

## 정량적인 OSS 선정을 위한 평가지표 연구

이후재<sup>1\*</sup>, 김두연<sup>2</sup>, 최일우<sup>3</sup>

<sup>1</sup>대림대학교 정보전략운영팀, <sup>2</sup>교육과학기술부, <sup>3</sup>강남대학교 교양학부

### A Study on Evaluation Criteria for quantitatively OSS Selection

Hoo-Jae Lee<sup>1\*</sup>, Doo-Yeon Kim<sup>2</sup> and Il-Woo Choi<sup>3</sup>

<sup>1</sup>Information Strategic Operation Team, Daelim University College

<sup>2</sup>Ministry of Education, Science And Technonlogy

<sup>3</sup>Division of General Studies, Kangnam University

**요 약** 기존의 OSS의 활용은 운영체제나 DBMS와 같은 시스템 애플리케이션의 사용이 주를 이루었다. 그러나 현재 많은 기업에서 시스템 소프트웨어가 아닌 응용 애플리케이션 중심으로 OSS를 활용하려고 한다. 그러나 OSS를 활용한 응용 애플리케이션 개발을 위해서는 기반이 되는 OSS의 선정이 무엇보다 중요하다. 기존의 OSS 평가 연구의 범위는 전체 OSS 품질을 포함하고 있다. 그러므로 OSS 선정에 대한 평가 연구는 미흡하다. 또한 평가의 결과는 정량적인 측정보다 정성적인 측정에 기반하고 있다.

본 연구에서는 기존의 OSS 평가지표들 중 선정을 위한 지표들만을 도출하고 이를 기반으로 프로젝트 특징에 따라 정량적인 평가가 가능한 평가지표를 제안한다. 제안하는 평가지표는 OSS 커뮤니티 내의 정보만을 가지고 평가할 수 있는 초기평가지표와 정량적인 측정이 가능한 상세평가지표로 구분하여 제안한다. 이를 통해 OSS 선정 시 정량적인 점수와 지표를 통해 객관적인 근거를 제공한다.

**Abstract** Utilization of the former OSS was mainly focused on the usage of system applications such as operating system and DBMS. However, nowadays many companies are trying to make use of OSS based on the application rather than the system software. However, selection of base OSS is the most important to develop of application for utilizing OSS. The scope of existing OSS evaluation studies is covered the entire OSS quality. Thus existing studies of evaluation of OSS selection is insufficient. Also, the result of assessment is based on qualitative measurement rather than quantitative ones.

In this paper, we derives only the indicators for selection among the existing OSS assessment indicators and suggests the assessment indicator that is capable of quantitative assessment in accordance with the characteristics of the project. The proposed assessment indicator is divided into an initial assessment indicator that can be assessed with only the information within the OSS community, and the detailed assessment indicator through metrics to make quantitative measurements possible. In this way, an objective basis can be provided through quantitative scores & indicators when selecting OSS.

**Key Words** : OSS, Selection, Assesment Indicator

### 1. 서론

OSS(Open Source Software)를 활용한 개발이 증가하면서 시스템 소프트웨어 뿐만 아니라 어플리케이션의 활

용 또한 증가하고 있다.

OSS를 활용한 개발[1]은 OSSD(Open Source Software Development)와 OBSD(OSS-Based Software Development) 두 가지로 분류한다. OSSD는 글로벌적으로 대형 소프트

\*Corresponding Author : Hoo-Jae Lee

Tel: +82-10-4145-2374 email: hooklee@naver.com

접수일 12년 01월 13일

수정일 12년 03월 19일

게재확정일 12년 04월 12일

웨어 시스템을 구축, 배포, 운영, 유지 보수하는 방법이고, OBSD는 OSS를 활용하여 소프트웨어를 구축하는 방법이다.

기존의 OSS 평가 연구들은 OSS 자체 품질에 대해 논의가 이루어졌다. 그러나 OSS 활용하기 위해서는 그 시작이 되는 OSS 선정에 관한 평가지표들을 정의하고 이를 정량적으로 측정할 수 있는 체계적인 방법이 필요하다.

그러나 기존의 연구들은 평가지표만을 제시할 뿐 구체적인 측정 방법이나 세부적인 측정리스트에 대해서는 언급하고 있지 않다. 그러므로 정량적인 측정을 통해 객관적인 기준을 제시할 수 있는 OSS 선정에 관한 평가지표의 연구가 필요하다.

본 연구에서는 모든 평가지표들을 측정하는 것은 어렵기 때문에 프로젝트 특성에 따른 OSS 선정 평가지표를 분류하고 이를 정량적으로 측정할 수 있는 메트릭을 제안한다.

## 2. 관련연구

Capgemini[2]의 OSS 성숙도 모델(Open Source Maturity Model, OSMM)은 조직이나 프로젝트에 적합한 OSS를 결정하기 위해 제안한 모델로, OSS 기반의 애플리케이션을 개발하는 조직이 보다 효율적으로 평가하고 구현할 수 있는 제품적도를 통해 조직적인 접근방법을 제공한다. 이 모델은 다른 OSS 컴포넌트를 평가할 경우를 고려해야 할 가장 가치 있는 지표를 강조하며, 획득 회사의 우선순위 및 비즈니스 모델에 따라 강점과 약점을 비교할 수 있는 방법을 제공해주고 있으나, 구체적인 평가 절차나 방법의 제시는 없다. 이 모델은 7개 평가단계로 구성되고 제품 평가 지표와 응용 평가 지표로 구분된다. 제품 평가 지표는 고객의 요구사항으로 평가를 실시하고 응용 평가 지표는 OSS에 대한 평가 점수를 부여하는 방법과 OSS를 선택하는 방법을 결정할 수 있는 방법을 설명한다. 제품 평가 지표는 제품, 통합, 사용, 수락의 4개 평가영역으로 분류하였으며 그 아래 12개 세부 항목과 요구사항 체크리스트를 정의한다.

Navicasoft사[3]에서 개발한 OSS 성숙도 모델은 OSS의 성숙도를 평가하는 공식적인 프로세스로서 개발조직에서 OSS를 성공적으로 구현할 수 있도록 한다. 이 모델은 성숙도 요소 평가(Assess Element Maturity), 가중치 부여(Assign Weighting Factors) 및 전반적인 성숙도 점수 계산(Calculate the Product's Overall Maturity Score)등 3개 단계로 구성된다. 평가 지표는 제품, 기술 지원, 문서화, 통합 교육 및 전문 서비스 등 6개의 평가영역으로 분

류하였으며 그 아래 19개 세부 항목과 요구사항 체크리스트를 정의한다. 이 모델은 주로 OSS를 개발하는 개발조직을 위하여 설계되고 매개의 후보에 대하여 모든 지표를 평가하기 때문에 후보의 수가 많을 경우에는 효율성이 떨어진다.

BRR(Business Readiness Rating)[4]은 개발자 또는 일반 사용자를 대상으로 하는 OSS 선정평가 모델로서 카네기 멜론과 인텔에 의해 공동으로 개발되었다. 이 모델에서는 빠른 평가, 카테고리 우선순위에 의한 평가, 데이터 수집 및 평가 진행, 최종점수 계산의 4개 단계로 나눈다.

QSOS(Qualification and Selection of Open Source software) 모델[5]은 16개의 평가 지표와 그 하위 44개 평가항목을 평가한다. 자격부여 단계에서는 가중치를 부여하고, 선정단계에서는 최후 점수를 계산하여 최종 OSS를 선정한다. 이 평가모델은 각 지표를 평가하고 총점수를 합산하여 매개 후보를 비교, 선정하는 방식으로서 후보가 많을 경우 효율적이지 못하다. 또한, 가중치 부여가 없으므로 사용자의 특수 요구사항을 반영하는데 부족함이 있다.

Rahul[6]의 OSS 컴포넌트 선정 모델은 수집(Collection), 배양(Incubation), 개정(Critical Revision)의 세 단계로 구성이 된다. 평가 지표는 비용, 수정 요구사항, 컴포넌트 특징, 라이선스, 유지보수, 지원의 6가지이다.

Meng Huang[7]은 OSS 기반의 개발 프로세스를 제안하면서, 선정단계에서 평가기준을 마련하기 위해 OSS의 속성을 제시하였다. Meng Huang의 프로세스가 COTS(Commercial On-The-Shelf) 프로세스의 기본바탕을 두고 있기 때문에 평가 속성도 COTS 프로세스에 근거하고 있다. Meng Huang은 OSS를 평가할 때 반드시 고려해야 할 평가 요소로 아키텍처 안정성, 개발도구, 코드의 이해성과 수정가능성, 합법성을 추가하였다.

IT서비스 전문 업체인 Optaros[8]는 기업들이 OSS 도입에서 지침서로 활용할 수 있는 보고서를 발표해 OSS의 체계적인 분류를 시도했다. 기업들이 사용할 수 있는 OSS 플랫폼, 컴포넌트, 프레임워크, 솔루션 등을 공통적인 특성에 따라 4개의 범주로 구분하고 세부적으로 29개 분야, 260종 이상의 OSS의 성숙도를 분류하고 있다. 또한, OSS를 기능성, 커뮤니티, 성숙도와 트렌드라는 판단 기준에 근거하여 평가하였고 여기에 기업 적응도라는 측정을 추가하여 각 제품들의 평가결과를 제시하였다.

Dan Wood[9]의 OSS 성숙도 모델은 평가 절차의 제시 없이 평가 지표를 제품 기준, 사용 기준, 통합 기준 등 3개의 평가영역으로 분류하였으며 그 아래 12개 세부 항목과 요구사항 체크리스트를 정의한다.

Kenwood[10]의 연구는 Linux 사용을 기반으로 하는 OSS 비즈니스 케이스를 분석한 연구로서, 사용자로 하여

금 적용한 OSS와 개발방법론이 프로젝트의 요구에 부합 되는지 확인하고 그 품질을 평가할 수 있도록 한다.

### 3. OSS 선정 평가 지표 도출

본 장에서는 제안한 OSS 선정 프로세스의 초기단계와 상세단계 수행 시 정량적으로 평가할 수 있는 평가 지표와 이를 측정할 수 있는 메트릭을 제안한다. OSS의 Life Cycle은 커뮤니티 기반으로 이뤄진다. 초기단계는 커뮤니티 사이트의 정보를 기초로 후보 OSS를 선정하는 과정이며, 상세단계는 선정된 후보 OSS 중 프로젝트에 따른 가중치를 적용한 평가지표의 측정값을 통해 최종 대상 OSS를 선정하는 과정이다.

초기단계와 상세단계의 평가 지표를 선정하기 위해 기존의 평가 지표들로부터 평가 지표를 도출하기 위한 과정을 기술한다.

이를 위해 기존 평가 지표 연구로부터 SMART(Specific, Measurable, Acceptable, Relevant, Timely)분석과 Mind Map을 통하여 평가 지표를 도출하고, 이를 기반으로 초기단계와 상세단계 평가 지표로 구분한다. 그리고 이를 측정할 수 있는 메트릭을 제안한다.

본 연구에서는 관련 연구를 기반으로 하여 평가 지표를 식별한다. 선정을 위한 연구에서 언급하고 있는 모든 평가 지표를 수집한다. 이 결과 그림 1에서 보는 바와 같이 지표의 연관성을 고려하여 중복되는 것을 제거하고 총 65개의 지표를 식별하였다.

| 평가지표 식별              |         |            |         |              |
|----------------------|---------|------------|---------|--------------|
| 기능성                  | 사용성     | 매일링 리스트    | 적용성     | 기술성          |
| 비기능적합성               | 이해가능성   | 변경 이력      | 실지용이성   | 응용성          |
| 상호운용성                | 사용용이성   | 개발도구지원     | 전포넌트호환성 | 암법성: 웹/아이전스  |
| 보안성                  | 운용성     | 성능         | 버전 호환성  | 비용           |
| 신뢰성                  | 지원      | 유지보수성      | 확장성     | 정책           |
| 결함(오류) 허용성           | 커뮤니티 지원 | 분석가능성      | 분할의 위험  | 버전 배포 활성     |
| 복구성                  | 커뮤니티    | 변경가능성      | 패키징     | 문서화          |
| 성숙성                  | 유료 지원   | 견고성/안정성    | 판매기성    | 사용자 공헌 프레임워크 |
| 경험                   | 교육 지원   | 시험가능성      | 시장주도기업  | 마케팅          |
| 개발 지원                | 이식성     | 수명         | 전문성     | 관련도서         |
| 공용 API / 외부 서비스      | 조직관리    | 스폰서        | 분기      | 웹 포스트        |
| 참조 가능한 적용            | 컨설팅     | 호환성        | 개발조직    | 통합성          |
| Third-Party Plug-ins | 표준      | 소프트웨어 요구사항 | 목표와 기원  | 모동화          |

[그림 1] 문헌기반의 평가 지표 수집  
[Fig. 1] Evaluation of related work-based Evaluation Criteria

수집된 65개의 평가 지표 중 OSS 선정과 관련 있는 평가 지표를 도출하기 위해서 OSS 전문가 집단을 대상으로 설문을 수행하여 SMART 분석을 통해 5가지 질문(지표의 정확성, 측정 가능성, 수용 가능성, 연관성, 적시성)에 모두 해당하는 것만을 평가 지표로 도출한다. 이를 통해 65개 중 14개의 평가 지표를 도출한다.

도출된 14개의 평가 지표 중 커뮤니티 지원, 합법성, 문서화(사용자 문서, 개발자 문서), 호환성은 다른 평가 지표와는 별개로 OSS 커뮤니티에서 제공되는 정보를 통해 평가가 가능하다고 판단하여 이를 OSS 후보를 검색하기 위해 사용하는 초기평가의 지표로써 사용한다.

초기평가 요소인 4개를 제외한 10개의 평가 지표는 상세평가 지표로써 활용하여야 한다. 그러나 10개라는 모든 지표를 적용하기에는 시간과 노력 측면에서 많은 어려움이 있다.

그러므로 본 연구에서는 초기평가 단계의 프로젝트 도메인의 특징[11-13]에 기반을 두어 조직의 기술 수준, 초기 요구사항의 안정성 및 명확도, 사용 기술 성숙도, 프로젝트 일정의 중요성, 소프트웨어 중요도라는 5가지 항목에 대한 가장 큰 프로젝트 중요도를 반영하여 평가 지표를 적용한다.

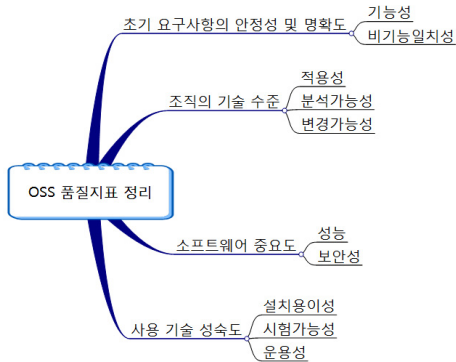
조직의 기술 수준 : 프로젝트 개발 조직이 소프트웨어 관련 기술에 대한 경험이 없을 경우 해당하는 지표들을 사용하여 OSS 평가 및 선정을 한다. 이 경우의 사례로는 개발 조직이 대다수가 초급 수준의 개발자로 프로젝트 팀이 구성이 되고 OBSD 개발 경험이 많지 않을 때 개발이 쉬운 OSS를 선정해야 한다.

사용 기술 성숙도 : 프로젝트에서 사용되는 기술(개발 환경)이 성숙되지 않은 신기술을 이용해서 프로젝트를 개발해야 하는 경우 해당하는 지표들을 사용하여 OSS 평가 및 선정을 한다. 이 경우의 사례로는 최신버전의 WAS(Web Application Server)나 최신 웹 프로그래밍 기법을 적용하는 프로젝트일 때 OSS의 수정 보안을 최소화할 수 있는 OSS를 선정해야 한다.

초기 요구사항의 안정성 및 명확도 : 초기의 요구사항이 모호한 경우 해당하는 지표들을 사용하여 OSS 평가 및 선정을 한다. 이 경우의 사례로는 초기 요구사항이 모호할 때 평가 지표들에 의해 요구사항을 명확히 정의해 가며 요구사항에 적합한 OSS를 선정해야 한다.

소프트웨어 중요성 : 프로젝트가 재정적인 큰 피해를 가져올 수 있거나 조직의 목표 달성에 관계 될 경우 해당하는 지표들을 사용하여 OSS 평가 및 선정을 한다. 이 경우의 사례로는 은행 업무 시스템 개발 시 보안, 오류, 성능 등에 대한 요소에 문제가 발생하면 심각한 문제가 발생되므로 이러한 요소들을 고려해야 한다.

프로젝트 일정의 중요성 : 완료시점에 따라 프로젝트의 성공 여부가 결정되는 경우에 해당하는 지표들을 사용하여 OSS 평가 및 선정을 한다. 이 경우의 사례로는 선거용지의 개표시스템을 개발할 때 개표일 까지 프로젝트가 완료되어야 하므로 프로젝트 일정 준수가 중요한 사항이 된다. 프로젝트 일정을 준수 할 수 있기 위해서는 통합, 설치, 호환 등에 대한 요소들을 고려해야 한다.



[그림 2] 프로젝트 특징에 따른 Mind Map 결과  
[Fig. 2] Mind Map Result based Project Characteristics

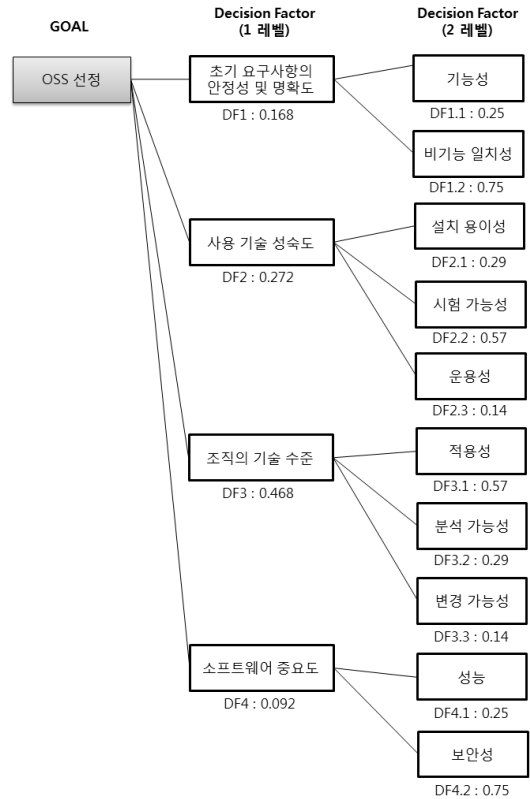
그러나 프로젝트 일정의 중요성은 OSS를 활용하여 개발하는 궁극적인 목적이 일정 단축에 해당됨으로 항목에서 제외한다.

이를 위해 프로젝트 특징의 4가지 항목에 대한 Mind Map을 수행하여 다음 그림 2와 같은 결과를 확인한다.

본 연구에서는 가중치를 적용하기 위해 AHP(Analytic Hierarchy Process) 기법을 적용한다. AHP 기법은 다면적 평가기준을 통한 의사결정을 지원한다.

의사결정을 위한 사항들을 식별하고 계층적으로 분류하여 의사결정 모형을 설정한다. 의사 결정 모형에서 동일 수준의 사항에 대해서 쌍대비교(Pairwise Comparison)를 통해 가중치 판단자료를 수집하고 각 사항의 상대적 가중치를 결정한다. 그림 3은 대학 ERP(Enterprise Resource Planning) 개발 프로젝트 수행조직들을 대상으로 프로젝트 중요도 설문을 통해 레벨1과 레벨2의 가중치를 반영한 의사 결정 모형이다.

상세평가 시 제안한 프로젝트 특징에 따라 분류한 평가 모델은 AHP를 이용하여 가중치를 구하고, 이를 기반으로 프로젝트 요구사항에 적합한 평가 지표만을 적용하여 상세평가를 수행한다. 프로젝트는 초기 요구사항의 안정성 및 명확도의 기능성과 사용기술 성숙도의 설치용이성이 기준이 된다고 판단되면 기능성과 설치용이성에 대한 메트릭을 측정하여 가중치 값에 적용한다.



[그림 3] AHP 기반의 가중치 결과  
[Fig. 3] Priority Result based AHP

## 4. OSS 선정 평가 지표

### 4.1 초기 평가 지표

초기단계의 선정 및 평가는 OSS 커뮤니티 내에서 제공하는 정보로써 판단할 수 있는 지표로 후보 OSS를 검색해야 한다. 그러므로 무엇보다 쉽게 체크리스트를 통해 평가를 해야 한다는 것이다. 초기평가 체크리스트는 표 1과 같이 15개의 체크리스트를 통해 커뮤니티 지원, 합법성, 문서화, 호환성이라는 초기평가 지표에 대해 평가한다.

[표 1] 초기단계 평가를 위한 체크리스트  
[Table 1] Checklists for Evaluation of Initial Stage

| 평가 지표   | 체크리스트                           |
|---------|---------------------------------|
| 커뮤니티 지원 | 해당 프로젝트에 대한 사용자들의 평가가 80점 이상인가? |
|         | 포럼, 위키 등 커뮤니티를 제공하는가?           |
|         | 버그보고 및 수정이 이뤄지고 있는가?            |

|     |                           |
|-----|---------------------------|
| 합법성 | 라이선스 정보가 있는가?             |
|     | 무료 사용인가?                  |
|     | 자유 배포가 가능한가?              |
| 문서화 | 문서화가 되어 있는가?              |
|     | 설치, Help 등 사용자 문서를 제공하는가? |
|     | 개발자 문서를 제공하는가?            |
|     | 문서가 자주 업데이트 되는 가?         |
|     | 문서작성 언어가 영어를 기본으로 하는가?    |
| 호환성 | 프로젝트의 개발 언어와 동일한가?        |
|     | 개발 언어의 운영 환경 버전이 동일한가?    |
|     | 프로젝트에 사용되는 WAS를 지원하는가?    |
|     | 프로젝트에 사용되는 DB를 지원하는가?     |

Project Management(PM) 관련 OSS에 대해서 초기 평가 수행서를 작성하고, 이들 중에서 후보 OSS를 선정한다. OSS 커뮤니티 사이트인 sourceforge.net에서 Project Management 분류 카테고리를 선정하여 Java 키워드로 검색을 하고 사용자 권장 항목을 추가로 필터링을 하여 사용자 80%이상 권장하는 OSS 10개가 검색 되었다. 커뮤니티 사이트와 연관 포럼 및 위키 등을 조사한 결과를 체크리스트에 작성하고 라이선스 정보가 불분명한 OSS와 문서 작성 언어가 영어가 아닌 OSS를 제외하여 5개의 후보 OSS를 선정하였다. 표 2는 선정 키워드인 Project Management, Java를 가지고 검색을 하여 선정된 10개의 OSS 중 3개의 OSS에 대해 초기평가를 수행한 예이다.

[표 2] PM 관련 후보 OSS에 대한 초기평가  
[Table 2] Initial assessment for PM-related OSS candidates

| 구 분                | Endeavour ALM | GTD-Free    | Project.net    |
|--------------------|---------------|-------------|----------------|
| 사용자들의 평가(추천 80%이상) | O (100%)      | O (100%)    | O (87%)        |
| 포럼, 위키 등 커뮤니티 제공   | O             | O           | O              |
| 버그리포트              | X             | X           | O              |
| 라이선스정보             | O             | O           | O              |
| 소스 무료사용 / 자유배포     | O             | X (소스제공 안됨) | O (JSP만 수정 가능) |
| 문서화                | X             | O           | O              |
| 사용자 문서             | X             | O           | O              |
| 개발자 문서             | X             | X           | O              |
| 문서 업데이트            | X             | 2010        | 2010           |
| 문서 작성 언어           | X             | 영어          | 영어             |
| 개발언어               | O             | Java Swing  | JSP, Java      |

|              |                   |                                  |        |
|--------------|-------------------|----------------------------------|--------|
| 개발언어 운영환경 버전 | -                 | -                                | -      |
| 지원 WAS       | -                 | -                                | -      |
| 지원 DB        | MySQL, PostgreSQL | Flat-file, Other file-based DBMS | Oracle |
| 후보 OSS       | X                 | X                                | O      |

#### 4.2 상세 평가 지표

상세평가는 선정된 후보 OSS를 프로젝트 특징에 의한 가중치 분석을 통해 선정된 평가 지표에 대한 정량적인 평가를 수행하여 대상 OSS를 선정 할 수 있는 자료 제공을 한다.

프로젝트의 특징에 따라 상세평가 지표를 선정하고, 선정된 상세평가 지표에 대한 평가 메트릭(14-17)으로 평가한다. 초기평가에서 프로젝트 특징 분석이 이뤄졌다면 프로젝트 특징에 의한 가중치 분석 태스크는 생략될 수도 있다. 프로젝트 특징에 의한 가중치 분석 태스크는 이미 후보 OSS가 선정이 되었을 때 초기평가를 실시하지 않고 상세평가를 실시할 때를 고려한 태스크이다. 그 외 사항은 각 태스크별 상세평가 지표 선정 및 메트릭에 의한 정량적인 평가를 실시하는 상세평가를 적용하여 수행한다. 프로젝트 특징별 상세평가 적용에 대한 평가 기준은 아래와 같다.

적용성 : 적용성은 적용성 비율 메트릭으로 측정한다. 요구사항 정의서에서 요구하는 최소한의 UI (User Interface) 기능과 OSS가 제공하는 UI 기능의 비율로 평가를 하고, OSS가 제공하는 UI 기능이 적을 경우 조직의 기술 수준에 따라 유지보수 시간과 노력이 좌우된다. 적용성은 0-100%의 값을 가지며 적용성의 값이 높을수록 적용성이 좋다는 의미이다.

$$\text{적용성} = \frac{\text{OSS가 제공하는 UI 기능}}{\text{요구되는 최소한의 UI 기능}} \times 100$$

분석 가능성(유지보수성) : 분석 가능성(유지보수성)은 분석 가능성 메트릭으로 측정한다. 전체 제공 소스의 수와 읽을 수 있는 소스의 수 비율로 측정을 하고, 읽을 수 있는 소스 코드의 수가 적을 경우 조직의 기술 수준에 따라 유지보수의 시간과 노력이 좌우된다. 분석 가능성은 0-100%의 값을 가지며 분석 가능성의 값이 높을수록 분석 가능성이 좋다는 의미이다.

$$\text{분석 가능성} = \frac{\text{읽을 수 있는 소스의 수}}{\text{전체 제공 소스의 수}} \times 100$$

변경 가능성(유지보수성) : 변경 가능성(유지보수성)은

변경 가능성 비율 메트릭으로 측정한다. 요구사항과 관련된 소수와 변경될 소스의 비율로 측정을 하고, 변경될 소스의 수가 많은 경우 조직 기술 수준에 따라 변경하는 노력과 시간에 영향을 준다. 변경 가능성은 0-100%의 값을 가지며 변경 가능성의 값이 높을수록 변경 가능성이 좋다는 의미이다.

$$\text{변경 가능성} = \left[ 1 - \frac{\text{변경될 소스의 수}}{\text{요구사항과 관계있는 소스의 수}} \right] \times 100$$

설치 용이성 : 설치 용이성은 설치에 소요되는 시간과 설치 후 발생하는 에러(Error) 및 경고(Warning)를 제거하는데 소요되는 시간의 합으로 설치 용이성 메트릭을 측정한다.

$$\text{설치 용이성} = \text{설치에 소요되는 시간} + \text{에러 및 경고 제거 소요 시간}$$

시험가능성 : 시험 가능성은 테스트 케이스 수와 수행된 테스트 케이스수의 메트릭을 통해 측정한다. 시험 가능성은 테스트 케이스 중 수행된 테스트 케이스와의 비율로 측정하고, 사용 기술의 성숙도는 시험에 소요되는 노력과 시간에 영향을 준다. 시험 가능성은 0-100%의 값을 가지며 값이 높을수록 시험 가능성이 좋다는 의미이다.

$$\text{시험 가능성} = \frac{\text{수행된 테스트 케이스 수}}{\text{테스트 케이스 수}} \times 100$$

운용성 : 운용성은 시스템에 대한 컨트롤 여부에 대한 측정이다. 운용성은 데이터베이스를 수정하지 않고 OSS의 기능만으로 데이터베이스를 컨트롤 할 수 있는 가에 대한 측정으로 평가하고, 사용 기술의 성숙도에 따라 운용성에 소요되는 노력과 시간에 영향을 준다. 운용성은 0-100%의 값을 가지며 값이 높을수록 운용성이 좋다는 의미이다.

$$\text{운용성} = \frac{\text{데이터를 변경할 수 있는 기능의 수}}{\text{데이터를 입력시 변경이 되어야하는 기능의 수}} \times 100$$

기능성 : 기능성은 요구되는 OSS 기능 Feature 개수의 일치 정도 메트릭을 통해 측정한다. 요구되는 OSS 기능 Feature 개수의 일치 정도는 개발 애플리케이션과 OSS 연관 애플리케이션을 식별을 통해 요구되는 OSS 기능의 Feature 개수 중 후보 OSS의 기능이 일치되는 Feature 개수의 비율로 측정한다. Feature의 기능 일치 정도는 0-100%의 값을 가지며 값이 높을수록 기능성이 좋다는 의미이다.

$$\text{기능성} = \frac{\text{후보 OSS 기능 일치 Feature 개수}}{\text{요구되는 OSS 기능 Feature 개수}} \times 100$$

비기능 일치성 : 비기능 일치성은 아키텍처 유사도와 비기능 범위 메트릭으로 측정한다. 목표 애플리케이션의 전체 비기능수와 OSS의 비기능수를 비교하여 측정한다. 비기능 일치성은 0-100%의 값을 가지며 비기능 일치성의 값이 높을수록 비기능 일치성이 좋다는 의미이다.

$$\text{비기능 일치성} = \frac{\text{구현된 후보 OSS 비기능 일치 항목 개수}}{\text{요구되는 비기능항목 개수}} \times 100$$

성능 : 성능은 각 후보 OSS 당 각각의 기능에 초당 처리량 성능테스트 결과의 평균값을 분산에 의한 범위 값 메트릭으로 비교하여 측정한다. 성능은 소프트웨어 중요도에 영향을 준다. 성능은 성능 평균값과 분산의 관계로 범위 값을 가지며 성능이 평균값보다 높을수록 성능이 좋다는 의미이다. 수식에서 i는 i 번째 성능 테스트를 나타내고 n은 성능 테스트 총 회수를 의미한다.

$$\text{성능} = \frac{\sum_{i=1}^n i \text{ 성능 테스트 결과}}{n}$$

보안성 : 보안성은 보안성 비율 메트릭으로 측정한다. 보안 비율은 DB(Data Base)와 연관되는 기능을 가진 전체 클래스 수 중 클라이언트의 요구사항을 받은 클래스가 직접 DB 연관 기능을 수행하지 않고, 디자인 패턴을 통한 DB 연관 기능과 그 외의 기능을 클래스로 분리한 클래스의 수로 측정한다. 보안성은 소프트웨어 중요도에 영향을 준다. 보안성은 0-100%의 값을 가지며 값이 높을수록 보안성이 좋다는 의미이다.

$$\text{보안성} = \frac{\text{디자인 패턴 적용 DB 연관 클래스 수}}{\text{DB 연관 클래스 수}} \times 100$$

상세평가 수행서에는 표 3과 같이 정량적인 평가를 위해 상세평가 메트릭을 적용하여 평가한 점수에 대해 집계를 하여 합계 점수를 작성한다. Project Management 관련 후보 OSS 평가에 대한 상세평가 수행서인 표 3에 그림 3의 레벨 1 또는 레벨 2의 AHP 기반의 가중치 값을 적용하여 합계점수가 높은 것을 대상 OSS로 선정한다. 표 3은 초기 평가된 후보 OSS 5개 중 3개의 OSS에 대해 상세평가를 수행한 예이다. 초기 요구사항 및 명확도의 기능성과 조직 기술수준의 적용성에 메트릭 측정과 가중치를 Project.net OSS에 적용하였다. 초기 요구사항 및 명확도의 가중치 0.168, 기능성은 0.25, 메트릭 측정값 1의 곱은 0.042 이다. 조직 기술수준 0.468, 적용성 0.57, 메트릭 측정값 3의 곱은 0.80028 이다. 이렇게 상세평가를 실시한 5개의 OSS 중 Project.net이 합계점수가 높았다.

[표 3] PM 관련 후보 OSS에 대한 상세평가

[Table 3] Detailed assessment of PM-related OSS candidates

| 선         | 프로젝트 특징         | 품질지표   | xplanner-plus | Onepoint Project | Project.net |
|-----------|-----------------|--------|---------------|------------------|-------------|
|           | 기술 수준           |        | 적용성           | 1                | 2           |
|           |                 | 분석가능성  | 1             | 2                | 1           |
|           |                 | 변경가능성  | 1             | 1                | 1           |
|           |                 | 계      | 3             | 5                | 5           |
| 사용 기술 성숙도 |                 | 설치용이성  | 4             | 2                | 4           |
|           |                 | 시험가능성  | 1             | 1                | 1           |
|           |                 | 운용성    | 1             | 1                | 2           |
|           |                 | 계      | 6             | 4                | 7           |
| 택         | 초기              | 기능성    | 1             | 1                | 1           |
|           | 요구사항의 안정성 및 명확도 | 비기능일치성 | 1             | 1                | 2           |
|           |                 | 계      | 2             | 2                | 3           |
|           | 소프트웨어 중요도       | 성능     | 2             | 2                | 3           |
|           | 보안성             | 2      | 2             | 3                |             |
|           | 계               | 4      | 4             | 6                |             |

### 5. 제안한 연구의 비교 평가

본 연구에서는 초기평가를 위한 체크리스트를 제안하였고, 초기평가를 통한 후보 OSS를 대상으로 상세평가를 하기위한 메트릭을 제안하였다. 제안된 메트릭이 올바르게 정의되었는지를 검증하기 위해 Kitchenham[18]의 접근에 기반을 두어 이론적인 검증을 수행하였다. 이론적인 분석은 측정이 정의된 범주 안에서 유효한지 유효하지 않은지를 판단하는 근거를 제공하며 유효한 측정은 Attribute, Unit, Instrument, Protocol에 대해서 유효성을 가져야 한다. 4개의 유효성에 대한 정의는 다음과 같다.

**Attribute 유효성 :** 측정 가능한 엔티티에 의해 실제 제출되는 것이 관심이 있는 어트리뷰트인지를 판단한다. 어트리뷰트 검증은 직접적으로 측정 가능한 어트리뷰트와 다른 어트리뷰트로부터 도출되는 간접적인 어트리뷰트를 모두 고려해야 한다.

**Unit 유효성 :** 어트리뷰트를 측정하기 위한 수단으로써 측정 단위가 적절히 사용되었는지를 판단한다.

**Instrument 유효성 :** 측정 도구가 적절한지 혹은 측정 도구가 적절히 측정되는지를 판단한다.

**Protocol 유효성 :** 수용 가능한 측정 프로토콜이 채택되었는지 안되었는지를 판단한다.

이러한 네 가지 측정 요소를 기반으로 이론적인 검증을 수행하였다.

**Attribute 유효성 :** 본 연구에서는 기존의 OSS 선정을 위한 평가 지표들을 수집하여 총 66개의 평가 지표들을

식별하였다. 이 66개의 평가 지표들 중 SMART 분석을 통해 14개의 평가 지표들이 OSS 선정과 연관이 있다고 판단되었으며, 이중 4개의 평가 지표는 초기평가 지표로 10개는 상세평가 지표로 구분하였다. 이렇듯 SMART 분석을 통해 상세평가 지표들을 기존의 평가 지표들로부터 도출하였기 때문에 OSS 선정을 위해 필요한 평가 지표라는 것을 확인하였다.

**Unit 유효성 :** 본 연구에서는 각 메트릭의 특성에 따라 동일한 측정 단위를 가지고 있지는 않다. 그러나 이러한 문제를 해결하기 위해 각 측정결과에 대한 범위를 정하였으며, 이를 unacceptable, bad, acceptable, very good, excellent의 5 단계로 통일하여 1~5점까지의 점수를 동일하게 적용하였다. 그러므로 제안한 메트릭의 측정단위가 동일하다고 판단할 수 있다.

**Instrument 유효성 :** 제안한 메트릭 중 설치용이성 하나만이 분자, 분모의 형태가 아니다. 나머지 9개의 메트릭은 모두 분자, 분모의 형식을 가지고 있고 정략적인 측정이 가능하며, 각 분자와 분모에 해당하는 측정 데이터가 구분되어 있다.

**Protocol 유효성 :** 다음 표 4에서 보는 바와 같이 제안한 메트릭의 분자와 분모를 측정하기 위해서 어떠한 자료로부터 분자와 분모가 도출되는지 확인할 수 있다.

[표 4] 측정 데이터의 도출 출처

[Table 4] Sources of derived measurement data

| 구분    | 출처   |  |
|-------|--|--|
|       | 분자   | 분모   |
| 적용성   | OSS가 제공하는 UI 기능은 OSS가 제공되는 화면을 보고 확인   | 요구되는 최소한의 UI 기능은 입력 산출물의 요구사항으로부터 확인 가능                                    |
| 분석가능성 | 읽을 수 있는 소스의 수는 Java, JSP, JS, HTML, HTM, XML, TXT, properties 등의 확장자를 자치며 어가 아닌 일반적인 편집 프로그램에서 읽을 수 있는 파일의 수로 확인 | 전체 제공 소스의 수는 OSS가 가지는 전체 파일의 수를 통해 확인                                      |
| 변경가능성 | 변경될 소스의 수는 분모에 해당하는 소스 중 요구사항과 일치하지 않아 기능 변경이나 추가가 요구되는 소스의 수로 확인  | 요구사항과 관계있는 소스의 수는 입력 산출물 요구사항과 관련 있는 OSS의 메뉴 구성을 확인 후 이에 해당하는 소스의 수로 확인 가능 |
| 설치용이성 | 설치에 소요되는 시간을 측정하여 확인하고 에러 및 경고 제거 소요시간은 실제 에러 및 경고를 제거하는 데 소요된 시간을 측정하여 확인                                       |  |

|        |   |  |
|--------|---|--|
| 시험가능성  | 수행된 테스트 케이스 수는 OSS 적용하여 테스트 한 결과 정상적인 테스트 수행을 마친 테스트 케이스의 수로 확인                 | 테스트 케이스 수는 입력 산출물의 요구사항을 기반으로 작성된 테스트 케이스 수                    |
| 운용성    | 데이터를 변경할 수 있는 기능의 수는 분모 중 입력된 데이터를 OSS에 제공되는 기능으로 변경이 가능하도록 제공되는 기능의 수를 조사하여 확인 | 데이터를 입력 시 변경이 되어야 하는 기능의 수는 OSS가 데이터 입력 기능을 가지는 기능의 수를 조사하여 확인 |
| 기능성    | 후보 OSS 기능 일치 Feature 개수는 분모 중 OSS가 제공하는 기능으로 확인                                 | 요구되는 OSS 기능 Feature 개수는 입력 산출물을 통해 확인                          |
| 비기능일치성 | 구현된 후보 OSS 비기능 일치 항목 수는 분모에 해당하는 OSS에서 제공하는 비기능의 맵핑을 통해 확인                      | 요구되는 비기능 항목 개수는 입력 산출물을 통해 확인                                  |
| 성능     | 성능 테스트 결과는 Apache JMeter에 의해 측정 가능  | N은 성능 테스트의 총 회수로 확인  |
| 보안성    | 디자인 패턴 적용 DB 연관 클래스 수는 분모에 해당하는 클래스 중 디자인 패턴을 적용한 클래스를 파악하여 확인                  | DB 연관 클래스 수는 DB와 연동되는 기능을 포함하는 클래스의 수를 파악하여 확인                 |

마지막으로 제안한 평가 지표와 기존의 평가 지표 연구와의 비교를 통해 본 연구의 차별성을 평가한다. 평가 지표들의 비교를 위해 평가방법, 평가척도, 평가가중치, 프로젝트 특성 고려, 정량적 평가라는 5가지 항목으로 비교하였다. 표 5는 이러한 평가요소들을 가지고 비교한 결과이다.

**[표 5]** 기존 평가 지표 연구와의 비교  
**[Table 5]** Comparison with existing evaluation metrics research

| 연구              | 평가 지표 | 평가 방법 | 평가 척도 | 평가 가중치 | 프로젝트 특징 고려 | 정량적 평가 |
|-----------------|-------|-------|-------|--------|------------|--------|
| Capgemini OSMM  | O     | O     | O     | X      | X          | △      |
| Navicasoft OSMM | O     | O     | X     | O      | X          | X      |
| QSOS            | O     | O     | O     | O      | X          | △      |
| OpenBRR         | O     | O     | O     | O      | △          | △      |
| Rahul           | O     | X     | X     | X      | X          | X      |
| Huang           | O     | O     | X     | O      | △          | X      |
| Optaros         | O     | X     | △     | X      | X          | X      |
| Dan Wood OSMM   | O     | O     | O     | X      | X          | △      |

|         |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|
| Kenwood | O | O | X | X | X | X |
| 제안 연구   | O | O | O | O | O | O |

\* O: 지원, △ 일부 지원, X: 미지원

결과에서 보는 바와 같이 기존 연구들은 평가방법은 있고 평가척도는 없는 경우와 방법과 척도는 있지만 가중치가 없는 경우가 많이 있는 것을 볼 수 있다. 특히 정량적인 평가가 거의 수행되지 않는다는 점이 본 연구와의 차이점이다.

## 6. 결론

OSS는 다양한 커뮤니티를 근간으로 각종 SW의 개발과 업그레이드가 상업용 SW에 비해 훨씬 수월하다. 대부분 인터넷을 통한 무료 다운로드 형태로 제공되어 신속한 개발과 제공이라는 장점을 가지고 있기 때문에 시장의 참여와 확산을 주도하고 있다.

그러나 이러한 다양한 OSS중 선정을 통해 OSS를 활용해야 하기 때문에 OSS 선정에 대한 중요성이 크다. 그러나 기존의 평가 연구들은 OSS 선정 뿐만 아니라 전체 OSS 품질에 대한 평가지표에 대한 연구가 수행되었으며, 이를 OSS 선정에 적용하는 것은 어렵다. 그러므로 본 연구에서는 커뮤니티 사이트 등의 정보를 기초로 후보 OSS를 선정하는 과정을 초기평가 지표, 프로젝트 가중치 평가 지표로 메트릭에 의한 평가를 하는 상세평가 지표로 구분하여 제안하였다. 또한, 기존 연구들의 평가 지표를 종합하고 이를 SMART 분석을 통해 OSS 선정과 관련되어 있다고 판단되는 지표만을 식별하고 이를 제안한 프로세스에 적합한 초기지표와 상세지표로 분류하였다. 상세지표에 해당하는 지표들은 프로젝트 특징을 기반으로 Mind Map을 실시하여 평가 지표를 분류하였다. 또한, 모든 상세지표에 대해 평가를 수행하는 것은 시간적, 인력적으로 낭비이기 때문에 프로젝트 특성에 우수한 상세지표만을 평가토록 제안하였다. 그러나 프로젝트 특성에 대한 가중치를 OSS 선정자의 임의대로 하는 것이 아닌 AHP를 활용하여 가중치를 객관적인 수치로 환산하여 제안하였다.

제안한 프로세스와 평가 지표에 대한 적용성 및 우수성을 평가하기 위해 평가항목을 도출하였으며, 이를 통해 이론적인 검증을 수행하였다. 또한, 기존 연구와의 비교를 통해 제안한 연구의 우수성을 평가하였다.

그러나 본 논문에서는 일반적인 응용 어플리케이션만을 대상으로 하였기 때문에 모든 응용 어플리케이션에 적용되기는 어렵다. 그러므로 모바일이나 유비쿼터스 같



은 다른 도메인의 OSS에도 적용하기 위해서는 각 적용 도메인에 대한 추가적인 연구와 이에 대한 특징들을 반영한 선정 기준들에 대한 연구가 추가적으로 필요하다.

### References

[1] Walt Scacchi, "OSS Development: Recent Research Results and Methods", *Advances in Computer*, No.69, pp.243-295, 2007.

[2] CapGemini, *Open Source Software Maturity Model*, 2003.

[3] Navica, *Making Open Source Ready for the Enterprise: The Open Source Maturity Model*, 2005.

[4] Intel, *Spike Source, The Business Readiness Rating model: An evaluation framework for open source*, 2005.

[5] Atos Origin, *Method for Qualification and Selection of Open Source software: QSOS*, 2006.

[6] T.R. Madanmohan and Rahul De', "Open Source Reuse in Commercial Firms", *IEEE Software*, Vol.21, No.6, pp.62-69, 2004.

[7] Meng Huang, Liguang Yang, Ye Yang, "A Development Process for Building OSS-Based Applications", *LNCS* 3840, pp.122-135, 2005.

[8] Optaros, *Open Source Catalogue 2007 U.S. Version 1.1*, 2007.

[9] Dan Woods, Gautam Guliani, *Open Source for the Enterprise*, O'Reilly, 2005.

[10] Carolyn A. Kenwood, *A Business Case Study of Open Source Software*, The MITRE Corporation, 2001.

[11] Pankaj Jalote, *CMM in Practice-Processes for Executing Software Projects at Infosys*, Addison-Wesley, 2002.

[12] Aaron J. Shenhar, Dov Dvir, "Toward a typological theory of project management", *Research Policy*, Elsevier, Vol.25, pp.607-632, 1996.

[13] Mark P. Ginsberg, Lauren H. Quinn, *Process Tailoring and the Software Capability Maturity Model*, Technical Report CMU/SEI-94-TR-024, 1995.

[14] ISO/IEC 9126-1, *Software Engineering-Product Quality-Part 1 : Quality model*, 2001.

[15] ISO/IEC 9126-2, *Software Engineering-Product Quality-Part 2 : External Metrics*, 2001.

[16] ISO/IEC 9126-4, *Software Engineering-Product Quality-Part 4 : Use in Metrics*, 2001.

[17] ISO/IEC 14598-6, *Software Engineering Product Evaluation-Part 6 : Documentation of Evaluation Modules*, 2001.

[18] Barbara Kitchenham, Shari Lawrence Pfleeger, Norman

Fenton, "Towards a Framework for Software Measurement Validation", *IEEE Transactions on Software Engineering*, Vol.21, No.12, pp.929-944, 1995.

### 이 후 재(Hoo-Jae Lee)

[정회원]



- 2004년 2월 : 송실대학교 정보과 대학원 (공학석사)
- 2011년 2월 : 송실대학교 일반대학원 (공학박사)
- 2004년 5월 ~ 현재 : 대림대학교 정보전략운영팀

<관심분야>

컴포넌트 기반 개발, 소프트웨어 재사용, 소프트웨어 아키텍처, 소프트웨어 프로세스, 오픈 소스 소프트웨어

### 김 두 연(Doo-Yeon Kim)

[정회원]



- 2007년 2월 : 송실대학교 일반대학원 (공학박사)
- 1994년 5월 ~ 1997년 5월 : 교육부(기획관리실, 교육정보관리국)
- 1997년 5월 ~ 2002년 5월 : 서울대학교(연구처, 학술정보원)
- 2002년 5월 ~ 2008년 2월 : 교육인적자원부(교육정보화지원과장)
- 2008년 2월 ~ 현재 : 교육과학기술부 교육정보화과장

<관심분야>

공개소프트웨어, 유비쿼터스, 정보화정책

### 최 일 우(II-Woo Choi)

[정회원]



- 1997년 2월 : 송실대학교대학원 (컴퓨터공학석사)
- 2004년 2월 : 송실대학교대학원 (컴퓨터공학박사)
- 2007년 3월 ~ 현재 : 강남대학교 교양학부 교수

<관심분야>

개발프로세스, 소프트웨어 재사용, USN, 모바일컴퓨팅