

컴포넌트 소프트웨어의 유지보수를 위한 비용 측정지표 및 비용 추정 모델

조 주 연[†] · 류 성 열^{††}

요 약

대부분의 정보시스템은 컴포넌트 기반으로 아웃소싱에 의해 개발되고 있으며, 개발된 소프트웨어들은 통합되어 유지보수 되고 있다. 그러나 유지보수의 생산성과 효율성 제고를 위한 기반 연구인 비용 측정지표 및 비용 추정 모델에 관한 연구가 부족하다.

본 연구는 컴포넌트 기반의 소프트웨어에 대한 유지보수의 비용에 대한 측정지표를 제안하고, 제안한 측정지표의 의한 컴포넌트 소프트웨어의 유지보수 비용 추정 모델을 연구한다.

컴포넌트 기반의 유지보수 비용 측정지표를 추정하기 위하여 기존에 제시된 지표들을 비교 정리하고, 정리된 측정지표가 컴포넌트 소프트웨어 유지보수 유형에 따라 어떻게 적용되는지를 분류하고 유형별 비용 측정지표를 제안한다. 또, 제안한 측정지표를 이용한 컴포넌트 소프트웨어 유지보수 유형에 따른 비용 추정 모델을 제안한다. 제안한 추정 모델의 사례연구를 실시하여 유효성을 검증한다.

키워드 : 유지보수, 측정지표, 유지보수 비용, 유지보수 비용 추정 모델

Cost Measurement Indicators and Maintenance Cost Estimation Model for Maintenance of Component-based Software

Ju-Yeon Jo[†] · Sung-Yul Rhew^{††}

ABSTRACT

Most information systems are component-based and developed by outsourcing, and developed software is maintained integrative. However, studies on cost measurement indicators and cost estimation model have not been performed sufficiently, which are foundational to enhance the productivity and efficiency of maintenance. This study suggests indicators to measure maintenance cost for component-based software and examines maintenance cost estimation model of component software by the measurement indicators suggested. In order to generate the indicators to measure the component-based maintenance cost, the previously proposed indicators are summarized comparatively. To estimate the measurement indicators of component-based software, it makes a comparison of the previously proposed indicators and arranges them. We classify the measurement indicators by how to apply according to maintenance types of component software and then we propose the cost measurement indicators. Moreover, we propose the cost estimation model according to the maintenance types of component software using the suggested measurement indicators suggested. With the suggested estimation model, a case study is performed and its validity is verified.

Keywords : Maintenance, Measurement Indicators, Maintenance Cost, Maintenance Cost Estimation Model

1. 서 론

다양한 기술의 발달로 사용자의 요구사항은 다양하고 복잡하게 나타나고 있으며, 이를 반영하기 위한 소프트웨어 시스템의 규모는 더욱 커지며 복잡해지고 있다. 다양한 사

용자 요구를 빠르고 정확하게 반영하기 위해 재사용 기술인 CBD 개발방법론(CBD : Component-Based Development)이 대두되었다[1].

하지만 재사용과 유지보수에 효율적인 컴포넌트를 사용하여 개발한 시스템이라도 소프트웨어의 결함과 오류, 기능개선, 환경 등으로 인해 유지보수 비용은 소프트웨어 생명주기의 약 70%를 차지할 만큼 증가하고 있다[2].

증가하는 비용 문제는 소프트웨어 위기를 가져온다. 이런 위기를 해결하기 위해 유지보수 비용 연구는 다른 연구 보다 선행되어야 한다. 비용연구의 핵심은 비용의 지표이며 이

* 본 연구는 숭실대학교 교내 연구지원비로 수행되었습니다.

† 준 회원 : 숭실대학교 컴퓨터학과 석사과정

†† 종신회원 : 숭실대학교 컴퓨터학과 교수

논문접수 : 2008년 12월 9일

수정일 : 1차 2009년 2월 26일

심사완료 : 2009년 2월 26일

지표들은 계속해서 측정되고 관리되어야 하지만 기반연구인 비용 측정 지표 및 비용 추정 모델에 관한 연구가 부족한 현실이다.

본 연구에서는 유지보수 관리를 효율적으로 하고 노력과 비용을 최소화하고자 기존에 제시된 유지보수 지표들을 비교 정리하고, 정리된 측정지표가 컴포넌트 소프트웨어 유지보수 유형에 따라 어떻게 적용되는지를 분류하고 유형별 비용 측정지표를 제안한다. 또, 제안한 측정지표를 이용한 컴포넌트 소프트웨어 유지보수 유형에 따른 비용 추정 모델을 제안한다. 제안한 추정 모델의 사례 연구를 실시하여 유효성을 검증한다.

제안하는 유지보수 측정 지표는 유지보수 비용을 예측하고 유지보수 관리를 효율적으로 할 수 있으며, 유지보수 생산성과 효율성을 높인다. 또, 컴포넌트 소프트웨어의 유지보수 비용을 정확하게 예측하여 수·발주자 모두를 만족시킬 수 있는 비용을 추정 할 수 있는 효과가 있다.

2. 관련 연구

2.1 소프트웨어 유지보수 비용 측정지표

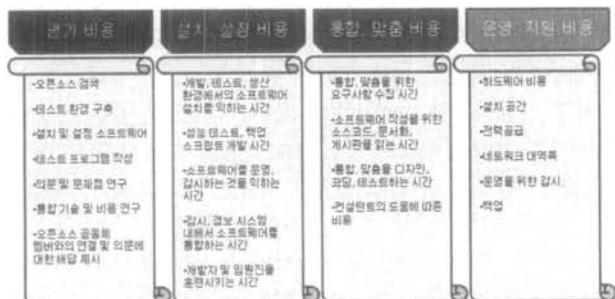
George E. Stark[6]는 소프트웨어 유지보수 시 비용에 영향을 미치는 요인을 크게 예상 기간, 유지보수 프로그램 기능추가/변경/삭제, 고객 만족도 이렇게 3가지 요소로 구분하였다. 유지보수 프로그램 기능추가/변경/삭제에 대한 구체적 예는 컴퓨터 언어, Logic, 입력, 데이터 처리, 출력, 인터페이스, 운영, 성능, 명세, 개선이다.

Hunt, B.; Turner, B.; McRitchie, K의 사례연구에서는 유지보수 시 유지보수기간, 인력, 도구, 형태, 자원, 프로그램 활동이 유지보수 비용에 영향을 미친다고 하였다[7].

Sommerville[8]는 유지보수 생산성 관련 요소로 모듈의 독립성, 프로그램 언어, 프로그램 스타일, 프로그램 검증, 문서화, 형상관리 이렇게 6개의 기술적 요소와 애플리케이션 영역, 요원의 안전성, 프로그램 나이, 외부환경, 하드웨어 안정성 이렇게 5개의 비기술적 요소로 도출하였다.

Gautam Guliani, Dan Woods는 오픈소스 소프트웨어의 각 비용 별 구성요소는 평가비용, 설치, 설정 비용, 통합 및출 비용, 운영·지원 비용으로 구분하여 (그림 1)과 같이 구성요소 별 각각의 지표를 제시하였다[9].

기존 연구에서는 유지보수 비용에 영향을 미치는 요인을



(그림 1) Gautam Guliani, Dan Woods 측정지표

구분하여 지표로 제시하였지만 요인별로 얼마나 영향을 미치는지 영향분석과 어떻게 측정하여 유지보수비용을 할당하였는지에 대한 비용 추정 방안이 제시되지 않은 문제점이 있다. 또, Gautam Guliani, Dan Woods의 측정지표는 오픈소스의 측정지표로 유지보수 비용에 관련되지 않은 지표도 제시하고 있으며 오픈소스 소프트웨어만을 대상으로 한 문제점이 있다.

2.2 컴포넌트 소프트웨어의 유형

컴포넌트 소프트웨어는 기 개발된 컴포넌트를 조립하여 개발되는 시스템을 의미한다. 컴포넌트 개발 방법에는 두 가지 형태의 개발 방법이 있다. 그중 하나는 컴포넌트 자체를 개발하는 CD(Component Development)이며 다른 방법은 이미 구축되어있는 컴포넌트를 사용하여 소프트웨어 시스템을 개발하는 CBSD(Component Based Software Development)이다. CBSD는 이미 개발되어있는 컴포넌트를 조립하여 비즈니스 요구사항을 충족시키는 것으로 컴포넌트를 그대로 조립하거나 특화(Customization)시킨 후 조립하여 사용하는 것이다[10].

기존 연구는 컴포넌트 개발에 따른 유형이지만 이는 컴포넌트 소프트웨어를 유지보수를 할 경우도 마찬가지로 적용된다. 따라서 기존 연구에 의해 유지보수의 유형을 분류하면 컴포넌트의 직접 재사용하는 경우와 커스터마이징(Customized)하는 경우, 컴포넌트를 개발하는 경우 3가지로 분류된다.

2.3 컴포넌트 소프트웨어 비용 추정 모델

컴포넌트 소프트웨어는 일반 소프트웨어 시스템과 프로세스 활동이 상이하여 비용 추정 모델이 따로 연구되고 적용되어왔다. 이러한 모델로는 SELECT Estimator, CotsCost, COCOTS등이 있다[1].

SELECT Estimator는 SELECT사의 제품군 중 비용 추정과 관련된 도구로 기본 생명주기를 기본으로 비즈니스 객체와 컴포넌트를 바탕으로 추정한다. 이 도구에서 채택하고 있는 비용 추정 모델은 Object Factory의 ObjectMetrix를 기반으로 소프트웨어를 요소로 분류하고, 분류된 요소를 바탕으로 소프트웨어 규모를 예측하고 있다[1, 5].

CotsCost 모델은 Loral Federal 시스템 사에 의해 개발된 COTS 기반 소프트웨어 비용 추정 모델이다. 이 모델은 기능 점수 기반의 규모 예측 방법을 사용하며, COCOMO 모델의 비용 유도자를 변형한 15가지 비용 유도자를 사용하여 비용을 예측하고 있다[1, 5].

COCOTS(CONstructive COTS Integration Cost Model)은 COCOMO모델의 확장 모델로서 COTS기반의 소프트웨어 비용 추정을 위해 사용되는 모델이다. 이 모델은 COTS 기반 소프트웨어 개발을 크게 3가지 활동, 즉 컴포넌트 평가(Assessment), 컴포넌트 수정(Tailoring), 글루 코드(Glue Code) 개발 및 테스트를 고려하고 있다. 또한 3가지 기본 활동에 컴포넌트 휘발성(Volatility)을 고려하여 비용을 추정한다[1, 8, 4].

황선명의 연구[1]에서는 화이트박스 컴포넌트에 한해 설계, 코드, 통합 수정 비율을 고려해 규모를 예측하고 평가와 이해도, 친숙도를 합하여 비용을 추정한다.

기존 컴포넌트 소프트웨어 비용 추정 모델은 컴포넌트의 개발과 재사용 관점의 비용 추정이므로 유지보수 보다는 개발과 재사용 관점에 비중을 두어 유지보수 조정 계수가 감안되지 않은 문제점이 있다. 또, CostCost와 COCOTS 모델은 블랙박스 컴포넌트인 경우의 비용추정 방안만을 제시하고 있어 그레이박스 컴포넌트, 화이트박스 컴포넌트의 추정 방안에 대한 연구가 요구되고 있다.

3. 컴포넌트 소프트웨어 유지보수의 비용 측정지표

본 장에서는 Administration Scorecard의 측정지표[11]와 기존 유지보수 측정 요소 및 지표, 컴포넌트 특성을 바탕으로 컴포넌트 소프트웨어의 유지보수 측정지표를 제안한다.

3.1 유지보수 비용 요소 비교

유지보수 비용 요소는 유지보수를 수행할 때 어떤 요인들이 비용에 영향을 줄 수 있는지 영향을 받을 수 있는지에 관한 요소들이다.

유지보수 비용 요소를 식별하기 위해서는 먼저 유지보수 관리자 스코어카드에서 제공하는 7가지 항목과 기존 연구에서의 요소를 비교한다.

유지보수 비용에 영향을 미치는 요소는 <표 1>에서와 같이 관리자 스코어카드의 측정지표와 관련연구 제 2.1절을 분석해 유지보수에 영향을 미치는 요소를 식별하였다. 식별된 16가지의 요소 중 중복되는 요소는 배제하였다.

<표 1> 유지보수 비용 요소 비교

측정 요소	Administration Scorecard	George E. Stark연구	Hunt, B.; Tumar, B.; McRitchie, K 연구	Sommerville 연구
예상기간	○	○	○	○
예상 자원	○		○	
유지보수 예비 크기	○			
작업계획 비율	○			
유지보수 프로그램 추가/변경/삭제	○	○	○	○
지연 작업	○			
잔여 관리 시간	○			
고객 만족도		○		
유지보수 인력			○	
유지보수 도구			○	
유지보수 형태			○	
명세		○		○
모듈의 독립성				○
프로그램 스타일				○
프로그램 검증				○
형상 관리				○

3.2 유지보수 비용 측정 지표

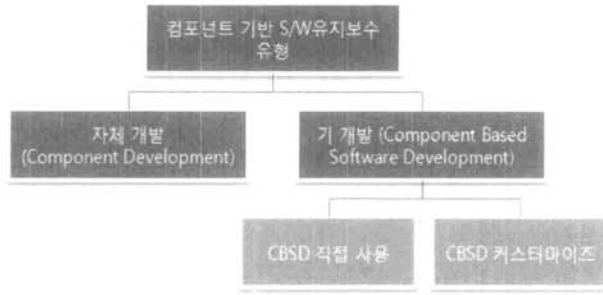
유지보수 비용 측정 지표는 유지보수 비용에 영향을 미치는 요소로 <표 1>에서 분석한 4개의 사례에 모두 만족하는 관리 요소 14가지이다. 이 중 프로그램 스타일과 검증은 유지보수 프로그램 추가/변경/삭제에 포함되므로 배제한다. 유지보수 비용 측정 지표는 <표 2>에서와 같이 총 14가지로 구성되며 유지보수의 총 기간과 프로세스별 기간, 유지보수 활동에 드는 자원, 유지보수 프로젝트 크기, 유지보수 작업의 우선순위와 그 비율을 추정하는 작업계획 비율, 유지보수 프로그램의 추가/변경/삭제에 대한 기능적 요소들, 지연되는 유지보수 기간과 인력, 크기, 유지보수 수행 후 잔여시간, 고객 만족 평가, 유지보수 기간당 인력, 도구의 활용도, 유지보수 형태, 유지보수 문서화 작업에 드는 인력, 시간과 문서 목록, 각 모듈의 독립성, 형상관리 항목과 형상관리 틀의 사용성이 있다.

<표 2> 유지보수 비용 측정 지표

유지보수 비용 측정 지표	지표 내용
기간	유지보수 기간(프로세스별 기간)
자원	유지보수 장비 목록, 장비 고장률
유지보수 크기	유지보수 프로젝트 크기
작업계획 비율	유지보수 우선순위
유지보수 프로그램 추가/변경/삭제	유지보수 단위, 프로그램 크기, 추가/변경/삭제된 LOC or FP
지연 작업	지연작업 기간, 인력, 크기
잔여 관리 시간	유지보수 활동 수행 후 잔여 시간
고객 만족도	고객 만족도 평가
유지보수 인력	유지보수 기간당 인력
유지보수 도구	유지보수 도구 활용도
유지보수 형태	유지보수 형태 분류 (수정, 적용, 완전, 예방)
명세	유지보수 활동에 따른 문서화 목록
모듈의 독립성	모듈의 독립성
형상관리	형상관리 항목과 틀의 사용성

3.3 컴포넌트 소프트웨어의 유지보수 유형별 측정 지표

관련연구 제 2.2절을 바탕으로 컴포넌트 소프트웨어 유지보수 유형을 분류하면 (그림 2)과 같이 크게 2가지로 나뉜다. 컴포넌트 자체를 개발하는 CD(Component Development), 다른 방법은 이미 구축되어있는 컴포넌트를 사용하여 소프트웨어 시스템을 개발하는 CBSD(Component Based Software Development)이다. CBSD는 이미 개발되어있는 컴포넌트를 조립하여 비즈니스 요구사항을 충족시키는 것으로 컴포넌트를 그대로 사용하여 조립하거나 커스터마이징(Customization)시킨 후 조립하여 사용하는 것이다.



(그림 2) 컴포넌트 소프트웨어의 유형

제안하는 측정지표는 컴포넌트 소프트웨어의 유형별로 컴포넌트 소프트웨어의 특성에 맞게 제 3.2절에서 제안한 유지보수 측정지표를 기반으로 제안하였다.

총 3가지 유지보수 유형에 따른 측정지표는 (그림 3)와 같다.

중속 측정지표	개발	CBSD 직접사용	CBSD 커스터마이즈
<ul style="list-style-type: none"> 컴포넌트 평가 기간 벤더의 성숙도에 따른 컴포넌트의 평가 레벨 (벤더의 성숙도는 업체의 실적으로 평가) 요구사항 수집 기간 지원 사용률 물의 라이선스 비용 유지보수 기간 요구원 작업 지연 기간, 인력, 크기 유지보수자 및 발주자 교육 시간 	<ul style="list-style-type: none"> 요구원(개발) 기능의 수 요구원(개발) 컴포넌트의 크기(ILOC, FP, 복안도) 모듈의 복잡성 컴포넌트 개발을 위한 유지보수자의 인원(인력) 개발된 컴포넌트의 테스트, 통합 시간 개발된 다른 명세 기간, 인력 	<ul style="list-style-type: none"> 컴포넌트 이해도 (유지보수자의 컴포넌트 이해도) 컴포넌트 검색 시간 컴포넌트 라이선스 비용 시스템 테스트, 통합의 기간, 인력 컴포넌트 명세 기간, 인력 	<ul style="list-style-type: none"> 컴포넌트 이해도 (유지보수자의 컴포넌트 이해도) 컴포넌트 설계수준비율 컴포넌트 코드수준비율 컴포넌트 통합수준비율 소프트웨어 커스터마이징을 위한 유지보수자의 인원(인력) 커스터마이징된 컴포넌트의 테스트, 통합 시간 커스터마이징에 따른 명세 기간, 인력

(그림 3) 컴포넌트 소프트웨어의 유지보수 유형별 측정지표

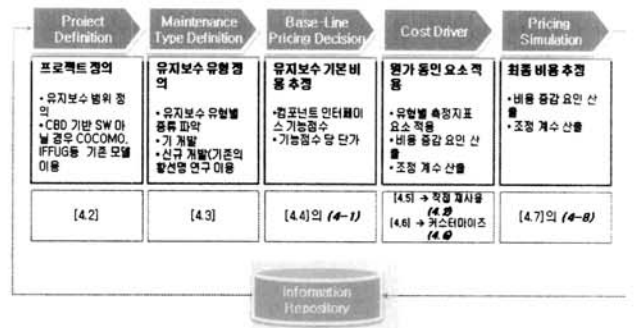
중속 측정지표는 유형에 관계없는 측정지표로 유지보수 시 사용되는 측정지표이다. 개발 측정지표는 요구사항에 맞는 기능의 컴포넌트를 개발하는데 사용되는 측정지표이다. CBSD 직접사용 측정지표는 기 개발된 컴포넌트를 수정 없이 그대로 직접 사용하는데 사용되는 측정지표이다. CBSD 커스터마이징 측정지표는 기 개발된 컴포넌트를 요구사항과 시스템에 맞게 수정하여 사용하는데 사용하는 측정지표이다.

중속 측정지표 중 컴포넌트 평가에 대한 레벨은 벤더의 성숙도에 따라 컴포넌트의 신뢰성과 만족도가 높기 때문에 과거의 실적을 파악하여 벤더의 실적 점수 또한 고려되어야 한다.

4. 컴포넌트 소프트웨어의 유지보수 비용 추정 절차 및 모델

4.1 컴포넌트 소프트웨어 유지보수 비용 추정 절차

비용 추정 절차는 프로젝트 정의, 유지보수 유형 정의, 유지보수 기본 비용 추정, 원가 동인요소 적용, 최종 비용 추정 총 5단계로 구성되어 있으며 전체 절차는 (그림 4)과 같다.



(그림 4) 컴포넌트 소프트웨어의 비용 추정 절차

4.2 프로젝트 정의(Project Definition)

프로젝트 정의는 유지보수 범위를 정의하고 CBD기반 소프트웨어가 아닐 경우를 배제시키는 단계이다.

유지보수 범위는 소프트웨어의 모든 오류를 제거하고 사용자 및 발주자에게 인도 완료된 시점을 그 범위로 제한하며, 유지보수 전문 업체에 유지보수 업무를 아웃소싱하는 경우에 한하며, 컴포넌트 소프트웨어의 유지보수로 그 범위를 제한한다. 컴포넌트 소프트웨어 중 소스코드가 완전 공개되어있는 화이트박스 컴포넌트 및 일부 공개 되어있는 그 레이박스 컴포넌트가 포함된다.

또, 유지보수 유형 중 신규 컴포넌트 개발을 제외한 기 개발된 컴포넌트의 사용만을 범위로 한다.

본 연구의 유지보수 비용 추정 지표는 포괄적으로 제시하나 비용 추정 모델의 범위는 위와 같이 한정한다.

유지보수 범위가 확정되면 측정 지표 값을 추정하고 그에 따른 비용을 추정하는 방법과 추정한 측정 지표 값에 조정 인자를 적용하여 최종 비용을 추정한다. 추정한 비용에 관한 내용은 추후 유사한 유지보수 비용 추정을 위해 정보저장소(Information Repository)에 저장한다.

4.3 유지보수 유형 정의 (Maintenance Type Definition)

유지보수 유형 정의는 3장에서 분류한 컴포넌트 소프트웨어의 유지보수 유형에 따라 소프트웨어를 분류하는 단계이다. 이 단계에서는 기존 컴포넌트를 직접 재사용 할 것인지 커스터마이징하여 사용할 것인지에 대하여 구분한다.

4.4 컴포넌트 소프트웨어 유지보수의 기본비용(Base-Line Pricing Decision)

기존 컴포넌트 유지보수 비용 추정은 경험을 바탕으로 유지보수 기간과 인력에 따라 비용을 추정하는 수준이었다 컴포넌트 소프트웨어의 특징과 유형별로 비용을 추정하지 못해 유지보수 업무에 따라 비용이 추정되지 못하는 한계기 있었다. 이러한 문제점을 해결하고자 컴포넌트 기능 점수를 산정해 기능점수 당 단가를 곱하여 산정한다. 비용 산정 시 기능점수당 단가는 유지보수 대가산정 기준[12]의 단가를 바탕으로 각 기업에 맞게 조정한다.

기본비용(원가 동인요소 미 고려)
 = 컴포넌트 기능점수(규모) × 기능점수당 단가 (4-1)

4.5 기 개발 컴포넌트의 직접 사용 유지보수 비용

기 개발된 컴포넌트를 직접 사용하는 경우 유지보수 기본 비용에 (그림 5)의 종속 측정지표에 따른 종속 요소 가중치와 (그림 3)의 CBSD 직접사용 측정지표를 바탕으로 원가 동인요소 가중치를 곱하여 추정한다.

(그림 5)에서는 종속 측정지표 요소 중 자원 사용률과 벤더의 성숙도 레벨을 분류하여 각각의 가중치 보정계수 구분하여 비용을 추정 할 수 있다.

직접 사용의 지표에 따른 가중치는 컴포넌트 이해도, 컴포넌트 검색 시간, 컴포넌트 라이선스 비용, 시스템 테스트/통합 기간과 인력, 컴포넌트 명세 기간과 인력의 가중치를 더하여 추정한다.

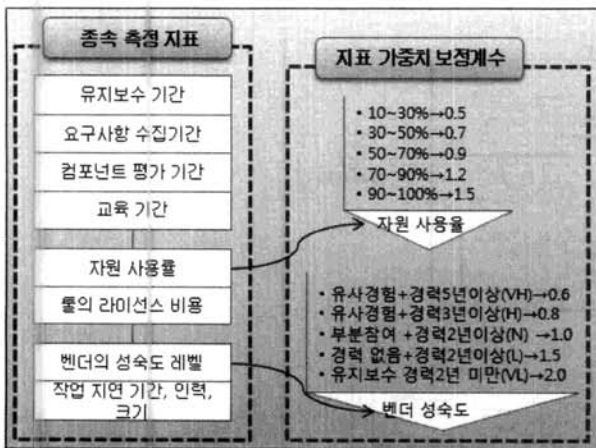
컴포넌트 이해도는 <표 3>의 황선명의 연구[1]에서의 SU(Software Understanding) 결정표에 따른다.

$$\text{직접사용비용} = \text{기본비용} \times \text{종속요소 가중치} \times \text{직접 사용 원가동인요소 가중치} \quad (4-2)$$

$$\text{종속요소 가중치} = \text{자원 사용률 보정계수} \times (\text{벤더 성숙도 수준 보정계수}) \quad (4-3)$$

$$\text{자원 사용률} = \frac{\text{사용하는 자원수}}{\text{전체 자원수}} \quad (4-4)$$

종속요소의 가중치는 제한한 지표를 바탕으로 자원사용률과 총자원 사용 수, 벤더의 성숙도 레벨의 보정계수를 곱하여 추정한다.



(그림 5) 종속 측정지표에 따른 종속요소 가중치

<표 3> 컴포넌트 이해도(SU)

항목	VL	L	N	H	VH
구조	매우 낮은 응집력, 높은 결함률, 스파게티 소스	보통 낮은 응집력, 높은 결함률	보통	높은 응집력, 낮은 결함률	강한 모듈성
명확성	프로그램과 데이터 유에 대한 일관성 결여	약간의 관련성 파악	보통	좋은 관련성 파악 가능	명백히 파악
설명	애매 모호한 코드	약간의 코멘트 및 해더	보통	좋은 코드 코멘트 존재	완전한 설명 존재
SU	50	40	30	20	10

$$\text{직접 사용 원가동인요소 가중치} = \text{SU} / 100 + \text{라이선스 비용} + (\text{테스트기간} \times \text{인력} \times \text{임금}) + (\text{명세기간} \times \text{인력} \times \text{임금}) \quad (4-5)$$

4.6 기 개발 컴포넌트의 커스터마이징 유지보수 비용

기 개발된 컴포넌트를 커스터마이징해서 사용하는 경우 유지보수 기본비용에 (그림 5)의 종속 요소 가중치와 (그림 3)의 CBSD 커스터마이징 측정지표를 바탕으로 원가 동인요소 가중치를 곱하여 추정한다. 직접 사용의 지표에 따른 가중치는 컴포넌트 이해도, 컴포넌트 설계(DM: Design Modified)/코드(CM: Code Modified)/통합 수정(IM: Integration Modified)비율, 커스터마이징 인력, 테스트/통합시간, 인력, 명세 기간/인력 가중치를 더하여 추정한다.

컴포넌트 이해도는 황선명의 연구[1]에서의 SU(Software Understanding) 결정표에 따른다.

$$\text{커스터마이징비용} = \text{기본비용} \times \text{종속요소 가중치} \times \text{커스터마이징 사용 원가동인요소 가중치} \quad (4-6)$$

$$\text{커스터마이징 원가동인요소 가중치} = \text{SU} / 100 + (0.4 \times \text{DM}) + (0.3 \times \text{CM}) + (0.3 \times \text{IM}) \times (\text{커스터마이징기간} \times \text{인력} \times \text{임금}) + (\text{테스트기간} \times \text{인력} \times \text{임금}) + (\text{명세기간} \times \text{인력} \times \text{임금}) \quad (4-7)$$

커스터마이징 비용은 직접 사용 비용과 달리 설계 수정, 코드 수정, 통합 수정의 비율과 그 가중치를 곱하여 커스터마이징 기간, 인력에 따른 비용과 테스트 기간, 인력에 따른 비용, 명세 기간과 인력에 따른 비용을 더하여 추정한다.

4.7 최종 비용

컴포넌트 소프트웨어 유지보수의 최종 비용은 기본비용에 종속요소 가중치와 유형별 원가 동인요소 가중치 비용을 더해 추정한다.

$$\text{최종비용} = \text{기본비용} \times \text{종속요소 가중치} + \text{유형별 원가동인요소 가중치비용} \quad (4-8)$$

5. 사례연구와 활용 방안

본 장에서는 실제 컴포넌트 소프트웨어를 선정하여 유지보수를 할 때의 비용을 예측한다. 본 연구에서 선정한 소프트웨어는 기존 최정운 논문에서 구축된 컴포넌트 기반의 신용카드 관리 시스템이다. 이 시스템은 다양한 기능을 구현하고 있으며 카드 기획관리, 회원관리, 카드관리, 가맹점관리, 카드 승인관리, 포인트 관리, 카드결제관리, 카드연체/사후관리, 제휴관리, 행사관리, 기타 서비스 관리 등으로 나타낼 수 있다.

프로젝트를 분석한 결과 회원관리 패키지과 카드관리 패키지 2가지로 구성되어 있으며 회원관리 패키지는 총 4개의

컴포넌트로 구성되며 회원모집관리 컴포넌트(CustReer), 회원심사관리 컴포넌트(CustExam), 회원기본정보관리 컴포넌트(CustInfo), 회원신용관리 컴포넌트(CustCred)이다. 카드관리 패키지는 카드심사관리 컴포넌트(CardExam), 카드기본정보관리 컴포넌트(CardInfo), 카드발급관리 컴포넌트(CardGive), 카드발주관리 컴포넌트(CardOrde), 카드배송관리 컴포넌트(CardSend)로 구성되어 있다.

신용카드 관리 패키지의 컴포넌트 다이어그램은 (그림 6)과 같으며 이 프로젝트의 유지보수 시 예측비용을 구한다.

단, 회원관리 패키지는 그대로 사용하며 카드관리 컴포넌트는 각 은행에 맞게 커스터마이징하여 사용한다. 직접 재사용하는 비용과 커스터마이징하는 경우의 비용을 추정한다.

5.1 기본비용 추정

구현된 시스템의 각 컴포넌트들의 인터페이스를 기능으로 분류해 확인해 본 결과 회원관리에는 4개의 기능이 구현되어 있으며 각 기능은 추가, 수정, 삭제의 기능을 가지고 있다. 또, 카드관리에는 5개의 기능이 구현되어 있으며 각 기능은 등록, 수정, 조회, 삭제의 기능을 가진다.

구현된 기능을 바탕으로 FP산정방법에 의해 비용을 산정할 수 있지만 이는 유지보수만을 위한 비용이 아니다. 그러므로 제 4장에서 제안한 비용추정식을 바탕으로 프로젝트의 유지보수 비용을 추정한다.

기본비용(원가 동인요소 미 고려)
 = 컴포넌트 기능점수(규모) × 기능점수당 단가
 = 103FP × 572,933원 = 59,012,099원

기본비용은 소프트웨어 대가 기준[12]에서의 기능점수 간 이법을 이용하여 기능점수를 구하고 그 단가를 곱하여 계산한다. 여기서 기능점수 당 단가는 유지보수 대가기준[12]에서 분석, 설계, 구현, 테스트단계의 비용을 합한 값이다.

5.2 컴포넌트의 직접 사용 유지보수 비용

이 중 회원가입 컴포넌트를 직접 재사용하는 비용을 구하면 다음과 같다.

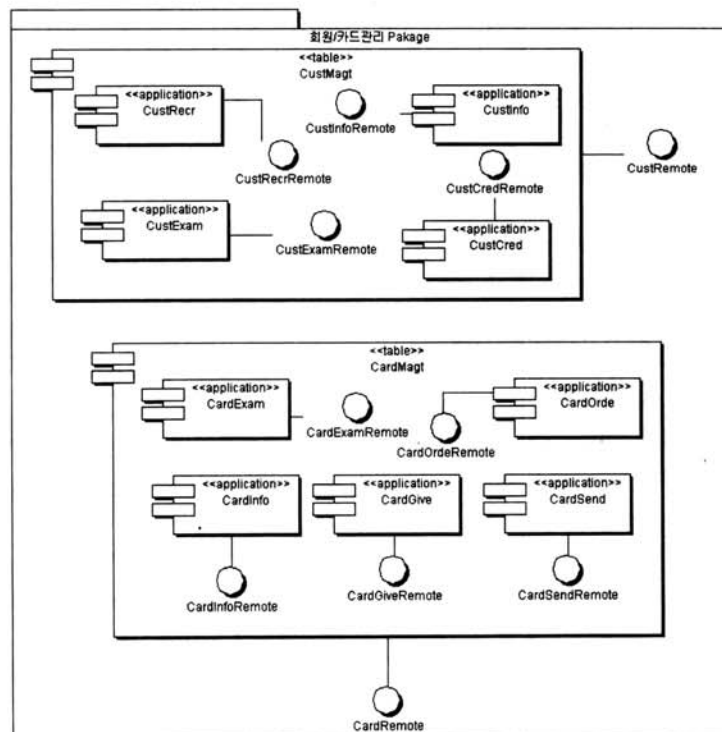
단, 유지보수의 가중치 요소는 아래와 같이 설정한다.

유지보수자의 컴포넌트 이해도가 보통 일 때(N) 테스트 기간은 10일, 테스트 인력은 3명이며 모두 중급기술자로 대가기준의[12] 노임단가를 기준으로 한다. 자원 사용율은 35%, 유지보수 벤더사는 유사경험이 있고 경력이 3년 이상 된 높은 레벨의 회사이다. 단, 이 컴포넌트는 오픈소스로 라이선스 비용은 제외한다.

종속요소 가중치 = 자원 사용율 0.7 × 벤더 0.8 = 0.56
 직접 사용 원가동인요소 가중치 비용
 = 0.3 × [(10 × 3 × 174,432) + (3 × 3 × 174,432)]
 = 0.3 × [5,232,960 + 1,569,888]
 = 0.3 × 6,802,848 = 2,040,854.4원

최종비용 = (59,012,099원 × 0.56) + 2,040,854.4원
 = 33,046,775원 + 2,040,854.4원 = 35,084,629.4원

컴포넌트 재사용 비용은 기본비용에 종속요소 가중치와 재사용 가중치 비용을 더한 것으로 최종 비용을 추정한다.



(그림 6) 신용카드 관리 패키지의 컴포넌트 다이어그램[13]

5.3 컴포넌트의 커스터마이징 유지보수 비용

이 중 회원가입 컴포넌트를 커스터마이징하는 비용을 구하면 다음과 같다.

단, 유지보수의 가중치 요소는 아래와 같이 설정한다.

유지보수자의 컴포넌트 이해도가 보통 일 때(N) 커스터마이징 인력은 3명, 임금은 모두 중급 기술자로 대가기준의 [12] 노임단가를 기준으로 한다. 테스트 기간은 10일, 테스트 인력은 3명이며 모두 중급기술자로 대가기준의 [12] 노임단가를 기준으로 한다. 자원 사용율은 35%, 유지보수 벤더사는 유사경험이 있고 경력이 3년 이상 된 높은 레벨의 회사이다. 단, 이 컴포넌트는 오픈소스로 라이선스 비용은 제외한다.

또, 컴포넌트의 기능 중 회원 모집관리, 회원심사 관리의 컴포넌트를 수정하여 사용할 것이며 수정 비율은 설계 10%, 코드 수정 30%, 통합 수정 30%로 한다.

$$\begin{aligned}
 \text{중속요소 가중치} &= \text{자원 사용율 } 0.7 \times \text{벤더 } 0.8 = 0.56 \\
 \text{커스터마이징 원가동인요소 가중치 비용} &= 0.3 + [(0.4 \times 0.1) + (0.3 \times 0.3) + (0.3 \times 0.3)] \\
 &\quad \times [(10 \times 3 \times 174,432) + (10 \times 3 \times 174,432) + (3 \times 3 \times 174,432)] \\
 &= 0.3 + 0.03 + 0.09 + 0.09 = 0.51 \\
 &= 0.51 \times [5,232,960 + 5,232,960 + 1,569,888] \\
 &= 0.51 \times 12,035,808 = 6,138,262.08 \\
 \text{최종비용} &= (59,012,099 \text{ 원} \times 0.56) + 6,138,262.08 \text{ 원} \\
 &= 33,046,775 \text{ 원} + 6,138,262.08 \text{ 원} \\
 &= 39,185,037.08 \text{ 원}
 \end{aligned}$$

컴포넌트 커스터마이징 비용은 기본비용에 중속요소 가중치와 커스터마이징 가중치 비용을 더한 것으로 최종 비용을 추정한다.

6. 결론 및 향후 연구

소프트웨어 프로젝트의 대형화, 복잡화에 따라 유지보수의 중요성이 점차 높아지고 그에 따른 비용이 증가하고 있다. 점점 증가하고 있는 소프트웨어 유지보수 비용을 최소화하고 관리하기 위해서는 유지보수를 위한 자료를 수집, 분석하는 활동을 해야 하며 분석한 자료를 지속적으로 관리하여야 한다.

본 연구에서는 기존의 유지보수 비용 산출근거의 부족을 해결하였고 유지보수 시 수주자와 발주자 모두에게 타당한 비용 추정 방안을 제안하였다. 또, 기존문헌을 토대로 유지보수 관리에 효과적으로 사용할 수 있는 측정지표를 제안하였다. 제안한 측정지표는 유지보수 프로젝트 수행 시 프로젝트 도메인, 특성에 따라 필요한 측정지표를 선정하여 측정이 가능하다. 이러한 측정을 바탕으로 유지보수 프로젝트를 관리할 수 있으며 유지보수 비용을 예측할 수 있는 장점이 있다.

하지만 본 연구에서는 컴포넌트 소프트웨어의 유지보수에 범위를 한정하였다. 향후 효과적인 유지보수를 위한 구체적인 측정지표와 비용 예측 방안의 연구가 더욱 진행되어야

할 것이다. 또한, 컴포넌트 소프트웨어 뿐 아니라 구조적 방법론으로 개발된 프로젝트, 객체지향방법론 등으로 개발된 프로젝트의 유지보수 비용추정 방안과 SLA에 기반한 서비스 비용에 관한 비용 또한 연구되어야 할 것이다.

참고 문헌

- [1] 김혜미, 황선명, "컴포넌트 소프트웨어 비용 추정 모델과 규모 예측", 한국정보처리학회 소프트웨어공학 논문지, Vol.6, No.2, pp.3-9, 2003.
- [2] Ian Sommerville, "Software Engineering 5th Ed." Addison-Wesley, 1996.
- [3] 박주석, "RBF망을 이용한 소프트웨어 유지보수 비용 추정", 정보과학회논문지, Vol.31, No.5, pp.555-562, 2004
- [4] Chris Abts, Betsy Clark, "COCOTS Tutorial", 15th Int. COCOMO Forum, Oct., 2000.
- [5] Boehm, B.W. et al., "Software Development Cost Estimation Approach-A Survey", USC Report, 1998.
- [6] George E. Stark, "Measurements for Managing Software Maintenance", 1996 International Conference on Software Maintenance (ICSM '96), 4-8 November 1996, Monterey, CA, USA, Proceedings. IEEE Computer Society 1996.
- [7] Hunt, B.; Turner, B.; McRitchie, K., "Software Maintenance Implications on Cost and Schedule", Aerospace Conference, 2008 IEEE 1-8 March 2008 pp.1-6, Mar., 2008.
- [8] Ian Sommerville, "Software Engineering 5th Ed." Addison-Wesley, 1996.
- [9] Gautam Guliani, Dan Woods, Open Source for the Enterprise, O'Reilly, 2005.
- [10] 전병선, "NET Enterprise System 객체지향 CBD 개발 방법론", 영진닷컴, 2004.
- [11] Darly Mather, The Maintenance Scorecard Creating Strategic Advantage, 2005.
- [12] 한국전산원, "소프트웨어 유지보수 대가기준 모형 연구", 한국전산원, Dec., 2004.
- [13] 최정운, "CBD 방법론을 적용한 신용카드 회원관리 시스템 구축", 숭실대 학위논문, 2005.

조주연



e-mail : achcjy@ssu.ac.kr
 2007년 한국교육개발원 컴퓨터공학사(학사)
 2007년 현재 숭실대학교 컴퓨터학과 석사과정
 관심분야: 소프트웨어 유지보수, 컴포넌트 기반 개발(CBD), 소프트웨어 재공학/역공학, SOA 등



류 성 열

e-mail : syrheo@ssu.ac.kr

1977년 숭실대학교 전자계산학과(학사)

1980년 연세대학교 전자계산학과(석사)

1996년 아주대학교 컴퓨터공학과(박사)

1981년~현 재 숭실대학교 교수

1982년~1995년 숭실대학교 전자계산연구소

및 중앙전자계산소 소장

1997년~1998년 George Mason University 객원 교수

1998년~2001년 숭실대학교 정보과학대학원 원장

2004년~현 재 한국품질재단 운영위원회 위원장

2008년~현 재 정보통신연구진흥원 이사

관심분야: SW공학, SW 요구공학, SW 유지보수, SW 재공학/역공학, 오픈소스