

# 유지보수 관리 체계의 정형화 및 비용 예측 모델에 관한 연구

류 성 열<sup>†</sup> · 백 인 섭<sup>††</sup> · 김 하 진<sup>†††</sup>

## 요 약

본 연구에서는 소프트웨어 위기의 주요 원인인 유지보수 문제를 해결하기 위한 방안을 제시하기 위한 것으로, 국내의 유지보수 문제점과 현황을 설문으로 기초 조사·분석하고 면담과 문헌을 참고하여 문제점을 해결하고자 하였다. 연구의 결과로 유지보수 생명주기를 정의하고 유지 보수를 효율적으로 수행하기 위한 기본 전략을 수립하고, 수립된 기본 전략을 구현하기 위한 유지보수 지원 시스템을 개략 설계하여 유지보수 시스템의 자동화 설계를 위한 기반을 구축하였다.

## A Study on the Formalization of Maintenance Management Systems and the Cost Predictive Model

Sung Yul Rhew<sup>†</sup> · InSub Baek<sup>††</sup> · HaJin Kim<sup>†††</sup>

## ABSTRACT

In this paper, we propose a solution to the software maintenance problem that is a primary factor of software crisis. We surveyed and analyzed the current software maintenance problems through questionnaires and interviews. As a result, we defined the software maintenance management life cycle and established a fundamental strategies to solve the software maintenance problems efficiently. We also designed a software maintenance management support system to construct an automated software maintenance management tool.

Furthermore, to improve the formalization and reliability of the software maintenance management procedure, we defined a cost predictive model using a fixed-single parameter based on comprehensive program size for the source code and delivered effort(person/month). We elaborated the model by considering an experience level of maintainer, a skill-level defined by the manager, and a reliability level required by the model of maintenance management.

### 1. 서 론

정보화 사회에서 컴퓨터 관련 업무의 생산성은 매우 중요하다. 그 중 컴퓨터 관련 비용의 80% 이상을

차지하는 소프트웨어의 생산성은 더욱 강조되고 있다. 소프트웨어 비용 중 70% 이상이 유지보수 관련 비용으로 소프트웨어 위기의 주 원인이 되고 있다. 소프트웨어 시스템도 환경의 변화에 따라 대규모화 되고 업무처리 방식이나 응용 방법이 복잡하고 다양 화됨에 따라 개발 방법이나 기법도 종래의 소프트웨어와는 상당한 차이점을 나타내고 있어 유지 보수를 더욱 어렵게 만들고 있다<sup>[4, 6, 8, 17, 18]</sup>.

† 정 회 원: 숭실대학교 정보과학대학 컴퓨터학부 교수  
†† 정 회 원: 아주대학교 정보 및 컴퓨터공학부 교수  
††† 비 회 원: 아주대학교 정보 및 컴퓨터공학부 교수  
논문접수: 1995년 11월 24일, 심사완료: 1996년 5월 15일

효율적인 유지보수 방안이 소프트웨어 생산성 향상을 위한 가장 중요한 문제임을 인식하고 이를 위한 대책이나 방법에 관한 연구도 많이 이루어지고는 있으나 실질적인 생산성 향상은 크게 이루어지고 있지 않은 현실이다. 특히 국내에서 개발 운용중인 소프트웨어 유지 보수의 생산성은 개발비를 상회하고 있을 뿐만 아니라 개발 자체를 무력화 시키고 있는 실정이다<sup>2,6</sup>.

본 연구는 우리의 환경에 적합한 유지보수 관리 체계와 방법론을 정립하여 정형화 함으로서 유지보수 시스템의 자동화를 위한 기반 작업을 수행하고, 유지보수에 소요되는 비용을 예측하는 방법을 연구하여 효율적인 관리 통제를 위한 방안을 연구하여, 소프트웨어의 생산성 향상을 도모하고자 한다.

연구 목적 달성을 위하여 국내 중 기업 이상의 업체를 대상으로 설문 조사하여 현상을 파악하고 근원적인 문제점을 찾는다. 유지보수의 조직과 역할 및 처리 방법을 파악하기 위하여 여러 유형의 유지보수 생명주기를 정의하고, 각 생명주기 단계별로 수행하여야 할 기능을 정의하며 유지 보수의 절차와 방법을 표준화하고 자동화 하고자 한다. 또 파악된 문제점을 정의된 생명주기 관점으로 해결하기 위한 전략을 수립하고, 수립된 전략을 바탕으로 유지보수 지원 시스템을 개략 설계한다.

유지보수 지원 시스템의 효율적인 운영 관리를 위해서는 시스템 운영을 위한 예측 방법이 있어야 한다. 유지보수의 예측은 업무의 수행에 필요한 비용, 시간 및 자원의 예측으로 관리 통제의 수단으로 활용된다.

## 2. 유지 보수의 방법과 비용 예측 모델

### 2.1 유지 보수의 방법

소프트웨어의 유지 보수는 개발단계에서 만들어진 소프트웨어를 사용자에게 인도한 다음부터 폐기될 때 까지 이루어지는 활동으로서, 개발 완료된 소프트웨어의 결점을 정정하거나, 운용 환경의 변화에 따른 수정 작업, 미래의 발생할 변화나 안전을 기하기 위한 보완 작업, 기존 소프트웨어의 성능을 향상시키거나 확대하기 위한 작업 및 기존 소프트웨어들의 통합 작업 등이 있다<sup>5,18,19,27</sup>.

유지보수 방법에 관한 연구로는 첫째, 원시 코드나

개발 시 작성된 문서를 효율적으로 이해하고 활용하기 위한 연구와<sup>13,16</sup> 둘째, 원시 코드를 재구성(restructuring)하거나, 원시 코드를 이용하여 설계 문서와 분석 관련 문서를 회복(Design Recovery)하고 이를 수정 보완하여 새로운 원시 코드를 재구성하는 재공학(ReEngineering) 방법에 관한 연구와<sup>2,6,25</sup> 셋째, 개발 및 유지 보수에 관련된 모든 문서를 데이터 베이스화 하여 관련 정보를 효율적으로 검색하고 재사용 및 문서화에 관한 연구와<sup>17</sup> 넷째, 객체 지향 중심 방법에 대처할 수 있는 연구<sup>6</sup> 등으로 구별할 수 있다.

### 2.2 유지 보수 비용 예측 방법

소프트웨어 개발 비용을 예측하기 위한 방법으로는 COCOMO 모델<sup>21</sup>, Nordon/Rayleigh 인력 분배 모델<sup>14</sup>, Putnam 모델<sup>14</sup> 및 과거 산정 모델<sup>23</sup> 등이 있으며 이 모델들은 개발 노력(person-month), 개발기간, 원시코드의 길이(LOC) 또는 기능점(Function Point)<sup>11</sup>과 경험적 데이터를 이용하여 모델을 만들고, 새로운 다른 프로젝트를 예측하는 경험적 추정 모델이다. 또 유지보수 비용을 예측하기 위한 방법으로는 COCOMO 모델<sup>21</sup>, Card의 모델<sup>14</sup>, Oman의 모델<sup>17</sup> 및 Sneed의 모델<sup>20</sup> 등이 있다.

이 모델들은 개발의 관점에서 개발의 총 비용을 예측하는 방법과 유지 보수의 관점에서 비용을 산정하는 방법이 있는데, 지금까지 연구된 대부분은 전자에 해당하는 것으로, 본 연구는 유지 보수의 관점에서 유지 보수의 비용과 시간 및 관련 사항을 예측하고자 한다.

유지 보수를 위한 자동화 도구에 관련된 연구는 외국의 경우에는 연구가 활발히 이루어지고 있으며 상용화된 것도 있다. 그러나 이러한 연구와 상용화된 도구를 국내의 환경에 응용하기 위한 연구는 거의 이루어지지 않고 있다<sup>26,27</sup>.

## 3. 요구 사항 분석과 개략 설계

### 3.1 현황 조사 및 분석<sup>[22,25,26]</sup>

국내 유지 보수의 현황을 파악하고 유지 보수의 절차와 방법을 조사하며 유지 보수의 근원적인 문제점을 분석하기 위하여 150여 업체에 설문지를 발송하고 58매를 회수하였다. 조사 대상 기업 중 전산담당 인

원이 10명 미만이고 자체 업무만을 개발 유지 보수하는 기업이거나, 설문지 내용을 제대로 파악하지 못하여 문항별 관련성에 위배되는 설문을 작성한 12개를 제외한, 46개만을 처리 대상으로 하였다. 설문 응답자의 20% 이상이 자료로 이용 될 수 없음은 소프트웨어 공학에 대한 기본 인식이 부족함을 단적으로 나타내고 있어 자료의 활용 및 신뢰성에 문제점이 있었다. 따라서 자료의 신뢰성을 향상하기 위하여 면담을 통하여 확인하고 기존의 연구 결과나 문헌을 참고 하였다<sup>10,29)</sup>.

3.2 유지보수 생명주기의 정의

유지보수 방법은 대상 업무의 개발 유형, 개인의 경력이나 경험 및 업무의 이해 정도에 따라 큰 차이를 보이고 있으나, 유지보수 전담 조직을 가지고 있는 40개 업체를 대상으로 프로세스의 중요도, 어려움 및 소요 공수를 산정한 결과는 <표 3.1>과 같다.

<표 3.1> 유지보수 프로세스와 소요공수  
<Table 3.1> Maintenance Process and Required Effort

프로세스	중요도	어려움	소요공수
① 사용자 요구사항 검토	2%	12%	5-8%
② 업무의 범위 파악	3%	5%	5-10%
③ 일정 계획의 수립	5%	25%	3-5%
④ S/W 시스템의 이해	15%	8%	10-20%
⑤ 프로그램 이해와 설계	28%	7%	20-30%
⑥ 코딩	1%	5%	5-10%
⑦ 시험 및 검증	26%	32%	20-30%
⑧ 변경관리 및 문서화	18%	0%	5-10%
⑨ 기타	2%	6%	5-8%

<표 3.1>에서 중요도는 제시된 프로세스 중에서 가장 중요한 프로세스를 선택한 수치이며, 어려움은 총 어려움 중 각 단계에서 발생한 상대적 수치이고, 소요 공수는 각 프로세스에서 소요되는 상대적 시간 비율이다.

유지보수 생명주기는 <표 3.1>의 9단계의 사용자 요구사항 검토 및 업무 파악을 하나로 묶었으며, 일정 계획 수립 단계를 제외한 6단계를 기본 단계로 하였다. 이 6단계를 조사 대상 업체에서 실제로 처리하는 순서대로 기재하도록 하여 동일 처리 방식을 가진 부분들을 그룹화하여 생명주기의 원형으로 삼았다.

1) 전형적 역추적 모형

대상 업무가 비교적 단순한 업무이거나 소규모 시스템인 경우나, 중규모 이상일지라도 개발 프로젝트에 직접 참여한 경우, 업무의 규모에 관계없이 장기간 유지보수 업무를 수행하여 업무와 프로그램에 상당한 지식이 있을 경우에는 ①→②→③→④→③→②→④→⑤→⑥의 단계로 ③, ②, ④의 과정을 프로그램 중심으로 역추적 하면서 유지보수를 수행한다.

2) 개발 프로세스 추적 모형

대상 업무가 중규모 이상이고 업무에 대한 사전 지식이 없거나 부족한 경우, 또는 유지보수의 규모가 방대하거나 시스템 통합을 위한 경우에 사용하는 모형으로 개발의 전 과정을 점진적으로 이해하여 ①→②→③→④→⑤→⑥의 단계로 수행하는 모형이다.

3) 임의 추적 모형

대상 업무의 규모나 특성에 관계없이 유지보수자의 개별적인 특성에 따라 특정한 수행 과정이 없이 ②, ③, ④, ⑤의 과정을 임의적으로 수행하는 방법으로 설문 응답의 70%가 이 모형에 속한다.

이 모형은 사용자가 제출한 변경 의뢰서의 핵심 되는 문장이나 단어에서 프로그램에 관련된 핵심어(Keyword)를 추출하여, 개발 및 유지보수 조직에서 관리하고 있는 문서를 임의적으로 참조하면서, 프로그램과의 상호관계를 파악(Mapping)하여 처리하는 모형이다.

4) 동적 유지보수 모형

1), 2), 3)의 유지보수 모형은 각각 독특한 처리 방법을 가지고 있으나 전체적으로는 개발 및 유지보수 과정에서 작성된 문서와 원시코드에 관련된다. 본 논문에서는 유지보수 업무의 기능 및 수행자 모형을 기본으로 처리 과정 중심 모형을 포괄적으로 운용하는 생명주기 모형을 추상화하여 '동적 유지보수 생명주기'라 정의하고 유지보수 관리 체계 설계의 원형으로 삼는다.

업무의 계획 단계는 관리자가 수행하여야 할 기본 기능으로 업무의 진행 계획 및 관리 통제를 위한 방법과 수단을 제공하며, 유지보수자의 업무 진행 내역

을 관리 통제 한다. 소프트웨어의 이해와 설계 단계는 처리 절차 중심의 1, 2, 3, 4 단계 중 유지보수자의 기능을 통합한 과정으로 전 과정의 50% 이상을 차지하는 가장 중요한 부분이다. 시험 및 품질 관리 단계는 유지보수자의 처리 내역을 관리자와 품질 관리자가 점검하고 재사용을 위한 준비를 수행하는 기능으로 처리 과정 중심의 3, 4 단계에 해당한다.

### 3.3 유지보수 지원 시스템의 기본 전략

#### (1) 역공학 방법에 기초한 유지보수 지원 방법

국내 유지보수의 환경에서 사용하는 문서로는 원시 코드가 80% 이상을 차지하므로 원시코드를 이용하여 문서화하고 유지보수를 지원할 수 있어야 한다<sup>125,261</sup>. 또 설문 조사의 결과 75% 이상이 임의 추적 모형을 택하고 있으므로 유지보수 절차는 임의 추적 모델을 기본 연구 대상으로 하되, 다른 생명주기를 포괄할 수 있도록 각 생명주기 단계별로 자동화 할 수 있도록 한다.

#### (2) 유지보수 관리 체계의 정형화

유지 보수의 관리 체계가 부분적이라도 정의된 기관은 72%이나 그 규정을 90% 이상 잘 지키지 않아 규정의 필요성에 대하여 부정적인 견해를 가지고 있다. 또 유지보수 관리자의 역할에 대해서는 부정적이며 거의 모든 일을 유지 보수자의 책임하에 처리 운영되고 있다. 그러나 계획 수립, 요구 사항의 분석 및 업무책임의 부분에는 70% 이상이 관리자의 역할로 생각하고 있다. 이는 기존 유지보수 환경이 체계화된 유지보수 방법론이나 방법론에 입각한 관리 체계나 도구를 갖고 있지 않음을 의미한다.

#### (3) 유지 보수를 위한 예측 전략의 필요

유지보수 관리 기능의 가장 중요한 부분은 관리와 통제이나, 이를 위한 적절한 방법이나 도구 특히 예측 방법이 없다. Jackson<sup>111</sup>에 따르면 유지보수자가 직면한 장애 중에서 잠재적인 변경에 대한 파급효과를 인지할 수 없음을 가장 큰 장애 요인으로 지적하고 있다. 즉 변경에 따른 의미적인 효과를 사용자가 이해할 수 있는 형태로 요약 보고할 수 있는 시스템의 필요성을 지적하고 있다.

따라서 유지보수시 예측 전략의 수립과 세부 지침 및 방법을 결정하는 것은 더욱 용이하게 하는 인자로 취급될 수 있다. 유지보수 대상 항목에 관한 구체적

인 분석 자료를 유지보수자에게 전달하는 방법이며, 기본적인 요구 사항으로는 사용자 중심의 예측결과를 이해 가능한 형태의 접속 관계를 통해서 제시할 수 있는 환경의 구축으로 정의한다.

#### (4) 유지보수자의 특성에 적합한 방법 및 절차 생성의 표준화 필요성

Krogstic<sup>121</sup>는 개개인의 능력차나 환경적인 요소가 유지보수의 성패를 좌우하는 인자이며 이에 따른 세부 유지보수 활동이 결정되어야 한다고 지적하고 있다. 즉, 유지 보수 활동의 세부 프로세스는 유지보수자의 경험, 능력 및 개인적인 특성에 따라 차별화 해야 하며, 개인의 특성을 최대한 보장하면서 전체적인 통합화로 유도하기 위한 방법과 절차를 생성할 수 있어야 한다. 특히 개별적인 유지보수 특성이 규정이나 절차에 의해 생산성에 저해가 되지 않도록 유연성을 부과 하여야 하며, 개인적인 특성이 자연스럽게 표준적인 절차로 유도되고 종합적인 생산성 향상을 위한 연구의 기초 자료로 생성될 수 있도록 하여야 한다.

#### (5) 변경 관리 및 문서화의 필요성

유지보수에서 가장 중요한 부문은 개발 환경 및 시스템의 이해이다. 그러나 이해를 도와줄 수 있는 문서가 표준화 되어 있지 않고, 수정에 따른 변경처리가 되어 있지 않아 거의 활용할 수 없는 상황이다. 참조하여야 할 문서는 표준화된 최소 분량으로 하고 별도의 문서화 작업 없이 자동화 될 수 있어야 한다<sup>131</sup>.

#### (6) 자료 저장소의 관리 기능

유지보수 관리 운영을 위한 지원 시스템의 모든 자료들은 물론 다른 시스템과의 호환성을 가지며 새로운 개발 및 운용 도구들과도 효율적으로 인터페이스할 수 있도록 설계되어야 미래 지향적인 유지보수 시스템이 구축될 수 있다<sup>141</sup>. 이를 위해서는 레포지토리의 설계가 필수적이다.

#### (7) 유지보수자 중심의 이해력 지원 및 통합화 도구의 지원

유지 보수의 핵심은 유지 보수자의 경력, 특성 및 환경에 적합한 자료를 제공하여 가장 효율적으로 시스템을 이해하게 하는 것이므로, 도구 설계의 기본 전략은 유지보수자 중심의 이해력을 향상 시킬 수 있어야 한다<sup>15,161</sup>.

### 3.4 유지보수 시스템의 개략 설계

3.4.1 시스템 설계 환경

시스템 설계 환경은 Client/Server 환경하에서 기본 처리 기능은 PC에서 수행하며 시스템의 관리 및 레포지토리의 운영은 Workstation에서 수행하도록 하며, 구현 언어로는, C/C++를 채택하였다.

3.4.2 유지보수 지원 시스템의 개략 설계

유지보수 지원 시스템은 (그림 1)과 같이 9개의 부시스템으로 구성되며 그 기능은 다음과 같다.

(1)원시 코드의 분석 및 재구성기

이 시스템은 원시 코드 분석 기능과 표준화 프로그램으로 전환하는 2개의 기능이 있다.

원시 코드 분석 기능은 다양한 형태로 코딩된 원시 코드를 최소 단위의 모듈 즉 Proper/Prime 단위로 분해한 C언어를 기반으로 한 표준 모듈 명세서와 모듈 단위의 프로그램 처리 절차표를 작성한다. 모듈 명세서에는 모듈명, 기능명, 유형, 기본 언어, 크기, 상·하위 모듈, 매개 변수, 전역 변수, 지역 변수 등이 상세히 기록된다.

표준화 프로그램 재구성기는 기 작성된 모듈 명세서와 프로그램 처리 절차표를 이용하여 표준 프로그램으로 재구성 한다.

(2)변경 의뢰서 분석기

유지 보수를 위하여 사용자가 작성한 변경 의뢰서에 기재된 사항을 기록하고 분석하여 유지보수 환경 분석 및 절차의 생성에 사용할 기본 자료를 만들고, 유지보수 예측 모델에 사용할 자료를 생성한다.

(3)시스템 환경 분석 및 절차 생성기

시스템 환경 분석기는 변경 의뢰서 분석기에서 작성된 유지 보수자의 특성, 경력 및 시스템 환경 자료와 원시 코드 재구성기에서 작성된 자료들을 결합하여 소프트웨어 시스템을 효율적으로 이해하기 위한 이해의 절차를 제시한다. 제시된 절차는 유지 보수자나 관리자의 특성에 따라 변경 가능하고, 그 절차를 따라 이해하며 필요한 참고 문서를 열람할 수 있도록 한다.

이 과정에서 유지보수 생명주기를 생성하고, 각 단계에서 사용할 문서를 지정하게 하여 관리자 및 개인의 진행 사항을 통제할 수 있도록 한다.

(4)문서 생성기

유지보수 과정에 사용되어질 모든 문서를 생성하는 단계로 원시 코드로 생성될 수 있는 자료는 물론 개발시 작성된 자료도 열람할 수 있도록 한다. 초기 단계에는 인적 개입이 필요하다.

(5)유지보수 예측기

변경 의뢰서 분석기에서 작성된 자료와 원시 코드 분석 및 재구성기에서 작성된 자료를 기준으로 완료 예정일자에 적합한 노력 즉 인원 배치 계획 혹은 유지 보수자에 따른 일정 계획을 예측한다.

(6)코드의 수정 및 시험·검증기

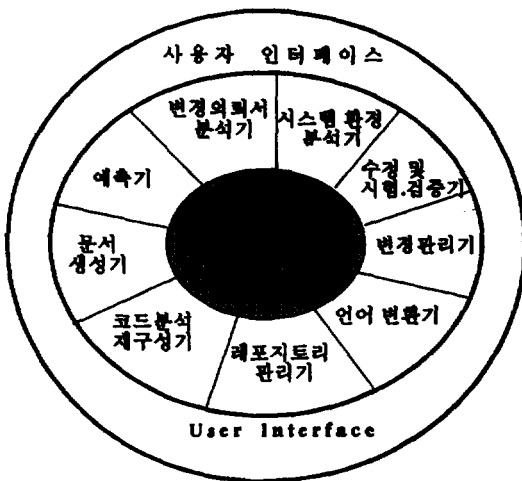
원시 코드를 수정하고 수정된 내용을 시험 검증한다. 시스템에 관련된 모든 모듈들이 검증될 수 있도록 강제 유도하며, 시험 사례들이 보관된다.

(7)변경 처리 및 문서화

시험 완료된 원시 코드는 Version Control될 수 있도록 하며, 재사용 가능한 모듈들은 등록한다. 또 완료된 원시 코드는 재 문서화하여 필요 시 열람 될 수 있도록 한다.

(8)언어 변환기

언어 변환기 유지 보수자가 다양한 언어에 효율적으로 대처하기 위하여 표준화된 언어를 유지보수자의 특성에 맞게 변환하여 처리할 수 있도록 하여야 한다. 특히 새로운 객체 지향 언어나 4GL과 같은 비



(그림 1) 유지보수 지원 시스템 구성도

(Fig. 1) A Configuration of the Maintenance Support System

절차 언어로의 전환 방법도 고려한다.

(9) 레포지토리 관리기

유지보수 지원 시스템의 모든 처리 과정과 처리데이터는 레포지토리에 의해서 관리 운영된다. 타 시스템이나 새로운 개발 도구의 활용은 레포지토리를 통해서 이루어 지도록 한다.

4. 유지보수 관리를 위한 비용 예측 모델

4.1 유지 보수의 비용 예측

유지 보수의 관점에서 비용을 예측하는 것은 개발의 관점에서 예측하는 것과는 목적과 처리 방법이 근본적으로 상이하다. 유지 보수의 관점에서 비용을 예측하는 것은 관리자에게 유지보수 과정을 관리 통제할 수 있도록 하며 유지보수 자체의 신뢰성을 향상시키고 유지 보수자에게 업무의 처리 과정과 문제점을 스스로 인식할 수 있게 하고자 한다.

유지 보수 비용의 예측은 다양한 개발 환경과 유지 보수자의 특성에 따라 그 변화가 심할 뿐 아니라 표준적인 관리 지침도 없으며 관련 연구를 수행하기 위한 자료가 전무한 실정이다. 그러나 3.2절과 같은 유지보수 생명주기와 각 단계별 소요 공수가 정의되고, 처리되어야 할 원시 코드가 표준화되면(3.4.2절의 (1)과 (2)에 의해) 예측이 가능해 진다. 즉 생명주기 각 단계에서 소요되는 공수를 이해하여야 할 프로그램의 크기로 산정하고, 이 프로그램의 크기와 실제 사용된 노력을 이용한 경험적 고정 단일 변수 모형을 만들어 프로그램의 크기에 따른 소요 공수를 계산할 수 있다.

유지보수 비용은 유지 보수자의 특성과 시스템 특성 즉 소프트웨어의 복잡도, 유지 보수자의 숙련도, 요구되는 신뢰도 및 완료 예정일자에 따라 소요되는 노력이 달라지므로 이를 조정할 수 있는 노력보정계수를 산정하여 점진적으로 정확도를 높일 수 있도록 설계한다.

4.2 유지보수 비용 모델의 설계

(1) 유지보수 비용 예측의 기본 모형

유지보수 비용 예측의 기본 모형은 COCOMO와 같이 프로그램의 크기와 노력에 대한 경험적 자료를 최소 자승법을 사용하여 (4-1)과 같은 고정 단일 변수

모형을 초기 모델로 사용하고, 노력보정계수를 사용하여 (4-2)와 같은 상세 모델을 작성한다.

$$\text{Effort} = a (\text{KLOC})^b \tag{4-1}$$

$$\text{Effort} = a (\text{KLOC})^b * \text{노력보정계수} \tag{4-2}$$

(2) 이해를 위한 프로그램의 크기의 산정

기본 모델 작성시에 사용하는 프로그램의 크기는 유지보수 기간에 발생하는 모든 노력의 합으로 표현하여야 하나, 본 연구에서는 변경 의뢰서에서 수정 제시한 소프트웨어를 이해하는데 필요한 노력을 이해하는데 필요한 코드의 크기로 환산하는 알고리즘을 (4-3)과 (4-4)와 같이 작성하였다.

모듈의 크기

$$\begin{aligned} &= \text{모듈의 라인수} + \text{내·외부 함수의 수} * 10 \\ &+ \text{전역 변수의 수} * \text{단말 모듈의 총수} \end{aligned} \tag{4-3}$$

이해를 위한 총 라인수

$$\begin{aligned} &= \text{합}(\text{모듈}i \text{의 크기} * \text{모듈}i \text{의 깊이}) \\ &\text{상수 } 10: \text{표본 함수의 평균 이해 라인수} \end{aligned} \tag{4-4}$$

(3) 노력 보정 계수의 산정 방법

유지보수 노력은 유지 보수자의 특성과 시스템 환경의 세부 속성에 따라 변화가 심하나, 본 연구에서는 유지 보수자의 경력 수준, 관리자가 정의한 숙련도, 유지 보수의 유형 및 요구되는 신뢰도만을 감안하되, 가중치의 값은 자료의 통계 분석에 의한 값이 아니고 전문가 판단에 의한 추정치이며 산장식은 (4-5)와 같다.

$$\text{노력보정계수} = A * B * C * D \tag{4-5}$$

A. 유지 보수자의 경력 수준

- 1) 직접 개발한 업무의 유지보수(0.3)
- 2) 유사한 개발 경험과 경력 5년 이상(0.6)
- 3) 부분적 개발 참여했고 경력 3년 이상(1.0)
- 4) 개발 경력이 없고 경력만 2년 이상(1.5)
- 5) 유지보수 경력 2년 미만(2.0)

B. 관리자가 정의한 숙련도

- 1) 탁월함(0.5)
- 2) 우수함(0.8)
- 3) 보통(1.0)
- 4) 약간 부족(1.3)

5) 아주 부족함(2.0)

C. 유지 보수의 유형

- 1) 적용보수(1.0)                      2) 수정보수(1.2)
- 3) 완전보수(1.3)                    4) 확장보수(1.4)

D. 요구되는 신뢰도

- 1) 보통수준(1.0)                    2) 중요함(1.5)
- 3) 매우 중요함(2.0)

4.3 예측 모델의 실험 및 평가

4.3.1 원시 자료의 생성과 모델 설정

(1) 원시 자료의 생성

유지보수 비용 예측 모델 설계를 위한 자료는, 현재 사용 중인 C 언어로 작성된 3개의 작성 형식이 서로 다른 프로그램을 선정 하였고, 경력 수준이 다른 3 사람에게 수정하도록 하였다. 이해 코드의 확산은 수작업으로 기본 자료를 작성하였고, 노력공수는 실제 수정에 소요된 일수이다. 실험 결과는 <표 4.1>과 같다.

<표 4.1> 유지보수 비용 예측 모델 실험 결과

<Table 4.1> A Experimental Result of the cost predictive model

프로젝트명	원시 코드	이해 코드	노력 공수	노력보정계수	
				A	B, C, D
PR #1	0.5	3.2	0.75	0.3	1.0
	0.5	3.2	2.5	1.0	1.0
	0.5	3.2	3.75	1.5	1.0
PR #2	1.2	8.4	1.8	0.3	1.0
	1.2	8.4	6.0	1.0	1.0
	1.2	8.4	9.0	1.5	1.0
PR #3	2.5	34.0	7.2	0.3	1.0
	2.5	34.0	24.0	1.0	1.0
	2.5	34.0	36.0	1.5	1.0

(2) 예측 모델의 설정

<표 4.1>의 결과 중 이해 코드와 노력공수를 변수로 하여 (4-1)식의 a, b 값을 구하면

$$Effort = 0.959536 (KLOC)^{0.803653} \text{ 이 된다.}$$

(3) 평가

(2)와 같은 유지보수 비용 예측 모델의 정확성을 검증하기 위하여, 세 사람에게 원시 코드 2.1 KLOC를 적용한 결과는 <표 4.2>와 같다.

실험의 결과 (4-1)식의 상수 값이 1에 근접하다는

<표 4.2> 유지보수 비용 예측 모델의 평가 결과  
<Table 4.2> A Evaluation Result of the cost predictive model

경험 수준	원시 코드	이해 코드	A	실험 결과	예측 결과	오차 (%)
A-1	2.1	20.4	0.3	6.0	4.35	27.5
A-3	2.1	20.4	1.0	13.0	14.51	11.6
A-5	2.1	20.4	1.5	18.0	21.76	20.8

것은 이해 코드 확산 알고리즘이 정확하다는 것을 의미하며, <표 4.2>에서 오차의 범위가 30% 이하의 오차가 발생한 것은 실험 데이터 선정에 문제가 있었다고 하더라도 초기 예측의 효과로는 탁월하다고 할 수 있으며, 노력보정계수의 조정에 의해 더욱 개선 될 수 있으리라 사료된다.

다만 실험의 조건이 제한적이고 업무의 성격과 범위가 소규모일 뿐만 아니라 충분한 자료를 작성할 수 없었던 것이 결점인 한 이틀 토대로 자료를 보완 수정해 나가면 좋은 모델이 될 것이다.

5. 결론 및 향후 과제

본 논문은 소프트웨어 위기를 초래하는 가장 큰 원인인 유지보수 문제를 해결하기 위한 방안 연구로서, 국내의 현실에 적합한 대안을 제시하고자 하였다. 국내 유지보수 문제점과 현황을 설문을 통하여 조사 분석하고, 설문에 기초한 문제의 해결책을 강구하고자 하였다. 그러나 국내 유지보수 환경의 미성숙으로 인하여 많은 해결책을 문헌과 면담에 의존할 수 밖에 없었으며, 문제 해결을 위하여 기본적인 몇 가지의 가정하에 연구를 계속하였다.

연구의 결과가 실험 자료의 부족으로 실험적 연구의 범위를 벗어나지 못했으나, 유지 보수의 생명주기를 정의할 수 있었고, 유지 보수를 효율적으로 진행하기 위한 기본 전략을 수립하였으며, 기본 전략에 따른 개략적인 시스템을 설계하였다. 또 유지보수 관리 체계 정립을 위하여 필수적인 요소인 비용 예측 모델을 수립하기 위한 방법을 제시 함으로서 관리자의 기능 즉 통제와 운영이라는 중요한 역할을 효율적으로 수행할 수 있게 되었다. 비록 제시된 모델이 제한된 여건과 충분하지 못한 데이터로 설정되었다고 하여도 모델의 응용 가능성은 충분히 인정될 수 있어 기업이나 업체에서 변용하여 활용할 수 있다.

향후 기본 설계된 시스템을 부 시스템별로 계속 연구해 나가면 좋은 자동화 시스템이 구축될 수 있을 것이라는 확신을 가지게 되었다. 특히 관리 체계를 정형화하기 위한 방법은 종래의 유지 보수에 대한 관리 방법을 근본적으로 개선할 수 있고 유지보수 자체의 신뢰성을 향상 시킬 수 있다.

본 연구는 완성된 연구의 결과로서의 의미보다 유지 보수를 위한 새로운 접근 방법을 제시하는데 있으며, 유지보수 통합 자동화 도구로 발전시키기 위한 기초 자료가 작성되기를 기대한다.

### 참 고 문 헌

- [1] A.J.Albrecht, J.E.Gafney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," IEEE Trans.on S.E., Vol. 16, SE-9, No. 6, pp.639-650, Nov. 1983.
- [2] T.J.Biggerstaff, "Design Recovery for Maintenance and Reuse," IEEE Computer, Jul. 1989, pp. 36-49
- [3] B.W.Boehm, Software Engineering Economics, Prentice Hall, 1981.
- [4] D.N.Card, D.V. Cotnoir, "Managing Software Managing Cost and Quality," Proc. of the 1987 IEEE Conf. on Software Maintenance, 1987, pp. 145-152
- [5] R. Cherinka C.M. Overstreet R.Sparks, "Building an Integrated Software Maintenance Environment-From a Maintainer's Perspective.," Proc. of the 1993 IEEE conf. on Software Maintenance, pp.31-40
- [6] W.C. Chu S.Patel, "Software ReStructuring by Enforcing Localization and Information Hiding," Proc. of the 1992 IEEE conf. on Software Maintenance, 1992, pp.165-172
- [7] D.A. Gustafson A.C. Melton Kyunghhee\_An Le-hong\_Lim, "Software Maintenance Models," IEEE Computer Society Press Tutorial: Software Maintenance and Computers, 1990, pp.15-22
- [8] G.T. Heineman G.E. Kaiser, "Incremental Process Support for Code ReEngineering," Proc. of the 1994 IEEE conf. on Software Maintenance, pp.282-290
- [9] D.Kung L.Gao ..., "Change Impact Identification in Object Oriented Software Maintenance," Proc. of the 1994 IEEE conf. on Software Maintenance, pp.202-211
- [10] D. Ince, "ISO 9001 and Software Quality Assurance," Quality Forum, 1994.
- [11] D. Jackson and D.A.Ladd, "Semantic Diff: A tool for summarizing the effects of modifications," IEEE, 1994.
- [12] J. Krogstic, "Software Maintenance in Norway: A survey Investigation," IEEE, 1994.
- [13] L.D. Landis P.M. Hyland ..., "Documentation in a Software Maintenance Environment," Proc. of 1988 IEEE Conf. on Software Maintenance, pp. 66-73
- [14] B. Londeix, "Cost Estimation for Software Development," Addison-Wesley, 1987.
- [15] P. Oman and J. Hagemester, "Metrics for Assessing a Software System's Maintainability," Proc. of IEEE Conference on Software Maintenance, pp.337-344, 1992.
- [16] P.Lino P.Aubet ..., "CARE: An Environment for Understanding and ReEngineering C Programs," Proc. of the 1993 IEEE conf. on Software Maintenance, pp.130-139
- [17] C. McClure, The Three Rs of Software Automation, Prentice Hall, Englewood Cliffs, NJ 07632, 1992.
- [18] N.F. Schneidewind, "The State of Software Maintenance," IEEE CSP Tutorial: Software Maintenance and Computers, 1990, pp.15-22
- [19] T.Skramstad M.K.Khan, "A Refined Software Life Cycle Model for Improved Maintenance," Proc. of the 1992 IEEE conf. on Software Maintenance, 1992, pp.193-197
- [20] H.H. Sneed, "Estimating the Costs of Software Maintenance Tasks," Proc. of the 1995 IEEE conf. on Software Maintenance, pp.168-181



- [21] C.R. Symons, Software Sizing and Estimating MK II FPA, Willy, 1993.
- [22] 과학기술처 "소프트웨어 관리체계에 대한 실태 조사, 분석연구," 1990.
- [23] 과학기술처 고시 94-1호, 1994.
- [24] 김제수, 소프트웨어 프로젝트 비용 산정 요소에 관한 경험적 연구, 숭실대학교, 석사학위논문, 1990. 6.
- [25] 류성열, 이성은, 이기오, 신현정, "역공학 방법에 기초한 유지보수 이해력 지원도구의 설계에 관한 연구," 숭실대학교 생산기술연구소 논문집 제25집(제2호), 1995, pp.85-92
- [26] 류성열, 소프트웨어 유지보수 형태에 대한 설문, 숭실대학교 소프트웨어공학 연구실, 1995. 9.
- [27] 우치수, "정보시스템 구축기술 개발(재공학방법론)," 과학기술처, 1995. 8.
- [28] 이주현, 실용 소프트웨어 공학론, 범영사, 1993.



**류 성 열**

- 1977년 숭실대학교 전산학(학사)
- 1980년 연세대학교 산업대학원(석사)
- 1996년 아주대학교 전산학(박사)
- 1981년 전국 금융협회 고문
- 1986년 숭실대학교 중앙전자계산소 소장

1994년 한국정보과학회 총무이사  
 1981년~현재 숭실대학교 정보과학대학 컴퓨터학부 교수  
 관심분야: S/W 재사용, S/W 리엔지니어링, 객체지향 시스템



**백 인 섭**

- 1969년 서울대 전기공학과(학사)
- 1975년 KAIST 전자계산학(석사)
- 1981년 프랑스 ENSIMAG 전자계산학(박사)
- 1981년~1983년 프랑스 국립 통계연구소(선임 연구원)

1983년~1988년 한국 데이콤 정보통신연구소장  
 1988년~1992년 한국 전산원(연구위원)  
 1992년~현재 아주대학교 정보 및 컴퓨터공학부 교수  
 관심분야: DB, SE



**김 하 진**

- 1962년 서울대학교 문리과대학 수학과 (이학사).
- 1978년 Grenoble 1 대학교 대학원 응용수학과 D. E. A. (이학석사).
- 1980년 Saint-Etienne 대학교 대학원 응용수학과 (이학박사).

1984년~1985년 프랑스 INRIA 초빙교수.  
 1989년~1992년 한국 정보과학회 부회장 및 회장.  
 1993년~1995년 아주대학교 공과대학 학장.  
 1974년~현재 아주대학교 공과대학 정보 및 컴퓨터공학부 교수  
 관심분야: 컴퓨터그래픽스, 수치해석 등.