

Musa-Okumoto와 Power-law형 NHPP 소프트웨어 신뢰모형에 관한 통계적 공정관리 접근방법 비교연구

김희철*

The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Musa-Okumo and Power-law Type

Hee-Cheul Kim*

요약 소프트웨어의 디버깅과정에서 오류 발생의 시간을 기반으로 하는 많은 소프트웨어 신뢰성 모델이 제안되어 왔다. 무한고장 모형과 비동질적인 포아송 과정에 의존한 소프트웨어 신뢰성 모형을 이용하면 모수 추정이 가능하다. 소프트웨어를 시장에 인도하는 결정을 내리기 위해서는 조건부 고장률이 중요한 변수가 된다. 무한 고장 모형은 실제 상황에서 다양한 분야에 사용된다. 특성화 문제, 특이점의 감지, 선형 추정, 시스템의 안정성 연구, 수명을 테스트, 생존 분석, 데이터 압축 및 기타 여러 분야에서의 사용이 점점 많아지고 있다. 통계적 공정 관리 (SPC)는 소프트웨어 고장의 예측을 모니터링 함으로써 소프트웨어 신뢰성의 향상에 크게 기여 할 수 있다. 컨트롤 차트는 널리 소프트웨어 산업의 소프트웨어 공정 관리에 사용되는 도구이다. 본 논문에서 NHPP에 근원을 둔 로그 포아송 실행시간 모형, 즉, Musa-Okumo 모형과 파워어 로우(Power-law) 모형의 평균값 함수를 이용한 통계적 공정관리 차트를 이용한 제어 메커니즘을 제안하였다.

Abstract There are many software reliability models that are based on the times of occurrences of errors in the debugging of software. It is shown that it is possible to do likelihood inference for software reliability models based on finite failure model and non-homogeneous Poisson Processes (NHPP). For someone making a decision about when to market software, the conditional failure rate is an important variables. The infinite failure model are used in a wide variety of practical situations. Their use in characterization problems, detection of outlier, linear estimation, study of system reliability, life-testing, survival analysis, data compression and many other fields can be seen from the many study. Statistical process control (SPC) can monitor the forecasting of software failure and thereby contribute significantly to the improvement of software reliability. Control charts are widely used for software process control in the software industry. In this paper, proposed a control mechanism based on NHPP using mean value function of Musa-Okumo and Power law type property.

Key Words : Musa-Okumoto Model, Infinite Model, Mission Time, NHPP, Estimation

1. 서론

정기간동안 고장이 나지 않고 운영 할 수 있는 확률이다.

소프트웨어 신뢰성은 일정한 환경조건에서 일

따라서 소프트웨어 개발과정에서 소프트웨어

This paper studies under a 2015 Namseoul University Academic Research Support

*Corresponding Author : Department of Industrial & Management Engineering, Namseoul University(kim1458@nsu.ac.kr)

Received November 26, 2015

Revised December 5, 2015

Accepted December 13, 2015

신뢰성은 중요한 문제이다. 이 문제는 사용자의 요구조건과 테스트 비용을 만족시켜야 한다. 소프트웨어 테스트(디버깅)면에서 비용을 줄이기 위해서는 소프트웨어의 신뢰성의 변동과 테스트 비용을 사전에 알고 있어야 효율적이다. 따라서 신뢰도, 비용 및 방출시간의 고려사항을 가진 소프트웨어 개발 과정은 필수 불가결 하다.

지금까지 많은 소프트웨어 신뢰성모형이 제안되었다. 이 중에서 비동질적 포아송 과정(Non-homogeneous Poisson process; NHPP)에 의존한 모형[1]은 에리탐색 과정측면에서는 우수한 모형이고 이러한 모형은 결함이 발생하면 즉시 제거되고 디버깅과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

이 분야에서 통계적 공정관리 (SPC; Statistical process control)는 소프트웨어 고장의 예측을 모니터링 함으로써 소프트웨어 신뢰성의 향상에 크게 기여 할 수 있다. 컨트롤 차트(관리도)는 널리 소프트웨어 산업의 소프트웨어 공정관리에 사용되는 도구이다. 이 분야에서 Rao, Prasad 와 Kantham는 Half Logistics 분포를 이용한 공정관리 차트를 이용한 메카니즘을 연구하기도 하였다 [2]. 또한 김희철[3]은 레일리형과 버르형 NHPP 소프트웨어 신뢰모형에 관한 통계적 공정관리 접근방법 비교연구 결과 레일리분포모형 보다는 버르분포모형이 상대적으로 효율적 모형임을 확인하였다.

따라서 본 연구에서는 Musa-Okumo모형과 Power-law형 패턴을 적용한 무한고장 NHPP 소프트웨어 모형에 대한 공정관리 접근방법을 비교 제시하였다.

2. 관련연구

2.1 Musa-Okumo 와 Power-law 무한고장 NHPP 모형

Musa-Okumoto 모형의 평균값 함수는 다음과 같이 로그형 함수로 알려져 있다[4, 5].

$$m_{M-O}(t) = a \ln(1 + bt) \tag{1}$$

단, $t \in [0, \infty]$ 이고 $a > 0$ 는 척도모수(Scale parameter)이고 $b > 0$ 는 형상모수(Shape parameter)를 의미하고 (1)식을 이용한 강도함수는 다음과 같이 유도되고 무한고장의 모형인 경우 강도함수는 위험함수와 같은 형태가 된다[3].

$$\lambda_{M-O}(t) = m'_{M-O}(t) = \frac{ab}{(1+bt)} = h(t) \tag{2}$$

이모형에 대한 이용하면 우도함수는 다음과 같이 유도된다[3].

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp(-m(x_n)) \\ = \left(\prod_{i=1}^n \frac{ab}{(1+bx_i)} \right) \exp[-a \ln(1+bx_n)] \tag{3}$$

단, θ 는 모수공간이고

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

그리고 최우추정법을 이용하기 위한 로그우도 함수는 (3)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln a + n \ln b - \sum_{i=1}^n \ln(1+bx_i) - a \ln(1+bx_n) \tag{4}$$

(4)식에서 a 와 b 에 대하여 편미분 하여 다음과 같은 식을 만족하는 \hat{a}_{MLE} 와 \hat{b}_{MLE} 을 수치 해석적 방법으로 계산할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial a} = \frac{n}{a} - \ln(1+bx_n) = 0 \tag{5}$$

$$\cong, \hat{a} = \frac{n}{\ln(1+\hat{b}x_n)}$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial b} = \frac{n}{\hat{b}} - \sum_{i=1}^n \frac{x_i}{1+\hat{b}x_i} - \frac{\hat{a}\hat{b}}{1+\hat{b}x_n} = 0 \tag{6}$$

무한고장 NHPP모형에서 평균값 함수 $m(t) = -\ln(1-F(t))$ 을 이용하면 분포함수는 다음과 같이 유도 할 수 있다[3, 6].

$$F(t) = 1 - e^{-m(t)} = 1 - e^{-a \ln(1+bt)} \quad (7)$$

Power-law 모형의 평균값 함수는 다음과 같은 함수로 알려져 있다 [7].

$$m_{P-L}(t) = \alpha t^\beta \quad (8)$$

단, $t \in [0, \infty]$ 이고 $\alpha > 0$ 는 척도모수(Scale parameter)이고 $\beta > 0$ 는 형상모수(Shape parameter)를 의미하고 (8)식을 이용한 강도함수(위험함수)는 다음과 같이 유도된다.

$$\lambda_{P-L}(t) = m'_{P-L}(t) = \alpha \beta t^{\beta-1} = h(t) \quad (9)$$

(8)식과 (9)식을 이용하면 우도함수는 다음과 같이 유도된다.

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \alpha \beta x_i^{\beta-1} \right) \exp[-\alpha x_n^\beta] \quad (10)$$

단, θ 는 모수공간이고

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

최우추정법을 이용하기 위한 로그우도함수는 (10)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta | \underline{x}) = n \ln \alpha + n \ln \beta - (\beta - 1) \sum_{i=1}^n \ln x_i - \alpha x_n^\beta \quad (11)$$

(11)식에서 α 와 β 에 대하여 편미분 하여 다음과 같은 식을 만족하는 $\hat{\alpha}_{MLE}$ 와 $\hat{\beta}_{MLE}$ 을 수치 해석적 방법으로 계산할 수 있다.

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \alpha} = \frac{n}{\alpha} - x_n^\beta = 0 \quad (12)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^n \ln x_i - \alpha x_n^\beta \ln x_n = 0 \quad (13)$$

그리고 $m(t) = -\ln(1-F(t))$ 을 이용하면 분포 함수는 다음과 같이 유도 할 수 있다[3].

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

3.제안된 Musa-Okumoto 및 Power-law NHPP 소프트웨어 신뢰모형에 관한 통계적 공정관리 접근 방법

공정이 안정된 상태에 있는가를 조사하기 위해, 또는 공정을 안정된 상태로 유지하기 위해 활용 하는 도표가 관리도이다. 관리도의 관리한계는 관리상한(UCL; Upper Control Limit), 중심선(CL; Control Limit), 관리하한(LCL; Low Control Limit)으로 구성된다. 즉, 관리한계를 벗어나면 공정의 이상을 발견하고 재발방지 조치를 취함으로써 공정을 안정하게 유지하여 품질의 유지 및 향상을 도모하기 위한 목적으로 이용된다.

소프트웨어 신뢰성 모형에 대한 누적함수($F(t)$)의 값을 각각 6-sigma 기준확률 0.99865, 0.00135, 0.5을 이용하면 관리상한(t_U), 관리하한(t_L) 중심선(t_C)을 추정할 수 있다고 하였다[2].

따라서 Musa-Okumoto 모형의 분포함수 (7)식을 이용하면 관리상한은 다음과 같은 관계로 정리 할 수 있다[2, 8].

$$F(t) = 1 - e^{-m(t)} = 1 - e^{-a \ln(1+bt)} = 0.99865 \quad (15)$$

(15)식을 t 에 관하여 정리하면 다음과 같이 일반적인 관리상한은 다음과 같이 유도된다[8].

$$t = \frac{1}{b} \left[\exp\left(\frac{-\ln(0.00135)}{a}\right) - 1 \right] = t_U \quad (16)$$

또한, 유사하게 중심선과 관리하한은 다음과 같이 유도된다.

$$t = \frac{1}{b} \left[\exp\left(\frac{-\ln(0.5)}{a}\right) - 1 \right] = t_C \quad (17)$$

$$t = \frac{1}{b} \left[\exp\left(\frac{-\ln(0.99865)}{a}\right) - 1 \right] = t_L \quad (18)$$

따라서 평균값함수의 차이를 이용한 관리 한계, 즉 관리상한($m_{M-O}(t_U)$), 중심선($m_{M-O}(t_C)$), 관리하한($m_{M-O}(t_L)$)은 각각 다음과 같이 유도된다[2, 8].

$$m_{M-O}(t_U) = a \ln(1 + b t_U) \quad (19)$$

$$m_{M-O}(t_C) = a \ln(1 + b t_C) \quad (20)$$

$$m_{M-O}(t_L) = a \ln(1 + b t_L) \quad (21)$$

유사한 방법으로 Power-law 모형의 분포함수 (14)식을 이용하면 관리상한은 다음과 같은 관계로 정리 할 수 있다.

$$F(t) = 1 - e^{-m(t)} = 1 - e^{-\alpha t^\beta} = 0.99865 \quad (22)$$

(22)식을 t 에 관하여 정리하면 다음과 같이 일반적인 관리상한은 다음과 같이 유도된다[2, 8].

$$t = \exp \left[\frac{\ln\left(\frac{-\ln(0.00135)}{\alpha}\right)}{\beta} \right] = t_U \quad (23)$$

또한, 유사하게 중심선과 관리하한은 다음과 같이 유도된다.

$$t = \exp \left[\frac{\ln\left(\frac{-\ln(0.5)}{\alpha}\right)}{\beta} \right] = t_C \quad (24)$$

$$t = \exp \left[\frac{\ln\left(\frac{-\ln(0.99865)}{\alpha}\right)}{\beta} \right] = t_L \quad (25)$$

따라서 평균값함수의 차이를 이용한 관리 한계, 즉 관리상한($m_{P-L}(t_U)$), 중심선($m_{P-L}(t_C)$), 관리하한($m_{P-L}(t_L)$)은 각각 다음과 같이 유도된다[2, 8]

$$m_{P-L}(t_U) = \alpha t_U^\beta \quad (26)$$

$$m_{P-L}(t_C) = \alpha t_C^\beta \quad (27)$$

$$m_{P-L}(t_L) = \alpha t_L^\beta \quad (28)$$

4. 소프트웨어 고장시간 공정 분석

표 1. 소프트웨어의 고장시간자료
Table 1. Failure time data of software

Failure number	Failure time (hours)	Failure number	Failure time (hours)
1	0.479	16	10.771
2	0.745	17	10.906
3	1.022	18	11.183
4	1.576	19	11.779
5	2.61	20	12.536
6	3.559	21	12.973
7	4.252	22	15.203
8	4.849	23	15.64
9	4.966	24	15.98
10	5.136	25	16.385
11	5.253	26	16.96
12	6.527	27	17.237
13	6.996	28	17.6
14	8.17	29	18.122
15	8.863	30	18.735

이 절에서는 소프트웨어 고장 시간자료[9] (Failure time data)를 이용하여 본 논문에서 제시하는 소프트웨어 신뢰모형들을 이용한 소프트웨어 고장시간을 분석하고자 한다. 이 자료의 고장시간은 18.735 시간단위에 30번의 고장이 발생된 자료이며 [표 1]에 나열 되어 있다.

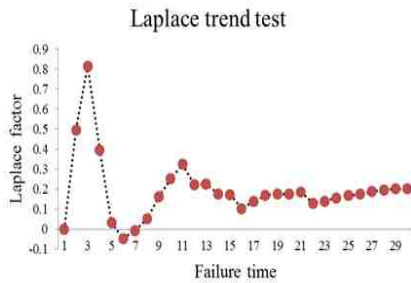


그림 1. 라플라스 추세검정
Fig. 1. Laplace trend test

또 한 제시하는 신뢰모형들을 분석하기 위하여 우선 자료에 대한 추세검정이 선행 되어야 한다[10]. 추세분석에는 일반적으로 라플라스 추세검정(Laplace trend test)을 사용한다. 이 검정을 실시한 결과 [그림 1]에서 라플라스요인(Factor)이 -2와 2사이에 존재함으로써 즉, 극단값(Extreme value)이 존재하지 않으므로 이 자료를 이용하여 신뢰성장모형을 제시하는 것이 효율적임을 시사하고 있다[3].

표 2. 각 모형에 대한 모수 추정 값과 관리한계 값
Table 2. Parameter estimation and control limites value for each model

model	MLE	control limits		
		$m(t_U)$	$m(t_C)$	$m(t_L)$
Musa-Okumoto	$\hat{a}_{MLE} = 21.3144$ $\hat{b}_{MLE} = 1.6471$	6.608	0.693	0.001
Power law	$\hat{\alpha}_{MLE} = 2.5250$ $\hat{\beta}_{MLE} = 3.9422$	34.575	3.323	2.526

Note. MLE : Maximum likelihood estimation.

모수추정은 최우추정법을 이용하고 모수추정을 용이하게 하기 위하여 원래의 고장시간 데이터를 변수변환($Failure\ time \times 10^{-1}$)하여 적용하였다. 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기 값을 0.01과 5을, 허용 한계 (Tolerance for width of interval)는 10^{-5} 을 주고

수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다. 최우추정법의 결과와 관리한계값들이 [표 2]에 요약되었다.

Musa-Okumoto모형의 경우 평균값함수의 차이를 이용한 관리상한은 (16)식과 (19)식과 모수 추정값을 이용하면 다음과 같이 계산된다.

$$t = \frac{1}{1.6471} \left[\exp\left(\frac{-\ln(0.00135)}{21.3144}\right) - 1 \right] = 0.22065 = t_U \tag{29}$$

$$m_{M-o}(t_U) = 21.3144 \times \ln(1 + 1.6471 \times 0.22065) = 6.60765 \tag{30}$$

Power-law모형의 경우에서도 평균값함수의 차이를 이용한 관리상한은 (23)식과 (26)식과 모수의 추정값을 이용하면 다음과 같이 계산된다.

$$t = \exp \left[\frac{\ln\left(\frac{-\ln(0.00135)}{2.5250}\right)}{3.9422} \right] = 1.94219 = t_U \tag{31}$$

$$m_{P-L}(t_U) = 2.5250 \times 1.94219^{3.9422} = 34.57522 \tag{32}$$

유사한 방법을 동원하면 Musa-Okumoto모형과 Power-law모형의 중심선과 관리하한 값을 추정할 수 있다.

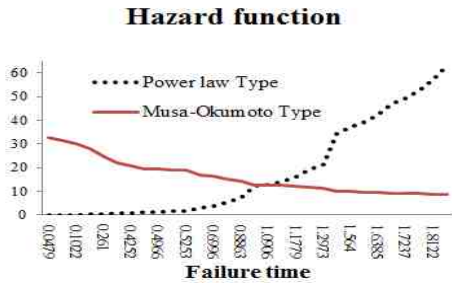


그림 2. 위험함수의 형태
Fig. 2. Pattern of hazard function

본 연구에서는 (2)와 (9)식을 이용하여 위험함수(강도함수)가 [그림 2]에 요약되었다.

이 그림에서 Musa-Okumoto모형의 위험함수의 특성은 비증가패턴을 보이고 있고 반면에 Power-law 위험함수의 특성은 비감소패턴을 보이고 있다.

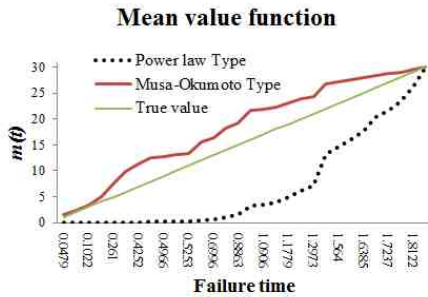


그림 3. 평균값 함수의 형태
Fig. 3. Pattern of mean value function

[그림 3]은 Musa-Okumoto모형과 Power law모형에 대한 평균값함수의 값을 나타내었다. 이 그림에서 Musa-Okumoto모형은 참값과의 차이에서 과대평가 추정이 이루어졌고 Power-law모형은 과소평가 추정 되었다.

표 3. 평균값에 대한 연속적인 차이
Table 3. Successive difference for mean value function

Failure number	Failure Time (t_i)	$m(i)$ $i = 1, 2, \dots, 30.$		$m(i+1) - m(i)$ $i = 1, 2, \dots, 29$ Successive difference	
		Musa-Okumoto	Power law	Musa-Okumoto	Power law
1	0.0479	1.6186	1.5×10^{-5}	0.8484	7.4
2	0.0745	2.4670	9.04×10^{-5}	0.8490	$\times 10^{-5}$
3	0.1022	3.3161	0.0003	1.6029	0.0002
4	0.1576	4.9190	0.0017	2.7031	0.0014
5	0.261	7.6220	0.0127	2.2112	0.0109
6	0.3559	9.8333	0.0430	1.4811	0.0303
7	0.4252	11.3144	0.0867	1.1983	0.0437
8	0.4849	12.5127	0.1456	0.2271	0.0588
9	0.4966	12.7398	0.1599	0.3258	0.0143
10	0.5136	13.0656	0.1826	0.2214	0.0227
11	0.5253	13.2870	0.1995	2.2724	0.0170
12	0.6527	15.5593	0.4697	0.7791	0.2702
13	0.6996	16.3384	0.6175	1.8338	0.1478
14	0.817	18.1721	1.1382	1.0127	0.5207
15	0.8863	19.1849	1.5690	2.5627	0.4308
16	1.0771	21.7476	3.3839	0.1702	1.8150
17	1.0906	21.9177	3.5542	0.3450	0.1703
18	1.1183	22.2627	3.9236	0.7238	0.3694
19	1.1779	22.9865	4.8149	0.8853	0.8912
20	1.2536	23.8718	6.1549	0.4948	1.3401
21	1.2973	24.3666	7.0451	2.3602	0.8902
22	1.5203	26.7268	13.1663	0.4334	6.1211
23	1.564	27.1601	14.7225	0.3312	1.5562
24	1.598	27.4913	16.0251	0.3879	1.3026
25	1.6385	27.8793	17.6869	0.5389	1.6618
26	1.696	28.4182	20.2630	0.2548	2.5761
27	1.7237	28.6730	21.5993	0.3294	1.3363
28	1.76	29.0024	23.4488	0.4649	1.8495
29	1.8122	29.4673	26.3125	0.5333	2.8636
30	1.8735	30.0006	29.9997	-----	3.6872

[표 3]은 Musa-Okumoto 모형과 Power law 모형의 평균값에 대한 연속적인 차이의 값을 계산한 표이고 그 결과값을 관리한게 값들에 적용시킨 평균값 차트가 [그림 4]에 요약 되었다.

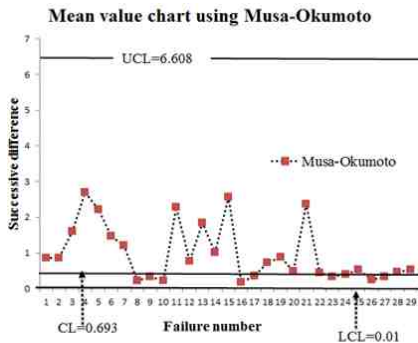


그림 4. Musa-Okumoto 모형의 평균값 차트
Fig. 4. Chart of mean value function for Musa-Okumoto model

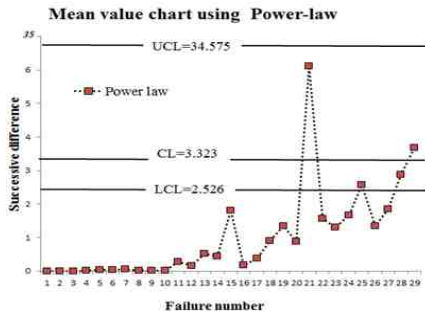


그림 5. Power-law 모형의 평균값 차트
Fig. 5. Chart of mean value function for Power-law model

SPC에서는 품질변동을 조사하여 그 원인을 우연원인(Random cause)과 이상원인(Assignable cause)으로 분리시켜, 이상원인은 현장에서 즉각 조치를 취하여 없앨 수 있는 요인을 의미하고 우연원인은 생산설비의 교체나 생산방식에 대한 시스템적 접근을 통하여 근본적으로 우연원인에 의한 품질변동의 크기를 축소시켜 주어야 하는 요인을 의미한다[8].

[표 3]과 [그림 4]에서 Musa-Okumoto모형은 모형에서는 평균차이에 대한 추정값이 관리하한(LCL)보다 낮게 타점된 곳이 없다. 반면에 Power law모형에서는 [표 3]과 [그림 5]에서 반이상의 추정값이 관리하한보다 낮게 나타나고 있어

이러한 추정값을 판단하여 이상원인으로 판명되는 되는 경우에는 시스템적 접근 예를 들면 작업방법 개선, 작업자 훈련, 작업환경의 균일화, 계측기 정도의 향상 등을 고려해 볼 수 있다.

결과적으로 사전에 이상원인을 제거함으로써 효율적인 통계적 공정분석을 통한 소프트웨어의 질을 개선 할 수 있다[2, 8]. 한편, Musa-Okumoto모형과 Power-law모형의 효율성 비교에 있어서 Power-law모형 보다는 Musa-Okumoto모형이 상대적으로 효율적이다. 그 근거는 첫째 관리하한보다 낮게 타점되는 이상값이 Musa-Okumoto모형에는 없지만 Power-law는 추정값이 존재하기 때문이다. 둘째, Musa-Okumoto 위험함수 형태는 감소하는 추세를 가진 반면 Power law 형태는 증가추세를 가진 특성을 가진다([그림 2]).

따라서 추정값을 판단하여 이상요인으로 판명되는 되는 경우에는 시스템적 접근방법을 고려하여 소프트웨어 고장품질을 향상 시킬 수 있다.

5. 결론

대용량 소프트웨어가 수검과 변경하는 과정에서 결점의 발생을 거의 피 할 수 없는 상황이 현실이다. 따라서 소프트웨어 관리자들이 소프트웨어 고장 원인 및 검사 도구에 효율적인 통계적 공정관리 과정을 이용한 NHPP 소프트웨어 모형에 대하여 연구 하였다.

신뢰성 분야에서 많이 사용되는 위험함수가 감소하는 추세를 따르는 Musa-Okumoto모형과 증가 추세를 따르는 Power-law 형태에 근거한 소프트웨어 NHPP 신뢰모형의 평균값함수를 이용한 통계적 공정관리 차트 제어 메커니즘을 비교 제안하였다. 즉, 관리 한계를 벗어나면 공정의 이상을 발견하고 제발방지 조치를 취함으로써 공정을 안정하게 유지하여 품질의 유지 및 향상을 도모하기 위한 목적으로 이용된다.

그 결과 Power-law모형 보다는 Musa-Okumoto모형이 상대적으로 효율적 모형으로 나타났다.

따라서 본 연구에 제안된 Musa-Okumoto모형

도 신뢰성 측면에서 더 효율적이기 때문에 이 분야에서 기존의 모형의 대안으로 사용할 수 있음을 확인 할 수 있었다. 이 연구를 통하여 소프트웨어 개발자들은 다양한 공정관리를 고려함으로써 소프트웨어 고장형태에 대한 사전지식을 파악하는데 어느 정도 도움을 줄 수 있으리라 사료된다.

REFERENCES

[1] Gokhale, S. S. and Trivedi, K. S. A, "time/structure based software reliability model", Annals of Software Engineering. 8, pp. 85-121. 1999.

[2] K,H Rao, R. S, Prasad and. R.L.Kantham "Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC", International Journal of Computer Applications (0975 -. 8887), Volume 21, No.7, pp. 1-5., May 2011.

[3]Hee-Cheul KIM, "The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Rayleigh and Burr Type", Journal of korea society of digital industry and information management, Volume 10, No.2, pp. 1-11. 2014.

[4] J. D. Musa, and K. Okumoto, "A logarithmic Poisson execution time model for software reliability measurement", Proc. 7th Inf. Conf. on Software Engineering, 23@238, 1984.

[5] K.Venkata Subba Reddy and Dr.B.Raveendra Babu , "A Log Based Approach for Software Reliability Modeling ", International Journal of Advanced Research in Computer Science and Software Engineering", Volume 4, pp. 49-5, 2014.

[6] Kuo, L. and Yang, T. Y, "Bayesian

Computation of Software Reliability," Journal of the American Statistical Association, Vol. 91, pp. 763-773, 1996.

[7]<http://www.itl.nist.gov/div898/handbook/apr/section1/apr172.htm>.

[8] K,H Rao, R. S, Prasad and. R.L.Kantham "Assessing Software Reliability Using SPC - An Order Statistics Approach", International Journal of Computer Science, Engineering and Applications (IJCEA) Vol.1, No.4, August 2011, pp. 121-131.

[9] Y. HAYAKAWA and G. TELFAR, "Mixed Poisson-Type Processes with Application in Software Reliability", Mathematical and Computer Modelling, 31, pp. 151-156, 2000.

[10] K. Kanoun and J. C. Laprie, "Handbook of Software Reliability Engineering", M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY, pp. 401-437, 1996.

저자약력

김희철 (Hee-Cheul Kim)

[정회원]



- 1992년 2월 : 동국대학교 통계학과 (이학석사)
- 1998년 8월 : 동국대학교 통계학과 (이학박사)
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수

<관심분야>

소프트웨어신뢰성 공학, 전산 통계, 웹프로그래밍