대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일~4일

보수정책을 고려한 소프트웨어 출시 시기의 결정
Determination of software release time considering maintenance policy

나일용*, 이진승**, 이창훈**

* 국방과학연구소
** 서울대학교 산업공학과

Abstract

본 연구에서는 소프트웨어의 출시 이후 주기적인 보수(periodic maintenance)를 고려한 출시 시기 결정 방법을 제시하였다. 출시시기 결정방법에 관한 연구는 소프트웨어의 신뢰성 확보, 사용자 편의성 등에 관한 연구와 더불어 중요한 연구 분야로 여겨지는 분야이다. 일반적으로 소프트웨어는 출시 이후에도 패치(patch), 서비스팩(service pack)등을 통해 지속적인 보수가 이루어지기 때문에, 출시 이후의 보수를 고려하여 출시시기를 결정하는 방법론이 필요하다. 이를 위해, 출시 이후의 보수정책을 반영한 소프트웨어 신뢰성 성장 모형(software reliability growth model)을 도출하였다. 위 모형을 기반으로 하여 비용과 신뢰성에 근거한 소프트웨어 출시 시점을 결정해 보았다. 제시된 모형의 타당성을 검증하기 위해 예제를 통해 기존의 논문에서 제시되었던 결과들과 비교분석을 해 보았다. 본 연구의 결과는 기존 연구에서 고려되지 않았던 보수 정책을 고려함으로써, 보다 현실에 가까운 모형을 제시하였다는 점에서 의의를 찾을 수 있고, 출시 후 보수 시점, 보수 정책 등의 결정에도 기여할 수 있을 것이다.

1. 서론

소프트웨어 산업의 성장에 따라, 과거와는 달리 출시 이후의 성능 보장 문제와 보수에 관한 문제가 소프트웨어의 뛰어난 성능만큼이나 중요한 문제로 인식되고 있다. 또한, 소프트웨어의 보수 정책은 출시시기에 영향을 줄 만큼 중요한 문제이기도 하다. 그러나, 기존의 출시시기 결정에 관한 연구[1-4]를 통해 살펴보면 테스트 기간동안의 소프트웨어의 신뢰성 평가에만 초점을 맞추고 있으며, 출시 이후의 보수에 관한 연구 및 보수 정책에 대해서는 제시되고 있지 않다.

하드웨어의 경우 제품 출시 이후 제품을 사용하는 시간에 따라 노화가 이루어지는 것이 대부분이다. 그러나 소프트웨어의 경우 출시 이후 또는 마지막 보수가 이루어진 후 시간이 지나더라도 항상 마지막 보수가 이루어진 시점의 고장률을 가지게 된다. 따라서 Yamada가 구해냈던 최적 출시 시점에서는 의도한 만큼의 신뢰성을 확보하지 못하는 결과를 가져오게 된다[2]. Xie & Yang은 그들의 논문에서 과거 Yamada가 비용/신뢰성을 고려한 최적 출시 시기 결정에 대하여 이러한 문제점을 지적하고 있으며, 올바른 비용/신뢰성 분석을 하기 위해서는 신뢰성 함수에 테스팅 단계의 고장률 함수가 아닌 실제 사용자가 사용하는 단계의 고장률 함수를 사용해야 한다고 주장하였다[4]. 그러나, 현실적으로 생각해 볼 때, 소프트웨어라고 할지라도 출시 이후 보수가 이루어지지 않는다는 가정은 문제가 있다. 최근의 추세를 볼 때, 대부분의 상용 소프트웨어들이 출시 이후에도 일정한 시점까지, 이를테면 소프트웨어의 수명주기가 다 할 때까지, 서비스 팩이나 패치 등을 통해서 보수를 해주고 있는 것이 사실이다. 비단, 상용소프트웨어가 아니라고 할지라도, 정도의 차이는 있을지라도 보수가 한번도 이루어지지 않는다고 할 수는 없다.

본 연구는 위에서 제시한 문제점들을 해결하기 위하여 출시 이후의 고장률이 주기적인 패치에 의해 결정되는 모형을 제시하였다. 출시 이후 패치로 인한 고장률의 변화가 직접적으로 비용에 영향을 주기 때문에 좀더 정확한 출시시기 결정이 가능해 진다. 그리고 제시된 모형의 신뢰성 함수와 비용 함수를 통해

최적 출시 시기를 결정함으로써, 소프트웨어 출시에 관한 보다 현실적인 정책 제시를 할 수 있게 된다.

2. 보수를 고려한 소프트웨어 신뢰성 성장 모형

기호 정의

$C(T_R)$	비용함수
$R(T_R)$	신뢰성함수
$m(t)$	GO 모형의 평균값 함수
$\lambda(t)$	GO 모형의 고장률 함수
R_0	최소 신뢰성 기준
T_R	소프트웨어 출시 시점
TM	출시 후 보수 회수
W	소프트웨어 수명주기(소프트웨어 신뢰성 보증 기간)
c_1	소프트웨어 출시 전 오류 수정 비용
c_2	소프트웨어 patch 비용
c_3	소프트웨어 출시 후 오류 수정 비용
c_4	소프트웨어 출시 전 오류 발견 비용
c_5	소프트웨어 출시 후 오류 발견 비용

소프트웨어 신뢰성 성장 모형으로 일반적으로 사용되고 있는 GO 모형은 소프트웨어의 고장(failure)은 남아있는 오류(error)의 개수에 비례한다는 기본 가정으로 만들어진 NHPP 모형이며, 고장의 평균값 함수 $m(t)$와 고장률 함수 $\lambda(t)$는 식 (1)과 (2)로 표현된다[5]. 여기서 a는 소프트웨어가 초기에 가지고 있는 오류의 개수, b는 남아있는 오류 하나당 고장을 유발시킬 수 있는 비례상수이다.

$$m(t) = a(1-e^{-bt}) \tag{1}$$
$$\lambda(t) = abe^{-bt} \tag{2}$$

본 논문에서 제시되는 모형은 기존의 GO 모형의 가정에 다음의 세 가지 가정을 첨가하여 만들어진다.

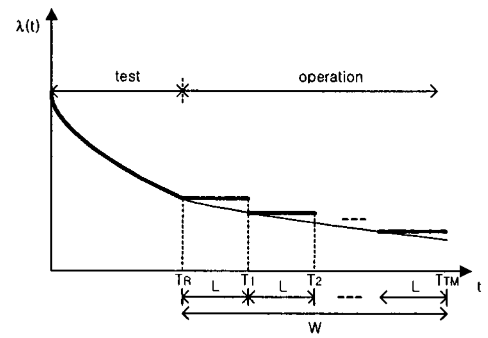
가정

① 소프트웨어는 출시 이후 일정시간 W 동안 품질을 보증한다

② 출시 후 일정 시간간격 L을 두고 보수(패치) 작업을 한다

③ 출시 후에도 내부적으로는 debugging 작업을 계속 진행한다.

출시 후 보수 정책을 고려한 소프트웨어 신뢰성 성장 모형은 [그림 1]과 같다. 주기 L마다 보수가 이루어 지는 것을 가정하고 있기 때문에 출시 후에는 보수주기 L마다 고장률이 감소하는 계단함수의 모양을 가지게 된다. 보수 후 감소량은 기존의 GO 모형을 따르는 것으로 가정을 하였다.



[그림 1] 출시 후 보수를 고려한 SRGM

소프트웨어 신뢰성이란 '특정 시점에서부터 일정 시간동안 고장이 발생하지 않을 확률'을 의미한다[6]. 즉, T시점 이후 x시간동안의 소프트웨어 신뢰성은 (3)과 같이 표현된다.

$$R(x|T) = \exp(-[m(T+x)-m(T)]) \tag{3}$$

그런데 위 식에서 $m(T+x)-m(T)$는 [그림 1]의 고장률 $\lambda(t)$의 그래프에서 $[T, T+x]$사이의 적분값과 같다. 즉, $\lambda(t)$의 $[T, T+x]$사이의 넓이를 S라 할 때 신뢰성은 아래와 같이 나타낼 수 있다.

$$R(x|T) = \exp(-S) \tag{4}$$

따라서 출시 후 보수를 고려한 보증기간 동안의 소프트웨어의 신뢰성은 아래와 같이 얻어진다.

$$R(W|T_R) = \mathrm{Exp}[-(\frac{W}{TM})\cdot(abe^{-bT_R+\frac{bW}{TM}})\cdot(\frac{1-e^{-bW}}{e^{\frac{bW}{TM}}-1})] \quad (5)$$

3. 보수를 고려한 소프트웨어 최적 출시 시기의 결정

비용/신뢰성을 고려한 소프트웨어의 최적 출시 시기는 아래의 조건을 만족하는 T_R이 된다.

$$\min\ C(T_R) \quad (6)$$
$$s.t.\ R(W|T_R) \ge R_0$$

먼저 비용함수를 도출해 보자. 소프트웨어의 테스트 및 보수에 관련된 비용은 아래와 같이 구성된다.

비용 = ($T_R + W$까지의 내부 debugging 비용) + (patch비용) + (출시 후 생기는 고장으로 인한 손실비용) + (출시 전 testing비용) + (출시 후 testing비용)

이에 근거하여 비용함수를 도출하면 다음과 같다.

$$C(T_R) = c_1 a(1-e^{-b(T_R+W)}) + c_2 TM + c_3 \sum_{i=1}^{TM} \frac{W}{TM} abe^{-b(T_R+(i-1)\frac{W}{TM})} + c_4 T_R + c_5 W \quad (7)$$

비용함수 $C(T_R)$에 대한 convexity 특성에 대해 알아보자. 비용함수 (7)을 T_R에 관해 이차미분하면 다음과 같다.

$$\frac{d^2C}{d^2T_R} = c_1 ab^2 e^{-bT_R} \times [-e^{-bW} + k\frac{W}{TM}\{-be^{\frac{bW}{TM}}(\frac{1-e^{-bW}}{1-e^{\frac{bW}{TM}}})\}],$$
$$(k = \frac{C_3}{C_1}) \quad (8)$$

식 (8)에서 $c_1 ab^2 e^{-bT_R} > 0$ 이므로, [ ] 안의 수식에만 주목해 보자. $bW = x$, $TM = \frac{1}{t}$ (상수)라 하여 [ ]안의 수식을 x에 대한 함수 $f(x)$라고 하면 아래와 같은 정리가 성립한다.

(정리 1)

함수 $f(x) = -e^{-x} - kxte^{xt}(\frac{1-e^{-x}}{1-e^{xt}})$는 $x>0$에 대해 단조 증가 함수이다.

(증명) $\frac{df(x)}{dx} = e^{-x} + kte^{xt}(\frac{1}{e^{xt}-1}) \times [(1-e^{-x})[1+xt-\frac{xte^{xt}}{e^{xt}-1}]+x^{e-x}]$에서, $1+xt-\frac{xte^{xt}}{e^{xt}-1}$을 제외한 나머지 부분은 모두 양수이다. $1+xt-\frac{xte^{xt}}{e^{xt}-1} = \frac{e^{xt}-xt-1}{e^{xt}-1}$는 $x>0$에서 항상 양수이므로 $\frac{df(x)}{dx}>0$이 된다. 따라서 $f(x)$는 $x>0$에서 단조 증가 함수이다. ■

bW'이 $\frac{d^2C}{d^2T_R} = 0$을 만족시키는 값이라 하면, **(정리 1)**에 의해 다음과 같은 결론을 얻을 수 있다.

- $bW > bW'$인 경우, $C(T_R)$은 convex 함수이며, $\frac{dC}{dT_R} = 0$가 되는 $T_R = T_R^1$에서 최소값을 갖는다.

위의 결론에 의하여 비용함수는 시간 T_R에 대한 convex함수이고, 식 (5)의 신뢰성 함수는 시간 T_R에 대한 단조 증가 함수이다 (증명 생략). 따라서 비용함수를 최소로 만드는 출시시기를 T_R^1, 신뢰성 하한을 만족시키는 출시시기를 T_R^2라 할 때, $T_R^1 > T_R^2$라면 T_R^2시점 이후에는 신뢰성이 만족되기 때문에 비용을 최소화하는 T_R^1에서 출시를 하게 된다. 반대로 $T_R^1 < T_R^2$라면 비용이 최소인 T_R^1 시점에서 일정한 신뢰성을 만족하지 못하기

때문에 T_R^1 시점이 아닌 T_R^2시점까지 기다려야 출시가 가능해 진다. 최적 출시 시점 T_R^*은 아래와 같이 결정된다.

$$T_R^* = \max(T_R^1, T_R^2) \qquad (9)$$

식 (5)와 (7)에서 구한 T_R^1및 T_R^2는 다음과 같다.

$$T_R^1 = -\frac{1}{b}\ln\left[\frac{c_4}{c_1 abe^{-bW} + c_3 Lab^2 e^{bL}\left(\frac{1-e^{-bW}}{e^{bL}-1}\right)}\right]$$

$$T_R^2 = -\frac{1}{b}\ln\left[\frac{R_0'}{abL} \Big/ \left(\frac{1-e^{-bW}}{e^{bL}-1}\right)\right] + L$$

4. 실험예제

실험에 사용된 모수는 기존의 연구와 비교를 하기 위해 [1],[6]에서 제시된 것을 사용하였고, 새롭게 도입된 c_2, c_5의 경우 c_1, c_3, c_4와의 관계를 고려하여 결정하였다. 실험 조건은 다음과 같다.

실험 조건

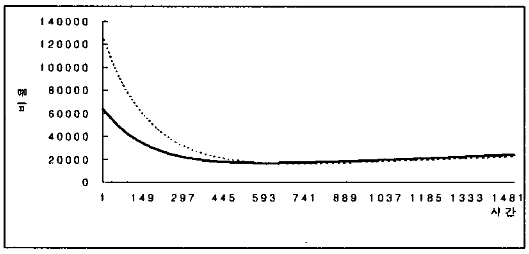
$\hat{a} = 34$, $\hat{b} = 0.00579$

$c_1 = 200$, $c_2 = 400$, $c_3 = 1500$, $c_4 = 10$,

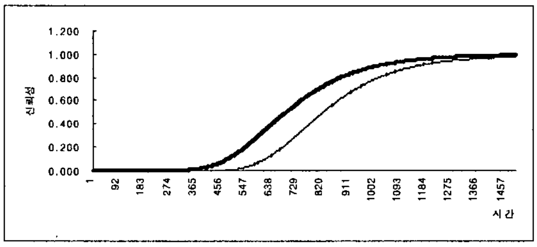
$c_5 = 1$, $W = 400$, $R_0 = 0.7$

출시 후 보수 회수가 5회 일 때를 가정하고 소프트웨어 출시 시기 결정해 보았다. 이 경우 최적 출시 시점이 803이 된다. 비용함수 및 신뢰성 함수는 [그림 2], [그림 3]의 굵은 실선과 같다.

기존의 논문에서 제시된 방법을 사용할 경우, 최적 출시 시점은 933이 된다. 따라서, 출시 이후 보수를 할 경우 출시 시점 측면에서는 180만큼의 출시시기를 앞당길 수가 있다. 비용 측면에서는 출시 후 보수를 하는 것이 약 700정도의 손해를 보게 되는 것을 알 수 있다. 하지만 본 연구에서의 비용함수는 출시 시기 단축에 따른 비용상의 이익을 고려하지 않았기 때문에, 위와 같은 비용 손실의 결과가 발생하는 것을 알 수 있다.



[그림 2] $TM=5$일 때, 비용 함수



[그림 3] $TM=5$일 때, 신뢰성 함수

5. 결론 및 추후 연구과제

본 연구에서는 기존에 소프트웨어 출시시기를 결정하는 문제에 있어서 고려되지 않았던 출시 이후의 보수라는 개념을 도입하여 출시시기를 결정하여 보았다. 제시된 모형에 대한 수식적인 증명과 더불어 적절한 예제 시스템으로 결과를 분석하였다. 따라서, 소프트웨어를 개발하는 측에서 출시 이후 보수라는 문제를 생각한다면, 기존의 연구 결과보다는 본 논문의 결과를 통해 출시 시기를 결정하는 것이 좀 더 바람직할 것이라 생각된다.

본 논문에서는 출시 이후 보수주기 동안의 고장률의 감소는 테스팅 환경내에서와 동일한 것으로 가정을 하고 분석을 하였다. 하지만, 출시가 된 이후에는 소프트웨어의 사용 환경이 달라지고, 개발할 때보다는 훨씬 많은 사용자에게 노출되기 때문에 고장의 발생 및 감소 패턴에 변화가 있게 될 것이다. 따라서, 추후 연구과제로는 출시 이후 고장 환경 변화를 고려한 모형화를 생각해 볼 수 있다.

Reference

[1] Kimura M., Toyata T. and Yamada S.,

"Economic analysis of software release problems with warranty cost and reliability requirement", Reliability Engineering and System Safety 66, 49-55, 1999

[2] Yamada S. and Osaki S., "Cost-Reliability optimal release policies for software systems", IEEE Transactions on Reliability 34, 422-424, 1985

[3] Xie M. and Hong G. Y., "Software release time determination based on unbounded NHPP model", Computer & Industrial Engineering 37, 165-168, 1999

[4] Xie M. and Yang B., "A study of operational and testing reliability in software reliability analysis", Reliability Engineering and System Safety 70, 323-329, 2000

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

[6] Musa J.D. and Iannino A., Okumoto K, "Software reliability: measurement, prediction, application", New York, McGraw-Hill, 1987

[7] Ross S. M. "Introduction to Probability Models 6th ed.", Academic Press, San Diego, 1997