

# 통계적 공정관리(SPC)를 이용한 무한고장 소프트웨어 신뢰성 모형에 대한 접근방법 연구

김희철\* · 신현철\*\*

## 요 약

소프트웨어의 디버깅에 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. 유한 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, 로그선형 모형 그리고 파레토 모형의 평균값 함수를 이용한 통계적 공정관리 차트를 이용한 제어 메커니즘을 제안하였다.

## Assessing Infinite Failure Software Reliability Model Using SPC (Statistical Process Control)

Kim Hee Cheul\* · Shin Hyun Cheul\*\*

## 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 asymptotic likelihood inference for software reliability models based on infinite 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 finite failure model are used in a wide variety of practical situations. Their use in characterization problems, detection of outliers, 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, we proposed a control mechanism based on NHPP using mean value function of log Poission, log-linear and Parto distribution.

**Key words :** Statistical Process Control (SPC), Non-Homogeneous Poisson Process, Finite Failure Model.

---

접수일(2012년 11월 15일), 수정일(1차: 2012년 12월 17일),  
게재확정일(2012년 12월 24일)

\* 남서울대학교 산업경영공학과

\*\* 백석문화대학교 인터넷정보학부

## 1. 서 론

소프트웨어 신뢰성에 관한 연구 활동은 지난 30년 전부터 행해져 오고 있고 많은 신뢰도 성장 모형들이 소프트웨어에 남아 있는 고장들의 수와 소프트웨어 신뢰도의 추정에 관한 문제들을 제안해 왔다.

일반적으로 소프트웨어 개발과정은 설계단계, 디자인, 코딩 그리고 테스트 단계를 거친다. 이러한 과정을 거친 후 소프트웨어 제품을 방출하게 되는데 방출 이후에 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[1][2]. 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 발생 할 수 있다. 이러한 상황을 반영하는 모형이 무한고장 모형이다.

이 분야에서는 Musa-Oikumoto의 대수 포아송 실행시간 모형[3]과 로그-파우어 모형[4]을 이용한 방출 문제에 대한 문제들이 이미 연구되었고 최근에도 이와 관련된 문제에 대한 연구는 Yang 과 Xie(2000) 와 Huang(2005)에 의해 보증기간 할인을 및 보증기간 이후의 사후대책 등에 대하여 연구되고 있다[5][6].

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. 이 분야에서 K,H Rao, R. S, Prasad and. R.L.Kan tham(2011)는 Half Logistics 분포를 이용한 공정관리 차트를 이용한 메카니즘을 연구하기도 하였다[7].

이 분야에서 본 논문에서는 신뢰성 분야에서 많이 사용되는 NHPP에 근원을 둔 로그 포아송 실행시간 모형, 로그선형 모형 그리고 파레토 모형의 평균값 함수를 이용한 통계적 공정관리 차트를 이용한 제어 메카니즘을 제안하였다.

## 2. 관련 연구

### 2.1 무한고장 NHPP

NHPP 모형에서 평균값 함수  $m(t)$  (Mean value function)와 강도함수(Intensity function)  $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[1].

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

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

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

이러한 시간 관련 모형(Time domain models)들은 강도 함수  $\lambda(t)$ 가 다르게 표현됨으로서 평균값 함수  $m(t)$ 도 역시 다르게 나타나는 NHPP에 의해서 확률 고장 과정으로 표현되고 유한 고장 모형과 무한 고장 범주로 분류한다[8]. 무한 고장 NHPP 모형들은 실제 상황에서는 수리 시점에서도 고장이 발생할 상황을 반영하기 위하여 기록 멈춤 통계량(Record breaking statistics)을 사용하는 RVS(Record Value Statistics) 모형을 사용할 수 있다고 하였고 이 RVS 모형과 NHPP 모형에 관해서 평균값함수는 다음과 같이 된다고 하였다[8, 9].

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

따라서 (2.1)식과 (2.3)식을 연관시키고  $f(t)$ 을 확률 밀도함수,  $F(t)$ 을 분포함수라고 하면 다음과 같은 관계식에 의해 NHPP의 강도함수는  $F(t)$ 의 위험함수 ( $h(t)$ )가 된다[8].

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

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

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

이라고 하고  $\theta$  을 모수공간이라고 하면 우도 함수는 다음과 같이 알려져 있다[10, 11].

$$L_{NHPP_{FR}}(\theta | D_t) = \prod_{i=1}^n \left( \frac{f(x_i)}{1-F(x_i)} \right) (1-F(x_n)) \\ = \left( \prod_{i=1}^n h(x_i) \right) (1-F(x_n)) \quad (6)$$

### 2.2 로그 포아송 실행시간모형

로그 포아송 실행시간(Log Poisson execution time)모형[10]은 1984년에 Musa 와 Okumoto에 의해서 소개된 무한 고장 소프트웨어 모형으로 평균값함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t) = \frac{1}{\theta} \ln(\lambda_0 \theta t + 1) \quad (7)$$

$$\lambda(t) = \frac{\lambda_0}{\lambda_0 \theta t + 1} \quad (8)$$

한편, (3)식에 (7)식과 (8)식을 이용하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(\theta, \lambda_0 | \underline{x}) = \left( \prod_{i=1}^n \frac{\lambda_0}{\lambda_0 \theta x_i + 1} \right) \cdot e^{-\frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1)} \quad (9)$$

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

모수 최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 다음과 같이 유도된다.

$$\ln L(\theta, \lambda_0 | \underline{x}) = n \ln \lambda_0 - \ln \sum_{i=1}^n (\lambda_0 \theta x_i + 1) - \frac{1}{\theta} \ln(\lambda_0 \theta x_n + 1) \quad (10)$$

(10)식을 이용하여 최우추정치  $\hat{\theta}_{MLE}$  와  $\hat{\lambda}_{0MLE}$  은 다음과 같이 구할 수 있다고 하였다.

$$\hat{\theta}_{MLE} = \frac{1}{n} \ln(\hat{\phi}_{x_n} + 1) \quad (11)$$

$$\hat{\lambda}_{0MLE} = \hat{\phi} / \hat{\theta}_{MLE} \quad (12)$$

단,  $\phi (= \hat{\lambda}_{0MLE} \cdot \hat{\theta}_{MLE})$  는 (13) 식의 근이 된다.

즉, 이  $\phi$  근을 구하기 위해서는 수치 해석적 방법으로 다음과 같은 식을 이용하여 계산할 수 있다[8].

$$\frac{\partial \ln L(\phi | \underline{x})}{\partial \phi} = \frac{n}{\phi} - \sum_{i=1}^n \frac{x_i}{\phi x_i + 1} - \frac{n x_n}{(\phi x_n + 1) \ln(\phi x_n + 1)} = 0 \quad (13)$$

### 2.3 로그-선형 모형

로그-선형 분포는 수명분포로 적용 할 수 있으며 다른 수명분포에 비해 실제적으로 효율적임을 보여 주었다[8, 12]. 이러한 로그-선형 수명분포의 강도함수의 패턴은 증가 추세를 가지며 다음과 같이 알려져 있다.

$$\lambda(t) = \exp(\alpha + \beta t) \quad (14)$$

단,  $0 < t < \infty, -\infty < \alpha < \infty, \beta > 0$ .

따라서 평균값 함수는 (1) 식  $m(t) = \int_0^t \lambda(s) ds$  을 이용하면 다음과 같이 유도 할 수 있다.

$$m(t) = \frac{e^\alpha (e^{\beta t} - 1)}{\beta} \quad (15)$$

우도함수도 (3)식을 이용하면 다음과 같이 표현된다.

$$L(\alpha, \beta | \underline{x}) = \exp \left( n\alpha + \beta \sum_{i=1}^n x_i - \frac{e^\alpha (e^{\beta x_n} - 1)}{\beta} \right) \quad (16)$$

단,  $\underline{x} = (x_1 \leq x_2 \leq x_3, \dots \leq x_n)$ . (16)식에 의해서  $\alpha$  와  $\beta$  에 관한 최우추정치  $\hat{\alpha}_{MLE}$  와  $\hat{\beta}_{MLE}$  는 다음 식을 이용하면 추정 할 수 있다고 하였다[9, 12].

$$\sum_{i=1}^n x_i + \frac{n}{\beta} - \frac{n x_n}{1 - e^{-\beta x_n}} = 0 \quad (17)$$

$$e^\alpha = \frac{n \beta}{e^{\beta x_n} - 1} \quad (18)$$

### 2.4 파레토 모형

확률분포의 꼬리 부분이 정규분포보다 두꺼운

(Heavy-tailed) 즉, 다소 극단적인 값들이 발생할 확률을 무시할 수 없는 파레토(Parto distribution) 분포의 확률 밀도 함수와 분포 함수는 각각 다음과 같다 [9][12].

$$f_{Pretto}(t | \alpha, \beta) = \frac{\alpha \beta^\alpha}{(\beta+t)^{(1+\alpha)}} \quad (\alpha, \beta > 0, t > 0) \quad (19)$$

$$F_{Pretto}(t | \alpha, \beta) = 1 - \left( \frac{\beta^\alpha}{(\beta+t)^\alpha} \right) \quad (20)$$

위 분포함수와 (3)식과 (4)을 이용하면 강도함수와 평균값 함수는 다음과 같이 표현된다.

$$\lambda(t) = \frac{\alpha}{t+\beta} \quad (21)$$

$$m(t) = \alpha \ln \left( 1 + \frac{t}{\beta} \right) \quad (22)$$

(6)식과 관련하여 무한 NHPP 고장 우도 함수는 다음과 같다[8].

$$L(\alpha, \beta | \underline{x}) = \left( \prod_{i=1}^n \frac{\alpha}{\beta+x_i} \right) \exp \left\{ -\alpha \ln \left( 1 + \frac{x_n}{\beta} \right) \right\} \quad (23)$$

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

최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 (23) 과 관련하여 다음과 같이 유도된다.

$$\ln L(\alpha, \beta | \underline{x}) = n \ln \alpha - \sum_{i=1}^n \ln(\beta+x_i) - \alpha \ln \left( 1 + \frac{x_n}{\beta} \right) \quad (24)$$

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

$$\frac{\partial \ln L(\alpha, \beta | \underline{x})}{\partial \alpha} = \frac{n}{\alpha} - \ln \left( 1 + \frac{x_n}{\beta} \right) = 0 \quad (25)$$

$$\begin{aligned} \frac{\partial \ln L(\alpha, \beta | \underline{x})}{\partial \beta} &= \\ - \sum_{i=1}^n \frac{1}{(\beta+x_i)} - \alpha \left( \frac{1}{\beta+x_n} - \frac{1}{\beta} \right) &= 0 \end{aligned} \quad (26)$$

## 2.5 식스 시그마에 의존된 관리 한계

공정이 안정된 상태에 있는가를 조사하기 위해, 또는 공정을 안정된 상태로 유지하기 위해 활용하는 도표가 관리도 이다. 관리도의 관리한계는 관리상한(UCL; Upper Control Limit), 중심선(CL; Control Limit), 관리하한(LCL; Low Control Limit)으로 구성된다.

즉, 관리 한계를 벗어나면 공정의 이상을 발견하고 재발방지 조치를 취함으로써 공정을 안정하게 유지하여 품질의 유지 및 향상을 도모하기 위한 목적으로 이용된다.

소프트웨어 신뢰성 모형에 대한 평균값 함수( $m(t)$ )의 값을 각각 기준 확률 0.99865, 0.00135, 0.5을 이용하면 관리상한(UCL), 관리하한(LCL) 중심선(CL)을 추정할 수 있다고 하였다[7].

로그 포아송 실행시간 평균값함수를 이용한 관리상한은 다음과 같이 유도된다.

$$m(t) = \frac{1}{\theta} \ln(\lambda_0 \theta t + 1) = 0.99865$$

이식을  $t$  에 관하여 정리하면 관리상한은 다음과 같이 유도된다.

$$t = \frac{e^{0.99865 \times \theta} - 1}{\lambda_0 \theta} = UCL$$

유사한 방법으로 관리하한(LCL) 중심선(CL)은 다음과 같이 유도된다.

$$t = \frac{e^{0.00135 \times \theta} - 1}{\lambda_0 \theta} = LCL$$

$$t = \frac{e^{0.5 \times \theta} - 1}{\lambda_0 \theta} = CL$$

유사한 방법으로 로그-선형 분포 모형과 파레토분포 모형의 관리 한계는 각각 <표 1>와 <표 2>에 요약되었다.

<표 1> 로그-선형 분포 모형의 관리 한계

Model	로그-선형 분포 모형
UCL	$\frac{\ln\left(1 + \frac{\beta}{e^\alpha} 0.99865\right)}{\beta}$
LCL	$\frac{\ln\left(1 + \frac{\beta}{e^\alpha} 0.00135\right)}{\beta}$
CL	$\frac{\ln\left(1 + \frac{\beta}{e^\alpha} 0.5\right)}{\beta}$

<표 2> 파레토분포 모형의 관리 한계

Model	파레토분포 분포 모형
UCL	$\exp\left(\ln(\beta) + \frac{0.99865}{\alpha}\right) - \beta$
LCL	$\exp\left(\ln(\beta) + \frac{0.00135}{\alpha}\right) - \beta$
CL	$\exp\left(\ln(\beta) + \frac{0.5}{\alpha}\right) - \beta$

### 3. 소프트웨어 고장 자료 분석 및 공정관리 분석

이 장에서 소프트웨어 고장 간격 시간 자료[13] (failure interval time data)를 가지고 제시하는 신뢰모형들을 분석하고자 한다. 이 자료의 고장 시간은 18.735 시간단위에 30번의 고장이 발생한 자료이며 <표 3>에 나열 되어 있다. 모수추정은 최우추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 기법 방법인 이분법(bisection method)을 사용하였다. 이러한 계산은 초기값을 0과 10을, 허용 한계(tolerance for width of interval)는  $10^{-5}$  을 주고 수렴성을 확인하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다. 최우추정법의 결과는 <표 4>에 요약되었다[12].

<표 3> 소프트웨어 고장 자료

Failure number	Failure Time(hours)	Failure Interval (hours)
1	5.649	5.649
2	8.92	3.271
3	20.29	11.37
4	29.955	9.665
5	34.715	4.76
6	75.95	41.235
7	78.171	2.221
8	78.625	0.454
9	83.022	4.397
10	89.114	6.092
11	89.804	0.69
12	92.86	3.056
13	93.66	0.8
14	110.655	16.995
15	111.988	1.333
16	122.545	10.557
17	127.045	4.5
18	128.712	1.667
19	128.99	0.278
20	131.768	2.778
21	131.829	0.061
22	141.712	9.883
23	164.212	22.5
24	342.85	178.638
25	356.144	13.294
26	399.144	43
27	446.494	47.35
28	476.644	30.15
29	497.144	20.5
30	497.661	0.517
31	591.161	93.5
32	665.644	74.483
33	686.444	20.8
34	765.944	79.5
35	772.977	7.033
36	774.944	1.967
37	791.561	16.617
38	815.978	24.417
39	837.145	21.167
40	861.945	24.8
41	1197.945	336

<표 4> 각 모형의 모수 추정값

Model	MLE	
Log Poission execution time	$\hat{\theta} = 0.020031$	$\hat{\lambda}_0 = 0.053067$
Log-linear	$a = -3.72358$	$\hat{b} = 0.000552$
Pareto	$\alpha = 6.345853$	$\hat{\beta} = 1.875762$

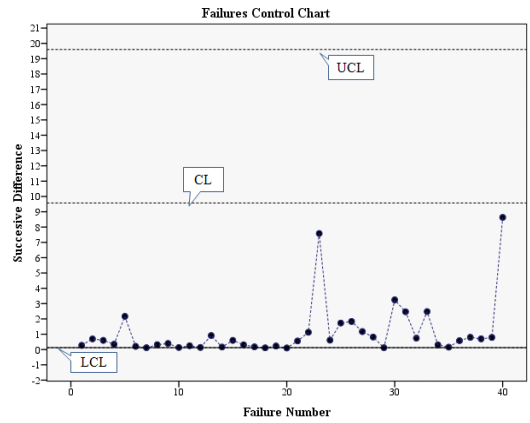
<표 4>의 모수 추정값을 적용하고 로그 포아송 실행시간 평균값함수와 <표 1>과 <표 2> 에 요약된 각 관리 한계값을 추정한 추정값은 <표 5>에 요약되었다.

<표 5> 각 모형의 관리 한계의 추정값

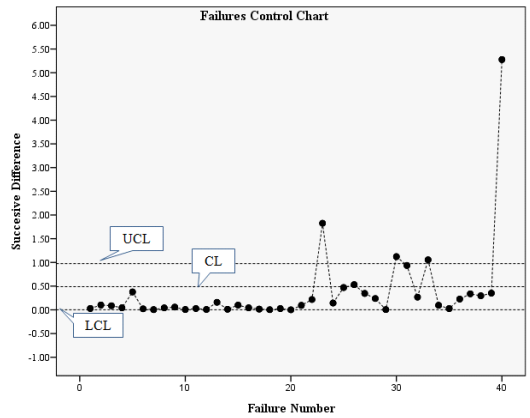
Model	UCL	LCL	CL
로그 포아송 실행시간	19.00815	0.02544	9.469393
로그-선형	0.974841	0.001318	0.488145
파레토펬포	0.170429	0.000213	0.081979

<표 6> 로그 포아송 실행시간 모형에 대한 평균값  
함수의 연속적인 차이 결과 추정값

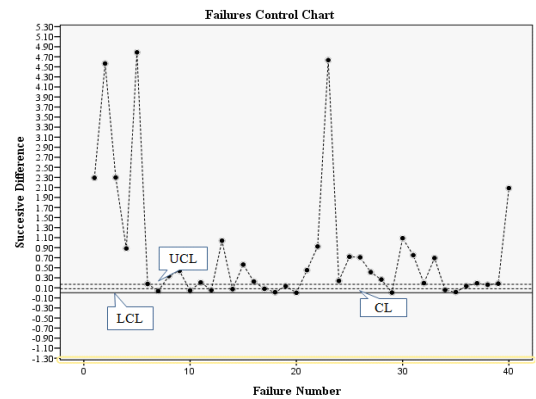
Failure Number	Failure Time ( $i$ )	$m(i)$	$m(i+1) - m(i)$ successive difference
1	5.649	0.298879024	0.172248554
2	8.92	0.471127578	0.594154721
3	20.29	1.065282299	0.499556263
4	29.955	1.564838563	0.244205678
5	34.715	1.809044241	2.066956842
6	75.95	3.876001083	0.108938267
7	78.171	3.98493935	0.022239103
8	78.625	4.007178453	0.214875139
9	83.022	4.222053592	0.296187238
10	89.114	4.51824083	0.033436623
11	89.804	4.551677453	0.14782167
12	92.86	4.699499123	0.038624592
13	93.66	4.738123715	0.813548814
14	110.655	5.551672529	0.063253355
15	111.988	5.614925884	0.498138712
16	122.545	6.113064596	0.210833668
17	127.045	6.323898264	0.077876699
18	128.712	6.401774962	0.012975425
19	128.99	6.414750387	0.129476008
20	131.768	6.544226395	0.002839301
21	131.829	6.547065697	0.457893929
22	141.712	7.004959625	1.027054351
23	164.212	8.032013977	7.481307249
24	342.85	15.51332123	0.514381149
25	356.144	16.02770238	1.628396295
26	399.144	17.65609867	1.733743483
27	446.494	19.38984215	1.073385028
28	476.644	20.46322718	0.716868813
29	497.144	21.18009599	0.01794667
30	497.661	21.19804266	3.143980417
31	591.161	24.34202308	2.370127864
32	665.644	26.71215095	0.642262872
33	686.444	27.35441382	2.381368547
34	765.944	29.73578236	0.205300401
35	772.977	29.94108276	0.057267963
36	774.944	29.99835073	0.481189587
37	791.561	30.47954031	0.698749783
38	815.978	31.1782901	0.597928851
39	837.145	31.77621895	0.691563604
40	861.945	32.46778255	8.532441007



(그림 1) 로그 포아송 실행시간 모형에 대한 관리도



(그림 2) 로그 선형 모형에 대한 관리도



(그림 3) 파레토 분포 모형에 대한 관리도

(그림 1)은 로그 포아송 실행시간 모형에 대한 평균값 함수의 연속적인 차이 결과 추정값을 이용하여 관리도를 그린 그림이다. 그리고 유사한 방법으로 (그림 2) 와 (그림 3)에도 각각 로그 선형 모형에 대한 관리도와 파레토 분포 모형에 대한 관리도를 그린 그림이다.

SPC에서는 품질변동을 조사하여 그 원인을 우연원인(Random cause)과 이상원인(Assignable cause)으로 분리시켜, 이상 원인은 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[15]. 따라서 로그 포아송 실행시간 모형 관리도에서는 <표 6>과 (그림 1)에서 보여 주듯이 7번째와 18번째 그리고 20번째와 29번째 추정값이 관리하한 *LCL*를 벗어나고 있다. 따라서 포아송 실행시간 모형을 사용하기 위해서는 이러한 4개의 자료를 판단하여 이상요인이면 즉시 제거하고 다시 또 관리한계를 설정하여 조사하면 된다. 그러나 중심선 하단 쪽에 모든 자료들이 타점되어 있기 때문에 시스템적인 접근을 통하여 자료들을 개선하여 사용하는 편이 효율적이다. 로그 선형 모형 관리도에서는 (그림 2)에서 보여 주듯이 20번째 추정값은 관리하한 *LCL* 보다 낮게 타점되어 있고 22, 30번째 그리고 33번째와 40번째 추정값이 관리상한 *UCL*보다 높게 타점되어 나타나고 있다. 그러나 런(Run), 주기, 경향이 비교적 없는 것으로 판단되므로 로그 선형 모형을 사용하기 위해서는 이러한 5개의 자료를 판단하여 이상요인이면 즉시 제거하고 다시 또 관리한계를 설정하여 조사하면 된다. 파레토 분포 모형 관리도에서는 (그림 3)에서 보여 주듯이 관리상한 *UCL*과 관리하한*LCL* 사이에 존재하는 자료가 벗어나는 자료보다 많기 때문에 이상요인과 우연요인 그리고 런(Run), 주기, 경향이 존재한다고 판단되기 때문에 파레토 분포 모형을 사용하기 위해서는 이상요인과 우연요인을 판단해야 할 뿐만 아니라 시스템적 접근 예를 들면 작업방법 개선, 작업자 훈련, 작업환경의 균일화, 계측기 정도의 향상 등을 고려해 볼 수 있다.

## 4. 결 론

본 논문에서는 신뢰성 분야에서 많이 사용되는 NHPP에 근원을 둔 로그 포아송 실행시간 모형, 로그 선형 모형 그리고 파레토 모형의 평균값 함수를 이용한 통계적 공정관리 차트를 이용한 제어 메커니즘을 제안하였다. 즉, 관리 한계를 벗어나면 공정의 이상을 발견하고 재발방지 조치를 취함으로써 공정을 안정하게 유지하여 품질의 유지 및 향상을 도모하기 위한 목적으로 이용된다. 본 연구에 사용된 모형에서 로그 포아송 실행시간 모형은 이상요인이면 즉시 제거하고 다시 또 관리한계를 설정하여 조사하거나 시스템적인 접근을 통하여 자료들을 개선하여 사용하는 편이 효율적이고 로그 선형 모형을 사용하기 위해서는 이상요인이면 즉시 제거하고 다시 또 관리한계를 설정하여 조사하면 된다.

파레토 분포 모형은 이상요인과 우연요인을 판단해야 할 뿐만 아니라 시스템적 접근 예를 들면 작업방법 개선, 작업자 훈련, 작업환경의 균일화, 계측기 정도의 향상 등을 고려해 볼 수 있다.

이 연구를 통하여 소프트웨어 개발자들은 효율적인 자료를 파악 하는데 어느 정도 도움을 줄 수 있으리라 사료 된다.

## 참고문헌

- [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 AL, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", IEEE Trans Reliab 28, pp.206-11, 1978.
- [3] Musa, J. D and Okumoto, K.(1984). "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", Proceeding the 7th International Conference on Software Engineering. pp. 230-238,

[4] Almering, V. and Genuchten, M, V and Cloudu, G. and Sonnemans, P. J. M.(2007) “ Using Software Reliability Growth Models in Practice ”. IEEE SOFTWARE. pp. 82-88.

[5] Yang, B. and Xie. M.(2000) “A study of operational and testing reliability in software reliability analysis”. RELIABILITY ENGINEERING & SYSTEM SAFETY, Vol,70, pp, 323-329.

[6] Huang, C. Y.(2005) “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.

[7] 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, pp. 121-131, August 2011.

[8] J. F. Lawless. Statistical Models and Methods for Lifetime Data. John Wiley & Sons, New York, 1981.

[9] L. Kuo and T. Y. Yang.”Bayesian Computation of Software Reliability”.Journal of the American Statistical Association, Vol.91, p p. 763-773, 1996.

[10] Musa, J. D, Iannino, A. and Okumoto, K. (1987) “Software Reliability: Measurement, Prediction, Application”. McGraw Hill, New York.

[11]Hee-Cheul KIM and Hyoung-Keun Park, “ The Comparative Study of Software Optimal Release Time Based on Burr Distribution”, International Journal of Advancements in Computing Technology, Volume 2, Number 3, pp. 119-128, 2010.

[12] 김희철, 박형근, “강도함수 특성에 근거한 소프트웨어 최적방출시기에 관한 비교 연구”, 한국산학기술학회, 제 11권 4호, pp. 1239-1247,

2010.

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

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

[15] 김희철, “공업 통계학”, 한울출판사, pp. 369-377, 2010.

---

[저 자 소 개]

---

**김희철 (Hee-cheul Kim)**



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

email : kim1458@nsu.ac.kr

**신현철 (Hyun-cheul Shin)**



1990년 2월 광운대학교 전자계산학과 졸업(공학석사)  
 2002년 2월 원광대학교 컴퓨터공학과 졸업(공학박사)  
 19945-현재 백석문화대학교 인터넷 정보학부 교수

email : hcshin@bcc.ac.kr