

감마족 분포를 적용한 NHPP 소프트웨어 개발비용 모형의 속성에 관한 비교 분석

배효정*

Comparative Analysis on the Attributes of NHPP Software Development Cost Model Applying Gamma Family Distribution

Hyo-Jeong Bae*

요약

본 연구에서는 감마족 분포(Erlang, Log-Logistic, Rayleigh)를 적용한 NHPP 소프트웨어 개발 비용 모형의 속성을 새롭게 분석하였고, 모형의 속성을 검증하기 위해 Goel-Okumoto 기본 모형과 비교한 후, 이를 근거로 최적의 모형도 제시하였다. 소프트웨어 신뢰도를 분석하기 위하여 시스템 운영 중 랜덤하게 발생한 고장 시간 데이터를 활용하였고, 모수의 계산은 최우추정법을 사용하여 해결하였다. 다양한 속성 분석(평균값 함수, 개발 비용, 최적의 방출시간)을 통하여 종합적으로 평가한 결과, Rayleigh 모형이 가장 우수한 성능을 가진 모형임을 확인하였다. 본 연구를 통하여, 기존 연구 사례가 없는 감마족 분포를 적용한 소프트웨어 개발비용 모형의 속성을 새롭게 규명하였다. 또한, 개발자들이 초기 단계에서 본 연구 데이터를 효율적으로 활용할 수 있도록 기초적인 설계 데이터도 제시할 수 있었다.

ABSTRACT

In this study, the attributes of the NHPP software development cost model applying the Gamma family distribution (Erlang, Log-Logistic, Rayleigh) were newly analyzed, and after comparing with the Goel-Okumoto basic model to verify the properties of the model, the optimal model was also presented based on this. To analyze software reliability, failure time data that occurred randomly during system operation was used, and the calculation of the parameters was solved using the maximum likelihood estimation. As a result of comprehensive evaluation through various attribute analysis (mean value function, development cost, optimal release time), it was confirmed that the Rayleigh model had the best performance. Through this study, the attributes of the software development cost model applying the Gamma family distribution, which has no previous research case, were newly identified. Also, basic design data could also be presented so that developers can efficiently utilize this research data at an early stage.

키워드

Attributes Analysis, Development Cost Model, Gamma Family Distribution, NHPP
속성 분석, 개발 비용 모형, 감마족 분포, 비동질 포아송 과정

* 교신저자 : 남서울대학교 드론공간정보공학과
• 접수일 : 2023. 07. 24
• 수정완료일 : 2023. 09. 02
• 게재확정일 : 2023. 10. 17

• Received : Jul. 24, 2023, Revised : Sep. 02, 2023, Accepted : Oct. 17, 2023
• Corresponding Author : Hyo-Jeong Bae
Dept. Drone and GIS Engineering, Namseoul University
Email : baehj@nsu.ac.kr

I. 서 론

4차 산업혁명 시대의 핵심인 소프트웨어 융합 기술의 급속한 확산으로 인해 다양하고 복잡한 데이터를 고장 없이 정확히 처리할 수 있는 신뢰성 있는 소프트웨어 개발의 필요성도 함께 증가하고 있다. 이러한 소프트웨어 개발 과정에서 가장 중요한 과제는 소프트웨어 신뢰성을 바탕으로 개발 비용을 분석하고 예측하여 가장 경제적인 비용을 찾아내는 것이다. 이러한 목적으로 NHPP (: Non-homogeneous Poisson process)를 적용한 소프트웨어 신뢰성 모형이 소프트웨어 신뢰도를 향상시키기 위해 광범위하게 연구되어 왔고, 다양한 모형 타입으로 진화해 오고 있다. 특히, 평균값 함수를 적용한 NHPP 소프트웨어 신뢰도 모형은 제어된 테스트 환경에서 고장율과 같은 신뢰도 속성을 예측하기 위해 다양한 방법으로 제안되고 있다[1]. 또한, 본 연구에서 제시한 NHPP 기반의 소프트웨어 비용 모델과 관련하여 Chatterjee, Singh, Roy, Shukla는 소프트웨어 잔존 고장을 기반으로 최적의 소프트웨어 방출시기에 대한 전략을 제시하였다[2]. Pham과 Zhang은 소프트웨어 신뢰성을 예측하여, 소프트웨어 비용모형으로 예상되는 총비용을 최소화하는 방법도 제안했다[3]. 또한 Sarada and Shenbagam은 소프트웨어 비용을 분석하기 위해 위상 유형 NHPP 모형을 활용하여 효율성을 입증했다[4]. 또한 Yang은 Lindley 타입 분포에 근거하여 소프트웨어 개발 비용 모형에서 비용과 시간의 속성을 비교하여 최적의 소프트웨어 출시 전략을 제시하였다[5]. 이에 Kim은 NHPP 방식의 Burr-Hatke-exponential 분포를 적용한 소프트웨어 개발비용 모형을 가지고, 비용과 시간의 관계를 분석하여 최적의 방출시점 데이터를 제시하였다[6]. 또한 Kim은 Gompertz 모델을 가지고, 수명 분포에 따른 비용 모델의 출시 시기와 관련된 문제를 연구하였다[7]. 또한 Yang은 NHPP 기반 신뢰성 비용모형의 성능 평가와 관련된 새로운 속성 문제를 제시하고 이를 지수형 분포와 비교하여 해결하였다[8].

따라서, 본 연구에서는 Goel-Okumoto 기본 모형과 함께 소프트웨어 신뢰성 분야에서 널리 사용되는 감마족 분포를 소프트웨어 개발비용 모형에 적용한 후, 적용된 모형의 속성을 새롭게 분석하였다. 또한, 분석된 속성데이터를 근거로, 본 연구에서 제안한 감마족 분포 가운데 최고의 모형도 함께 제시할 것이다.

II. 관련 연구 및 기술

2.1 NHPP 소프트웨어 신뢰성 모형

NHPP 신뢰성 모형은 단위 시간당 발생하는 횟수를 기준으로 모수인 평균값 함수를 이용해서 신뢰도를 분석하는 확률 기반의 예측형 모형이다. 본 연구에서는 소프트웨어 고장시간을 NHPP 모형에 적용한, NHPP 소프트웨어 신뢰성 모형을 가지고 분석한다. 만일, 소프트웨어 고장의 누적수를 $N(t)$, 평균값 함수(Mean Value Function)를 $m(t)$ 라고 하면, 고장 수 $N(t)$ 는 모수 $m(t)$ 를 갖는 포아송 확률밀도함수(Probability Density Function)를 따른다.

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

단, $n = 0, 1, 2 \dots \infty$

여기서, $m(t)$ 는 NHPP 모형의 고장 발생 기댓값의 속성을 나타내는 평균값 함수이다.

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

또한, $m(t)$ 를 미분하면, 고장발생 강도의 속성을 나타내는 강도 함수 $\lambda(t)$ 는 식 (3)과 같다.

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

NHPP 모형은 결함을 수리하는 동안에는 고장이 발생하지 않는다는 유한고장과 결함을 수리하는 동안에도 고장이 발생한다는 무한고장으로 분류된다.

본 연구에서는 유한 고장(Finite Failure)을 기반으로 분석한다. 따라서, 유한고장 NHPP 모형에서 시간 t 까지 발견될 수 있는 고장의 기댓값을 θ , 누적분포 함수를 $F(t)$, 확률밀도함수를 $f(t)$ 라고 하면 신뢰도의 성능을 나타내는 속성 함수는 식 (4), (5)와 같다.

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

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

따라서, 본 연구에서 적용할 NHPP 모형의 우도함수(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 Goel-Okumoto Basic 모형

Goel-Okumoto 모형은 지수 분포 특성을 갖는 모

형으로서 신뢰도 분석에서 가장 널리 알려져 있는 기본 모형이다. Goel-Okumoto 모형에서 고장 발생의 기댓값을 θ 라 하면, 속성 함수는 식 (7), (8)을 만족한다[9].

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

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

따라서, NHPP Goel-Okumoto Basic 모형의 로그우도함수는 식 (9)와 같다.

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

식 (9)를 모수 θ 와 b 로 편미분하여 정리하면, 최우추정값 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 을 이분법으로 구할 수 있다.

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

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

2.3 NHPP Erlang 모형

어랑분포(Erlang distribution)는 신뢰성 분석 분야에서 널리 사용되는 수명분포이며, 감마족 분포에 속한다. 특히, 감마분포에서 형상모수(a)가 상수인 경우를 어랑분포라고 하며, 형상모수(a)에 따른 분포함수는 식 (12), (13)과 같다[10].

$$F(t) = \left(1 - e^{-bt} \sum_{i=0}^{a-1} \frac{(bt)^i}{i!} \right) \quad \dots(12)$$

$$f(t) = \frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt} \quad \dots(13)$$

단, $a, b > 0$, $a = 1, 2, 3 \dots$, $t \in [0, \infty]$ 을 만족한다.

식 (12)와 (13)을 이용하면, 속성 함수는 식 (14), (15)를 만족한다[9].

$$m(t) = \theta \left(1 - e^{-bt} \sum_{i=0}^{a-1} \frac{(bt)^i}{i!} \right) \quad \dots(14)$$

$$\lambda(t) = \theta \frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt} \quad \dots(15)$$

본 연구에서 적용된 모형들은 수명분포의 타입을 결정하는 형상모수(a)가 2인 경우를 적용하고자 한다. 따라서, NHPP Erlang 모형의 로그우도함수는 식 (16)과 같이 쓸 수 있다.

$$\ln L_{NHPP}(\Theta | \underline{x}) = \quad \dots(16)$$

$$n \ln \theta - n \ln \Gamma(a) + na \ln b + (a-1) \sum_{i=1}^n \ln x_i - b \sum_{i=1}^n x_i - \theta + \theta e^{-bx_n} \left(\sum_{i=0}^{a-1} \frac{(bx_n)^i}{i!} \right)$$

식 (16)를 모수 θ 와 b 로 편미분하여 정리하면, 최우추정값 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 을 이분법으로 구할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \theta} = \quad \dots(17)$$

$$\frac{n}{\theta} - 1 + e^{-\hat{b}x_n} (1 + \hat{b}x_n) = 0$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial b} = \quad \dots(18)$$

$$\frac{2n}{\hat{b}} - \sum_{i=1}^n x_i - \hat{\theta} \hat{b} x_n^2 e^{-\hat{b}x_n} = 0$$

2.4 NHPP Log-Logistic 모형

로그-로지스틱 분포(Log-Logistic distribution)는 고장 발생률(강도함수)가 증감하는 특성을 가지고 있어서, 신뢰성 분야에 널리 응용되는 분포다. 따라서, 형상모수(k)에 따른 분포함수는 식 (19), (20)과 같다[11].

$$F(t|\tau, k) = \frac{(\tau t)^k}{[1 + (\tau t)^k]} \quad \dots(19)$$

$$f(t|\tau, k) = \frac{\tau k (\tau t)^{k-1}}{[1 + (\tau t)^k]^2} \quad (\tau, k > 0) \quad \dots(20)$$

식 (19)와 (20)을 이용하면, 속성함수는 식 (21), (22)를 만족한다.

$$m(t) = \theta \frac{(\tau t)^k}{[1 + (\tau t)^k]} \quad \dots(21)$$

$$\lambda(t) = \theta \frac{\tau k (\tau t)^{k-1}}{[1 + (\tau t)^k]^2} \quad \dots(22)$$

본 연구에서는 고장분포의 타입을 결정하는 형상모수(k)가 2인 경우를 적용하고자 한다. 따라서, NHPP Log-Logistic 모형의 로그우도함수는 다음과 같이 유도된다.

$$\ln L_{NHPP}(\Theta | \underline{x}) = n \ln 2 + n \ln \theta + 2n \ln \tau \quad \dots(23)$$

$$+ \sum_{i=1}^n x_i - 2 \sum_{i=1}^n \ln [1 + (\tau x_i)^2] - \theta \frac{(\tau x_n)^2}{[1 + (\tau x_n)^2]} = 0$$

식 (23)를 모수 θ 와 b 로 편미분하여 정리하면, 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{\tau}_{MLE}$ 값을 이분법으로 구할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - \frac{(\hat{\tau} x_n)^2}{[1 + (\hat{\tau} x_n)^2]} = 0 \quad \dots(24)$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \tau} = \frac{2n}{\hat{\tau}} - 2\hat{\tau} \sum_{i=1}^n \frac{x_i^2}{[\ln(1 + (\hat{\tau} x_i)^2)]}$$

$$- \hat{\theta} \left(\frac{2 \hat{\tau} x_n^2 (1 + \hat{\tau}^2 x_n - \hat{\tau}^2 x_n^2)}{[(1 + (\hat{\tau} x_n)^2)]^2} \right) = 0 \quad \dots(25)$$

단, $x = (x_1, x_2, x_3, \dots, x_n)$ 이다.

2.5 NHPP Rayleigh 모형

레이리 분포는 신뢰성 테스트 분야에서 널리 사용되는 수명분포로서, 감마족 분포에 속한다. 이러한, 레이리 분포는 와이블(Weibull distribution) 분포에서 형상모수(α)가 2를 갖는 특수한 경우이다. 형상모수(α)에 따른 분포함수는 식 (26), (27)과 같다[12].

$$F(t) = 1 - e^{-\frac{t^\alpha}{2\beta^2}} \quad \dots(26)$$

$$f(t) = \frac{t^{\alpha-1}}{\beta^2} e^{-\frac{t^\alpha}{2\beta^2}} \quad \dots(27)$$

$\frac{1}{2\beta^2} = b$ 로 치환하여 정리하면 식(28), (29)와 같다.

$$F(t) = 1 - e^{-bt^\alpha} \quad \dots(28)$$

$$f(t) = 2bt^{\alpha-1} e^{-bt^\alpha} \quad \dots(29)$$

단, $b > 0, t \in [0, \infty]$ 의 값을 갖는다.

본 연구에서는 고장분포의 타입을 결정하는 형상모수(α)가 2인 경우를 적용한다. 따라서, 속성함수는 식 (30), (31)을 만족한다[9].

$$m(t|\theta, b) = \theta (1 - e^{-bt^2}) \quad \dots(30)$$

$$\lambda(t|\theta, b) = 2\theta b t e^{-bt^2} \quad \dots(31)$$

따라서, NHPP Rayleigh 모형의 로그우도함수는 식 (32)와 같다.

$$\ln L_{NHPP}(\Theta | \underline{x}) = n \ln 2 + n \ln \theta + n \ln b + \sum_{i=1}^n \ln x_i - b \sum_{i=1}^n x_i^2 - \theta (1 - e^{-bx_n^2}) \quad \dots(32)$$

식 (32)를 모수 θ 와 b 로 편미분하여 정리하면, 최

우추정량 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 값을 이분법으로 구할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b} x_n^2} = 0 \quad \dots(33)$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial b} = \frac{n}{\hat{b}} - \sum_{i=1}^n x_i^2 - \hat{\theta} x_n^2 e^{-\hat{b} x_n^2} = 0 \quad \dots(34)$$

단, $x = (x_1, x_2, x_3, \dots, x_n)$ 이다.

2.6 NHPP 소프트웨어 개발비용 모형

본 연구에서 지금까지 전개한 NHPP 모형의 속성함수 $m(t)$ 를 소프트웨어 개발비용 모형에 적용한, NHPP 소프트웨어 개발비용모형은 식 (35)와 같이 각 구성요소별 비용의 총합으로 구성된다[13].

$$E_t = E_1 + E_2 + E_3 + E_4 \quad \dots(35)$$

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

단, E_t 는 개발 총비용을 나타내고, $m(t)$ 는 제안된 NHPP 모형의 속성(평균값) 함수를 말한다.

따라서, 전체 소프트웨어 비용을 구성하는 각 비용 속성은 다음과 같다.

① E_1 는 소프트웨어 초기 개발비용을 나타내며, 상수로 간주된다.

② E_2 는 단위시간당 테스트 비용으로, 실제로 적용되는 산업분야별로 비용은 상이하다.

$$E_2 = C_2 \times t \quad \dots(36)$$

단, C_2 는 단위시간당 테스트 비용이고, t 는 테스트 시점이다.

③ E_3 는 한 개의 고장을 제거하는 비용이다.

$$E_3 = C_3 \times m(t) \quad \dots(37)$$

단, C_3 는 테스트과정에서 발견된 1개의 고장을 제거하는 비용이고, $m(t)$ 는 고장 발생의 기댓값이다.

④ E_4 는 소프트웨어 시스템에서 잔존하는 모든 결함들을 제거하는 비용이다.

$$E_4 = C_4 \times [m(t+t') - m(t)] \quad \dots(38)$$

단, C_4 는 소프트웨어 방출 후, 사용단계에서 운용자가 발견한 고장을 수리하는 비용이고, t' 는 소프트웨어를 정상적으로 운용할 수 있는 시간이다.

여기서, C_4 는 C_2 와 C_3 보다는 현실적으로 높은 비용 구조를 갖기 때문에, 본 연구에서도 C_4 비용을 C_2 와 C_3 비용보다 높다는 상황을 적용하였다.

모든 소프트웨어 개발자들은 비용이 최소가 되는 시점에 소프트웨어를 출시하고자 할 것이다. 따라서, 최적의 소프트웨어 방출시간은 식 (39)와 같이 소프트웨어 개발 비용이 최소가 되는 시점이 된다. 즉, 식 (39)를 만족해야 한다.

$$\frac{\partial E_t}{\partial t} = E' = (E_1 + E_2 + E_3 + E_4)' = 0 \quad \dots(39)$$

III. 소프트웨어 고장시간을 이용한 개발비용 모형의 속성 분석

본 연구에서는 표 1과 같이 소프트웨어 시스템 운영 중에 랜덤하게 수집된 고장 시간 데이터[14] (Software Failure Time Data)를 이용하여 제안된 모형의 속성을 분석하였다. 본 데이터는 738.68시간 동안 고장이 30번 발생했음을 나타내고 있다.

표 1. 수집된 소프트웨어 고장 시간 데이터
Table 1. Collected software failure time data

Failure number	Failure time(hours)	Failure time(hours) × 10 ⁻²
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 1에서 제시한 고장시간 데이터가 본 연구에 적용 가능한지를 검증하기 위해 그림 1과 같이 라플라스 추세 테스트(Laplace Trend Test)를 사용하였다.

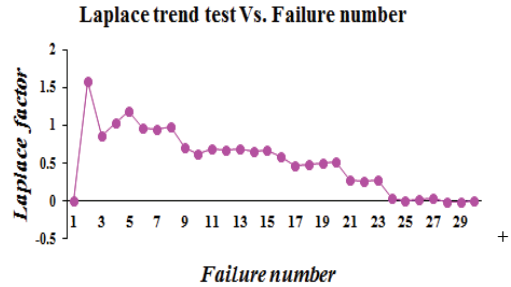


그림 1. 라플라스 추세 테스트의 결과
Fig. 1 Results of laplace trend test

일반적으로, 라플라스 추세 테스트에서 적용된 데이터가 “-2와 2”사이에 분포하면, 극단값이 존재하지 않고, 안정적이기 때문에 사용할 수 있다고 한다[15].

따라서, 적용된 고장시간 데이터의 분석 결과가 그림 1과 같이 “-2와 2사이에 분포”하기 때문에, 극단값이 존재하지 않고 안정적이며, 신뢰할 수 있는 데이터므로 본 연구에 사용할 수 있다고 판단된다.

본 연구에서 제안된 모형의 모수($\hat{\theta}$, \hat{b})의 추정은 최우 추정법(maximum likelihood estimation method, MLE)을 적용하였고,, 그 결과는 표 2와 같다[16].

표 2. 각 모형에 대한 모수추정
Table 2. Parameter estimation of each model

Type	NHPP Model	MLE	
		$\hat{\theta}$	\hat{b}
Basic	Goel-Okumoto	33.4092	0.3090
	Erlang	30.5978	0.7922
Gamma Family Distribution	Log-Logistic	32.2412	0.4953
	Rayleigh	24.0116	0.3707

3.1 평균값 함수(m(t))의 속성 분석

그림 2는 참값(Real Value)을 예측하고 추정하는 성능을 나타내는 $m(t)$ 함수에 대한 속성 분석을 나타내고 있다. 이 시뮬레이션에서 Rayleigh 모형이 참값에 대해 가장 작은 오차(에러)를 보여서, 가장 우수한 성능 속성을 가지고 있음을 알 수 있다[17].

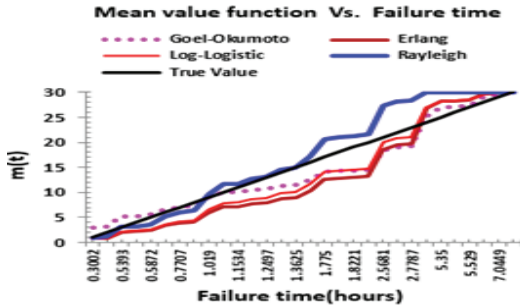


그림 2. $m(t)$ 의 속성 분석
Fig. 2 Attribute analysis of $m(t)$

3.2 개발 비용과 방출 시간의 속성 분석

본 연구에서는 실제 소프트웨어 개발 조건과 동일하게 시뮬레이션하기 위해 소프트웨어 개발 비용을 [가정 1] ~ [가정 4]로 가정하였다. 이를 위해 전체 소프트웨어 개발 비용(E_1)을 구성하는 각 비용 요소 (C_2, C_3, C_4)를 각각 2배씩 증가시켜서 총 소프트웨어 개발 비용의 변화에 따른 속성을 비교하였다[18].

① [가정1 : 기본 조건]

$$E_1 = 40\$, c_2 = 5\$, c_3 = 1.5\$, c_4 = 10\$, t' = 40(H) \quad \dots(40)$$

[가정 1]과 같은 기본 조건을 적용한 시뮬레이션 결과는 그림 3과 같다. 이 결과에서, 비용곡선의 패턴은 초기 단계는 크게 감소한 후, 방출시간이 흐를수록 점진적으로 증가하는 추이를 보이고 있다.

그 이유는 초기 단계에서는 소프트웨어에 내재한 결함이 발견될 확률이 높기 때문에 비용은 감소하지만, 후반 단계에서는 발견되고 남아있는 결함이 검출될 확률이 더욱 감소하기 때문에, 비용은 증가하게 된다. 결국, 방출시간이 흐를수록 개발비용 곡선의 추이는 점진적으로 증가하는 형태가 된다.

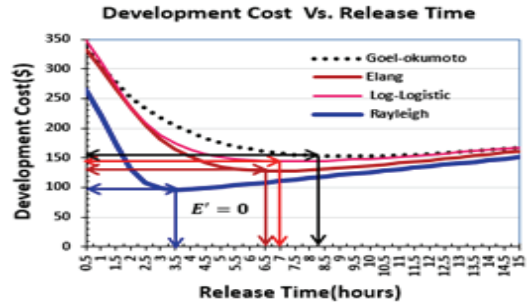


그림 3. 가정1의 개발비용과 방출시간의 속성 분석
Fig. 3 Attributes analysis of development cost and release time under assumption 1

그림 3을 분석한 결과, 제안된 모든 모형들은 유사한 패턴을 보였지만, Rayleigh 모형은 개발 비용이 낮고 방출 시간이 빠르기 때문에 가장 효율적이다.

② [가정2: 가정1에서 C_2 값이 2배로 증가한 경우]

$$E_1 = 40\$, c_2 = 10\$, c_3 = 1.5\$, c_4 = 10\$, t' = 40(H) \quad \dots(41)$$

[가정 2]는 단위 시간당 테스트 비용 (C_2)을 [가정 1]에 비해 2배로 증가시킨 상황이다. 그림 4를 분석한 결과, 개발 비용은 증가했지만 방출 시간은 전혀 변동되지 않았다. 따라서, 이 경우 소프트웨어가 출시되기 전에 단위 시간당 테스트 비용이 증가하지 않도록 빠르고 정확한 테스트가 필요함을 알 수 있다. 또한 Rayleigh 모형은 제안된 다른 모형보다 개발 비용이 낮고 방출 시간이 빠르기 때문에 가장 효율적이다.

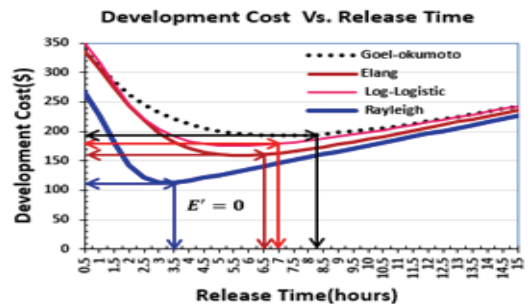


그림 4. 가정2의 개발비용과 방출시간의 속성 분석
Fig. 4 Attributes analysis of development cost and release time under assumption 2

③ [가정3: 가정1에서 C_3 값이 2배로 증가한 경우]
 $E_1 = 40\$, c_2 = 5\$, c_3 = 3\$, c_4 = 10\$, t' = 40(H) \dots(42)$

[가정 3]은 테스트 과정에서 발견된 1개의 결함을 제거하는 비용 (C_3)을 [가정 1]에 비해 2배로 증가시킨 상황이다. 그림 5를 분석한 결과, 개발 비용은 증가했지만 출시 시간은 전혀 변동되지 않았다.

따라서, 이 경우 소프트웨어 테스트 단계에서 가능한 많은 결함을 한 번에 제거해서, 결함을 제거하는 비용이 증가하지 않도록 해야 한다.

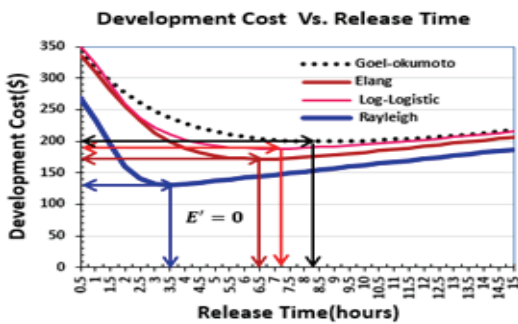


그림 5. 가정3의 개발비용과 방출시간의 속성 분석
 Fig. 5 Attributes analysis of development cost and release time under assumption 3

또한, Rayleigh 모형은 다른 모형보다 개발 비용이 낮고 출시 시간이 빠르기 때문에 상대적으로 효율적임을 알 수 있다.

④ [가정4 : 가정1에서 C_4 값이 2배로 증가한 경우]
 $E_1 = 40\$, c_2 = 5\$, c_3 = 1.5\$, c_4 = 20\$, t' = 40(H) \dots(43)$

[가정 4]는 소프트웨어 출시 후 운영 단계에서 운영자가 발견한 결함을 수정하는 비용(C_4)을 [가정 1]에 비해 2배로 증가시킨 상황이다. 그림 6을 분석한 결과, [가정 2, 3]과 다르게 개발 비용이 증가함에 따라 출시 시간도 지연되는 것을 확인할 수 있다.

즉, 이 경우 소프트웨어를 출시하기 전에 모든 결함을 줄이기 위해서 운영 단계가 아닌 개발 테스트 단계에서 가능한 모든 결함을 제거해야 한다[19].

또한, Rayleigh 모형은 다른 모형들보다 개발 비용

이 낮고 출시 시간이 빠르기 때문에 가장 효율적인 모형임을 확인할 수 있다.

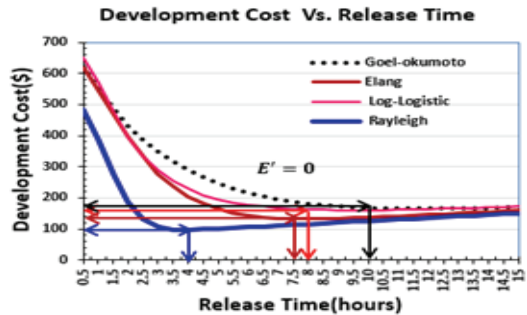


그림 6. 가정4의 개발비용과 방출시간의 속성 분석
 Fig. 6 Attributes analysis of development cost and release time under assumption 4

3.3 제안된 개발비용 모형의 속성 평가

본 연구에서 수행한 속성 연구는 제안된 개발비용 모형의 성능에 영향을 주는 중요한 요인이 된다.

표 3은 감마족 분포를 적용한 NHPP 소프트웨어 개발비용 모형의 속성을 상대적으로 비교하여, 평가한 결과를 요약하여 보여주고 있다[20].

표 3. 제안 모형의 속성 평가

Table 3. Attributes estimation of Proposed model

Type	NHPP Model	Attributes Evaluation		
		m(t)	Cost	Release Time
Gamma Family Distribution	Goel-Okumoto	Good	Worst	Worst
	Erlang	Bad	Good	Good
	Log-Logistic	Bad	Good	Good
	Rayleigh	Best	Best	Best

결론적으로 감마족 분포를 적용한 비용 모형중 Rayleigh 모형이 가장 우수한 모형임을 알 수 있었다.

따라서, 소프트웨어 개발자들이 초기 단계에서 이러한 사전 정보를 이용하면 비용에 대한 속성 정보와 함께 기본적인 설계 데이터로 활용할 수 있을 것이다.

IV. 결론 및 향후 연구과제

소프트웨어 개발자가 초기 단계에서 수집된 신뢰할 수 있는 고장 시간 데이터로 소프트웨어 시스템의 신뢰성을 모델링할 수 있다면 실제 운영 중에 발생할 수 있는 고장을 미리 예측하고 보다 안정적인 소프트웨어 제품을 생산할 수 있을 것이다. 따라서 소프트웨어 제품의 고장을 미리 예측함으로써 개발자는 보다 경제적인 비용으로 고품질의 소프트웨어를 효율적으로 개발할 수 있다. 따라서, 본 연구에서는 감마족 수명분포 분포 특성을 갖는 NHPP 기반 소프트웨어 개발 모델의 비용 속성을 고장시간 데이터를 적용하여 새롭게 탐색하고 분석하였다.

본 연구의 결과는 다음과 같다.

첫째, 소프트웨어 개발비용에 영향을 미치는 $m(t)$ 의 속성을 분석한 결과, Rayleigh 모형이 참값 예측에 오차가 가장 작아서 효율적인 것으로 나타났다.

둘째, 본 연구에서 적용한 [가정 2~4]의 조건에서 비용 요소(C_2, C_3, C_4)를 각각 2배로 하여 개발 비용과 함께 최적의 소프트웨어 방출시기의 속성을 분석한 결과, Rayleigh 모형이 모든 조건에서 최고의 속성을 보였다.

셋째, 본 연구에서 제시한 소프트웨어 개발비용 모델의 성능속성($m(t)$, 개발비용, 최적의 방출시기)을 종합적으로 평가한 결과 Rayleigh 모형이 가장 우수한 것으로 확인되었다.

결론적으로 소프트웨어 개발자가 프로젝트 초기단계부터 테스트 계획을 수립하고, 테스트 설계를 하는 과정에서 이 속성 정보를 활용한다면, 신뢰성 분석과 함께 비용 속성을 효율적으로 탐색할 수 있는 데이터로 활용할 수 있을 것이다. 또한 향후 소프트웨어 산업별로 신뢰할 수 있는 고장시간 데이터를 수집하고 이를 다양한 분포에 적용한 후 최적의 비용 모델을 찾는 후속 작업이 필요할 것이다.

감사의 글

이 논문은 2023년도 남서울대학교 학술연구비 지원에 의해 연구되었음.

References

- [1] K. Song, I. Chang, and H. Pham, "A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments," *Applied Science*, vol. 7, no. 983, 2017, pp. 1-16.
- [2] 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," *Proceedings of the National Academy of Sciences, India Section A*, vol. 90, no. 1, 2020, pp. 11-26.
- [3] H. Pham and 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 and 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] T. 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.
- [6] H. 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. Kim and K. 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. Yang, "Comparative study on the attributes analysis of software development cost model based on exponential-type lifetime distribution," *International Journal of Emerging Technology and*

- Advanced Engineering*, vol. 11, no. 10, 2021, pp. 166-176.
- [9] 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.
- [10] T. Yang, "A Comparative Study on Reliability Attributes of Software Reliability Model Based on Type-2 Gumbel and Erlang Life Distribution," *ARPN Journal of Engineering and Applied Science*, vol. 14, no. 10, 2019, pp. 3366-3370.
- [11] H. Kim, "The Property of Learning effect based on Delayed Software S-Shaped Reliability Model using Finite NHPP Software Cost Model," *Indian Journal of Science and Technology*, vol. 8, no. 34, 2015, pp. 1-7.
- [12] H. Jung, "A proposal of Software Quality Measurement Model and Testcase on the basis of ISO/IEC 25," *The Journal of Korean Institute of Information Technology*, vol. 9, no. 10, 2011, pp. 197-205.
- [13] 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.
- [14] 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 (0975 - 8887)*, vol. 21, no. 7, 2011, pp.1-5.
- [15] Y. Ju, J. Kim and E. Kim, "Development of External Expansion Devices and Convergence Contents for Future Education based on Software Teaching Tools," *The Journal of The Korea Institute of Electronic Communication Sciences*, vol. 16, no. 6, 2021, pp.1317-1322.
- [16] S. Lee, "A Routing Algorithm based on Deep Reinforcement Learning in SDN," *The Journal of The Korea Institute of Electronic Communication Sciences*, vol. 16, no. 6, 2021. pp. 1153-1160.
- [17] Y. Song, Y. Lee and Y. Goo, "A Study on the Weapon System Software Reliability Testing for the Joint Tactical Data Link System Project Case," *The Journal of The Korea Institute of Electronic Communication Sciences*, vol. 17, no. 4, 2022. pp. 663-670.
- [18] J. Seo, "A Study on Image Classification using Deep Learning-Based Transfer Learning," *The Journal of The Korea Institute of Electronic Communication Sciences*, vol. 18, no. 3, 2023. pp. 413-420.
- [19] C. Lee. and H. Baek, "A Study on The Need for AI Literacy According to The Development of Artificial Intelligence Chatbot" *The Journal of The Korea Institute of Electronic Communication Sciences*, vol. 18, no. 3, 2023. pp. 421-426.
- [20] S. Park, "Comparative Analysis on the Performance of NHPP Software Reliability Model with Exponential Distribution Characteristics" *The Journal of The Korea Institute of Electronic Communication Sciences*, vol. 17, no. 4, 2022. pp. 641-648.

저자 소개



배호정(Hyo-Jeong Bae)

2004년 부경대학교 지능기계공학과 졸업(공학석사)

2017년 부경대학교 지능기계공학과 졸업(공학박사)

1992~2020년 (재)포항산업과학 연구원(RIST)

2021년~현재 남서울대학 드론공간정보공학과 교수.

※ 관심분야 : 지능제어, 지능 소프트웨어공학, AI

