한국정보전자통신기술학회논문지(jkiiect)'19-12 Vol.12 No.6
논문 19-12-06-554 http://dx.doi.org/10.17661/jkiiect.2019.12.6.554

레일리 분포와 역-레일리 분포에 근거한 NHPP 소프트웨어 신뢰성 모형의 개발비용 속성 분석에 관한 연구

양태진*

A Study on Development Cost Attributes Analysis of NHPP Software Reliability Model Based on Rayleigh Distribution and Inverse Rayleigh Distribution

Tae-Jin Yang*

요 약 본 연구에서는 소프트웨어 신뢰성 분야에서 많이 사용하는 유한고장 NHPP Rayleigh 분포 모형과 NHPP Inverse Rayleigh 분포 모형을 소프트웨어 개발비용 모형에 적용한 후, 개발비용과 최적의 방출시간에 대한 속성을 비교, 분석하였다. 소프트웨어 개발비용의 속성을 분석하기 위하여 소프트웨어 고장시간 자료를 사용하였고, 모수추정은 최우추정법을 적용하였으며, 비선형 방정식은 이분법을 사용하여 계산하였다. 그 결과, Rayleigh 모형이 Inverse Rayleigh 모형보다 소프트웨어 개발비용이 비교적 적고, 소프트웨어 방출시점도 빨라서 상대적으로 우수한 모형임을 확인할 수 있었다. 본 연구를 통하여 기존 연구사례가 없는 Rayleigh 모형과 Inverse Rayleigh 모형의 개발비용 속성을 새롭게 분석하였으며, 더불어 소프트웨어 개발자들이 소프트웨어 신뢰도 향상 방법 및 개발비용의 속성을 탐색하는 데 필요한 기본지침으로 활용할 수 있을 것으로 기대한다.

Abstract In this study, after applying the finite failure NHPP Rayleigh distribution model and NHPP Inverse Rayleigh distribution model which are widely used in the field of software reliability to the software development cost model, the attributes of development cost and optimal release time were compared and analyzed. To analyze the attributes of software development cost, software failure time data was used, parametric estimation was applied to the maximum likelihood estimation method, and nonlinear equations were calculated using the bisection method. As a result, it was confirmed that Rayleigh model is relatively superior to Inverse Rayleigh model because software development cost is relatively low and software release time is also fast. Through this study, the development cost attributes of the Rayleigh model and the Inverse Rayleigh model without the existing research examples were newly analyzed. In addition, we expect that software developers will be able to use this study as a basic guideline for exploring software reliability improvement method and development cost attributes.

Key Words : Finite failure NHPP, Inverse Rayleigh distribution, Rayleigh distribution, Software development cost, Software release time, Software reliability model.

1. 서론

4차산업 혁명시대의 핵심인 소프트웨어 기술이 여러 산업분야로 급격히 확장되어, 대용량의 정보를 고장 없이 빠르고, 정확하게 처리할 수 있는 소프트웨어 개

Funding for this paper was provided by Namseoul University year 2019
*Corresponding Author : Department of Electronic Engineering, Namseoul University(solomon645@nsu.ac.kr)
Received October 22, 2019 Revised October 31, 2019 Accepted November 19, 2019

발에 대한 필요성이 더욱 가중되고 있다. 이러한 문제를 해결하기 위해 소프트웨어 개발자들은 소프트웨어 신뢰성 향상과 경제적인 개발비용을 탐색하기 위해 지금도 많은 연구를 하고 있다. 이런 이유로, 소프트웨어 신뢰성향상을 위해 비동질 포아송 과정(Non-Homgeneous Poisson Process ; NHPP)을 이용한 소프트웨어 신뢰성 모형이 많이 제안되어 왔다[1]. 특히, Goel and Okumoto[2]는 지수 형태의 NHPP 신뢰성 모형을 제안하였고, Shyur[3]은 변환점을 이용하여 일반화된 신뢰성 모형을 제시하였으며, Kim[4]은 레일리 분포를 이용한 유한고장과 무한고장의 신뢰성을 비교 하였다. 더불어, Pham과 Zhang[5]은 테스팅 커버리지를 가지고 NHPP 소프트웨어 신뢰성에 근거한 비용모형을 제안하였으며, Zhang[6]은 소프트웨어 신뢰성 모형과 고장시간을 기반으로 새로운 소프트웨어 신뢰성 비용모형을 제안하기도 하였고, Yang[7]은 Erang 분포의 형상모수에 근거한 신뢰성 비용 모형을 연구하기도 하였다. 또한, Prasad와 Rao[8]는 소프트웨어 신뢰성 모형에 기반하여 효율적인 고장 예측 방법 및 학습과정을 설명하기도 하였다.

따라서, 본 연구에서는 유한고장 NHPP모형에 근거하여 레일리(Rayleigh) 분포와 역 레일리(Inverse Rayleigh) 분포를 적용하여 소프트웨어 개발비용 모형의 속성을 새롭게 분석하고, 이 분석을 통해 경제적인 개발비용 탐색정보와 최적의 방출시간을 예측하는데 필요한 정보를 제안하고자 한다.

2. 관련 연구

2.1 NHPP 소프트웨어 신뢰성 모형

$N(t)$을 시간 t까지 탐지된 소프트웨어의 고장의 누적수이고, $m(t)$를 이에 대한 평균값(고장 발생 기대값) 함수, $\lambda(t)$을 강도(결함당 고장 발생률) 함수로 표시하면, 누적 고장수 $N(t)$는 모수 $m(t)$을 가진 포아송 확률밀도 함수를 따른다고 하였다.

$$P\{N(t)=n\}=\frac{[m(t)]^n \cdot e^{-m(t)}}{n!} \quad (1)$$

단, $n=0,1,2 \ldots \infty$의 값을 갖는다.

여기서, NHPP 모형의 평균값 함수 $m(t)$와 강도함수 $\lambda(t)$는 다음과 같다.

$$m(t)=\int_0^t \lambda(s)ds \quad (2)$$

$$\frac{d\,m(t)}{dt}=\lambda(t) \quad (3)$$

시간관련 NHPP 모형들은 결함을 수리하는 시점에는 고장이 발생하지 않는다는 유한고장(Finite failure)과 결함을 수정하는 시점에도 고장이 발생한다는 무한고장(Infinite failure)으로 분류된다. 본 연구에서는 유한고장인 경우를 적용하여 NHPP 신뢰성모형의 개발비용 속성을 분석하고자 한다. 그러므로, 유한고장시 시간$(0,t]$까지 탐색될 수 있는 결함의 기댓값(=잔존 고장률)을 θ라고 하고, 결함 탐색률을 β라고 하면, 평균값 함수 $m(t)$와 강도함수 $\lambda(t)$는 다음과 같다.

$$m(t)=\theta \cdot F(t) \quad (4)$$

$$\lambda(t)=\theta \cdot F(t)'=\theta \cdot f(t) \quad (5)$$

임의 $[0,t]$ 시간에서 n번째까지 고장자료를

$$x_n=\sum_{i=1}^{n} t_i \quad (6)$$

단, $i=1,2,\cdots,n\,;0 \le x_1 \le x_2 \le \cdots \le x_n$ 이다.

Θ을 모수공간이라고 하면, NHPP모형의 우도함수(Likelyhood function)는 다음과 같다.

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

단, $\underline{x}=(x_1,x_2,x_3,\cdots,x_n]$ 이다.

2.2 유한고장 NHPP 레일리 모형

레일리 분포는 와이블(Weibull distribution) 수명분포에서 형상모수(α)가 2를 갖는 특수한 경우이다. 이러한 레일리 분포는 신뢰성 측정 및 수명(Life) 테스트 분야에 적합한 분포로 널리 알려져 있다. 레일리분포의 형상모수를 고려한 확률밀도함수$f(t)$와 누적분포함수 $F(t)$는 다음과 같다[9].

$$f(t)=\frac{t^{\alpha-1}}{\beta^2}\,e^{-\frac{t^{\alpha}}{2\beta^2}} \quad (8)$$

$$F(t) = 1 - e^{-\frac{t^{\alpha}}{2\beta^2}} \quad (9)$$

단, $\beta > 0,\ t \in [0, \infty]$의 값을 갖는다.

식(8)과 식(9)을 좀 더 간편하게 만들기 위해서 $\frac{1}{2\beta^2} = b$로 치환하여 정리하면 다음과 같다.

$$f(t) = 2b\, t^{\alpha-1} e^{-b t^{\alpha}} \quad (10)$$

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

단, $b > 0,\ t \in [0, \infty]$의 값을 갖는다.

식(10)과 식(11)에서 형상모수가 $\alpha = 1$인 경우는 지수분포(Exponential distribution)가 되고, $\alpha = 2$인 경우는 레일리 분포가 된다.

그러므로, 평균값 함수와 강도함수는 식(10)과 식(11)에서 형상모수 $\alpha = 2$를 대입하면 다음과 같다.

$$m(t|\theta, b) = \theta\, F(t) = \theta\left(1 - e^{-b t^2}\right) \quad (12)$$

$$\lambda(t\,|\theta, b) = \theta\, f(t) = 2\theta\, b\, t\, e^{-b t^2} \quad (13)$$

단, $\theta > 0,\ b > 0$ 조건을 만족한다.

따라서, 식(7)에 식(12)과 식(13)을 대입한 후, 로그우도함수를 구하면 다음과 같다.

$$\ln L_{NHPP}(\Theta|\underline{x}) = n\ln 2 + n\ln\theta + n\ln b + \sum_{i=1}^{n}\ln x_i - b\sum_{i=1}^{n} x_i^2 - \theta\,(1 - e^{-b x_n^2}) \quad (14)$$

그러므로, 식(14)을 모수 θ와 b에 대하여 편미분을 하면, 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{b}_{MLE}$값을 이분법(bisection Methode)으로 계산할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\Theta\,|\,\underline{x})}{\partial\theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}\, x_n^2} = 0 \quad (15)$$

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

단, $x = (x_1, x_2, x_3, \cdots, x_n)$ 이다.

2.3 유한고장 NHPP 역-레일리 모형

역-레일리 분포(Inverse-Rayleigh distribution)는 신뢰성 분야에서 널리 응용되는 분포로서, 여러 종류의 수명분포들은 역-레일리 분포로 설명될 수 있다. 이 분포의 확률밀도함수 $f(t)$와 누적분포함수 $F(t)$는 다음과 같다.

$$f(t) = \frac{2b}{t^3}\left(e^{-\frac{b}{t^2}}\right) \quad (17)$$

$$F(t) = e^{-\frac{b}{t^2}} \quad (18)$$

단, $b > 0,\ t \in [0, \infty]$의 값을 갖는다.

그러므로, 평균값 함수와 강도함수는 다음과 같다.

$$m(t|\theta, b) = \theta\, F(t) = \theta\left(e^{-\frac{b}{t^2}}\right) \quad (19)$$

$$\lambda(t\,|\theta, b) = \theta\, f(t) = \theta\left(\frac{2b}{t^3} e^{-\frac{b}{t^2}}\right) \quad (20)$$

따라서, 식(7)에 식(19)과 식(20)을 대입한 후, 로그우도함수를 구하면 다음과 같다.

$$\ln L_{NHPP}(\Theta|\underline{x}) = n\ln 2 + n\ln\theta + n\ln b + b\sum_{i=1}^{n}\ln\left(\frac{1}{x_i^3}\right) - b\sum_{i=1}^{n}\frac{1}{x_i^2} - \theta\, e^{-\frac{b}{x_n^2}} \quad (21)$$

그러므로, 식(21)을 모수 θ와 b에 대하여 편미분을 하면, 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{b}_{MLE}$값을 이분법(bisection Methode)으로 계산할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\Theta\,|\,\underline{x})}{\partial\theta} = \frac{n}{\hat{\theta}} - e^{-\frac{\hat{b}}{x_n^2}} = 0 \quad (22)$$

$$\frac{\partial \ln L_{NHPP}(\Theta\,|\,\underline{x})}{\partial b} = \frac{n}{\hat{b}} + \sum_{i=1}^{n}\ln\left(\frac{1}{x_i^3}\right) - \sum_{i=1}^{n}\left(\frac{1}{x_i^2}\right) + \frac{\hat{\theta}}{x_n^2} e^{-\frac{\hat{b}}{x_n^2}} = 0 \quad (23)$$

단, $x = (x_1, x_2, x_3, \cdots, x_n)$ 이다.

3. 소프트웨어 개발비용 모형

소프트웨어 개발비용 모형은 각 구성요소별 비용의 총합으로 다음과 같이 구성된다[6].

$$\begin{aligned} E_t &= E_1 + E_2 + E_3 + E_4 \\ &= E_1 + C_2 \times t + C_3 \times m(t) \\ &\quad + C_4 \times [m(t+t') - m(t)] \end{aligned} \quad (24)$$

단, E_t : 소프트웨어 개발 총비용을 나타낸다.

E_1은 초기 개발비용을 의미하며, E_2는 단위 시간당

테스팅 비용으로, 실제로 산업별 단위시간당 비용은 상이하다.

$$E_2 = C_2 \times t \quad (25)$$

단, C_2는 단위시간당 비용이고, t는 테스팅 시점이다. E_3는 기본 결함을 감지하고, 한 개의 고장을 제거하는 비용이다.

$$E_3 = C_3 \times m(t) \quad (26)$$

단, C_3는 테스팅 과정에서 발견된 한 개의 고장을 제거하는 비용, $m(t)$는 t시점에서 탐색할 수 있는 평균값 함수로 고장(결함)의 기댓값이다.

E_4는 소프트웨어 시스템에 잔존하는 모든 고장들을 제거하는 비용이다.

$$E_4 = C_4 \times [m(t+t') - m(t)] \quad (27)$$

단, C_4는 소프트웨어 출시이후에 사용 단계에서 운용자가 탐색한 고장을 수정하는 비용, t'는 소프트웨어를 정상적으로 사용할 수 있는 시간이다.

여기서, C_4 비용은 C_2 비용과 C_3 비용 보다는 높다는 판단이 현실적이다. 그러므로, 본 연구에서도 C_4 비용이 C_2 비용과 C_3 비용 보다는 높다는 현실적 상황을 적용하여 연구하였다.

또한, 소프트웨어 방출시간(t)는 소프트웨어 개발 총비용(E_t)이 최소가 되는 시점이 소프트웨어를 방출하는 시점이 된다. 그러므로, 소프트웨어 개발비용과 방출시간의 관계는 다음과 같은 식으로 표현 할 수 있다[8, 11].

$$\frac{\partial E_t}{\partial t} = E' = (E_1 + E_2 + E_{3+} E_4)' = 0 \quad (28)$$

4. 고장시간을 이용한 비용속성의 분석

본 연구에서는 [표 1]과 같은 소프트웨어 고장시간 자료를 사용하였는데, 이 자료의 고장시간은 187.35 시간 동안(단위)에 30번의 고장이 발생된 자료이다. 본 연구에서는 해당 고장시간 자료를 가지고 속성을 분석하고자 한다[10].

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

Failure number	Failure time(hours)	Failure interval-time	Failure time (hours)$\times 10^{-1}$
1	4.79	4.79	0.479
2	7.45	2.66	0.745
3	10.22	2.77	1.022
4	15.76	5.54	1.576
5	26.10	10.34	2.610
6	35.59	9.49	3.559
7	42.52	6.93	4.252
8	48.49	5.97	4.849
9	49.66	1.17	4.966
10	51.36	1.70	5.136
11	52.53	1.17	5.253
12	65.27	12.74	6.527
13	69.96	4.69	6.996
14	81.70	11.74	8.170
15	88.63	6.93	8.863
16	107.71	19.08	10.771
17	109.06	1.35	10.906
18	111.83	2.77	11.183
19	117.79	5.96	11.779
20	125.36	7.57	12.536
21	129.73	4.37	12.973
22	152.03	22.30	15.203
23	156.40	4.37	15.640
24	159.80	3.40	15.980
25	163.85	4.05	16.385
26	169.60	5.75	16.960
27	172.37	2.77	17.237
28	176.00	3.63	17.600
29	181.22	5.22	18.122
30	187.35	6.13	18.735

본 연구에서는 라플라스 추세검정(Laplace trend test)을 사용하여, [표 1]의 고장시간자료에 대한 신뢰성을 확인하였다. 일반적으로 임의 자료에 대한 라플라스 요인(Factor)의 추정 값이 "-2와 2"사이에 분포하면 안정적이기 때문에 신뢰성을 가진다고 한다. 그러므로, [그림 1]과 같이 라플라스 추세검정 결과가 "0과 2"사이에 분포했기 때문에 해당 자료를 사용하는 것은 유효하다[11].

또한, 모수추정은 최우추정법(Maximum Likelihood Estimation ; MLE)을 이용하였고, 수렴성을 쉽게 하기 위하여 원래의 고장자료를 수치변환($Failure\ time \times 10^{-1}$)하여, 적용하였다.

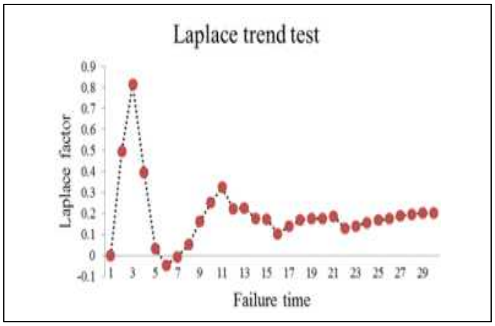

그림 1. 라플라스 검정의 추정결과
Fig. 1. Estimation results of Laplace test

또한, 비선형 방정식의 모수 추정은 이분법을 사용하였고, 그 결과는 [표 2]와 같다[12].

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

Model	MLE	
	$\hat{\theta}$	$\hat{b}$
Rayleigh	30.0412	1.8835
Inverse-Rayleigh	30.0100	0.1652

본 연구에서는 실제 개발환경을 투입하여 [가 정 1]과 [가정 2]와 같은 상황으로 소프트웨어 개발비용의 속성을 분석하고자 한다.

[가정1 : 기본조건]

$E_1 = 5\$,\ c_2 = 5\$,\ c_3 = 1.5\$,\ c_4 = 10\$,\ t' = 10$ (29)

[가정 1]과 같은 기본 조건을 적용한 개발비용의 시뮬레이션 결과는 [그림 2]와 같다. 이 결과에서 비용곡선의 추이는 초기단계는 급격하게 감소한 후, 단기간 일정한 패턴을 유지하다가 방출시간이 흐를수록 점진적으로 증가하는 패턴을 보이고 있다. 그 이유는 초기단계에서 결함을 제거하는 과정 중에 소프트웨어에 내재한 결함의 수는 줄어들기 때문에 비용은 감소하지만, 후반단계에서는 남아있는 결함이 발견될 확률은 감소하기 때문에, 반대로 개발 비용은 방출시간에 비례하여 증가한다. 결국, 방출시간이 흐를수록 개발비용 곡선의 패턴은 증가하는 형태가 된다.

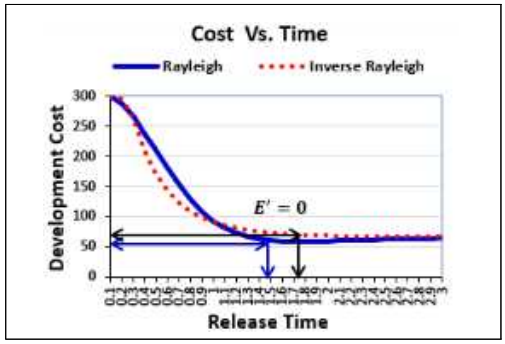

그림 2. [가정1]을 적용한 개발비용 곡선
Fig. 2. The cost curve applied to the condition of [$Supposition\,1$]

[그림 2]와 같이 시뮬레이션 결과를 보면, 레일리 모형의 개발 비용은 50$, 방출시간은 1.48H이며, 역-레일리 모형의 개발 비용은 70$, 방출시간은 1.72H이다. 그러므로, 레일리 모형이 역-레일리 모형보다 개발비용이 적고, 방출시점도 빨라서 효율적인 모형임을 알 수 있다.

[가정2 : 가정1에서 C_3 비용이 증가한 경우]

$E_1 = 5\$,\ c_2 = 5\$,\ c_3 = 3\$,\ c_4 = 10\$,\ t' = 10$ (30)

[가정 2]는 [가정 1]과 비교하여, 다른 조건은 모두 동일하지만, 테스팅 과정에서 발견된 한 개의 고장을 제거하는 비용(C_3)을 2배 증가(1.5$→3$)시킨 상황으로서, 시뮬레이션 결과는 [그림 3]과 같다.

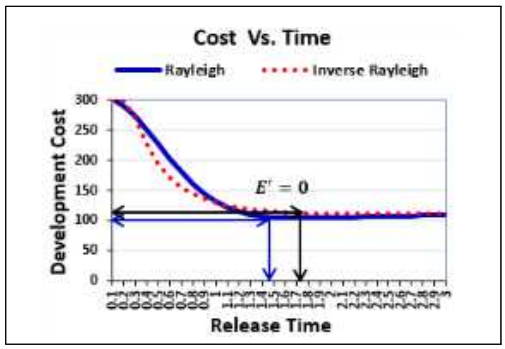

그림 3. [가정2]을 적용한 개발비용 곡선
Fig. 3. The cost curve applied to the condition of [$Supposition\,2$]

[그림 3]과 같이 시뮬레이션 결과를 보면 레일리 모형의 개발 비용은 100$, 방출시간은 1.48H이며, 역-레일리 모형의 개발 비용은 120$, 방출시간은 1.72H이다. 그러므로, 레일리 모형이 역-레일리 모

형보다 개발비용이 적고, 방출시점도 빨라서 효율적인 모형임을 알 수 있다.
또한, 개발비용은 증가하였지만, 방출시간에는 변함이 없었다. 이 경우에는 테스팅 단계에서 한 개의 고장을 제거하는 비용이 증가하지 않도록 가능한 한 번에 많은 결함을 제거해야 한다.

[가정3 : 가정1에서 C_4 비용이 증가한 경우]

$$E_1 = 5\$,\ c_2 = 5\$,\ c_3 = 1.5\$,\ c_4 = 20\$,\ t' = 10 \quad (31)$$

[가정 3]과 같은 상황 조건을 적용한 비용 곡선의 시뮬레이션 결과는 [그림 4]와 같다.

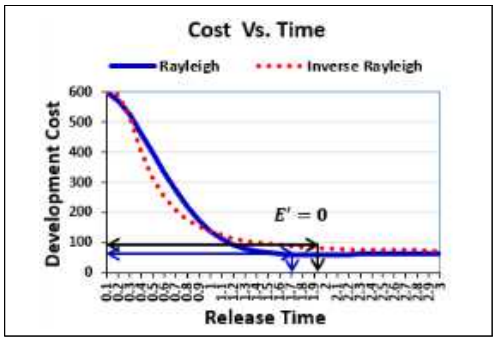

그림 4. [가정3]을 적용한 개발비용 곡선
Fig. 4. The cost curve applied to the condition of [*Supposition* 3]

[가정 3]은 [가정 1]과 비교하여, 다른 조건은 모두 동일하지만, 소프트웨어 출시 이후에 실제 운용단계에서 사용자가 발견하는 고장을 수정하는 비용(C_4)을 2배 증가(10$→20$)시킨 상황이다. [그림 4]의 시뮬레이션 결과를 보면, 개발비용 증가와 함께 방출시간도 함께 지연된다는 것을 알 수 있다. 이 경우에는 소프트웨어 출시 이전에 결함을 감소시킬 수 있도록 운영단계 보다는 테스팅 단계에서 가능한 모든 결함을 제거해야 한다. 즉, 레일리 비용모형이 역-레일리 비용모형보다 개발비용이 적고, 방출시점도 빨라서 상대적으로 우수한 모형임을 알 수 있다.

5. 결론 및 향후 연구과제

실제 소프트웨어 개발 과정이나 테스팅 단계에서 발생할 수 있는 고장시간에 대한 신뢰성 분석자료를 소프트웨어 개발비용 모형에 적용하면, 더욱 경제적인 개발 비용을 탐색할 수 있다. 그러므로, 본 연구에서는 유한고장 NHPP 신뢰성모형을 가지고 레일리 분포와 역레일리 분포의 개발 비용모형의 속성을 탐색하고, 분석하였다.

본 연구의 결과는 다음과 같다.

첫째, 기본 조건하에서 비용곡선은 초기단계에서 급격히 감소한 후, 일정한 추세를 보이다가 고장시간이 흐른 후반단계에는 점점 증가하는 형태를 보였다. 그 이유는 초기결함을 제거하는 과정에서 잔존결함의 수는 크게 감소하지만, 후반단계에서는 잔존결함이 발견될 확률이 점점 줄어들어, 결국 비용이 증가하기 때문임을 알 수 있었다.

둘째, 소프트웨어 출시이전에 발견된 한 개의 결함을 제거하는 비용(C_3)이 증가하면 개발비용은 증가하였지만, 반대로 방출시간에는 변함이 없었다. 하지만, 소프트웨어 출시이후에 운용자가 발견하는 결함수정 비용(C_4)이 증가하는 경우에는, 전체 고장시점에서의 개발비용도 증가하고, 방출시간도 함께 지연된다는 것을 알 수 있었다.

셋째, 본 연구에서 적용한 모형을 분석한 결과, 레일리 비용모형이 역-레일리 비용모형보다 개발비용이 비교적 적고, 방출시점도 빨라서 상대적으로 우수한 모형임을 알 수 있었다.

본 연구결과를 이용하면 소프트웨어 개발자 및 운용자들에게 경제적인 소프트웨어 개발비용 분석과 함께 경제적인 소프트웨어 개발비용 분석과 함께 최적의 방출시간을 예측하는데 필요한 더불어 추후에 소프트웨어 개발자를 위해 다양한 형태의 소프트웨어 고장시간 자료를 가지고, 여러 신뢰성 분포 모형에 근거하여, 이를 소프트웨어 개발비용에 적용한 후속성 분석을 통해 최적의 소프트웨어 개발 모형을 탐색하여 제시하는 후속 연구가 필요하겠다.

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 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

[3] Shyur H-J, "A stochastic software reliability model with imperfect debugging and change-point", J. Syst. Software 66, pp.135-141, 2003.

[4] K. S. Kim, H. C. Kim "The Comparative Study for Software Reliability Model Based on Finite and Infinite Failure Property using Rayleigh Distribution", Journal of Digital Con., Vol.12, No.12, pp.277-284, 2014.

[5] Pham H, Zhang X., "NHPP software realiability and cost models with testing coverage", Eur.J. Oper. Res, 145, pp.445 454, 2003.

[6] 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.

[7] 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.

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

[9] 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.

[10] Hayakawa, Y. and Telfar, G., "Mixed poisson-type processes with application in software are reliability", Mathematical and Computer Modelling, Vol. 31, pp.151 -156, 2000.

[11] T. J. Yang, "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", The Journal of Korea Institute of Information Electronics and Communication Technology, Vol 11, No 4, 2018, pp.460-466.

[12] T. J. Yang, "The Performance Analysis Comparative Study depend on Software Reliability Model and Curve Regression Model", Medwell Journals, Vol.12, No.5, pp.313-317, 2017.

저자약력

양 태 진 (Tae-Jin Yang) **[정회원]**

- 1992년 2월 : 한양대학교 전자공학과 공학석사
- 1995년 2월 : 한양대학교 전자공학과 공학박사(수료)
- 1986년 3월 ~ 1992년 2월 : 현대그룹 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