

# 지수족 수명 분포를 이용한 소프트웨어 개발 모형의 성능 평가

배효정\*

## Performance Evaluation of Software Development Model using Exponential Family Life Distribution

Hyo-Jeong Bae\*

### 요약

본 연구에서는 NHPP 기반 소프트웨어 개발 모형에 시스템 고장의 신뢰성 분석에 적합한 지수족 수명 분포를 적용한 후, 개발자가 요구한 고장시간 데이터를 대입하여 모형의 성능을 새롭게 평가하였다. 이에 따라, 첫째, 모형의 성능을 결정하는  $m(t)$  함수를 분석한 결과, 모든 모형이 참값을 예측하는 능력에서 과대하게 추정하는 속성을 보였다. 하지만, 가장 작은 오차를 보인 Burr-Hatke-Exponential 모형은 효율적이었다. 둘째, 모형의 성능에 관한 속성 ( $m(t)$ , 개발비용, 방출 시간)을 종합 평가한 결과, Lindley 모형이 가장 우수함을 확인하였다. 따라서, 본 연구를 통해 지수족 수명 분포의 성능을 새롭게 탐색하였고, 개발자는 관련 데이터를 활용함으로써 초기 단계에서 필요한 속성을 파악하고, 이를 개발 테스트 단계에서 참조 데이터로 활용할 수 있다.

### ABSTRACT

In this study, an exponential family life distribution suitable for reliability analysis of system failures was applied to the NHPP-based software development model, and then the performance of models were newly evaluated by substituting the failure time data requested by the developer. First, as a result of analyzing the  $m(t)$  function that determines the model performance, all models showed the property of overestimating the ability to predict the true value. However, the Burr-Hatke-Exponential model, which showed the smallest error, was efficient. Second, the Lindley model was found to be the best when total evaluating the properties of the model performance ( $m(t)$ , development cost, release time). Therefore, through this study, the performance of the exponential family life distribution was newly explored, and by utilizing related data, developers can identify required properties at an early stage and use them as basic reference data during the development testing phase.

### 키워드

Development Model, Exponential Family, Life Distribution, NHPP, Performance Evaluation  
개발 모형, 지수족, 수명 분포, 비동질 푸아송 과정, 성능 평가

\* 교신저자 : 남서울대학교 드론공간정보공학과  
• 접수일 : 2024. 07. 19  
• 수정완료일 : 2024. 08. 30  
• 게재확정일 : 2024. 10. 12

• Received : Jul. 19, 2024, Revised : Aug. 30, 2024, Accepted : Oct. 12, 2024  
• Corresponding Author : Hyo-Jeong Bae  
Dept. Drone and GIS Engineering, Namseoul University  
Email : baehj@nsu.ac.kr

## 1. 서 론

AI(Artificial Intelligence)이 산업현장뿐만 아니라 우리들의 실제 생활 속에서 활용되고 있는 AI 융합 시대가 더욱 확산되고 있다. 따라서 지금까지 우리가 해왔던 프로세스가 AI 기반의 소프트웨어 형태로 융합되고, 그 과정에서 불필요한 부분들은 제거될 것이다. 이러한 근거로, 개발자들은 효율성이 높고 결함이 없으며, 가장 고 품질이 요구되는 AI 소프트웨어를 개발하는 과정에서 신뢰성과 비용 문제를 가장 중요한 주제로 간주하고 있다. 이를 해결하기 위해 현재까지 많은 소프트웨어 개발 모형들이 다양한 형태로 제안되고 있다. 특히, 개발자들은 NHPP(Non-homogeneous Poisson Process)를 이용한 비용 연구에 주목하고 있다[1]. 본 연구의 주제인 소프트웨어 개발 모형과 관련하여 Ra는 NHPP 모형을 이용하여 개발자가 필요로 하는 속성을 파악하고 관련 속성도 제시하였다[2]. 또한 Chatterjee, Singh, Roy, Shukla는 NHPP 기반의 새로운 소프트웨어 개발 모형을 제안하고, 잔존 결함 수를 활용하는 최적의 알고리즘 전략을 제시하였다[3]. 또한 Pham과 Zhang은 신뢰도를 정량적으로 예측하는 데 사용할 수 있는 NHPP 모형을 제안하고, 개발 비용을 최소화할 수 있는 알고리즘을 단계적으로 제시하였다[4]. Okamura와 Dohi는 새롭고 혁신적인 위상형 소프트웨어 신뢰성 모형을 제안하고 이를 실제 데이터에 적용하여 효율성을 입증하였다[5]. Kim and Yang은 소프트웨어 개발비용 모형을 NHPP 기반의 신뢰성 모형으로 변환한 후, 이를 기반으로 비용의 속성을 새롭게 분석하기도 하였다[6]. Park은 개발비용 및 방출 시간과 관련된 속성 문제를 분석할 수 있는 방법을 제시하였다[7].

따라서, 본 연구에서는 NHPP 기반의 소프트웨어 개발 모형에 다양한 지수형 분포를 적용하였고, 이를 근거로 모형의 성능 문제와 함께 속성도 함께 비교, 평가하였다. 더불어, 검증한 데이터를 기반으로 제안된 모형 중에서 가장 효율적인 모형을 제시하였다.

## II. 관련 연구

### 2.1 NHPP 소프트웨어 신뢰성 모형

NHPP 모형은 단위 시간당 발생하는 고장의 횟수를 기반으로 향후에 발생할 고장을 예측하는 확률 기

반형 모형이다. 여기서, 고장 시간이 서로 다른 간격으로, 연속적으로 발생하는 소프트웨어 시스템에서 단위 시간당 발생하는 고장의 횟수  $N(t)$ 는 푸아송 분포를 따른다. 즉, 소프트웨어 고장의 누적수  $N(t)$ 는 식 (1)과 같이 푸아송 확률 밀도 함수를 따르게 된다.

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

단,  $n = 0, 1, 2 \dots \infty$

여기서,  $m(t)$ 는 고장 발생의 기댓값을 의미하는 속성인 평균값 함수이다.

$$m(t) = \int_0^t \lambda(s) ds \quad \dots(2)$$

또한,  $\lambda(t)$ 는 고장 발생의 강도를 의미하는 속성인 강도 함수이다.

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

NHPP 모형은 유한 고장(결함 수리 중에는 고장이 발생하지 않는다고 가정)과 무한 고장(결함 수리 중에도 고장이 발생한다고 가정)으로 분류한다.

본 연구에서는 현실적인 유한 고장을 적용하여 분석한다. 따라서, 시간  $t$ 까지 발견될 수 있는 잔존 고장의 기댓값을  $\theta$ 라고 하고, 확률 분포함수( $F(t)$ )를 이용하면, NHPP 모형의 신뢰도 성능을 의미하는 속성 함수는 식 (4), (5)와 같다.

$$m(t) = \theta \cdot F(t) \quad \dots(4)$$

$$\lambda(t) = \theta \cdot F(t)' = \theta \cdot f(t) \quad \dots(5)$$

NHPP 모형의 우도 함수(Likelihood Function)는 식 (6)과 같다.

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

단,  $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$

### 2.2 NHPP Exponential-basic 모형

NHPP의 기본 모형으로 널리 알려진 Exponential-basic 모형은 결함당 고장 발생 시간이 지수 형태의 분포 특성을 갖고 있으며, 또한 Goel-Okumoto 기본

모형이라고도 한다. 따라서, 확률 분포함수를 대입하면, 모형 성능의 속성 함수는 식 (7), (8)과 같다.

$$m(t) = \theta (1 - e^{-bt}) \quad \dots(7)$$

$$\lambda(t) = \theta b e^{-bt} \quad \dots(8)$$

따라서, 식 (6)에 대입한 후 로그(Log)를 취하여 정리하면, 로그 우도 함수는 식 (9)와 같다[8].

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

식 (9)를 정리하면, 모수인 최우 추정값  $\hat{\theta}_{MLE}$ 와  $\hat{b}_{MLE}$ 는 식 (10), (11)로 계산된다.

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

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

### 2.3 NHPP Burr-Hatke-exponential 모형

Burr-Hatke-Exponential 모형은 고장 발생률을 나타내는 강도 함수가 수명 분포와 유사하게 증가 또는 감소하는 특성을 갖고 있기 때문에, 시스템의 신뢰도 테스트에 매우 적합한 것으로 알려져 있다.

따라서, 확률 분포함수를 식 (4), (5)에 대입하여 정리하면, 속성 함수는 식 (12), (13)과 같다[9].

$$m(\theta, b) = \theta \left[ 1 - \frac{e^{-bt}}{(1+bt)} \right] \quad \dots(12)$$

$$\lambda(\theta, b) = \theta \left[ b e^{-bt} \cdot \frac{2+bt}{(1+bt)^2} \right] \quad \dots(13)$$

식 (6)에 대입하면, 로그 우도 함수는 식 (14)와 같다.

$$\ln L_{NHPP}(\Theta|\underline{x}) = n \ln \theta + n \ln b - b \sum_{i=1}^n x_i + \sum_{i=1}^n \ln(2+x_i) - 2 \sum_{i=1}^n \ln(1+bx_i) - \theta \left[ 1 - \frac{e^{-bx_n}}{(1+bx_n)} \right] \quad \dots(14)$$

식 (14)를 정리하면, 모수인 최우 추정값  $\hat{\theta}_{MLE}$ 와

$\hat{b}_{MLE}$ 는 식 (15), (16)으로 계산된다.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial \theta} = \frac{n}{\theta} - \left[ 1 - \frac{e^{-bx_n}}{(1+bx_n)} \right] = 0 \quad \dots(15)$$

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial b} = \frac{n}{b} - \sum_{i=1}^n x_i - \sum_{i=1}^n \frac{x_i}{2+bx_i} - 2 \sum_{i=1}^n \frac{x_i}{1+bx_i} - \hat{\theta} x_n e^{-bx_n} \frac{2+bx_i}{(1+bx_i)^2} = 0 \quad \dots(16)$$

### 2.4 NHPP Lindley 모형

Lindley 분포는 지수 분포와 감마 분포가 혼합된 지수형 수명 분포이다. 이 분포는 시스템 신뢰도의 수명 주기 테스트 분야에 적용되며, 주로 공학 및 의료 분야에 적합한 것으로 알려져 있다. 특히 최근에는 신뢰도 수명 테스트를 기반으로 한 다양한 활용 분야에서 Lindley 분포는 기존의 다른 분포와 비교하여 분석하는 사례가 증가하고 있다.

따라서, 확률 분포함수를 대입하면, 성능의 속성 함수는 식 (17), (18)과 같다[10].

$$m(t) = \theta \left[ 1 - \left( \frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] \quad \dots(17)$$

$$\lambda(t) = \theta \left[ \frac{b^2}{b+1} (1+t) \times e^{-bt} \right] \quad \dots(18)$$

식 (6)에 대입하면, 로그 우도 함수는 식 (19)와 같다.

$$\ln L_{NHPP}(\Theta|\underline{x}) = -\theta \left[ 1 - \left( \frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] + n \ln \theta + 2n \ln b - n \ln(b+1) + \sum_{i=1}^n (1+x_i) - b \sum_{i=1}^n x_i \quad \dots(19)$$

따라서, 식 (19)를 정리하면, 모수인 최우 추정값  $\hat{\theta}_{MLE}$ 와  $\hat{b}_{MLE}$ 는 식 (20), (21)과 같이 계산된다.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial \theta} = \frac{n}{\theta} - \left[ 1 - \left( \frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] = 0 \quad \dots(20)$$

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial b} = \frac{2n}{b} - \frac{n}{b+1} - \sum_{i=1}^n x_i - \hat{\theta} e^{-\hat{b}x_n} (x_n - b^2 x_n^2 + b - b^3 x_n^3 - b^3) = 0 \quad \dots(21)$$

### 2.5 NHPP Rayleigh 모형

Rayleigh 분포는 전자과의 특정한 기능적 모델링에 사용되었지만, 최근에는 신뢰도 분석 및 수명 테스트에 적합한 모형으로 알려져 있다. 최근 연구에 따르면 지수분포 특성을 갖는 Rayleigh 분포는 다른 형태의 분포에 비교하여 효율적인 것으로 알려져 있다.

그러므로, 확률 분포함수를 대입하면, 성능의 속성 함수는 식 (22), (23)과 같다.

$$m(t) = \theta(1 - e^{-bt^2}) \quad \dots(22)$$

$$\lambda(t) = 2\theta b t e^{-bt^2} \quad \dots(23)$$

식 (6)에 대입하면, 로그 우도 함수는 식 (24)와 같다.

$$\ln L_{NHPP}(\Theta | \underline{x}) = n \ln 2 + n \ln \theta + n \ln b + \sum_{i=1}^n \ln x_i - b \sum_{i=1}^n x_i^2 - \theta(1 - e^{-bx_n^2}) \quad \dots(24)$$

식 (24)를 정리하면, 모수인 최우 추정량  $\hat{\theta}_{MLE}$  와  $\hat{b}_{MLE}$ 는 식 (25), (26)으로 계산된다.

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-bx_n^2} = 0 \quad \dots(25)$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial b} = \frac{n}{b} - \sum_{i=1}^n x_i^2 - \hat{\theta} x_n^2 e^{-\hat{b}x_n^2} = 0 \quad \dots(26)$$

### 2.6 NHPP 소프트웨어 개발 모형

본 연구에서 이용한 NHPP 모형의 성능 속성 함수인  $m(t)$ 를 적용한 소프트웨어 개발 모형은 식 (27)와 같이 각 구성요소의 총합으로 구성된다[11].

$$E_i = E_1 + E_2 + E_3 + E_4 \quad \dots(27)$$

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

단,  $E_i$ 는 소프트웨어 개발 모형 식을 나타낸다.

따라서, 소프트웨어 개발 모형 식을 구성하는 각 요소의 속성은 다음과 같이 정의할 수 있다.

①  $E_1$ 는 소프트웨어 초기 단계 개발비용이다.

②  $E_2$ 는 소프트웨어 전체 테스트 비용이다.

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

단,  $C_2$ 는 단위시간당 비용이고,  $t$ 는 시간이다.

③  $E_3$ 는 고장 한 개를 제거하는 비용이다.

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

단,  $C_3$ 는 테스트 과정에서 발견된 고장 1개를 수리하는 비용이고,  $m(t)$ 는 NHPP 모형의 성능 속성을 나타내는 평균값 함수이다.

④  $E_4$ 는 소프트웨어 시스템에 남아있는 고장을 수리하는 비용이다.

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

단,  $C_4$ 는 사용자가 발견한 결함 수리에 소요되는 비용이고,  $t'$ 는 시스템을 운용할 수 있는 시간이다.

또한, 모든 개발자들은 개발과정에서 최소의 비용을 가지고, 최적기에 소프트웨어를 방출할 것이다.

따라서, 본 연구에서 적용한 소프트웨어 개발 모형은 식 (31)을 만족한다.

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

## III. 소프트웨어 고장시간을 이용한 개발 모형의 성능 평가

본 연구에서는 개발 모형의 성능을 평가하기 위하여 개발 과정에서 개발자가 요구하는 표 1과 같은 소프트웨어 고장 시간(Software Failure Time)를 사용하였다[12]. 따라서, 본 데이터는 소프트웨어 시스템을 정상적으로 운영하는 738.68시간 동안 발생한 총 30번의 고장을 수집하여 정리한 것이다.

표 1. 소프트웨어 고장시간  
Table 1. Software failure time

Failure number	Failure time(hours)	Failure time(hours) × 10 <sup>-2</sup>
1	30.02	0.30
2	31.46	0.31
3	53.93	0.53
4	55.290	0.55
5	58.720	0.58
6	71.920	0.71
7	77.070	0.77
8	80.900	0.80
9	101.90	1.01
10	114.87	1.14
11	115.34	1.15
12	121.57	1.21

13	124.97	1.24
14	134.07	1.34
15	136.25	1.36
16	151.78	1.51
17	177.50	1.77
18	180.29	1.80
19	182.21	1.82
20	186.34	1.86
21	256.81	2.56
22	273.88	2.73
23	277.87	2.77
24	453.93	4.53
25	535.00	5.35
26	537.27	5.37
27	552.9	5.52
28	673.68	6.73
29	704.49	7.04
30	738.68	7.38

표 1과 같이 고장시간 데이터의 유효성을 검증하기 위해 라플라스 추세 분석(Laplace Trend Analysis)를 사용하였고, 그 결과는 그림 1과 같다[13].

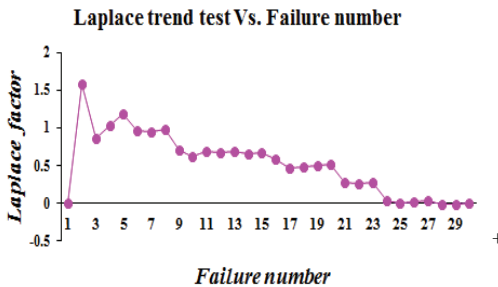


그림 1. 라플라스 추세 분석  
Fig. 1 Laplace trend analysis

통상적으로, 인용된 데이터의 테스트 결과가 “-2와 2” 사이에 존재하면, 안정적이고 신뢰할 수 있다고 한다. 따라서, 그림 1과 같이 고장시간 데이터의 분석 결과가 “0과 2” 사이에 분포하므로, 표 1에서 인용한 데이터는 본 연구에 사용할 수 있다.

표 2는 MLE(Maximum Likelihood Estimation) 방법을 사용하여 제안된 모형의 모수인 최우 추정 값  $\hat{\theta}_{MLE}$ 와  $\hat{b}_{MLE}$ 을 계산한 결과값을 보여주고 있다.

표 2. 제안된 모형의 모수 추정  
Table 2. Parameter estimation of proposed model

Type	NHPP Model	MLE	
		$\hat{\theta}$	$\hat{b}$
Basic	Exponential-basic	29.0332	0.4809
Exponential Family Life Distribution	Burr-Hatke-exponential	29.0996	0.2991
	Lindley	30.4691	1.3460
	Rayleigh	24.0116	0.3707

### 3.1 평균값 함수(m(t))의 성능 분석

그림 2는 소프트웨어 신뢰성 모형의 성능에 중요한 영향을 미치는 속성인 m(t)를 이용하여 모형의 신뢰도 성능을 분석한 것이다. 이 데이터를 참조하면 참값을 추정하는 예측 능력을 분석할 수 있다. 따라서 속성 함수 m(t)를 평가할 때 가장 작은 오차로 참값을 예측하는 경향을 보이는 Burr-Hake-exponential 모형과 Exponential-basic 모형이 효율적이다[14].

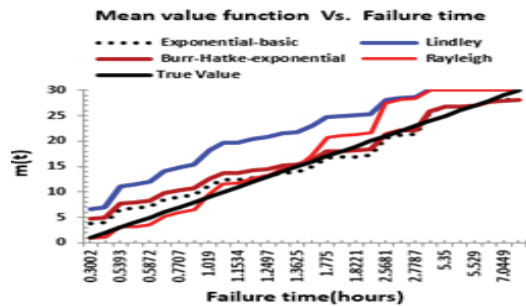


그림 2. m(t)의 성능 분석  
Fig. 2 Performance analysis of m(t)

### 3.2 비용과 방출 시간의 성능 분석

본 연구에서 소프트웨어 개발과 유사한 상황을 투입하기 위한 소프트웨어 개발 비용의 조건은 [가정 1] ~ [가정 4]와 같다. 이를 위해 식 (27)과 같은 개발 모형의 식( $E_i$ )에서 비용을 구성하는 각 속성( $C_2, C_3, C_4$ )를 각각 2배씩 증가시켜서 분석하였다[15].

① [가정 1 : 기본 조건]

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

그림 3은 [가정 1]과 같은 개발 조건을 적용한 시뮬레이션 결과이다. 이 조건에서 비용곡선은 초기에는 급격히 감소한 후, 방출 시간이 흐를수록 비례하여 증가하는 형태를 보였다.

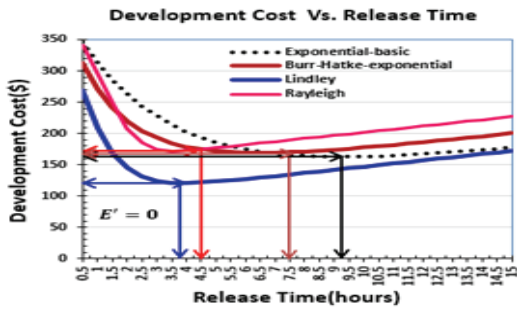


그림 3. 가정 1의 비용과 방출 시간의 성능 분석  
Fig. 3 Performance analysis of cost and release time of assumption 1

즉, 초기 단계에서 결함이 발견될 확률이 높기 때문이 고장이 쉽게 수리되므로 비용은 감소하지만, 후기에는 시간이 갈수록 잔존 결함이 검출될 확률은 점점 감소하여, 결국 비용은 증가하게 된다.

그림 3을 보면, 제안된 모형들은 유사한 패턴을 보였지만, Lindley 모형은 비용이 가장 낮고, 방출 시간도 빠르기 때문에 성능이 가장 효율적이다.

② [가정 2: 가정 1에서  $C_2$  비용이 2배로 증가한 경우]  $E_1 = 50\$$ ,  $C_2 = 10\$$ ,  $C_3 = 1.5\$$ ,  $C_4 = 10\$$ ,  $t' = 50(H)$  ⋯(33)

[가정 2]는  $C_2$  비용을 2배로 증가시킨 조건이다. 그림 4는 [가정 2]와 같은 개발 조건을 적용한 시뮬레이션 결과이다.

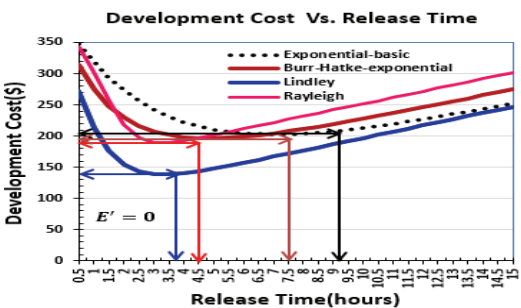


그림 4. 가정 2의 비용과 방출 시간의 성능 분석  
Fig. 4 Performance analysis of cost and release time of assumption 2

분석 결과, 비용은 증가하지만 방출 시간은 변동이 없었다. 이 경우 소프트웨어를 출시하기 전에 테스트

비용이 증가하지 않도록 가능한 적은 시간 내에 정확한 테스트를 해야 한다. 결과적으로, Lindley 모형은 다른 모형보다 비용이 가장 낮고 방출 시간도 빠르기 때문에 성능이 가장 우수하다.

③ [가정 3: 가정 1에서  $C_3$  비용이 2배로 증가한 경우]  $E_1 = 50\$$ ,  $C_2 = 5\$$ ,  $C_3 = 3\$$ ,  $C_4 = 10\$$ ,  $t' = 50(H)$  ⋯(34)

[가정 3]은 개발 테스트 과정에서 검출된 한 개의 결함을 수리하는 비용( $C_3$ )을 2배로 증가시킨 것이다.

그림 5는 [가정 3]과 같은 개발 조건을 적용한 시뮬레이션 결과이다. 분석 결과, 비용은 증가했지만 방출 시간은 변동이 없었다. 이 경우 초기 테스트 단계에서 가능한 한 번에 많은 결함들을 제거해야 한다.

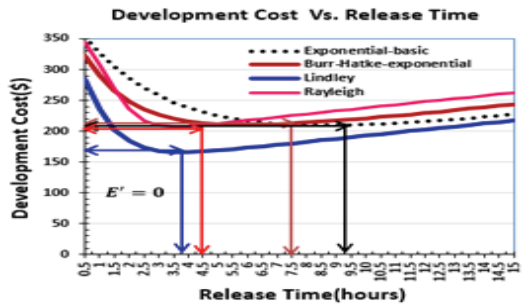


그림 5. 가정 3의 비용과 방출 시간의 성능 분석  
Fig. 5 Performance analysis of cost and release time of assumption 3

결과적으로, Lindley 모형은 다른 모형보다 비용이 가장 낮고 방출 시간도 빠르기 때문에 모형의 성능이 상대적으로 우수함을 알 수 있다.

④ [가정 4: 가정 1에서  $C_4$  비용이 2배로 증가한 경우]  $E_1 = 50\$$ ,  $C_2 = 5\$$ ,  $C_3 = 1.5\$$ ,  $C_4 = 20\$$ ,  $t' = 50(H)$  ⋯(35)

[가정 4]는 소프트웨어 방출 후 운영 단계에서 사용자가 발견한 결함을 수리하는 비용 ( $C_4$ )을 2배로 증가시킨 조건이다.

그림 6은 [가정 4]와 같은 개발 조건을 적용한 시뮬레이션 결과이다. 앞에서 분석한 결과와 다르게 비용이 증가함에 따라서, 방출 시간도 함께 지연되는 것으로 나타났다. 즉, 이 경우 소프트웨어를 방출하기 전에 개발 테스트 단계에서부터 남아있는 모든 결함들을 수리해야만 한다.



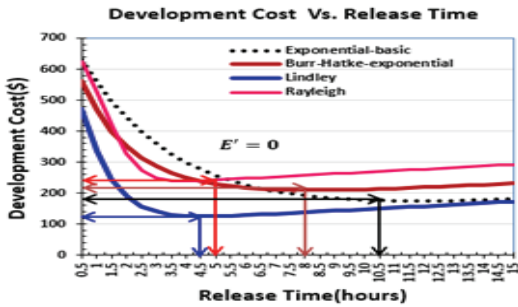


그림 6. 가정 4의 비용과 방출 시간의 성능 분석  
Fig. 6 Performance analysis of cost and release time of assumption 4

결과적으로, Lindley 모형의 성능은 다른 모형보다 비용이 가장 낮고 방출 시간도 빠르기 때문에 제안된 모형 중 가장 우수하였다.

### 3.3 제안된 개발모형의 성능 평가

본 연구에서는 제안된 개발 모형의 성능 평가를 위해 모형의 신뢰도 성능 속성인  $m(t)$ , 개발 비용과 방출 시간에 대한 분석 데이터를 근거로 전개하였다. 표 3은 본 연구의 주제인 모형의 성능에 영향을 미치는 속성 데이터를 상호 비교한 것이다. 결과적으로, Lindley 모형의 성능은 제안된 지수족 분포 모형 중 가장 효율적이며, 신뢰도 속성, 비용 및 방출 시간을 종합 평가한 결과, 가장 우수함을 보였다.

표 3. 제안된 모형의 성능 평가

Table 3. Performance evaluation of proposed model

Type	NHPP Model	Performance Evaluation		
		$m(t)$	Cost	Release Time
Basic	Exponential-basic	Good	Bad	Bad
Exponential Family Life Distribution	Burr-Hatke-exponential	Best	Bad	Good
	Lindley	Good	Best	Best
	Rayleigh	Good	Bad	Bad

## IV. 결론 및 향후 연구과제

소프트웨어 개발 전에 신뢰할 수 있는 소프트웨어 고장시간 데이터를 수집한 후 이를 초기 프로세스에 적용할 수 있다면, 개발자는 모형의 속성을 분석하고

탐색한 후 보다 효율적으로 연구할 수 있을 것이다. 이를 위해, 본 연구에서는 개발자가 요구하는 소프트웨어 고장시간 데이터를 지수형 분포에 적용한 후, 모형의 성능을 새롭게 평가하고, 관련 속성 데이터를 체계적으로 비교, 분석하였다.

본 연구의 결과는 다음과 같다.

첫째, 개발 모형에 중요한 영향을 미치는 속성  $m(t)$ 를 이용하여 참값을 예측한 결과, 모든 모형이 참값을 과대하게 추정하는 경향을 보였지만, Burr-Hatke-Exponential 모형과 Exponential-basic 모형은 작은 오차를 보여서 효율적이었다.

둘째, 비용의 속성을 분석한 결과, Lindley 모형은 개발된 소프트웨어를 가장 낮은 비용으로 빠르게 방출할 수 있다는 점에서 효율적임을 알 수 있었다.

결론적으로, 소프트웨어 개발 모형의 성능에 관한 속성 데이터( $m(t)$ , Cost, Release time)를 종합적으로 분석한 결과, Lindley 모형의 성능이 가장 우수함을 확인하였다. 따라서, 본 연구는 소프트웨어 개발 초기 과정에서 개발자가 필요로 하는 속성 데이터를 분석하고, 이를 개발 단계에서 기본적인 참고 자료로 활용할 수 있을 것으로 기대한다. 또한, 본 연구와 관련하여 소프트웨어 개발 모형에 최적화할 수 있는 통계 분포를 탐색한 후, 이를 기반으로 최적의 모형을 찾는 후속 연구가 필요할 것이다.

### 감사의 글

이 논문은 2024년도 남서울대학교 학술연구비 지원에 의해 연구되었음.

### References

- [1] T. Yang, "Comparative study on the Attributes Analysis of Software Development Cost Model Based on Exponential-type Lifetime Distribution," *Int. J. of Emerging Technology and Advanced Engineering*, vol. 11, no. 10, 2021, pp. 166-176. [https://doi.org/10.46338/ijetae1021\\_20](https://doi.org/10.46338/ijetae1021_20)
- [2] Y. Ra, "A Comparative Analysis on the Performance of Finite Failure NHPP Software

- Reliability Model Based on Rayleigh-type Lifetime Distribution," *J. of Theoretical and Applied Information Technology*, vol. 99, no. 24, 2021, pp. 6162-6172.
- [3] S. Chatterjee, J. B. Singh, A. Roy, and 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," *Proc. of the National Academy of Sciences*, vol. 90, 2020, pp. 11-26. <https://doi.org/10.1007/s40010-018-0551-1>
- [4] H. Pham and X. Zhang, "NHPP Software Reliability and Cost Models with Testing Coverage," *European J. of Operational Research*, vol. 145, issue. 2, 2003, pp. 443-454. [https://10.1016/S0377-2217\(02\)00181-9](https://10.1016/S0377-2217(02)00181-9)
- [5] H. Okamura and T. Dohi, "Phase-type software reliability model: parameter estimation algorithms with grouped data," *Annals of Operations Research*, vol. 244, no. 1, 2016, pp. 77. <https://10.1007/s10479-015-1870-0>
- [6] T. Yang, "A Comparative Study on Reliability Attributes for Software Reliability Model Dependent on Lindley and Erlang Life Distribution," *J. of Korea Institute of Information, Electronics, and Communication Technology*, vol. 10, no. 5, 2017, pp. 469-475. <http://dx.doi.org/10.17661/jkiiect.2017.10.5.469>
- [7] S. Park, "Performance Attributes Analysis of NHP-Based Software Development Cost Model With Inverse-Type Distribution Properties," *J. of Theoretical and Applied Information Technology*, vol. 101, no. 19, 2023, pp.5929-5939.
- [8] T. Yang, "Performance Analysis on the Reliability Attributes of NHPP Software Reliability Model Applying Exponential and Inverse-Exponential Lifetime Distribution," *J. of Theoretical and Applied Information Technology*, vol. 100, no. 22, 2022, pp. 6645-6656.
- [9] H. Kim, "A Comparative Study on the Cost of Software Development Model Based on Burr-Hatke-Exponential Distribution," *Int. J. of Engineering Research and Technology*, vol. 12, no. 11, 2019, pp. 2036-2040.
- [10] H. Kim, "A Comparative Study on the Finite Failure Software Reliability Model with Modified Lindley Type Lifetime Distribution," *Int. J. of Engineering Research and Technology*, vol. 12, no. 6, 2019, pp. 760-764.
- [11] Y. Zhang and K. Wu, "Software Cost Model Considering Reliability and Time of Software in Use," *J. of Convergence Information Technology*, vol. 7, no. 13, 2012, pp. 135-142. <https://10.4156/jcit.vol7.issue13.16>
- [12] R. S. Prasad, K. R. H. Rao, and R. R. L. Kantha, "Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC," *Int. J. of Computer Applications*, vol. 21, no. 7, 2011, pp. 1-5. <https://10.5120/2527-3440>
- [13] S. Park, "Comparative Analysis on the Performance of NHPP Software Reliability Model with Exponential Distribution Characteristics," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 17, no. 4, 2022, pp. 641-648. <http://dx.doi.org/10.13067/JKIECS.2022.17.4.6.41>
- [14] S. Lee, "A Routing Algorithm based on Deep Reinforcement Learning in SDN," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 16, no. 6, 2021, pp. 1153-1160. <http://dx.doi.org/10.13067/JKIECS.2021.16.6.1153>
- [15] S. Park, "Comparative Study on the Performance of Finite Failure NHPP Software Development Cost Model Based on Inverse-Type Life Distribution," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 18, no. 5, 2023, pp. 935-944. <http://dx.doi.org/10.13067/JKIECS.2023.18.5.935>

저자 소개



**배효정(Hyo-Jeong Bae)**

2004년 부경대학교 지능기계공학과 졸업(공학석사)

2017년 부경대학교 지능기계공학과 졸업(공학박사)

1992~2020년 (재)포항산업과학 연구원(RIST)

2021년~현재 남서울대학교 드론공간정보공학과 교수.

※ 관심분야 : 지능제어, 지능 소프트웨어공학, AI