International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 11 (2019), pp. 2036-2040
© International Research Publication House. http://www.irphouse.com

A Comparative Study on the Cost of Software Development Model Based on Burr–Hatke- Exponential Distribution

Hee-Cheul Kim[1]
[1] *Associate Professor, Department of Industrial & Management Engineering,*
Namseoul University, South Korea
ORCID: 0000-0002-4786-5672

Abstract

In this research, the software cost model considering Burr–Hatke exponential distribution, by applying software failure time data, was analysed. Software failure model was used NHPP and the parameter estimation was applied maximum likelihood estimation. The results of this study can be summarized as follows. If cost per unit time and cost of eliminating one defect in the testing process are the same price, it can be seen that software release period is fast, but the cost is higher than other cases, which is uneconomical circumstance. Also, if cost per unit time is larger price than cost of eliminating one defect in the testing process, it can be seen that software release period is later than the same price, but the cost is lower than other belongings, which is economical circumstance. In the case of price cost of eliminating one defect in the testing process less than cost per unit time, regard as the ineffective model in terms of software release period and cost was found. Before the optimal release time, as the value of defect correction costs that are observed by the user during the software operation phase after the software is released increases, the cost increases, but after the optimal release time for all models, the cost is nearly the same. Before the optimal release time, as the time to operate and maintain software after launching the software system increases, the cost increases, but after the optimal release time for all models, the cost is almost the same. Through this study, software operators are considered to be helpful to identify the software development costs.

Keywords: Software Cost Model, NHPP, Burr–Hatke-Exponential Distribution, Finite Failure Model, NHPP

I. INTRODUCTION

Software reliability is a likelihood that can be operated lacking a malfunction born over a sector of time at conservation settings. Therefore, software reliability is an important factor affecting system reliability and in terms of hardware reliability, design attributes have different properties. Therefore, it may cause enormous losses outstanding to a failure of the computer organization, software breakdown in our society. Therefore, software reliability in the software growth progression is a major underlying problem. This environment needs to analyse the operator software requirements. So as to efficiently achieve the costs in terms of software difficult, a cost and reliability of the testing of the software is the efficient development process if the trend was estimated in advance. Therefore, reliability, cost, and software considering the estimated time for the release time, the development process are indispensable process. Therefore, in terms of the error search process, the software reliability mode[1, 2] using the non-homogeneous Poisson process (NHPP) was regard as excellent model and if a new fault occurs, it is immediately was removed and have the presumption that new defects does not occurred from the debugging process. Huang [3] was presented the effective integration of software reliability prediction technology.In another aspect, the S-shape model can be analysed from the knowledge process in the software manager may be utilized in software failure discovery implement [4]. Kim [5] also studied the reliability characteristics of the Burr-XII and type-2 Gumbel distributions for the lifetime distribution. Also, was studied the characteristics of the software development cost with the inverse exponential distribution as the life distribution [6]. And software development cost model based on NHPP Gompertz distribution was studied [7].

In this working out, software failure time data was applied to NHPP life distribution in the reliability model analysis to compare the cost of a software reliability model considering Burr–Hatke-exponential distribution.

II. NHPP SOFTWARE RELIABILITY MODEL USING BURR-HATKE EXPONENTIAL DISTRIBUTION

The hazard function of basic exponential distribution, which represents the rate of failure per unit time, has a constant independent of the failure time. But Burr-Hatke-exponential distribution in which these patterns of hazard functions exhibit an increasing function or a decreasing function [8]. The probability density function and cumulative distribution function of this distribution are defined as follows. The probability density function and cumulative distribution function of this distribution are defined as follows.

$$F(t \mid \lambda) = 1 - \frac{e^{-\lambda t}}{1+\lambda t}, \quad f(t \mid \lambda) = \lambda e^{-\lambda t} \frac{2+\lambda t}{(1+\lambda t)^2} \tag{1}$$

Note that $t \in (0, \infty]$ and $\lambda > 0$ are the shape parameter. In finite failure NHPP model, θ was specified the expected value of faults that would be discovered observing time $(0, t]$. Thus the intensity function and the average value function are known as follows [5].

$$\lambda(t \mid \theta, \lambda) = \theta f(t) = \theta \lambda e^{-\lambda t} \frac{2+\lambda t}{(1+\lambda t)^2} \tag{2}$$

$$m(t\,|\,\theta,\lambda)=\theta F(t)=\theta\left[1-\frac{e^{-\lambda t}}{1+\lambda t}\right] \qquad (3)$$

In equation (2) and equation (3), time t and x_n are replaced with the last failure time point, the likelihood function is known as follows [9].

$$m(t)=\theta\,F(t) \qquad (4)$$

Note that $\underline{x}=(x_1\le x_2\le x_3\le\ldots\le x_n)$,

$\Theta=\{\theta,\beta\}$ Specifies parameter space.

The log-likelihood function by means of the equation (4) can be detailed ensuing relation [10, 11].

$$m(t)=\theta\,F(t) \qquad (5)$$

The estimator $\hat{\theta}_{MLE}$ and $\hat{\lambda}_{MLE}$ must be assessed the following structure for the maximum likelihood estimation about all parameter by means of the equation (5).

$$\frac{\partial \ln L_{NHPP}(\Theta\,|\,\underline{x})}{\partial\theta}=\frac{n}{\theta}-\left(1-\frac{e^{-\lambda x_n}}{1+\lambda x_n}\right)=0 \qquad (6)$$

$$\begin{aligned}\frac{\partial \ln L_{NHPP}(\Theta\,|\,\underline{x})}{\partial\lambda}&=\frac{n}{\lambda}-\sum_{i=1}^{n}x_n+\sum_{i=1}^{n}\frac{x_i}{2+\lambda x_i}\\&-2\sum_{i=1}^{n}\frac{x_i}{1+\lambda x_i}-\theta x_n e^{-\lambda x_n}\frac{(2+\lambda x_i)}{(1+\lambda x_i)^2}=0\end{aligned} \qquad (7)$$

III. SOFTWARE DEVELOPMENT COST MODEL

The estimated whole charge of software development was achieved as next special features [12, 13].

$$\begin{aligned}E&=E_1+E_2+E_3+E_4\\&=E_1+C_2\times t+C_3\times m(t)+C_4\times\left[m(t+t')-m(t)\right]\end{aligned} \qquad (8)$$

The based on equation (8), E mean estimated total cost of software development. Also, E_1 denotes initial software plan and software growth costs (data to analyse, the number of software development professionals, CPU time, etc.). The software testing costs per unit time (constant) is E_2 . $E_2=C_2\times t$, using C_2 (means cost per unit time) and t (represents the failure period). Also, E_3 denotes cost of the removal of a defect of the activity, such as to detect the fault and remove the defective. $E_3=C_3\times m(t)$, using C_3 (represents cost of eliminating one defect in the testing process) and $m(t)$ (indicates the predictable amount of errors noticed at period t). And E_4 denotes cost of the fixative failure, in the working stage from similarly connected to the software dependability forming; it must satisfy $E_4=C_4\times\left[m(t+t')-m(t)\right]$ in this situation. Thus C_4 represents defect correction costs that are observed by the user during the software operation phase after the software is released and t' represents time to operate and maintain software after launching the software system. From special feature, a cost of C_4 is larger than cost of C_2 & C_3 . In decision, optimum software release period t was expressed next estimating equation.

$$\begin{aligned}E'&=(E_1+E_2+E_3+E_4)'\\&=E_1+C_2\times t+C_3\times m'(t)+C_4\times\left[m'(t+t')-m'(t)\right]=0\end{aligned} \qquad (9)$$

III. SOFTWARE DEVELOPMENT COST ANALYSIS

Table 1. Failure time data

Failure Number	*Failure Time (hours)*	*Failure Time* $\times 10^{-2}$	*Failure Number*	*Failure Time (hours)*	*Failure Time* $\times 10^{-2}$
1	30.02	0.3002	16	151.78	1.5178
2	31.46	0.3146	17	177.5	1.775
3	53.93	0.5393	18	180.29	1.8029
4	55.29	0.5529	19	182.21	1.8221
5	58.72	0.5872	20	186.34	1.8634
6	71.92	0.7192	21	256.81	2.5681
7	77.07	0.7707	22	273.88	2.7388
8	80.9	0.809	23	277.87	2.7787
9	101.9	1.019	24	453.93	4.5393
10	114.87	1.1487	25	535	5.35
11	115.34	1.1534	26	537.27	5.3727
12	121.57	1.2157	27	552.9	5.529
13	124.97	1.2497	**28**	**673.68**	**6.7368**
14	134.07	1.3407	**29**	**704.49**	**7.0449**
15	136.25	1.3625	**30**	**738.68**	**7.3868**

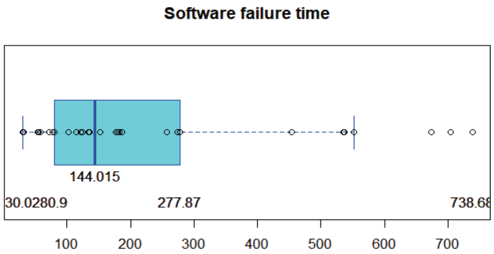

Fig. 1. Box plot test

In this section, the reliability structures of the software reliability model were studied using the software failure time data [14]. The failure time data is revealed in Table 1. Furthermore, a trend test should be headed in order to assure reliability of data. In this study, the trend analysis was used was the Box-plot test [15]. Therefore, in the outcome of Figure 1, three data (28th, 29th, 30th) were excluded from the parameter estimation due to the occurrence of an abnormal value (extreme value). The maximum likelihood estimation so as to the parameter estimation calculation was applied. So as to enable the parameter estimation, in this research, a mathematical translation data ($\textit{Failure Time} \times 10^{-2}$) was used and calculation method of nonlinear equations was used bisection method. These controls solve the root exactly, since the initial values were specified 0.0001 and 1.000 and the tolerance value for the measurement of interval (10^{-5}) were specified, with an accomplished replication of 100 times using *C*-language checking satisfactory convergent. The results of the maximum likelihood estimation were estimated

$$\hat{\theta}_{MLE} = 29.0996\,,\ \hat{\lambda}_{MLE} = 0.2991\,. \tag{10}$$

In this study, was analysed the cost curve assuming that the defect repair cost observed by the user is larger than the cost of removing one defect and the testing cost per unit time. Therefore, was used the cost curve that reflects the following assumptions.

$$< \textit{Assumption} > \\ E_1 = 5000\,\$, C_2 = 3\,\$, C_3 = 2\,\$, C_4 = 5\,\$, t' = 5(hours) \tag{11}$$

The estimated cost curve of assumptions summarized in Figure 2.

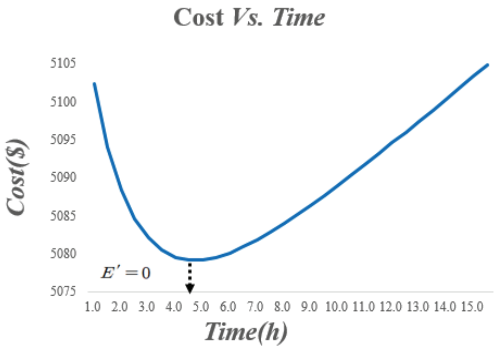

Fig. 2. Cost curve based on the situation of <Assumption>

The cost curve of the model presented that in the early stages while showing a tendency to decrease gradually increase after a certain time (release time) to failure as shown in Figure 2. The reason is that the number of defects remaining in the software structure has been reduced more and more in the course of removal of the defectives. Thus, the observed probability from remaining defects is lowered. In the initial step of difficult, there are unmoving numerous errors in software which are simply noticed. The cost to remove an error in the step is far lesser than that of eliminating an error in the procedure period. Accordingly, the price of the software decreases throughout the procedure of faults correcting. But from the later step the number of remaining faults in software is previously less, and in this testing stage the time of noticing a fault is very long and the cost of removing a fault become advanced than that in task step, therefore the cost curve increases continuously with time. From the tendency of the cost curve, can be foreseeable the optimal software release period and it is also the most faithful phenomenon. The most circumstances are reliable in a development from the real software development [13]. The other assumptions are the same, the cost curve for the relationship between cost per unit time (C_2) and cost of eliminating one defect in the testing (C_3) is shown in Figure 3. In this figure, if C_2 and C_3 are the same price ($C_2 = C_3 = 2.5\,\$$), it can be seen that software release period is fast, but the cost is higher than other cases, which is uneconomical circumstance. Also, if C_2 is larger price than C_3($C_2 = 3\,\$, C_3 = 2\,\$$), it can be seen that software release period is later than the same price, but the cost is lower than other cases, which is economical circumstance. In the case of price C_3 less than C_2, regard as the ineffective model in terms of software release period and cost was found.

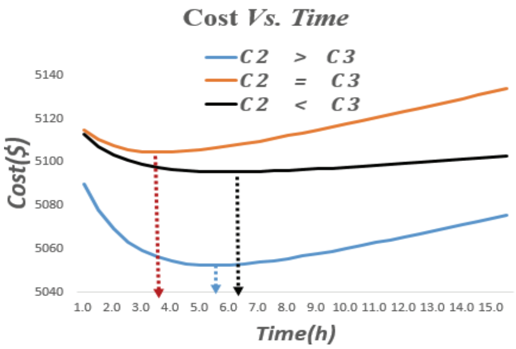

Fig. 3. Cost curve for the relationship in the testing process

The other assumptions are the same, the cost curve for the relationship between defect correction costs that are observed by the user during the software operation phase after the software is released (C_4) is shown in Figure 4. In this figure, before the optimal release time, as the value of C_4 increases, the cost increases, but after the optimal release time for all models, the cost is almost the same.

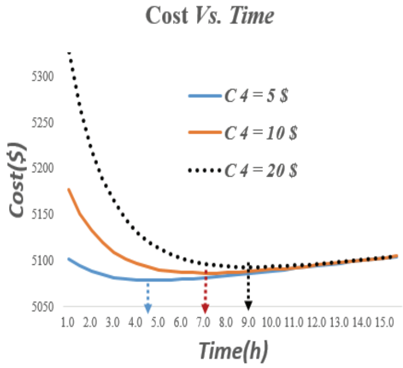

Fig. 4. Cost curve for the relationship after released time

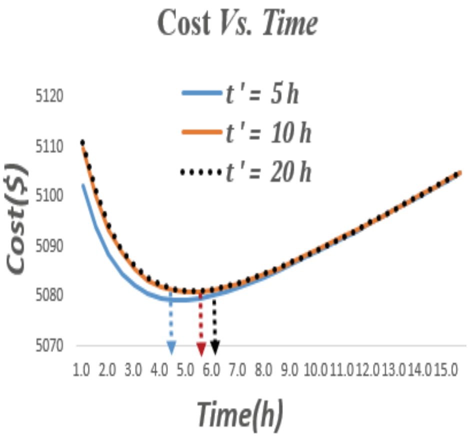

Fig. 5. Cost curve for software to maintain

The other assumptions are the same, the cost curve for the relationship t' represents time to operate and maintain software after launching the software system in Figure 5. In this figure, before the optimal release time, as the time of t' increases, the cost increases, but after the optimal release time for all models, the cost is almost the same.

IV. CONCLUSION

In this research, the cost of the optimal release period can be predicted. Therefore, more efficient models to reduce the cost of testing, it should ensure that profits can increase the release software. Software cost model using finite failure NHPP model based on Burr–Hatke exponential distribution used

Therefore, a more efficient model should reduce testing costs and allow software to increase total cost benefits. The cost curves of Burr-Hatke-exponential distribution model used in this study were compared and analyzed. The results of this study can be summarized as follows.

First, if cost per unit time and cost of eliminating one defect in the testing process are the same price, it can be seen that software release period is fast, but the cost is higher than other cases, which is uneconomical circumstance.

Second, also, if cost per unit time is larger price than cost of eliminating one defect in the testing process, it can be seen that software release period is later than the same price, but the cost is lower than other cases, which is economical circumstance. In the case of price cost of eliminating one defect in the testing process less than cost per unit time, regard as the ineffective model in terms of software release period and cost was found.

Third, before the optimal release time, as the value of defect correction costs that are observed by the user during the software operation phase after the software is released increases, the cost increases, but after the optimal release time for all models, the cost is almost the same.

Fourth, before the optimal release time, as the time to operate and maintain software after launching the software system increases, the cost increases, but after the optimal release time for all models, the cost is almost the same.

Through this study, software operators can identify the cost of software development by using the features of life distribution to identify the circumstance of software development.

ACKNOWLEDGEMENTS

Funding for this paper was provided by Namseoul University

REFERENCES

[1] Gokhale, S.S., and Trivedi, K.S. A time/structure based software reliability model. Annals of Software Engineering 8. 1999; 5-12.

[2] Goel, A. L, Okumoto K, Time-dependent fault detection rate model for software and other performance measures. IEEE Trans. Reliability. 1978; 28:206-211.

[3] Huang, C.Y., Performance analysis of software reliability growth models with testing-effort and change-point. J. Syst. Software 76. 2005; 181-194.

[4] 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. 2008:1410-1421.

[5] Hee-Cheul Kim. The 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. 2019; 12(1):73-78.

[6] 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. 2017; 15(4): 1-7.

[7] Kim H-C, Kim K-S, Software Development Cost Model based on NHPP Gompertz Distribution. Indian Journal of Science and Technology.2016; 9(20):1-6.

[8] Yadav, A. S. Altun, E., Yousof, H. M. Burr–Hatke Exponential Distribution: A Decreasing Failure Rate Model, Statistical Inference and Applications. Annals of Data Science. 2019; 1-20.

[9] Al. Turk, L. I, Characteristics and Application of the NHPP Log-Logistic Reliability Model. International Journal of Statistics and Probability. 2019; 8(1):44-55.

[10] S. Yamada, S. Osaki, Software reliability growth modelling: models and applications. IEEE Transactions on Software Engineering. 1985; 11(12): 1431-1437.

[11] Kim K-S, Kim H-C, The Performance Analysis of the Software Reliability NHPP Log-linear Model Depend on Viewpoint of the Learning Effects, Indian Journal of Science and Technology. 2016; 9(37):1-5.

[12] Kim, H.C., The Property of Learning effect based on Delayed Software S-Shaped Reliability Model using Finite NHPP Software Cost Model. Indian Journal of Science and Technology, 2015; 8(34): 1-7.

[13] Zhang, Y., and Wu, K., Software Cost Model Considering Reliability and Time of Software in Use. Journal of Convergence Information Technology, 2012; 7(13): 135-142.

[14] Satya Prasad R, Rao KRH, Kantha RRL., Software reliability measuring using modified maximum likelihood estimation and SPC, International Journal of Computer Applications. 2011; 21(7):1–5.

[15] Hee-Cheul Kim, A Comparative Study on Software Reliability Models with Shape Parameter of Type-2 Gumble Life Distribution, International Journal of Soft Computing, 2017; 12(5-6):351-354.