

극값 분포 특성을 가진 소프트웨어 신뢰성 보증 모형에 관한 비교연구

정회원 김희철*

The Comparative Study of the Warranty Cost Model for Software Reliability Time Based on Extreme Value Distribution

Hee-cheul Kim* *Regular Member*

요약

본 연구에서는 소프트웨어 제품을 개발하는 과정에서 보증기간을 고려하여 사용자에게 인도하는 시기를 결정하는 방출시기에 관한 모형의 비교 연구이다. 실증 비교 결과 지수화 소프트웨어 신뢰성장모형에서는 보증기간이 증가함에 따라 초기에는 최적방출시기가 증가했으나 점차 거의 유사한 방출시기를 나타내고 있으며 이와 반대로 불완전한 디버깅 소프트웨어 신뢰성장 모형에서는 초기에는 최적방출시기가 감소하였으나 보증기간이 길수록 소폭 증가하는 패턴을 보이고 있으며 제안된 극값모형은 점차적으로 증가하는 형태를 가진다는 결론을 도출 할 수 있다. 이러한 연구결과를 통하여 소프트웨어 개발자들은 보증기간과 방출최적시기의 관계를 파악함으로써 소프트웨어 개발 시 사전정보로 활용 할 수 있으리라 사료 된다.

Key Words : Software Release Policies; Total Average Cost; Extreme Value Distribution.

ABSTRACT

In this research, the process of developing software products to users in transfer by considering the warranty period to determine the timing of the release period is a comparative study of models. For the results of demonstration, exponential software reliability model increases the warranty period, the higher the initial period, but shows almost a similar release. In contrast, the optimal release time of imperfect debugging software reliability model, lower the initial warranty period, but the pattern is expected to rise slightly larger. The proposed model, extreme value distribution model, pattern of the optimal release time gradually increase, have a form that can be drawn. These research results through, warranty period and release the software developers understand the relationship between the optimal time for software development by using advance information could do is feed.

I. 서론

소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 유발 할 수 도 있다. 따라서 소프트웨어 신뢰도는 현대의 소프트웨어 생산품 개발에서 중요한 분야 가운데 하나이다. 모든

소프트웨어 회사들은 소비자들에게 오류가 없는 제품을 방출하기를 기대하지만 점점 소프트웨어 제품이 복잡해짐에 따라 신뢰도가 높은 제품을 생산하는데 어려움이 점점 가중되어 가고 있다. 그러므로 신뢰도, 비용, 방출시기 면에서 소프트웨어 제품에 대한 관리의 필요성이 제기된다.

※ 이 논문은 2008학년도 남서울대학교 학술연구비 지원에 의하여 연구되었음

* 남서울대학교 산업경영공학과(kim1458@nsu.ac.kr)

논문번호 : KICS2009-04-160, 접수일자 : 2009년 4월 16일, 최종논문접수일자 : 2009년 6월 5일

소프트웨어 개발과정은 설계단계, 디자인, 코딩 그리고 테스트 단계를 거친다. 이러한 과정을 거친 후 소프트웨어 제품을 방출하게 되는데 방출이후에 발견되지 않은 고장들이 나타난다면 이것들에 대한 보전 비용(Maintenance cost)은 크게 증가 할 것이다. 결국, 소프트웨어 시스템 시험을 끝내고 그것을 사용자에게 넘기는 시기 결정은 매우 중요한 사항이 된다. 이러한 소프트웨어 방출시간에 대한 연구들은 대부분 유한 고장 NHPP(Non-Homogeneous Poisson Process)모형을 사용하였다.

이 분야에서는 많은 연구가 이루어졌는데 주요 연구 결과들을 살펴보면 Yang과 Xie.^[1]는 소프트웨어 평가모형에 대해 운영 및 테스트 신뢰도에 대하여 연구하였고 Tal 과 McCollin., Ben Dell^[2]은 소프트웨어 신뢰도를 이용한 통계적인 기준으로 최적 테스트 기준에 대하여 논의되었으며 Popstojanova와 Trivedi^[3]는 소프트웨어 시스템에 대한 신뢰도 평가 모형에 대한 구조적 측면을 연구하였다. 그러나 이러한 유한(finite)고장 NHPP모형은 소프트웨어가 유한개의 고장이 있고 고장 제거 단계에서는 새로운 고장이 발생하지 않는다는 가정을 한 모형이다. 그러나 실제 고장 제거 단계에서도 새로운 고장이 발생할 수 있다. 최근에 이와 관련된 문제에 대한 연구는 Huang^[4]과 김희철^[5]에 의해 연구되고 있다. 이러한 연구 분야를 확장한 보증기간 문제에 대한 연구도 수행되었는데, Yamada와 Kimura, Terane, Osaki.^[6]은 라이프 싸이클 분포 및 할인율을 적용한 소프트웨어 최적 방출문제에 대한 연구를 시행하였고 Prince Williams와 Vivekanandan^[7]은 할인율을 가진 소프트웨어 신뢰도에 대한 장기간에 이루어지는 보증 비용모형에 대하여 연구되었고 Pham^[8]은 임의의 라이프 사이클 과 패널티(penalty) 비용에 대한 불완전한 디버깅 속성을 가진 소프트웨어 보증 비용 모형에 대하여 연구되었다.

따라서 본 연구에서는 여러 분포 중에서 표본의 최대값 혹은 최소값을 찾는 데 사용되는 분포 극값 분포^[11]을 적용한 유한고장 NHPP 모형을 이용하여 보증기간을 고려한 최적 방출시기에 관한 문제를 다루고자 한다. 본 논문의 2절에서는 관련 연구로서 무한고장 NHPP 과 보증비용 및 신뢰도 모형 그리고 기준모형에 대하여 약술하였고 3장에는 제안된 극값 비용 모형에 대하여 서술하고 4장과 5장에서는 각각 수치적인 예와 그 결론을 나열 하였다.

II. 관련연구

2.1 무한고장 NHPP

NHPP 모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도 함수(Intensity function) $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다^[9, 10, 11].

$$m(t) = \int_0^t \lambda(s)ds, \quad \frac{dm(t)}{dt} = \lambda(t) \quad (1)$$

따라서 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률 밀도 함수(Probability density function; Pdf)로 알려져 있다. 즉,

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, \quad n = 0, 1, 2, \dots, \infty \quad (2)$$

이처럼 시간 관련 모형(Time domain models) 들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이러한 모형들은 고장 강도 함수 $\lambda(t)$ 가 다르게 표현됨으로서 평균값 함수 $m(t)$ 도 역시 다르게 나타나고 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다^[11,12]. 유한 고장 NHPP 모형들은 충분한 테스트 시간이 주어지면 결함들(Faults)의 기대값이 유한 값($\lim_{t \rightarrow \infty} m(t) = \theta < \infty$)을 가지고 반면에 무한 고장 NHPP 모형들은 무한 값을 가진다고 가정되고 유한 고장 모형 모형은 각 수리 시점에서 새로운 결함이 발생하지 않는다는 가정을 한다. 그러나 실제 상황에서는 수리 시점에서도 고장이 발생할 수도 있다. 이러한 상황을 추가하기 위한 무한 고장 NHPP에 근거한 평균값함수는 다음과 같이 된다고 하였다^[11].

$$e^{-m(t)} = 1 - F(t) \quad (3)$$

따라서 (1)식 과 (3)식과 연관시키고 $f(t)$ 을 확률밀도 함수, $F(t)$ 을 분포함수라고 하면 무한 고장 NHPP 모형의 평균값 함수와 고장 강도 함수는 각각 다음과 같이 유도된다.

$$m(t) = -\ln(1 - F(t)) \quad (4)$$

$$\lambda(t) = m'(t) = f(t)/(1 - F(t)) \quad (5)$$

시간 $(0, t]$ 까지 조사하기 위한 시간 절단(Time truncated) 모형은 n 번째 까지 고장 시점 자료를

$$x_k = \sum_{i=1}^k t_k \quad (k = 1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (6)$$

이라고 하고 θ 을 모수공간이라고 하면 우도 함수는 다음과 같이 알려져 있다¹²⁾.

$$L_{NHPP_{NPN}}(\theta | D_t) = \prod_{i=1}^n \left(\frac{f(x_i)}{1-F(x_i)} \right) (1-F(x_n))$$

$$= \left(\prod_{i=1}^n \lambda(x_i) \right) (1-F(x_n)) \quad (7)$$

$$= \left(\prod_{i=1}^n \lambda(x_i) \right) e^{-m(t_n)}$$

2.2 보증비용모형

소프트웨어 개발 비용을 추정하고자 할 때 소프트웨어 판매이후에 일어나는 비용인 보증비용(Warranty cost)도 고려해야 한다^{13,14)}. 이러한 보증 비용은 소프트웨어 방출시기에 의존하게 되며 그림 1과 같이 표현 된다.

본 연구에서 사용되는 소프트웨어 유지 비용 요인들의 관계는 다음과 같은 식으로 사용된다^{13, 14)}.

$$WC(T) = c_0 + c_t \int_0^T e^{-\alpha t} dt + C_w(T) \quad (8)$$

- 단, $WC(T)$: 소프트웨어 총 기대유지 비용
- c_0 : 최초 테스트 비용(최소한의 요구조건)
- c_t : 단위 시간당 테스트 비용
- T : 소프트웨어방출시간(추가적 총 테스트 시간)
- α : 총 소프트웨어 비용에 대한 할인율

(8)식에 있는 $C_w(T)$ 은 보증기간동안 유지비용을 나타내는데 이것은 2가지 경우로 구분해 볼 수 있다. 보증기간이 상수이고 소프트웨어 테스트 이후에 소프트웨어 신뢰성장이 이루어지지 않은 경우와 보증

기간이 상수이고 소프트웨어 테스트 이후에 소프트웨어 신뢰성장이 이루어지는 경우로 구분해 볼 수 있는데 본 연구에서는 신뢰성장이 이루어지는 경우를 연구의 대상으로 시행 하였다(그림1).

따라서 $C_w(T)$ 은 다음과 같이 정의 할 수 있다¹³⁾.

$$C_w(T) = c_w \int_T^{T+T_w} \lambda(t) e^{-\alpha t} dt \quad (10)$$

- 단, T_w : 소프트웨어 보증기간
- c_w : 보증기간동안 결함 당 유지 비용

2.3 요구 신뢰도와 최적방출시점

NHPP 모형에서 테스트 시점 T 이후에 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간 $(T, T+x]$ (단, x 는 임무시간(Mission time)) 동안 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도 $\hat{R}(x | T)$ 는 다음과 같이 됨이 알려져 있다^{12,13)}.

$$\hat{R}(x | T) = \exp[-\{m(x+T) - m(T)\}] \quad (11)$$

R_0 을 원하는 신뢰성 수준이라고 가정하면 최적 방출문제는 $\hat{R}(x | T) \geq R_0$ 을 만족시키면서 동시에 $WC(T)$ 을 최소화 시키는 방출시기가 최적방출시점이 된다¹⁵⁾.

2.4 기존모형

2.4.1 지수화 소프트웨어 신뢰성장모형

Kimura와 Toyota, Osaki¹⁴⁾는 결함 탐색 현상에 근거한 NHPP 지수화 소프트웨어 신뢰성장모형을 제안 하였다. 이 모형에서 사용된 평균값함수와 강도함수는 다음과 같다.

$$m(t) = a(1 - e^{-bt}), \quad \lambda(t) = abe^{-bt} \quad (12)$$

단, a 와 b 는 각각 초기결함기대수와 결함 당 탐색 비율을 의미한다.
다음과 같다¹⁴⁾.

$$WC(T) = c_0 + c_t \left(\frac{1 - e^{-\alpha T}}{\alpha} \right) + \frac{c_w ab(1 - e^{-(\alpha+b)T_w}) e^{-(\alpha+b)T}}{(\alpha+b)} \quad (13)$$

(13)을 T 에 관하여 미분하고 0으로 놓고 T 에 관하여 정리하면 방출시간은 다음과 같이 표현된다.

$$T = T_c = \frac{1}{b} \ln \left(\frac{ab(1 - e^{-(\alpha+b)T_w}) c_w}{c_t} \right) \quad (14)$$

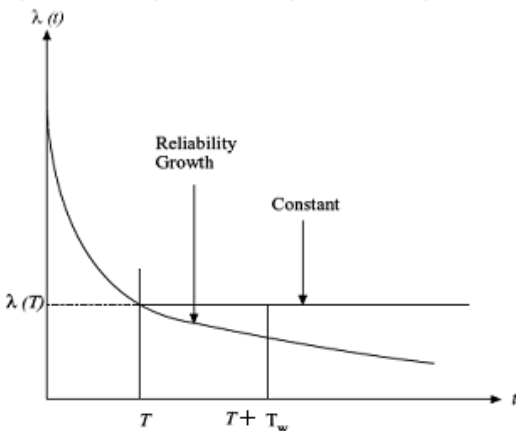


그림 1. 보증 기간에 대한 소프트웨어 신뢰성장 패턴
Fig. 1. Software reliability growth pattern during the warranty period

물론 (13)식은 $\frac{d^2 WC(T)}{dT^2} > 0$ 을 만족한다.

신뢰도 측면에서 $\hat{R}(x | T) = R_0$ 만족하는 방출시간을 T_R 이라고 하면 (11)식에 의해서 다음과 같이 표현된다.

$$\hat{R}(x | T) = \exp[-e^{-bT} m(t)] = R_0 \quad (15)$$

따라서 T_R 은 다음과 같이 유도된다.

$$T_R = T = \frac{1}{b} \left[\ln m(x) - \ln \ln \left(\frac{1}{R_0} \right) \right] \quad (16)$$

2.4.2 불완전한 디버깅 소프트웨어 신뢰성장 모형

이 모형은 Prince Williams^[13]이 제안 하였는데 앞 절에서의 지수화 소프트웨어 신뢰성장모형에서 불완전한 디버깅 확률 p 을 삽입하여 확장한 모형인데 평균값함수와 강도함수는 다음과 같다.

$$m(t) = \frac{a}{p} (1 - e^{-bpt}), \quad \lambda(t) = abe^{-bpt} \quad (17)$$

그리고 유사한 방법으로 T 에 관하여 정리한 방출시간은 다음과 같이 표현된다^[13].

$$T = T_c = \frac{1}{bp} \ln \left(\frac{ab(1 - e^{-\alpha T_w}) c_w}{c_t} \right) \quad (18)$$

그리고 앞에서와 유사한 방법으로 T_R 은 다음과 같이 표현된다.

$$T_R = T = \frac{1}{bp} \left[\ln m(x) - \ln \ln \left(\frac{1}{R_0} \right) \right] \quad (19)$$

T_{op} 가 최적 방출시기라고 표현하면 최적 방출 모형은 다음과 같이 구성된다^[13, 14].

P : $\lambda(0) > \lambda(T_c)$ 이고 $R(x|0) < R_0$ 을 만족하면

$$T_{op} = \max \{ T_c, T_R \}$$

P.1 : $\lambda(0) > \lambda(T_c)$ 이고 $R(x|0) \geq R_0$ 을 만족하면

$$T^* = T_c$$

P.2 : $\lambda(0) \leq \lambda(T_c)$ 이고 $R(x|0) < R_0$ 을 만족하면

$$T_{op} = T_R$$

P.3 : $\lambda(0) \leq \lambda(T_c)$ 이고 $R(x|0) \geq R_0$ 을 만족하면

$$T_{op} = 0$$

III. 제안된 극값 모형

이 절에서 극값 분포(Extreme value distribution)^[11] 모형에 대하여 제안 하고자 한다. 이 극값 분포는 Gumbel 분포라고도 하는데 여러 표본 중에서 최대값 혹은 최소값을 찾는 데 사용되는 분포이고 기후 혹은 날씨를 모형화 할 때 많이 사용되는 분포로 알려져 있다. 이 극값 분포의 확률 밀도 함수와 분포 함수는 각각 다음과 같다^[11].

$$f(t | \beta_0, \beta_1) = \exp \left(\beta_0 + \beta_1 t - \frac{e^{\beta_0 + \beta_1 t} - e^{\beta_0}}{\beta_1} \right) \quad (20)$$

단, 모수 $\beta_0, \beta_1 > 0, t \geq 0$.

$$F(t | \beta_0, \beta_1) = 1 - \exp \left(- \frac{e^{\beta_0 + \beta_1 t} - e^{\beta_0}}{\beta_1} \right) \quad (21)$$

(4)식과 (5)식을 이용하여 무한 NHPP로 접근하면 평균값 함수와 강도 함수는 다음과 같이 표현 할 수 있다.

$$m(t) = -\ln(1 - F(t)) = \frac{e^{\beta_0 + \beta_1 t} - e^{\beta_0}}{\beta_1} \quad (22)$$

$$\lambda(t) = m'(t) = f(t)/(1 - F(t)) = \exp(\beta_0 + \beta_1 t) \quad (23)$$

즉, (10)식을 이용하면 $C_w(T)$ 는 다음과 같이 표현된다.

$$C_w(T) = c_w \int_T^{T+T_w} \lambda(t) e^{-\alpha t} dt = \frac{c_w}{\beta_1 - \alpha} e^{(\beta_0 + T(\beta_1 - \alpha))} (e^{(\beta_1 - \alpha)T_w} + 1) \quad (24)$$

따라서 소프트웨어 총 기대유지 비용은 다음과 같이 유도된다.

$$WC(T) = c_0 + c_t \left(\frac{1 - e^{-\alpha T}}{\alpha} \right) + \frac{c_w}{\beta_1 - \alpha} e^{(\beta_0 + T(\beta_1 - \alpha))} (e^{(\beta_1 - \alpha)T_w} - 1) \quad (25)$$

(25)을 T 에 관하여 미분하고 0으로 놓고 T 에 관하여 정리하면 방출시간은 다음과 같이 표현된다.

$$T = T_c = \frac{\ln \frac{c_t}{c_w} + (\beta_1 - \alpha)T_w}{\alpha} \quad (26)$$

물론 (25)식은 $\frac{d^2 WC(T)}{dT^2} > 0$ 을 만족하고 신뢰도 측면

에서 $\hat{R}(x | T) = R_0$ 만족하는 방출시간을 T_R 이라고 하면 (11)식에 의해서 다음과 같이 표현된다.

$$T_R = T = \frac{1}{\beta_1} \left[\ln \left(\frac{\beta_1 \ln R_0}{1 - e^{-\beta_1 x}} \right) - \beta_0 \right] \quad (27)$$

앞 절과 유사하게 T_{op} 가 최적 방출시기라고 표현하면 최적 방출 모형은 다음과 같이 구성된다.

P : $\lambda(0) > \lambda(T_c)$ 이고 $R(x|0) < R_0$ 을 만족하면

$$T_{op} = \max \{ T_c, T_R \}$$

P.1 : $\lambda(0) > \lambda(T_c)$ 이고 $R(x|0) \geq R_0$ 을 만족하면

$$T^* = T_c$$

P.2 : $\lambda(0) \leq \lambda(T_c)$ 이고 $R(x|0) < R_0$ 을 만족하면

$$T_{op} = T_R$$

P.3 : $\lambda(0) \leq \lambda(T_c)$ 이고 $R(x|0) \geq R_0$ 을 만족하면

$$T_{op} = 0$$

IV. 수치적인 예

수치적인 예에서 지수화 소프트웨어 신뢰성장모형과 불완전한 디버깅 소프트웨어 신뢰성장 모형의 c_0 (최초 테스트 비용(최소한의 요구조건))을1000, c_w (보증기간동안 결함 당 유지 비용)을 30, 모수 $a=200$ 과 $b=0.03$, $p=0.5$ (불완전한 디버깅 확률), $\alpha=0.001$ (총 소프트웨어 비용에 대한 할인율)을 가정하고 여러 값에 대한 c_t (단위 시간당 테스트 비용)과 T_w (소프트웨어 보증기간)의 관계를 이용한 최적 방출시기가 계산되었다. 그리고 제안된 극값 분포모형에서의 모수는 $c_w=0.9$, $\beta_0=-10$, $\beta_1=0.05$ 그리고 다른 조건은 동일하여 적용 시켰을 때 방출시기를 계산하여 비교되었다.

표 1. 지수화 소프트웨어 신뢰성장모형에 대한 방출 시간
Table 1. Release time of exponential software reliability model

T_w/c_t	1	5	10	20	30	40
1	263.37	209.72	186.61	163.51	149.99	140.40
5	77.58	85.17	89.07	93.47	96.35	98.55
10	118.32	115.21	113.72	112.11	111.10	110.35
20	104.25	105.14	105.58	106.05	106.35	106.58
30	108.47	108.19	108.05	107.90	107.81	107.74
40	107.15	107.24	107.28	107.33	107.36	107.38
50	107.56	107.53	107.52	107.50	107.50	107.49
100	107.43	107.44	107.44	107.45	107.45	107.45

표 2. 불완전한 디버깅 소프트웨어 신뢰성장 모형에 대한 방출 시간
Table 2. Release time of imperfect debugging software reliability phenomenon model

T_w/c_t	1	5	10	20	30	40
1	27.97	109.30	139.29	117.13	90.09	64.52
5	104.51	170.28	158.99	122.97	92.27	65.61
10	137.40	182.54	162.01	123.80	92.57	65.76
20	170.17	191.96	164.19	124.39	92.78	65.86
30	189.24	196.58	165.22	124.66	92.88	65.91
40	202.70	199.54	165.86	124.83	92.94	65.94
50	213.09	201.69	166.32	124.95	92.99	65.96
100	244.92	207.59	167.56	125.28	93.10	66.02

표 3. 극값 분포모형에 대한 방출시간
Table 3. Release time of Extreme value distribution

T_w/c_t	1	5	10	20	30	40
1	4.18	57.83	80.93	104.04	117.55	127.14
5	6.85	60.49	83.60	106.70	120.22	129.81
10	10.18	63.83	86.93	110.04	123.55	133.14
20	16.85	70.49	93.60	116.70	130.22	139.81
30	23.51	77.16	100.26	123.37	136.89	146.47
40	30.18	83.83	106.93	130.04	143.55	153.14
50	36.85	90.49	113.60	136.70	150.22	159.81
100	70.18	123.83	146.93	170.04	183.55	193.14

표 1과 표 2에 요약된 지수화 소프트웨어 신뢰성장모형과 불완전한 디버깅 소프트웨어 신뢰성장 모형에서는 단위 시간당 테스트 비용(c_t)이 고정 되었을 때 소프트웨어 보증기간(T_w)이 증가 할수록 방출시간은 늘어나고 반대로 소프트웨어 보증기간(T_w)이 고정 되었을 때 단위 시간당 테스트 비용(c_t)이 증가 할수록 방출시간은 감소되고 있음을 알 수 있다.

표 3에 요약된 제안된 극값 분포모형에서는 단위 시간당 테스트 비용(c_t)이 고정 되었을 때 소프트웨어 보증기간(T_w)이 증가 할수록 방출시간은 늘어나지만 반대로 소프트웨어 보증기간(T_w)이 고정 되었을 때 단위 시간당 테스트 비용(c_t)이 증가 할수록 방출시간은 증가하고 있음을 알 수 있다.

그리고 임무시간(Mission time) $x=10$ 과 단위 시간당 테스트 비용 $c_t=10$, 신뢰성 수준 $R_0=0.95$ 를 가정하고 최적 방출시간을 요약한 내용은 표 4에 나타내었다. 이 표에서 신뢰도를 고려한 방출시간 T_R 에 불완전한 디버깅 소프트웨어 신뢰성장 모형은 영향을 주지 못하지만

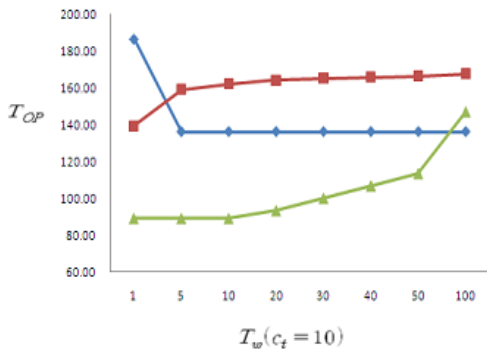
표 4. 각 모형에 대한 최적방출시간 계산 $c_t = 10, x = 10$
 Table 4. Calculation for optimal release time of each model

T_w	T_C	T_{OP1}	T_C	T_{OP2}	T_C	T_{OP3}
1	186.61	186.61	139.29	139.29	80.93	89.34
5	89.07	136.18	158.99	158.99	83.60	89.34
10	113.72	136.18	162.01	162.01	86.93	89.34
20	105.58	136.18	164.19	164.19	93.60	93.60
30	108.05	136.18	165.22	165.22	100.26	100.26
40	107.28	136.18	165.86	165.86	106.93	106.93
50	107.52	136.18	166.32	166.32	113.60	113.60
100	107.44	136.18	167.56	167.56	146.93	146.93
T_R	136.18		87.81		89.34	

(주 T_{OP1} : Exponential software reliability model
 T_{OP2} : Imperfect debugging software reliability model
 T_{OP3} : Extreme value distribution model)

지수화 소프트웨어 신뢰성장모형은 T_w 가 1인 경우에만 영향을 준다. 그러나 제안된 극값 분포모형에서는 T_w 가 1, 5, 10인 경우에 영향을 준다. 따라서 기존의 모형보다 제안된 극값모형이 T_R 에 더 영향을 주기 때문에 보다 더 신뢰성 수준에 더 많은 관리가 필요하다고 할 수 있다.

그림 2에서는 $c_t = 10$ 경우 T_w 에 대한 최적 방출 시간 T_{OP} 와의 관계를 요약 하였다. 이 그림에서 지수화 소프트웨어 신뢰성장모형에서는 보증기간이 증가함에 따라 초기에는 높았으나 거의 유사한 방출시기를 나타내고 있으며 이와 반대로 불완전한 디버깅 소프트웨어 신뢰성장 모형에서는 초기에는 최적방출시기가 낮았



(주: —●— Exponential software reliability model
 —■— Imperfect debugging software reliability model
 —▲— Extreme value distribution model)

그림 2. 각 모형에 대한 최적방출시간 비교
 Fig. 2. Comparative of optimal release time of each model

으나 보증기간이 클수록 소폭 상승하는 패턴을 보이고 있으며 제안된 극값모형은 점차적으로 증가하는 형태를 보이고 있다.

V. 결 론

본 연구는 여러 표본 중에서 의 최대값 혹은 최소값을 찾는데 사용되는 분포이고 기후 혹은 날씨를 모형화 할 때 많이 사용되는 분포로 알려진 극값 분포를 적용한 무한고장 NHPP 모형을 이용하여 보증기간을 고려한 최적 방출시기에 관한 모형의 비교 문제를 알아보았다. 즉, 지수화 소프트웨어 신뢰성장모형과 불완전한 디버깅 소프트웨어 신뢰성장 모형에서는 단위 시간당 테스트 비용이 고정 되었을 때 소프트웨어 보증기간이 증가 할수록 방출시간은 늘어나고 반대로 소프트웨어 보증기간이 고정 되었을 때 단위 시간당 테스트 비용이 증가 할수록 방출시간은 감소되고 있음을 알 수 있다. 그러나 제안된 극값 분포모형에서는 단위 시간당 테스트 비용이 고정 되었을 때 소프트웨어 보증기간이 증가 할수록 방출시간은 늘어나지만 반대로 소프트웨어 보증기간이 고정 되었을 때 단위 시간당 테스트 비용이 증가 할수록 방출시간은 증가하고 있음을 알 수 있다. 결국 지수화 소프트웨어 신뢰성장모형에서는 보증기간이 증가함에 따라 초기에는 증가하였으나 거의 유사한 방출시기를 나타내고 있으며 이와 반대로 불완전한 디버깅 소프트웨어 신뢰성장 모형에서는 초기에는 최적방출시기가 감소하였으나 보증기간이 클수록 소폭 증가하는 패턴을 보이고 있으며 제안된 극값모형은 점차적으로 증가하는 형태를 보인다고 결론을 도출 할 수 있다. 이러한 연구결과를 통하여 소프트웨어 개발자들은 보증기간과 방출최적시기의 관계를 파악함으로써 소프트웨어 개발 시 사전정보로 활용하는데 어느 정도 도움을 줄 수 있으리라 사료 된다.

참 고 문 헌

- [1] Yang, B. and Xie. M. "A study of operational and testing reliability in software reliability analysis". *Reliability Engineering and System Safety*, Vol, 70, pp,323-329, June 2000.
- [2] Tal, U, and McCollin,C. and Ben Dell, A.. "An optimal statistical testing policy for software reliability", *Demonstration of safety critical system*, Vol 137(3), pp. 544-557, 2002.

- [3] Popstojanova, K. G. and Trivedi, K. S. "Architecture-based approach to reliability assessment of software systems", *Performance Evaluation* Vol. 45, pp.179-204, 2001.
- [4] Huang, C. Y. "Cost-Reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency, *The Journal of Systems and Software*. Vol, 77, pp, 139-155, 2005.
- [5] 김희철. "Log-Logistic 분포모형에 근거한 소프트웨어 최적방출시기에 관한 비교 연구", *한국컴퓨터정보학회논문지*, 제13권 7호, pp.1-9, 2008.
- [6] Yamada, S and Kimura, M. and Terane, E., Osaki, S. "Optimal software release problems with life cycle distribution and discount rate", *Trans. IPS Japan*, Vol. 34(5), pp. 1188-1197, 1993(in Japanese).
- [7] Prince Williams, D. R. and Vivekanandan, P. "Life-time warranty cost model for software reliability with discount rate", *Journal of Computer Science*, Vol. 45, pp.53-59, 2005.
- [8] Pham, H. "A software cost model with imperfect debugging, random life cycle and penalty cost", *Int. J. System Sci.* Vol. 27, pp.455-463, 2003.
- [9] 김희철, "Burr 커버리지 함수에 기초한 ENHPP 소프트웨어 신뢰성장모형에 관한 연구", *한국컴퓨터정보학회논문지*, 12권 4호, pp. 33-42, 2007.
- [10] 김희철. "지수화 지수 분포에 의존한 NHPP 소프트웨어 신뢰성장 모형에 관한 연구", *한국컴퓨터정보학회논문지*, 제11권 5호, pp. 9-18, 2006.
- [11] Kuo, L. and Yang, T. Y. "Bayesian Computation of Software Reliability". *Journal of the American Statistical Association*, Vol.91, pp.763-773, June, 1996.
- [12] 김희철. "일반화감마분포를 이용한 NHPP 소프트웨어 신뢰도 모형에 관한 연구", *한국컴퓨터정보학회 논문지*, 10권 6호, pp. 27-35, 2005.
- [13] Prince Williams, D. R. "Study of the warranty cost model for software reliability with an imperfect debugging phenomenon", *Turk J Engin*, Vol.1,15(3), pp.369-381, 2007.
- [14] Kimura, M. and Toyota. T., Osaki, S. "Economic analysis of software release problems with warranty cost and reliability requirement", *Reliability Engineering and System Safety*, 66, pp.49-55, 1999.

김희철 (Heel-cheul Kim)

정회원



1992년 동국대학교 통계학과 석사
 1998년 동국대학교 통계학과 박사
 2000년 3월~2004년 2월 송호
 대학 정보산업계열 조교수
 2005년 3월~현재 남서울대학교
 산업경영공학과 전임강사
 <관심분야> 소프트웨어 신뢰성
 공학, 웹 프로그래밍, 전산통계