

A Comparative Study on the Cost Analysis of Software Development Model Based on Shape Parameters of NHPP Software Reliability Model

Yoo-Chan Ra¹

¹ Professor, Department of Information Communication Engineering,
Namseoul University, South Korea.

Abstract

In this study, after applying the NHPP Rayleigh distribution model and the NHPP Erlang distribution model which are widely used in the field of software reliability together with the Goel-Okumoto basic model to the software development cost model, the attributes of cost were compared and analyzed. For this, 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 the Rayleigh distribution development model is the best efficient model because it has a lower software development cost and faster software release point than proposed other models. Through this study, the software development cost attributes of the Rayleigh model and the Erlang model without the existing research examples were newly analyzed. Also, we were able to present software developers and operators with the necessary prior information to predict the economic software development cost and the optimal release time.

Keywords : Erlang Lifetime Distribution, NHPP Model, Rayleigh Lifetime Distribution, Shape Parameter, Software Development Cost, Software Development Model.

1. INTRODUCTION

As the software-based convergence technology is rapidly spreading to the high-tech industries, the need for high-quality software development is also increasing. In the course of developing such software, it is possible to carry out more economical development work if the optimal development cost and release time can be predicted. For this reason, many development cost models based on software reliability have been proposed to date. In particular, to estimate the reliability performance under given software failure time data, the software reliability models based on the non-homogeneous Poisson process (NHPP) using the intensity function and the mean value function were developed and proposed [1]. Yamada and Osaki [2] stated that the results of the mean value function can be estimated using the maximum likelihood estimation method, while Huang [3] presented and explained a graph showing the confidence interval of the mean value function. Pham and Zhang [4] proposed a software reliability cost model based on the software reliability model and

software failure time data. Also, using this NHPP reliability model, Kim [5] analyzed the cost of the software development model based on the Burr-Harke exponential distribution, and also proposed a new software development model with NHPP Gompertz distribution [6]. In particular, Yang [7] analyzed the attributes of the finite failure NHPP software reliability model based on Weibull life distribution.

Therefore, in this study, after applying the NHPP Rayleigh distribution model and the NHPP Erlang distribution model widely used in the software reliability field together with Goel-Okumoto basic model to the software development model, we were newly analyzed the cost attributes of the proposed models and will present the best software development model through this analysis results.

2. RELATED RESEARCH

2.1 Goel-Okumoto basic model : NHPP model

The Goel-Okumoto is a well-known basic model in the software reliability field. This model assumes the exponential distribution as the lifetime distribution per fault. Therefore, the rate of occurrence of faults is constant. Let $f(t)$ and $F(t)$ for the Goel-Okumoto model be a probability density function and a cumulative density function, respectively. Assuming that the expected value of the number of failures of the observation point $[0, t]$ is θ , the finite failure strength function and the mean value function are as follows [8].

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

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

Note. $\theta > 0, b > 0$.

Considering Eq. 1 and Eq. 2, the likelihood function of the finite-failure NHPP model is derived as follows.

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

Note. $\underline{x} = (0 \leq x_1 \leq x_2 \leq \dots \leq x_n)$.

The log-likelihood function, using Eq. 3, is simplified to the following log conditional expression.

$$\ln L_{NHPP}(\theta|\underline{x}) = n \ln \theta + n \ln b - b \sum_{k=1}^n x_k - \theta(1 - e^{-bx_n}) \quad (4)$$

Note. $\underline{x} = (0 \leq x_1 \leq x_2 \leq \dots \leq x_n,$

θ is parameter space.

When the Eq. 4 is partially differentiated concerning θ and b , the result is as follows.

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + \exp(-\hat{b}x_n) = 0 \quad (5)$$

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial b} = \frac{n}{\hat{b}} - \sum_{i=1}^n x_i - \hat{\theta} x_n \exp(-\hat{b}x_n) \quad (6)$$

Note. $x = (x_1, x_2, x_3 \dots x_n)$

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and \hat{b}_{MLE} can be calculated using the bisection method.

2.2 Erlang distribution : NHPP model

The Erlang distribution is the lifetime distribution of the Gamma family widely used in the reliability field. The probability density function and the cumulative density function considering the shape parameter (a) and the scale parameter (b) are as follows [9].

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

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

Note. $a, b > 0, a = 1, 2, 3 \dots, t \in [0, \infty]$

Assuming that the expected value of the number of failures of the observation point $[0, t]$ is θ , the finite failure strength function and the mean value function are as follows.

$$\lambda(t|\theta, b) = \theta f(t) = \theta \left(\frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt} \right) \quad (9)$$

$$m(t|\theta, b) = \theta F(t) = \theta \left(1 - e^{-bt} \sum_{i=0}^{a-1} \frac{(bt)^i}{i!} \right) \quad (10)$$

The log-likelihood function to maximum likelihood estimation by using Eq. 9 and Eq.10 is derived as follows.

$$\ln L_{NHPP}(\theta|\underline{x}) = n \ln \theta - n \ln \Gamma(a) + n a \ln b + (a-1) \sum_{i=1}^n \ln x_i - b \sum_{i=1}^n x_i - \theta + \theta e^{-bx_n} \left(\sum_{i=0}^{a-1} \frac{(bx_n)^i}{i!} \right) \quad (11)$$

Note. $\underline{x} = (0 \leq x_1 \leq x_2 \leq \dots \leq x_n,$

θ is parameter space.

When the Eq. 11 is partially differentiated concerning θ and b , the result is as follows.

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}x_n} \left(\sum_{i=0}^{a-1} \frac{(\hat{b}x_n)^i}{i!} \right) = 0 \quad (12)$$

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial b} = \frac{an}{\hat{b}} - \sum_{i=1}^n x_i + \frac{\partial \left[\hat{\theta} e^{-\hat{b}x_n} \left(\sum_{i=0}^{a-1} \frac{(\hat{b}x_n)^i}{i!} \right) \right]}{\partial b} = 0 \quad (13)$$

In this study, we consider a case where the shape parameter(a) is 2. Therefore, the Eq.12 and Eq. 13 is derived as follows.

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}x_n} (1 + \hat{b}x_n) = 0 \quad (14)$$

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

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and \hat{b}_{MLE} can be calculated using the bisection method.

2.3 Rayleigh distribution : NHPP model

The Rayleigh distribution is the lifetime distribution of the Weibull family widely used in the reliability field. The probability density function and the cumulative distribution function considering the shape parameter(α) are as follows.

$$f(t) = \frac{t^{\alpha-1}}{\beta^2} e^{-\frac{t^\alpha}{2\beta^2}} \quad (16)$$

$$F(t) = 1 - e^{-\frac{t^\alpha}{2\beta^2}} \quad (17)$$

Note. $\beta > 0, t \in [0, \infty]$.

In order to simplify Eq. 16 and Eq. 17, if substitution by the equation $\frac{1}{2\beta^2} = b$ is as follows.

$$f(t) = 2bt^{\alpha-1} e^{-bt^\alpha} \quad (18)$$

$$F(t) = 1 - e^{-bt^\alpha} \quad (19)$$

Note. $b > 0, t \in [0, \infty]$.

In the Weibull distribution such as Eq. 18 and Eq. 19, the Rayleigh distribution is obtained when the shape parameter(α) is 2. Therefore, the intensity function and the mean value function of the finite fault NHPP Rayleigh model are as follows [10].

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

$$m(t|\theta, b) = \theta F(t) = \theta(1 - e^{-bt^2}) \quad (21)$$

Note. $\theta > 0, b > 0$.

The log-likelihood function to maximum likelihood estimation by using Eq. 20 and Eq. 21 is derived as follows.

$$\ln L_{NHPP}(\theta|\underline{x}) = n \ln 2 + n \ln \theta + n \ln b + \sum_{i=1}^n \ln x_i - b \sum_{i=1}^n x_i^2 - \theta \left(1 - e^{-bx_n^2}\right) \quad (22)$$

Note. $\underline{x} = (0 \leq x_1 \leq x_2 \leq \dots \leq x_n)$, Θ is parameter space.

When the Eq. 22 is partially differentiated concerning θ and b , the result is as follows.

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + \exp(-\hat{b}x_n^2) = 0 \quad (23)$$

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial b} = \frac{n}{b} - \sum_{i=1}^n x_i^2 - \hat{\theta} x_n^2 \exp(-\hat{b}x_n^2) = 0 \quad (24)$$

Note. $x = (x_1, x_2, x_3 \dots x_n)$.

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and \hat{b}_{MLE} can be calculated using the bisection method.

2.4 Software Development Model

The software development cost model using the mean function $m(t)$ of the proposed NHPP model consists of the sum of the costs for each component as follows [11].

$$E_t = E_1 + E_2 + E_3 + E_4 \quad (25)$$

$$= E_1 + C_2 \times t + C_3 \times m(t) + C_4 \times [m(t + t') - m(t)]$$

Note that E_t is the total cost of software development.

① E_1 stands for initial software development costs, and is considered a constant.

② E_2 is the testing cost per unit time, and the actual cost per unit time is different for each industry field. E_2 is expressed by the following Eq. 26.

$$E_2 = C_2 \times t \quad (26)$$

Note that C_2 is the testing cost per unit time, t is the testing time point.

③ E_3 represents the cost of removing a defect by detecting a basic defect, and is expressed by the following Eq. 27.

$$E_3 = C_3 \times m(t) \quad (27)$$

Note that C_3 is the cost of removing one defect found in the testing process, and the mean value function $m(t)$ is the expected value of the defect that can be detected at time t .

④ E_4 represents the cost of eliminating all remaining defects in the software operating system, and is expressed by the following Eq. 28.

$$E_4 = C_4 \times [m(t + t') - m(t)] \quad (28)$$

Note that C_4 is the defect correction cost discovered by the operator at the software operation stage after the software is released, and t' is the time when the software can be operated normally after software release. In reality, C_4 has a higher cost than C_2 and C_3 . Therefore, this study is applied to the realistic supposition that C_4 is higher than C_2 and C_3 .

Therefore, the optimal software release time for the software development cost can be derived as follows.

$$\frac{\partial E_t}{\partial t} = E' = (E_1 + E_2 + E_3 + E_4)' = 0 \quad (29)$$

In other words, it can be seen that the optimal release time is the time point at which the lowest development cost is satisfied.

3. SOFTWARE FAILURE TIME ANALYSIS AND SOLUTIONS

Let compare and analyze the attributes of the proposed development cost models using the software failure time data [12] as shown in Table 1. This software failure is the data that was occurred 30 times in 738.68 unit time.

Table 1. Software Failure time data

Failure number	Failure time (hours)	Failure interval-time	Failure time (hours) × 10 ⁻²
1	30.02	30.02	0.30
2	31.46	1.44	0.31
3	53.93	22.47	0.53
4	55.29	1.36	0.55
5	58.72	3.43	0.58
6	71.92	13.20	0.71
7	77.07	5.15	0.77
8	80.90	3.83	0.80
9	101.90	21.00	1.01
10	114.87	12.97	1.14
11	115.34	0.47	1.15
12	121.57	6.23	1.21
13	124.97	3.40	1.24

14	134.07	9.10	1.34
15	136.25	2.18	1.36
16	151.78	15.53	1.51
17	177.50	25.72	1.77
18	180.29	2.79	1.80
19	182.21	1.92	1.82
20	186.34	4.13	1.86
21	256.81	70.47	2.56
22	273.88	17.07	2.73
23	277.87	3.99	2.77
24	453.93	176.06	4.53
25	535.00	81.07	5.35
26	537.27	2.27	5.37
27	552.90	15.63	5.52
28	673.68	120.78	6.73
29	704.49	30.81	7.04
30	738.68	34.19	7.38

Laplace trend test was used to verify the reliability of the software failure time data as shown in Figure 1 [13].

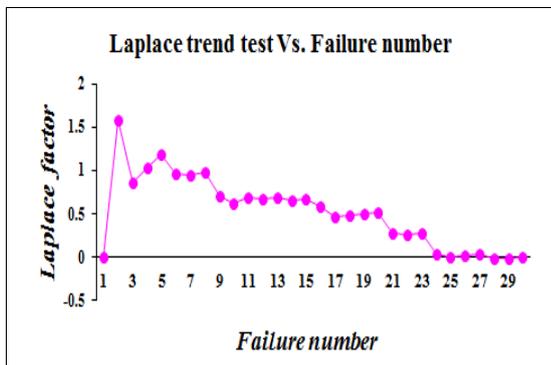


Fig. 1. Estimation Results of Laplace Trend Test

The estimated value of the Laplace factor was distributed between 0 and 2, as shown in Figure 1. Therefore, it is possible to apply this data because there is no extreme value. The maximum likelihood estimation (MLE) was used to perform parameter estimation. And, numerical conversion data (Failure time [hours] $\times 10^{-2}$) to facilitate the parameter estimation was used, and the results are shown in Table 2 [14].

Table 2. Parameter Estimation of Each Model

Model	MLE	
Goel-Okumoto	$\hat{\theta} = 33.4092$	$\hat{b} = 0.3090$
Rayleigh	$\hat{\theta} = 24.0116$	$\hat{b} = 0.3707$
Erlang	$\hat{\theta} = 30.5978$	$\hat{b} = 0.0079$

Explanatory notes. MLE : Maximum Likelihood Estimation.

In this study, we assumed the cost of software development as [Supposition 1] ~ [Supposition 4] to simulate the same as the actual software development environment. To do this, we will analyze and predict software development cost and release time by changing each cost component (C_2, C_3, C_4) that constitutes the total software development cost (E_t).

[Supposition 1 : Basic conditions]

$$E_1 = 20\$, C_2 = 5\$, C_3 = 1.5\$, C_4 = 10\$, t' = 200 \quad (30)$$

The result of the cost curve using [Supposition 1] is as shown in Figure 2. In this figure, the transition of the development cost curve shows a constant pattern for a short period of time after showing a decreasing pattern in the initial stage, but it shows an increasing pattern as the release time passes. The reason is that in the process of eliminating defects during the initial stage, the development cost is decreased because the number of defects inherent in the software is reduced. But the development cost is increased because the probability of finding the remaining defects during the latter stage is gradually lowered. As a result, the pattern of the development cost curve gradually increases as the release time passes.

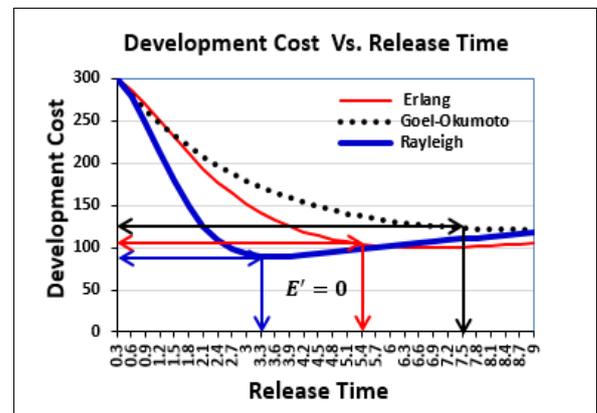


Fig. 2. The development cost curve applied to the condition of [Supposition 1]

As shown in Figure 2, although all of the proposed models show a similar pattern, the Rayleigh model is the best efficient because it has lower development cost and faster release time than other models (Erlang and Go-Okumoto basic).

[Supposition 2 : Assumed that the cost C_2 is increased in supposition 1]

$$E_1 = 20\$, C_2 = 10\$, C_3 = 1.5\$, C_4 = 10\$, t' = 200 \quad (31)$$

[Supposition 2] is a case where the software testing cost (C_2) per unit time is doubled compared with [supposition 1].

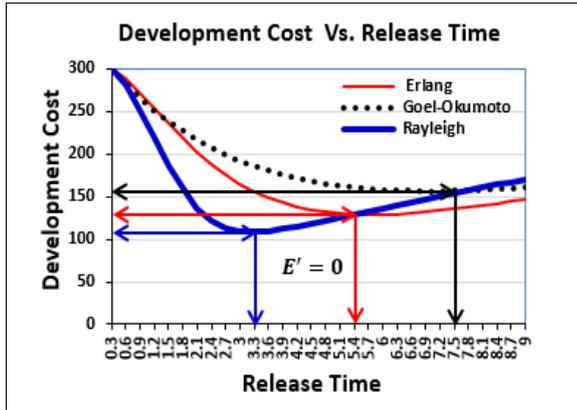


Fig. 3. The development cost curve applied to the condition of [Supposition 2]

As shown in Figure 3, the development cost has increased, but the release time has not changed. Therefore, in this case, it can be seen that fast and precise testing is required so that the testing cost per unit time does not increase before the software is released. Also, the Rayleigh model is relatively efficient because it has lower development costs and faster release time than other models.

[Supposition 3 : Assumed that the cost C_3 is increased in supposition 1]

$$E_1 = 20\$, C_2 = 5\$, C_3 = 3\$, C_4 = 10\$, t' = 200 \quad (32)$$

[Supposition 3] is a case where the cost(C_3) of removing one defect found in the software testing process is doubled compared to [Supposition 1]. As shown in Figure 4, the development cost has increased, but the release time has not changed. Therefore, in this case, as many defects as possible should be removed at once so that the cost of removing one defect in the software testing step is not increased.

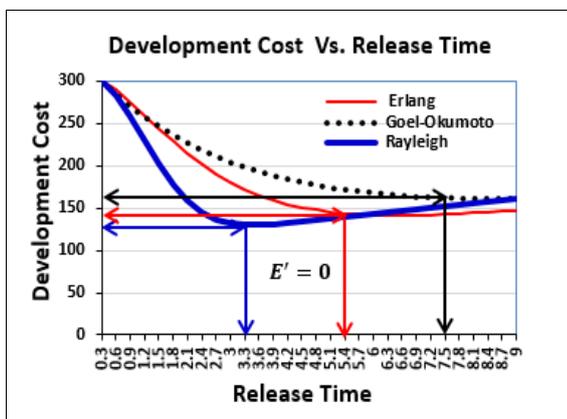


Fig. 4. The development cost curve applied to the condition of [Supposition 3]

As shown in the simulation result, the Rayleigh model is relatively efficient because it has lower development costs and faster release time than other models.

[Supposition 4 : Assumed that the cost C_4 is increased in supposition 1]

$$E_1 = 20\$, C_2 = 5\$, C_3 = 1.5\$, C_4 = 20\$, t' = 200 \quad (33)$$

[Supposition 4] is a case where the cost(C_4) of correcting defects discovered by the software operator during the operation stage after releasing the software is doubled compared to [Supposition 1].

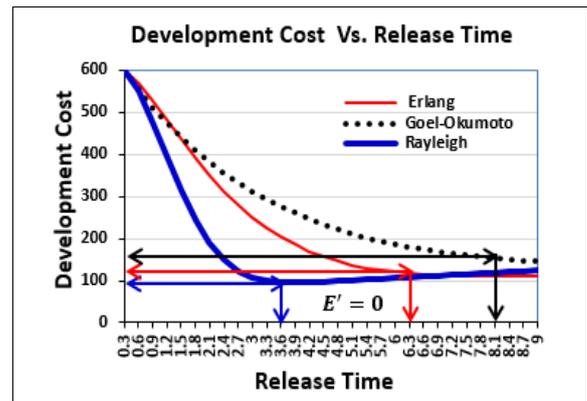


Fig. 5. The development cost curve applied to the condition of [Supposition 4]

As shown in Figure 5, it can be seen that the release time is delayed with increasing development costs. Therefore, in this case, we must eliminate all possible defects at the testing stage rather than the operational stage to reduce all defects before releasing the software. Also, the Rayleigh model is the best efficient model because it has lower development costs and faster release time than other models.

In conclusion, by using this prior information, we can predict the optimal software release time together with the development cost attributes.

IV. CONCLUSION

If software development costs can be quantitatively modeled together with release time during the software development process, the attributes of development costs can be efficiently analyzed and predicted. Therefore, this study was analyzed and predicted software development cost along with software release time through the proposed NHPP reliability models.

The results of this study can be summarized as follows. First, under basic conditions (Supposition 1), the software development cost curve shows a constant pattern for a short

time after a significant decrease in the initial stage but shows a pattern of increasing again in the latter stage as the release time passes. The reason is that the number of remaining defects is gradually reduced in the course of removing defects, so the probability of finding residual defects is getting lower. For this reason, it can be seen that the cost eventually increases.

Second, before software release, if the testing cost (C_2) per unit time and the cost (C_3) of removing one defect found in the testing process increases, the development cost has increased as well, but the release time has not changed. But, after software release, if the defect correction cost (C_4) discovered by the operator increases, the development cost increases and the release time is also delayed.

Third, as a result of a comprehensive analysis of the proposed model used in this study, the Rayleigh model is relatively most efficient because it has low software development cost and fast release time compared to other models.

Using the results of this study, it is possible to provide software developers and operators with the necessary prior information for predicting the most economical software development costs and the optimal release time.

Besides, further studies will be needed to find out the optimal software development cost model through analysis with other models having the same type of failure time distribution.

Acknowledgements

Funding for this paper was provided by Namseoul University.

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 model -ing : 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] Pham H, Zhang X., "NHPP software reliability and cost models with testing coverage," *Eur.J. Oper. Res*, 145, pp. 445-454, 2003.
- [5] H. C. Kim, "A Comparison Analysis about Reliability Features of Software Reliability Model Using Burr- XII and Type-2 Gumbel Lifetime Distribution," *International Journal of Engineering Research and Technology*, Vol. 12, No. 1, pp. 73-78, 2019.
- [6] H. C. Kim, K. S. Kim "Software Development Cost Model based on NHPP Gompertz Distribution," *Indian Journal of Science and Technology*, Vol. 9, No. 20, pp. 1-6, 2016.
- [7] Tae-Jin Yang, "A Study on the Reliability Performance Analysis of Finite Failure NHPP Software Reliability Model Based on Weibull Life Distribution," *International Journal of Engineering Research and Technology*, Vol. 12, No. 11, pp. 1890-1896, 2019.
- [8] 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.
- [9] V. K. Rohatgi, *Statistical inference*, John Wiley & Sons, INCL, New York, pp. 398-416. 1984.
- [10] 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.
- [11] 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.
- [12] R. Satya Prasad, K. R. H. Rao and R. R. L. Kantham, "Software Reliability Measuring using Modified Maximum Like-lihood Estimation and SPC," *International Journal of Computer Applications (0975-8887)*, Vol. 21, No. 7, pp. 1-5, 2011.
- [13] Tae-Jin Yang, J.G. Park "A Comparative Study of the Software NHPP Based on Weibull Extension Distribution and Flexible Weibull Extension Distribution," *International Journal of Soft Computing*, Vol. 11, No. 4, pp. 276~281, 2016.
- [14] Tae-Jin Yang, "The Analysis and Predict of Software Failure Time Based on Nonlinear Regression," *Journal of Engineering and Applied Sciences*, Vol. 13, Issue 12, pp. 4376-4380, 2018.