

<https://doi.org/10.7236/JIIBC.2022.22.3.193>

JIIBC 2022-3-28

와이블 수명분포에 근거한 소프트웨어 개발모형의 비용 분석에 관한 비교 평가

Comparative Evaluation on the Cost Analysis of Software Development Model Based on Weibull Lifetime Distribution

배효정*

Hyo-Jeong Bae*

요약 본 연구에서는 소프트웨어 신뢰성 분야에서 많이 사용하는 Weibull 수명 분포((Goel-Okumoto, Rayleigh, Type-2 Gumbel)에 근거하여 유한 고장 NHPP 소프트웨어 신뢰성 모형을 소프트웨어 개발 모형에 적용한 후, 비용의 속성을 비교하여, 분석하였다. 본 연구를 위하여 소프트웨어 시스템의 정상 운용 중 검출된 고장시간 데이터를 수집하여 사용하였고, 제안된 모형의 모수 추정은 최우추정법을 적용하였으며, 비선형 방정식의 계산은 이분법을 사용하여 해결하였다. 그 결과, 첫째, 소프트웨어 개발 모형의 단위 시간당 테스트 비용과 단일 고장을 제거하는 비용이 증가하면 비용은 증가하였지만 방출시간은 변하지 않았고, 정상적인 시스템 운용 중에 검출된 고장 수리 비용이 증가하면 비용 증가와 함께 방출 시간도 지연됨을 알 수 있었다. 둘째, 제안된 모형들을 종합적으로 비교 분석한 결과, Type-2 Gumbel 모형이 Rayleigh 모형과 Goel-Okumoto 기본 모형 보다 개발비용이 적고, 방출 시간 포인트도 상대적으로 빨라서 가장 효율적인 모형임을 알 수 있었다. 셋째, 본 연구를 통하여 Weibull 분포 모형의 개발비용 속성을 새롭게 분석하였으며, 분석된 데이터는 소프트웨어 개발자들이 개발 비용과 방출 시간에 대한 속성을 탐색하는 데 필요한 설계 데이터로 활용할 수 있을 것으로 기대한다.

Abstract In this study, the finite-failure NHPP software reliability model was applied to the software development model based on the Weibull lifetime distribution (Goel-Okumoto, Rayleigh, Type-2 Gumbel), which is widely used in the software reliability field, and then the cost attributes were compared and evaluated. For this study, failure time data detected during normal operation of the software system were collected and used, the most-likelihood estimation (MLE) method was applied to the parameter estimation of the proposed model, and the calculation of the nonlinear equation was solved using the binary method. As a result, first, in the software development model, when the cost of testing per unit time and the cost of removing a single defect increased, the cost increased but the release time did not change, and when the cost of repairing failures detected during normal system operation increased, the cost increased and the release time was also delayed. Second, as a result of comprehensive comparative analysis of the proposed models, it was found that the Type-2 Gumbel model was the most efficient model because the development cost was lower and the release time point was relatively faster than the Rayleigh model and the Goel-Okumoto basic model. Third, through this study, the development cost properties of the Weibull distribution model were newly evaluated, and the analyzed data is expected to be utilized as design data that enables software developers to explore the attributes of development cost and release time.

Key Words : Goel-Okumoto, Rayleigh, Software Development Cost, Type-2 Gumbel, Weibull Distribution

*정회원, 남서울대학교 드론공간정보공학과
접수일자 2022년 4월 23일, 수정완료 2022년 5월 23일
게재확정일자 2022년 6월 10일

Received: 23 April, 2022 / Revised: 23 May, 2022 /

Accepted: 10 June, 2022

*Corresponding Author: baehj@nsu.ac.kr

Dept. of Drone and GIS Engineering Namseoul Univ., Korea

1. 서 론

최근 소프트웨어 응용 기술의 급속한 확산으로 소프트웨어 융복합 시스템의 규모는 끊임없이 확장하고 더욱 복잡해지고 있다. 이런 이유로, 대용량의 빅데이터를 고장 없이 정확히 처리할 수 있는 소프트웨어에 개발에 대한 필요성은 더욱 가중되고 있다. 따라서, 이러한 고품질의 신뢰성 있는 소프트웨어를 개발하는 데 있어서 가장 중요한 토픽은 비용에 관한 주제일 것이다^[1]. 지금까지, 소프트웨어 개발 모형의 비용 분석을 위하여 신뢰성 분야에 적합한 확률 분포 함수를 강도 함수와 평균값 함수에 적용한 비동질 푸아송 과정(Non-homogeneous Poisson process ; NHPP) 신뢰성 모형에 관한 연구가 많이 제안되어 왔다^[2]. 이 과정에서, Kim and Shin^[3]은 Weibull 분포를 적용한 학습 효과 기법을 이용하여 NHPP 소프트웨어 신뢰도 모형을 제안하였고, Pham and Zhang^[4]은 NHPP 모형을 기반으로 테스트 범위를 포함하는 소프트웨어 비용 모형을 개발하고, 소프트웨어 제품의 신뢰성을 정량적으로 추정하고, 예측하는 데 사용할 수 있다고 제안하였다. 또한, Nam and Kim^[5]는 NHPP를 기반으로 소프트웨어 수명 주기가 끝날 때까지 예상되는 총 소프트웨어 비용에 대한 방정식을 공식화하였고, Song and Chang^[6]은 소프트웨어 테스트 데이터를 기반으로 NHPP 모델의 적합도를 조사하여, 제안된 모형의 최적 방출 시간을 비교한 후, 비용을 효율적으로 분석할 수 있는 방법을 제안하기도 하였다. 더불어, Kim and Yang^[7]은 로그 파워 강도 함수를 가진 무한 고장 NHPP 신뢰성 모형에 기반하여 소프트웨어 제품 테스트 과정에서 강도 함수를 고려한 소프트웨어 신뢰성 비용 모형을 제시하기도 하였다.

따라서, 본 연구에서는 신뢰도 테스트 분야에서 널리 사용되는 수명 분포인 Weibull 분포(Goel-Okumoto, Rayleigh, Type-2 Gumbel)에 근거하여 유한 고장을 기반으로 NHPP 소프트웨어 개발 모형의 비용 속성을 새롭게 분석하고, 평가하였다. 이와 함께, 소프트웨어 개발자들이 개발 단계에서 최적의 개발비용 모형을 탐색하는데 필요한 새로운 설계 데이터를 제시하고자 한다.

II. 관련 연구 및 기술

1. NHPP 소프트웨어 신뢰성 모형

NHPP 소프트웨어 신뢰성 모형은 단위 시간당 발생되

는 소프트웨어의 고장 발생의 횟수를 기준으로 신뢰도를 예측하고, 분석하는 모형이다. 따라서, NHPP 모형에서 소프트웨어 고장의 누적수 $N(t)$ 는 모수 $m(t)$ 와 모수 $\lambda(t)$ 을 가진 푸아송 확률밀도함수(Probability density function)를 따른다.

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

단, $n = 0, 1, 2 \dots \infty$

여기서, NHPP 모형의 고장발생 기댓값을 나타내는 평균값 함수(Mean Value Function) $m(t)$ 와 고장 발생률을 의미하는 강도 함수(Intensity Function) $\lambda(t)$ 는 다음과 같이 정의된다^[8].

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

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

일반적으로, NHPP 모형은 소프트웨어의 고장을 수리하는 동안에는 고장이 발생하지 않는다고 가정하는 유한 고장과 고장을 수리하는 동안에도 고장이 발생한다고 가정하는 무한 고장으로 구분한다.

본 논문에서는 유한 고장을 기반으로 분석하고자 한다. 따라서, 유한 고장 NHPP 모형에서 임의 테스트 시간 t 까지 검출될 수 있는 잔존 고장률을 θ 라고 하면, 식(2)과 식(3)의 상관 관계식을 적용하여 정리하면, 다음과 같이 유도할 수 있다.

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

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

단, $f(t)$ 는 확률밀도함수, $F(t)$ 는 누적분포함수이다.

식(4)와 식(5)을 적용하여 정리하면, NHPP 모형의 우도 함수(Likelihood function)는 다음과 같다.

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

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

2. NHPP Goel-Okumoto 기본 모형

Goel-Okumoto 모형은 소프트웨어 신뢰도 테스트 분야에서 가장 잘 알려진 기본 모형이다. Goel-Okumoto

기본 모형에서 잔존 고장율인 고장 발생의 기댓값을 θ 라고 하면, 다음 식과 같이 유도된다고 하였다^{[9][14]}.

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

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

식(7)과 식(8)를 식(6)에 대입하여 정리하면 NHPP Goel-Okumoto 기본 모형의 우도 함수는 다음과 같다.

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

단, $\underline{x} = (0 \leq x_1 \leq x_2 \leq \dots \leq x_n)$.

식(9)에서, 양변에 로그를 취하여 정리하면, 로그 우도 함수는 다음과 같이 간략화된다.

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

따라서, 식(10)을 각각 모수 θ 와 모수 b 로 편 미분하여 정리하면, 식(11)과 식(12)과 같다. 여기서, 최우추정량 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 는 이분법으로 계산할 수 있다.

$$\frac{n}{\theta} = 1 - \exp(-\hat{b} x_n), \quad (11)$$

$$\frac{n}{\hat{b}} = \sum_{i=1}^n x_i + \hat{\theta} x_n \exp(-\hat{b} x_n) \quad (12)$$

3. NHPP Rayleigh 모형

Rayleigh 분포는 잘 알려진 Weibull 분포로서, 신뢰도 테스트 분야에 적합한 모형으로서 잘 알려져 있다. 이 분포는 Weibull 분포에서 고장의 형태를 결정하는 형상모수(α)가 2일 때 Rayleigh 분포가 되는 특수한 형태이다. 형상모수를 고려한 확률밀도함수 $f(t)$ 와 누적분포 함수 $F(t)$ 는 다음과 같다.

$$f(t) = \frac{t^{\alpha-1}}{\beta^2} e^{-\frac{t^\alpha}{2\beta^2}} \quad (13)$$

$$F(t) = 1 - e^{-\frac{t^\alpha}{2\beta^2}} \quad (14)$$

단, $\beta > 0, t \in [0, \infty)$

식(13)와 식(14)을 간략화된 수식으로, 표현하기 위해 $\frac{1}{2\beta^2} = b$ 로 치환하면, 다음과 같이 정리된다.

$$f(t) = 2b t^{\alpha-1} e^{-bt^\alpha} \quad (15)$$

$$F(t) = 1 - e^{-bt^\alpha} \quad (16)$$

단, $b > 0, t \in [0, \infty)$

식 (15)와 식(16)에서 형상모수(α)가 2인 경우에 Rayleigh 분포가 되므로, 테스트 시간 t 까지 검출될 수 있는 잔존 고장율인 결합의 기댓값을 θ 라고 하면, 다음 식과 같이 유도할 수 있다고 하였다^[10].

$$m(t|\theta, b) = \theta (1 - e^{-bt^2}) \quad (17)$$

$$\lambda(t|\theta, b) = 2\theta b t e^{-bt^2} \quad (18)$$

단, $\theta > 0, b > 0$

NHPP 로그 우도 함수는 구하기 위해서 식(17)과 식 (18)식을 식(6)에 대입한 후, 양변에 로그를 취하면, 다음과 같이 간략화된다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln 2 + n \ln \theta + n \ln b \quad (19)$$

$$+ \sum_{i=1}^n \ln x_i - b \sum_{i=1}^n x_i^2 - \theta(1 - e^{-bx_n^2})$$

따라서, 식(19)를 각각 모수 θ 와 모수 b 로 편 미분하여 정리하면, 식(20)과 식(21)과 같다. 따라서, 최우추정량 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 는 이분법으로 계산할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-\hat{b} x_n^2} = 0 \quad (20)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial b} = \frac{n}{b} - \sum_{i=1}^n x_i^2 \quad (21)$$

$$-\hat{\theta} x_n^2 e^{-\hat{b} x_n^2} = 0$$

단, $x = (x_1, x_2, x_3, \dots, x_n)$

4. NHPP Type-2 Gumbel 모형

Type-2 Gumbel 분포는 잘 알려진 Weibull 분포의 한 형태로서, 소프트웨어 신뢰도 테스트 및 평가 분야에 효율적인 모형으로 알려져 있다. 따라서, 형상모수(a)를 고려한 신뢰도 함수는 다음과 같이 정의된다^[11].

$$m(t|a, b) = \theta e^{-bt^{-a}} \quad (22)$$

$$\lambda(t|a, b) = \theta (a b t^{-a-1} e^{-bt^{-a}}) \quad (23)$$

단, $a, b > 0, t \in [0, \infty)$

NHPP 로그 우도 함수는 구하기 위해서 식(22)과 식(23)을 식(6)에 대입한 후, 양변에 로그를 취하면, 다음과 같이 정리할 수 있다.

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

본 연구에서는 Rayleigh 분포와 동일한 조건으로 비교 분석하기 위해서 형상모수가 $a=2$ 인 경우를 고려한다. 식(24)를 각각 모수 θ 와 모수 b 로 편 미분하여 정리하면, 식(25)과 식(26)과 같다. 여기서, 최우추정량 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 는 이분법으로 계산할 수 있다.

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

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

5. 소프트웨어 개발 모형

소프트웨어 개발 모형은 다음과 같이 각 구성 요소별 단위 비용의 총합으로 구성되어 있다 [12].

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

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

단, E_t : 소프트웨어 개발 총비용이다.

- ① E_1 은 초기에 투입되는 개발비용이다.
- ② E_2 는 단위 시간당 테스트 비용으로, 적용되는 산업별 단위시간당 비용은 상이하다.

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

단, C_2 는 단위시간당 비용이고, t 는 테스트 시간이다.

- ③ E_3 는 기본적인 고장을 검출하고, 고장 1개를 제거하는 비용이다.

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

단, C_3 는 테스트 과정에서 검출된 고장 1개를 제거하는 비용이며, $m(t)$ 는 제안된 NHPP 모형의 t 시점에서 검출할 수 있는 고장의 기댓값인 평균값 함수이다.

- ④ E_4 는 시스템의 모든 잔존 고장들을 제거하는 비용이다.

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

단, C_4 는 소프트웨어 방출 이후에 사용자가 검출한 고장을 수리하는 비용, t' 는 소프트웨어를 방출한 후, 시스템을 정상적으로 운용할 수 있는 시간이다.

본 연구에서는 C_4 비용이 C_2 비용과 C_3 비용보다는 크다는 현실적인 판단을 적용하였다. 또한, 소프트웨어 방출 시간은 개발 총비용(E_t)이 최소가 될 때, 최적의 방출 시점이 된다. 따라서, 소프트웨어 개발비용과 방출 시간의 관계는 다음과 같다.

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

III. 소프트웨어 고장시간 분석

본 연구에서는 표 1과 같은 소프트웨어 고장시간 자료 [13] (Failure time data)를 사용하여 제안된 모형의 비용 속성을 비교하여, 분석하였다. 이 고장시간 자료는 총 738.68 테스트 시간 동안 30번의 고장이 발생하였음을 나타내고 있다.

표 1. 수집된 소프트웨어 고장시간
Table 1. Collected Software Failure Time

Failure number	Failure time (hours)	Failure numbe	Failure time(hours)
1	30.02	16	151.78
2	31.46	17	177.50
3	53.93	18	180.29
4	55.290	19	182.21
5	58.720	20	186.34
6	71.920	21	256.81
7	77.070	22	273.88
8	80.900	23	277.87
9	101.90	24	453.93
10	114.87	25	535.00
11	115.34	26	537.27
12	121.57	27	552.9
13	124.97	28	673.68
14	134.07	29	704.49
15	136.25	30	738.68

따라서, 본 논문에서는 표. 1과 같이 수집된 고장시간 자료가 신뢰도 분석에 적용할 수 있는지를 판별하기 위하여 라플라스 추세 테스트를 사용하였다.

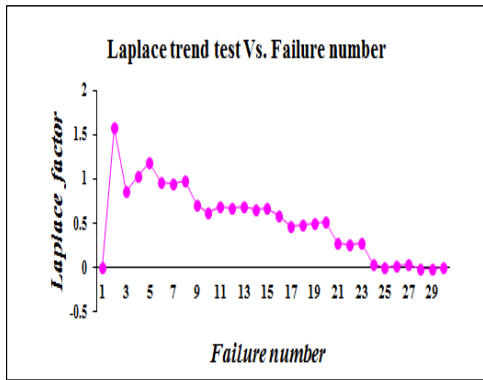


그림 1. 라플라스 추세 테스트의 시뮬레이션 결과
 Fig. 1. Simulation Result of Laplace Trend Test

그림 1과 같이 분석 결과, 라플라스 요인(Factor)의 시뮬레이션 결과값이 -2와 2사이에 분포하였기 때문에, 극단값이 존재하지 않는다. 따라서, 이 고장시간 데이터는 신뢰할 수 있고, 본 연구에 적용할 수 있다.

본 연구에서 제안된 NHPP 모형의 모수추정은 최우추정법(Maximum Likelihood Estimation, MLE)을 사용하였고, 그 결과는 표 2와 같다 [14].

표 2. 제안된 모형의 모수 추정 값
 Table 2. Parameter Estimation of the Proposed Model

Model	MLE	
	$\hat{\theta}_{MLE}$	\hat{b}_{MLE}
Goel-Okumoto	33.4092	0.3090
Rayleigh	24.0116	0.3707
Type-2 Gumbel	30.3852	0.696

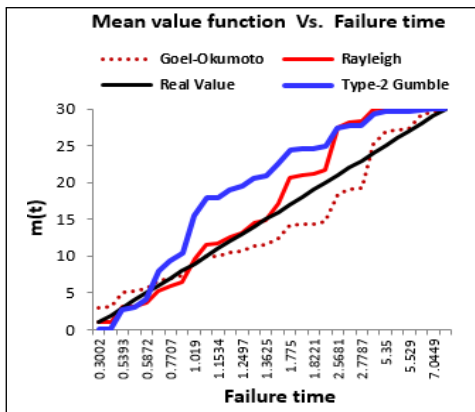


그림 2. 평균값 함수의 예측능력 분석
 Fig. 2. The Predictive Ability Analysis of the Mean Value Function

또한, 제안된 모형들의 비용분석과 함께 신뢰도를 평가하기 위하여 먼저, 평균값 함수 $m(t)$ 에 대한 분석을 수행하였다. 그림 2는 평균값 함수에 대한 분석을 나타내고 있다. 그 결과, Rayleigh 모형은 제안 모형 중 참값에 대한 오차가 가장 작기 때문에, 신뢰도의 적합성 측면에서는 효율적임을 보여주고 있다.

또한, 표 3은 관련 연구에서 분석한 NHPP 모형의 평균값 함수 $m(t)$ 를 가지고, 소프트웨어 개발 모형의 비용 분석에 적용되는 과정을 보여주고 있다 [15].

표 3. 평균값함수와 소프트웨어개발모형의 상관 관계
 Table 3. Correlation between Mean Value Function and Software Development Model

NHPP Model	제안 모형의 평균값 함수	소프트웨어 개발 모형
Goel-Okumoto	$m(t) = \theta(1 - e^{-bt})$	$E_3 = C_3 \times m(t)$
Rayleigh	$m(t) = \theta(1 - e^{-bt^2})$	$E_4 = C_4 \times$
Type-2 Gumbel	$m(t) = \theta e^{-bt^\alpha}$	$[m(t+t') - m(t)]$

본 연구에서는 실제와 동일한 개발 환경을 투입하기 위하여, [가정 1]~[가정 3]과 같은 비용 조건을 적용한 후, 소프트웨어 개발 비용의 속성을 분석하였다.

[가정 1 : 기본 조건]

$$E_1 = 50\$, c_2 = 5\$, c_3 = 1.5\$, c_4 = 10\$, t' = 50 \quad (32)$$

[가정 1]과 같은 비용 조건을 적용한 시뮬레이션 결과는 그림 2와 같다. 시뮬레이션 결과에서 비용곡선의 추이는 처음에는 크게 감소한 후, 일정한 형태를 유지하다가 방출 시간이 흐를수록 점진적으로 증가하는 형태를 보이고 있다. 그 이유는 초기 단계에서 고장을 제거하는 과정 중에 남아있는 고장의 수는 감소하기 때문에 비용은 크게 감소하지만, 시간이 지날수록 남아있는 고장이 발견될 확률은 점진적으로 감소하기 때문이다. 결국, 방출 시간이 흐를수록 비용도 함께 증가하는 형태가 된다.

그림 3과 같이 시뮬레이션 결과를 보면, Type-2 Gumbel 모형의 개발 비용은 150\$, 방출 시간은 3.15H이며, Rayleigh 모형의 개발 비용은 170\$, 방출 시간은 3.75H이다. 그러므로, Type-2 Gumbel 모형이 Rayleigh 모형보다 개발비용이 적고, 방출 시점도 빨라서 효율적임을 알 수 있다.

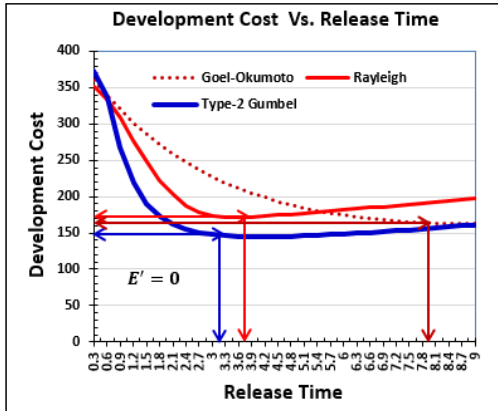


그림 3. [가정1]을 적용한 비용 분석
Fig. 3. The Cost Analysis Applied to the Condition of [Assumption 1]

[가정2 : 가정1에서 C_3 비용이 증가한 경우]

$$E_1 = 50 \$, c_2 = 5 \$, c_3 = 3 \$, c_4 = 10 \$, t' = 50 \quad (33)$$

[가정 2]는 [가정 1]의 기본 조건에서 테스트 과정 중에 발견된 1 개의 고장을 제거하는 비용(C_3)을 2배 증가(1.5\$- 3\$)시킨 상황으로서, 시뮬레이션 결과는 그림 4와 같다.

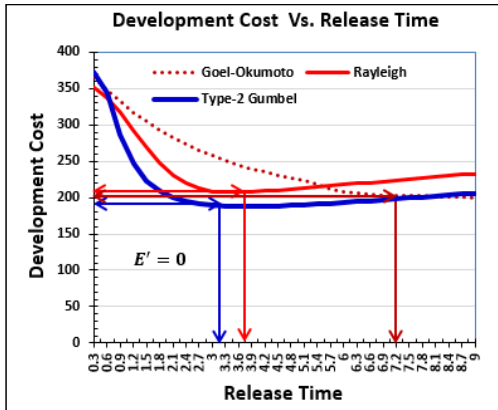


그림 4. [가정2]을 적용한 비용 분석
Fig. 4. The Cost Analysis Applied to the Condition of [Assumption 2]

그림 4와 같이 시뮬레이션 결과를 보면, Type-2 Gumbel 모형의 개발 비용은 190\$, 방출 시간은 3.15H이며, Rayleigh 모형의 개발 비용은 210\$, 방출 시간은 3.75H이다. 그러므로, Type-2 Gumbel 모형이 Rayleigh 모형보다 개발비용이 적고, 방출 시점도 빨라

서 효율적인 모형임을 알 수 있다.

이러한, 결과를 분석하면 개발비용은 증가하였지만, 방출 시간에는 변함이 없었다. 즉, 개발 비용이 증가하지 않도록 하기 위해서는 테스트 단계에서부터 한 번에 가능한 많은 고장을 제거해야만 한다 [16].

[가정3 : 가정1에서 C_4 비용이 증가한 경우]

$$E_1 = 50 \$, c_2 = 5 \$, c_3 = 1.5 \$, c_4 = 20 \$, t' = 50 \quad (34)$$

[가정 3]과 같은 비용 조건을 투입한 시뮬레이션 결과는 그림 4와 같다. [가정 3]은 [가정 1] 기본 조건에서 소프트웨어 방출 이후에, 운용 단계에서 사용자가 검출할 수 있는 고장을 수리하는 비용(C_4)을 2배 증가(10\$- 20\$)시킨 상황이다.

그림 5의 분석 결과를 보면, 개발비용 증가와 함께 방출 시간도 지연된다. 따라서, 이러한 상황 조건에서는 소프트웨어 방출 이전에 고장을 감소시킬 수 있도록 테스트 단계에서 가능한 모든 고장을 제거해야 한다.

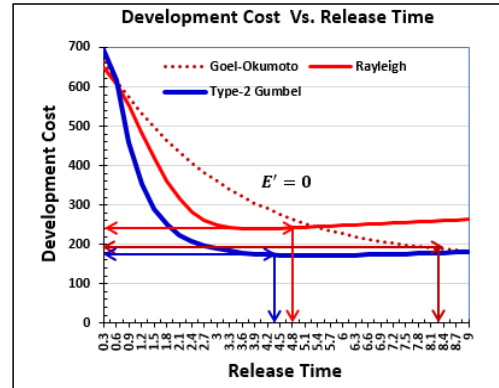


그림 5. [가정3]을 적용한 비용 분석
Fig. 5. The Cost Analysis Applied to the Condition of [Assumption 3]

그러므로, 종합적으로 분석하면 Type-2 Gumbel 모형이 Rayleigh 모형과 Goel-Okumoto 모형 보다 개발 비용이 적고, 소프트웨어 방출 시점도 빨라서 가장 우수한 모형임을 알 수 있다.

V. 결 론

사전에 수집된 고장시간에 대한 분석 자료를 소프트웨

어 개발 과정이나 테스트 작업시에 개발 모형에 적용하여 분석하면, 더욱 효율적으로 비용에 대한 속성을 탐색할 수 있다. 따라서, 본 연구에서는 유한 고장 NHPP 신뢰성 모형을 가지고 Weibull 분포에 적용한 후, 해당 분포의 개발비용 모형을 비교하여 분석하고, 평가하였다.

본 연구의 결과는 다음과 같다.

첫째, 개발 비용은 초기에는 크게 감소한 후, 일정한 패턴을 보이다가, 방출 시간이 흐른 후반 단계에는 점진적으로 증가하는 형태를 보였다. 그 이유는 초기고장을 제거하는 과정에서 잔존 고장의 수는 크게 감소하지만, 시간이 지날수록 잔존 고장이 발견될 확률이 점점 줄어들면서, 결국 비용이 증가하기 때문이다.

둘째, 소프트웨어 방출 이전에 발견된 1개의 고장을 제거하는 비용이 증가하면, 개발 비용은 증가하지만 방출 시간은 전혀 변하지 않았다. 하지만, 소프트웨어 방출 이후에 정상적인 운용 중에 사용자가 발견한 고장 수리 비용이 증가하면, 전체 시점에서 개발비용이 상승하고, 방출시간도 지연된다는 것을 알 수 있었다.

셋째, 본 연구에서 적용된 Weibull 분포의 모형들을 종합적으로 분석한 결과, Type-2 Gumble 모형이 Rayleigh 모형과 Goel-Okumoto 기본 모형 보다 개발 비용이 적고, 방출 시점도 빨라서 상대적으로 가장 우수한 모형임을 알 수 있었다.

본 연구 결과를 이용하면 소프트웨어 개발 및 테스트 단계에서 개발자들에게 경제적인 개발 비용에 관한 분석과 함께 최적의 방출 시간을 예측할 수 있는 설계 데이터를 제시할 수 있다. 더불어 추후에 소프트웨어 산업별로 수집한 고장시간 데이터를 가지고, 다양한 신뢰성 분포 모형에 적용한 후, 평가를 통해 최적의 개발 모형을 탐색하는 후속 연구가 필요하겠다.

References

- [1] T. Dohi, H. Suzuki and S. Osaki, "Transient Cost Analysis of Non-Markovian Software System with Rejuvenation", International Journal of Performance Engineering, Vol. 2, No. 3, pp. 233-243, 2006.
DOI : <https://doi.org/10.23940/ijpe.06.3.p233.mag>
- [2] 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, pp. 2036-2040, 2019.
- [3] H.C. Kim and H. C. Shin, "The Study of NHPP Software Reliability Model from the Perspective of Learning Effects", Journal of Information and Security, Vol. 11, No. 1, pp. 25-32, 2013.
- [4] H. Pham and X. Zhang , "NHPP software reliability and cost models with testing coverage", European Journal of Operational Research, Vol 145, Issue 2, pp. 443-454. 2003.
DOI : [https://doi.org/10.1016/S0377-2217\(02\)00181-9](https://doi.org/10.1016/S0377-2217(02)00181-9)
- [5] K H. Nam and D, H. Kim, "The Software Reliability Growth Models for Software Life-Cycle Based on NHPP", The Korean Journal of Applied Statistics, Vol. 23, No. 3, pp. 573-584, 2010.
DOI : <https://doi.org/10.5351/KJAS.2010.23.3.573>
- [6] K Y, Song and I, H. Chang, "The Optimal Release Time using Software Reliability Model with a Burr Type III Fault Detection Rate Function", Journal of the Korean Data Analysis Society, Vol. 19, No. 2, pp. 577-586, 2017.
DOI : <https://doi.org/10.37727/jkdas.2017.19.2.577>
- [7] T, J. Yang and H, C. Kim, "A Comparative Study on Infinite NHPP Reliability Cost Model Based on Intensity Function of Log Power Form", Journal of Knowledge Information Technology and Systems, Vol. 9, No. 2, pp. 295-302, 2014.
- [8] S. S. Gokhale and Trivedi "A time/structure based software reliability model", Software Engineering. 8, pp. 85-121. 1999.
DOI : <https://doi.org/10.1023/A:1018923329647>
- [9] T. J. 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, pp. 3366-3370. 2019.
DOI : <https://doi.org/10.36478/jeasci.2019.3366.3370>
- [10] H. C. Kim, "The Comparative Study for Statistical Process Control of Software Reliability Model on Finite and Infinite NHPP using Rayleigh Distribution". International Journal of Soft Computing, Vol. 11, No. 3, pp. 165-171. 2016.
- [11] T. J. Yang, "A Study on the Attributes of Software Reliability Cost Model with Shape Parameter Change of Type-2 Gumbel Life Distribution". The Journal of Korea Institute of Information Electronics and Communication Technology., Vol. 12, No. 3, pp. 211-217. 2019.
DOI : <https://doi.org/10.17661/jkiict.2019.12.3.211>
Information Electronics and Communication Technology., Vol. 12, No. 3, pp. 211-217. 2019.
DOI : <https://doi.org/10.17661/jkiict.2019.12.3.211>
- [12] Ye Zhang, and Kaigui Wu, "Software Cost Model Considering Reliability and Time of Software in Use", Journal of Convergence Information Technology, Vol. 7, No. 13, pp. 135-142, 2012.
DOI : <https://doi.org/10.4156/jcit.vol7.issue13.16>
- [13] R. Satya 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, pp. 1-5, 2011.
DOI : <https://doi.org/10.5120/2527-3440>

- [14] H. C. Kim and H. K. Park, "The Comparative Study of Software Optimal Release Time Based on Weibull Distribution Property", Journal of the Korea Academia-Industrial cooperation Society, Vol. 10, Issue. 8, pp. 1903-1901, 2009.
DOI : <https://doi.org/10.5762/KAIS.2009.10.8.1903>
- [15] T. J. Yang, "A Performance Comparative Evaluation for Finite and Infinite Failure Software Reliability Model using the Erlang Distribution", The Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 9, No. 4, pp. 351-358, 2016.
DOI : <https://doi.org/10.17661/jkiict.2016.9.4.351>
- [16] S. U. Lee and M, B, Choi, "An Establishment of the Process System for Software Requirements Engineering", The Journal of IIBC, Vol. 14, No. 1, pp. 37-49, 2014.
DOI : <http://dx.doi.org/10.7236/JIIBC.2014.14.1.37>

저 자 소 개

배 효 정(정회원)



- 2004년 : 부경대 지능기계공학 석사
- 2017년 : 부경대 지능기계공학 박사
- 1992 ~ 2020년 : (재)포항산업과학연구원(RIST)
- 2021년 ~ 현재 : 남서울대 드론공간 정보공학과 교수. 주관심 분야는 지능제어, 지능 소프트웨어공학, AI 등

※ 이 논문은 2021년도 남서울대학교 학술연구비 지원에 의해 연구되었음.