

IFCS에 의한 오픈소스 소프트웨어 선정기법

전 혜 영[†] · 김 정 숙^{††} · 류 성 열^{†††}

요 약

OSS를 활용한 소프트웨어 개발의 사용영역이 날로 확대되고 있으며 기존의 OSS 선정기법에 관한 연구들은 이론적으로 잘 정립되어 있으나 실용적이지 못하며, 평가지표의 수가 많고 선정기법이 어려워 실제 적용하기에 비효율적이다.

본 논문에서는 이런 문제점을 해결하기 위해 선정평가지표를 최적화하고, 선정단계를 실용화하여 보다 쉽게 사용할 수 있는 선정기법을 제안한다. 기존의 연구를 바탕으로 공통으로 사용되고 있는 평가지표를 식별하고 배제와 그룹화를 통해 최종 14개의 평가지표와 31개의 평가항목을 추출하여 이해하기 쉽도록 재 정의한다. 그리고 정의된 평가지표를 초기평가(I), 기능평가(F), 특성평가(C), 보완평가(S) 등 4개 단계로 나누어 평가하는 단계적 선정기법을 제안한다.

제안한 OSS 선정기법의 적정성을 평가하기 위하여 OSS 사이트에서 MySQL 데이터베이스 관리 툴 29개를 후보로 선정하고 제안한 선정기법을 적용한 결과 평가시간의 감소되고 사용방법이 간결해졌다.

키워드 : 오픈 소스 소프트웨어, 선정기법, 평가지표

A Method of Selecting Open Source Software by IFCS

Hye-Young Jeon[†] · JING-SHU JIN^{††} · Sung-Yul Rhew^{†††}

ABSTRACT

The application field of OSS is being expanded farther and the existing studies are well established theoretically but not actually widely used. Because so many criteria and complexity of selection methods result in low efficiency and not proper for use.

In this master's thesis, to solve these problems I propose a selecting technique for OSS with less criteria, which is more effective and easy to use than before. Based on the existing studies, we identify assessment criteria which common used and redefine 14 evaluation attributes as well as 31 assessment criteria through elimination and grouping. And evaluate the criteria in four steps such as Initial assessment, Functional assessment, Characteristical assessment and Supplemental assessment.

To evaluate the adequacy of suggested technique, 29 database management tools for MySQL are selected as candidates, and applied to the technique I proposed. As the result, time and effort for evaluation is reduced sharply and we can use this technique more easily.

Keywords : Open Source Software, Selecting Technique, Assessment Criteria

1. 서 론

최근 OSS(OpenSource Software : 오픈소스 소프트웨어)는 무료 사용, 소스코드의 수정, 복제 및 자유로운 배포 등의 특징으로 여러 분야에서 활용되고 있다[1, 2]. 새로운 소프트웨어를 개발 시 저비용으로 효율성을 높일 수 있기 때문에 OSS를 활용하고 있고 시스템적인 측면에서는 상용 소

프트웨어를 대체하여 도입 및 유지보수에 투여되는 절차와 예산을 감소시키기 위해 OSS를 활용하고 있다.

여러 분야에서 폭넓게 OSS를 활용하다 보니 OSS를 활용할 수 있는 범위와 종류 또한 매우 많아졌다. 그래서 OSS를 사용하기 이전에 업무에 맞게 필요한 OSS를 선택하는 것도 매우 중요한데 이미 OSS를 활용하기 위한 선정기법이 다양하게 연구되고 있다[3-6]. 그러나 기존의 연구들은 이론적으로 잘 정립되어 있으나 평가지표의 수가 많고 선정기법이 어려워 실제 적용하기에 적합하지 않아 실용적이지 못하다는 문제점이 있다. 이에 OSS를 선정하기 위한 평가지표를 최적화하고 선정단계를 실용화하여 보다 쉽게 사용할 수 있는 선정기법이 필요하다.

본 논문에서는 기존의 연구를 바탕으로 OSS 선정에 사

* 본 과제는 정보통신산업진흥원의 SW공학 요소기술 개발과 전문인력 양성 사업의 결과물임을 밝힙니다.

† 정 회 원 : 숭실대학교 컴퓨터학과 박사과정

†† 정 회 원 : 숭실대학교 컴퓨터학과 공학석사

††† 중신회원 : 숭실대학교 정보과학대학원 원장

논문접수 : 2009년 12월 29일

수 정 일 : 1차 2010년 3월 2일

심사완료 : 2010년 3월 20일

용될 최소한의 평가지표를 추출하고 선정 단계를 실용화 하여 보다 효율적인 선정기법을 제안한다. 그래서 OSS를 선정할 때 전문가뿐만 아니라 일반 사용자도 쉽게 사용할 수 있도록 구성하였다.

2. 관련연구

2.1 Navicasoft의 오픈소스 성숙도 모델

2005년, Navicasoft 회사에서 개발한 OSMM(Open Source Maturity Model : 오픈소스 성숙도 모델)은 OSS의 성숙도를 평가하는 공식적인 프로세스이다[3]. 이 모델은 선정, 평가 및 구현 등 3개 단계로 구성되어 있고 소프트웨어, 기술지원, 문서화, 통합, 교육 및 전문서비스 등 6개의 평가 영역에서 19개 평가지표와 총 40개의 세부평가항목을 정의하였다[3]. 선정단계에서는 요구사항과 자원을 정의하고 평가 단계에서는 각 지표의 점수와 가중치를 부여하며 구현단계는 모든 점수를 합산하여 OSMM의 최종 점수를 계산한다.

이 모델은 OSS 개발조직을 위해 설계되어 비전문가들은 사용하기 어렵다. 또한 OSS 평가후보의 수가 많을 경우 평가 과정이 매우 복잡하여 효율성이 떨어진다.

2.2 Kim의 논리적 품질평가 모델

이 연구에서는 ISO/IEC9126을 기반으로 응용 오픈소스 소프트웨어 개발 및 유지보수에 활용할 수 있는 품질평가 모델을 제시하고, 정량적 품질평가 기법을 제시하였다[4]. 이 품질평가 모델은 OSS 품질을 기능성, 이식성, 유지보수성, 사용성, 합법성 및 성숙성 등 6개의 주특성과 하위 12개의 부특성으로 구성하였다. 평가는 각 지표에 대하여 가중치를 부여하고 최종 점수를 합산하는 방식으로 수행된다.

Kim의 논리적 품질평가 모델은 아직 초기 연구단계여서 평가기준이 명시되어 않아 실제로 OSS 선정평가에 적용하기에는 부족함이 있다.

2.3 QSOS 평가 모델

QSOS(Qualification and Selection of Open Source software) 모델은 아토스 오리진 회사에서 개발한 OSS를 선정하고 평가할 수 있는 모델로서 IT 조직에서 OSS를 활용하여 개발을 수행하는데 사용될 수 있다[5]. 이 모델은 주로 정의, 평가, 자격부여 및 선정 등 4개 부분으로 프로세스를 제시하였다. 정의단계에서는 평가를 진행하기 위한 일련의 지표와 활동을 정의하고 후보를 식별한다. 평가단계는 <표 1>과 같이 16개의 평가지표와 그 하위 44개 평가항목을 평가한다. 자격부여단계에서는 가중치를 부여하고 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.

<표 1> QSOS에서 정의한 OSS 평가지표

기능성	성숙성	적용성	개발 리더쉽
활성	독립성	부산물	품질보증
모듈화	문서화	패키징	이용가능성
서비스	라이선스	서비스	유지보수성

2.4 BRR 평가 모델

BRR(Business Readiness Rating)은 카네기 멜론과 인텔에서 공동으로 개발한 것으로 개발자뿐 아니라 일반 사용자가 OSS를 선정할 때 평가 할 수 있는 모델로 사용하기 위해 개발하였다[6]. 이 모델에서는 평가를 크게 4단계로 정의하였다. 첫 단계는 라이선스, 표준, 참조, 구현언어, 관련도서 등 기준에 의해 평가할 후보를 준비한다. 2단계에서는 12개 카테고리(<표 2> 참조)에 대해 우선순위를 부여하여 가장 중요한 7개 또는 더 적은 수의 카테고리만 선정평가지표로 선정하고 이에 대해 가중치를 부여한다. 3단계는 각 평가항목의 가중치를 부여하고 평가를 진행한다. 4단계에서는 2, 3 단계에서 얻는 점수를 합산하여 최종 BRR 점수를 계산한다.

이 모델은 효율성을 위하여 12개 카테고리 중 7개 또는 이하의 지표를 평가하지만 어느 지표를 선정할 지에 관한 취사결정에 따른 문제점이 있고 가중치를 2번 부여함으로써 계산이 복잡해질 수 있다. 또한 이 모델의 일부 평가지표는 이해하기 어려운 문제점도 있다.

<표 2> BRR에서 정의한 OSS 평가 카테고리

평가 카테고리	설 명
기능성	기능이 사용자 요구사항에 얼마나 적합한가?
유용성	사용자 인터페이스가 좋은가? 사용하기 쉬운가? 설치하기 쉬운가? 배치, 유지보수하기 쉬운가?
품질	디자인, 소스코드, 테스트의 품질은 어떠한가? 에러 발생율은 어떠한가?
보안	보안성은 어떠한가? 안전한가?
성능	성능은 어떠한가?
확장성	대형 환경으로의 확장성은 어떠한가?
아키텍처	소프트웨어 아키텍처가 어떠한가? 모듈화, 적합성, 유연성, 확장성은 어떠한가? 통합하기 쉬운가?
지원	지원이 얼마나 잘 되어 있는가?
문서화	문서화가 잘 되어 있는가? 문서의 품질은 어떠한가?
적용	커뮤니티, 시장 및 생산에서 얼마나 적용되고 있는가?
커뮤니티	소프트웨어를 지원하는 커뮤니티의 활동은 어떠한가?
전문성	개발 프로세스와 프로젝트 조직의 전문성 정도는 어떠한가? 개발자들의 기술성은 어떠한가?

2.5 기타 OSS 선정에 대한 연구

앞의 4개의 OSS 선정평가 모델 외 Meng Huang[7], David A. Wheeler[8], Carolyn A. Kenwood[9], 그리고 Karin van den Berg[10] 등에서는 OSS 품질의 특성과 평가에 대하여 연구하였다. 본 논문에서는 이 몇 개의 연구에서 제시한 OSS 품질특성을 정리하고 선정평가지표를 수립하기 위해 사용한다.

3. OSS 선정평가지표의 최적화

본 장에서는 관련연구를 바탕으로 최적화 과정을 통해 본 연구에서 제안하는 선정기법에 사용되는 OSS 선정평가지표를 도출하였다. OSS 선정평가지표는 아래 선정평가지표의 수집, 최적화 및 최소화 등 3개 단계를 거쳐 도출된다.

3.1 OSS 품질특성에 따른 선정평가지표의 수집

최적화의 첫 번째 단계로 기존의 8개 연구문헌을 기반으로 <표 3>에 나타난 것과 같이 총 60개의 OSS 품질특성을 식별하였다.

3.2 OSS 선정평가지표의 최적화

3.2.1 공통 품질특성의 선정

이 단계에서는 앞 단계에서 수집된 선정평가지표들 중에서 2개 이상의 문헌에서 공동으로 언급된 품질특성을 본 연구에서 사용될 선정평가지표로 선정한다. 이러한 이유는 서

로 다른 OSS를 평가함에 있어서 공통성을 찾기 위해서이며, 기타 선정되지 않은 특성들이 필요 없다는 의미는 아니다. 이 단계를 거친 결과 총 45개의 품질 특성이 선정되었다.

3.2.2 프로젝트의 특성에 따른 요소 배제

이 단계에서는 “비용” 지표를 배제하였는데 그 이유는 다음과 같다. 기존의 연구들에서는 무료로

사용할 수 있는 OSS와 사용 소프트웨어를 비교분석하였으나 본 연구에서는 OSS를 선정범위로 하였다. OSS도 일정 금액을 요구하는 제품이 있으나 대다수 OSS는 무료로 사용할 수 있으며 또한 제5장 사례연구에서도 도입비용이 들지 않는 OSS를 선정범위로 하였으므로 “비용” 지표를 제외로 하였다. 이 단계를 거쳐 총 44개의 OSS 품질특성이 선정되었다.

3.3 그룹화에 의한 선정평가지표의 최소화

이 단계는 앞 단계에서 얻은 품질특성들을 분석한 결과 성질이 유사한 품질특성을 그룹화하거나 새로운 유형으로 재분류함으로써 선정평가지표를 최소화하여 44개의 품질특성은 14개로 재 정의하였다.

이 부분은 앞서 정의한 14개 품질특성에 의하여 선정평가지표를 수립하는 과정으로서 평가지표와 세부평가항목에 대한 도출과정만 설명하고 실제 평가방법은 제4장에서 설명한다. 최종적으로 <표 4>와 같이 14개의 선정평가지표와 그 하위 31개 세부평가항목을 정의하였다.

<표 3> 기존 연구에서 식별된 OSS 품질특성

품질속성	연구문헌							
	OSMM	QSOS	Wheeler	Huang	Kenwood	Kim	Berg	BRR
1	기능성	*	*	*	*	*	*	*
2	기능적합성	*	*	*	*	*	*	*
3	비기능적합성	*	*	*	*	*	*	*
4	정확성	*	*	*	*	*	*	*
5	상호운용성	*	*	*	*	*	*	*
6	보안성	*	*	*	*	*	*	*
7	신뢰성	*	*	*	*	*	*	*
8	모듈화용성	*	*	*	*	*	*	*
9	복구성	*	*	*	*	*	*	*
10	성숙성	*	*	*	*	*	*	*
11	사용성	*	*	*	*	*	*	*
12	미해기능성	*	*	*	*	*	*	*
13	사용 용이성	*	*	*	*	*	*	*
14	운용성	*	*	*	*	*	*	*
15	지원	*	*	*	*	*	*	*
16	커뮤니티 지원	*	*	*	*	*	*	*
17	유료 지원	*	*	*	*	*	*	*
18	교육지원	*	*	*	*	*	*	*
19	개발지원	*	*	*	*	*	*	*
20	개발도구지원	*	*	*	*	*	*	*
21	성능	*	*	*	*	*	*	*
22	유사모수성	*	*	*	*	*	*	*
23	분석가능성	*	*	*	*	*	*	*
24	변경가능성	*	*	*	*	*	*	*
25	견고성/안정성	*	*	*	*	*	*	*
26	시험가능성	*	*	*	*	*	*	*
27	미작성	*	*	*	*	*	*	*
28	적용성	*	*	*	*	*	*	*
29	결지 용이성	*	*	*	*	*	*	*
30	공존성	*	*	*	*	*	*	*
31	컴포넌트 호환성	*	*	*	*	*	*	*
32	버전 호환성	*	*	*	*	*	*	*
33	확장성	*	*	*	*	*	*	*
34	문말의 위법	*	*	*	*	*	*	*
35	커뮤니티	*	*	*	*	*	*	*
36	시장 점유	*	*	*	*	*	*	*
37	수명	*	*	*	*	*	*	*
38	기술성	*	*	*	*	*	*	*
39	응용성	*	*	*	*	*	*	*
40	합법성(법적/라이선스)	*	*	*	*	*	*	*
41	비용	*	*	*	*	*	*	*
42	정책	*	*	*	*	*	*	*
43	버전배포 활성	*	*	*	*	*	*	*
44	문서화	*	*	*	*	*	*	*
45	사용자 공헌 프레임워크	*	*	*	*	*	*	*
46	사용자 문서	*	*	*	*	*	*	*
47	개발자 문서	*	*	*	*	*	*	*
48	관련 도서	*	*	*	*	*	*	*
49	웹 포스트	*	*	*	*	*	*	*
50	통합성	*	*	*	*	*	*	*
51	모듈화	*	*	*	*	*	*	*
52	표준	*	*	*	*	*	*	*
53	패키징	*	*	*	*	*	*	*
54	소프트웨어 요구사항	*	*	*	*	*	*	*
55	목표와 기원	*	*	*	*	*	*	*
56	이커넥터	*	*	*	*	*	*	*
57	third-party plug-ins	*	*	*	*	*	*	*
58	공용 API/외부 서비스	*	*	*	*	*	*	*
59	참조 가능한 적용	*	*	*	*	*	*	*
60	전문성	*	*	*	*	*	*	*

<표 4> OSS 선정평가지표 및 평가항목

평가지표	평가항목	설명
합법성	라이선스	라이선스를 체크하여 합법적으로 사용 가능 여부
기능성	기능 일치정도	제공하는 기능이 요구하는 기능과 일치한 정도
커뮤니티	수명	등록 날짜부터 사용 가능한 기간
	버전배포	버전배포 주기와 수량
	시장점유율	다운로드 수
	커뮤니티 활성	최근 활동 상황과 활약 정도
신뢰성	오류 대응	오류를 대응하는 방식
	오류의 발생, 복구	오류가 발생하고 복구되는 능력을 평가
지원	서비스 유형	지원하는 서비스의 유형을 평가
	메일링 리스트	커뮤니케이션을 진행하는 형식
	전문성	지원조직의 전문성
보안성	보안 서비스	보안관련 기능 또는 서비스 등 제공여부
문서화	문서종류	제공하는 문서내용의 종류
	문서형식	제공하는 문서 포맷의 유형
사용성	설치 용이성	설치에 드는 노력과 관련 요소
	사용용이성	기능 수행에 드는 노력과 관련 요소
	시험용이성	사용자를 위한 테스트를 제공하는 능력
성능	성능 테스트	성능 테스트를 수행하여 성능에 대한 품질 제공 여부
이해 가능성	기능	제공된 기능을 식별하는 능력
	인터페이스	인터페이스의 이해용이 정도
	문서	문서에 대한 이해용이 정도
	소스코드	소스코드에 대한 이해용이 정도
유지 보수성	분석가능성	결함이나 장애원인을 진단하거나 수정될 부분을 식별하는 능력
	변경가능성	문제해결을 위하여 소스코드를 변경할 수 있는 능력
적용성	표준 적용	적용한 표준이 있는지 여부
	참조가능한 적용	유명 브랜드회사 또는 시스템에 의한 적용, 관련도서, 뉴스보도 등에 의한 참조 여부
이식성	이식용이성	OSS가 환경변화에 따라 이식하기 쉬운 정도
개발성	모듈화	OSS(소스코드)가 개개의 기능에 의하여 모듈로 구분되는 능력
	통합용이성	개발언어, 플랫폼, 프레임워크 등 개발요소들의 통합 용이 정도
	코드 주석	소스코드에 대한 설명을 명시한 정도
	확장성	시스템 크기와 사용자 수의 증대에 유연하게 대응하는 능력

4. IFCS에 의한 OSS 선정기법

본 장에서는 3장에서 추출하고 정의한 지표를 적용한 선정기법을 제안하고 평가를 수행하기 위한 활용방안을 제시한다.

4.1 IFCS에 의한 OSS 선정기법의 설계

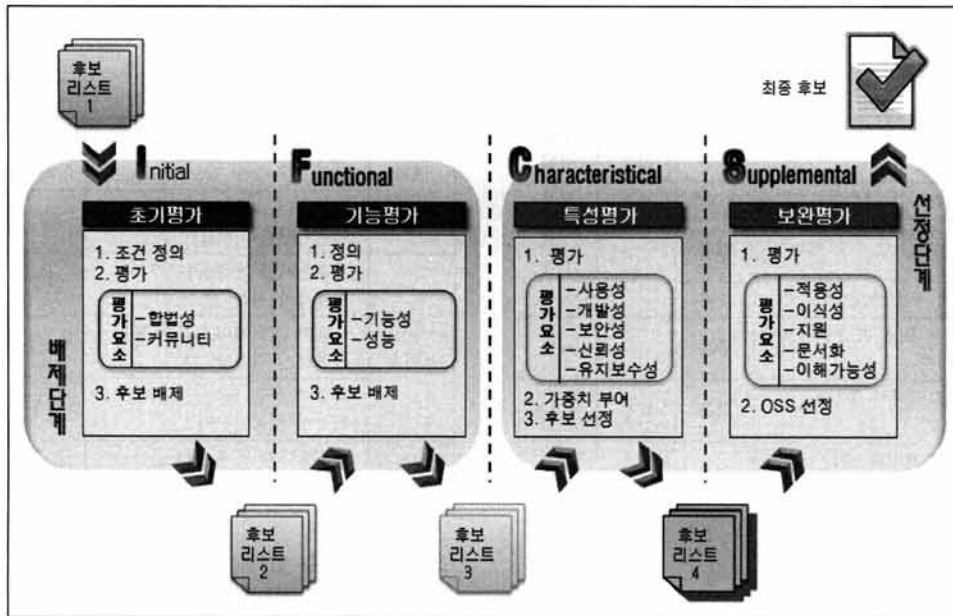
OSS 후보를 처리하는 측면으로 후보배제와 후보선정 단계로 나눌 수 있고 평가지표를 적용하는 측면으로 초기평가(Initial assessment), 기능평가(Functional assessment), 특성평가(Characteristical assessment) 및 보완평가(Supplemental assessment) 등 4개 단계로 정의하였다. 이렇게 설계한 이유는 후보자에 평가지표를 단계별로 적용하여 평가하고 조건에 부합되지 않는 후보를 배제하거나 또는 조건에 부합되는 후보를 선정함으로써 선정평가의 효율성을 높이려는 것이다.

4.2 초기평가(Initial assessment) 단계

초기평가는 OSS 선정의 첫 번째 단계로서 초기단계에 사용자가 정의한 조건에 부합되지 않는 후보를 배제하고 믿음직한 평가환경을 구성하는데 목적이 있다. 초기평가는 주로 사용자 조건정의, 지표평가 및 후보배제 등 3개 활동과 절차로 수행되며 지표로는 합법성과 커뮤니티를 평가한다.

합법성평가에서는 일반적으로 많이 사용되고 있는 OSS 라이선스와 각 라이선스에서 규정한 사항들을 기술한 합법성 평가표<표 5>를 사용할 수 있다. 오픈소스 프로젝트는 소프트웨어와 이를 개발, 유지해나가는 커뮤니티로 구성되는데 커뮤니티의 품질은 OSS의 품질을 반영한다. 커뮤니티가 활발한 OSS는 사용자에게 보다 높은 믿음성을 주므로 초기에 커뮤니티가 가장 활발하지 않은 후보를 배제하여 믿음직한 선정환경을 조성한다.

사용자는 <표 6>에 제시한 평가항목에 대하여 조건을



(그림 1) IFCS에 의한 OSS 선정기법

<표 5> 합법성 평가 활용방안

구분	GPL	LGPL	BSD	Apache	MPL
무료사용	O	O	O	O	O
자유배포	O	O	O	O	O
코드공개	O	O	O	O	O
코드수정	O	O	O	O	O
코드공개	O	O	X	X	O

<표 6> 커뮤니티 평가 활용방안

평가항목	사용자 정의조건
수명	
시장점유율	
커뮤니티 활성	
버전 배포	

정의하고 이런 조건에 근거하여 부합되지 않는 후보를 배제한다.

4.3 기능평가(Functional assessment) 단계

기능평가는 OSS 선정의 두 번째 단계로서 초기평가에서 얻은 후보리스트(2)를 입력물로 기능과 성능에 대하여 평가한다. OSS가 아무리 훌륭하다 해도 사용자가 요구하는 기능을 수행하지 못한다면 아무런 의미도 없게 된다. 그리고 기능을 수행하는 정도는 성능을 통하여 나타낼 수 있으므로 성능이 매우 낮은 후보도 이 단계에서 미리 고려하여 배제하여야 한다.

본 논문에서 기능성이란 사용자가 정의한 기능과의 일치 정도로서 사용자가 필요한 기능을 정의<표 7>하고 후보들

<표 7> 기능성 평가 활용방안

평가항목	사용자 정의조건
기능 일치성	1. 필요한 기능을 정의하시오 2.

이 수행할 수 있는 기능을 정의기능과 대조하면서 적합하지 않은 후보를 제거하도록 한다. 결과로 사용자 정의 기능의 50% 이상을 만족시키지 못하는 후보는 배제된다.

성능 지표는 OSS의 실행, 데이터 용량, 응답시간 등 성능을 평가하는데 이는 제품이 제공하는 기능을 얼마나 잘 수행하는지를 나타낼 수 있다. 일반적으로 성능 테스트는 전문적인 도구를 의뢰하므로 오픈소스 프로젝트에서는 성능 테스트를 수행하고 결과를 커뮤니티에 공개한다. 사용자는 프로젝트의 홈페이지에 성능에 대한 설명, 테스트 수행결과 등이 있는지 살펴보거나 커뮤니티에서 성능에 대한 사용 후기, 토론 등이 있는지 조사하여 성능 평가 정보를 참고할 수 있다.

4.4 특성평가(Characteristical assessment) 단계

특성 평가는 OSS 선정의 세 번째 단계이다. 특성 평가 단계에서는 OSS를 활용함에 있어서 사용자의 관점이 서로 다르고 OSS 특성에 대한 치중점이 다르기 때문에 이를 평가하기 위함이다. 예를 들면 OSS를 활용하여 재개발을 할 것인가 아니면 일반적인 응용인가, 높은 성능 특히 보안성을 위주로 선정할 것인가 아니면 신뢰성 또는 유지보수성을 중요시 하는가 등 사용자의 요구에 따라 평가의 결과가 달라진다. 특성평가는 지표평가, 가중치 부여 및 후보선정 등 3개 단계로 수행된다.

〈표 8〉 개발성 평가항목 및 평가기준

평가 항목	평가 기준	점수		
		0	1	2
모듈화	-OSS(소스코드)가 모듈로 구분되어 있는가?	구분 안됨	명확치 않음	구분 잘됨
	-모듈은 이해하기 쉬운가?	어려움	보통	쉬움
코드 주석	-주석이 있는가? 이해하기 쉬운가?	없음		이해 쉬움
통합 용이성	-개발언어에 종속적인가?	종속		독립
	-개발 플랫폼 또는 프레임워크에 종속적인가?	종속		독립
	-기타 프로그램과의 통합이 쉬운가?	쉬움	보통	어려움
확장성	-공용 API를 제공하는가? 확장개발을 지원하는가?	없음		있음

4.4.1 항목평가

지표평가에서는 개발성, 보안성, 사용성, 신뢰성 및 유지보수성 등을 평가하는데 각 지표의 평가항목에 대하여 (0) 점, (1)점, (2)점 등 3개 등급으로 평가한다.

• 개발성

OSS는 소스코드를 공개하는 특성상 이를 이용하여 일정한 라이선스 하에 재개발을 허용한다. 개발성은 OSS를 활용한 재개발에 영향을 주는 소스코드의 능력을 평가하는 것으로 모듈화, 코드 주석, 통합용이성 및 확장성 등 항목을 평가한다.

• 보안성

OSS는 소스공개, 공동개발 등 특성으로 하여 보안성에 대한 논의가 많아지고 있다. 특히 안전성을 중요한 품질로 요구하는 OSS 활용에 있어서는 보안성에 대한 평가를 진행해야 하며 보안과 관련된 서비스를 제공하여 보안상의 문제가 발생했을 경우 이를 발견, 기록하고 추적 및 해결하는 능력이 있는지 여부를 평가한다.

〈표 9〉 보안성 평가항목 및 평가기준

평가 항목	평가 기준	점수		
		0	1	2
모듈화 및 코드 주석	-보안을 위한 로그인 기능이 있는가?	없음		있음
	-보안버그 디렉토리가 있는가?	없음		있음
	-보안 관련 문제를 해결하는 서비스를 제공하는가?	없음		있음

• 사용성

사용성은 사용자가 OSS 사용하는데 필요한 노력의 정도를 평가한다. 사용에 필요한 노력은 소프트웨어의 설치에서부터 직접적인 사용까지 포함한다. OSS는 이런 노력을 감소할 수 있는 도움말, Q&A를 제공할 수 있고 테스트 케이

〈표 10〉 사용성 평가항목 및 평가기준

평가 항목	평가 기준	점수		
		0	1	2
설치 용이성	-설치문서가 제공되는가?	없음		있음
	-설치/제거하기 쉬운가?	어려움	보통	쉬움
	-기타 프로그램과의 의존성은 어떠한가?	종속		독립
사용 용이성	-사용문서를 제공하는가? 이해하기 쉬운가?	없음	이해 어려움	알기 쉬움
	-사용자 인터페이스의 메뉴는 사용하기 쉬운가?	어려움	보통	쉬움
	-사용 도움말 또는 Q&A 기능이 제공되는가?	없음		있음
시험 용이성	-테스트 케이스를 제공하는가?	없음		있음
	-테스트는 수행하기 쉬운가?	어려움	보통	쉬움

스를 제공할 수 있다.

• 신뢰성

신뢰성은 OSS가 정상적인 성능을 유지하는 능력으로서 일반적으로 오류/버그가 발생하는 확률 및 이를 복구하는 능력으로 신뢰성을 평가한다. 오픈소스 프로젝트에서는 소프트웨어 오류 등 문제를 등록하고 추적, 해결하는 Bugs 시스템을 제공하거나 포럼, 위키 등을 이용하여 문제를 토론하고 해결하는 방식을 채택하고 있다.

〈표 11〉 신뢰성 평가항목 및 평가기준

평가 항목	평가 기준	점수		
		0	1	2
오류 대응	-Bugs 시스템을 제공하는가?	없음		있음
	-오류를 대응하기 위한 게시판, 포럼, 위키 등을 제공하는가?	없음		있음
오류발생 및 복구	-오류보고 및 수정활동이 활발히 진행되는가?	없음	활발치 않음	활발함

• 유지보수성

OSS의 유지보수란 소프트웨어를 최적의 상태로 활용, 유지하기 위하여 기술을 지원하고 운영 시 발생하는 오류 및 결함 등 문제를 해결하며, 추가 또는 변경되는 사용자 요구사항을 만족시키는 기능과 특징을 개발하는 것이다. 이러한 활동을 위하여 OSS가 분석과 삭제 등 변경이 용이해야 한다.

4.4.2 가중치부여 및 선정

OSS는 활용하는 사용자의 목적에 따라 이를 평가하는 시점이 다르기 때문에 어떤 지표를 더 중요시 하는가 하는 가중치 부여가 필요하다. 본 논문에서는 5개 지표에 대하여 각각 우선순위를 부여하고 가중치를 부여한 다음 매개 지표의 점수를 합산하여 총점수를 얻고 총점수가 상위인 5개의 후보(후보리스트(4))를 선정한다.

<표 12> 유지보수성 평가항목 및 평가기준

평가 항목	평가 기준	점수		
		0	1	2
분석 가능성	-문제영역(코드, 문서 등)을 쉽게 식별할 수 있는가?	어려움	보통	쉬움
	-유지보수를 위한 전문가조직이 있는가?	없음		있음
변경 가능성	-변경문서를 제공하는가?	없음		있음
	-문제영역만 독립적으로 변경하기 쉬운가?	어려움	보통	쉬움

4.5 보완평가(Supplemental assessment) 단계

보완평가단계에서는 특성평가단계를 통해 얻은 5개 후보에 대하여 적용성, 이식성, 지원, 문서화 및 이해가능성 등 지표의 보완평가를 수행한다. 이 5개 평가지표는 비슷한 후보들에서 공통으로 필요하면서도 미세한 차이로 인해 평가 결과가 달라진다.

• 적용성

일반적으로 OSS가 표준을 사용하고 유명한 브랜드회사 또는 시스템에 의해 적용되거나 참조되면 믿음성이 높다고 볼 수 있다. 그리고 관련된 도서 또는 뉴스 등에 대한 참조도 간접적으로 OSS를 평가할 수 있다.

• 이식성

이식성은 OSS가 한 환경에서 다른 환경으로 이동할 수 있는 능력을 평가하는데 사용자 또는 개발자의 능력과 실제 환경이 서로 다르므로 이식성이 좋아야 한다.

• 이해가능성

OSS는 이해하기 쉬워야 한다. 무료이고 기능이 좋고 해도 이해하기 어렵다면 훈련에 드는 노력과 비용이 높아지고 일반 사용자는 이를 쉽게 사용하지 못하게 된다. 이 지표는 OSS의 기능, 인터페이스 및 문서에 대한 이해정도를 평가한다.

• 지원

이 지표는 오픈소스 프로젝트에서 개발자 또는 사용자를 위하여 여러 가지 서비스를 제공하는 능력을 평가한다. 서비스를 제공하는 것은 무료인 OSS의 주요한 수익창출의 정책이다. 서비스 유형은 비용측면으로 유료지원과 무료지원으로 나눌 수 있고 서비스 내용으로는 기술지원, 교육지원 등이 있다. 메일링 리스트는 커뮤니케이션의 수단으로 이를 제공하는지 여부와 활성화 되었는지 여부가 지원능력을 평가할 수 있다. 또한 메일링 리스트가 분야별 구성은 서비스를 제공하는 조직의 전문성도 평가할 수 있다.

• 문서화

초기의 OSS는 문서가 없거나 또는 잘 되어 있지 않았으

<표 13> 보완평가 평가기준 및 활용방안

평가 항목	평가 기준	후보				
		A	B	C	D	E
적용성	-기타 유명브랜드 또는 기사에서 적용하거나 참조하고 있는가?					
	-품질 또는 기타 관련 표준을 적용하였는가?					
	-출판된 관련 도서가 있는가?					
이식성	-독립적으로 실행될 수 있는 응용프로그램인가?					
	-다양한 운영체제 환경에서 인식되는가?					
	-개발언어에 종속되지 않고 이식 가능한가?					
이해 가능성	-Screenshot 또는 데모를 제공하는가?					
	-인터페이스의 메뉴가 알기 쉬운가?					
	-제공된 문서가 알기 쉬운가?					
	-프로그램은 영어 이외 기타 언어 실행이 가능한가?					
지원	-포럼, 위키 등 커뮤니티를 제공하는가?					
	-기술지원 또는 교육지원을 제공하는가?					
	-메일링 리스트를 제공하는가?					
	-메일링 리스트가 분야별로 구성되어 있는가?					
문서화	-최근 1년간 커뮤니티 활동이 활발하게 진행되고 있는가?					
	-설치, Help 등 사용자 문서를 제공하는가?					
	-개발자 문서를 제공하는가?					
	-문서는 영어 이외 기타 언어를 지원하는가?					
	-txt 이외 기타 문서형식을 제공하는가?					

므로 기존의 연구에서는 이 지표를 중요한 특성으로 간주하지 않았다. 그러나 OSS가 발전하면서 문서화도 품질속성의 한가지로 되었다. OSS에서 어떤 문서가 어떤 형식(html, xml, txt 등)으로 제공하는지 하는 것을 문서화 지표에서 평가한다.

보완평가는 <표 13>에서 제시한 평가표를 사용하여 체크하는 방식으로 수행된다. ● 는 해당된다는 의미고 ○는 해당되지 않음을 의미한다.

5. 사례연구

본 논문에서 제안하는 IFCS에 의한 다단계 선정기법의 효율성을 검증하기 위하여 사례연구를 이론적 증명과 설문 조사를 통하여 제안한 선정기법이 기존의 연구보다 쉽고 효율적이었음을 보여주었다.

최근 개인 또는 중소기업은 DB2, Oracle 등 고가의 데이터베이스 대신 오픈소스 데이터베이스인 MySQL를 채택하여 사용하면서 이를 관리하는 오픈소스 툴들이 많이 개발되고 있다. 본 논문에서는 MySQL의 관리 툴을 선정하는 사

례연구로 하였다. 본 사례에서의 MySQL 관리 툴은 무료로 사용할 수 있어야 하고 리눅스나 윈도우 등 다양한 운영체제에서 사용 가능해야 하며 PHP로 개발되어야 한다. OSS의 후보군은 오픈소스 커뮤니티 사이트인 소스포지에서 'MySQL', 'Front-ends', 'Database management tool', 'PHP' 등 조건을 사용하여 총 29개의 후보를 식별하였다.

5.1 IFCS에 의한 다단계 선정기법의 적용 및 결과

5.1.1 초기평가

본 논문에서는 OSS의 라이선스를 GPL, LGPL, BSD, Apache 및 MPL로 규정하고 평가한 결과 MyQueery Builder, tbvs, Isolde 등 3개 후보가 배제되었다. 그리고 <표 14>와 같이 조건을 정의하고 평가를 수행하여 ALUMNI, PHPDBB, Adminature 등 9개의 후보를 배제하였다.

<표 14> 커뮤니티평가 사용자 정의

평가항목	사용자 정의 조건
수명	최소1년이상 수명, 현재까지 활약(다운로드 가능함) 되어야 함
시장점유율	다운로드 회수가 500번 이상이어야 함
커뮤니티 활성	활약 정도는 60%이상이어야 함 최근 1년 이내 커뮤니티 활동이 존재해야 함
버전배포 상태	최근 5년간 버전 업데이트가 있어야 함

5.1.2 기능평가

기능평가에서는 <표 15>와 같이 10개의 기능요구사항을 정의하고 5개 이상 기능을 수행하지 못하는 ezradius, MyAntarMuka, eSQueLIA, Mailadmin 및 Polymer, phpMyIDE 등 6개의 후보를 배제하였다.

<표 15> 기능성 요구사항 정의(일부)

평가 항목	사용자 정의 조건
기능 일치성	1. DB, 테이블, 뷰, 인덱스 및 테이블의 데이터를 열람할 수 있어야 함 2. DB, 테이블, 인덱스 그리고 데이터를 생성, 복사, 수정, 삭제할 수 있어야 함 3. CSV와 SQL로부터 데이터를 도입 가능해야 함 4. 데이터를 Excel, Word 또는 SQL방식으로 출력(Export) 가능해야 함

5.1.3 특성평가

특성평가는 앞의 두 개 단계의 평가를 거쳐 남은 11개의 후보에 대해 제안한 선정평가지표와 절차에 따라 평가를 수행한 결과 <표 16>에 나타난 것과 같다. 본 연구에서는 사용성(30%), 유지보수성(25%), 보안성(20%), 신뢰성(15%) 그리고 개발성(10%)로 가중치를 부여하였다.

특성평가를 진행한 결과 11개 후보 중 총점수가 높은 순으로 phpMyAdmin, MySQL Control Center, Adminer, PHP Mini SQL Admin, alc MySQL-Admin 등 5개의 후보를 선정하였다.

<표 16> 특성평가의 결과

후보 OSS		점수
1	phpMyAdmin	7.25
2	phpMyEdit	5.20
3	MySQL Sidu	4.60
4	MySQL Control Center	6.70
5	Vfront	5.55
6	Graphical Botanical Database Management	5.80
7	MySQL-Admin	6.40
8	eSKUel	5.25
9	DataBrowser	5.70
10	Adminer(phpMinAdmin)	6.95
11	PHP Mini SQL Admin	6.65

5.1.4 보완평가

특성평가에서 선정된 5개의 후보에 대하여 보완평가를 수행하여 phpMyAdmin이 평가가 제일 높았지만 Adminer과 MySQL Control Center도 사용제품으로 대체하기에 괜찮은 후보이다. 결과는 <표 17>에서 보여준 것과 같다.

<표 17> 보완평가의 결과(일부)

평가지표	평가 기준	후보				
		A	B	C	D	E
적용성	기타 유명 브랜드 또는 기사에서 적용하거나 참조하고 있는가?	●	●	●	○	○
	품질 또는 기타 관련 표준을 적용하였는가?	○	○	○	●	●
	출판된 관련 도서가 있는가?	●	○	○	○	○
이식성	독립적으로 실행될 수 있는 응용 프로그램인가?	○	●	○	●	○
	다양한 운영체제 환경에서 인식되는가?	●	●	●	●	●
	개발언어에 종속되지 않고 이식 가능한가?	●	●	●	○	●

5.2 기존 선정기법과의 비교 분석

본 논문에서는 실제 선정사례를 통하여 제안한 선정기법이 기존의 연구보다 효율적이라는 결과를 얻었다. 이를 검증하기 위하여 아래와 같은 가설을 정하고 이론적인 증명을 하였다. 후보를 29개로 정하고 BRR의 경우 약 34개 항목을 평가한다고 가정한다<표 18>.

비록 이론적 가설이지만 실제로 너무 많은 평가를 진행하는 것은 시간과 노력이 많이 들것은 물론이고 더욱이 사용자의 반감을 초래할 수 있다.

또한 본 논문에서 정의한 평가지표들이 이해하기 쉽고 사용하기 용이하다는 것을 증명하기 위하여 OSS 사용자 25명을 대상으로 설문조사를 진행하였다. 설문조사는 크게 2개

<표 18> 선정기법에 따른 평가지표 및 항목 비교

	OSMM	QSOS	BRR	IFCS
평가지표/평가항목	6/40	16/44	20/34	14/31
총 평가회수	1160	1276	986	585

〈표 19〉 설문조사 결과

설문	문항	인원(명)
사용자 수준조사	OSS 개발자로서 커뮤니티에서 활약	3
	OSS를 사용하여 개발한 적 있다	4
	일반 응용으로 사용한다	18
기존 선정기법	매우 쉽다	0
	쉽다	0
	보통	6
	어렵다	10
	매우 어렵다	9
제안한 선정기법	매우 쉽다	0
	쉽다	5
	보통	14
	어렵다	3
	매우 어렵다	3

부분으로 구성되는데 한 부분은 사용자의 수준과 기존의 선정기법에 대한 이해정도를 조사하고 다른 한 부분은 본 논문에서 제안하는 선정평가지표와 해당 설명에 대한 이해정도를 조사하였다. 설문 조사결과 <표 19>에서 보여준 것과 같이 본 논문에서 제시한 선정평가 지표가 비교적 이해하기 쉽다는 결론을 얻을 수 있었다.

6. 결론 및 향후연구

최근 OSS가 널리 사용되면서 이를 활용하기 위한 다양한 연구들이 수행되어 왔다. 그러나 현재까지 OSS를 선정하는 표준이 정립되어 있지 않고 나름대로 정의하여 사용하고 있다. 특히 기존에 제안되었던 여러 선정기법은 효율성이 낮거나 범용성이 약하거나 또는 실제 평가를 수행하기 어려운 등 문제점이 있다.

본 논문에서는 이런 문제점을 해결하기 위하여 기존 연구를 바탕으로 OSS 평가지표를 최소화하여 재정립하고 이를 IFCS 등 4가지 평가활동에 의하여 단계적으로 적용하는 선정기법을 제안하였다. 즉, 제안한 14개 평가지표와 그 하위 31개 평가항목을 초기평가, 기능평가, 특성평가 및 보완평가 등 4개 단계로 나누어 적용하였다.

IFCS에 의해 OSS 선정기법을 적용한 MySQL 데이터베이스 관리 툴을 선정하는 사례연구를 수행한 결과 본 연구의 선정기법은 기존의 연구보다 효율적이고 이해하기 쉽다는 장점이 있었다.

본 논문에서의 결과는 특정사례를 수행한 결과로서 이를 모든 OSS 선정에 적합하다고 말하기에는 아직 무리가 있다. 향후 다양한 OSS의 평가대상을 포함하여 지속적인 사례연구를 바탕으로 선정기법을 개선해 나가고 이를 기반으로 IFCS의 유용성과 효율성을 검증해 나가도록 할 것이다.

참고 문헌

- [1] 송위진, "오픈소스 소프트웨어의 기술혁신 특성 리뷰", 한국기술혁신학회지, 제5권, 제2호, 2002.
- [2] 한국소프트웨어진흥원, '공개소프트웨어 도입가이드라인 연구', 2003.
- [3] Golden, 'The Open Source Maturity Model: OSMM', Navicasoft, 2004.
- [4] 김지혁(Kim, Ji-Hyeok), 류성열, "응용 오픈소스 소프트웨어 특성에 적합한 논리적 품질평가 모델에 관한 연구", The KIPS transactions. Part D, Vol.16, No.1, 2009, pp.73-82.
- [5] Atos Origin, 'Method for Qualification and Selection of Open Source software: QSOS,' 2004.
- [6] Intel, SpikeSource, 'The Business Readiness Rating model: An evaluation framework for open source,' 2004.
- [7] Meng Huang, 'Li guang Yang, and Ye Yang, A Development Process for Building OSS-Based Applications,' LNCS 3840, pp.122-135, 2005.
- [8] David a Wheeler, 'How to Evaluate Open Source Software Programs,' 2008.
- [9] Carolyn A. Kenwood, 'A Business Case Study of Open Source Software,' The MITRE Corporation, 2004.
- [10] Karin van den Berg, 'Finding Open options-An Open Source software evaluation model with a case study on Course Management Systems,' Tilburg University, 2005.



전혜영

e-mail : jeonhye0@ssu.ac.kr

2008년 숭실대학교 정보과학대학원 소프트웨어공학(공학석사)

현 재 숭실대학교 컴퓨터학과 박사과정

2009년~현 재 성결대학교 컴퓨터학과 강사

2001년~2009년 한국 Sun Microsystems

책임 컨설턴트 및 전임강사

관심분야: 오픈소스 소프트웨어, 소프트웨어 아키텍처, 개발 방법론, 프레임워크 등



김정숙

e-mail : avile0706@naver.com

2007년 중국 목단강사범대학교 교육기술학과(공학사)

2010년 숭실대학교 컴퓨터학과(공학석사)

관심분야: 오픈소스 소프트웨어, 요구공학 등



류 성 열

e-mail : syrheo@ssu.ac.kr

1981년~현 재 송실대학교 컴퓨터학부 교수

1982년~1995년 송실대학교 전자계산원 원
장 중앙전자계산소 소장

1997년~1998년 George Mason University
객원교수

1998년~2001년 송실대학교 정보과학대학원 원장

관심분야: 소프트웨어 요구공학, 소프트웨어 유지보수, 오픈소스
소프트웨어 등