The Korean Communications
in Statistics Vol. 12 No. 3, 2005
pp. 697-704

Optimization of Software Cost Model with Warranty and Delivery Delay Costs[1)]

Chong Hyung Lee[2)], Kyu Beom Jang[3)], and Dong Ho Park[4)]

Abstract

Computer software has gradually become an indispensable elements in many aspects of our daily lives and an important factor in numerous systems. In recent years, it is not unusual that the software cost is more than the hardware cost in many situations. In addition to the costs of developing software, the repair cost resulting from the software failures are even more significant. In this paper, a cost model with warranty cost, time to remove each fault detected in the software system, and delivery delay cost is developed. We use a software reliability model based on non-homogeneous Poisson process(NHPP). We discuss the optimal release policies to minimize the expected total software cost. Numerical examples are provided to illustrate the results.

Keywords : Non-homogeneous Poisson Process, optimal software release time, software cost model, warranty cost, scheduled software delivery time

1. 서론

컴퓨터의 고장은 가전제품의 고장을 일으켜 생활에 불편을 줄 수가 있으며 은행체계 등의 혼잡을 야기하여 경제적인 손실을 주기도 하며, 더 나아가서는 항공체계나 병원시스템의 고장으로 생명까지도 위협할 수가 있다. 컴퓨터의 주된 고장원인 중의 하나는 소프트웨어 고장에 의한 것으로 소프트웨어의 신뢰성 향상은 현대의 소프트웨어 개발에 있어서 매우 중요한 요인들 중 하나가 되었다. 현재 소프트웨어의 품질을 보증하기 위해 일정한 기간동안에 소프트웨어가 고장이 없이 작동하는 확률로서 정의되는 소프트웨어의 신뢰성이 널리 사용되고 있다. 소프트웨어의 고장은 소프트웨어에 내재된 결함에 의해서 발생되는 것으로, 정상적인 프로그램 운용에서 받아들여질 수 없는 작동오류로 정의된다. 일반적으로 결함 없는 완전한 소프트웨어를 출하하는 것은 불가능하기

1) 본 논문은 2005 년도 건양대학교 학술 연구비 지원에 의하여 이루어진 것임.
본 논문은 2004 년도 한림대학교 교비 연구비(HRF-2004-27)에 의하여 연구되었음.
2) Assistant Professor, Department of Computer Engineering, Konyang University, Nonsan 320-711, Korea
Email : chlee@konyang.ac.kr
3) Doctoral Student, Department of Statistics, Hallym University, Chunchon 200-702, Korea
4) Professor, Department of Statistics, Hallym University, Chunchon 200-702, Korea

때문에 소프트웨어의 출하시점을 결정하는 문제는 프로그램 개발자들에게 있어서 대단히 중요한 관심사 중의 하나이다. 만약 테스트기간이 너무 짧을 경우에는 낮은 신뢰성으로 인하여 빈번한 고장이 발생하며 이를 수정하기 위해 많은 수리비용이 필요하다. 또한 테스트기간이 너무 길게 지속될 경우 신뢰성은 증가하나 테스트비용의 과다 소요와 제품의 출하가 늦어지게 된다. 그러므로 최적의 출하시점을 결정하는 것은 매우 중요하다.

Goel과 Okumoto (1979)는 NHPP에 기초한 소프트웨어 성장 모형을 제안하였고 Yamada, Tokuno와 Osaki (1992), Pham (1996)은 불완전수리를 반영하여 Goel-Okumoto의 NHPP 모형을 확장하였다. 비용모형에 관한 연구에서 Yamada와 Osaki (1985)는 기대비용과 요구되는 신뢰수준을 동시에 만족하는 최적 출하시점에 대한 문제를 다루었으며, Yun과 Bai (1990), Pham (1996), Lee, Nam과 Park (2002)은 소프트웨어 수명주기에 랜덤수명주기를 반영한 비용모형을 제안하고 최적 출하시점에 대한 결정문제를 다루었다.

본 논문에서는 Goel-Okumoto의 NHPP 모형에 기초하여 소프트웨어 수명주기에서 고려할 수 있는 여러 가지 비용에 관한 소프트웨어 비용 모형과 최적의 출하시점을 결정하는 방법을 제안하려고 한다. 2절에서는 본 논문에서 사용하는 기호와 가정을 기술하며, 3절에서는 보증기간과 다양한 소요비용을 반영한 소프트웨어 비용모형을 제시한다. 4절에서는 최적 출하시점을 결정하는 방법을 제안하고 5절에서는 제안된 출하시점 결정방법을 설명하기 위한 수치예제를 제공한다.

2. 기호와 가정

2.1 기호

$c_1(c_2)$ 테스트기간 동안의 단위시간당 테스트(결함제거)비용

c_3 보증기간 동안의 단위시간당 결함제거비용

$N(t)$ 시간 t까지 발견된 결함수, $m(t) = E[N(t)]$

t^* 소프트웨어 최적테스트 기간

Y_i 테스트기간 동안의 i번째 결함을 제거하는데 필요한 시간을 나타내는 확률변수

W_i 운용단계의 보증기간 동안의 i번째 결함을 제거하는데 필요한 시간을 나타내는 확률변수

$t_W(t_S)$ 보증(배달약속)기간

$c_p(t)$ 배달이 t시간 지연되는데 따르는 비용

$C_j(t)$ 유형 j의 소요비용, $j = 1, 2, 3, 4,$ $C(t) = \sum_{j=1}^{4} C_j(t)$

2.2 가정

1. 테스트 기간동안에 결함을 제거하는 비용은 그 기간동안 발견된 모든 결함을 제거하는데 소요되는 총 시간에 비례한다.

2. 보증기간 동안에 결함을 제거하는 비용은 구간 $[t, t_W]$동안에 발견된 모든 결함을 제거하는데 소요되는 총 시간에 비례한다.
3. 테스트기간과 보증기간 동안에 결함을 제거하는데 소요되는 최대 허용시간을 가정하며 각각 평균이 μ_Y와 μ_W인 절단된 지수분포(truncated exponential distribution)를 따른다.
4. 배달계획에 따른 시간 t_S 보다 테스트기간이 더 길어지면 배달지연에 따른 추가비용이 소요된다.

3. 소프트웨어 비용 모형

이 절에서는 2.2 절에서 가정한 내용에 따라 소프트웨어의 비용모형을 유도한다. $C_1(t)$, $C_2(t)$, $C_3(t)$, $C_4(t)$가 각각 시점 t까지 소요되는 테스트비용, 테스트기간 동안의 결함제거비용, 보증기간 동안의 결함제거비용 및 배달지연 손실비용을 나타낸다고 하면 수명주기 동안의 전체 소프트웨어 비용 $C(t)$에 대한 기대값은 $EC(t) = EC_1(t) + EC_2(t) + EC_3(t) + EC_4(t)$에 의하여 계산될 수 있다. 또한 소프트웨어 비용모형을 정립하기 위하여 Goel-Okumoto의 NHPP 모형의 평균고장수, $m(t)$를 사용하였다. Goel-Okumoto의 NHPP 모형의 $m(t)$는 다음과 같다.

$$m(t) = a(1 - e^{-bt}).$$

여기에서 a는 테스트 시작시점에서 소프트웨어에 내재되어 있는 결함수이고 b는 결함의 발견율을 나타낸다.

각각의 비용에 대한 기대값은 다음과 같이 계산된다.

1. 테스트 비용 $EC_1(t) = c_1 t$.
2. 테스트기간 동안의 기대되는 결함제거비용
$$EC_2(t) = E\left[c_2 \cdot \sum_{i=1}^{N(t)} Y_i\right] = c_2 \cdot E[N(t)] \cdot E[Y_i] = c_2 \cdot m(t) \cdot \mu_Y.$$
여기에서 μ_Y는 확률변수 Y_i의 기대값이다.
3. 보증기간 동안의 기대되는 결함제거비용
$$EC_3(t) = E\left[c_3 \cdot \sum_{i=N(t)}^{N(t+t_W)} W_i\right] = c_3 \cdot E[N(t+t_W) - N(t)] \cdot E[W_i]$$
$$= c_3 \cdot [m(t+t_W) - m(t)] \cdot \mu_W.$$
여기에서 μ_W는 확률변수 W_i의 기대값이다.
4. $[t_S,\ t]$동안의 배달지연에 따른 기대비용
$$EC_4(t) = I(t) \cdot c_p(t - t_S).$$

여기에서 배달계획에 따른 시간 t_S는 고정된 값이며 배달지연에 따른 기대비용을 반영하기 위하여 $0 \le t_S \le t$로 가정한다. 그리고

$$I(t) = \begin{cases} 1, & t \ge t_S \\ 0, & t < t_S \end{cases}$$

이며, 배달지연에 따른 비용함수는 선형증가인 경우($c_p(t) = k_1 t,\ k_1 > 0$)와 지수증가인 경우 ($c_p(t) = k_2(e^{\gamma t} - 1),\ \gamma > 0, k_2 > 0$)로 나누어 고려한다.

그러므로 소프트웨어의 기대비용은 다음과 같이 계산될 수 있다.

$$\begin{aligned} EC(t) &= EC_1(t) + EC_2(t) + EC_3(t) + EC_4(t) \\ &= c_1 t + c_2\mu_Y m(t) + c_3\mu_W[m(t+t_W) - m(t)] + I(t)c_p(t - t_S) \end{aligned} \tag{3.1}$$

4. 소프트웨어 최적출하정책

소프트웨어의 소요비용을 최소로 하는 최적출하시점을 결정하기 위하여 식 (3.1)로부터 다음과 같은 식을 얻어낼 수 있다.

$$\begin{aligned} y(t) &= \frac{d}{dt}EC(t) \\ &= c_1 + c_2 ab\mu_Y e^{-bt} + c_3 ab\mu_W e^{-bt}(e^{-bt_W} - 1) + \frac{d}{dt}I(t)c_p(t - t_S). \end{aligned}$$

$$w(t) = \frac{d^2}{dt^2}EC(t) = e^{-bt}[u(t) - K].$$

여기에서

$$u(t) = e^{bt} \cdot \frac{d^2}{dt^2}I(t)c_p(t - t_S), \qquad K = ab^2 c_2\mu_Y + ab^2 c_3\mu_W(e^{-bt_W} - 1).$$

본 논문에서는 배달지연에 따른 기대비용을 나타내기 위해 사용된 비용함수 $c_p(t)$를 두 가지 경우로 4.1절, 4.2절에서 각각 나누어 고려하였다. 첫번째는 비용함수가 시간에 비례하는 경우이고 두번째는 시간에 따라 지수적으로 증가하는 경우이다.

4.1 $c_p(t)$가 선형증가함수인 경우

배달지연에 따른 비용함수가 시간에 비례하는 경우를 고려하여 $c_p(t) = k_1 t$를 반영하고 소프트웨어 비용모형의 패턴을 파악하기 위하여 $w(t)$를 유도한다.

$$w(t) = e^{-bt}ab^2[c_3\mu_W(1 - e^{-bt_W}) - c_2\mu_Y] \tag{4.1}$$

식 (4.1)에서, 만약 $c_3\mu_W(1 - e^{-bt_W}) \ge c_2\mu_Y$이면 모든 $t(\ge t_S)$에 대하여 $y(t)$는 증가함수가 되며 만약 $c_3\mu_W(1 - e^{-bt_W}) < c_2\mu_Y$이면 $y(t)$는 감소후 증가하는 형태의 함수가 된다. 각각의 경우

에 대해 기대되는 소요비용 $EC(t)$를 최소로 하는 최적의 테스트기간 t^*를 찾는 방법은 다음과 같이 요약할 수 있다.

정리 1. c_1, c_2, c_3, μ_Y, μ_W, t_W, t_S가 주어있다면, 소프트웨어 기대비용을 최소로 하는 최적출하시점, t^*는 다음과 같이 얻어진다.

(a) $y(t)$가 증가함수인 경우

(1) $y(t_S) \geq 0$이면 $t^* = t_S$이다.

(2) $y(t_S) < 0$이고 $t \in (t_S, t_1]$에서 $y(t) \leq 0$이고 $t \in (t_1, \infty)$에서 $y(t) > 0$이 성립되는 값 t_1이 존재하므로 $t^* = t_1$이다.

(b) $y(t)$가 감소후 증가하는 경우

(1) $y(t_S) > 0$ 이고 모든 t에 대해 $y(t) \geq 0$이면 $t^* = t_S$이다.

(2) $y(t_S) > 0$ 이고 $t \in (t_S, t_2]$에서 $y(t) \geq 0$이고 $t \in (t_2, t_2']$에서 $y(t) \leq 0$, $t \in (t_2', \infty)$에서 $y(t) > 0$인 t_2와 t_2'이 존재하므로

만일 $EC(t_S) \leq EC(t_2')$이면, $t^* = t_S$,

만일 $EC(t_S) > EC(t_2')$이면, $t^* = t_2'$이다.

(3) $y(t_S) \leq 0$이고 $t \in (t_S, t_3]$에서 $y(t) \leq 0$이고 $t \in (t_3, \infty)$에서 $y(t) > 0$이 성립되는 값 t_3이 존재하므로 $t^* = t_3$이다.

증명. $y(t)$가 감소후 증가하는 (b)-(2)의 경우, $y(t_S) > 0$이면 $t \in (t_S, t_2]$에서 $y(t) \geq 0$이므로 $EC(t)$는 증가하고, $t \in (t_2, t_2']$에서 $y(t) \leq 0$이므로 $EC(t)$는 감소하게 되며 $t \in (t_2', \infty)$에서 $y(t) > 0$이므로 $EC(t)$는 다시 증가하게 된다. 그러므로 소프트웨어 수명주기 동안에 소요되는 기대비용을 최소로 하는 시점 t^*는 $EC(t_S)$과 $EC(t_2')$을 비교하여 작은 값을 갖는 시점이다. 다른 경우에도 유사한 방법으로 출하시점 t^*가 존재함을 보일 수 있다.

4.2 $c_p(t)$가 지수증가함수인 경우

이 절에서는 배달지연에 따른 비용함수가 지수적으로 증가하는 경우를 고려하기 위하여 $c_p(t) = k_2(e^{\gamma t} - 1)$를 반영하고 소프트웨어 비용모형의 패턴을 파악하기 위하여 아래의 식을 유도한다.

$$\frac{d}{dt}u(t) = k_2\gamma^2 e^{-\gamma t_S} e^{(b+\gamma)t}(b+\gamma) \tag{4.2}$$

식 (4.2)에서, $b > 0$, $k_2 > 0$, $\gamma > 0$이고 모든 $t(>0)$에 대하여 $d[u(t)]/dt > 0$이므로 $u(t)$는 $u(0) > 0$인 증가함수이다. 따라서 주어진 $u(t)$의 형태와 $w(t)$에 주어진 K를 비교하여 $y(t)$의 형태를 알 수 있다. 만약 $u(t_S) \geq K$이면 모든 $t(\geq t_S)$에 대하여 $y(t)$는 증가함수가 되며 만약

$u(t_S) < K$이면 $y(t)$는 감소후 증가하는 형태의 함수가 된다. 이러한 그래프 형태를 갖는 $y(t)$인 경우에 대하여 $EC(t)$를 최소로 하는 시점 t^*는 정리 1에서 제시된 경우와 동일한 방법으로 구할 수 있다.

5. 수치예제

다음의 수치예제는 앞에서 제안된 결과를 이용하여 비용모형에서 사용된 여러 가지 계수들에 따른 기대비용과 최적출하시점을 계산한 결과들이다. Goel-Okumoto의 NHPP 모형에서 사용된 모수들에 대한 추정값은 <표 1>의 고장자료를 이용하여 최우도추정법에 의하여 $\hat{a} = 142.32$, $\hat{b} = 0.1246$으로 얻어진다.

<표 1> 고장수와 누적 고장수

시간	1	2	3	4	5	6	7	8	9	10	11	12	13
고장수	27	16	11	10	11	7	2	5	3	1	4	7	2
누적 고장수	27	43	54	64	75	82	84	89	92	93	97	104	106
시간	14	15	16	17	18	19	20	21	22	23	24	25	
고장수	5	5	6	0	5	1	1	2	1	2	24	25	
누적고장수	111	116	122	122	127	128	129	131	132	134	135	136	

수치예제에서 사용되는 소프트웨어 비용모형의 모수는 다음과 같이 가정한다.

$$c_1 = 1,\ c_2 = 5,\ c_3 = 10,\ t_W = 10,\ t_S = 3,\ \mu_Y = 0.1,\ \mu_W = 0.3.$$

<표 2>와 <표 3>은 배달지연에 따른 비용함수가 선형증가인 경우와 지수증가인 경우에 있어 최적출하시점을 계산한 수치를 각각 보여주고 있다. 수치예제를 통해서 k_1과 k_2의 값이 증가하면 소프트웨어 기대비용이 증가하다가 감소하는 것을 볼 수 있고 발견된 결함의 평균수와 최적출하시점은 감소하는 결과를 볼 수 있다.

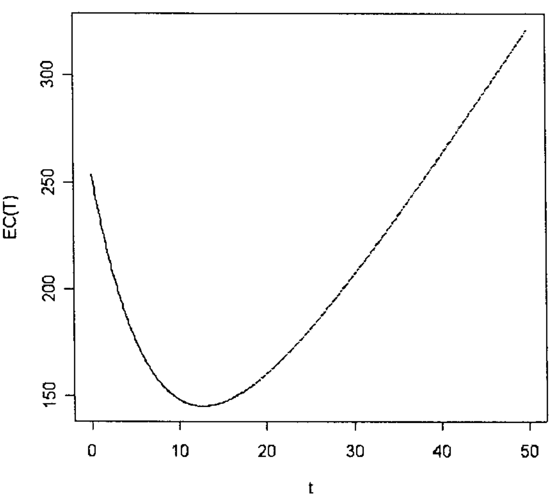
<표 2> 배달 지연에 따른 비용함수가 선형증가인 경우($c_p(t) = k_1 t$)

배달지연에 따른 비용함수 $c_p(t) = k_1 t$	최적출하시점 t^*	소프트웨어 기대비용 $EC(t^*)$	발견된 결함의 평균수 $m(t^*)$
(k_1)			
0.1	26.2681	107.883	136.9272
0.5	23.7789	113.867	134.9661
1	21.4701	120.152	132.5149
3	15.9071	136.891	122.7097
5	12.6530	145.232	112.9047
7	10.3441	148.119	103.0990
10	7.7883	145.114	88.3919
20	2.5987	94.272	39.3664

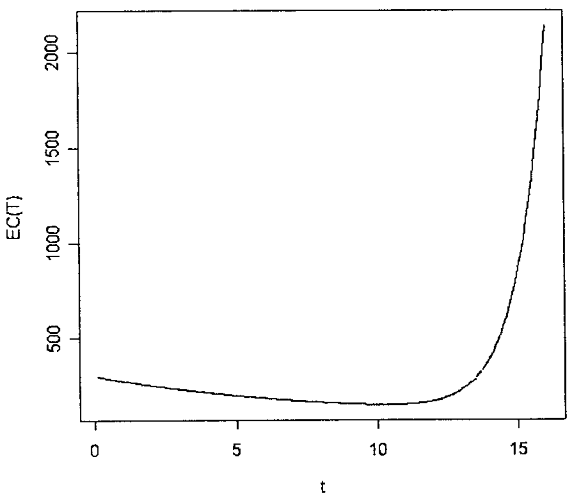
<표 3> 배달 지연에 따른 비용함수가 지수증가인 경우($c_p(t) = k_2(e^{\gamma t} - 1)$, $\gamma = 1$로 가정)

배달지연에 따른 비용함수 $c_p(t) = k_2(e^t - 1)$	최적출하시점 t^*	소프트웨어 기대비용 $EC(t^*)$	발견된 결함의 평균수 $m(t^*)$
(k_2)			
0.1	13.7462	131.064	116.6506
0.5	12.3480	138.260	111.7653
1	11.7439	141.553	109.3767
3	10.7844	146.296	105.1931
5	10.3374	147.763	103.0666
7	10.0428	148.171	101.599
10	9.7303	147.835	99.982
20	9.1224	143.361	96.651

<그림 1>은 배달지연에 따른 비용함수가 선형증가인 경우($k_1 = 5$)에서의 소프트웨어 기대비용의 그래프로서 처음에는 감소하다가 일정시점 이후 증가하는 패턴을 보여주고 있다. <그림 2>는 배달지연에 따른 비용함수가 지수증가인 경우($k_2 = 5$, $\gamma = 1$)에서의 소프트웨어 기대비용의 그래프로서 <그림 1>과 유사한 패턴을 가지고 있으며 일정시점 이후 급격하게 증가하는 것을 보여주고 있다.



<그림 1> 배달 지연에 따른 비용함수가 선형증가인 경우에서의 소프트웨어 기대비용 ($c_p(t) = 5t$)



<그림 2> 배달 지연에 따른 비용함수가 지수증가인 경우에서의 소프트웨어 기대비용 ($c_p(t) = 5(e^t - 1)$)

참고문헌

[1] Goel, A.L. and Okumoto, K.(1979). Time-dependent error-detection rate model for software reliability and other performance measures, *IEEE Transactions on Reliability*, vol 28, 206-211.

[2] Lee, C.H., Nam, K.H. and Park, D.H.(2002). Software profit model under imperfect debugging and optimal software release policy, *IEICE Transactions on Information and Systems*, vol E85-D No.5, 833-838.

[3] Pham, H.(1996) A software cost model with imperfect debugging, random life cycle and penalty cost, *International Journal of Systems Science*, vol 27, 455-463.

[4] Yamada, S., Tokuno, K. and Osaki, S.(1992). Imperfect debugging models with fault introduction rate for software reliability assessment, *International Journal of Systems Science*, vol 23, 2241-2252.

[5] Yamada, S. and Osaki, S.(1985). Cost-reliability optimum release policies for software systems, *IEEE Transactions on Reliability*, vol 34, 422-424.

[6] Yun, W.Y. and Bai, D.S.(1990). Optimum software release policy with random life cycle, *IEEE Transactions on Reliability*, vol 39, 167-170.

[2005년 3월 접수, 2005년 10월 채택]