

수명분포가 와이블 분포의 형상모수를 고려한 소프트웨어 개발 비용모형에 관한 속성분석 연구

김 희 철*

A Study on Attribute Analysis of Software Development Cost Model about Life Distribution Considering Shape Parameter of Weibull Distribution

Hee-Cheul Kim*

요 약 소프트웨어 안정성은 운영 환경에서 시간의 흐름에 따른 오작동이 없이 운영 될 수있는 가능성이라고 할 수 있다. 소프트웨어의 고장 분석을 위한 유한 고장 NHPP에서, 고장 발생률은 일정하거나 단조롭게 증가하거나 단조 감소하는 추이를 나타낼 수 수 있다. 본 연구에서는 NHPP 모형에 근거하고 소프트웨어 고장시간 자료를 바탕으로 와이블 분포의 형상모수를 고려한 지수분포 Rayleigh 분포, 역-지수 분포를 수명분포로 하여 소프트웨어 개발 비용모형에 관한 속성을 비교 평가분석을 하였다. 또한 모수 추정은 최우 추정방법을 적용하고 데이터 추세검정은 박스-플롯방법을 이용하였다. 본 연구의 결과는 Rayleigh 모형이 역-지수 모형이나 Goel-Okumoto 모형에 비교해서 방출 시기는 Rayleigh 모형이 가장 빠르고 방출시점의 비용도 가장 경제적인을 알 수 있다. 이 연구의 결과를 이용하면 소프트웨어 개발자 및 운용자들은 최적방출시간과 경제적인 개발비용을 예측 하는데 활용 할 수 있으리라 판단된다.

Abstract Software stability is the possibility of operating without any malfunction in the operating environment over time. In a finite failure NHPP for software failure analysis, the failure occurrence rate may be constant, monotonically increasing, or monotonically decreasing. In this study, based on the NHPP model and based on the software failure time data, we compared and analyzed the attributes of the software development cost model using the exponential distribution Rayleigh distribution and inverse exponential distribution considering the shape parameter of the Weibull distribution as the life distribution. The results of this study show that the Rayleigh model is the fastest release time and has the economic cost compared to the inverse-exponential model and the Goel-Okumoto model. Using the results of this study, it can be expected that software developers and operators will be able to predict the optimal release time and economic development cost.

Key Words : Box-Plot, Goel-Okumoto model NHPP, Software cost model, Weibull distribution.

1. 서론

소프트웨어 안정성은 운영 환경에서 시간의 흐름에 따른 오작동이 없이 운영 될 수있는 가능성이라고 할 수 있다. 따라서 소프트웨어 안정성은 소프트웨어 시스템 안정성에 영향을 미치는 중요한 요소이며 하드웨어 안정성 측면에서 디자인 속성과는 다른 속성을 가진다.

따라서 컴퓨터 시스템의 고장은 우리사회의 소프트웨어 고장에 막대한 손실을 초래할 수 있다. 따라서 소프트웨어 개발과정에서 소프트웨어 안정성은 주요 근본적인 문제이다.

따라서 소프트웨어 인도 예상시간을 고려한 신뢰성, 비용 및 소프트웨어, 개발 프로세스가 반드시 필요하다 [1]. 따라서 오류 탐색 과정에서 소프트웨어 신뢰도 모

Funding for this paper was provided by Namseoul University year 2018

*Corresponding Author : Associate Professor, Department of Industrial & Management Engineering, Namseoul University (kim1458@nsu.ac.kr)

Received October 30, 2018

Revised November 06, 2018

Accepted November 12, 2018

텔은 비동질 포아송 공정 (NHPP)을 사용하여 새로운 결함이 발생하면 즉시 제거되고 새로운 결함은 디버깅 프로세스에서 발생하지 않은 다는 가정을 하였다[2]. Huang [3]은 효과적인 소프트웨어 신뢰성 예측 기술의 통합측면에서, S-형상 모델은 소프트웨어의 신뢰성 예측 기술을 제시 하였다. 또한 S-형상 모델은 소프트웨어 관리자가 소프트웨어 오류 탐지 환경을 해결할 수 있는 학습과정을 설명하기도 하였다 [4]

본 연구에서는 NHPP 모형에 근거하고 소프트웨어 고장시간 자료를 바탕으로 수명분포가 와이블 분포의 형상모수를 고려한 소프트웨어 개발 비용모형에 관한 속성을 비교 평가분석을 하고자 한다.

2. 배경연구

2.1 와이블 분포의 형상모수를 고려한 모형

와이블 분포의 확률밀도함수와 누적분포는 다음과 같다 [1, 5].

$$f(t) = \alpha \beta t^{\alpha-1} e^{-\beta t^\alpha}, \quad t > 0 \quad (1)$$

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

(1)식과 (2)식에서 형상모수 $\alpha=1$ 인 경우는 지수분포(Exponential distribution)가 되고 $\alpha=2$ 인 경우는 랄리분포(Rayleigh distribution)가 된다 [6]. 따라서 지수분포를 적용한 경우 관찰시간 $(0, t]$ 까지 관찰된 결함의 기대 값을 θ 라고 하고 결함탐색률을 β 라 하면 NHPP모형의 유한고장 강도함수와 평균값 함수는 다음과 같은 형태로 알려져 있다[2, 7].

$$\lambda(t|\theta, \beta) = \theta f(t) = \theta \beta e^{-\beta t}, \quad (\theta > 0, \beta > 0) \quad (3)$$

$$m(t|\theta, \beta) = \theta F(t) = \theta (1 - e^{-\beta t}) \quad (4)$$

이 모형은 Goel-Okumoto 모형으로 알려져 있다[8]. 유사하게 랄리분포(Rayleigh distribution)를 적용한 경우 NHPP모형의 유한고장 강도함수와 평균값함수는 다음과 같은 특징으로 유도 된다 [9].

$$\lambda(t|\theta, \beta) = 2\theta \beta t \exp(-\beta t^2) \quad (5)$$

$$m(t|\theta, \beta) = \theta [1 - \exp(-\beta t^2)] \quad (6)$$

2.2 역-와이블분포의 형상모수를 고려한 모형

역-와이블분포의 누적분포는 다음과 같다 [9, 10].

$$F(t) = \exp[-(\beta t)^{-\gamma}], \quad t > 0 \quad (7)$$

형상모수 $\gamma=1$ 인 경우는 역-지수분포(Inverse exponential distribution)가 된다.

따라서 역-지수분포의 확률밀도함수와 누적분포는 다음과 같은 형태로 유도 된다 [9, 10].

$$f(t) = \beta^{-1} t^{-2} \exp[-(\beta t)^{-1}], \quad t > 0 \quad (8)$$

$$F(t) = \exp[-(\beta t)^{-1}], \quad t > 0 \quad (9)$$

역-지수분포를 적용한 경우 NHPP모형의 유한고장 강도함수와 평균값 함수는 다음과 같은 속성으로 요약 있다[2, 9].

$$\lambda(t|\theta, \beta) = \theta \beta^{-1} t^{-2} \exp[-(\beta t)^{-1}] \quad (10)$$

$$m(t|\theta, \beta) = \theta \exp[-(\beta t)^{-1}] \quad (11)$$

2.3 모수추정

관측시간 $(0, t]$ 까지 관찰하기 위한 정시중단(Time truncated)특징은 n 번째까지 고장시점 자료 (t_i 은 고장간격시간)는 다음과 같은 속성으로 요약 할 수 있다 [1, 2].

$$x_n = \sum_{i=1}^n t_i \quad (i=1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (12)$$

(10)식을 이용한 NHPP모형에 대한 우도함수는 다음과 같다 [7, 8].

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp[-m(x_n)] \quad (13)$$

단, θ 는 모수공간이고 $\underline{x} = (0 \leq x_1 \leq x_2 \leq \dots \leq x_n)$.

Goel-Okumoto인 경우 (13)식을 이용한 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 모수 추정은 과정조건을 만족해야 한다 [8]

$$\frac{n}{\hat{\theta}} = 1 - e^{-\hat{\beta} x_n} \quad (14)$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^n x_i + \hat{\theta} x_n e^{-\hat{\beta} x_n} \quad (15)$$

Goel-Okumoto 모형과 유사하게 Rayleigh형 모형에서도 최우추정값 $\hat{\theta}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 을 다음과 같은 특징으로 추정 할 수 있다 [9].

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-\beta x_n^2} = 0 \quad (16)$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \beta} = \quad (17)$$

$$\frac{n}{\beta} - \sum_{i=1}^n x_i^2 - \theta x_n^2 e^{-\beta x_n^2} = 0$$

또한 역지수모형에서도 최우추정값 $\hat{\theta}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 을 다음과 같은 방법으로 추정 할 수 있다 [9].

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - e^{-(\beta x_n)^{-1}} = 0 \quad (18)$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \beta} = \quad (19)$$

$$-\frac{n}{\beta} + \frac{1}{\beta^2} \sum_{i=1}^n \frac{1}{x_i} - \theta \frac{1}{\beta^2 x_n} e^{-(\beta x_n)^{-1}} = 0$$

2.4 소프트웨어 개발 비용 모형

소프트웨어 비용 모형의 다음과 같은 속성으로 구조화 된다[11, 12].

$$\begin{aligned} E &= E_1 + E_2 + E_3 + E_4 \\ &= E_1 + C_2 \times t + C_3 \times m(t) + C_4 \times [m(t+t') - m(t)] \end{aligned} \quad (20)$$

단, E : 소프트웨어 개발 예상 총비용
 E_1 은 데이터 분석료, 투입된 소프트웨어 개발전문가 인건비, 사무실 임대료 등과 같은 소프트웨어 설계 및 초기 소프트웨어 관련된 개발 비용을 의미하고 상수로 가정한다. 또한 E_2 는 단위 시간당 소프트웨어 테스트 비용을 나타내며 다음과 속성으로 표현된다.

$$E_2 = C_2 \times t \quad (21)$$

단, C_2 는 단위 시간당 비용을 의미하고 t 는 테스트 시점을 의미한다. 그리고 E_3 는 내재되어 있는 결함을 감지하고 하나의 결함을 제거할 때 투입되는 비용을 나타내며 다음과 관계로 표현된다.

$$E_3 = C_3 \times m(t) \quad (22)$$

단, C_3 는 테스트 과정에서 탐색된 하나의 결함을 제거할 때 투입되는 비용, $m(t)$ 는 t 시점에서 탐색되어 질 수 있는 결함의 기대수를 나타낸다. 또한 E_4 는 소프트웨어 시스템 운영과정에서 남아있는 모든 결함들을 제거하는 비용(상수)을 의미하고 다음과 관계로 표현된다.

$$E_4 = C_4 \times [m(t+t') - m(t)] \quad (23)$$

C_4 는 소프트웨어 방출 된 이후에 소프트웨어 운영 과정에서 운용자가 탐색된 결함수정 비용, t' 는 소프트웨어 시스템을 출시 한 이후 소프트웨어를 유지할 수 있는 시간을 나타낸다. 일반적으로는 C_4 는 C_2 와 C_3 보다 높은 비용이 투입되어야 한다. 따라서 최적의 최적 소프트웨어 방출시간 (t)는 다음과 같은 속성으로 나타낼 수 있다[11, 12].

$$\frac{\partial E}{\partial t} = E' = (E_1 + E_2 + E_3 + E_4)' = 0 \quad (24)$$

3. 소프트웨어 고장시간 분석 및 개발 비용 속성 비교분석

장에서 소프트웨어 고장시간 정보자료[13] (Failure time information data)를 적용하여 신뢰 모형들의 특징 및 개발비용을 비교 분석하고자 한다.

이 자료는 [표 1]에 수록하였고 이상값(Ouliers)이 존재하는지를 탐색하기 위하여 자료에 관한 추세검정을 활용 할 수 있는데 본 연구에서는 박스-플롯(Box-plot)을 이용하였다[7]. 박스-플롯 추세검정의 결과는 [그림 1]에 요약되었다. 이 결과에서 상한 (= 15.04 + 1.5*(15.04 - 4.849) = 30.3265)과 하한 (= 4.849 - 1.5*(15.04 - 4.849) = - 40.6408)사이를 벗어

난 자료가 존재하지 않기 때문에 이상값 혹은 극단값이 탐색되지 않았다. 그러므로 이 자료를 활용하여 신뢰성 모형의 속성을 예측하는 과정이 유용성을 가진다[7].

모수추정방법은 전통적인 최우추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 방법인 이분법(Bisection method)을 이용하였다. 이러한 계산에서 모수 β 값을 추정하기 위하여 초기값을 0.01과 2.00을, 허용한계(Tolerance for width of interval)는 10^{-5} 을 투여하고 수렴 성을 확인하면서 R 소프트웨어를 이용하여 모수추정을 수행하였다. 그 결과는 [표 2]에 나타내었다.

표 1. 고장시간자료
Table 1. Failure time data

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	2.610	20	12.536
6	3.559	21	12.973
7	4.252	22	15.203
8	4.849	23	15.640
9	4.966	24	15.980
10	5.136	25	16.385
11	5.253	26	16.960
12	6.527	27	17.237
13	6.996	28	17.600
14	8.170	29	18.122
15	8.863	30	18.735

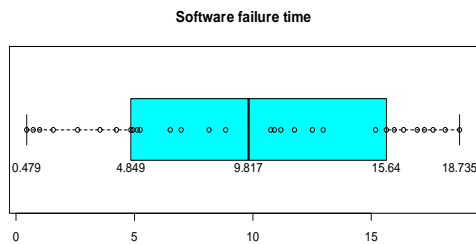


그림 1. 박스플롯의 결과
Fig. 1. Result of Box-plot

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

Model	MLE	
Goel-Okumoto	$\hat{\theta} = 32.9261$	$\hat{\beta} = 0.1297$
Rayleigh	$\hat{\theta} = 30.0412$	$\hat{\beta} = 0.0188$
Inverse exponential	$\hat{\theta} = 41.2881$	$\hat{\beta} = 0.1692$

본 연구에서는 (18)식을 이용하여 효율적인 값을 가정하여 비용 곡선의 속성을 비교 및 분석하고자 한다.

[Supposition] (25)

$$E_1 = 10\$, c_2 = 0.5\$, c_3 = 0.8\$, c_4 = 0.9\$, t' = 3.5$$

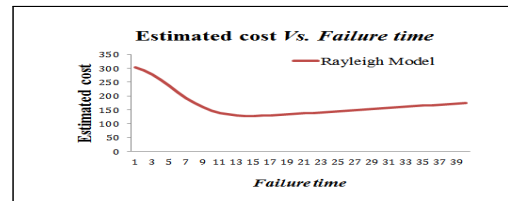


그림 2. [Supposition]을 적용한 비용곡선

Fig. 2. The cost curve applied to the condition of [Supposition]

수명분포가 Rayleigh 분포이고 (25)식의 가정을 적용한 비용곡선의 패턴은 [그림 2]에 나열되었다. 이 그림에서의 비용곡선 양상은 초기단계의 상황은 감소 패턴을 나타내다가 일정시간 이후에는 일정패턴을 나타내다가 고장시간이 지나는 후반상황에서는 점차 증가패턴을 보이는 양상으로 나타나고 있다. 그 이유는 소프트웨어 시스템의 내재하는 결함의 수는 결함의 제거하는 과정에서 점점 줄어들기 때문에 내재되어 있는 결함이 탐색 될 확률은 점점 낮아지게 되기 때문이다. 따라서 후반 상황의 테스트 단계에서는 결함을 탐색할 시간이 상대적으로 길고 결함을 제거하는 비용은 상대적으로 높게 나타나기 때문에 비용곡선의 양상은 테스트 시간이 지남에 따라 계속 증가하는 패턴이 된다. 이러한 비용곡선의 추이 정보를 사

용하면 최적의 소프트웨어 방출시간을 예측 할 수 있다. 이러한 테스트과정은 일반적인 패턴이며 일반적으로 실제 소프트웨어 개발의 작업과정에서 나타나는 패턴이라고 알려져 있다[11, 12].

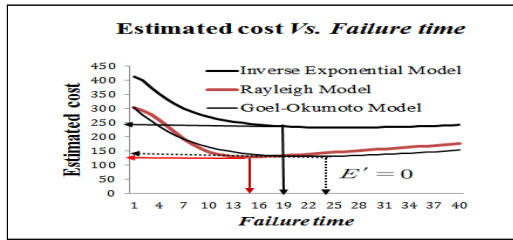


그림 3. 각 모형에서 [Supposition]을 적용한 비용곡선
Fig. 3. The cost curve applied to the condition of [Supposition] for each model

[그림 3]에서 기본 모형인 Go-Okumoto 모형과 Rayleigh 모형 그리고 역지수 모형을 수명분포를 적용시킨 비용곡선의 패턴에 대한 그림이 요약되었다. 이 그림에서 모든 모형이 대하여 비용곡선 양상은 초기단계의 상황은 감소패턴을 나타내다가 일정시간 이후에는 일정패턴을 나타내다가 고장시간이 지나는 후반상황에서는 점차 증가패턴을 보이는 양상으로 나타나고 있다. 그러나 소프트웨어 방출시기는 Rayleigh 모형이 역-지수 모형이나 전 고장시간에서 기본 모형인 Goel-Okumoto 모형과 Rayleigh 모형이 역-지수 모형보다 보다 비용이 감소하고 있어 상대적으로 효율적 모형으로 간주 된다. Goel-Okumoto 모형에 비해 빠르게 나타나고 있다. 그리고 역-지수 모형과 Goel-Okumoto 모형과의 방출시기에 대한 비교해서 역-지수 모형이 빠르긴 하지만 비용측면에서 높게 나타나고 있다. 결국 예상 한 것처럼 Rayleigh 모형이 역-지수 모형이나 Goel-Okumoto 모형에 비교해서 방출 시기는 Rayleigh 모형이 가장 빠르고 출시시점의 비용도 가장 경제적인임을 알 수 있다.

4. 결론

소프트웨어 개발 작업 단계에서 테스트 과정이나 실제 소프트웨어 운용과정에서 내장된 고장발생 현상 혹은 고

장 발생추이를 계량적으로 분석 할 수 있으면 소프트웨어 성능을 비교 분석하여 상대적으로 유용성 비교 평가를 할 수 있다. 본 연구에서는 소프트웨어 고장 시간 자료를 활용하여 NHPP 모형을 기반으로 하여 수명분포가 와이بل 분포의 형상모수를 고려한 Goel-Okumoto 모형과 Rayleigh 모형 그리고 역-지수 모형을 적용하여 소프트웨어 개발 비용모형에 관한 속성을 분석 하였다.

본 연구의 결과는 다음과 같이 요약 할 수 있다.

첫째, 본 연구에서 제한한 모형들의 비용곡선의 양상은 초기단계의 상황은 감소패턴을 나타내다가 일정시간 이후에는 일정패턴을 나타내다가 고장시간이 지나는 후반상황에서는 점차 증가패턴을 보이는 양상으로 나타나고 있다. 둘째, 전 고장시간에서 기본 모형인 Goel-Okumoto 모형과 Rayleigh 모형이 역-지수 모형보다 보다 비용이 감소하고 있어 상대적으로 효율적 모형으로 간주 된다. 셋째, Rayleigh 모형이 역-지수 모형이나 Goel-Okumoto 모형에 비교해서 방출 시기는 Rayleigh 모형이 가장 빠르고 출시시점의 비용도 가장 경제적인임을 알 수 있다. 이 연구의 결과를 이용하면 소프트웨어 개발자 및 운용자들은 최적방출시간과 경제적인 개발비용을 예측 하는데 활용 할 수 있으리라 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.
- [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] Kuei-Chen, C., Yeu-Shiang, H., & Tzai-Zang, L., "A study of software reliability

growth from the perspective of learning effects”, *Reliability Engineering and System Safety*, Vol. 93, pp. 1410-1421, 2008.

[5] Pham, Hoang, "Springer Handbook of Engineering Statistics", pp.63-78, *Springer-Verlag, London*, 2006.

[6] https://en.wikipedia.org/wiki/Weibull_distribution

[7] Hee-Cheul 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. 35 1-354, 2017.

[8] Goel, A.L., & Okumoto, K. "Time Dependent Error Detection Rate Model for software Reliability and other performance measure". *IEEE Trans. Reliability*, Vol. R-28, No. 3, pp. 206-211, 1979.

[9] Hee-Cheul Kim, "The Comparative Study for Statistical Process Control for Software Reliability Property Utilizing Inverse Rayleigh and Rayleigh Distribution", *Journal of Engineering and Applied Sciences*, Vol 13(Special Issue 1), pp.2500-2504, 2018

[10] Rao, R. K. H., R. S, Prasad and R. R. L., "Assessing software reliability using SPC: An order statistics approach", *Int. J. Comput. Sci. Eng. Applic.*, Vol. 1, pp.123-131, 2011.

[11] Zhang Y, Wu K. "Software cost model considering reliability and time of software in use". *Journal of Convergence Information Technology*, Vol 7, No 13, pp.135-142, 2012.

[12] Hee-Cheul Kim, "A Comparative Study on the Cost of Software Development Model Based on Inverse Exponential Distribution", *International Journal of Applied Business and Economic Research*, Vol.15, No. 14, pp.1-7, 2017.

[13] Y. Hayakawa, G. Telfar "Mixed poisson-type processes with application in software are reliability", *Mathematical and Computer Modelling*, Vol. 31, pp.1 51-156, 2000.

[14] Kim tai-dal, 'Software development project management using Agile methodology', *The Journal of The Institute of Internet, Broadcasting and Communication* VOL. 16 No. 1, 2016

저자약력

김희철 (Hee-Cheul Kim)

[정회원]



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

〈관심분야〉

소프트웨어 신뢰성 공학, 전산 통계, 웹-프로그래밍