ABSTRACT

In this study, properties of software development cost were analyzed by applying the Weibull family lifetime distributions (Lindley, Rayleigh, Type-2 Gumbel) which are utilized in the reliability evaluation field to the software development model. Also, the Weibull family 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, and nonlinear equations were calculated using the binary method. As a result, first, 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. However, 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 Lindley distribution model is 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 use this analytical information efficiently, they can explore the cost of economic development by predicting relevant attributes.

Keywords: lindley, cost attributes, rayleigh, software development model, type-2 gumbel, weibull family distribution.

1. INTRODUCTION

With the rapid growth of software convergence technology, the most important issue is to develop reliable software that can accurately process various and complex large amounts of big data without failure. 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, to estimate the reliability performance, the software reliability models based on finite failure NHPP model using the mean value function were developed [2]. Chatterjee and Singh [3] analyzed the NHPP software reliability and optimal release policy with logistic-exponential testing coverage, while Bajta and Idri, Fernandez-Aleman [4] presented and explained the techniques based on the software cost estimation methods for effective software development. Rashid and Nisar, Mahmood [5] presented a comparative study on the software development cost estimation technique with a software life-cycle model. Kim [6] compared the cost properties of the software development model using Gompertz distribution. Also, Yang [7] compared and presented the research results on the characteristics of the NHPP software reliability model using the Weibull family 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 Poisson distribution is a probabilistic model for the number of occurrences of an event in a given time or event domain. It is a distribution model of a random variable with a very low probability of a specific event occurring among many events. Accordingly, in the NHPP model, if \( N(t) \) is the cumulative number of software failure times found up to time \( t \), then \( m(t) \) is expressed as an average value function representing the expected value of fault occurrence. That is, the equation of the NHPP software reliability model is as follows. That is

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

Note: \( n = 0, 1, 2, \ldots \infty \).

Also, the intensity function \( \lambda(t) \) means the fault occurrence rate per defect.
This study is based on the finite failure NHPP model that does not occur new defects during the removal or correction process of software defects. Therefore, when the residual fault value in the finite failure software system is \( \Theta \), the \( m(t) \) and \( \lambda(t) \) functions are as follows:

\[
m(\theta, b) = \theta F(t) = \theta (1 - e^{-bt}) \tag{4}
\]

\[
\lambda(\theta, b) = \theta f(t) = \theta be^{-bt} \tag{5}
\]

The likelihood function of the finite failure NHPP model using Equations (4) and (5) is as bellows:

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

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

2.2 Goel-Okumoto: NHPP Basic Model

The Goel-Okumoto model is the best-known basic type among the Weibull family lifetime distribution. In the finite failure NHPP, if \( f(t) \) is a probability density function (PDF) and \( F(t) \) is a cumulative distribution function (CDF), then the mean value function \( m(t) \) and the intensity function \( \lambda(t) \) are as follows.

\[
m(\theta, b) = \theta F(t) = \theta (1 - e^{-bt}) \tag{7}
\]

\[
\lambda(\theta, b) = \theta f(t) = \theta be^{-bt} \tag{8}
\]

Note that \( \theta > 0, \; b > 0 \).

After substituting Equations (7) and (8) into Equation (6) to get the likelihood function, and then taking the logarithm of both sides, the log-likelihood function can be solved as follows:

\[
ln(\theta|x) = nln\theta + nlnb - b \sum_{k=1}^{n} x_k - \theta (1 - e^{-bx_n}) \tag{9}
\]

Thus, the calculation of the parameters \( \theta_{\text{MLE}} \) and \( b_{\text{MLE}} \) using maximum likelihood estimation (MLE) in Equations (10) and (11) can be solved by the bisection method.

2.3 Lindley Distribution: NHPP Model

The Lindley distribution is widely known as a suitable model in the reliability evaluation field tests among the Weibull family distribution. In the finite failure NHPP, if \( f(t) \) is a probability density function and \( F(t) \) is a cumulative distribution function, then the average value function \( m(t) \) and the intensity function \( \lambda(t) \) are as follows [7] [8].

\[
m(\theta, b) = \theta F(t) = \theta \left[ 1 - \left( \frac{b + 1 + bt}{b + 1} \right) \times e^{-bt} \right] \tag{12}
\]

\[
\lambda(\theta, b) = \theta f(t) = \theta \left[ \frac{b^2}{b + 1} + (1 + t) \times e^{-bt} \right] \tag{13}
\]

After substituting Equations (12) and (13) into Equation (6) to solve the likelihood function, and then taking the logarithm of both sides, the log-likelihood function can be solved as follows:

\[
ln(\theta|x) = -\theta \left[ 1 - \left( \frac{b + 1 + bt}{b + 1} \right) \times e^{-bt} \right] + nln\theta + 2nlnb - nln(b + 1) + \sum_{i=1}^{n} (1 + x_i) - b \sum_{i=1}^{n} x_i \tag{14}
\]

Thus, the calculation of the parameters \( \theta_{\text{MLE}} \) and \( b_{\text{MLE}} \) using maximum likelihood estimation (MLE) in Equations (15) and (16) can be solved by the bisection method as bellows.

2.4 Rayleigh Distribution: NHPP Model

The Rayleigh distribution is widely used in the lifetime reliability test among the Weibull family distribution. In the finite failure NHPP, if \( f(t) \) is a probability density function and \( F(t) \) is a cumulative
distribution function, then the average value function \( m(t) \) and the intensity function \( \lambda(t) \) are as follows [7] [9].

\[
m(\theta, b) = \theta F(t) = \theta \left(1 - e^{-bt^a}\right)
\]

(17)

\[
\lambda(\theta, b) = \theta f(t) = 2\theta bt^a e^{-bt^a}
\]

(18)

Note. \( \theta > 0, \ b > 0 \).

After substituting Equations (17) and (18) into Equation (6) to solve the likelihood function, if arranged in the same way as in Equation (9), the log-likelihood function can be solved as follows:

\[
\ln(\theta | x) = n \ln \theta + n \ln b + \sum_{i=1}^{n} \ln x_i - b \sum_{i=1}^{n} x_i^2
\]

(19)

Note. \( x = (0 \leq x_1 \leq x_2 \leq \ldots \leq x_n) \).

Thus, the calculation of the parameters \( \hat{\theta}_{MLE} \) and \( \hat{b}_{MLE} \) using maximum likelihood estimation (MLE) in Equations (20) and (21) can be solved by the bisection method.

\[
\frac{\partial (\theta | x)}{\partial \theta} = \frac{n}{\theta} - 1 + e^{\theta(bx_n^a)} = 0
\]

(20)

\[
\frac{\partial (\theta | x)}{\partial b} = \frac{n}{b} - \sum_{i=1}^{n} x_i^2 - \theta x_n^2 e^{\theta(bx_n^a)} = 0
\]

(21)

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

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].

\[
E_t = E_1 + E_2 + E_3 + E_4 = E_1 + C_3 \times t + C_3 \times m(t) + C_3 \times [m(t + t') - m(t)]
\]

(27)

Note that \( E_1 \) is the initial development cost as a constant.

① \( E_1 \) is the initial development cost as a constant.

② \( E_2 \) is the testing cost per unit time.

\[
E_2 = C_2 \times t
\]

(28)

Note that \( C_2 \) is the testing cost per unit time.

③ \( E_3 \) is the cost of removing one fault.

\[
E_3 = C_2 \times m(t)
\]

(29)

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 failure.

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

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

(30)
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. The optimal software releasing time point is as follows:

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

3. SOLUTIONS USING SOFTWARE FAILURE TIME DATA

As shown in Table-1, the cost properties of the proposed distribution model are compared and analyzed using the software failure time data [12] that occurred 30 times during the testing time.

Table-1. Software failure time data.

<table>
<thead>
<tr>
<th>Failure Number</th>
<th>Failure Time (hours)</th>
<th>Failure Time (hours) $\times 10^{-2}$</th>
<th>Failure Number</th>
<th>Failure Time (hours)</th>
<th>Failure Time (hours) $\times 10^{-2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30.02</td>
<td>0.3</td>
<td>16</td>
<td>151.78</td>
<td>1.51</td>
</tr>
<tr>
<td>2</td>
<td>31.46</td>
<td>0.31</td>
<td>17</td>
<td>177.50</td>
<td>1.77</td>
</tr>
<tr>
<td>3</td>
<td>53.93</td>
<td>0.53</td>
<td>18</td>
<td>180.29</td>
<td>1.8</td>
</tr>
<tr>
<td>4</td>
<td>55.29</td>
<td>0.55</td>
<td>19</td>
<td>182.21</td>
<td>1.82</td>
</tr>
<tr>
<td>5</td>
<td>58.72</td>
<td>0.58</td>
<td>20</td>
<td>186.34</td>
<td>1.86</td>
</tr>
<tr>
<td>6</td>
<td>71.92</td>
<td>0.71</td>
<td>21</td>
<td>256.81</td>
<td>2.56</td>
</tr>
<tr>
<td>7</td>
<td>77.07</td>
<td>0.77</td>
<td>22</td>
<td>273.88</td>
<td>2.73</td>
</tr>
<tr>
<td>8</td>
<td>80.90</td>
<td>0.8</td>
<td>23</td>
<td>277.87</td>
<td>2.77</td>
</tr>
<tr>
<td>9</td>
<td>101.90</td>
<td>1.01</td>
<td>24</td>
<td>453.93</td>
<td>4.53</td>
</tr>
<tr>
<td>10</td>
<td>114.87</td>
<td>1.14</td>
<td>25</td>
<td>535</td>
<td>5.35</td>
</tr>
<tr>
<td>11</td>
<td>115.34</td>
<td>1.15</td>
<td>26</td>
<td>537.27</td>
<td>5.37</td>
</tr>
<tr>
<td>12</td>
<td>121.57</td>
<td>1.21</td>
<td>27</td>
<td>552.90</td>
<td>5.52</td>
</tr>
<tr>
<td>13</td>
<td>124.97</td>
<td>1.24</td>
<td>28</td>
<td>673.68</td>
<td>6.73</td>
</tr>
<tr>
<td>14</td>
<td>134.07</td>
<td>1.34</td>
<td>29</td>
<td>704.49</td>
<td>7.04</td>
</tr>
<tr>
<td>15</td>
<td>136.25</td>
<td>1.36</td>
<td>30</td>
<td>738.68</td>
<td>7.38</td>
</tr>
</tbody>
</table>

Figure-1. Results of the Laplace trend test.

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.

In Figure-1, the estimated result of the Laplace trend test existed between 0 and 2. Therefore, this failure data can be used because there are no extreme values [13].

In this study, the parameter estimation was calculated using the maximum likelihood estimation (MLE) method with numerical conversion data to facilitate parameter estimation as in Table-1 [14].

Table-2. below represented the parameter estimated results of the proposed models using MLE.

<table>
<thead>
<tr>
<th>Type</th>
<th>NHPP Model</th>
<th>MLE (Maximum Likelihood Estimation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Model</td>
<td>Goel-Okumoto</td>
<td>$\hat{\theta} = 33.4092$, $\hat{b} = 0.3090$</td>
</tr>
<tr>
<td>Weibull family Lifetime Distribution</td>
<td>Lindley</td>
<td>$\hat{\theta} = 30.4691$, $\hat{b} = 1.3460$</td>
</tr>
<tr>
<td></td>
<td>Rayleigh</td>
<td>$\hat{\theta} = 24.0116$, $\hat{b} = 0.3707$</td>
</tr>
<tr>
<td></td>
<td>Type-2 Gumbel</td>
<td>$\hat{\theta} = 30.3852$, $\hat{b} = 0.6960$</td>
</tr>
</tbody>
</table>
The calculating methods of the mean value function that determines the cost attributes of software development cost 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.

Table-3. Calculating methods of the mean value function m(t)

<table>
<thead>
<tr>
<th>Type</th>
<th>NHPP model</th>
<th>m(t) of Weibull Family Lifetime Distribution</th>
<th>m(t) of Software Development Cost Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic model</td>
<td>Goel-Okumoto</td>
<td>( m(t) = \theta (1 - e^{-bt}) )</td>
<td></td>
</tr>
<tr>
<td>Weibull family Lifetime</td>
<td>Lindley</td>
<td>( m(t) = \theta \left[ 1 - \left( \frac{b + 1 + bt}{b + 1} \right) \times e^{-bt} \right] )</td>
<td>( E_3 = C_3 \times m(t) )</td>
</tr>
<tr>
<td>Lifetime distribution</td>
<td>Rayleigh</td>
<td>( m(t) = \theta (1 - e^{-bt^2}) )</td>
<td>( E_4 = C_4 \times [m(t + t') - m(t)] )</td>
</tr>
<tr>
<td>Type-2 Gumbel</td>
<td></td>
<td>( m(t) = \theta (e^{-bt - a}) )</td>
<td></td>
</tr>
</tbody>
</table>

Also, the estimated result values of mean value function \( m(t) \) are shown in Table-4. After all, these estimated values of Table-4 can be substituted and used as the mean value function \( m(t) \) to calculate the total software development cost of the proposed distribution model.
In this study, to test the same cost conditions as the actual development environment, the software development cost is presented as [Assumption 1] ~ [Assumption 5].

### 3.1 [Assumption 1: Basic Conditions]

\[ E_1 = 50\$, \ C_2 = 5\$, \ C_3 = 1.5\$, \ C_4 = 10\$, \ t' = 50 \] (32)

If substituting the cost parameter value given in Equation (32) and the estimated result of the mean value function \( m(t) \) in Table-4 into the software development cost model equation as in Equation (27), the test simulation result is as bellows:

<table>
<thead>
<tr>
<th>Failure Number</th>
<th>Release Time (hours)</th>
<th>Basic Model</th>
<th>Weibull family Lifetime Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Goel-Okumoto</td>
<td>Lindley</td>
</tr>
<tr>
<td>1</td>
<td>0.3</td>
<td>2.957820066</td>
<td>6.620405287</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>5.653775151</td>
<td>12.2048333</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
<td>8.111048972</td>
<td>16.71095179</td>
</tr>
<tr>
<td>4</td>
<td>1.2</td>
<td>10.35077272</td>
<td>20.2385719</td>
</tr>
<tr>
<td>5</td>
<td>1.5</td>
<td>12.39220679</td>
<td>22.94117459</td>
</tr>
<tr>
<td>6</td>
<td>1.8</td>
<td>14.25290636</td>
<td>24.97708068</td>
</tr>
<tr>
<td>7</td>
<td>2.1</td>
<td>15.94887243</td>
<td>26.49110959</td>
</tr>
<tr>
<td>8</td>
<td>2.4</td>
<td>17.49468937</td>
<td>27.60531347</td>
</tr>
<tr>
<td>9</td>
<td>2.7</td>
<td>18.90365034</td>
<td>28.418248</td>
</tr>
<tr>
<td>10</td>
<td>3.0</td>
<td>20.18787163</td>
<td>29.00711344</td>
</tr>
<tr>
<td>11</td>
<td>3.3</td>
<td>21.35839683</td>
<td>29.43106636</td>
</tr>
<tr>
<td>12</td>
<td>3.6</td>
<td>22.4252918</td>
<td>29.73468794</td>
</tr>
<tr>
<td>13</td>
<td>3.9</td>
<td>23.39773125</td>
<td>29.9513929</td>
</tr>
<tr>
<td>14</td>
<td>4.2</td>
<td>24.28407762</td>
<td>30.10482899</td>
</tr>
<tr>
<td>15</td>
<td>4.5</td>
<td>25.091953</td>
<td>30.21356858</td>
</tr>
<tr>
<td>16</td>
<td>4.8</td>
<td>25.82830466</td>
<td>30.29026193</td>
</tr>
<tr>
<td>17</td>
<td>5.1</td>
<td>26.49946883</td>
<td>30.34420023</td>
</tr>
<tr>
<td>18</td>
<td>5.4</td>
<td>27.1120511</td>
<td>30.38203817</td>
</tr>
<tr>
<td>19</td>
<td>5.7</td>
<td>27.66878613</td>
<td>30.40852029</td>
</tr>
<tr>
<td>20</td>
<td>6.0</td>
<td>28.17700277</td>
<td>30.42701567</td>
</tr>
<tr>
<td>21</td>
<td>6.3</td>
<td>28.64022541</td>
<td>30.43990817</td>
</tr>
<tr>
<td>22</td>
<td>6.6</td>
<td>29.06243752</td>
<td>30.44887924</td>
</tr>
<tr>
<td>23</td>
<td>6.9</td>
<td>29.44726988</td>
<td>30.45511148</td>
</tr>
<tr>
<td>24</td>
<td>7.2</td>
<td>29.79803184</td>
<td>30.45943454</td>
</tr>
<tr>
<td>25</td>
<td>7.5</td>
<td>30.11773975</td>
<td>30.4624291</td>
</tr>
<tr>
<td>26</td>
<td>7.8</td>
<td>30.40914293</td>
<td>30.46450074</td>
</tr>
<tr>
<td>27</td>
<td>8.1</td>
<td>30.67474729</td>
<td>30.46593216</td>
</tr>
<tr>
<td>28</td>
<td>8.4</td>
<td>30.91683686</td>
<td>30.46692012</td>
</tr>
<tr>
<td>29</td>
<td>8.7</td>
<td>31.1374935</td>
<td>30.46760127</td>
</tr>
<tr>
<td>30</td>
<td>9.0</td>
<td>31.33861472</td>
<td>30.46807043</td>
</tr>
</tbody>
</table>
Figure-2. Shape of the cost curve tested by the condition of [Assumption 1].

Analysis of Figure-2 represented that the development cost curve shows a decreasing shape in the initial stage and a constant shape for a short period, but eventually increases with the release time. It is 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 [16].

As shown in Figure-2, the Lindley model showed the best performance among the proposed models (Goel-Okumoto, Rayleigh, Type-2 Gumbel) as it showed the properties of the lowest development cost and the fastest release time.

3.2 [Assumption 2: Assume that only the Cost \( C_2 \) has Doubled Under the Conditions of Assumption 1]

\[ E_1 = 50\$, \ C_2 = 10\$, \ C_3 = 1.5\$, \ C_4 = 10\$, \ t' = 50 \]

[Assumption 2] is a case where only the cost \( C_2 \) has doubled under the condition of [Assumption 1]. If substituting the cost parameter value given in Equation (33) and the estimated result of the mean value function \( m(t) \) in Table-4 into the software development cost model equation as in Equation (27), the test simulation result is as shown in Figure-3.

Figure-3. Shape of the cost curve tested by the condition of [Assumption 2].

Analysis of Figure-3 showed that the development cost increased but the release time did not change at all. Thus, it was confirmed that accurate testing is necessary so that the cost does not increase.

Also, as a result of simulation analysis, the Lindley model with the lowest development cost and the fastest release time among the proposed models was the best.

3.3 [Assumption 3: Assume that only the Cost \( C_3 \) has Doubled under the Conditions of Assumption 1]

\[ E_1 = 50\$, \ C_2 = 5\$, \ C_3 = 3\$, \ C_4 = 10\$, \ t' = 50 \]

[Assumption 3] is a case where only the cost \( C_3 \) has doubled under the condition of [Assumption 1]. If substituting the cost parameter value given in Equation (34) and the estimated result of the mean value function \( m(t) \) in Table-4 into the software development cost model equation as in Equation (27), the test 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, as a result of simulation analysis, the Lindley model with the lowest development cost and the fastest release time among the proposed models was the best.
3.4 [Assumption 4: Assume that only the Cost $C_4$ has Doubled under the Conditions of Assumption 1]

$$E_1 = 50\$, \ C_2 = 5\$, \ C_3 = 1.5\$, \ C_4 = 20\$, \ t' = 50$$ \hspace{1cm} (35)

Assumption 4 is a case where only the cost ($C_4$) has doubled under the condition of [Assumption 1].

If substituting the cost parameter value given in Equation (35) and the estimated result of the mean value function $m(t)$ in Table-4 into the software development cost model equation as in Equation (27), the test 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. 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 [17].

3.5 [Assumption 5: Assume that only the Time $t'$ has Doubled under the Conditions of Assumption 1]

$$E_1 = 50\$, \ C_2 = 5\$, \ C_3 = 1.5\$, \ C_4 = 10\$, \ t' = 100$$ \hspace{1cm} (36)

Assumption 5 is a case where only the time ($t'$) has doubled under the conditions of [Assumption 1].

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

4. CONCLUSIONS

If software developers and operators can quantitatively analyze the attributes of software development costs during the testing process, they can effectively predict economic development costs. Therefore, in this study, the attributes of software development cost were analyzed and predicted based on the Weibull family distribution model.

The results of this study are as follows:

First, under the 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. However, 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 Lindley model showed the best performance among the proposed Weibull family distribution models because it has the characteristics of the lowest software development cost and the fastest release time.

In conclusion, if software developers and operators 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 lifetime model distributions, additional research is needed to find the optimal cost model by applying the software failure time data applied in this paper to the development cost model.

ACKNOWLEDGEMENTS
Funding for this paper was provided by Namseoul University.

REFERENCES