

대수형 학습효과에 근거한 소프트웨어 신뢰모형에 관한 통계적 공정관리 비교 연구

김경수*, 김희철**

백석문화대학교 인터넷 정보학부*, 남서울대학교 산업경영공학과**

The Assessing Comparative Study for Statistical Process Control of Software Reliability Model Based on Logarithmic Learning Effects

Kyung-Soo Kim*, Hee-Cheul Kim**

Dept. of Internet information, BaekSeok Culture University*

Dept. of Industrial & Management Engineering, Namseoul University**

요 약 소프트웨어의 디버깅 오류의 발생 시간에 의존하는 많은 소프트웨어 신뢰성 모델이 연구되었다. 소프트웨어 오류 탐색 기법은 사전에 알지 못하지만 자동적으로 발견되는 에러를 고려한 영향요인과 사전 경험에 의하여 세밀하게 에러를 발견하기 위하여 테스트 관리자가 설정해놓은 요인인 학습효과의 특성에 대한 문제를 비교 제시 하였다. 본 연구에서는 학습효과 비동질적인 유한고장모형 분석을 위한 모수 추정은 우도함수를 이용하였다. 소프트웨어 시장에 인도하기 위한 결정에 대하여 조건부 고장률은 중요한 변수가 되고 이러한 고장 모델은 실제 상황에서 많이 사용되고 있다. 통계적 공정 관리 (SPC)는 소프트웨어 오류의 예측을 모니터링 함으로써 소프트웨어의 신뢰성 향상에 크게 기여할 수 있다. 이러한 컨트롤 차트는 널리 소프트웨어 산업의 소프트웨어 프로세스 제어를 위해 사용된다. 본 연구에서는 로그 위험 학습 효과 속성의 비동질적인 포아송 과정의 평균값 기능을 사용한 컨트롤 메커니즘을 제안하였다.

주제어 : 대수형 학습효과, 비동질적 포아송 과정, 통계적 공정관리, 평균값 함수, 추세검정

Abstract There are many software reliability models that are based on the times of occurrences of errors in the debugging of software. Software error detection techniques known in advance, but influencing factors for considering the errors found automatically and learning factors, by prior experience, to find precisely the error factor setting up the testing manager are presented comparing the problem. 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). 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 logarithmic hazard learning effects property.

Key Words : Logarithmic Learning Effects, Non-Homogeneous Poisson Process, Statistical Process Control. Mean Value Function, Laplace trend test

Received 8 October 2013, Revised 8 December 2013

Accepted 20 December 2013

Corresponding Author: Hee-Cheul Kim(Namseoul University)

Email: kim1458@nsu.ac.kr

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1738-1916

1. 서론

현대사회에 있어서 인간이 의존하고 있는 시스템의 대부분은 소프트웨어 시스템(System)을 포함하고 있으며 이러한 시스템의 기능이 확대되어 소프트웨어가 대규모화, 복잡화, 다양화하고 수요량도 급속히 증대하고 있다. 그러나 시스템이 고장(Failure)이 발생하면 고장의 원인이 되는 결함(Fault)을 찾아 제거하는 디버깅 과정을 거치면서 소프트웨어에 잔존하는 결함의 수는 점차 감소하며 고장이 발생하는 시간 간격이 점차 증가함으로 소프트웨어 신뢰성은 성장한다.

소프트웨어 테스트 단계에서 소프트웨어 고장수(Number of failure)와 고장간격시간에 의해 소프트웨어 고장현상을 수리적으로 모형화 하면 소프트웨어에 대한 평가를 쉽게 할 수 있으며 신뢰도 모형에 의해 소프트웨어 고장수, 소프트웨어 고장발생간격시간, 소프트웨어 신뢰도 및 고장률 등의 신뢰성 평가측도들이 추정되어 미래의 고장시간을 예측할 수도 있다.

소프트웨어 고장시간은 수명자료가 된다. 따라서 비음(Nonnegative)의 값을 가지기 때문에 수명분포가 한정되어 있다. 시간이 지날수록 비증가 패턴을 가지는 지수분포(Exponential,)나, 혹은 와이불(Weibull), 로그정규(Lognormal), 감마분포(Gamma distribution)등을 사용하여 모형을 적용하는 것이 일반적이다. 이러한 분포들은 일반화 감마 분포(Generalized gamma distribution)의 특수한 경우로 알려져 있다[1].

그리고 최근에는 S-형태 모형을 사용하면 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 익숙해지는 학습 과정을 설명할 수 있다고 하였고[2], KIM[3]은 S-형태 모형과 비 S-형태모형의 차이를 규명하기 위하여 학습과정을 이용한 S-형태 모형과 극값 분포 모형을 비교하는 연구도 이루어졌다.

또한, 통계적 공정관리(SPC, Statistical process control)에서는 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 메카니즘을 의미한다. 즉, 공정 관리 (SPC)는 소프트웨

어 고장의 예측을 모니터링 함으로써 소프트웨어 신뢰성의 향상에 크게 기여 할 수 있다. 모니터링 하는 과정에서 이상요인과 우연요인을 분리시킬 수 있는 관리도는 널리 소프트웨어 산업의 소프트웨어 공정 관리에 사용되는 도구이다. 이 분야에서 Rao, Prasad 와 Kantham은 Half Logistics 분포를 이용한 공정관리 차트를 이용한 메카니즘을 연구하기도 하였다[4].

따라서 본 연구는 위험함수가 S-형태가 아닌 로그 선형 패턴을 적용한 유한 고장 NHPP 소프트웨어 모형에 대하여 자동적으로 발견되는 에러를 고려한 자동 에러 탐색요인(Autonomous errors-detected factor)과 사전에 설정되는 학습요인(Learning factor)으로 구성된 영향요인들(Influential factors)의 특성에 대한 공정관리 접근 방법을 비교 제시하였다.

2. 관련연구

2.1 소프트웨어 신뢰성

$N(t)$ 을 시간 t 까지 검출된 소프트웨어의 누적고장수라고 하고, $m(t)$ 를 이에 대한 기대치를 나타내는 평균값 함수(MVF, Mean Value Function)로 가정하고 $\lambda(t)$ 을 강도함수(Intensity function) (즉, t 에서의 순간 결함 검출율)이면 비동질 포아송 과정(NHPP, non-homogeneous Poisson Processes)은 누적 고장수인 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률밀도함수(Probability density function)로 알려져 있다. 즉,

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

따라서 NHPP 모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도 함수 $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[5].

$$m(t) = \int_0^t \lambda(s) ds, \quad \frac{dm(t)}{dt} = \lambda(t) \quad (2)$$

유한 고장 NHPP 모형에서는 시간 $(0, t]$ 까지 탐색되어 질 수 있는 결함의 기대값을 θ 라고 표현하면 유한 고

장 NHPP모형의 평균값 함수와 강도함수는 다음과 같이 표현할 수 있다[5, 6].

$$m(t) = \theta F(t), \lambda(t) = \theta F'(t) \quad (3)$$

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

$$x_k = \sum_{i=1}^k t_k \quad (k=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (4)$$

이라고 하면 n 번째까지 고장시점이 관찰된 고장 절단 모형일 경우에 데이터 집합 D_{x_n} 은 $\{x_1, x_2, \dots, x_n\}$ 으로 구성되며, 이 시간 절단 모형에서 θ 을 모수공간이라고 표시하면 유한고장 우도함수는 다음과 같이 알려져 있다 [5, 6].

$$L_{NHPP}(\theta | \underline{x}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp(-m(x_n)) \quad (5)$$

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

NHPP 모형에서 테스트 시점 x_n 에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간 $(x_n, x_n + t]$ (단, t 는 임무시간(Mission time)사이에서 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도(Reliability) $\hat{R}(t | x_n)$ 는 식 (6)과 같이 표현된다.[7].

$$\begin{aligned} \hat{R}(t | x_n) &= e^{-\int_{x_n}^{x_n+t} \lambda(\tau) d\tau} \\ &= \exp[-\{m(t+x_n) - m(x_n)\}] \end{aligned} \quad (6)$$

2.2 학습효과를 고려한 강도함수와 누적함수

소프트웨어 테스트 작업에 있어서 학습효과는 테스트 관리자에 의해 동일한 혹은 조작 가능한 작업이 될 수 있으므로 이러한 효과들을 어떠한 방법으로 반영하는가는 소프트웨어 신뢰성에 중요한 과정이 된다.

소프트웨어 에러들을 발견하기 위하여 자동 에러 탐색요인(Autonomous errors-detected factor) γ 과 학습요인(Learning factor) η 이 포함된 영향요인들(Influential

factors)이 고려될 수 있다. 따라서 $f(t)$ 을 t 시점에서 에러를 발견될 확률을 나타내는 확률밀도함수이고 $F(t)$ 을 $(0, t]$ 시점까지의 누적분포함수라고 가정하면 영향요인들을 고려한 모형은 다음과 같이 표현된다[8, 10].

$$f(t) = (\gamma + \eta F(t)) (1 - F(t)) \quad (7)$$

단, $\gamma > 0, \eta > 0$.

(7)식에서 자동 에러 탐색요인 γ 는 사전에 알지 못하지만 테스트 과정에서 테스트 관리자가 자동적으로 에러를 발견하는 요인이지만 학습요인 η 은 과거에 발견된 에러패턴을 바탕으로 세밀하게 에러를 발견하기 위하여 테스트 관리자가 설정해놓은 요인을 의미한다.

한편 (7)식을 로그 선형 추세를 분석하기 위하여 다음과 같이 위험함수 형태로 변경 할 수 있다[8, 9].

$$h(t) = (\gamma + \eta F(t)) \quad (8)$$

$$\text{단, } h(t) = \frac{F'(t)}{1 - F(t)} = \frac{f(t)}{1 - F(t)}$$

(8)식에서 누적함수와 확률밀도함수는 다음과 같이 수정 가능하다.

$$F(t) = \frac{h(t) - \gamma}{\eta}, \quad f(t) = F'(t) = \frac{h'(t)}{\eta} \quad (9)$$

2.3 대수형 위험함수를 이용한 NHPP 소프트웨어 모형

대수형 위험함수 다음과 같이 정의 된다[1, 10].

$$h(t) = \ln(a + bt) \quad (10)$$

단, $a, b > 0$

따라서 학습효과를 고려한 분포함수는 (9) 식과 (10) 식을 이용하면 로그선형 분포함수와 확률밀도 함수는 각각 다음과 같이 유도 된다[9, 13].

$$F(t) = \frac{h(t) - \gamma}{\eta} = \frac{\ln(a + bt)}{\eta} - \frac{\gamma}{\eta} \quad (11)$$

$$f(t) = F'(t) = \frac{h'(t)}{\eta} = \frac{b}{\eta(a+bt)} \quad (12)$$

2.4 식스 시그마에 의존된 관리 한계

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

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

소프트웨어 신뢰성 모형에 대한 누적 함수(F(t)의 값을 각각 6-sigma 기준 확률 0.99865, 0.00135, 0.5을 이용하면 관리상한(t_U), 관리하한(t_L) 중심선(t_C)을 추정할 수 있다고 하였다[4].

따라서 로그선형 분포함수 (11)식을 이용하여 t에 관하여 정리하면 다음과 같이 일반적인 관리상한은 다음과 같이 유도된다.

$$t = \frac{\exp(\eta \times 0.99865 + \gamma) - a}{b} = t_U \quad (13)$$

또한, 유사하게 중심선과 관리하한은 다음과 같이 유도된다.

$$t = \frac{\exp(\eta \times 0.5 + \gamma) - a}{b} = t_C \quad (14)$$

$$t = \frac{\exp(\eta \times 0.00135 + \gamma) - a}{b} = t_L \quad (15)$$

따라서 평균값 함수의 차이를 이용한 관리 한계, 즉 관리상한, 중심선, 관리하한은 각각 다음과 같이 정리된다[4].

$$m(t_U), \quad m(t_C), \quad m(t_L) \quad (16)$$

3. 제안된 로그선형 학습효과에 근거한 NHPP 소프트웨어 신뢰모형에 관한 통계적 공정관리 접근방법

학습효과를 고려한 유한 고장 NHPP모형의 로그 선형

모형의 평균값 함수와 강도함수는 (3)식과 (11), (12)식을 이용하면 다음과 같이 유도된다[10].

$$m(t) = \theta F(t) = \theta \left(\frac{\ln(a+bt)}{\eta} - \frac{\gamma}{\eta} \right) \quad (17)$$

$$\lambda(t) = \theta F'(t) = \theta \frac{b}{\eta(a+bt)} \quad (18)$$

따라서 우도함수는 (5)식에 (17)식과 (18)식을 대입하면 다음과 같다.

$$L_{NHPP}(\Theta | \underline{x}) = \quad (20)$$

$$\prod_{i=1}^n \left(\frac{\theta b}{\eta(a+bx_i)} \right) \exp \left[-\theta \left(\frac{\ln(a+bx_i) - \gamma}{\eta} \right) \right]$$

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

모수추정방법은 최우추정법(MLE)을 사용하였고 최우추정법을 이용하기 위한 로그 우도 함수는 (20)식과 관련하여 다음과 같이 유도된다.

$$\ln L_{NHPP}(\Theta | \underline{x}) = n \ln \theta + n \ln b - n \ln \eta - \sum_{i=1}^n \ln(a+bx_i) - \theta \left(\frac{\ln(a+bx_i) - \gamma}{\eta} \right) \quad (21)$$

본 연구에서는 로그형 모형의 특성을 유지하면서도 모수 추정을 용이하게 위하여 (21)식에서 형상모수 a=0.5으로 가정한 모형을 사용하고자 한다. 즉, θ 와 b에 대하여 편미분하여 다음과 같은 식을 만족하는 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 을 수치 해석적 방법으로 추정할 수 있다[10].

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - \left(\frac{\ln(a+bx_i) - \gamma}{\eta} \right) = 0 \quad (22)$$

$$\text{즉, } \hat{\theta} = \frac{n\eta}{\ln(a+bx_i) - \gamma}$$

$$\frac{\partial \ln L_{NHPP}(\Theta | \underline{x})}{\partial b} = \quad (23)$$

$$\frac{n}{b} - \sum_{i=1}^n \frac{x_i}{a+bx_i} - \frac{\theta x_n}{\eta(a+bx_n)} = 0$$

또한, (6)식을 이용한 신뢰도는 다음과 같다.

(24) 이 효율적임을 시사하고 있다[12, 14].

$$\hat{R}_{in}(t|x_n) = \exp \left[- \left(\frac{\theta}{\eta} (\ln(a+b(x_n+t)) - \gamma) \right) + \left(\frac{\theta}{\eta} (\ln(a+b x_n) - \gamma) \right) \right]$$

따라서 평균값 함수의 차이를 이용한 관리 한계, 즉 관리상한($m(t_U)$), 중심선($m(t_C)$), 관리하한($m(t_L)$)은 각각 다음과 같이 유도된다[4].

$$m(t_U) = \theta \left(\frac{\ln(a+b t_U)}{\eta} - \frac{\gamma}{\eta} \right) \tag{25}$$

$$m(t_C) = \theta \left(\frac{\ln(a+b t_C)}{\eta} - \frac{\gamma}{\eta} \right) \tag{26}$$

$$m(t_L) = \theta \left(\frac{\ln(a+b t_L)}{\eta} - \frac{\gamma}{\eta} \right) \tag{27}$$

$$\text{단, } t_U = \frac{\exp(\eta \times 0.99865 + \gamma) - a}{b},$$

$$t_C = \frac{\exp(\eta \times 0.5 + \gamma) - a}{b},$$

$$t_L = \frac{\exp(\eta \times 0.00135 + \gamma) - a}{b}.$$

4. 제안된 소프트웨어 고장자료 분석 및 공정관리 분석

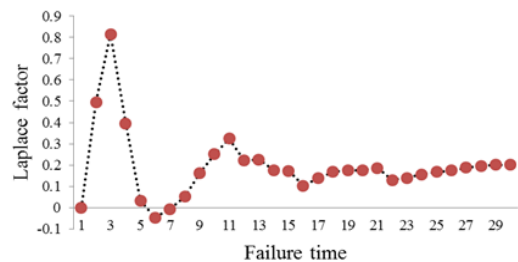
이 장에서는 소프트웨어 고장 간격 시간 자료[11] (Failure interval time data)를 가지고 제시하는 신뢰모형들을 분석하고자 한다. 이 자료의 고장 시간은 18.735 시간단위에 30번의 고장이 발생한 자료이며 <표 1>에 나열 되어 있다. 또한 제시하는 신뢰 모형들을 분석하기 위하여 우선 적용자료에 대한 추세검정이 선행 되어야 한다[12] [13].

또한 신뢰모형을 분석하기 위해서는 자료에 대한 추세검정이 선행 되어야 하는데[12], 본 연구에서는 일반적으로 추세 검정에 사용되는 라플라스 추세 검정을 사용한다.< 그림 1>은 이 결과를 보여준다. 이 그림에서 라플라스 추세 검정의 결과는 라플라스 요인(Factor)이 -2와 2사이에 존재함으로써 즉, 극단값이 존재하지 않으므로 신뢰성장(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰 성장모형을 제시하는 것

<Table 1> Software failure time

Failure Number	Failure Interval (second)	Failure Time(second)
1	0.479	0.479
2	0.266	0.745
3	0.277	1.022
4	0.554	1.576
5	1.034	2.610
6	0.949	3.559
7	0.693	4.252
8	0.597	4.849
9	0.117	4.966
10	0.170	5.136
11	0.117	5.253
12	1.274	6.527
13	0.469	6.996
14	1.174	8.170
15	0.693	8.863
16	1.908	10.771
17	0.135	10.906
18	0.277	11.183
19	0.596	11.779
20	0.757	12.536
21	0.437	12.973
22	2.230	15.203
23	0.437	15.640
24	0.340	15.980
25	0.405	16.385
26	0.575	16.96
27	0.277	17.237
28	0.363	17.600
29	0.522	18.122
30	0.613	18.735

Laplace trend test



[Fig. 1] Laplace trend test

모수추정은 최우추정법을 이용하여 추정된 Kim Hee Cheul 과 Shin Hyun Cheul[10]의 결과를 이용하였고, 영향요인에 따른 최우추정법의 결과와 관리 한계 추정값

이 <표 2>에 요약되었다.

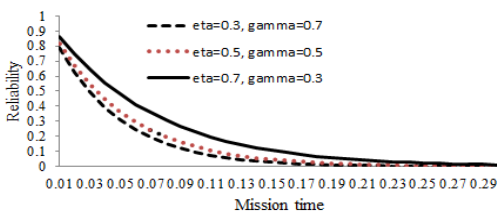
<그림 2>에서는 임무시간에 대한 신뢰도 그림에서도 η 가 증가 할수록 신뢰도가 상승하는 패턴을 보이고 있다. 즉, 소프트웨어 에러들을 발견하기 위하여 자동 에러 탐색요인 γ 보다 학습요인 η 가 큰 경우가 신뢰도가 높게 나타나고 있다.

<표 3>은 자동 에러 탐색요인 γ 보다 학습요인 η 가 작은 경우($\eta = 0.3, \gamma = 0.7$)에 평균값에 대한 연속적인 차이의 값을 계산한 표이고 그 결과값을 관리 한계값들에 적용시킨 평균값 차트가 <그림 3>에 요약 되었다.

(Table 2) The parameters estimation and control limits bases on learning factors

Learning factors		MLE	Control limits		
η	γ		$m(t_U)$	$m(t_C)$	$m(t_L)$
0.3	0.7	$\hat{b}_{MLE} = 0.0808$ $\hat{\theta}_{MLE} = 6.0568$	6.0486	3.0284	0.0082
0.5	0.5	$\hat{b}_{MLE} = 0.0613$, $\hat{\theta}_{MLE} = 6.3854$	6.3768	3.1927	0.0086
0.7	0.3	$\hat{b}_{MLE} = 0.0454$, $\hat{\theta}_{MLE} = 6.7900$	6.7808	3.3950	0.0092

Reliability Vs. Mission time



Note: eta : η , gamma : γ

[Fig. 2] Reliability Pattern of learning factors

통계적 공정관리(SPC)에서는 품질변동을 조사하여 그 원인을 우연원인(Random cause)과 이상원인(Assignable cause)으로 분리시켜, 이상 원인은 현장에서 즉각 조치를 취하여 없앨 수 있는 요인을 의미하고 우연 원인은 생산설비의 교체나 생산방식에 대한 시스템적 접근을 통하여 근본적으로 우연원인에 의한 품질변동의 크기를 축소시켜 주어야 하는 요인을 의미한다[4].

따라서 <그림 3>에서의 자동 에러 탐색요인 보다 학습요인이 작은 경우는 1번째부터 4번째 추정값이 관리상한(UCL)높게 타점되어 나타나고 있어 이러한 4개의 자료를 판단하여 이상요인 인지 우연 요인지를 파악하여 이상요인이면 즉시 제거하고 다시 또 관리한계를 설정하여 조사하면 된다.

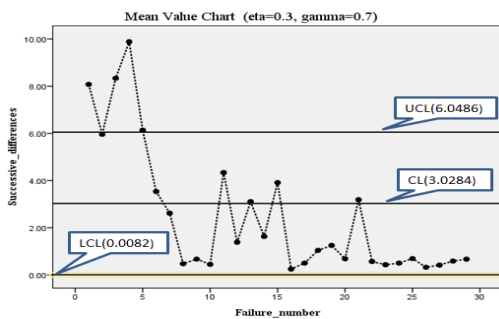
(Table 3) Successive difference of $m(i)$

Failure Number	Failure Time (i)	$m(i)$ $i = 1, 2, \dots, 29$.	$m(i+1) - m(i)$ Successive difference
1	47.9	15.6434	8.07526
2	74.5	23.7187	5.95842
3	102.2	29.6771	8.33527
4	157.6	38.0124	9.88024
5	261	47.8926	6.13624
6	355.9	54.0289	3.53557
7	425.2	57.5644	2.61685
8	484.9	60.1813	0.47536
9	496.6	60.6566	0.67134
10	513.6	61.3280	0.44941
11	525.3	61.7774	4.33814
12	652.7	66.1155	1.38825
13	699.6	67.5038	3.10653
14	817	70.6103	1.63187
15	886.3	72.2422	3.91154
16	1077.1	76.1537	0.25005
17	1090.6	76.4038	0.50356
18	1118.3	76.9073	1.04268
19	1177.9	77.9500	1.25114
20	1253.6	79.2011	0.68846
21	1297.3	79.8896	3.18842
22	1520.3	83.0780	0.56986
23	1564	83.6479	0.43250
24	1598	84.0804	0.50338
25	1638.5	84.5838	0.69378
26	1696	85.2776	0.32590
27	1723.7	85.6035	0.41927
28	1760	86.0227	0.58805
29	1812.2	86.6108	0.66939
30	1873.5	87.2802	-----

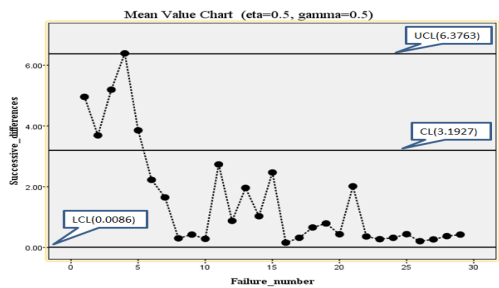
유사한 방법으로 자동 에러 탐색요인과 학습요인이 같은 경우($\eta = 0.5, \gamma = 0.5$)의 <그림 4>에서의 평균값 차트에서는 4번째 추정값이 관리상한보다 높게 타점되어 나타나고 있기 때문에 역시 4번째의 자료를 판단하여 이상요인이면 즉시 제거하고 다시 또 관리한계를 설정하여 조사하면 된다. 그러나 <그림 5>에서는 학습요인이 자동 에러 탐색요인 보다 큰 경우($\eta = 0.7, \gamma = 0.3$)

을 적용한 그림이다. 이 그림에서는 모든 추정값이 관리 상한선과 관리 하한선(LCL) 사이에 타점되어 나타나고 있기 때문에 이상요인과 우연 요인이 나타나지 않은 관리 상태로 나타나고 있다.

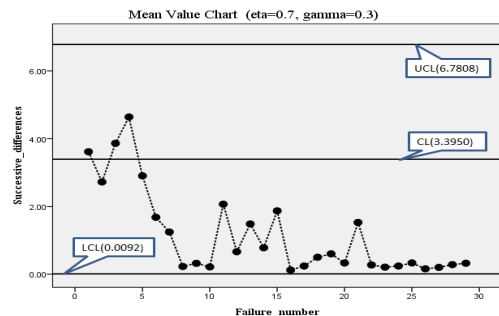
즉, 자동 에러 탐색요인이 학습요인보다 크거나 같은 경우에는 관리 상태가 되지 않아 이상요인과 우연요인 그리고 런(Run), 주기, 경향이 존재 유무를 판단하여 이상요인으로 판명되는 되는 경우에는 시스템적 접근 예를 들면 작업방법 개선, 작업자 훈련, 작업환경의 균일화, 계측기 정도의 향상 등을 고려해 볼 수 있다.



[Fig. 3] Mean Value Chart($\eta=0.3, \gamma=0.7$)



[Fig. 4] Mean Value Chart($\eta=0.5, \gamma=0.5$)



[Fig. 5] Mean Value Chart($\eta=0.7, \gamma=0.3$)

결과적으로 사전의 자동 에러 탐색요인보다 학습요인을 비중 있게 설정하면 효율적인 통계적 공정 분석을 통한 소프트웨어의 질을 개선 할 수 있다.

5. 결론

대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피 할 수 없는 상황이 현실이다.

본 연구에서는 신뢰성 분야에서 많이 사용되는 대수형 학습효과에 근거한 소프트웨어 NHPP 신뢰모형의 평균값 함수를 이용한 통계적 공정관리 차트 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 로그선형 학습효과모형의 결과는 자동 에러 탐색요인이 학습요인보다 크거나 같은 경우에는 관리 상태가 되지 않아 이상요인과 우연요인 이 발생할 가능성이 있지만 학습요인이 자동 에러 탐색요인 보다 큰 경우 모든 추정값이 관리 상한과 관리 하한 사이에 타점되어 나타나고 있기 때문에 이상요인과 우연 요인이 나타나지 않은 관리 상태로 나타나고 있다.

따라서 사전의 자동 에러 탐색요인보다 학습요인을 비중 있게 설정하면 효율적인 통계적 공정 분석을 통한 소프트웨어의 질을 개선 할 수 있다.

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

REFERENCES

- [1] J. F. Lawless. "Statistical Models and Methods for Lifetime Data". John Wiley & Sons, New York, 1981.
- [2] 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.

[3] Hee-Cheul, Kim. "The Comparative Study of NHPP Delayed S-Shaped and Extreme Value Distribution Software Reliability Model using the Perspective of Learning Effects", International Journal of Advancements in Computing Technology(IJACT) Vol. 5, No. 9, pp. 1210-1218, 2013.

[4] 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.

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

[6] Hee-Cheul KIM and Hyoung-Keun Park, "Exponentiated Exponential Software Reliability Growth model," International Journal of Advancements in Computing Technology, Volume 1, Number 2, pp.57-64, 2009.

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

[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] Hee-Cheul KIM, "The Comparative Study of NHPP Half-Logistic Distribution Software Reliability Model using the Perspective of Learning Effects," Journal of Next Generation Information Technology, Volume 4, Number 8, pp. 132-139, 2013.

[10] Kim Hee Cheul and Shin Hyun Cheul, "The Comparative Study for NHPP Software Reliability Model based on the Property of Learning Effect of Log Linear Shaped Hazard Function," Journal of Information and Security, Volume 12, Number 3, pp. 19-26, 2012. 06.

[11] Y. HAYAKAWA and G. TELFAR "Mixed Poisson-Type Processes with Application in Software Reliability," Mathematical and Computer

Modelling, 31, pp. 151-156, 2000

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

[13] Kim, Hee-Cheul, "The Comparative Study of NHPP Extreme Value Distribution Software Reliability Model from the Perspective of Learning Effects," Journal of the Korea Society of Digital Industry and Information Management, Volume7, Number 2, pp. 1-8, 2011.

[14] Hee-Cheul, Kim. "The Comparative Study of NHPP Delayed S-Shaped and Extreme Value Distribution Software Reliability Model using the Perspective of Learning Effects", International Journal of Advancements in Computing Technology(IJACT) Vol. 5, No. 9, pp. 1210-1218, 2013.

김 경 수(Kim, Kyung Soo)



- 2001년 8월 순천대학교 전산학과 (공학 박사)
- 2005년 3월 ~ 2007년 2월 : VCU DBLab Visiting Scholar
- 1998년3월~현재: 백석문화대학교 인터넷정보학부 교수
- 관심분야 : 소프트웨어신뢰성 공학, 정보보안, 웹프로그래밍

· E-Mail : kkskim@bscu.ac.kr

김 희 철(Kim, Hee Cheul)



- 1992년 2월 : 동국대학교 통계학과 (이학 석사)
- 1998년 8월 : 동국대학교 통계학과 (이학박사)
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수
- 관심분야 : 소프트웨어신뢰성 공학, 전산 통계

· E-Mail : kim1458@nsu.ac.kr