Journal of Theoretical and Applied Information Technology

15th October 2023. Vol.101. No 19

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

PERFORMANCE ATTRIBUTES ANALYSIS OF NHPP-BASED SOFTWARE DEVELOPMENT COST MODEL WITH INVERSE-TYPE DISTRIBUTION PROPERTIES

SEUNG KYU PARK

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

ABSTRACT

In this study, after applying the Inverse-type distribution (Inverse-Exponential, Inverse-Rayleigh), which is known to be suitable for reliability research because it can explain various types of life distribution, to the NHPP-based software development cost model, and the attributes that determine the performance of the model were analyzed. Also, to evaluate the efficiency of the proposed model, the optimal model compared with the Goel-Okumoto basic model was also presented. Using the randomly collected failure time data, software failure phenomena were identified and applied to attribute analysis, and maximum likelihood estimation (MLE) was used for the solution of parameters. In conclusion, first, as a result of analyzing the properties of m(t) that affect development cost, the Inverse-exponential model and the Goel-Okumoto basic model were efficient with small prediction errors for the true value. Second, as a result of analyzing the properties of release time along with development cost, the performance of the Inverse-Rayleigh model was the best. Third, as a result of comprehensively evaluating the performance attributes (m(t), cost, release time) of the cost model presented in this work, it was confirmed that the Inverse-Rayleigh model was the best. Therefore, if software developers can efficiently utilize this data in the early process, it is expected that they will be able to efficiently explore and analyze the attributes that affect development cost performance.

Keywords: *Goel-Okumoto, Inverse-Exponential, Inverse-Rayleigh, NHPP, Performance Attributes, Software Development Cost Model*

1. INTRODUCTION

In the era of the 4th industrial revolution led by creation and innovation, high-tech technology that combines software and artificial intelligence is rapidly entering our daily lives. In this era of artificial intelligence, highly reliable software that can process various and complex data without errors is required. For this reason, software developers are concentrating on reliability research to develop high-quality software, but development costs are also becoming a major problem. Therefore, to solve this problem, software developers are investing a lot of time and effort to develop high-quality and reliable software at an economical cost. Thus, many software reliability models applying non-homogeneous Poisson process (NHPP) are being studied in various forms and are evolving into improved models. Especially, the NHPP model using the reliability attributes such as software failure rate is attracting attention [1]. Also, regarding the NHPP-based software reliability cost model presented in this study, Chatterjee, Singh, Roy and Shukla [2] suggested a strategy for the optimal release method based on the software residual failures, and the proposed method was verified by applying to actual data. Pham and Zhang [3] quantitatively predicted product reliability with a model including test coverage, and also proposed a release policy that minimized the expected total cost according to requirements with a software cost model. Moreover, Y. Sarada and R. Shenbagam [4] utilized a phase-type NHPP model to investigate software system cost analysis and operational availability, and then proved useful in reducing the time and effort required to select a reliability model. Also, Kim and Yang [5] presented an optimal software release strategy by comparing the attributes of cost and time in a cost model that can be utilized for software system solutions by applying the characteristics of the Gamma family distribution. In this regard, Kim [6] presented optimal release time data by analyzing the correlation between cost and time with a software development cost model to which NHPP-type Burr-Hatke-exponential distribution was applied. Moreover, after Kim [7] presented the comparison problem of the Gompertz model, he studied problems related to the release time of the cost model according to the life distribution that may occur in the process of analyzing software products. Also, Yang [8] presented a new attribute problem related to performance evaluation of NHPP-based Inverse-Exponential reliability model and solved it by comparing with Exponential-type distribution.

Therefore, the performance attributes of the proposed cost model in this work were newly analyzed and evaluated by applying the Inverse-type distribution, which is well known to be suitable for reliability research because it can explain various types of life distribution. We also suggest the optimal model through the analyzed data.

2. RELATED RESEARCH

2.1 NHPP Software Reliability Model

The NHPP is well known as a probability-based model that predicts the number of occurrences in the future based on the number of successful occurrences by applying a given time, or by applying a certain number of defects per unit.

This model is known to be very efficient in terms of error detection because it assumes that defects are not only removed immediately when they occur but also that no new defects are generated.

That is, if assuming that the accumulated number of software faults is N(t) and the mean value function is m(t), then it is known that N(t) follows the Poisson probability density having the parameter m(t) as in Equation (1).

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

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

Also, it can be seen that the differentiation of m(t) becomes an intensity function λ(t) representing the fault occurrence strength at time t.
As such, time-related models can be explained as stochastic failure processes by NHPP. Thus, m(t) and λ(t) satisfy the relationship as follows.

$$\mathrm{m(t)} = \int_0^t \lambda(s)ds \tag{2}$$

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

These NHPP models are classified into finite failures in which failures do not occur during repairs and infinite failures in which failures continue to occur even during repairs.

In this paper, we will develop this work based on the finite failure NHPP model by applying the actual software development situation. More specifically, finite failure is a model that assumes that no new defect occurs because it is not used during the repair period, but there is a remaining residual failure.

When given sufficient testing time in the NHPP model, if the detectable residual failure rate is θ, the cumulative distribution function is F(t), and the probability density function is f(t), then m(t) and λ(t) can be expressed as the following functional expressions, respectively [9].

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

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

As such, time-domain models can be explained as stochastic failure processes by NHPP. Accordingly, if Equations (4) and (5) are applied and the parameter space of the failure model observed up to the n-th fault is Θ, then the likelihood function of the NHPP model is as follows.

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

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

2.2 NHPP Goel-Okumoto Basic Model

In the field of software reliability, the Goel-Okumoto model is well known as the basic model. In particular, in the Goel-Okumoto basic model, the lifetime distribution following the distribution of failure occurrence time per software defect assumes an exponential distribution. Therefore, the attributes functions of the reliability performance are as follows [10].

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

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

That is, if applying the values of m(t) and λ(t) to Equation (6) and rearranging it, the following equation can be written.

$$\ln L_{NHPP}(\Theta|\underline{x}) = nln\theta + nlnb - b\sum_{k=1}^{n} x_k$$
$$-\ \theta(1 - e^{-bx_n}) \qquad (9)$$

Accordingly, using Equation (9), the estimators $\hat{\theta}_{MLE}$ and $\hat{\tau}_{MLE}$ for the parameters must satisfy the following conditional expression.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}x_n} = 0 \qquad (10)$$

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

2.3 NHPP Inverse-Exponential Model

The Inverse-Weibull distribution, which is known to be suitable for reliability research, is widely applied in the fields of medicine and ecology. In particular, in reliability works, it is well known that the Inverse-Weibull distribution can model very general failure rates. Therefore, the F(t) function of the Inverse-Weibull distribution is as follows [11].

$$F(\mathrm{t}) = e^{-(bt)^{-\gamma}} \qquad (12)$$

It is known that the Inverse-Exponential distribution proposed in this study is established when the shape parameter (γ) is 1 in the Inverse-Weibull distribution as shown in equation (12). Accordingly, the F(t) function is derived as in Equation (13), and if differentiating this equation, the f(t) function can be developed as in Equation (14).

$$\mathrm{F(t)} = e^{-(bt)^{-1}} \qquad (13)$$

$$\mathrm{f(t)} = \mathrm{F(t)}' = b^{-1}t^{-2}e^{-(bt)^{-1}} \qquad (14)$$

Therefore, the attributes functions of the reliability performance are as follows.

$$\mathrm{m}(\mathrm{t}|\theta, \mathrm{b}) = \theta e^{-(bt)^{-1}} \qquad (15)$$

$$\lambda(\mathrm{t}|\theta, \mathrm{b}) = \theta b^{-1}t^{-2}e^{-(bt)^{-1}} \qquad (16)$$

The likelihood function can be obtained by substituting the performance functions obtained in Equations (15) and (16) into Equation (6). Therefore, the log-likelihood function for calculating the parameter ($\hat{\theta}_{MLE}$, $\hat{b}_{MLE}$) by applying the maximum likelihood estimation (MLE) can be developed as in Equation (17).

$$\ln L_{NHPP}(\Theta|\underline{x}) = nln\theta - nlnb \qquad (17)$$
$$+2\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} (bx_i)^{-1} - \hat{\theta}e^{-(bx_n)^{-1}} = 0$$

Accordingly, if Equation (17) developed to calculate the parameters (θ, b) is rearranged after partial differentiation with the parameters θ and b, respectively, Equation (17) can be rewritten as Equations (18) and (19).

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

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial b} = -\frac{n}{\hat{b}} + \frac{1}{\widehat{b^2}}\sum_{i=1}^{n} \frac{1}{x_i} \qquad (19)$$
$$-\theta\frac{1}{b^2x_n}e^{-(\hat{b}x_n)^{-1}} = 0$$

Therefore, if using the bisection method, which is a numerical analysis method applied in this study, it is possible to calculate the maximum likelihood estimator parameters ($\hat{\theta}_{MLE}$, $\hat{b}_{MLE}$).

2.4 NHPP Inverse-Rayleigh Model

The Inverse-Rayleigh distribution is widely applied in reliability research because it can explain various types of lifetime distributions. In particular, it was confirmed that the lifetime distribution of various reliability test devices can be approximated and utilized by applying the Inverse-Rayleigh distribution. Therefore, the distribution functions considering the scale parameter (b) are as follows [12].

$$F(\mathrm{t}) = exp\left(-\frac{b}{t^2}\right) \qquad (20)$$

$$f(\mathrm{t}) = \frac{2b}{t^3}exp\left(-\frac{b}{t^2}\right) \qquad (21)$$

Therefore, the attributes functions of the reliability performance are as follows.

$$\mathrm{m}(\mathrm{t}|\theta, \mathrm{b}) = \theta exp\left(-\frac{b}{t^2}\right) \qquad (22)$$

$$\lambda(\mathrm{t}|\theta, \mathrm{b}) = \theta\left[\frac{2b}{t^3}\ exp\left(-\frac{b}{t^2}\right)\right] \qquad (23)$$

The likelihood function can be obtained by applying the performance functions obtained in Equations (22) and (23) to Equation (6). Therefore, the log-likelihood function for calculating the parameter ($\hat{\theta}_{MLE}$, $\hat{b}_{MLE}$) by applying the MLE can be developed as follows.

$$\ln L_{NHPP}(\Theta|\underline{x}) = nln2 + nln\theta + nlnb$$

$$+b\sum_{i=1}^{n}\ln\left(\frac{1}{x_i^3}\right) - b\sum_{i=1}^{n}\frac{1}{x_i^2} - \theta exp\left(-\frac{b}{x_n^2}\right) \quad (24)$$

If Equation (24) developed to calculate the parameters (θ, b) is rearranged after partial differentiation with the parameters θ and b, respectively, Equation (24) can be rewritten as Equations (25) and (26). Also, if using the bisection method, it is possible to calculate the maximum likelihood estimator parameters ($\hat{\theta}_{MLE}$, $\hat{b}_{MLE}$).

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{n}{\hat{\theta}} - exp\left(-\frac{\hat{b}}{x_n^2}\right) = 0 \quad (25)$$

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

$$+\frac{\hat{\theta}}{x_n^2} exp\left(-\frac{\hat{b}}{x_n^2}\right) = 0 \quad (26)$$

2.5. Software Development Cost Model Applying the NHPP-Based Reliability Model

In this work, we will analyze the performance of the proposed model after applying the m(t) attribute of the NHPP model to the development cost model. It is said that the total software development cost (E_t) of the NHPP-Based software development cost model is composed of the sum of each cost component (E_1~E_4) required in the development process as shown in Equation (27) [13].

$$E_t = E_1 + E_2 + E_3 + E_4 \;= E_1 + C_2 \times t$$

$$+C_3 \times m(t) + C_4 \times [m(t+t') - m(t)] \quad (27)$$

Note that E_t represents the total software development cost.

① E_1 is the development cost invested in the initial stage.

② E_2 is the testing cost per unit time.

$$E_2 = C_2 \times t \quad (28)$$

Note that C_2 is the testing cost.

③ E_3 is the cost of eliminating one defect.

$$E_3 = C_3 \times m(t) \quad (29)$$

Note that C_3 is the cost of removing one error found in the testing phase, and m(t) is an attribute function representing the reliability performance.

④ E_4 is the cost of eliminating all remaining flaws.

$$E_4 = C_4 \times [m(t+t') - m(t)] \quad (30)$$

Note that C4 is the cost of repairing failures found during normal system operation, and t' is the time the system can operate normally after the developed software is released.

Also, software developers will want to release their software at the time when development costs are minimized. That is, the optimal release time is the time point when the total development cost (E_t) becomes the minimum.

Therefore, in this study, we will analyze the attribute relationship to determine the optimal release time according to the development cost trend of the proposed model.

$$\frac{\partial \mathrm{E_t}}{\partial t} = \mathrm{E}' = (\mathrm{E_1} + E_2 + E_3 + E_4)' = 0 \quad (31)$$

3. PERFORMANCE ATTRIBUTES ANALYSIS OF SOFTWARE DEVELOPMENT COST MODEL

This study is related to performance attribute analysis based on reliability of NHPP-based software development cost model. Also, the attribute related to reliability has a characteristic that the probability of occurrence of a specific event (software failure, etc.) must be low.

For this reason, in this work, the performance attributes of the proposed cost model with Inverse-type distribution properties were analyzed and evaluated by applying failure time data as in Table 1.

Table 1 shows the software failure time data applied in this study [14].

Table 1: Software Failure Time Data.

The cited failure time is a collection of failures that occurred randomly during normal operation of the software system, and it is judged that they occurred due to insufficient testing and basic design errors in the early development process. In addition, the applied data is that 30 failures occurred for a total of 187.35 hours, and was collected by the number of failures based on the order of occurrence of failures.

Failure number	Failure time (hours)	Failure time Interval	Failure time (hours)× 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

In general, the occurrence of software failures has the property of being constant, increasing monotonically, or decreasing monotonically regardless of the testing time. Therefore, as a scale method for analyzing this type of failure, a test technique such as the Laplace trend test is widely used [15].

Therefore, in this study, the trend of failure time data presented in Table.1 was verified by applying the Laplace trend test. In general, for such a trend test, if the analysis results are existed between "-2 and 2", the cited data is said to be stable and reliable.

Figure 1 shows the results of the Laplace trend test, and also shows that the cited failure time data is distributed between "0 and 2".

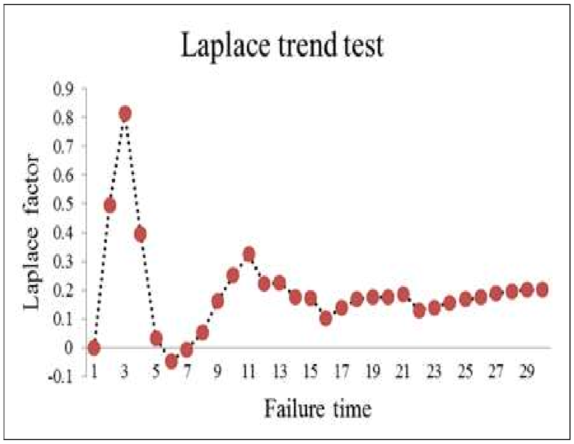

Figure 1: Analysis Results of the Laplace Trend Test.

Accordingly, it can be seen that the software failure time cited in Table.1 can be used in this work because it can be judged to be stable data without extreme values.

Table 2: Parameter Estimator Solution Applying the MLE.

Type	NHPP Model	*Parameter Estimates of the Proposed Model*	
		$\hat{\theta}_{MLE}$	$\hat{b}_{MLE}$
Basic	Goel-Okumoto	32.9261	0.1297
Inverse-type distribution	Inverse-Exponential	41.2881	0.1692
	Inverse-Rayleigh	30.0100	1.6520

3.1. Parameter Calculation of the Proposed NHPP Reliability Model.

As the data presented in Table 1 are numerically converted so that the parameters of the model applied in this work can be easily calculated. Also, the solution of the parameter estimator was calculated by applying the MLE [16].

Table 2 shows the results of calculating the parameters $\left(\hat{\theta}_{MLE},\ \hat{b}_{MLE}\right)$ of the proposed NHPP model by applying MLE. Therefore, if the parameter values obtained in Table 2 are applied to Table 3, the m(t) value of the proposed NHPP model can be obtained.

3.2. Performance of Mean Value Function (m(t))

Table 3 summarizes the method for calculating the cost elements (E_3, E_4) of the NHPP-based software development cost model proposed in this work by applying the equation for obtaining m(t) in the NHPP reliability model.

Table 4 shows in detail the estimation ability of m(t), which represents the predictive power for the true value given equal to the number of failures. Also, m(t) is also known as an attribute function that determines the performance of the software development cost model.

Table 3: Applying m(t) to Calculation of Software Development Cost Model.

Type	NHPP Model	$m(t)$ of Software Reliability Model	$m(t)$ of Software Development Cost Model
Basic	Goel-Okumoto	$m(t) = \theta(1 - e^{-bt})$	$E_3 = C_3 \times m(t)$ $E_4 = C_4 \times [m(t + t') - m(t)]$
Inverse-type distribution	Inverse-Exponential	$m(t) = \theta e^{-(bt)^{-1}}$	
	Inverse-Rayleigh	$m(t) = \theta exp\left(-\frac{b}{t^2}\right)$	

Table 4: Performance Attribute Values Applying m(t).

Failure Number	Failure time (hours)× 10^{-1}	True Value	Basic Model	Inverse-type Distribution Model	
			Goel-Okumoto	Inverse-Exponential	Inverse-Rayleigh
1	0.479	1	1.9833304	0.000180826	0.022402619
2	0.745	2	3.03265707	0.0148088	1.529725269
3	1.022	3	4.08757242	0.127152711	6.171169908
4	1.576	4	6.087035513	0.97090109	15.43155317
5	2.61	5	9.45549807	4.28938387	23.54753178
6	3.559	6	12.17366983	7.845569763	26.34050043
7	4.252	7	13.95757064	10.28412305	27.38941079
8	4.849	8	15.3708973	12.20361469	27.97387557
9	4.966	9	15.63528464	12.55913601	28.0655465
10	5.136	10	16.01235751	13.06374933	28.18821237
11	5.253	11	16.26708425	13.40290443	28.26608262
12	6.527	12	18.80438397	16.69450096	28.86855356
13	6.996	13	19.63779326	17.7392949	29.01398079
14	8.17	14	21.51465663	20.02889247	29.27638411
15	8.863	15	22.49559635	21.19443287	29.38546636
16	10.771	16	24.78221812	23.85188828	29.58569753
17	10.906	17	24.92357228	24.01444705	29.59606464
18	11.183	18	25.20597499	24.33897063	29.61618333
19	11.779	19	25.78026724	24.99860162	29.65479731
20	12.536	20	26.44852342	25.76762543	29.69618234
21	12.973	21	26.80545455	26.18011141	29.7168665
22	15.203	22	28.34272074	27.98935819	29.79626959
23	15.64	23	28.59527673	28.29503943	29.80800676
24	15.98	24	28.78210766	28.52345327	29.8164833
25	16.385	25	28.99416702	28.78540434	29.82590261
26	16.96	26	29.27673326	29.13958574	29.83813881
27	17.237	27	29.40551645	29.30322664	29.84360286
28	17.6	28	29.567428	29.51118859	29.85037786
29	18.122	29	29.78729468	29.79802889	29.8594184
30	18.735	30	30.02718608	30.11770269	29.86908831

Figure.2 applies the data values of m(t) analyzed in Table.4, and shows in detail the trend of m(t) predicting a given true value. Moreover, the failure time data cited in Figure. 2 was used after converting the original failure time to 1/10 to facilitate calculation. That is, as shown in Figure 2, as a result of analyzing the properties of m(t) that affect the performance of the development cost model, it was confirmed that the Inverse-Exponential model and the Goel-Okumoto model, which had a small prediction error for the true value, were efficient.

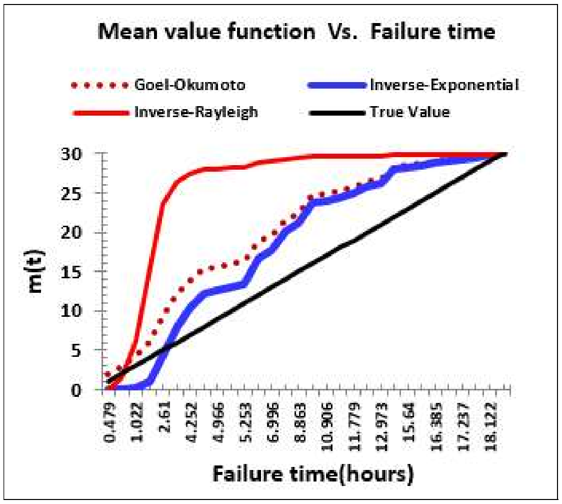

Figure 2: Performance Trend Applying m(t).

3.3. Analysis of software development cost model applying m(t)

The cost of the software development model to be applied in this study was set as [Assumptions.1 to 4] in order to input under conditions similar to the actual development environment [17]. Also, as described in Equation (31), the optimal software release time according to the trend of development cost is the time point when the development cost is minimized.

3.3.1. Assumption 1: basic conditions.

In this section, after setting [Assumption.1] as in Equation (32) as the basic condition, it is analyzed by comparing with [Assumption.2 ~Assumption.4].

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

$$t' = 50(\text{hours}) \tag{32}$$

Figure 3 is the result of analyzing the attribute relationship between development cost and release time by substituting the value of m(t) presented in Table 3 into Equation (27).

As a result of the analysis, the development cost showed a tendency to decrease rapidly at first and gradually increase over time.

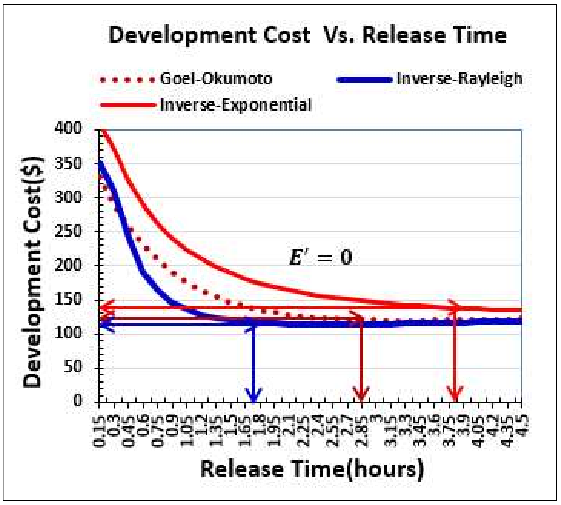

Figure 3: Analysis Results Applying [Assumption 1].

This is because, in the process of removing defects, the probability of finding remaining defects in the software is very high in the beginning, but the probability of finding defects in the later stage decreases as time goes by. Thus, in the end, the cost gradually increases.

3.3.2. Assumption 2: under the condition of Assumption 1, the situation where only the C_2 cost is doubled.

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

$$t' = 50(\text{hours}) \tag{33}$$

The condition of [Assumption 2] is a situation in which only the test cost (C_2) per unit time is doubled (5$→10$) in the same basic condition as [Assumption 1].

Figure 4 is the result of analyzing the performance attribute relationship between development cost and release time under the condition of [Assumption 2] by substituting the m(t) of the proposed models into Equation (27).

As a result, when the cost of the Inverse-Rayleigh model is $125, the release time is 1.725H, when the cost of the Goel-Okumoto model is \$140, the release time is 2.85H, and when the cost of the Inverse-Exponential model is \$160, the release time is 3.825H. This result means that the Inverse-Rayleigh model among the proposed models is an efficient model that can release software the fastest at the lowest cost.

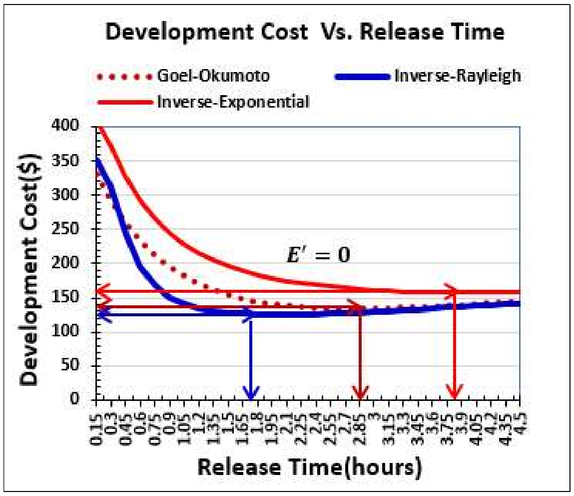

Figure 4: Analysis Results Applying [Assumption 2].

Also, the result of [Assumption 2] compared with [Assumption 1] showed a situation where only the cost attribute increased and the release time attribute did not change at all.

3.3.3. Assumption 3: under the condition of Assumption 1, the situation where only the C_3 cost is doubled.

$$E_1 = 50\$,\quad C_2 = 5\$,\quad C_3 = 3\$,\quad C_4 = 10\$$$
$$t' = 50(\text{hours}) \tag{34}$$

The condition of [Assumption 3] is a situation in which only the cost (C_3) of removing one error found in the development test stage is doubled (1.5\$ → 3\$) under the condition of [Assumption 1].

Figure 5 is the result of analyzing the performance attribute relationship between development cost and release time under the condition of [Assumption 3] by substituting the m(t) of the proposed models into Equation (27).

Therefore, as a result of analyzing the cost at the time of software release and the optimal release time of the proposed model under the conditions of [Assumption 3], Figure 5 showed the situation where only the cost attribute increases and the time attribute does not change at all. That is, to reduce the development cost in this situation, as many defects as possible should be removed at once during the testing process.

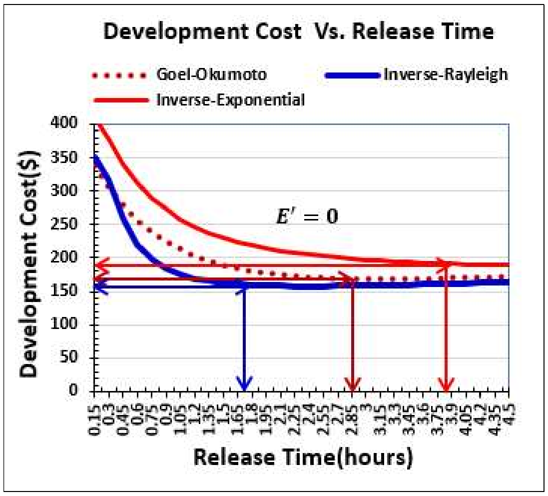

Figure 5: Analysis Results Applying [Assumption 3].

As a result of the analysis, it can be seen that the Inverse-Rayleigh model is an efficient model that can release software the fastest with the lowest release cost.

3.3.4. Assumption 4: under the condition of Assumption 1, the situation where only the C_4 cost is doubled.

$$E_1 = 50\$,\quad C_2 = 5\$,\quad C_3 = 1.5\$,\quad C_4 = 20\$$$
$$t' = 50(\text{hours}) \tag{35}$$

The condition of [Assumption 4] is a situation in which only the cost (C_4) of repairing a failure found by the user during the actual operation stage after software release is doubled (\$10 → \$20) under the condition of [Assumption 1].

Figure 6 is the result of analyzing the performance attribute relationship between development cost and release time under the condition of [Assumption 4] by substituting the m(t) of the proposed models into Equation (27).

As a result, when the cost of the Inverse-Rayleigh model is \$110, the release time is 2.25H, when the cost of the Goel-Okumoto model is \$120, the release time is 3.075H, and when the cost of the Inverse-Exponential model is \$130, the release time is

4.275H. This means that among the proposed models, the Inverse-Rayleigh model is an efficient model that can release software the fastest with the lowest release cost.

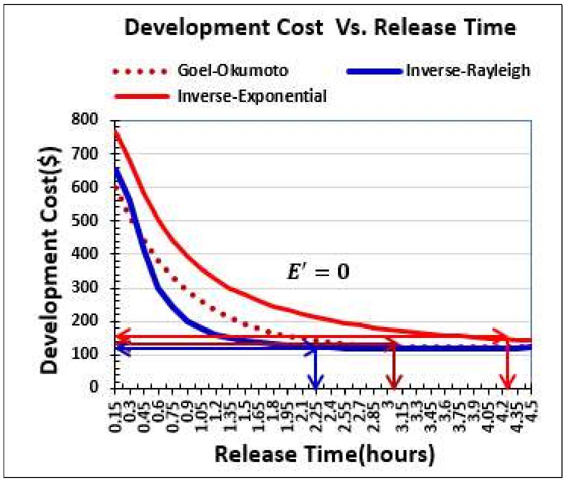

Figure 6: Analysis Results Applying [Assumption 4].

As a result of comparison with [Assumption 1], [Assumption 4] showed a situation in which the release time is delayed along with the increase in development cost. Therefore, in this case, it is necessary to eliminate all possible defects during the development testing phase so that all defects can be eliminated before the software is released [18].

3.4 Performance Attributes Evaluation of the Proposed Software Development Cost Model

Table 5 briefly summarizes the performance evaluation results according to the attributes of the cost models proposed in this work. As a result of comprehensively evaluating the performance attributes (m(t), cost, release time) of the software development cost model proposed in this study, the Inverse-Rayleigh model was found to be the best.

Table 5: Performance Attributes Evaluation.

NHPP model	Performance Attributes		
	m(t)	Cost	Release Time
Goel-Okumoto	Best	Good	Good
Inverse-Exponential	Best	Worst	Worst
Inverse-Rayleigh	Good	Best	Best

4. CONCLUSION

If a software developer can model the reliability of a software system with reliable failure time data collected at an early stage, it will be possible to predict failures that may occur during actual operation in advance and produce more reliable software products. Therefore, by predicting the failure of software products in advance, developers will be able to efficiently develop high-quality software at a more economical cost. Accordingly, in this work, the cost performance attributes of the NHPP-based software development model with Inverse-type distribution characteristics were newly explored and analyzed by applying failure time data.

The results of this study are as follows.
First, as a result of analyzing the properties of m(t) that affect development cost, it was found that the Goel-Okumoto basic model and the Inverse-Exponential model are efficient with small errors in predicting the true value.
Second, as a result of analyzing the properties of the release time along with the development cost by doubling the cost factors (C2, C3, C4) under the conditions of Assumptions 2 to 4 applied in this work, Inverse-Rayleigh model showed the best performance under all conditions.
Third, as a result of comprehensively evaluating the performance attributes (m(t), development cost, release time) of the software development cost model presented in this work, the Inverse-Rayleigh model was confirmed to be the best.

In conclusion, if software developers use this research information in the early stage, it will be able to utilize it as basic design data that can efficiently explore cost attributes along with reliability analysis. Also, in the future, research work to find the optimal cost model after collecting reliable failure time data for each software industry and applying them to various distributions will be necessary continuously.

ACKNOWLEDGEMENTS

Funding for this paper was provided by Namseoul University.

REFERENCE

[1] R. Lai, and M. Trivedi, "A Detailed Study of NHPP Software reliability models", *Journal of Software*, Vol.7, No.6, 2012, pp. 1296-1306.

[2] S, Chatterjee, J. B. Singh, A. Roy & A. Shukla, "NHPP-Based Software Reliability Growth Modeling and Optimal Release Policy for N-Version Programming System with Increasing Fault Detection Rate under Imperfect Debugging", *Proceedings of the National Academy of Sciences, India Section A,* Vol.90, No.1, 2020, pp. 11-26.

[3] H. Pham, X. Zhang "NHPP Software Reliability and Cost Models with Testing Coverage", *European Journal of Operational Research,* Vol.145, 2003, pp.443-454.

[4] R. Shenbagam, Y. Sarada, "On a Cost and Availability Analysis for Software System Via Phase Type Non-homogeneous Poisson Process" *Communications in Statistics – Theory and Methods,* 2023, pp.1-22.

[5] J. M. Kim, Y. J. Yang, "Comparative Study on the Cost Analysis of Software Development Model Applicable to System Solutions Based on Gamma Family Distribution", *International Journal of Electrical Engineering and Technology,* Vol 12, Issue. 9, 2021, pp. 43-54.

[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.11, 2019, pp. 2036-2040.

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

[8] T. J. Yang, "Performance Analysis on the Reliability Attributes of NHPP Software Reliability Model Applying Exponential and Inverse-Exponential Lifetime Distribution", *Journal of Theoretical and Applied Information Technology,* Vol. 100, No.22, 2022, pp. 6645-6656.

[9] H. J. Bae, "Performance Attributes Analysis of Software Development Cost Model with Gamma Family Distribution Characteristics" *Journal of Theoretical and Applied Information Technology,* Vol.101, No.10, 2023, pp.3816-3826.

[10] T. J. Yang, "Comparative Analysis on the Reliability Performance of NHPP Software Reliability Model Applying Exponential-Type Lifetime Distribution", *International Journal of Performability Engineering.* Vol. 18, No. 10, 2022, pp. 679-689.

[11] T. J. Yang, "Comparative Study on the Performance Evaluation of Infinite Failure NHPP Software Reliability Model with Log-Type Distribution Property", *ARPN Journal of Engineering and Applied Sciences.* Vol. 17, No. 11, 2022, pp. 1209-1218.

[12] 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.

[13] T. J. Yang and 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, 2016. pp. 276-281.

[14] Y. Hayakawa, G. Telfar, "Mixed Poisson-type Process with Application in Software Reliability", *Mathematical and Computer Modelling.* Vol. 31, No.10-12, 2000, pp. 151-156.

[15] 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.

[16]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 Tech.,* Vol. 13, No. 3, 2020, pp.438-443.

[17] 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.

[18] T. J. Yang, "Performance Analysis on the Reliability Attributes of NHPP Software Reliability Model Applying Exponential and Inverse-Exponential Lifetime Distribution", *Journal of Theoretical and Applied Information Technology,* Vol. 100, No. 22, 2022, pp. 6645-6656.