

오픈소스 소프트웨어의 선정 절차에 관한 연구

이 영 민[†] · 류 성 열^{**} · 김 종 배^{***}

요 약

소프트웨어의 품질, 개발 속도 및 비용 등 기존 소프트웨어 개발의 한계점을 해결하기 위한 새로운 대안으로써 오픈소스 소프트웨어를 활용한 개발 방법이 시도되고 있다. 특히, 중소기업들이 소프트웨어를 개발할 경우 대규모 개발 방법론 적용에 어려움을 겪고 있어 소규모 개발 방법에 대한 요구가 증가하고 있는 상황에서, 오픈소스를 활용한 소프트웨어 개발 방식은 비용을 절감하고 개발 기간을 단축시킬 수 있는 효율적인 방법이 될 수 있다. 이에 따라, 오픈소스 소프트웨어에 대한 다양한 분석들이 이루어졌지만 아직까지 산업현장에서 소프트웨어 개발에 오픈소스를 활용하기 위한 구체적인 절차나 방법이 없다.

본 연구에서는 이러한 문제들을 해결하기 위해 기존 연구에 대한 종합과 개발 프로젝트의 경험을 토대로 하여, 오픈소스의 식별과 선정을 위한 효과적인 절차와 방법을 제시하였다. 오픈소스 소프트웨어 선정을 위해서는 개발될 소프트웨어에 대한 정확한 요구사항을 식별하고, 요구사항과 일치하는 오픈소스를 조사하여 후보 목록을 추출하고, 평가기준을 수립하여, 이에 따라 활용 대상 오픈소스 소프트웨어를 선정하는 방법을 제시하였다. 또한 실제 프로젝트의 적용 실험과 평가를 통해 각 활동의 유효성과 개선점을 도출하였다.

키워드 : 오픈소스, 오픈소스 소프트웨어, 선정 절차

A Study on Selection Process of Open Source Software

Lee Young Min[†] · Rhew Sung Yul^{**} · Kim Jong Bae^{***}

ABSTRACT

The development methods which utilize OSSs have been tried as new alternative to solve limits of the previous software developments such as the quality of software, and time and cost of developments. Especially, small and medium companies are suffering from difficulty in applying large-scale development methodology whenever they develop softwares. Therefore, in the current situation that demand for small-scale development methodology is increasing, the methods of utilizing OSSs can become an efficient way to save costs and reduce a development period. Accordingly, analysis of open-source is no fixed procedure or method to utilize open-source for software developments in the field.

In this thesis, to solve such problems, we propose the procedures and methods for identifying and selecting suitable open-source, and effective methods for improvement and integration through least modification on the basis of synthesis of existing researches and experiences in development projects. For selection of OSSs, we did identify correct requirements for the software to be developed, investigate the open-source just matching with such requirements, draw a candidate index, establish assessment criteria, and the consequently present a method to select OSSs. And, we deduced the validity and improvement of each detailed activity from practical application to an actual project and assessment.

Keywords : Open Source, Open Source Software, OSS, Selection Process

1. 서 론

최근 소프트웨어 개발의 문제점인 소프트웨어의 개발 일정 및 비용 등의 문제를 해결하기 위한 대안으로 OSS(OpenSourceSoftware, 오픈소스 소프트웨어)를 활용한 개발

방법이 시도되고 있지만, 소프트웨어 개발에 OSS를 활용하기에는 OSS를 선정하는 절차에 많은 어려움이 있다.

기존의 OSS 활용에 대한 관련연구는 조사 결과 몇몇 연구[1,2,3,4,5,6,7]들이 존재 하였다. 기존 연구는 두 가지 분야로 나뉘지는데, 개발 절차에 관한 연구와 평가 및 선정 기준에 관한 연구이다. 이들 연구를 분석한 결과 절차에 관한 연구는 개념적 활동만을 제시했고, 선정기준에 관한 연구는 기존의 컴포넌트 선정기준을 대부분 그대로 가져온 것에 불과하였다. 본 연구에서는 OSS를 활용하여 소프트웨어를 개발하고자 할 경우, 적절한 OSS를 식별 및 평가, 선정

※ 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌음.
† 준 회 원 : 숭실대학교 컴퓨터학과 박사과정
** 중 심 회 원 : 숭실대학교 컴퓨터학부 교수
*** 정 회 원 : 숭실대학교 컴퓨터학과 공학박사
논문접수: 2008년 8월 25일
수 정 일 : 1차 2008년 10월 17일
심사완료: 2008년 10월 19일

하기 위한 절차와 방법을 제시하고, 실제 프로젝트들에 적용하여 실험함으로써 각 활동의 유효성과 개선점을 도출하고자 한다.

2. 관련 연구

본 장에서는 OSS 선정 방법을 연구하기 위해 관련 문헌들을 중심으로 수행한 연구의 결과를 요약한다.

2.1 T.R. Madanmohan의 오픈소스 선정 모델[2]

T.R. Madanmohan의 오픈소스 컴포넌트 선정 모델은 수집(Collection), 배양(Incubation), 개정(Critical Revision)의 세 단계로 구성된다. 이 연구에서는 활용 가능한 오픈소스 컴포넌트의 식별과 평가를 위한 모델을 제시하고 있지만, 실제 오픈소스를 식별하기 위한 구체적인 절차와 방법의 제시는 없다.

2.2 Meng Huang의 오픈소스 선정 연구[3]

Meng Huang의 연구는 오픈소스 선정과정을 포함하여 전체적인 OSS 기반의 개발 프로세스를 정의하였지만, 선정을 위한 평가과정에서 기능적인 부분을 상대적으로 간과하는 경향이 있고, 그 과정 또한 개발자나 조직의 경험에 의존해야 한다는 약점을 가지고 있기 때문에 구체적인 평가 기준을 제시해야 한다.

2.3 Capagemini의 오픈소스 성숙도 모델[4]

Capagemini의 OSMM(Open Source Maturity Model, 오픈소스 성숙도 모델)은 조직이나 프로젝트에 적합한 오픈소스를 결정하기 위해 개발한 모델이다. OSMM은 오픈소스 기반의 어플리케이션을 개발하는 조직이 보다 효율적으로 평가하고 구현할 수 있는 제품 척도를 통해 조직적인 접근 방법을 제공한다. OSMM은 다른 OSS 컴포넌트를 평가할 경우를 고려해야 할 가장 가치 있는 지표를 강조해주며, 획득 회사의 우선순위 및 비즈니스 모델에 따라 강점과 약점을 비교할 수 있는 방법을 제공[14]해주고 있으나, 구체적인 평가 절차나 방법의 제시는 없다.

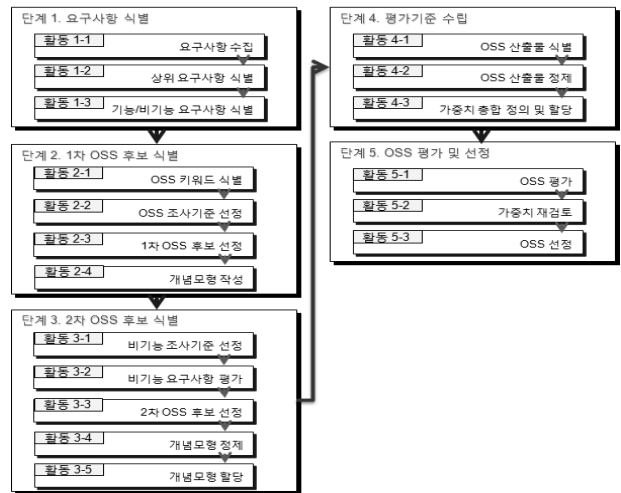
2.4 Kenwood의 이익과 위험에 따른 분류[7]

Kenwood는 OSS에 대한 사례연구에서 OSS의 도입을 위한 분류 체계를 제안하였다. 첫째는 소프트웨어의 생명주기에 근거한 비용요소 분류법이고, 두 번째는 품질적 요소에 따른 분류체계이다.

Kenwood의 연구는 OSS 도입 시 고려해야 할 평가 요소들로 활용될 수 있는 주요한 속성들을 제시하고 있으나, 각 속성들에 대한 평가 비율과 절차는 제시하지 않는다.

3. OSS 선정 절차의 설계

현재 OSS는 많은 커뮤니티 및 사이트에서 제공하고 있다. 그러므로 방대한 양의 OSS로부터 개발에 이용할 오픈



(그림 1) OSS 선정 절차

소스로부터 개발에 이용할 OSS를 선정하기 위한 단계가 우선시 되어야 한다. 이를 위해, 제한한 OSS 선정 절차에서는 요구사항을 분석 및 식별하여, 이를 기반으로 OSS 후보자를 식별하고, 최종 OSS를 선정하기 위해 평가기준 수립 및 OSS 평가 및 선정 등의 단계를 수행한다. OSS 선정 절차에 대한 전체 단계 및 활동은 (그림 1)과 같다.

3.1 OSS의 후보 식별 (단계 1 ~ 단계 3)

단계 1에서 단계 3까지는 개발에 이용할 OSS의 후보를 식별하기 위해 요구사항을 식별하고, 식별된 요구사항을 기반으로 1차/2차 OSS 후보자를 선정하는 단계이다.

3.1.1 단계 1: 요구사항 수집

단계 1은 시스템을 개발하기 위한 사용자 상위수준의 요구사항을 수집하고, 수집된 요구사항을 기반으로 개발될 소프트웨어의 상위 요구사항을 식별하는 단계이다. <표 1>은 단계 1: 요구사항 수집 단계의 주요 활동과 입/출력물을 보여주고 있고, 이 단계는 요구사항 수집 활동과 상위 요구사항 수집 활동, 기능/비기능 식별 활동으로 구성된다. 식별된 요구사항은 OSS 식별단계의 입력 데이터로 활용된다.

활동 1-1: 요구사항 수집은 비전 기술서 등을 통해 시스템의 개발 배경을 이해하고, 실제 시스템 사용자와의 면담, 설문 등을 통해 요구사항을 파악하는 작업이다. <표 2>는 사용자 요구 수집서의 예시이다.

활동 1-2: 상위요구사항 식별 활동에서는 요구사항 수집 작업을 통해 파악된 추상적인 수준의 기능 및 비기능 요구사항을 기반으로 상위 요구사항에 대한 명세를 작성한다. <표 3>은 상위 수준 요구사항의 예시이다.

활동 1-3: 기능/비기능 요구사항 식별에서는 상위 요구사항 식별 활동을 통해 파악된 상위 수준의 요구사항을 기능 및 비기능 요구사항으로 구분한다. <표 4>는 기능/비기능 요구사항 명세서의 예를 보여주고 있다.

3.1.2 단계 2: 1차 OSS 후보 식별

단계 2는 사용자의 직접적인 요구사항을 기반으로 OSS

<표 1> 단계 1: 요구사항 식별

| 활동 | 입력물 | 출력물 | 설명 |
|----------------|--------------------------|-----------------|------------------------|
| 요구사항 수집 | 비전 기술서 | 사용자 요구 수집서 | 요구사항 분석을 위해 필요한 자료를 수집 |
| 상위 요구사항 식별 | 비전기술서 사용자 요구 수집서 | 상위 요구사항 명세서 | 상위의 요구사항을 식별 |
| 기능/비기능 요구사항 식별 | 사용자 요구 수집서 상위요구사항 명세서 | 기능/비기능 요구사항 명세서 | 기능 및 비기능 요구사항을 식별 |

<표 2> 활동 1-1: 사용자 요구 수집서의 예

| 항목 | 설명 |
|---------|---|
| 사용자 정의서 | 시스템 사용자들의 목록으로 사용자 그룹 또는 개인명과 각 사용자에 대한 간략한 설명을 기록한다. |
| 면담 계획서 | 사용자의 요구사항을 파악하기 위해서 사용자와의 면담이 필요한 경우, 면담일정계획에 대한 내용들을 기술한다. |
| ... | ... |

<표 3> 활동 1-2: 상위 수준 요구사항의 예

| |
|--|
| R1. 시스템은 별도의 프로그램 설치 없이 웹 브라우저를 통해 사용할 수 있어야 하며 내·외부망의 구분 없이 접근할 수 있어야 한다. |
| R2. 시스템에 대한 보안성을 보장하여야 한다. (생략)... |

<표 4> 활동 1-3: 기능/비기능 요구사항 명세서의 예

| 번호 | 구분 | 상위 요구사항 |
|-----|-----|--|
| R1 | 비기능 | 시스템은 별도의 프로그램 설치 없이 웹 브라우저를 통해 사용가능하고, 내외부망의 구분 없이 접근 가능해야 한다. |
| R2 | 비기능 | 시스템에 대한 보안성을 보장하여야 한다. |
| ... | ... | ... |

<표 5> 단계 2 : 1차 OSS 후보 식별

| 활동 | 입력물 | 출력물 | 설명 |
|--------------|-------------------|--------------------------------|----------------------------------|
| OSS 키워드 식별 | 상위 요구사항 명세서 | OSS 1차 조사서 (키워드 테이블) | 키워드를 식별 |
| OSS 조사 기준 선정 | 키워드 테이블 | OSS 1차 조사서 (OSS 커뮤니티 정보 조사기준표) | OSS 커뮤니티의 정보를 기반으로 조사 기준을 수립 |
| 1차 OSS 후보 선정 | OSS 커뮤니티 정보 조사기준표 | OSS 1차 조사서 (1차 OSS 후보 리스트) | OSS 검색 후 1차 후보를 선정 및 기초자료 조사를 수행 |
| 개념모형 작성 | 비전기술서 | OSS 1차 조사서 (개념모형도) | 개발될 시스템의 모형을 구성 |

를 조사하기 위한 키워드를 식별하고, 조사기준을 선정하여, 이에 따라 OSS 후보 선정과 개념모형의 작성을 수행하는 단계이다. <표 5>는 단계 2: 1차 OSS 후보 식별 단계의 주요 활동과 입/출력물을 정의하고 있다.

활동 2-1 : OSS 키워드 식별 작업은 상위 요구사항 명세서를 기반으로 OSS 조사를 위한 키워드를 식별한다. 이러한 키워드는 OSS를 제공하는 커뮤니티와 사이트에서 1차적인 검색을 위해 사용된다. <표 6>은 키워드 테이블의 구성을 설명하고 있다.

활동 2-2 : OSS 조사기준 선정은 OSS 조사를 위한 일정한 제한 및 기준을 세운다. OSS에 대한 조사기준이 없다면 키워드로 검색되는 방대한 양의 OSS를 전부 조사해야 함으

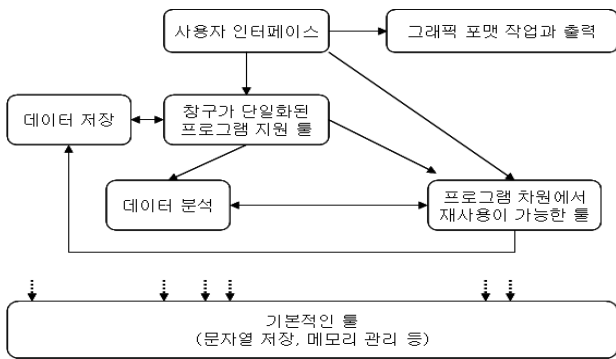
로 많은 시간이 소요 될 수 있다. <표 7>은 OSS 조사기준의 예를 보여주고 있는데, 이러한 기준을 OSS 검색을 위한

<표 6> 활동 2-1: 키워드 테이블

| 키워드명 | 설명 |
|--------|----------------------|
| 검색 키워드 | 키워드가 포함하는 기능의 간략한 설명 |

<표 7> 활동 2-2: OSS 조사기준표의 예

| 번호 | 항목 | 설명 | 비고 |
|-----|----------------|----------------------|------|
| 1 | Activity | 50% 이하는 1차 조사과정에서 배제 | 제외조건 |
| 2 | Project Admins | 프로젝트 관리자(권위) | 참조자료 |
| ... | ... | ... | ... |



(그림 2) 활동 2-3: 개념모형도의 예

1차적 선별 조건으로 활용할 수 있다.

활동 2-3 : 1차 OSS 후보 선정은 키워드 테이블과 OSS 조사 기준표를 기반으로 1차적으로 OSS를 검색하고, 기초 자료를 조사한다. 검색된 OSS 기반으로 1차 후보자 리스트를 작성하고, 조사한 커뮤니티 자료를 기반으로 기초자료를 작성한다.

활동 2-4 : 개념모형작성에서는 상위 요구사항을 기반으로 식별된 OSS 후보 리스트를 이용하여 개념모형도를 작성한다. 개념모형도의 예는 (그림 2)와 같다.

3.1.3 단계 3: 2차 OSS 후보 식별

단계 3은 비기능 요구사항을 기반으로 2차 OSS 후보자를 식별하고, 이를 기반으로 수집되고 정의된 시스템의 요구사항을 상세하게 정제하고 평가를 준비하는 활동이다. 비기능 조사기준 선정, 비기능 평가, 2차 OSS 후보 선정, 개념모형 정제, 개념모형 할당 단계로 정제와 평가준비가 이루어진다. <표 8>은 단계 3 : 2차 OSS 후보 식별 단계의 주요 활동과 입/출력물을 정의하고 있다.

활동 3-1 : 비기능 조사기준 선정 활동은 비기능 요구사

<표 8> 단계 3 : 2차 OSS 후보 식별

| 활동 | 입력물 | 출력물 | 설명 |
|----------------|---|-----------------------------|---|
| • 비기능 조사 기준 선정 | • 사용자 요구 수집서 • 상위 요구사항 명세서 • 기능/비기능 명세서 | • OSS 2차 조사서 (비기능 조사 기준표) | 비기능 요구사항 명세서를 기반으로 비기능 조사 기준을 수립 |
| • 비기능 요구 사항 평가 | • 비기능 조사기준표 | • OSS 2차 조사서 (비기능 평가 테이블) | 비기능 요구사항을 평가 |
| • 2차 OSS 후보 선정 | • OSS 1차 조사서 • 비기능 평가 테이블 | • OSS 2차 조사서 (2차 OSS 후보리스트) | OSS의 비기능 요구사항을 평가하여 2차 후보를 선정 |
| • 개념모형 정제 | • 개념모형도 | • OSS 2차 조사서 (개념모형도) | 개념모형을 정제 |
| • 개념모형 할당 | • OSS 2차 조사서 | • OSS 후보 분류서 | OSS의 평가 시 개념모형별로 평가하기 위해 개념모형에 OSS 후보를 할당 |

<표 9> 활동 3-1 : 비기능 조사기준 선정표의 예

| 번호 | 조사항목 | 내용 | 제외조건 | 비고 |
|-----|--------|----------------|----------------------|------|
| 1 | 설치 용이성 | 프로그램 설치가 용이한가 | 설치 프로그램을 제공하지 않으면 배제 | 제외조건 |
| 2 | 사용 용이성 | 인터페이스 사용이 용이한가 | | 참조자료 |
| ... | ... | ... | ... | ... |

<표 10> 활동 3-2: 비기능 요구사항 평가 테이블

| 번호 | 이름 | 설치용이성 | 보안성 | 라인선스 |
|-----|----------------|-------|-----|------|
| 1 | Work project | X | O | O |
| 2 | Project Logger | X | O | X |
| ... | ... | ... | ... | ... |

<표 11> OSS 할당 테이블의 예

| 개념모형 단위 | OSS 후보 리스트 |
|---------------------|---------------------------------|
| 프로젝트(Project) 관리 모듈 | Work project |
| | Metis Project Management Module |
| | dotProject |
| | ... |
| 일정(Schedule) 관리 모듈 | DesignWorks Project Site |
| | ACal Project |
| | dotProject |
| | ... |

항 명세서를 기반으로 1차 OSS 후보자들의 비기능 요구사항을 조사하는 기준을 선정한다. <표 9>는 비기능 조사기준 선정표의 예이다.

활동 3-2 : 비기능 요구사항 평가 활동에서는 비기능 조사기준 선정표를 기반으로 1차 OSS 후보자들의 비기능 요구사항을 평가한다. <표 10>은 비기능 요구사항 평가 테이블의 예이다.

활동 3-3 : 2차 OSS 후보 선정 활동은 비기능 요구사항 평가 테이블을 기반으로 사용자 요구사항에 적합한 OSS 후보를 2차적으로 선정하는 활동이다. 비기능 요구사항을 반영하여 OSS를 식별하여 좀 더 명확한 OSS 후보자를 선정할 수 있다.

활동 3-4, 3-5 : 개념모형 정제 및 할당 활동은 3-1에서

〈표 12〉 단계 4 : 평가기준 수립

| 활동 | 입력물 | 출력물 | 설명 |
|------------------|----------------------------|------------------|----------------------------|
| • OSS 산출물 식별 | • 기능/비기능 명세서 • OSS 산출물 | • OSS 산출물 식별서 | OSS 후보자들의 산출물을 식별 |
| • OSS 산출물 정제 | • OSS 산출물 식별서 | • OSS 산출물 정제서 | OSS 후보자들의 산출물을 정제 |
| • 가중치 총합 정의 및 할당 | • 요구사항 수집서 • OSS 후보 분류서 | • 평가 기준서(가중치테이블) | 가중치를 정의하고, 가중치를 후보 OSS에 할당 |

3-3 활동까지 분석된 비기능 요구사항 및 OSS 후보를 개념 모형에 정제 및 할당하는 활동이다. <표 11>은 OSS 할당 테이블의 예이다.

3.2 OSS 선정 (단계 4 ~ 단계 5)

단계 4에서 단계 5까지의 OSS 선정은 단계 3에서 선정된 OSS 후보군으로부터 요구사항에 적합한 OSS를 선정하기 위한 평가기준을 수립하고, 수립된 기준에 따라 OSS 후보를 평가하여, 최종 OSS를 선정하는 과정이다.

3.2.1 단계 4: 평가기준 수립

단계 4는 OSS를 평가하기 위해 OSS 후보자들의 산출물을 식별하고 정제한다. 그리고 각 모듈의 세부기능에 가중치를 결정하여, OSS 후보에 맵핑하여 평가를 하는 단계이다. <표 12>는 평가기준 수립 단계의 주요 활동과 입/출력물을 정의하고 있다.

활동 4-1, 4-2 : OSS 산출물 식별 활동에서는 선정된 OSS 후보자가 제공하는 산출물을 식별한다. OSS 산출물 OSS 마다 제공하는 정보와 양식이 모두 다르기 때문에 평가 기준의 정보로 활용하기 위해서는 OSS 산출물의 정제 과정이 필요하다. OSS 산출물 정제 활동에서는 식별된 OSS 산출물을 정제하여 OSS를 평가하기 위한 정보를 얻는 활동이다. <표 13>은 OSS 산출물을 조사하여 TTA 소프트웨어 표준 문서화 지침서를 기반으로 OSS 산출물의 분류표를 작성하고, OSS 마다 각기 다른 산출물 정보의 분류 기준을 제시하였다[9][10][11].

활동 4-3 : 가중치 총합 정의 및 할당 활동에서는 정의된 속성을 기반으로 각 속성들에 대한 가중치를 정의하여 가중치 정의 테이블을 작성하고, 이를 기반으로 하여 가중치를 개념모듈에 할당한다. 평가기준서는 가중치 정의 테이블과 가중치 할당 테이블로 구성된다. <표 14>는 각 속성 별 가중치 할당 테이블의 예시이고, <표 15>는 기능성에 대한 가중치 할당 테이블의 예이다.

3.2.2 단계 5: OSS 평가 및 선정

단계 5는 평가기준서에 따라 OSS 후보자를 평가하여 프로젝트에서 사용하고자 하는 OSS를 선정하는 것이다. 또한 이 과정에서 변경된 사항에 대해 사용자와의 리뷰를 통한 가중치 재검토 작업을 수행한다. <표 16>은 단계 5 : OSS 평가 및 선정 단계의 주요 활동과 입/출력물을 정의하고 있다.

활동 5-1 : OSS 평가 활동에서는 평가 기준서에 따라

〈표 13〉 OSS 산출물 분류표

| 분류 | 정보 항목 |
|--------|-----------|
| 릴리즈 정보 | 라인선스 정보 |
| | 버전변경 정보 |
| | 기능 정보 |
| 모듈 정보 | 클래스 정보 |
| | 모듈 정보 |
| | 변수 정보 |
| | 파일 정보 |
| 설치 정보 | 시스템 요구사항 |
| | 설치 정보 |
| 사용 정보 | 기본 정보 |
| | 환경설정 정보 |
| | 인터페이스 정보 |
| | 공통모듈 정보 |
| | 데이터 정보 |
| | 기능별 세부 정보 |

〈표 14〉 속성에 대한 가중치 할당 테이블의 예

| 구분 | 속성 | 가중치 |
|-----|-----|-----|
| 1 | 기능성 | 200 |
| 2 | 정확성 | 150 |
| ... | ... | ... |

〈표 15〉 기능성에 대한 가중치 할당 테이블의 예

| 번호 | 요구사항 | 가중치 |
|----|----------------------------------|-----|
| 1 | 프로젝트 Life Cycle(단계, 활동, 활동, 산출물) | 60 |
| 2 | 노트(메모) | 15 |
| 3 | 북마크(관련 사이트) | 15 |
| 4 | 버그트래킹 | 15 |
| 5 | help requests(가이드) | 15 |
| 6 | 문서(파일)관리기능 | 40 |
| 7 | 프로젝트 공정표 | 40 |
| 소계 | | 200 |

〈표 16〉 단계 5 : OSS 평가 및 선정

| 활동 | 입력물 | 출력물 |
|-----------|-------------------------|--------------|
| • OSS 평가 | • OSS 후보 분류서 • 평가기준서 | • OSS 평가결과서 |
| • 가중치 재검토 | • OSS 평가 결과서 | • 평가기준서 (정제) |
| • OSS 선정 | • OSS 평가결과서 | • OSS 선정 결과서 |

〈표 17〉 평가 결과서(기능)의 예

| 번호 | 요구사항 | Work project | dot Project | eGroup Ware | php Collab | 가중치 |
|-----|----------------------------------|--------------|-------------|-------------|------------|-----|
| 1 | 프로젝트 Life Cycle(단계, 활동, 작업, 산출물) | 0.6 | 0.8 | 0.7 | 0.7 | 60 |
| 2 | 노트(메모) | 0.5 | 0.7 | 0.3 | 0.4 | 15 |
| ... | ... | ... | ... | ... | ... | ... |
| | 소계 | 95 | 149.5 | 120 | 138 | 200 |

OSS를 평가한다. <표 17>은 기능에 대한 평가 결과서의 예이다. 평가 결과서(기능)의 예는 기능 요구사항에 대한 평가의 결과를 보여주고 있는데, 앞의 평가기준 수립 단계에서 기능별로 할당된 가중치와 각 요구 기능과 후보 OSS의 기능을 비교하여 배점한 결과를 곱하여 후보별 점수를 합산하는 방식을 사용하고 있다. 이러한 평가결과서를 바탕으로 활용 대상 OSS를 선정할 수 있는데, 이때 OSS 선정 결과서에는 평가의 결과뿐만 아니라, 그 과정과 근거 등의 기록도 함께 유지해야 한다. 이는 향후 변경 설계와 통합의 과정에서 재검토와 조정에 필요할 수 있기 때문이다.

활동 5-2 : 가중치 재검토 활동에서는 변경된 사항이 발생 시 사용자와의 리뷰를 통한 가중치 재검토 활동을 수행한다. 필요시에는 가중치의 재적용을 통해 평가를 다시 수행하여 OSS 선정에 따르는 위험을 최소화한다.

활동 5-3 : OSS 선정 활동에서는 OSS 평가 결과서에 따라 최종적으로 사용할 OSS를 선정한다.

4. 적용 실험 및 분석

본 연구에서 제시하는 OSS 선정절차의 효용성을 검증하기 위해 산업 현장에서 활용 가능한 OSS 선정 사례 연구의 내용을 요약된 과정 및 결과로 제시한다.

4.1 프로젝트 관리 시스템을 위한 OSS 선정 사례연구

개발 방법론과 프로젝트 관리 기능을 지원하는 기존의 상용제품들은 대부분 고가일 뿐 아니라, 대규모 개발 방법론을 기반으로 하고 있어 중소기업에 적용하기에는 무리가 있다. 인력과 기술이 취약한 중소기업에서 쉽게 적용할 수 있고, 프로젝트의 규모에 따라 그 절차를 조정할 수 있는 유연한 지원도구의 필요성이 제기되었다.

웹 기반의 프로젝트 관리 시스템은 중소기업 소프트웨어 개발회사들이 가진 문제를 해결하기 위한 시스템으로서, 웹을 통해서 프로젝트의 전 과정에 대한 관리감독이 가능하고, 방법론에 대한 가이드를 제공하며, 작은 규모의 프로젝트에도 쉽게 빠르게 적용할 수 있는 지원도구이다. 이러한 프로젝트 관리 시스템을 선정하기 위하여 3장에서 제시한 OSS 선정절차를 기반으로 OSS 커뮤니티에서 웹 기반의 프로젝트 관리 시스템을 선정하는 사례 연구를 진행하였고, 마르미 방법론[8]에서 제시하고 있는 산출물 양식을 활용하여, 각 활동의 산출물을 제시한다.

4.1.1 상위 요구사항 분석 및 기능/비기능 요구사항 식별
요구사항 식별 단계에서는 제사용할 OSS의 조사를 위해

사용자를 통해 <표 18> 시스템 요구사항을 조사하고, 조사된 추상적인 요구사항을 기반으로 <표 19> 상위 요구사항에 대한 명세를 작성하였다. 그리고 <표 20> 목표 시스템의 기능/비기능 요구사항을 식별하였다.

4.1.2 키워드 및 조사기준을 통한 1차 OSS 후보 식별

시스템에 적합한 OSS를 체계적으로 검색하기 위해 요구사항 명세서 및 기능 요구사항의 핵심 키워드를 기반으로 프로젝트 관리, 방법론, 프로세스, 일정 관리 등의 <표 21>와 같은 키워드를 선정하고, platform(LINUX)과 Language(PHP)와 같은 제한사항을 고려하였다. 이를 바탕으로 OSS 커뮤니티 사이트인 소스포지(<http://sourceforge.net>)[12]를 통

〈표 18〉 시스템 요구사항

| 번호 | 요구사항 | 설명 |
|-----|------------|-------------------|
| 1 | 사용자 인터페이스 | 웹 브라우저를 통한 접근 제공 |
| 2 | 보안성 | 권한을 통한 접근 |
| 3 | 방법론/프로세스지원 | 단계개발 프로세스와 방법론 지원 |
| 4 | 일정 관리 | 프로젝트의 일정을 관리 |
| ... | ... | ... |

〈표 19〉 상위 요구사항 명세서

| | |
|-----|---|
| R1. | 시스템은 별도의 프로그램 설치 없이 웹 브라우저를 통해 사용가능 하고, 내외부망의 구분 없이 접근 가능해야 한다. |
| R2. | 시스템에 대한 보안성을 보장하여야 한다. |
| R3. | 시스템은 마르미III(V4.0) 방법론 기반의 단계개발프로세스에서 제시하고 있는 모든 단계별 절차와 활동, 그리고 직·간접적인 산출물을 관리(추가, 수정, 삭제, 조회)할 수 있어야 한다. |
| R4. | 시스템은 프로젝트의 일정 계획, 진도, 담당자를 관리(추가, 수정, 삭제, 조회)할 수 있어야 한다. (생략)... |

〈표 20〉 목표시스템 기능/비기능 요구사항 식별

| 번호 | 구분 | 초기 기능 요구사항 |
|-----|--------------------|---|
| R1 | 비기능 | 시스템은 웹 브라우저를 통해 사용 가능하고, 내 외부망의 구분 없이 접근 가능해야 한다. |
| R2 | 비기능 | 시스템에 대한 보안성을 보장하여야 한다. |
| R3 | 단계별 절차와 활동, 산출물 관리 | 시스템은 마르미III 방법론 기반의 단계개발프로세스의 모든 단계별 절차와 활동, 산출물을 관리(추가, 수정, 삭제, 조회)할 수 있어야 한다. |
| R4 | 일정 계획, 진도, 담당자 관리 | 시스템은 프로젝트의 일정 계획, 진도, 담당자를 관리(추가, 수정, 삭제, 조회)할 수 있어야 한다. |
| ... | ... | ... |

〈표 21〉 프로젝트 관리 도구의 키워드 테이블

| 키워드명 | 설명 | 비고 |
|---------------------|---------|------------|
| Project Management | 프로젝트 관리 | 목표시스템의 키워드 |
| Methods | 방법론 | R3에서 도출 |
| Process | 프로세스 | R3에서 도출 |
| Schedule Management | 일정 관리 | R4에서 도출 |
| ... | ... | ... |

〈표 22〉 OSS 조사기준표

| 번호 | 항목 | 설명 | 비고 |
|-----|----------------|----------------------|------|
| 1 | Activity | 50% 이하는 1차 조사과정에서 배제 | 제외조건 |
| 2 | Project Admins | 프로젝트 관리자(권위) | 참조자료 |
| ... | ... | ... | ... |
| 18 | Site Rank | 랭크순위 | 참조자료 |

〈표 23〉 1차 PM OSS 후보 리스트

| 번호 | 이름 | 활동율 |
|-----|-------------------------------------|---------------------------|
| 1 | Work project | All time activity: 87.25% |
| 2 | Project Logger | All time activity: 86.84% |
| ... | ... | ... |
| 13 | PrjPlanner-Agile Project Management | All time activity: 97.08% |

해 OSS를 조사하였다. 1차 조사에서는 분석 활동의 효율성을 높이기 위해 우선 키워드를 통해 검색된 목록에서 함께 제공되는 간략 정보(About)를 통해 연관성이 없는 프로젝트를 제외하는 것이 효율적이다. 이를 위해 <표 22>와 같이 조사기준표를 작성하였다. 그리고 이러한 1차 조사 활동의 결과 <표 23>과 같은 1차 OSS 후보 리스트를 작성하였다.

4.1.3 비기능 요구사항을 통한 2차 OSS 후보 식별

비기능 요구사항을 기반으로 라인선스(Proprietary), 보안성(접근제어) 등의 <표 24>의 비기능 조사기준을 선정하고, 이를 기반으로 <표 25>과 같이 1차 OSS 후보자들을 평가한다. 그리고 평가 결과를 기반으로 <표 26>와 같이 2차 OSS 후보자를 선정하였다.

<표 27>은 OSS식별 활동을 통해 추가로 발견된 기능 요구사항을 초기요구사항과의 조정을 통해 정제하였다.

〈표 24〉 비기능 조사기준 선정표

| 번호 | 조사항목 | 내용 | 제외조건 | 비고 |
|----|--------|---------------------|------------------------------|------|
| 1 | 설치 용이성 | 프로그램 설치가 용이한가 | 설치 프로그램을 제공하지 않으면 배제 | 제외조건 |
| 2 | 사용 용이성 | 인터페이스 사용이 용이한가 | | 참조자료 |
| 3 | 접근성 | 사용자가 시스템에 접근하기 용이한가 | | 참조자료 |
| 4 | 보안성 | 보안에 대한 고려가 되었는가 | 권한을 통한 접근제어가 없는 경우는 배제 | 제외조건 |
| 5 | 라인선스 | 다양한 라인선스를 지원하는가 | Proprietary License(독점) 는 배제 | 제외조건 |

〈표 25〉 PM OSS 후보의 비기능 요구사항 평가

| 번호 | 이름 | 설치용이성 | 보안성 | 라인선스 |
|-----|-------------------------------------|-------|-----|------|
| 1 | Work project | X | O | O |
| 2 | Project Logger | X | O | X |
| 3 | DesignWorks Project Site | X | X | O |
| ... | ... | ... | ... | ... |
| 12 | phpaga | O | O | O |
| 13 | PrjPlanner-Agile Project Management | O | O | O |

〈표 26〉 2차 PM OSS 후보 리스트

| 번호 | 이름 |
|----|---------------------------------------|
| 1 | airTODO - Project Management Tool |
| 2 | ACal Project |
| 3 | dotProject |
| 4 | eGroupWare: Enterprise Collaboration |
| 5 | phpCollab |
| 6 | more.groupware |
| 7 | Group-Office groupware |
| 8 | phpaga |
| 9 | PrjPlanner - Agile Project Management |

〈표 27〉 목표시스템 기능(정제된 기능 리스트)

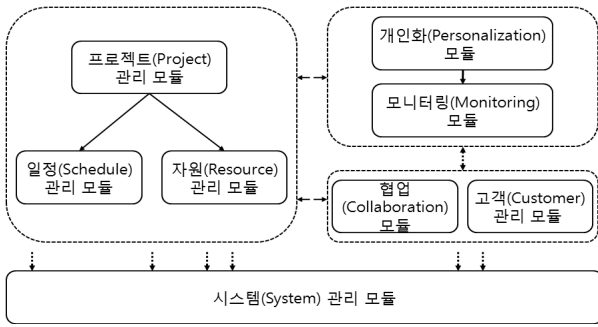
| 정제된 요구 기능 리스트 |
|---|
| <ul style="list-style-type: none"> • 프로젝트 Life Cycle-단계, 활동, 활동, 산출물, 파일, 노트(메모), 북마크 • 프로젝트 일정-계획, 진도, ToDo리스트 (생략)... |

이와 같이 정제된 요구 기능은 목표 시스템의 개념모형으로 설계되었는데, (그림 3)과 같은 목표시스템의 개념모형 설계에서도 조사 과정에서 획득한 후보 OSS들의 아키텍처를 반영될 수 있었다.

다음으로 <표 28>과 같이 개념모형에서 설계된 모듈별로 OSS 후보자를 할당하였다.

4.1.4 OSS 선정을 위한 평가기준 수립

평가기준 수립 단계는 OSS 후보자들의 산출물을 기반으로 OSS를 평가하기 위한 정보를 정제한다. 그리고 OSS를



(그림 3) 프로젝트 관리시스템의 개념모형도

<표 28> 모듈별 평가 후보 목록

| 개념모형 단위 | OSS 후보자 리스트 |
|---------------------|-----------------------------------|
| 프로젝트(Project) 관리 모듈 | airTODO - Project Management Tool |
| | dotProject |
| | ... |
| 일정(Schedule) 관리 모듈 | ACal Project |
| | dotProject |
| | ... |
| ... | ... |

<표 29> OSS 산출물 분류표

| 분류 | 정보 항목 |
|--------|----------|
| 릴리즈 정보 | 라인선스 정보 |
| | 버전변경 정보 |
| | 기능 정보 |
| 모듈 정보 | 클래스 정보 |
| | 모듈 정보 |
| | 변수 정보 |
| | 파일 정보 |
| 설치 정보 | 시스템 요구사항 |
| | 설치 정보 |

평가하기 위한 각 속성의 가중치를 결정한 후, 이러한 가중치를 OSS 후보에 할당하는 활동으로 구성된다.

<표 29> OSS 산출물 분류표를 기반으로, <표 30>과 같이 OSS 후보자들의 산출물을 정제하였다.

8개의 모듈의 세부 기능을 평가하기 위해 가중치 부여 기법을 사용했다. 점수는 개발조직의 경험과 업무의 중요성에 근거하여 표준점수를 부여했으며, 각 OSS 후보들의 점수와 표준점수가 비교된다. <표 31>는 8개 모듈 중 프로젝트 관리 모듈 항목에 표준점수를 부여한 것이다.

4.1.5 OSS 평가 및 선정

OSS 평가 단계는 이전의 활동에서 수집된 정보를 바탕으로 비판적 검토를 통해 후보들을 탈락시켜 최종의 후보만을 남긴다. 이를 보완한 OSS 평가 단계에서는 기능평가 점

<표 30> PM OSS 후보의 산출물 리스트

| 번호 | 이름 | 산출물 |
|----|--------------------------------------|-----------------------------|
| 1 | airTODO - Project Management Tool | 릴리즈 정보, 모듈 정보, 설치 정보, 사용 정보 |
| 2 | ACal Project | 릴리즈 정보, 사용 정보 |
| 3 | dotProject | 릴리즈 정보, 모듈 정보, 설치 정보, 사용 정보 |
| 4 | eGroupWare: Enterprise Collaboration | 릴리즈 정보, 모듈 정보, 사용 정보 |
| 5 | phpCollab | 릴리즈 정보 |
| 6 | more.groupware | 릴리즈 정보, 사용 정보 |
| 7 | Group-Office groupware | 릴리즈 정보, 사용 정보 |
| 8 | phpaga | 릴리즈 정보 |
| 9 | PrjPlanner-Agile Project Management | 릴리즈 정보, 사용 정보 |

<표 31> 기능 평가 기준

| 모듈 | 번호 | 요구사항 | 점수 |
|----------------------|----|---------------------------------|-----|
| 프로젝트 (Project) 관리 모듈 | 1 | 프로젝트 LifeCycle(단계, 활동, 작업, 산출물) | 60 |
| | 2 | 노트(메모) | 15 |
| | 3 | 북마크(관련 사이트) | 15 |
| | 4 | 버그트래킹 | 15 |
| | 5 | help requests(가이드) | 15 |
| | 6 | 문서(파일)관리기능 | 40 |
| | 7 | 프로젝트 공정표 | 40 |
| 소계 | | | 200 |

<표 32> 평가 결과서(기능)

| 번호 | 요구사항 | air TODO | Acal Project | dot Project | eGroup Ware | ... | 가중치 |
|----|---------------------------------|----------|--------------|-------------|-------------|-----|-----|
| 1 | 프로젝트 LifeCycle (단계, 활동, 작업 산출물) | 0.8 | 0.5 | 0.8 | 0.7 | ... | 60 |
| 2 | 노트(메모) | 0.7 | 0.4 | 0.7 | 0.3 | ... | 15 |
| 3 | 북마크(관련 사이트) | 0.7 | 0.5 | 0.6 | 0.4 | ... | 15 |
| 4 | 버그트래킹 | 0.7 | 0.5 | 0.6 | 0.5 | ... | 15 |
| 5 | help requests(가이드) | 0.5 | 0.6 | 0.6 | 0.8 | ... | 15 |
| 6 | 문서(파일)관리기능 | 0.8 | 0.3 | 0.8 | 0.6 | ... | 40 |
| 7 | 프로젝트 공정표 | 0.8 | 0.8 | 0.8 | 0.6 | ... | 40 |
| 소계 | | 151 | 104 | 149.5 | 120 | ... | 200 |

수와 기능 이외의 속성이 얻는 평가점수를 가지고 평가를 한다. <표 32>는 OSS 후보의 기능 및 속성을 평가한 평가 결과서이다.

각 OSS 후보의 기능 평가와 기능 이외의 속성 평가가 이루어진 후에 최종적으로 점수와 등급을 합산하여 사용자의 요구사항을 가장 만족하는 OSS를 선정했다. OSS 후보 중에 dotProeject, airTODO - Project Management Tool,

〈표 33〉 OSS 최종 선정

| 순위 | 이름 | 점수 |
|----|-----------------------------------|------|
| 1 | airTODO - Project Management Tool | 1468 |
| 2 | dotProject | 1464 |
| 3 | more.groupware | 1418 |

more.groupware가 가장 상위 점수를 획득하였고, 그중에서도 airTODO - Project Management Tool이 가장 높았기 때문에, 최종 OSS로 선정했다. <표 33>

4.2 종합 분석

4장에서는 산업 현장에서 활용 가능한 실용적인 OSS 선정 절차를 정립하기 위하여 OSS를 활용하기 위한 선정 방법을 제시하고, 이를 실제 프로젝트에 적용하여 각각의 유효성과 한계를 도출하였다. 본 절에서는 연구의 성과를 보다 명확히 하기 위해, 이론적 측면에서의 평가를 추가적으로 수행한 뒤 그 결과들을 종합하였다.

첫째, 본 연구에서 제시한 OSS의 선정 방법과 절차가 적절한 품질의 OSS를 선정하기 위한 객관적인 기준을 제시하고 있는지를 평가하였다. 이를 위해, 소프트웨어의 품질을 평가하기 위한 표준으로 제시되고 있는 ISO 9126 [13]의 품질 특성과의 연관성을 비교 측정하였다.

<표 34>의 비교 평가의 결과에서 알 수 있듯이 본 연구에서 제안한 OSS 평가 기준으로서의 품질 속성은 기존의 연구들에서 제안한 속성들에 비해 ISO 9126에서 권고하고 있는 기준에 보다 가까우며, 그 만큼 OSS 선정 기준의 객관성을 확보하고 있다고 평가 할 수 있다.

둘째, 역시 본 연구에서 제시한 OSS의 선정 방법과 절차가 적절한 품질의 OSS를 선정하기 위한 효율적인 절차를 제시하고 있는지를 평가하였다. 이를 위해 소프트웨어의 선정과 획득을 위한 표준 프로세스 등과의 비교 평가를 수행하는 것도 의미가 있으나, 상용 소프트웨어의 선정 절차와 OSS의 선정 절차는 서로 다를 수 밖에 없기 때문에[14], 본 연구에서는 기존의 OSS 선정 절차 등에 대한 연구의 결과들과 비교하는 것으로 평가를 대신한다.

<표 35>의 비교를 통해, 본 연구에서 제안한 절차와 방법은 기존의 연구들에 비해 절차가 세분화 되어 있으며, 특히 평가속성을 제외기준과 평가기준으로 구분하여 초기 조사의 과정에서 분석 대상을 크게 줄일 수 있으며, 조사 과정에서의 기능 분석을 통해 요구사항을 상세화하여 후보 선정의 정확성을 기할 수 있고, 상향식 방법을 통한 가중치 배분으로 하위 품질 속성들 간의 형평성을 유지할 수 있다는 점에서 보다 객관적이고 효율적인 절차라고 평가할 수 있다.

지금까지의 적용 실험을 통한 평가와 이론적 평가 결과를 통해 위와 같은 성과를 도출하였지만, 다음과 같은 한계를 발견할 수 있었다.

첫째, 실험을 통해 기존 OSS에 근거하여 시스템을 구축할 때의 핵심적인 절차는 적절한 OSS의 선정과 이를 위한 평가 기준의 수립임을 알 수 있었는데, 이러한 평가기준은 프로젝

〈표 34〉 ISO 9126과의 연관성

| ISO9126 품질특성 | ISO 9126 품질 부특성 | OSMM | Meng | Kenwood | 제안속성 |
|--------------|-----------------|------|------|---------|------|
| 기능성 | 적합성 | x | o | x | o |
| | 정확성 | x | o | o | x |
| | 상호운용성 | o | o | o | o |
| 신뢰성 | 보안성 | x | o | o | o |
| | 성숙성 | o | o | o | o |
| | 결함허용성 | x | x | o | x |
| 사용성 | 회복성 | x | x | x | o |
| | 이해성 | x | o | x | o |
| | 학습성 | o | o | x | o |
| | 운용성 | x | x | x | x |
| 효율성 | 선호도 | o | x | o | o |
| | 시간반응성 | x | o | o | o |
| 유지보수성 | 자원효율성 | x | x | x | x |
| | 분석성 | x | x | x | o |
| | 변경성 | o | o | o | o |
| | 안정성 | x | o | o | o |
| 이식성 | 시험성 | x | x | x | x |
| | 적용성 | o | o | x | o |
| | 설치성 | o | o | x | o |
| | 공존성 | o | x | x | o |
| 지원정도 | 대체성 | x | x | x | x |
| | | 38% | 57% | 43% | 71% |

〈표 35〉 OSS 선정 절차 비교

| 구분 | Rahul | Meng | 제안 절차 |
|-------|---|---|--|
| 핵심 절차 | <ul style="list-style-type: none"> 수집(Collection) 배양(Incubation) 개정(Critical Revision) | <ul style="list-style-type: none"> 고수준요구사항 고수준아키텍처 평가기준 및 가중치를 통한 평가 | <ul style="list-style-type: none"> 키워드식별 조사기준선정 개념모형작성 평가기준수립 가중치제정도 |
| 특징 | <ul style="list-style-type: none"> 세 단계를 반복적으로 수행 오픈소스 컴포넌트의 식별과 평가를 위한 모델 | <ul style="list-style-type: none"> COTS 프로세스에 근거한 절차 하향식 방법을 통한 가중치의 할당 | <ul style="list-style-type: none"> 평가속성을 제외기준과 평가기준으로 구분 조사의 과정을 반복함으로써 요구사항을 구체화 상향식 방법을 통한 가중치 배분으로 하위 속성들 간의 형평성을 유지 |

트를 구성하는 여러 가지 요인들에 따라 적절히 조정되어야 한다는 것을 확인할 수 있었다. 요구되는 기능에 대한 만족도뿐만 아니라, 라인선스나 개발 및 운영 환경의 적합성, 비용과 일정 등 여러 측면에서의 취사선택이 필요하였다. 그러나 취사선택 과정은 OSS를 선택하는 과정의 복잡성을 심화

시켰다. 특히, OSS 선정 시 제공되는 산출물이 많지 않고, 그 명세들을 신뢰할 수 없는 경우가 종종 있었다.

둘째, 본 연구가 OSS 선정 절차의 객관성과 효율성에 중점을 두고 수행되었으나, 그 평가가 정성적인 수준에서 이루어졌고, 정량적인 측면에서도 개발 기간과 노력의 절감 효과에 대한 평가만 수행되었다는 한계가 있다. 본 연구에서는 '비용'이라는 요소에 초점을 두지 않았기 때문인데, 보다 과학적인 측정을 위해서는 '비용'요소를 중심으로 한 절차의 측정과 개선 노력이 필요하다.

5. 결론 및 향후 연구 과제

최근 소프트웨어 개발의 문제점인 소프트웨어의 개발 속도 및 비용 등의 문제점을 해결하기 위한 새로운 대안으로써 OSS를 활용한 개발 방법이 시도되고 있다. 이에 따라, OSS를 활용하기 위한 다양한 분석들이 이루어졌고, 지속적으로 OSS를 활용한 여러 연구가 진행되고 있다. 그러나 실제로 OSS를 소프트웨어 개발에 활용하기에는 기존의 OSS 선정 절차에 문제점이 있다.

따라서 본 연구에서는 이러한 문제들을 해결하기 위해 OSS를 선정할 시에 개발될 소프트웨어에 대한 정확한 요구사항을 식별하고, 요구사항과 일치하는 OSS를 조사하여 후보 리스트를 추출 및 평가기준에 따라 활용 대상 OSS를 선정하는 방법을 제시하였다. 또한, 적용 실험과 평가를 통해 각 활동의 유효성과 개선점을 도출하였다.

그러나, 실제 적용의 과정에서 본 연구의 한계도 있었는데, OSS 선정 시에 사용자 요구사항의 변경, 비용과 일정 등의 다양한 변수들이 존재하고, 이러한 변수들은 OSS 커뮤니티와 같은 외부적인 요인들에 영향을 받는 만큼 OSS 선정 시의 복잡성을 더욱 심화시켰다. 그리고 OSS를 평가하는 과정에서 OSS 산출물의 표준화가 이뤄지지 않아 산출물을 정제하는 과정에서 오랜 시간을 소요하였다.

향후 연구과제로는 보다 정확하고, 객관적인 OSS 선정 및 평가를 위해 OSS 문서의 표준화 연구와 OSS 평가 매트릭에 대한 연구가 수행되어야 한다.

참 고 문 헌

[1] 김종배, 류성열, "오픈소스 소프트웨어 재사용 절차의 경험적 연구", 한국통신학회 논문지, 제13권 제2호, pp.43-58, April, 2006.
 [2] T.R, Madanmohan and Rahul De', "Open Source Reuse in Commercial Firms," IEEE, November/December, 2004.
 [3] Meng Huang, et al, "A Development Process for Building OSS-Based Applications," SPW 2005, LNCS 3840, pp.122-135, 2005.
 [4] Widdows, C., Duijnhouwer, F.W., "Open Source Maturity Model," <http://www.seriouslyopen.org>, pp.6-10, 2003.
 [5] Gacek, C. & Arief, "The many meanings of open source," IEEE Software, Vol.21, No.1, pp.34, 2004.
 [6] Widdows, C., Duijnhouwer, F.W., "Open Source Maturity

Model," <http://www.seriouslyopen.org>, pp.6-10, 2003.
 [7] Carolyn, A., ABusiness Case Study of Open Source Software, MITRE PRODUCT, 2001.
 [8] Magic and Robust Methodology Integrated(MaRMI) III Ver4.0, Electronics and Telecommunications Research Institute, 2003.
 [9] 한국정보통신기술협회, TTA.KO-11.0001~0006 단계별 소프트웨어 문서 작성 지침, 1998.
 [10] 한국정보통신기술협회, TTA.KO-11.0009 정보시스템 문서화 지침서, 1998.
 [11] 한국정보통신기술협회, TTA.IS-6592 컴퓨터기반 응용시스템 문서화 지침, 1998.
 [12] Sourceforge.net, "<http://www.sourceforge.net>".
 [13] Craig Meyers, Patricia Oberndorf: Managing Software Acquisition, SEI Series in Software Engineering, 2001.
 [14] Eric von Hippel, and George von Krogh, Exploring the Open Source Software Phenomenon: Issues for Organization Science, http://opensource.mit.edu/online_papers.php, 2002.



이 영 민

e-mail : leerin99@ssu.ac.kr

2005년 단국대학교 컴퓨터학과(학사)

2007년 숭실대학교 컴퓨터학과(공학석사)

2007년~현재 숭실대학교 컴퓨터공학과

박사과정

관심분야: 소프트웨어 개발 방법론, 요구공학,

소프트웨어 제공학/역공학, 오픈소스

소프트웨어



류 성 열

e-mail : syrheew@ssu.ac.kr

1981년~현재 숭실대학교 교수

1982년~1995년 숭실대학교 전자계산연구소

및 중앙전자계산소 소장

1997년~1998년 George Mason University

객원 교수

1998년~2001년 숭실대학교 정보과학대학원 원장

2004년~현재 한국품질재단 운영위원회 위원장

2006년~현재 공정거래위원회 성과관리위원회위원

2008년~현재 정보통신연구진흥원 이사

관심분야: 소프트웨어 유지보수, 소프트웨어 재사용, 오픈소스

소프트웨어



김 종 배

e-mail : kjb@enterprise.co.kr

2002년 숭실대학교 컴퓨터학과(공학석사)

2006년 숭실대학교 컴퓨터학과(공학박사)

관심분야: 소프트웨어 개발 방법론, 에이전트

시스템, 오픈소스 소프트웨어