Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COMPARATIVE EVALUATION ON THE ATTRIBUTES OF SOFTWARE DEVELOPMENT COST MODEL WITH EXPONENTIAL AND INVERSE-EXPONENTIAL DISTRIBUTION PROPERTY

HYO JEONG BAE

Professor, Department of Drone and GIS Engineering, Namseoul University, 91 Daehak-ro, Seonghwan-eup, 31020, Seobuk-gu, Cheonan-si, Chungnam, Korea
E-mail: baehj@nsu.ac.kr

ABSTRACT

In this study, attributes of software development cost were evaluated by applying the exponential type distributions (Exponential-exponential, Inverse-exponential) which are utilized in the reliability testing field to the software development model. Also, the proposed distribution models were compared with the Goel-Okumoto basic model to verify cost property, and the optimal development cost model was presented. For this study, a total solution was performed using software failure time data generated during desktop application operation, parameter calculations were solved using the maximum likelihood estimation (MLE) method. As a result, fist, when the testing cost per unit time and the cost of eliminating a single fault detected during the development testing process increase, the development cost increases, but the release time does not change. But, if the fault correction cost detected by the operator during normal system operation increases, the development cost increased along with the delay of the release time. Second, it can be confirmed that the Exponential-exponential distribution model is the most efficient among the proposed models as it has the best performance in terms of development cost and releasing time. Third, if software developers and operators can utilize this analysis information efficiently, they can predict and design a reasonable development process by analyzing the related cost and time attributes.

Keywords: *Goel-Okumoto, Exponential-exponential, Inverse-Exponential, mean value function, Software Development Model, Cost Attributes.*

1. INTRODUCTION

The most important task in the era of software convergence with the rapid emergence of the 4th industrial technology is to develop reliable and high-quality software that can be used without failure in various and complex industrial fields. The most important problem in the process of developing such software is the development cost. Therefore, the problem of developing reliable software at an economical cost becomes the most important research topic for software developers. For this reason, studies on software reliability and software development cost are still being actively conducted. Recently, software developers and researchers are actively researching to find the most economical software development cost together with software reliability that determines software quality [1]. Recently, to analyze and predict the reliability performance of software, a new type of software reliability model using the Non-homogeneous Poisson process (NHPP) has been presented. In particular, the reliability performance using the intensity function of the NHPP model has been mainly analyzed and evaluated to test the reliability of the software [2]. Pham [3] proposed a new distribution function and its application method with the failure function of the NHPP software reliability model. Banga and Bansal, Singh [4] presented a new estimation solution using a hybrid approach to predict software fault detection. Yang [5] presented comparative estimation results on the cost characteristics of the software development model with exponential-type lifetime distribution. Kim [6] analyzed and presented research data on the cost properties of the software development model considering Burr-Hatke-exponential distribution. Also, Yang [7] compared and presented the research

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

results on the cost estimation of the NHPP software development model using the Lindley-type lifetime distribution, which has not been studied so far.

Therefore, in this study, the Weibull family lifetime distribution model, which is frequently utilized in the reliability evaluation test field, will be applied to the software development model to analyze the properties of development costs and present a new optimal cost model.

2. RELATED RESEARCH

2.1 NHPP Software Reliability Model

The NHPP model is a stochastic distribution model in which the number of occurrences N(t) at time t follows a Poisson distribution with parameters. It is primarily useful for modeling predicting the probability that a number of mutually independent events can occur steadily over time (t).

In the NHPP model, N(t) refers to the accumulated number of software flaws detected up to the test time t, and m(t) refers to the expected value at which flaws can occur. Therefore, the NHPP model is as follows.

That is,

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

Note that $n=0,1,2,\cdots\infty$.

The mean value function m(t) and the intensity function λ(t) of the NHPP model are as follows.

$$\mathrm{m(t)}=\int_{0}^{t}\lambda(s)ds \quad (2)$$

$$\frac{dm(t)}{d(t)}=\lambda(t) \quad (3)$$

We will use the NHPP reliability model with software failure time based on the NHPP model to analyze the attributes of software development costs.

The time domain NHPP model is divided into a finite failure in which no more failures occur when repairing a failure and an infinite failure in which failures can continue to occur even when the failure is repaired.
In this study, we will analyze based on the finite failure cases.

In this NHPP model, if f(t) is a probability density function and F(t) is a cumulative distribution function, then the mean value function $m(\mathrm{t})$ and the intensity function $\lambda(t)$ are as follows.

$$\mathrm{m(t}|\theta,\mathrm{b})=\theta F(\mathrm{t}) \quad (4)$$

$$\lambda(\mathrm{t}|\theta,\mathrm{b})=\theta F(\mathrm{t})'=\theta f(t) \quad (5)$$

Therefore, using Equations (5) and (6), the likelihood function of the NHPP model is calculated as follows.

$$L_{NHPP}(\Theta|\underline{x})=\left(\prod_{i=1}^{n}\lambda(x_i)\right)exp[-m(x_n)] \quad (6)$$

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

2.2 Goel-Okumoto Basic NHPP Model

The Goel-Okumoto model is a well-known basic model in the field of software reliability tests and evaluation, and is an exponential type model.

In this Goel-Okumoto basic model, let f(t) be a probability density function and F(t) be a cumulative distribution function. Then, when the residual failure rate at the observation point [0, t] is θ, the average value function m(t) and the intensity function λ(t) are calculated as follows [8].

$$\mathrm{m(t}|\theta,\mathrm{b})=\theta F(\mathrm{t})=\theta(1-e^{-bt}) \quad (7)$$

$$\lambda(\mathrm{t}|\theta,\mathrm{b})=\theta\mathrm{f(t)}=\theta\mathrm{b}e^{-bt} \quad (8)$$

Note that $\theta>0,\ \mathrm{b}>0$.

If Equations (7) and (8) are substituted into Equation (6) to obtain Goel-Okumoto's NHPP model equation, the likelihood function is calculated as follows.

$$L_{NHPP}(\theta,\mathrm{b}|\underline{x})=\left(\prod_{i=1}^{n}\theta be^{-bx_i}\right) exp[-\theta(1-e^{-bx_n})] \quad (9)$$

Note that $\underline{x}=(0\le x_1\le x_2\le\cdots\le x_n)$.

If taking the logarithm of both sides in Equation (9), then the log-likelihood function can be solved as follows.

$$\ln L_{NHPP}(\Theta|\underline{x}) = n\ln\theta + n\ln b - b\sum_{k=1}^{n} x_k - \theta(1 - e^{-bx_n}) \quad (10)$$

Therefore, the estimator $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ satisfying Equations (11) and (12) can be calculated by the binary method.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}x_n} = 0 \quad (11)$$

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

2.3 Exponential-exponential NHPP Model

A specific type of Weibull exponential distribution among software reliability models is the Exponential-exponential distribution.

In this Exponential-exponential model, let f(t) be a probability density function and F(t) be a cumulative distribution function. Then, when the residual failure rate at the observation point [0, t] is θ, the average value function m(t) and the intensity function λ(t) are as follows [9].

$$m(t|\theta,a,b) = \theta[1 - exp(-ae^{bt} + a)] \quad (13)$$

$$\lambda(t|\theta,a,b) = \theta[ab\, exp(bt - ae^{bt} + a)] \quad (14)$$

After substituting Equations (13) and (14) into Equation (6), if taking the logarithm of both sides, then the log-likelihood function can be solved as follows.

$$lnL_{NHPP}(\Theta|\underline{x}) = -\hat{\theta}\left[1 - exp\left(-ae^{\hat{b}x_n} + a\right)\right] + \sum_{i=1}^{n} \ln[\,\hat{\theta}(ab\, exp\left(bx_i - ae^{\hat{b}x_i} + a\right))] \quad (15)$$

Note that $\Theta = \{\theta, a, b\}$ is the parameter space.

If solving the partial derivatives of the parameters θ and b in Equation (15), then the parameters $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ can be solved by the binary method as below.

$$\frac{\partial lnL_{NHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{n}{\hat{\theta}} - [1 - exp\left(-2e^{\hat{b}x_n} + 2\right)] = 0 \quad (16)$$

$$\frac{\partial lnL_{NHPP}(\Theta|\underline{x})}{\partial b} = \frac{n}{\hat{b}} + \sum_{i=1}^{n} x_i - 2\sum_{i=1}^{n} x_i\, e^{bx_i} - 2\hat{\theta}x_n\, exp\left(bx_n - 2x_n e^{\hat{b}x_n} + 2\right) = 0 \quad (17)$$

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

2.4 NHPP Inverse-exponential NHPP Model

The Inverse-exponential distribution is widely applied distribution in reliability tests and the medical field. The Inverse-exponential distribution is obtained when the shape parameter(γ) is 1. Here, the cumulative distribution function F(t) is as follows.

$$F(t) = e^{-(bt)^{-\gamma}} \quad (18)$$

Note that $b > 0$, γ is a shape parameter.

When the shape parameter condition ($\gamma = 1$) are applied in the Inverse-exponential distribution, the probability density function f(t) and the cumulative distribution function F(t) are as follows.

$$f(t) = F(t)' = b^{-1}t^{-2}e^{-(bt)^{-1}} \quad (19)$$

$$F(t) = e^{-(bt)^{-1}} \quad (20)$$

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

Therefore, the mean value function m(t) and the intensity function λ(t) of the Inverse-Exponential distribution are as follows [10].

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

$$\lambda(t|\theta, b) = \theta f(t) = \theta b^{-1}t^{-2}e^{-(bt)^{-1}} \quad (22)$$

After substituting Equations (21) and (22) into Equation (6), if taking the logarithm of both sides, then the log-likelihood function can be solved as follows.

$$\ln L_{NHPP}(\Theta|\underline{x}) = n\ln\theta - n\ln b)$$

$$+2\sum_{i=1}^{n} x_i - \sum_{i=1}^{n}(bx_i)^{-1} - \hat{\theta} e^{-(bx_n)^{-1}} = 0 \qquad (23)$$

If solving the partial derivatives of the parameters θ and b in Equation (23), then the parameters $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ can be calculated by the binary method as below.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - e^{-(\hat{b}x_n)^{-1}} = 0 \qquad (24)$$

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial b} = -\frac{n}{\hat{b}} + \frac{1}{\widehat{b^2}}\sum_{i=1}^{n}\frac{1}{x_i} - \theta \frac{1}{b^2 x_n} e^{-(\hat{b}x_n)^{-1}} = 0 \qquad (25)$$

Note that $x = (x_{1,} x_{2,}, x_3 \cdots x_n)$.

Therefore, the estimator $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ in Equations (24) and (25) can be solved by the binary method.

2.6. Software Development Model using the NHPP Model

When the mean value function m(t) of the proposed NHPP model is applied to the software development cost model, it is composed of the sum of each cost element as follows [11].

$$E_t = E_1 + E_2 + E_3 + E_4 = E_1 + C_2 \times t + C_3 \times m(t) + C_4 \times [m(t+t') - m(t)] \qquad (26)$$

Note that E_t is the total cost of software development.

① E_1 is the initial development cost as a constant.

② E_2 is the testing cost per unit time.

$$E_2 = C_2 \times t \qquad (27)$$

Note that C_2 is the testing cost per unit time.

③ E_3 is the cost of removing one fault.

$$E_3 = C_3 \times m(t) \qquad (28)$$

Note that C_3 refers to the cost of eliminating one fault detected in the development testing stage, and the mean values function m(t) refers to the expected value of software failure.

④ E_4 is the cost of removing all remaining faults in the software system.

$$E_4 = C_4 \times [m(t+t') - m(t)] \qquad (29)$$

Note that C_4 is the fault correction cost found in the software operation stage. and t' is the normal operating time of the software system.

Also, it can be seen that the time point at which the software development cost is the minimum becomes the optimal software releasing time point.

Therefore, the equation of the optimal software releasing time is as follows.

$$\frac{\partial \mathrm{E_t}}{\partial t} = \mathrm{E}' = (\mathrm{E_1} + E_2 + E_3 + E_4)' = 0 \qquad (30)$$

3. THE ANALYSIS OF DEVELOPMENT COST USING SOFTWARE FAILURE TIME

In this paper, the cost properties of the proposed distribution model are compared and analyzed using the software failure time data.

The software failure time data applied in this paper means random faults caused by software design and analysis errors and insufficient testing during the normal system operation of desktop applications.

Table 1 shows the software failure time data used in this study [12]. This software failure time is the data collected for 30 failures during a total of 738.68 testing hours.

Table 1: Software Failure Time Data

Failure Number	Failure Time (hours)	Failure Time (hours) $\times 10^{-2}$	Failure Number	Failure Time (hours)	Failure Time (hours)$\times 10^{-2}$
1	30.02	0.30	16	151.78	1.51
2	31.46	0.31	17	177.50	1.77
3	53.93	0.53	18	180.29	1.80
4	55.29	0.55	19	182.21	1.82
5	58.72	0.58	20	186.34	1.86
6	71.92	0.71	21	256.81	2.56
7	77.07	0.77	22	273.88	2.73
8	80.90	0.80	23	277.87	2.77
9	101.90	1.01	24	453.93	4.53
10	114.87	1.14	25	535.00	5.35
11	115.34	1.15	26	537.27	5.37
12	121.57	1.21	27	552.90	5.52
13	124.97	1.24	28	673.68	6.73
14	134.07	1.34	29	704.49	7.04
15	136.25	1.36	30	738.68	7.38

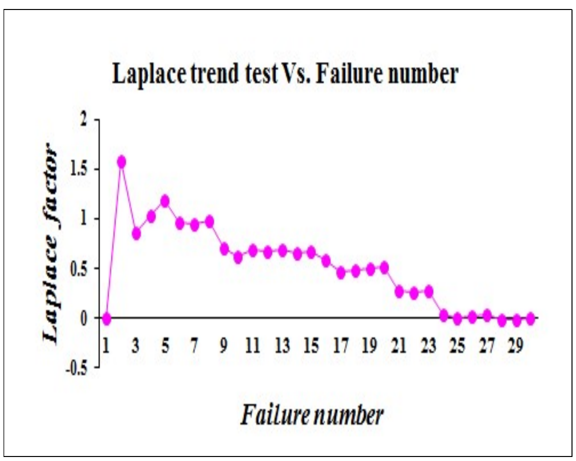

Figure 1: Simulation Results of Laplace Trend Test

In Figure 1, the analysis result of the Laplace trend test existed between 0 and 2. Therefore, since there is no extreme value, this failure data can be utilized [13].

In this study, the parameters (θ, b) were calculated using the maximum likelihood estimation (MLE) method with numerical conversion data to facilitate parameter estimation as in Table 1.

Table 2 represented the parameter estimated results of the proposed models using MLE [14].

Table 2: Parameter Estimates Using the MLE

Type	NHPP Model	MLE (Maximum Likelihood Estimation)	
Basic	Goel-Okumoto	$\hat{\theta} = 29.0332$	$\hat{b} = 0.4809$
Exponential	Exponential-exponential	$\hat{\theta} = 30.6612$	$\hat{b} = 0.1879$
	Inverse-exponential	$\hat{\theta} = 30.3914$	$\hat{b} = 1.6984$

The calculating methods of the mean value function m(t) that determines the cost attributes of the software development model are shown in Table 3 [15]. For this purpose, the calculation method of the mean value function of the proposed NHPP model is also specified.

Also, the estimated values of $m(t)$ are shown in Table 4. After all, these estimated values of Table 4 can be substituted and used as the mean value function to calculate the total software development cost of the proposed model.

Table 3: Calculating Methods Using The m(t).

Type	NHPP Model	$m(t)$ of the Proposed Distribution Model	$m(t)$ of Software Development Cost Model
Basic	Goel-Okumoto	$m(t) = \theta(1 - e^{-bt})$	$E_3 = C_3 \times m(t)$ $E_4 = C_4 \times [m(t + t') - m(t)]$
Exponential	Exponential-exponential	$m(t) = \theta[1 - exp(-ae^{bt} + a)]$	
	Inverse-exponential	$m(t) = \theta e^{-(bt)^{-1}}$	

Table 4: Detailed Estimates of The mean value function m(t).

Failure Number	Failure Time (hours)	Basic Model	Exponential Distribution Model	
		Goel-Okumoto	Exponential-exponential	Inverse-exponential
1	0.3002	3.902911727	3.359735182	4.275220699
2	0.3146	4.076336794	3.515814738	4.676781919
3	0.5393	6.632535295	5.889433139	10.20005987
4	0.5529	6.77856297	6.029323417	10.47769162
5	0.5872	7.142639363	6.380201078	11.15014884
6	0.7192	8.489039854	7.704513116	13.40311193
7	0.7707	8.991594801	8.20991945	14.15676451
8	0.809	9.357351823	8.581653235	14.67815893
9	1.019	11.24735524	10.5567584	17.05338358
10	1.1487	12.3228185	11.72285765	18.20305471
11	1.1534	12.36054516	11.7643388	18.24111475
12	1.2157	12.8526505	12.30904842	18.72460318
13	1.2497	13.11506106	12.60228867	18.97296441
14	1.3407	13.79664656	13.37310457	19.58950527
15	1.3625	13.95554641	13.55472525	19.72763863
16	1.5178	15.04058091	14.81457873	20.61948962
17	1.775	16.66853877	16.77014504	21.81171712
18	1.8029	16.83332882	16.97249334	21.92397045
19	1.8221	16.9454552	17.11063578	21.99954648
20	1.8634	17.18316356	17.40472894	22.15767188
21	2.5681	20.58933018	21.79177735	24.16461133
22	2.7388	21.25479672	22.6807804	24.51239386
23	2.7787	21.40262523	22.87921846	24.58817954
24	4.5393	25.76090808	28.58621714	26.69430825
25	5.35	26.81737454	29.70258394	27.2241806
26	5.3727	26.84143189	29.72471856	27.2368424
27	5.529	27.00013623	29.86603853	27.32135266
28	6.7368	27.89583648	30.47279772	27.84798728
29	7.0449	28.05246443	30.53774434	27.95463395
30	7.3868	28.2011578	30.58621718	28.06298223

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 4 shows the trend of the mean value function (expected value of failure) indicating the ability to estimate the true value. In this simulation, all of the proposed models were found to have an error estimate in predicting the true value, but the Exponential-exponential model showed the smallest error pattern. That is, the Exponential-exponential model is efficient because the estimated error value for the true value is smaller than the Inverse-exponential model.

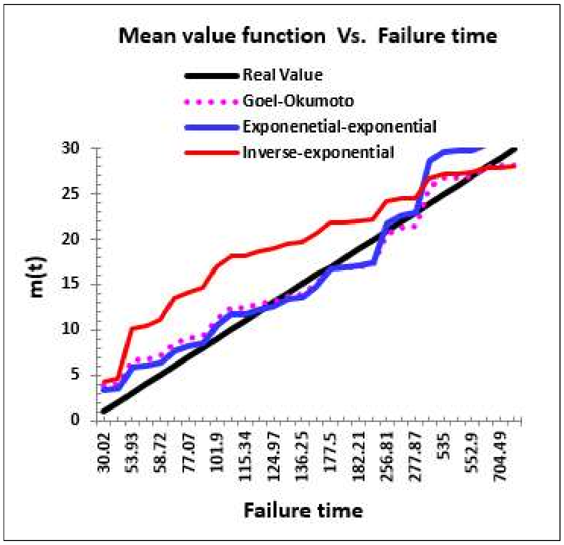

Figure 2: Simulation Result of The Mean Value Function.

In this study, cost conditions in the software development model as in Equation (26) were proposed as [Assumption 1] ~ [Assumption 5]. These cost conditions are presented in consideration of the actual development environment [16].

3.4.1. [Assumption 1: basic conditions]

$$E_1 = 50\$, \quad C_2 = 5\$, \quad C_3 = 1.5\$, \quad C_4 = 10\$$$

$$t' = 100\text{H} \tag{31}$$

If substituting the cost value given in Equation (31) and the estimated value of the mean value function m(t) in Table 4 into the software development cost model equation as in Equation (26), the simulation result is as bellows.

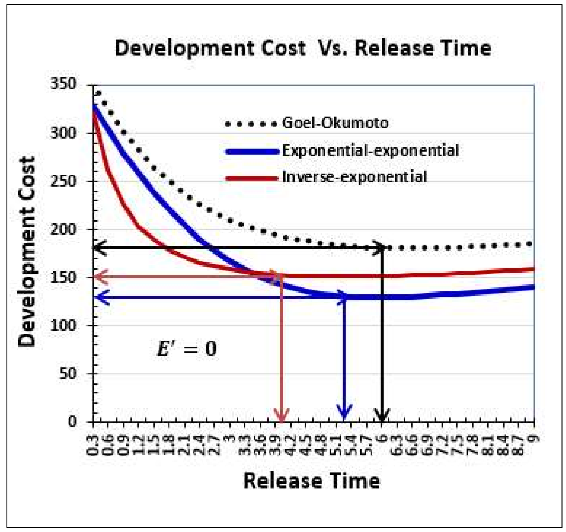

Figure 3: Cost Trend Simulated by The [Assumption 1].

As a result of analyzing Figure 2, it was found that the development cost curve showed a sharp decrease in the beginning, then showed a constant shape for a short period, and the cost gradually increased as the release time passed. For this reason, it was confirmed that the cost increases because the probability of finding a residual failure in the software gradually decreases in the later stage than in the early stage [17].

As shown in Figure 2, the Exponential-exponential model has a slightly delayed release time compared to the Inverse-exponential model, but it is the most efficient among the proposed models because the development cost is the lowest.

3.4.2. [Assumption 2: The C_2 only has doubled increase under Assumption 1]

$$E_1 = 50\$, \quad C_2 = 10\$, \quad C_3 = 1.5\$, \quad C_4 = 10\$$$

$$t' = 100\text{H} \tag{32}$$

[Assumption 2] is the case where only the cost (C_2) is doubled under the same conditions as in [Supposition 1].

If substituting the cost value given in Equation (32) and the estimated value of the mean value function m(t) in Table 4 into the software development cost model equation as in Equation (26), the simulation result is as shown in Figure 3.

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

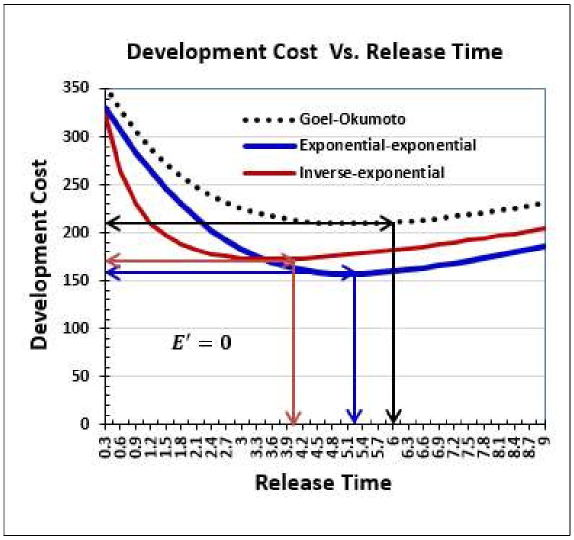

Figure 4: Cost Trend Simulated by The [Assumption 2].

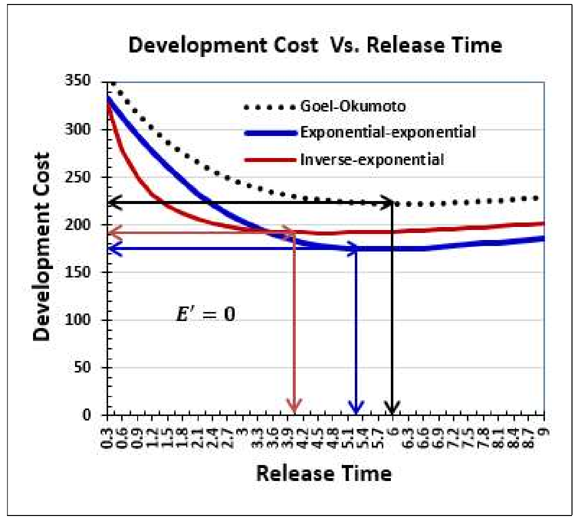

Figure 5: Cost Trend Simulated by The [Assumption 3].

As a result of analyzing Figure 3, it was found that although the development cost increased compared to the [Assumption 1] condition, the release time did not change at all. Therefore, it was confirmed that accurate testing is required so that the cost does not increase.
Therefore, the Exponential-exponential model has a slightly delayed release time compared to the Inverse-exponential model, but it is the most efficient among the proposed models because the development cost is the lowest.

3.4.3. [Assumption 3: The C_3 only has doubled increase under Assumption 1]

$$E_1 = 50\$, \quad C_2 = 5\$, \quad C_3 = 3\$, \quad C_4 = 10\$$$

$$t' = 100\text{H} \tag{33}$$

[Assumption 3] is the case where only the cost (C_3) is doubled under the same conditions as in [Assumption 1].

If substituting the cost value given in Equation (33) and the estimated value of the mean value function m(t) in Table 4 into the software development cost model equation as in Equation (26), the simulation result is as shown in Figure 4. Analysis of Figure 4 showed that the development cost increased but the release time did not change at all.

In this case, it was confirmed that as many faults as possible should be eliminated at once so that the cost of removing one fault does not increase during the testing stage of the development process.

Also, the Exponential-exponential model showed the best performance among the proposed models because the release time was slightly delayed compared to the Inverse-exponential model but the development cost was the lowest.

3.4.4. [Assumption 4: The C_4 only has doubled increase under Assumption 1]

$$E_1 = 50\$, \quad C_2 = 5\$, \quad C_3 = 1.5\$, \quad C_4 = 20\$$$

$$t' = 100\text{H} \tag{34}$$

[Assumption 4] is the case where only the cost (C_4) is doubled under the same conditions as in [Assumption 1].

If substituting the cost value given in Equation (34) and the estimated value of the mean value function m(t) in Table 4 into the software development cost model equation as in Equation (26), the simulation result is as shown in Figure 5.

Analyzing Figure 5, it can be seen that the release time is also delayed as the development cost increases [18].

Thus, in this case, it can be seen that to reduce possible faults before releasing the software, it is necessary to eliminate as many faults as possible in the testing stage rather than the actual operation stage.

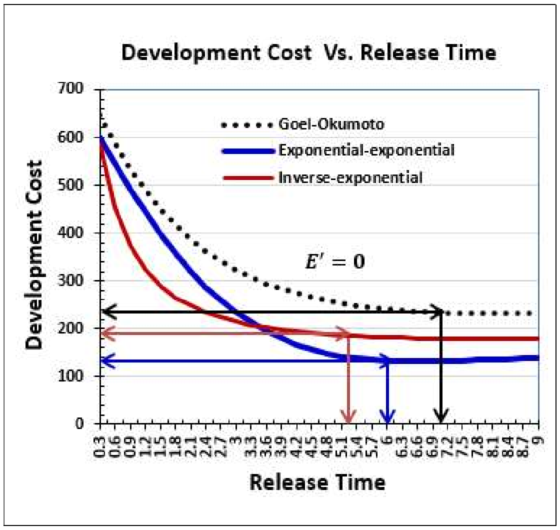

Figure 6: Cost Trend Simulated by The [Assumption 4].

As can be seen from the simulation results, the Exponential-exponential model was relatively efficient compared to the proposed other model because the development cost was low and the release time was fast.

4. CONCLUSION

If the developer can analyze and evaluate the characteristics of development cost in detail from the testing process, which is the stage before releasing the software, the attributes of the total development cost can be reasonably predicted.
Therefore, in this study, in order to predict the cost design of software developers, the property of software development cost was analyzed and evaluated using the exponential distribution model, which is widely applied in the field of reliability testing.

The results of this study are as follows.
First, under the basic condition of [Assumption 1], development cost decreased in the initial stage but increased again in the later stage. The reason is that the probability of finding residual faults in the later stages is gradually decreasing.

Second, in the testing process, if the test cost per unit time (C_2) and the cost of removing one fault (C_3) increased, the development cost increased but the release time did not change at all. But, after the software system was released, if the fault correction cost (C_4) found by the operator increased, the development cost increased and the release time was also delayed.

Third, when the simulation results are analyzed comprehensively, the Exponential-exponential model showed the best performance among the proposed models because the release time was slightly delayed but the development cost was the lowest.

In conclusion, if software developers can efficiently use this research data, it is possible to predict the optimal release time together with the trend of development costs. Also, after exploring more diverse distribution models, additional research is needed to find the optimal cost model by applying the software failure time data applied in this paper to the searched development cost model.

ACKNOWLEDGEMENTS

Funding for this paper was provided by Namseoul University.

REFERENCES

[1] S. S. Gokhale, and K.S. Trivedi, "A time/ structure-based software reliability model", *Annals of Software Engineering,* Vol. 8, No. 1, 1999. pp. 85-121.
[2] K. Y. Song, I. H. Chang, and H. Pham "A software reliability model with a Weibull fault detection rate function subject to operating environments", *Applied Sciences,* Vol. 7, No. 10, 2017, pp.1-16.
[3] H. Pham, "Distribution Function and Its Application in Software Reliability", *International Journal of Performability Engineering,* Vol.15, No. 5, 2019, pp. 1306-1313.
[4] M, Banga, A. Bansal, A. Singh, "Proposed Hybrid Approach to predict Software Fault Detection", *International Journal of Performability Engineering,* Vol. 15, No. 8, 2019, pp. 2049-2061.
[5] T. J. Yang, "Comparative study on the Attributes Analysis of Software Development Cost Model Based on Exponential-Type Lifetime Distribution", *International Journal of Emerging*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Technology and Advanced Engineering, Vol. 11, No. 10, 2021, pp. 166-176.

[6] H. C. Kim, "A Comparative Study on the Cost of Software Development Model Based on Burr–Hatke- Exponential Distribution", *International Journal of Engineering Research and Technology,* Vol. 12, No. 10, 2019, pp. 2036-2040.

[7] T. J. Yang, "A Comparative Study on the Cost and Release Time of Software Development Model Based on Lindley-Type Distribution", *International Journal of Engineering Research and Technology,* Vol. 13, No. 9, 2020. pp.2185-2190.

[8] H. C. Kim, "Software Development Model Based on NHPP Gompertz Distribution", *Indian Journal of Science and Technology,* Vol. 8, No. 12, 2015. pp.1-5.

[9] T. J. Yang, "Comparative Study on the Performance Attributes of NHPP Software Reliability Model based on Weibull Family Distribution", *International Journal of Performability Engineering,* Vol. 17, No. 4, 2021, pp. 343-353.

[10] T. J. 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, 2019, pp.1890-1896.

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

[12] R.S. Prasad, K.R.H. Rao, and R. R. L. Kantha, "Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC", *International Journal of Computer Applications,* Vol. 21, No. 7, 2011, pp. 1-5.

[13] T.J. Yang, "A comparative study on the attributes of finite failure NHPP software reliability with logistic distribution property", *International Journal of Engineering Research and Technology,* Vol. 13, No. 3, 2020, pp. 438-443.

[14] K.C. Chiu, Y.S. Huang and T.Z. Lee, "A Study of Software Reliability Growth from the Perspective of Learning Effects", *Reliability Engineering & System Safety,* Vol. 93, No. 10, 2008, pp. 1410-1421.

[15] T. J. Yang, "A Comparative Study on Reliability Attributes of Software Reliability Model Based on Type-2 Gumbel and Erlang Life Distribution", *Journal of Engineering and Applied Sciences,* Vol. 14, No. 10, 2019, pp. 3366-3370.

[16] S. Maza and O. Megouas, "Framework for Trust worthiness in Software Development", *International Journal of Performability Engineering,* Vol. 17, No. 2, 2021. pp. 241-252.

[17] A. Boranbayev, S. Boranbayev, M. A mirtaev, M. Bai,ukhamedov, A.N urbekov, "Development of a Real Time Human Face Recognition Software System", *Journal of Theoretical and Applied Information Technology,* Vol. 99, No. 7, 2021, pp. 1693 -1705.

[18] A. Alhaddad, B. Al-ibrahi, A. bualkishik, "A Systematic Mapping Study on Software Effort Estimation", *Journal of Theoretical and Applied Information Technology,* Vol. 98, No. 17, 2020, pp. 3619-3643.