

A Study on Optimal Renewal Cycle for Governmental Agency Software

Hoon-sang An[†] · Jongho Bae^{**} · Youngsung Kim^{***} · Chulhyun Park^{****}

ABSTRACT

The demands for additional functionalities and improvements in performance of software increase over time. In particular, increases in software complexity and requirements for quality control attributed to continued maintenance lead to the deterioration of software quality and raises in software life cycle costs. In order to prevent this, software operators have to conduct timely redevelopment of the software. However, the scope of previous studies on timely redevelopment of software is limited to enterprises. We, in this study, suggest a model to derive the optimal cycle for the redevelopment of governmental agency software using Renewal process and discuss its correlation with previous studies.

Keywords : Governmental Agency Software, Software Maintenance, Software Redevelopment, Renewal Process

공공기관 소프트웨어의 최적 재개발 주기 도출에 대한 연구

안 훈 상[†] · 배 종 호^{**} · 김 영 성^{***} · 박 철 현^{****}

요 약

운용 중인 소프트웨어에 대한 기능의 추가 및 성능 개선 요구는 시간이 지날수록 증가한다. 특히, 지속되는 유지보수에 따른 소프트웨어의 복잡도와 품질관리 소요의 증가는 품질저하와 수명주기 비용의 상승을 초래할 수 있으므로, 소프트웨어 운용자는 적시 재개발을 통해서 이를 예방해야 한다. 그러나 소프트웨어의 적시 재개발에 대한 기존 연구는 기업 분야에 한정되어 있다. 따라서 우리는 본 논문에서 리뉴얼 프로세스를 사용하여 공공기관에 적용 가능한 최적의 소프트웨어 재개발 주기를 산출하는 모델을 제시하고 기존 연구와 연관성을 논의한다.

키워드 : 공공기관 소프트웨어, 소프트웨어 유지보수, 소프트웨어 재개발, 리뉴얼 프로세스

1. 서 론

정부 및 공공분야 소프트웨어 사업의 규모 및 비용 산정 기준은 소프트웨어 유지보수에 있어 일정 규모이상 변경이면 재개발로 인정하고 이에 대한 대가 산정방법을 제시하고 있다[1]. 특히 소프트웨어 운용자는 Fig. 1의 소프트웨어 수명 주기에서처럼 유지보수가 지속되면 결심지점(Decision Point)에서 소프트웨어의 유지보수를 계속할 것인지, 재개발 또는 폐기할 것인지 결정하여 총수명주기 비용을 절감할 필요가 있다. 즉, 유지보수 비용 증가에 따라서 정기적인 소프트웨어 교체 또는 재개발은 유지보수 비용을 줄일 수 있는 방법

이 될 수 있다[2]. 또한 유지보수가 계속되면서 외부 환경변화 또는 사용자 요구에 따른 기능추가 등에 의해 시스템의 복잡도가 증가하면 특정 순간 재개발을 하는 것이 비용뿐만 아니라 품질관리와 업무관리 측면에서 효율적일 수 있다. 그러나 기업·기관들의 내부 생산성과 영업성, 노출에 대한 우려 때문에 정보시스템 유지보수 서비스 기업들의 데이터 공개가 이루어지지 않고 있어 국내에서 이루어지는 유지보수 수행 결과에 대한 관련자료 확보가 어려웠으며[1], 이에 따라 소프트웨어 재개발에 대한 연구 실적은 제한적이다. 단, 문홍근[2]과 박성식[3]은 금융기관 정보시스템의 최적수명주기 도출을 통한 재개발 시점 판단 모델을 제시하였으며, 이병철[1]은 기업 정보시스템 유지보수 계약에서 재개발 판정에 대한 실증적 연구를 진행한 바 있다. 그러나 문홍근[2]과 박성식[3]의 연구에서 사용된 '연차별 수명주기 비용법(Equivalent Annual Cost Method)'은 수익을 통한 자산회수에 기초한 비용 산정 방법으로 자산회수와 관련이 적은 공공기관의 비용 산정에 적용하기 제한된다. 또한, 이들은 소프트웨어를 포함하는 IT 시스템 전체에 대해 연구하여 소

[†] 비 회 원 : 육군본부 정보체계관리단 전산체계개발과장
^{**} 비 회 원 : 충남대학교 정보통계학과 교수
^{***} 비 회 원 : 육군본부 정보체계관리단 전산체계개발과
^{****} 정 회 원 : 육군 제 6보병사단
Manuscript Received : August 12, 2015
First Revision : October 12, 2015
Second Revision : October 26, 2015
Accepted : October 26, 2015
* Corresponding Author : Chulhyun Park(kmanp@hanmail.net)

트웨어에 대한 자세한 분석은 다루지 않았다. 반면, 안전행 정부(현 행정자치부)[4]는 공공기관 소프트웨어 재개발 및 폐기를 위한 비용 및 업무측면에서 평가기준을 제시한 바 있다. 그러나 이 방법은 평가기준의 모호함과 사용자 편의성 등을 측정하기 위한 객관적 근거의 부족으로 실용성에 대한 문제가 제기되었다.

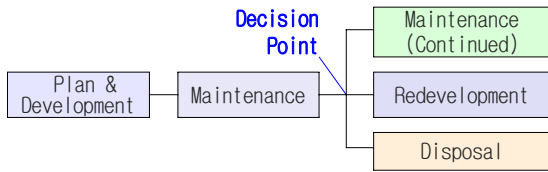


Fig. 1. Software Life Cycle

따라서, 본 논문은 현재 다수 공공기관에서 적용 중인 소프트웨어 비용 산정 방법과 한국소프트웨어산업협회[5]에서 제시하는 효율제 유지보수 방식을 사용하여 다년간 유지보수 비용의 추이를 분석한다. 그리고 그 결과를 연차별 수명주기 비용법에 대응할 수 있는 Ross[6]의 리뉴얼 프로세스(Renewal process) 모델에 적용하여 소프트웨어 재개발을 위한 최적의 주기를 구한다. 우리는 이 연구를 통하여 공공기관에서 운용 중인 소프트웨어에 대한 연간 또는 중기(5년) 이상 유지보수 예산 수립 시, 계속 유지보수·재개발·폐기 주기를 판단하기 위한 모델을 제시한다.

2. 소프트웨어 유지보수 및 재개발 비용산정 방법

2.1 소프트웨어 유지보수의 특성

Lientz & Swanson[7]과 Nosek & Palvia[8]는 유지보수 유형을 오류 수정(Fault Repair), 적응 유지보수(Software Adaption), 추가 요구사항 구현(Functionality Addition or Modification)으로 구분하고 이중 약 65%는 추가 요구사항 구현과 관련된다고 주장하였다. 일반적으로 소프트웨어는 개발 이후 체계가 안정화되기 전까지 다수의 오류 수정 수요가 발생하나 체계가 안정되면서부터는 수정 수요가 감소한다. Gartner[9]는 체계의 도입기, 적응기까지 운영비용 적용 가중치가 1 근처에 머물다가 활용기, 성숙기(안정화 단계)에서 0.9로 감소하고, 이후 대규모 개선요구 발생 시 운영비용 적용 가중치가 다시 1에 가까워지는 모델을 제시하였다. Putnam[10], Warburton[11]은 시간에 따라 시스템 안정화를 위한 유지보수 수요가 증감하는 모델을 소개하였고, Bhatt et al.[12]은 Fig. 2와 같이 추가 요구사항 구현이 소프트웨어의 유지보수 노력을 증가시키는 특성을 설명하였다. 즉, 시간이 지날수록 오류 수정 등 시스템 안정화를 위한 유지보수 수요는 감소하나 추가 요구사항 구현(Fig. 2는 External factor로 표현)은 유지보수의 노력과 복잡도를 증가시키는 요인이 된다. 그리고 소프트웨어의 크기와 복잡도가 커질수록 유지보수에 대한 노력과 비용 역시 증가하게 된다[13].

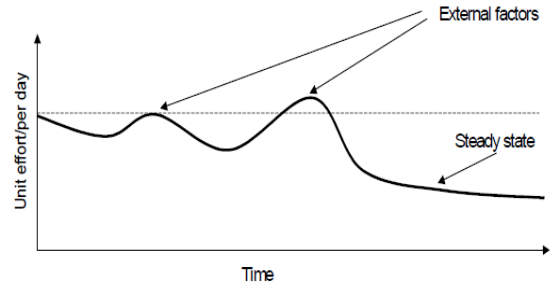


Fig. 2. Dynamics of Software Maintenance [12]

2.2 소프트웨어 유지보수 비용 산정

유지보수 비용에 관한 초기의 연구는 대개 구축개발 비용의 일정비율을 유지보수 비용으로 산정하는 방식을 적용하고 있다[3]. 유지나 등[14]은 소프트웨어 유지보수 비용 산출 방법을 개발 비용에 근거한 산출법과 유지보수 노력에 근거한 산출법의 2가지로 제시하였다. 개발비용에 근거한 산출법은 프로그램 개발규모·투입인력으로 산출하며, 유지보수 노력에 근거한 산출법은 프로그램 크기·복잡도·기능·품질·프로그래머 능력을 통하여 산출한다. 반면, 현재 다수의 공공기관은 후자의 요소들을 기능점수(FP: Function Point)[15, 16]화하거나 투입인력에 따른 노임단가를 적용하여 개발비용을 산정 후, 이를 현재가로 재산정하고 여기에 일정 효율을 곱하는 형태의 유지보수 효율제 방식을 적용하고 있다. 효율제 방식에 대해서는 다음에서 알아본다.

1) 소프트웨어 사업대가 산정 가이드 [5]

현재 다수 공공부처 및 기관은 소프트웨어 비용 산정을 위하여 소프트웨어산업협회의 소프트웨어 사업대가 산정 가이드를 활용 중에 있다. 이 가이드는 소프트웨어사업 대가 산정 프로세스를 기획단계, 구현단계, 운영단계로 구분하고, 운영단계의 대가 산정 유형을 소프트웨어 유지보수비와 운영비로 구분하고 있다. 이중 유지보수 대가 산정 방식은 기본적으로 효율제 방식, 고정비/변동비 방식, SLA(Service Level Agreement) 기반 정산 방식의 3가지 방식으로 구성된다[17]. 일반적으로 다수의 공공기관에서 사용하는 효율제 방식의 대가 산정식은 다음과 같다.

유지보수 비용
 = 대상 소프트웨어의 계약시점에 재산정된
 개발비 × 유지보수 효율(5~15%) + 직접경비

가이드에 따르면 ‘계약시점에 재산정된 개발비’는 대상 소프트웨어가 유지보수 되면서 현재 시점까지 축적된 추가·변경·삭제 등의 노력이 최초 개발비에 반영된 값이다. 본 논문은 유지보수 기간 동안 투입된 노력 중 오류 수정·보완 등 시스템 안정화를 위한 노력은 제외하고 추가 요구사항 구현을 위한 노력들을 활용하여 재산정된 개발비를 계산

한다.¹⁾ 재산정시에는 최초 개발 시점 이후 물가상승율과 변경된 기능점수(또는 노임공수) 당 단가가 적용된다. 즉 개발 이후 추가 요구사항 구현 노력들이 비용으로 반영되어 대부분 최초 개발비 보다 재산정된 개발비가 더 크게 산정된다. 그리고 유지보수 효율은 유지보수 횟수, 자료처리 건수, 타 시스템 연계, 실무지식 필요, 분산처리 여부의 요소들로 난이도를 환산한 후 계산식에 의해 백분율화 된 값으로 나타난다. 이는 국산과 외산의 기준이 차이가 있으며 국산 소프트웨어는 과거 지식경제부 고시[18]를 유지하여 5~15%를 적용하고 있다.

2.3 재개발 비용 산정

개발된 소프트웨어의 일부를 다시 개발하거나 발주자가 보유한 소프트웨어 자산을 재사용하여 개발하는 것을 재개발이라고 정의한다[1]. 안전행정부[4]는 “장비적인 운영유지비의 절감이나 활용도 개선을 위하여 신기술 및 신규소프트웨어의 도입을 통한 비용측면의 개선이나 노력이 필요한 경우” 재개발이 필요하다고 설명하였다. 소프트웨어산업협회[5]는 재개발을 “개발된 소프트웨어의 일부를 다시 개발하거나, 발주자가 보유한 소프트웨어 자산을 재사용한 개발과 관련된 제반활동”이라고 정의하였다. 또한 재개발을 기존 시스템을 폐기하고 새롭게 정보시스템을 구축하는 순수 재개발과 기존시스템에 새로운 기능을 추가 개발하는 것이 주목적이지만 일부 기존 시스템의 기능이 수반되는 확대개발, 기존 시스템의 기능 변경이 주목적이거나 일부 새로운 기능이 기존 시스템에 추가되기도 하는 보완개발로 분류하였다. Sakhthivel[19]과 문홍근[2], 박성식[3]이 제시한 최적수명주기의 경제수명은 노후 시스템의 교체 또는 순수 재개발을 위한 적기를 의미하므로 기능이 추가되거나 프로그램 일부가 수정되는 경우는 고려하지 않는다. 이들은 ‘연차별 수명주기 비용법’으로 경제성이 최적이 되는, 즉, 총수명주기 비용을 가장 크게 절약할 수 있는 지점을 산출하여 시스템을 폐기하고 교체 또는 신규개발할 것을 제안하였다. 그러나 본 논문은 소프트웨어를 폐기하고 새롭게 구축하는 순수 재개발은 배제하고 안전행정부[4]와 소프트웨어산업협회[5]의 정의와 부합되는 확대개발 및 보완개발로 재개발의 범위를 한정한다. 즉, 공공기관에서 개발한 소프트웨어의 재사용성에 중점을 두고, 특정 수준 이상의 요구 사항 구현을 처리하기 위하여 경제적으로 최적인 주기를 재개발 주기로 판단한다.

- ① 연차별 개발비용 = 초기 개발금액 × 자본회수계수
- ② 연차별 유지보수비용 = 누적 유지보수비용 × 자본회수계수
- ③ 연차별 수명비용 = ① + ②

1) 'n년도 재산정된 개발비 = 최초 개발비 + $\sum_{i=1}^n$ i년도 추가요구사항 구현 비용'을 전제로 하며, 시스템 안정화는 무상 하자보수 기간 동안 집중적으로 진행되므로 총수명주기 비용에 큰 영향을 미치지 않음을 가정한다.

먼저 Sakhthivel[19]과 문홍근[2], 박성식[3]이 제시한 연차별 수명주기 비용법은 다음과 같이 연차별 수명비용, 즉, 연차별 개발비용과 연차별 유지보수비용의 합이 최소가 되는 연차를 최적 경제수명으로 판단한다.

여기서 자본회수계수는 매년 일정한 동일 금액으로 회수하여 종년에는 투자된 모든 비용이 회수되는 비율이다. 따라서 자본회수가 비용 산정에 적용되지 않는 다수의 공공기관의 경우는 위의 방식과 다른 비용 산정 방식이 필요하다.

자본회수와 관련성이 적은 공공기관에서 주로 사용하는 소프트웨어산업협회[5]의 소프트웨어 재개발비 산정기준은 다음과 같다.

$$\text{소프트웨어 재개발 비용} = \text{총 재개발 원가} + \text{직접경비} + \text{이윤}$$

여기서 총 재개발 원가는 재사용 대상 소프트웨어 중 수정 대상 소프트웨어의 규모와 추가개발 소프트웨어 규모의 합으로부터 계산한다.

또한, 소프트웨어산업협회[5]는 재개발과 유지보수의 구분을 “소프트웨어의 기능변경 수준이 일정 수준 이상이면 재개발, 그 이하이면 유지관리(유지보수)로 구분하는 것이 바람직함”이라고 하였다. 예를 들면, 개발된 소프트웨어의 기능 변경소요가 최초 개발소요의 50% 이상이거나, 기능 변경량이 75FP 이상이면 재개발 기준을 적용한다고 설명했다. 기능변경 소요가 최초 개발소요를 초과한다면 재개발이 아닌 신규개발(또는 순수 재개발)이 되므로, 우리는 재개발 비용은 (현재시점에서 재산정한) 개발비용을 초과하지 않는다는 것을 전제로 한다.

안전행정부[4]는 정보시스템을 비용측면과 업무측면에서 성과 측정하여 각각 5점 만점 점수로 환산하고 이중 업무측면이 3점 이상이나 비용측면이 3점 이하, 즉, 업무측면의 불편성 보다는 비용측면에서 비효율적일 경우 재개발을 하도록 기준을 제시하였다. 그리고 비용측면이 3점 이상이나 업무측면이 3점 이하인 경우 기능고도화를 하도록 명시하였으나, 다수의 경우 재개발과 기능고도화에 필요한 노력이 동일하기 때문에 그 차이가 다소 모호하다. 또한, 업무측면에서 “사용상의 편의성” 항목의 기준이 명확하지 않는 등 현 공공기관 소프트웨어나 정보시스템에 적용하기에 제한되는 사항들이 식별되었다.

3. 리뉴얼 프로세스를 사용한 재개발 적기 산출

3.1 리뉴얼 프로세스

리뉴얼 프로세스(Renewal process)는 시간에 따라 랜덤하게 발생하는 사건(또는 Renewal)을 다루는 포아송 프로세스(Poisson process)의 일반화된 형태이다. Ross[6]는 시간이 지날수록 비용이 증가하는 시스템에서 장기적인 비용을 최소화하기 위한 적정 리뉴얼 시점을 추정하는 방법을 소개하

였다. 자가 차량 교체를 예로 들면, 기존 보유 차량이 조금이라도 고장이 덜 난 상태에서 판매해야 새 차 구매를 위한 비용을 더 마련할 수 있다. 여기서 리뉴얼은 기존 차량을 팔고 새 차량을 구매하는 것이며, 기존 보유 차량이 완전 고장시점에 근접할수록 판매 수익이 줄어들어 총 소요비용은 늘어가는 특성을 갖는다.

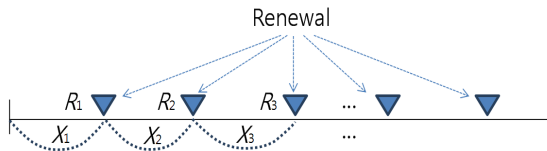


Fig. 3. Renewal, Cost(R_i) and Renewal Interval(X_i)s

Fig. 3에서 $X_1, X_2, \dots, X_n, \dots$ 은 리뉴얼과 리뉴얼 사이 간격(Inter-Renewal Time, Renewal Intervals) 또는 ‘사이클’이라 하며, 차량 교체 시 이번 차량 교체에서 다음번 차량 교체까지 사이의 시간이라 할 수 있다. 그리고 R_n 을 n 번째 사이클에서의 소요비용이라고 하자. 여기서 $R_1, R_2, \dots, R_n, \dots$ 은 i.i.d.(independent and identically distributed)한 확률변수로 가정한다. 또한 $R(t)$ 를 시각 t 까지 총 소요비용이라 할 때 $R(t) = \sum_{n=1}^{M(t)} R_n$ 으로 정의한다. 여기서 $M(t)$ 는 시각 t 까지 총 리뉴얼 횟수이다. 이 리뉴얼 프로세스는 다음과 같은 성질을 갖는다[6, p.433].

$$\text{만일 } E[R_n] = E[R_1] < \infty \text{ 이고 } E[X_n] = E[X_1] < \infty$$

$$\text{이면, } \lim_{t \rightarrow \infty} \frac{R(t)}{t} = \frac{E[R_1]}{E[X_1]} \quad (1)$$

여기서 $\lim_{t \rightarrow \infty} (R(t)/t)$ 는 오랜 시간 동안의 평균 소요 비용을 의미한다. 본 논문에서는 소프트웨어 재개발을 리뉴얼로 정의하고, 최초 개발 후 1회 재개발 시점까지를 한 사이클로 두어, $\lim_{t \rightarrow \infty} (R(t)/t)$ 를 최소로 만드는 사이클의 길이, 즉, 최적의 재개발 주기를 결정한다.

3.2 최적 재개발 주기 산출

Sakthivel[19]과 문홍근[2], 박성식[3]은 최적 경제수명을 도출하기 위하여 정보시스템 유형별 투자비용 산정 \Rightarrow 연차별 수명주기 비용 산출 \Rightarrow 유형별 경제적 수명주기 분석의 절차를 진행하였다. Sakthivel[19]은 연차별 수명주기 비용 곡선을 이용하여 최소 안장점(Fig. 4. 참조)의 비용과 기존 소프트웨어의 최근 유지보수 비용과의 비교를 통해 곡선의 안장점인 7년을 교체 또는 순수 재개발 주기로 제시하였다. 그러나 그의 모델은 가설 및 추론에 근거하고 있으며, 현장에서의 실증 데이터에 근거하지 못하고 있다[3]. 이에 따라 문홍근[2]과 박성식[3]은 금융 분야의 실증 데이터를 분석하여 최적수명주기, 즉, 경제수명(Economic Life)을 산출하였다.

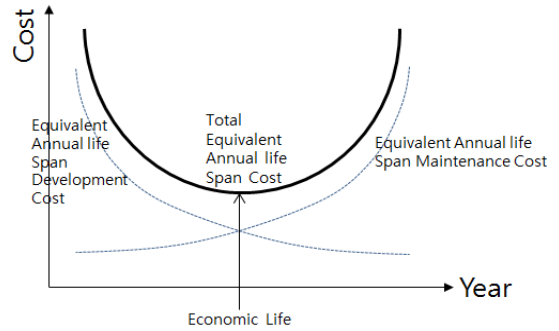


Fig. 4. Annual Life Span Curve [1]

이제 자본회수와 관련이 적은 공공기관의 경우를 살펴보면, Fig. 5의 예시와 같이 매년 재산정 되는 개발비 \times 요율로 이루어지는 연간 유지보수 비용(Annual Cost for Software Maintenance)은 증가 곡선을 그린다. 개발 이후 시스템이 안정화되면 오류 수정·보완 소요는 다소 감소할 수 있지만, 추가 요구사항 구현 노력이 매년 축적되기 때문에 유지보수 계약시점의 재산정 개발비는 지속적으로 증가한다.²⁾ 따라서 연간 유지보수 비용도 상승곡선을 그리며, 대다수의 경우 재개발(Redevelopment) 이후 유지보수 비용이 감소한 후 다시 증가하게 된다. 즉, 추가 요구사항이 계속 증가하면 특정 시점에서 재개발하는 것이 총수명주기 비용 절약에 도움이 됨을 유추할 수 있다. 또한 소프트웨어는 유지보수가 지속될수록 복잡도가 증가하기 때문에[20, 21] 동일한 추가 요구사항이라도 시간이 지날수록 구현을 위한 소요 노력과 비용이 증가하는 특성을 갖는다. 즉, 가능하면 시스템이 더 복잡해지기 전에 재개발 등을 통해서 요구사항 구현을 해결하는 것이 바람직하다.

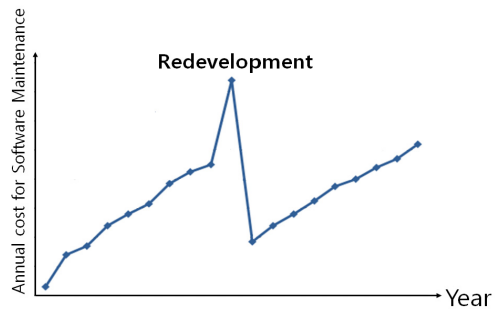


Fig. 5. Annual Cost for Software Maintenance

이제 n 번째 연도의 추가 요구사항 구현에 투입되는 노력의 환산 비용을 F_n , 오류 수정·보완 등 시스템 안정화를 위한 노력의 환산비용을 K_n , 유지보수 요율을 r_n , 최초 개발비를 S , 재산정 개발비를 S_n' , 유지보수 비용을 C_n , 그리고 n 번째 연도의 재개발 비용을 D_n 이라 할 때 F_n 과 K_n 은 등분산을 가정하고 다음과 같이 나타낼 수 있다.

²⁾ 기능점수/노임공수로 비용 산정 시 시세에 따라 증가하는 기능점수 당 단가/노임단가도 유지보수 비용 상승의 원인이 된다.

$$C_n = \sum_{i=1}^n F_i r_i + K_n r_n + S \cdot r_n, \quad S'_n = S + \sum_{i=1}^n F_i > D_n, \\ F_n = f(n) + \epsilon_{f_n}, \quad K_n = k(n) + \epsilon_{k_n}.$$

여기서 $f(n)$, $k(n)$ 은 각각 n 에 대한 함수이며 ϵ_{f_n} 과 ϵ_{k_n} 은 오차를 의미하는 확률변수로서 각각 분산이 일정하고 기대값이 0이다. $|\epsilon_{f_n}|$ 과 $|\epsilon_{k_n}|$ 은 각각 $f(n)$ 과 $k(n)$ 에 비하여 매우 작은 값이므로 F_n 과 K_n 은 양수로 가정한다. 그리고 재개발 비용 D_n 은 재산정 개발비(S'_n)를 초과하지 않는 값으로 결정된다. 이때 D_n 은 재개발 조건에 따라 S'_n 의 30%, 40%, 50%, ... 등 다양하게 상정할 수 있는데, D_n 이 S'_n 의 100% 이상인 경우 재산정 개발비를 초과하므로 소프트웨어 폐기 후 신규개발(순수 재개발) 하는 경우로 판단한다. 전술한 바와 같이 측정되는 추가 요구사항 구현과 관련된 $\sum_{i=1}^n F_i r_i$ 은 n 에 따라 증가하고, 시스템 안정화와 관련된 $K_n r_n$ 은 감소하는 형태를 갖는다. 또한, Equation (1)에서 첫 번째 사이클의 길이를 N (년)이라 하면($1 \leq n \leq N-1$) E[한 사이클의 길이] = $E[X_1] = N$ 년이 된다. 그리고 1회 사이클에서 소요비용 R_1 은 $N-1$ 번째 연도까지 유지보수 비용 합계 $\sum_{n=1}^{N-1} C_n$ 과 N 번째 연도 재개발 비용 D_N 의 합이므로,

$$R_1 = \sum_{n=1}^{N-1} C_n + D_N \\ = \sum_{n=1}^{N-1} \sum_{i=1}^n F_i r_i + \sum_{n=1}^{N-1} K_n r_n + S \sum_{n=1}^{N-1} r_n + D_N.$$

따라서,

$$\lim_{t \rightarrow \infty} \frac{R(t)}{t} = \frac{E[R_1]}{E[X_1]} \\ = E \left[\sum_{n=1}^{N-1} \sum_{i=1}^n F_i r_i + \sum_{n=1}^{N-1} K_n r_n + S \sum_{n=1}^{N-1} r_n \right] / N \quad (2)$$

이고, $\lim_{t \rightarrow \infty} (R(t)/t)$ 를 최소가 되게 하는 N 값이 첫 번째 재개발을 위한 최적의 유지보수 연차가 된다. 참고로 시스템이 계속 사용되는 경우, 우리는 동일한 방법을 사용하여 이후 적정 재개발 주기 또한 파악할 수 있다.

4. 실증 데이터 분석

이제 앞에서 구한 결과를 실제 공공기관에서 운용중인 모델에 적용하여 최적의 재개발 주기를 산출한다. 모델은 현재 OO부처 산하에서 운용 중인 교육정보관리시스템으로서, 2002년도 284백만원(=S)의 비용으로 개발되었으며, 총 라인수는 약 20만 라인, 개발언어는 ABAP4, Delphi 등이다. 그

리고 유지보수 12년차인 2014년 약 8억원(S'_{12} 의 52%)의 예산을 투입하여 재개발하였으며, 과거 10년간 실제 투입되었던 유지보수 노력을 기능점수를 활용한 비용으로 환산하여 Table 1과 같이 산출하였다.³⁾

Table 1. Cost Factors from Maintenance Efforts

Cost Unit : Million Won

n (years)	$F_n r_n$	$K_n r_n$	r_n	C_n	S'_n
1	1.19	1.98	0.07	23.05	300.99
2	32.06	1.36	0.08	57.49	701.68
3	6.72	0.68	0.048	41.08	841.61
4	0.34	0.41	0.048	41.15	848.69
5	3.97	0.57	0.05	46.97	928.00
6	1.77	0.35	0.05	48.52	963.41
7	3.06	0.08	0.05	51.30	1024.52
8	9.46	0.12	0.07	81.29	1159.70
9	17.24	0.21	0.07	98.64	1406.03
10	6.20	0.21	0.07	104.83	1494.64

Table 1의 값들을 활용하여 회귀분석하면 $\sum_{i=1}^n \widehat{F_i r_i} = 7.6n$, $\widehat{K_n r_n} = 1.84 \times e^{(-0.2893n)}$, $\widehat{r_n} = 0.06$ 을 얻고, 이 추정량들을 $E \left[\sum_{i=1}^n F_i r_i \right]$, $E [K_n r_n]$, r_n 으로 사용한다. 그리고 D_N 은 재개발시 조건에 따라 S'_N 의 0.5배, 0.6배, 0.7배...(이 값들을 P 라 하자.) 등으로 다양하게 상정할 수 있는데, $\widehat{S'_n} = S + 112.07n$ 이므로 $E [D_N] = (284 + 112.07N) \times P$ 로 둔다.

이제 $P = 0.5, 0.6, 0.7$ 으로 하고 각 비용 요소들의 추정값을 이용하여 Equation (2)를 계산하면,

$$\lim_{t \rightarrow \infty} \frac{R(t)}{t} = \left\{ \sum_{n=1}^{N-1} 7.60n + \sum_{n=1}^{N-1} 1.84e^{-0.2893n} \right. \\ \left. + (284 \times 0.06)(N-1) \right. \\ \left. + (284 + 112.07N) \times P \right\} / N.$$

위 식에서 $\lim_{t \rightarrow \infty} (R(t)/t)$ 를 최소로 만드는 N 은 D_N 이 S'_N 의 50% 또는 60%일 때 6, 70%일 때 7로 나타났다. 여기서 우리는 개발 후 1년간의 무상 유지보수(하자보수)기간을 고려하여, 재산정 개발비의 50% 또는 60%로 재개발할 경우 유지보수 7년차에, 70%로 재개발 할 경우 8년차에 재개발 하는 것이 총수명주기 비용을 최소화할 수 있는 최적의 재개발 주기임을 확인하였다.

3) 소프트웨어산업협회의 기능점수 방식에 의한 비용산정 모델 중 간이법을 사용하여 계산하였으며 편의 상 직접경비는 미반영하였다. 또한 기능점수 당 단가는 1~5년차까지 461,081원, 6~10년차까지 497,427원으로 실제 값을 적용하였다.

참고로 Table 2는 문홍근[2]과 박성식[3]이 제시한 금융 분야의 실증 데이터 분석을 통한 연차별 수명주기 비용 산출결과의 예시이다. 이는 10년 간 비용 데이터를 분석한 결과 연차별 수명주기 비용값(Total Equivalent Annual life Span Cost)이 가장 작은 7년차가 시스템 교체 및 신규개발을 위한 최적의 경제수명임을 보여준다. 이 시점은 Fig.4에서처럼 감소하는 연차별 수명주기 개발비용과 증가하는 연차별 수명주기 유지보수비용의 합이 최소가 되는 안장점이다. 여기서 자본회수계수를 배제하고 누적 유지보수비용과 초기 개발비용을 활용하여 앞의 리뉴얼 프로세스 모델을 적용하면 경제수명은 5년으로 나타난다. 따라서 자본회수 부분을 보정하여 기업에도 적용할 수 있는 리뉴얼 프로세스 모델의 추가적인 연구가 요구된다.

Table 2. An Example of Calculation of Equivalent Annual life Span Costs [3]

Cost Unit : Million Won

<i>n</i> (years)	Equivalent Annual life Span Development Costs	Equivalent Annual life Span Maintenance Costs	Total Equivalent Annual life Span Costs
1	4,075	-	4,075
2	2,076	-	2,076
3	1,409	25	1,434
4	1,077	88	1,165
5	877	139	1,016
6	744	162	906
7	649	180	829
8	578	482	1,061
9	523	768	1,292
10	479	1,049	1,528

Table 3. Comparison between Two Methods

	Total Equivalent Annual Life span Cost Method	Renewal Process Model
Decisive Factors	Capital Recovery with Return + Equivalent Cost	Total Maintenance Cost + N th Year Redevelopment Cost
Application	Enterprise	Governmental Agency, etc.
Curve Fitting	Polynomial Equation including All Points	Regression
Objective	Disposal and Replacement	Redevelopment and Reuse

5. 기존 연구결과와 비교 분석

전술한 바와 같이 Sakthivel[19]과 문홍근[2], 박성식[3]이 제시한 연차별 수명주기 비용법은 수익을 통한 자본회수에

기초하여 (소프트웨어를 포함하는) 정보시스템에 대한 연차별 비용을 산출하였으며, 본 논문에서는 자본회수가 적용되지 않는 유지보수 노력과 재개발 노력을 비용으로 산출하였다. 연차별 수명주기 비용법은 기업에 적용하기에 적절한 산출 방법이며, 리뉴얼 프로세스는 공공기관 외에도 매년 총 유지비용이 증가하는 경우에 적용 가능한 모델이다. 또한 Table 3에서처럼 데이터의 최적 Curve Fitting을 위하여 문홍근[2]과 박성식[3]은 산출된 연차별 수명주기 비용점이 모두 포함되는 다항식 함수를 사용하였으며, 본 논문은 비용 요소들을 확률변수로 가정한 회귀분석을 사용했다. 즉, 기존 연구는 확정적인(Deterministic) 다항식을 적용하여 보다 정확한 Curve Fitting을 구현하였으나 이상치(Outlier)가 포함되면 함수 도출이 제한될 수 있다. 또한 여기서 수명은 시스템의 교체 또는 폐기 후 신규개발을 위한 시기를 의미하기 때문에 소프트웨어를 2회 이상 재개발 및 재사용하는 경우에 적용하기 제한된다. 반면, 본 논문은 비용 요소들을 확률변수로 가정한 회귀분석, 즉, 확률적(Stochastic)인 방법을 사용하였기 때문에 이상치에 대한 유의성 판단이 더 용이하다. 그리고 소프트웨어 폐기 후 신규개발($D_N = S'_N$ 일 때)을 위한 시기 뿐만아니라 2회 이상의 재개발 및 재사용 주기를 예측하는데 유용하게 활용될 수 있다.

6. 결 론

현재 소프트웨어는 개발보다 이후 유지보수 및 재개발을 위한 비용의 비중이 더 크다. 소프트웨어는 유지보수를 하면서 복잡도가 지속적으로 증가하기 때문에 적시 재개발이 이루어지지 않으면, 소프트웨어의 품질 저하 뿐만아니라 총 수명주기 비용이 오히려 증가할 수 있다. 그러나 데이터 획득의 어려움 등의 제한사항으로 인하여 적시 재개발 주기 판단을 위한 연구는 금융 기업 분야에 한정하여 진행되었다. 따라서 본 논문은 공공기관의 소프트웨어 유지보수 기간 중 재개발 및 폐기 등을 위한 최적 주기 산출 모델을 다음과 같은 방법으로 도출하였다.

첫째, 소프트웨어 개발 이후 매년 진행되는 유지보수를 시스템 안정을 위한 오류 수정·보완과 추가 요구사항 구현으로 구분하였으며, 현재 다수 공공기관에서 사용 중인 소프트웨어산업협회의 비용 산정 방법을 적용하였다.

둘째, 한 번의 재개발을 한 사이클로 하고 이에 대한 평균 비용을 구한 후 리뉴얼 프로세스에 적용하여 최적의 재개발 주기를 산출했다. 이때 재개발 비용은 계약시점의 재산정 개발비를 초과하지 않음을 전제로 하였다. 이에 대해 현재 운용 중인 모델을 적용한 결과 실제 재개발이 이루어지고 있는 시점보다 4~5년 앞서 재개발을 했을 경우 총 수명주기 비용을 절감할 수 있음을 확인했다.

우리는 향후 로지스틱·Pearl·Gompertz 모델[22] 등 기존에 제시된 다양한 비용 추정 방법을 사용하여 소프트웨어의

최적 수명주기를 연구할 예정이다. 또한, 리뉴얼 프로세스를 기업의 소프트웨어 경제수명 산출에 적용하여 표준화된 수명주기 도출 방법을 연구하고자 한다.

References

[1] B. C. LEE, "An Empirical Study on Decision Criteria of Additional Software Development in Information System Maintenance Contract," Department of Computing, Ph.D. Dissertation, Soongsil Univ., 2013.

[2] Heoung Keun Moon, "Study on Development of Cost Data Based Optimal Life Cycle Model for Financial Information Systems," Department of Information Technology & Management, Ph.D. Dissertation, Konkuk Univ., 2012.

[3] Sung sik Park, "An Empirical Study on the Establishment of the Optimal Economic Life Span Model for the Information Systems," Department of Information Technology & Management, Ph.D. Dissertation, Konkuk Univ., 2011.

[4] S. Korea's Ministry of Security and Public Administration, "A Manual for Operational Performance Measurement of Information Systems," pp.27-31, 2014.

[5] Korea Software Industry Association, "A Guide for Payment of Software Business," 2014.

[6] S. M. Ross, "Stochastic Process," 9th ed. New York: Wiley, 1983.

[7] B. P. Lientz and E. B. Swanson, "Software Maintenance Management Reading," MA: Addison-Wesley, Ch.21, 1980.

[8] J. T. Nosek and P. Palvia, "Software maintenance management: Changes in the last decade," *Software Maintenance: Research and Practice*, Vol.2, No.3, pp.157-174, 1990.

[9] Gartner, "IT service maturity model, Strategic Analysis Report," 2002.

[10] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," *IEEE Trans. on Software Eng.*, Vol.SE-4, No.4, pp.345-361, 1978.

[11] R. D. H. Warburton, "Managing and Predicting the Costs of Real-Time Software," *IEEE Trnas. on Software Eng.*, Vol.SE-9, No.5, pp.562-569, 1983.

[12] P. Bhatt, G. Shroff, and A. K. Misra, "Dynamics of Software Maintenance," *ACM SIGSOFT Software Engineering Notes*, Vol.29, No.5, 2004.

[13] E. B. Swanson and E. Dans., "System Life Expectancy and the Maintenance Effort: Exploring Their Equilibration," *MIS Quarterly*, Vol.24, No.135, pp.277-297, 2000.

[14] J. Yoo, B. S. Kim, M. S. Choi, and J. Oh, "Estimation of Vendor Cost for Software Maintenance and Repair Outsourcing," *Asia Pacific Journal of Information Systems*, Vol.16, No.2, pp.143-158, 2006.

[15] IFPUG, "Function Point Counting Practices Manual 4.2," 2004.

[16] NESMA, "Function Point Analysis for Software Enhancement Guideline," Ver.2.2.1, 2009.

[17] C. S. Yoo and E. J. Chung, "A Study on the Improvement for Management Efficiency about the Approach of Applying IT to our Defense Area," *The Korean Journal of Defense Analysis*, Vol.27, No.2, pp.67-99, 2011.

[18] Notification No.2010-52 of the Ministry of Knowledge and Economy, 2010.

[19] S. Sakthivel, "A Decision Model to Choose between Software Maintenance and Software Redevelopment," *Journal of Software Maintenance*, Vol.6, pp.121-143, 1994.

[20] H. Leung and Z. Fan, "Software Cost Estimation," Department of Computing, HongKong Polytechnic University, 2002.

[21] Young-Joon Moon and Sung-Yul Rhew, "Software Replacement Time Prediction Technique Using the Service Level Measurement and Replacement Point Assessment," *Journal of KIPS*, Vol.2, No.8, pp.527-534, 2013.

[22] S. K. Park, S. W. Lee, and J. H. Park, "A Software Cost Estimation Using Growth Curve Model," *Journal of KIPS*, Vol.11-D, No.3, pp.597-604, 2004.



안 훈 상

e-mail : husker@naver.com

1987년 육군사관학교 전자공학과(학사)

1998년 美 네브라스카 주립대학교 전산학과(석사)

2012년~현 재 아주대학교 NCW학과 박사과정

2015년~현 재 육군본부 정보체계관리단 전산체계개발과장
 관심분야: 사이버전 및 대응전략, 빅데이터, 암호장비체계,
 육군 소프트웨어 개발



배 종 호

e-mail : bae-jongho@cnu.ac.kr

1995년 포항공과대학교 수학과(학사)

1997년 포항공과대학교 수학과(석사)

2001년 포항공과대학교 수학과(박사)

2014년~현 재 충남대학교 정보통계학과 교수

관심분야: 확률과정론, 대기행렬이론



김 영 성

e-mail : topys2000@naver.com
1999년 호남대학교 컴퓨터공학과(학사)
2010년 국방대학교 전산정보학(석사)
2015년~현 재 육군본부 정보체계관리단
전산체계개발과
관심분야: 서버·네트워크 침해대응 시스템,
사이버방호체계, 육군 소프트웨어
개발



박 철 현

e-mail : kmanp@hanmail.net
1999년 육군사관학교 물리학과(학사)
2003년 국방대학교 무기체계학(석사)
2016년 충남대학교 통계학과(박사)
2015년~현 재 육군 제 6보병사단
관심분야: SW 예산·기술·지식재산권
정책, 데이터 분석