]에 요약하였다.

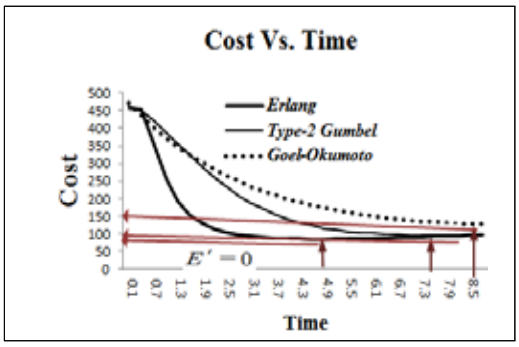

그림 3. [가정2]을 적용한 비용곡선
Fig. 3. The cost curve applied to the condition of [*Supposition* 2]

[그림 3]은 소프트웨어 운영 단계에서 운용자가 발견한 결함수정 비용 C_4을 5$에서 10$로 상승시킨 경우를 나타낸 비용곡선이다. 이 경우도 Erlang 모형과 Type-2 Gumble 모형도 유사한 패턴을 보이고 있지만, 고장시간의 전제 구간에서 기본 모형인 Go-Okumoto 모형보다 비용이 감소하고 있어 상대적으로 효율적 모형으로 간주 된다. Erlang 모형과 Type-2 Gumble 모형을 비교한 결과, Erlang 모형이 소프트웨어 출시 시점의 비용이 가장 적고 출시시기도 제일 빨라서 가장 경제적인 모형임을 알 수 있다. 또한, [가정1]과 비교 했을 때 모든 모형들이 고장시점과 출시 시점에서의 비용이 모두 높게 나타남을 알 수 있다.

따라서, 이 경우에는 소프트웨어 출시시기 이후에 결함을 감소시킬 수 있도록 운영단계보다 테스팅 단계에서 가능한 모든 결함들을 제거해야 한다.

4. 결론

소프트웨어 개발과정에서 테스트 작업이나 실제소프트웨어 사용과정에서 고장발생 속성 혹은, 고장 발생 현상을 정량적으로 모형화한 후, 소프트웨어 유용성을 비교분석하면 상대적으로 효율성 평가를 할 수 있다. 본 연구에서는 소프트웨어 고장 시간 자료를 바탕으로 NHPP 모형에서 Type-2 Gumbel과 Erlang 분포의 형상모수를 따르는 수명분포에 근거한 소프트웨어 개발 비용모형에 관한 특성을 비교한 후, 평가 분석을 하였다.

본 연구의 결과를 다음과 같이 요약 할 수 있다.

첫째, Go-Okumoto 모형 및 제시된 모형인 Erlang 모형과 Type-2 Gumble 모형에 대한 비용곡선의 추세도 초기 단계의 추세는 감소하고 일정한 추세를 보이다가 고장시간이 지나는 후반부에 가서는 점차 증가하는 추세를 보이고 있다. 그 이유는 소프트웨어 시스템의 남아있는 결함의 수는 결함의 제거하는 과정에서 점점 줄어들기 때문에 남아 있는 잔존 결함이 관측될 확률은 점점 낮아지게 되기 때문임을 알 수 있었다.

둘째, Erlang 모형과 Type-2 Gumble 모형과 비교한 결과 Erlang 모형이 출시시점에서의 비용이 가장 적고, 출시 시기도 제일 빨라서, 가장 경제적인 비용 모형임을 알 수 있었다.

셋째, 실제 오류를 탐지 및 제거 단계에서 오류 수정 비용은 운용 단계에서 남아있는 모든 오류를 제거하는 비용보다 낮지만 운영 시간이 증가함에 따라 총비용이 증가함을 알 수 있다. 따라서 이 경우에는 소프트웨어 출시시기 이후에 결함을 감소시킬 수 있도록 운영단계보다 테스팅 초기 단계에서 가능한 결함들을 제거해야 한다. 결론적으로 본 연구결과를 이용하면 소프트웨어 개발자 및 사용자들은 최적의 소프트웨어 출시(방출) 시간과 가장 효율적인 소프트웨어 개발비용을 예측하는데 필요한 사전 정보로 활용할 수 있다고 판단된다.

REFERENCES

[1] K. Y. Song, I. H. Chang, H. Pham, "A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments", Applied Science, Vol.7, No.983, pp. 1-16, 2017.

[2] S. Yamada, S. Osaki, "Software reliability growth modeling : models and applications", IEEE Transactions on Software Engineering, Vol.11, No.12, pp.1431-1437, 1985.

[3] C. Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point", Journal of Systems and Software, Vol.76, No.2, pp.181-194, 2005.

[4] S. Chatterjee, J. B. Singh, "A NHPP based software reliability model and optimal release policy with logistic-exponential test coverage under imperfect debugging", International Journal of System Assurance Engineering and Management, Vol.5, Issue 3, pp.399-406, 2013.

[5] Goel A L, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", IEEE Transactions on Software Engineering, Vol.28, pp.206-211, 1978.

[6] H. C. KIM, "The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Rayleigh and Burr Type", Journal of korea society of digital industry and information management, Vol.10, No.2, pp.1-11, 2014.

[7] https://en.wikipedia.org/wiki/Type-2_Gumbel_distribution

[8] Ye Zhang, and Kaigui Wu, "Software Cost Model Considering Reliability and Time of Software in Use", Journal of Convergence Information Technology, Vol.7, No.13, pp.135-142, 2012.

[9] R. Satya Prasad, K. R. H. Rao and R.R.L Kantha, "Software Reliability Measuring using Modified Maximum Likelihood Estimat ion and SPC", International Journal of Computer Applications (0975-8887) Vol.21, No.7, pp.1-5, 2011.

[10] H. C. Kim, "A Comparative Study on Software Reliability Models with Shape Parameter of Type-2 Gumble Life Distribution", International Journal of Soft Computing" Vol.12, No.5-6, pp. 351-354, 2017.

[11] T. J. Yang, "The Comparative Study of NHPP Software Reliability Model Based on Log and Exponential Power Intensity Function", The Journal of Korea Institute of Information, Electronics, and Commmunication Technology, Vol.8, No.6, pp.445-452, 2015.

[12] T. J. Yang, "A Comparative Software Development Cost Model Considering the Change in the Shape Parameter of the Eralng Distribution", The Journal of Korea Institute of Information, Electronics, and Communication Technology" Vol.9, No.6, pp.566-572, 2016.

[13] T. J. Yang, "The Performance Analysis Comparative Study depend on Software Reliability Model and Curve Regression Model", Medwell Journals, Vol.12, No.5, pp313-317, 2017.

저자약력

양 태 진 (Tae-Jin Yang) **[정회원]**

- 1992년 2월 : 한양대학교 전자공학과 공학석사
- 1995년 2월 : 한양대학교 전자공학과 공학박사(수료)
- 1986년 3월 ~ 1992년 2월 : 현대 그룹 신규사업부 기술과장
- 1993년 3월 ~ 2013년 12월 : 호서대학교 호서전문학교 정보통신과 교수
- 2014년 3월 ~ 현재 : 남서울대학교 전자공학과 교수

<관심분야> 소프트웨어 신뢰성공학, 인공지능 (Artificial - Intelligent), Intelligent-Network & Network-Security