

다항 위험함수에 근거한 NHPP 소프트웨어 신뢰모형에 관한 통계적 공정관리 접근방법 비교연구

김희철*, 신현철**

The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on polynomial hazard function

Hee-Cheul Kim*, Shin-Hyun Cheul**

요약 소프트웨어 디버깅과정에서 오류의 발생 시간에 기초한 많은 소프트웨어 신뢰성 모형이 이미 연구되었다. 유한고장모형과 비동질적인 포아송과정을 이용하면 소프트웨어의 신뢰성 모형에 대한 모수 추정을 가능하게 한다. 소프트웨어를 사용자에게 인도하는 경우 인도시기를 결정할 때 조건부 고장률은 중요한 변수가 된다. 이러한 유한 고장 모형은 실제 다양한 상황에서 사용될 수 있다. 특성화 문제, 이상치의 검출, 선형 추정, 시스템 신뢰성 연구, 수명 시험, 생존 분석, 데이터 압축 및 많은 다른 분야의 연구에서 이들의 사용은 많은 연구에서 볼 수 있다. 통계 공정 관리(SPC)는 소프트웨어 오류의 예측을 모니터링 함으로써 소프트웨어의 신뢰성의 향상에 크게 기여할 수 있다. 관리도는 널리 소프트웨어 업계에서 소프트웨어 품질관리에 사용된다. 본 논문에서는 NHPP와 다항 위험 함수의 평균값을 기초한 관리 메카니즘을 제시하였다.

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 parameter inference for software reliability models based on finite failure model and non-homogeneous Poisson Processes (NHPP). For someone making a decision to market software, the conditional failure rate is an important variables. In this case, finite 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 polynomial hazard function

Key Words : Polynomial Hazard Functione, Non-Homogeneous Poisson Process, Statistical Process Control.

1. 서론

소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리사회에 엄청난 손실을 유발할 수도 있다. 따라서 소프트웨어 개발과정에서 소프트웨어 신뢰성은 중요한 문제이다. 이 문제는 사용자

의 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 개발과정은 필수 불가결하다. 결국 소

*Corresponding Author : Division of Industrial & Management Engineering, Namseoul University

**Division of Internet information, BaekSeok Culture University

Received August 9, 2015

Revised August 20, 2015

Accepted August 28, 2015

소프트웨어 제품의 결함내용을 예측하기 위한 모형개발이 필요하다. 지금까지 많은 소프트웨어 신뢰성모형이 제안되었다. 이 중에서 비동질적 포아송 과정(NHPP)에 의존한 모형은 에리 탐색 과정 측면에서는 우수한 모형이고, 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

Gokhale과 Trivedi[1]은 고양된 비동질적인 포아송 과정 모형(Enhanced NHPP) 모형을 제시하였고 Goel과 Okumoto[2]은 지수적 소프트웨어 신뢰성모형(Exponential software reliability growth model)을 제안 하였다. 이 모형은 결함의 누적수가 S-형태나 지수적 형태(S-shaped or exponential-shaped)를 가진 평균값함수(Mean value function)를 이용하였다. 이러한 모형에 의존한 일반화모형은 Yamada와 Ohba[3]에 의해 지연된 S-형태 신뢰 성장모형(Delayed S-shaped reliability growth model)과 변곡된 S-형태 신뢰 성장모형(Inflection S-shaped reliability growth model)이 제안되었다. Zhao[4]는 소프트웨어 신뢰도에서 변환점 문제를 제시하였고 Shyur[5]는 변환점을 이용한 일반화한 신뢰도 성장모형을 제안하였다. Pham와 Zhang[6]는 테스트 커버리지(Coverage)를 측정하여 소프트웨어 안정도를 평가 할 수 있는 소프트웨어 안정도 모형을 제시했다. 비교적 최근 Huang[7]은 일반화 로지스틱 테스트 노력함수(Generalized logistic testing-effort function)와 변환점 모수(Change-point parameter)를 통합하여 효율적인 소프트웨어 신뢰성을 예측하는 기술을 제시하기도 하였다. 그리고 최근에는 S-형태 모형은 소프트웨어 관리자들이 소프트웨어 및 검사도구에 익숙해지는 학습과정을 설명할 수 있다고 하였다[8]. 그리고 김희철[9]은 학습과정을 이용한 지수형 모형과 로그형 모형을 비교하는 연구도 이루어 졌다.

또한, 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, Prasad와 Kantham는 Half Logistics 분포를 이용한 공정관리 관리도를 이용한 메카니즘을 연구하기도 하였다[10]. 또한 김희철[11]은 Rayleigh형과 Burr형 NHPP 소프트웨어 신뢰모형에 관한 통계적 공정관리 접근방법 비교연구 결과 레일리 분포모형 보다는 버르분포 모형이 상대적으로 효율적 모형임을 확인하였다.

따라서 본 연구에서는 다항 위험함수에 근거한 무한고장 NHPP 소프트웨어 모형에 대한 공정관리 접근방법을 비교 제시하였다.

2. 관련연구

2.1 무한고장 NHPP 모형

NHPP모형에서 평균값함수 $m(t)$ (Mean value function)와 강도함수 $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[1, 12].

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

$N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률밀도함수 (Probability density function)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 시간관련모형(Time domain model)들은 NHPP에 의해서 확률고장과정으로 설명이 가능하다. 이러한 NHPP모형들은 유한고장모형과 무한고장 범주로 분류한다[13].

무한고장 NHPP모형들은 수리시점에서도 고장이 발생할 상황을 반영하기 위하여 기록 멈춤 통계량(Record breaking statistics)을 사용하는 RVS(Record Value Statistics)모형을 사용할 수 있다고 하였고 이 RVS모형과 NHPP모형에 관해서 평균값함수는 다음과 같이 된다고 하였다[1, 12].

$$m(t) = -\ln(1-F(t)) \quad (3)$$

따라서 (3)식을 이용하고 $f(t)$ 을 확률밀도함수, $F(t)$ 을 분포함수라고 하면 NHPP의 강도함수는 다음과 같이 위험함수($h(t)$)가 된다.

$$\lambda(t) = m'(t) = f(t)/(1-F(t)) = h(t) \quad (4)$$

시간 $(0, t]$ 까지 조사하기 위한 시간절단(Time truncated)모형은 n 번째까지 고장시점 자료를

$$x_k = \sum_{i=1}^k t_k \quad (k=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (5)$$

이라고 하면 데이터 집합 D_t 는 $\{n, x_1, x_2, \dots, x_n; t\}$ 와 같이 구성된다. n 번째까지 고장시점이 관찰된 고장 절단모형일 경우에 데이터 집합 D_{x_n} 은 $\{x_1, x_2, \dots, x_n\}$ 으로 구성되며, 이 시간절단 모형에서 θ 을 모수공간이라고 표시하면 무한고장 우도함수는 다음과 같이 알려져 있다[12].

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

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

2.2 다항 위험함수를 이용한 NHPP모형

다항 위험함수(Polynomial hazard function) 형태는 다음과 같이 알려져 있다[13].

$$h(t) = \beta_0 + \beta_1 t + \dots + \beta_{m-1} t^{m-1}, \quad t \geq 0 \quad (7)$$

단, $\beta_i (i=0,1,2,\dots,m-1)$ 는 다항위험함수의 계수이고 (4)식에서 보여 주듯이 무한고장 NHPP의 강도함수는 위험함수($h(t)$)가 된다고 하였기 때문에 다항형태의 무한고장 NHPP 강도함수 $\lambda(t)$ 는 (7)식과 동일한 위험함수 $h(t)$ 가 된다.

따라서 평균값함수 $m(t) = \int_0^t \lambda(s) ds$ 은 다음

과 같이 유도된다.

$$m(t) = \beta_0 t + \frac{1}{2} \beta_1 t^2 + \frac{1}{3} \beta_2 t^3 + \dots + \frac{1}{m} \beta_{m-1} t^m \quad (8)$$

(8) 식에서 $m=1$ 이면 지수분포가 되고 $m=2$ 일 경우에는 선형 위험률 분포(Linear hazard rate distribution)라고 알려져 있다[10]. 그리고 $m>2$ 경우에는 비 단조 위험 함수(non-monotone hazard function) 형태를 가진다고 하였다[6]. 따라서 본 연구에서는 $m=2$ 인 경우를 이용한 다항모형을 적용하고자 한다. 이 경우의 다항 모형의 우도함수는 다음과 같이 표현된다.

$$L_{NHPP_{poly}}(\theta | \underline{x}) = \prod_{i=1}^n (\beta_0 + \beta_1 x_i) e^{-\beta_0 x_n - \frac{1}{2} \beta_1 x_n^2} \quad (9)$$

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

모수추정은 최우추정법(MLE)을 이용하였다. 이러한 추정 방법을 사용하기 위한 로그 우도 함수는 (9)식을 연관 하면 다음과 같이 유도된다.

$$\ln L_{NHPP_{m=2}}(\theta | \underline{x}) = \sum_{i=1}^n \ln(\beta_0 + \beta_1 x_i) - \beta_0 x_n - \frac{1}{2} \beta_1 x_n^2 \quad (10)$$

(10)식에서

$$\frac{\partial \ln L_{NHPP_{poly}}(\theta | \underline{x})}{\partial \beta_0} = 0, \quad \frac{\partial \ln L_{NHPP_{poly}}(\theta | \underline{x})}{\partial \beta_1} = 0$$

을 만족하는 $\hat{\beta}_{0,MLE}$ 와 $\hat{\beta}_{1,MLE}$ 는 다음과 같이 구할 수 있다[11, 12]

$$\frac{\partial \ln L_{NHPP_{poly}}(\theta | \underline{x})}{\partial \beta_0} = \sum_{i=1}^n \frac{1}{\beta_0 + \beta_1 x_i} - x_n = 0 \quad (11)$$

$$\frac{\partial \ln L_{NHPP_{poly}}(\theta | \underline{x})}{\partial \beta_1} = \sum_{i=1}^n \frac{x_i}{\beta_0 + \beta_1 x_i} - \frac{1}{2} x_n^2 = 0 \quad (12)$$

단, $x = (x_1, x_2, x_3, \dots, x_n)$

그러나 위 식들은 근을 구하는데 근이 수렴되지 않으므로 본 연구에서는 β_0 의 추정치를 고정된 상수로 간주하여 전개하고자 한다.

3. 제안된 다항 위험함수 NHPP 신뢰성 모형에 관한 통계적 공정관리 접근 방법

공정이 안정된 상태에 있는가를 조사하기 위해, 또는 공정을 안정된 상태로 유지하기 위해 활용하는 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 [10].

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[10].

따라서 다항 위험함수모형의 평균값함수 (8) 식을 이용하면 관리상한은 다음과 같은 관계로 정리할 수 있다[10].

$$F(t) = 1 - e^{-m(t)} = 1 - e^{-\beta_0 t - \frac{1}{2}\beta_1 t^2} = 0.99865 \quad (13)$$

(13)식은 다음과 같이 정리된다.

$$\frac{1}{2}\beta_1 t^2 + \beta_0 t - \ln(0.00135) = 0 \quad (14)$$

(14)식을 t 에 관하여 정리하기 위하여 2차함수의 근의 공식을 이용하면 일반적인 관리상한은 다음과 같이 유도된다[10, 14].

$$t = \left(\frac{-\beta_0 + \sqrt{\beta_0^2 - 2\beta_1 \times (-\ln(0.99865))}}{\beta_1} \right) = t_U \quad (15)$$

단, $t > 0$.

또한, 유사하게 중심선과 관리하한은 다음과 같이 유도된다.

$$t = \left(\frac{-\beta_0 + \sqrt{\beta_0^2 - 2\beta_1 \times (-\ln(0.5))}}{\beta_1} \right) = t_C \quad (16)$$

$$t = \left(\frac{-\beta_0 + \sqrt{\beta_0^2 - 2\beta_1 \times (-\ln(0.00195))}}{\beta_1} \right) = t_L \quad (17)$$

따라서 평균값함수의 차이를 이용한 관리한계, 즉 관리상한($m(t_U)$), 중심선($m(t_C)$), 관리하한($m(t_L)$)은 각각 다음과 같이 유도된다[10, 14].

$$m(t_U) = \beta_0 t_U + \frac{1}{2}\beta_1 t_U^2 \quad (18)$$

$$m(t_C) = \beta_0 t_C + \frac{1}{2}\beta_1 t_C^2 \quad (19)$$

$$m(t_L) = \beta_0 t_L + \frac{1}{2}\beta_1 t_L^2 \quad (20)$$

4. 소프트웨어 고장자료 공정분석

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

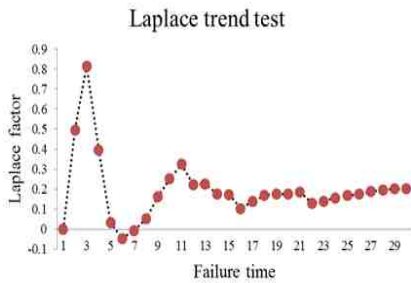


그림 2. 라플라스 추세검정
Fig. 2. Laplace trend test

이 장에서 소프트웨어 고장시간 자료[15] (Failure time data)을 가지고 제시하는 신뢰모형들을 분석하고자 한다. 이 자료의 고장시간은 18.735시간단위에 30번의 고장이 발생한 자료이며 [표 1]에 나열 되어 있다. 또한 제시하는 신뢰모형들을 분석하기 위하여 우선 적용 자료에 대한 추세검정이 선행되어야 한다[12, 16]. 추세분석에는 일반적으로 라플라스 추세검정 (Laplace trend test)을 사용한다. 이 검정을 실시한 결과 [그림 1]에서 라플라스 추세검정의 결과는 라플라스 요인(Factor)이 -2와 2사이에 존재함으로써(극단값이 존재하지 않음) 신뢰성장 (Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰성장모형을 제시하는 것이 효율적임을 시사하고 있다[12, 16].

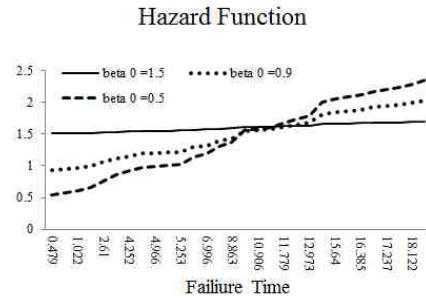
표 2. 각 모형에 대한 모수추정값 및 관리한계
Table 2. Parameter estimation and control limits of each model

β_0	$\hat{\beta}_{1MLE}$	Control limits		
		$m(t_U)$	$m(t_C)$	$m(t_L)$
0.5	0.0987	8.503	7.317×10^{-1}	1.351×10^{-3}
0.9	0.0601	6.014	6.825×10^{-2}	1.350×10^{-3}
1.5	0.0104	1.839	6.201×10^{-2}	9.012×10^{-5}

Note. MLE : Maximum likelihood estimation.

모수 추정은 최우추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러

한 계산은 초기 값을 0.01과 5을, 허용한계 (Tolerance for width of interval)는 10^{-5} 을 주고 수렴성을 확인하면서 충분한 반복횟수인 100 번을 C-언어를 이용하여 모수 추정을 수행하였다. 최우추정법의 결과와 관리한계 값들이 [표 2]에 요약되었다.



Note. $\beta_0 = \beta_0$

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

본 연구에서는 (7)식을 이용하여 위험함수(강도함수)가 <그림 2>에 요약되었다.

이 그림에서 다항위험함수모형의 위험함수의 특성은 비증가패턴을 보이고 있고 특히 $\beta_0 = 1.5$ 인 경우에는 변화 추이가 거의 없는 반면에 $\beta_0 = 0.9$ 와 $\beta_0 = 0.5$ 는 경우에는 고장시간이 지날 수록 상승패턴을 보이고 있다.

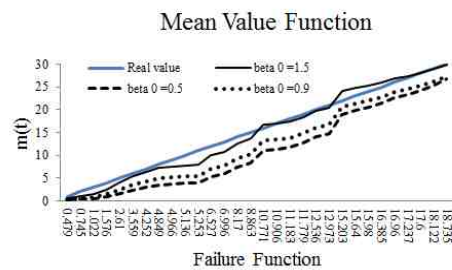


그림 3. 평균값함수
Fig. 3. Mean value function

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

Failure Number	Failure Time (i)	$m(i+1) - m(i), i = 1, 2, \dots, 29$ Successive difference		
		$\beta_0 = 0.5$	$\beta_0 = 0.9$	$\beta_0 = 1.5$
1	0.0479	0.14906757	0.249183799	0.400693037
2	0.0745	0.162654802	0.264008243	0.418045187
3	0.1022	0.34802906	0.541850725	0.838484318
4	0.1576	0.730602789	1.060666136	1.573507285
5	0.261	0.763413702	1.030024149	1.453942781
6	0.3559	0.613632685	0.786361341	1.06764772
7	0.4252	0.566633207	0.700570575	0.923753144
8	0.4849	0.115171319	0.139808068	0.181471446
9	0.4966	0.169750729	0.204606067	0.263930168
10	0.5136	0.118485567	0.141826166	0.181820668
11	0.5253	1.377630982	1.597581986	1.989040144
12	0.6527	0.547491863	0.612685724	0.736479892
13	0.6996	1.465671025	1.591636764	1.853585397
14	0.817	0.929020935	0.978406263	1.100880119
15	0.8863	2.802733513	2.842923244	3.056800694
16	1.0771	0.211917593	0.20943817	0.217717254
17	1.0906	0.440455526	0.433165523	0.447316996
18	1.1183	0.973372121	0.947644828	0.96516383
19	1.1779	1.286858554	1.234413973	1.231213566
20	1.2536	0.768625819	0.728280362	0.713466652
21	1.2973	4.215782888	3.895116024	3.671728896
22	1.5203	0.883658596	0.79832565	0.725587633
23	1.564	0.70055198	0.62906154	0.56590416
24	1.598	0.849371164	0.758390141	0.67566069
25	1.6385	1.233706056	1.093659919	0.96220155
26	1.696	0.60597128	0.533950698	0.464757359
27	1.7237	0.80557176	0.706707222	0.610258321
28	1.76	1.181223725	1.030138864	0.879963797
29	1.8122	1.421481378	1.230629897	1.036985373
30	1.8735	-----	-----	-----

[그림 3]은 다항위험함수모형에 대한 평균값 함수의 값을 나타내었다. 이 그림에서 $\beta_0 = 1.5$ 인 경우에는 거의 참값과 변화 추이가 거의 비슷한 반면 $\beta_0 = 0.9$ 와 $\beta_0 = 0.5$ 는 경우에는 고장시간이 지날수록 과소평가 추정 되었다.

[그림 3]은 다항위험함수모형에 대한 평균값 함수의 값을 나타내었다. 이 그림에서 $\beta_0 = 1.5$ 인 경우에는 거의 참값과 변화 추이가 거의 비슷한 반면 $\beta_0 = 0.9$ 와 $\beta_0 = 0.5$ 는 경우에는 고장시간이 지날수록 과소평가 추정 되었다.

[표 3]은 다항위험함수모형의 평균값에 대한 연속적인 차이의 값을 계산한 표이고 그 결과값을 관리한계 값들에 적용시킨 평균값 차트가 [그림 4]와 [그림 5] 및 [그림 6]에 요약 되었다.

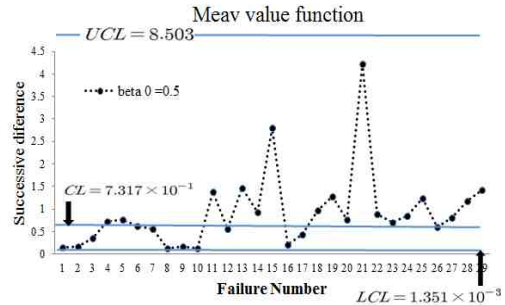


그림 4. $\beta_0 = 0.5$ 인 모형의 평균값 관리도
Fig. 4. Mean value function control chart of $\beta_0 = 0.5$

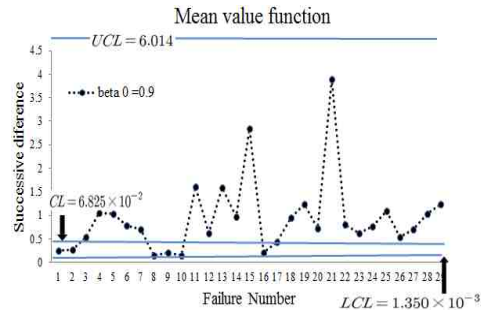


그림 5. $\beta_0 = 0.9$ 인 모형의 평균값 관리도
Fig. 5. Mean value function control chart of $\beta_0 = 0.9$

[표 3]과 [그림 4], [그림 5]에서 $\beta_0 = 0.5$ 인 경우와 $\beta_0 = 0.9$ 인 경우의 다항위험함수모형에서는 평균차이에 대한 추정값이 관리한(LCL)보다 낮게 타점된 곳이 없고 또한 상한(UCL) 보다 높게 나타난 점이 없기 때문에 정상적인 형태로 나타나고 있다. 그러나 [표 4]에서 보여주듯이 $\beta_0 = 0.9$ 인 경우가 $\beta_0 = 0.5$ 인 경우보다 평균이 높고 범위가 작기 때문에 고장간격시간이 상대적으로 길게 나타나 효율적 모형으로 판단된다.

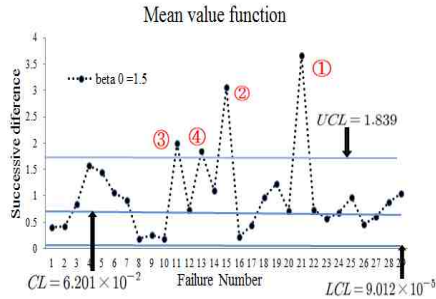


그림 6. $\beta_0 = 1.5$ 인 모형의 평균값 관리도
 Fig. 6. Mean value function control chart of $\beta_0 = 1.5$

[표 3]과 [그림 6]에서는 $\beta_0 = 1.5$ 인 경우에는 관리하한(LCL)보다 낮게 타점된 곳이 없지만 상한(UCL) 보다 높게 나타난 점이 4군데나 발생하였다. 즉 일반적인 관리도에서는 이러한 추정 값을 판단하여 이상 원인으로 판명되는 경우에는 시스템적 접근 예를 들면 작업방법 개선, 작업자 훈련, 작업환경의 균일화, 계측기 정도의 향상 등을 고려해 볼 수 있다.

그러나 평균값에 대한 연속적인 차이에 대한 관리도에서는 상한보다 높게 표시된 타점들은 고장간격시간이 보다 길게 나타남을 의미함으로써 효율적인 현상으로 받아들일 수 있다[10]. 따라서 $\beta_0 = 1.5$ 인 경우가 $\beta_0 = 0.5$ 인 경우와 $\beta_0 = 0.9$ 인 경우보다 효율적 모형으로 판단된다.

표 4. 연속적인 평균값 차이에 대한 기초통계정보
 Table 4. Statistic information of successive difference of mean value function model

통계량	$\beta_0 = 0.5$	$\beta_0 = 0.9$	$\beta_0 = 1.5$
평균	0.9117	0.9300	1.0072
표준오차	0.1575	0.1478	0.1496
범위	4.1006	3.7553	3.4903

또한 [표 4]의 요약 표에서도 $\beta_0 = 1.5$ 인 경우가 $\beta_0 = 0.9$ 인 경우와 $\beta_0 = 0.5$ 인 경우보다 평균이 높고 범위가 작기 때문에 고장간격시간이 상대적으로 길게 나타나 효율적 모형으로 판단된다.

결국, 다항위험함수모형의 효율성 비교에 있어서 다항위험함수의 절편(intercept) 모수가 클수록 상대적으로 효율적이다. 그 근거는 첫째 관리하한보다 낮게 타점되고장간격시간이 상대적으로 길게 나타나는 관리상한을 벗어나는 타점이 많이 존재하기 때문이다. 둘째 품질변동의 측도인 평균과 범위가 높게 나타나기 때문이다.

셋째, 비교적 다항위험함수의 절편(intercept) 모수가 클수록 변화 추이가 거의 없는 반면에 작을수록 과소평가되는 특성을 가진다([그림 2]).

따라서 평균값 차이 관리도를 이용한 시스템적 접근방법을 고려하면 소프트웨어 고장품질을 향상시킬 수 있다.

5. 결론

대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 수 없는 상황이 현실이다. 따라서 소프트웨어 관리자들이 소프트웨어 고장원인 및 검사도구에 효율적인 통계적 공정관리 과정을 이용한 NHPP 소프트웨어 모형에 대하여 연구 하였다.

신뢰성분야에서 많이 사용되는 위험함수가 증가추세를 따르는 다항위험함수 모형에 근거한 소프트웨어 NHPP 신뢰모형의 평균값함수를 이용한 통계적 공정관리 차트 제어 메커니즘을 비교 제안하였다. 즉, 다항위험함수모형의 효율성 비교에 있어서 다항위험함수의 절편(intercept) 모수가 클수록 품질변동의 측도인 평균과 범위가 높게 나타나기 때문에 상대적으로 효율적 모형으로 나타났다.

이 연구를 통하여 소프트웨어 개발자들은 다양한 공정관리를 고려함으로써 소프트웨어 고장 형태에 대한 사전지식을 파악하는데 도움을 줄 수 있으리라 사료 된다.

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] Goel A L, Okumoto K, “Time-dependent fault detection rate model for software and other performance measures”, *IEEE Trans. Reliab.* 28, pp.206-11, 1978.
- [3] Yamada S, Ohba H, “S-shaped software reliability modeling for software error detection”, *IEEE Trans. Reliab.* 32, pp.475-484, 1983.
- [4] Zhao M, “Change-point problems in software and hardware reliability”, *Commun. Stat. Theory Methods*, 22(3), pp.757-768, 1993.
- [5] Shyur H-J, “A stochastic software reliability model with imperfect debugging and change-point”, *J. Syst. Software* 66, pp.135-141, 2003.
- [6] Pham H, Zhang X, “NHPP software reliability and cost models with testing coverage”, *Eur. J. Oper. Res.* 145, pp.445-454, 2003.
- [7] Huang C-Y, “Performance analysis of software reliability growth models with testing-effort and change-point”, *J. Syst. Software* 76, pp. 181-194, 2005.
- [8] Kuei-Chen, C., Yeu-Shiang, H, and Tzai-Zang, L., “A study of software reliability growth from the perspective of learning effects”, *Reliability Engineering and System Safety* 93, pp. 1410 - 1421, 2008.
- [9] Kim, Hee Cheul, “The Comparative Study of NHPP Software Reliability Model Exponential and Log Shaped Type Hazard Function from the Perspective of Learning Effects, *The Journal of Korea Society of Digital Industry and Information Management*, 8(2): 1-10, 2012.
- [10] 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.*
- [11] Kim, Hee Cheul, “The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Rayleigh and Burr Type , *The Journal of Korea Society of Digital Industry and Information Management*, 10(2): 1-11, 2014..
- [12] Hee-Cheul Kim and Kyung-Soo Kim., “Software Development Cost Model based on NHPP Gompertz Distribution”, *Indian Journal of Science and Technology*, Vol.8(12), 68322, June 2015, pp. 1-5.
- [13] J. F. Lawless. *Statistical Models and Methods for Lifetime Data.* John Wiley & Sons, New York, 1981.
- [14] 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 (IJCSEA) Vol.1, No.4, August 2011, pp. 121-131.*
- [15] Y. HAYAKAWA and G. TELFAR, “Mixed Poisson-Type Processes with Application in Software Reliability”, *Mathematical and Computer Modelling*, 31, 2000, pp.151-156.
- [16] K. Kanoun, J. C. Laprie, “Handbook of Software Reliability Engineering,” M. R. Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY, 1996, pp. 401-437.

저자약력

김 희 철 (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

신 현 철(Shin-Hyun Cheul)

[정회원]



- 2002년 2월 : 원광대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 : 한국정보처리학회 이사
- 1994년 3월 ~ 현재 : 백석문화대학교 인터넷정보학부 교수

<관심분야> 통신공학, 컴퓨터과학, 소프트웨어공학