

오픈소스 소프트웨어 활성화를 위한 성숙도 및 적용성 평가 모델(OSMAAM)의 설계 및 구현에 관한 연구

정 윤 재*, 오 승 운**, 김 형 채***, 박 진 흥****, 김 근 동*****

요약 본 연구는 오픈소스 소프트웨어(OSS, Open Source Software)의 실질적인 활용을 증대시키기 위해 성숙도뿐 아니라 적용성까지 평가하는 모델과 구현 방법을 제시한다. 이 모델은 기존 모델에서 평가될 수 없었던 적용성을 측정하기 때문에, OSS를 신규 도입하거나, 상용 소프트웨어를 OSS로 대체, 또는 OSS를 상용 소프트웨어와 비교 시 활용할 수 있도록 사용자들을 지원한다. 이 모델의 설계 및 구현을 위해 7단계의 방법론을 제시하였고, 설문을 실시하여 모델의 타당성과 신뢰성을 검증하였다. 또한, 이 방법론을 기반으로 인프라 부분에서 OSS에 대한 성숙도와 적용성을 평가하는 실제적인 예를 제공함으로써 이 모델의 실용성을 입증하였다. 향후 일반 사용자뿐만 아니라 정보화 담당자들까지도 산업적인 적용을 위해 최적의 OSS를 이 모델을 활용하여 선정할 수 있을 것이다. 따라서 본 연구의 결과가 OSS 활성화 정책에 크게 기여할 것으로 기대된다.

주제어: 오픈소스 소프트웨어(OSS), OSS 성숙도 평가 모델, OSS 속성

A Study for Design and Implementation of Open Source Software Maturity and Applicability Assessment Model(OSMAAM) for OSS Proliferation

Jung, Yunjae, Oh Seung Woon, Kim Hyeong Chae, PARK JIN HONG, Kim Keun Dong

Abstract This study presents a model and its implementation methodology assessing not only maturity but also applicability in order to increase the practical usage of open source software(OSS). Since this model measures applicability which can not be assessed by the previous models, it is able to support users in many ways when they introduce OSS for the first time, replace commercial software with OSS, and compare OSS with commercial software. For the purpose of design and implementation of this model, we presented a methodology with 7 phases and conducted a survey to evaluate the appropriateness and reliability of the model. In addition, we proved the practical usability of the model by providing a practical example assessing maturity and applicability of OSS in the area of the infra category on the basis of the methodology. In the future, not only general users but also officers in charge of informatization will be able to select optimal OSS by using this model for industrial application. Therefore, the result of this research is expected to greatly contribute to OSS promotion policy.

Keywords: OSS, OSS maturity assessment, OSS maturity assessment method

2012년 11월 19일 접수, 2012년 11월 20일 심사, 2013년 2월 22일 게재확정

* 삼성SDS 오픈소스기획그룹 수석(yunjae.jung@samsung.com)

** 제2저자 겸 교신저자, 한국정보화진흥원 책임연구원(kmanager@nia.or.kr)

*** 아비도스 이사(hckim@abydos.co.kr)

**** 정보통신산업진흥원 책임연구원(jhpark1@nipa.kr)

***** 락플레이스 이사(kdkim@rockplace.co.kr)

I. 서론

오픈소스 소프트웨어(OSS, Open Source Software)¹⁾는 품질과 규모 측면에서 발전을 거듭하여 지금은 IT 전반적인 영역에서 없어서는 안 되는 중추적인 역할을 담당하고 있다(한국소프트웨어진흥원, 2007). 더욱이 최근에 들어서는 임베디드, 모바일, 클라우드 컴퓨팅, 빅 데이터 영역에서도 OSS가 핵심적인 리더십을 발휘하고 있다(Wong, 2011; Campbell, 2011; Bennett, 2011; Lurie, 2012). 그러나 OSS가 이처럼 오랜 시간에 걸쳐 꾸준히 발전하였음에도 불구하고 상업적인 적용 사례 및 공공부분의 OSS 확대는 기대만큼 폭발적으로 증가하고 있지 않는 것으로 보인다(심화영, 2012). OSS의 신뢰성 부족, 전문인력의 부족 등이 이러한 현상의 주요한 요인으로 분석되기도 하였으나(권문주 외, 2008) 수많은 OSS 중에서 어떤 것이 알맞은 OSS인지, 적용 분야에 꼭 맞는다고 선별된 OSS가 상용 비공개소스 소프트웨어(CSS, Closed Source Software)에 비교해서 손색은 없는지 등 기초적인 정보의 부재도 문제가 되는 것이 사실이었다. 이러한 문제를 해결하기 위해서 지금까지 다양한 노력이 기울여졌으며, 그 결과 OSS 분류체계(한국정보통신기술협회, 2011)와 성숙도 모델이 등장하게 되었다(Rashid, 2011, Del Bianco, 2009; 김정숙, 2009; 김지혁, 2009; Wittmann 2009; 송지훈 2008; 이영민, 2008; Taibi, 2007; Atos Origin, 2006; SpikeSource, 2005; Fuggetta, 2003). 그럼에도 불구하고, 이러한 성숙도 모델의 실제적 활용이 미진하고 커뮤니티 활동으로 연결되지 못하는 상황²⁾은 이 모델들의 효과적인 사용이 어려운 현실을 반영하고 있는 것으로 보인다. 더구나 OSS 활성화를 위한

정책적 방향에서 성숙도 모델만 제시하는 것이(정보통신산업진흥원, 2010) 아닌 현 업무에 적용할 수 있는 OSS 적용 모델이 필요한 것이다.

본 연구는 기존 OSS 성숙도 모델의 문제를 실용성, 정확성, 공정성 측면에서 해결 하고자 한다. 실용성 관점에서는 OSS를 선택하고 적용하는데 실질적인 도움을 주지 못하는 것이 문제이다. 실제로 OSS를 단순히 개인적인 목적이나 단체 내부적인 용도를 넘어서 사업 제안과 프로젝트 실행 단계에서 CSS³⁾를 대체하거나 전환하는데 실용적 지표를 제공해 주는데 부족한 부분이 없지 않다. 정확성 관점에서는 측정의 오류를 낮출 수 있도록 평가 기여도에 따라 속성을 정의하고, 한명이 여러 차례 측정하거나 여러 명이 측정하더라도 동일한 결과가 나오도록 정확한 정량화 공식이 수립되어야 한다. 현실적으로는 속성 값을 찾을 수 없거나, 여러 개의 속성 값이 존재할 경우에는 unknown⁴⁾으로 처리할 수밖에 없는 경우가 많아 사실상 정확한 측정이 어렵게 된다. 공정성 관점에서는 속성의 정량화 방법과 가중치 할당이 투명하고 객관적이어야 한다. 그러나 일반적으로 정성적인 속성을 정량화시키는 방법이 애매하며, 속성의 중요도인 가중치를 할당하는데 있어서도 주관성을 배제하지 못하는 것이 근본적인 딜레마로 판단된다.

본 논문에서는 이런 문제를 해결하기 위해 기존의 성숙도 평가 모델의 한계를 극복한 OSS의 활용과 보급에 실용적 프레임워크(OSMAAM, Open Source Software Maturity and Applicability Assessment Model) 및 측정방안을 제시하고자 한다. 이를 기반으로 정보화담당자가 쉽게 OSS를 선택할 수 있도록 하여 OSS 활성화 정책에 도움이 되고자 한다.

1) OSS를 '공개소프트웨어' 라고도 하나 본 연구에서는 '오픈소스 소프트웨어' 로 용어를 통일함

2) OpenBRR (<http://openbrr.org/>) 등은 2012년 11월 현재 사이트가 중단됨

3) CSS는 OSS와 상반되는 개념으로 상용 소프트웨어를 지칭함

4) 정확한 값을 알 수 없으므로 Null이나 N/A로 표기함

II. 이론적 배경

OSS 성숙도 평가 모델은 OSS의 속성을 기술하고 평가 기준을 제시하여 전반적인 품질 수준을 측정하

는 프레임워크이다(Petrinja, 2010). 제품 자체적인 속성으로는 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성 등이 대표적이며 제품 외적 속성에는 커뮤니티의 활동성, 지속성 그리고 OSS의 라이선스

〈표 1〉 OSS 성숙도 평가 모델에 관련된 기존 연구 목록

No	연구자(연도)	모델명 또는 연구제목	평가 속성분류	계량화 평가
1	Frans-Willem Duijnhouwer, Chris Widdows(2003)	Open Source Maturity Model	4개 대분류 내 12개 인디케이터(Product indicators)와 15개 인디케이터(Application indicators)	1 to 5(점수)
2	MAMPU(2008)	(OSS) IMPLEMENTATION GUIDELINES	8개 영역별 원칙	원칙에 대한 설명만 제시
3	SpikeSource, Carnegie Mellon West, and Intel Corporation(2005)	Business Readiness Rating for Open Source(Open BRR)	11개 대분류, 28개 매트릭	1 to 5(점수)
4	Atos Origin (2006)	Method for Qualication and Selection of Open Source software (QSOS) v1.6	3개 정의와 3개 평가 카테고리 구성	정의 및 1 to 2 (점수)
5	Taibi Davide, Luigi Lavazza, and Sandro Morasca(2007)	OpenBQR	6개 대분류	각 분류별 가중치 포함한 다양한 점수 부여
6	Gökem çtin(2008)	A Measurement Based Framework for Assessment of Usability-Centricness of Open Source Software Projects	6개 카테고리, 21개 인디케이터	절대평가, Yes or No, 및 5점 스케일복합
7	송지훈 외(2008)	오픈소스 소프트웨어 특성을 고려한 소프트웨어 평가모델	8개 평가요소	없음
8	Wittmann Marion, Nambakam Ranga(2009)	Quality Platform for Open Source Software(Qualipso) version 3.0	3개 레벨, 25개 평가 카테고리	1 to 3(점수)
9	김정숙(2009)	IFCS에 의한 오픈소스 소프트웨어선정기법	4가지 평가, 14개 평가요소	절대평가, 0 to 2 (점수) 평가 등 복합
10	김지혁 외(2009)	응용 오픈소스 소프트웨어 특징에 적합한 논리적 품질평가 모델에 관한 연구	6개 주특성, 12개 부특성 및 12개 매트릭	매트릭별 별도계산
11	Wheeler David A. (2011)	How to Evaluate Open Source Software / Free Software (OSS/FS) Programs	13개 이슈	없음
12	Raza Arif(2011)	A Usability Maturity Model for Open Source Software	4개 관점, 11개 사용요인	0 to 4(점수)
13	Rashid Tariq(2011)	Open Source Software for Government Assessment Model	7개 원칙, 35개 세분류	Positive, Intermediate, Negative

등이 이에 속한다. 이들 속성이 정량적인 값으로 변환되고 중요도에 따라 가중치를 할당 받음으로써 최종 점수가 계산된다. 여기서 하나의 속성이 여러 하부 속성들로 세분화되면 다단계의 계층 구조가 만들어진다. 결국 이 다단계 계층 체계에서 하위 속성의 평가 점수를 바로 상위 속성의 평가 점수에 반영시켜 순차적으로 합산해 나가면 최종적인 평가 점수가 계산된다. 이렇게 속성 하나의 성숙도가 하부 속성에 의해서 계산되는 방법을 보면 다음과 같다.

$$M(d) = \sum M(d-1)_i \times W(d-1)_i \text{ where } d \geq 2 \text{ and } i = 1 \dots n \text{ and}$$

$M(d-1)_i$ = Maturity of attribute i in depth $d-1$

$W(d-1)_i$ = Weight of attribute i in depth $d-1$

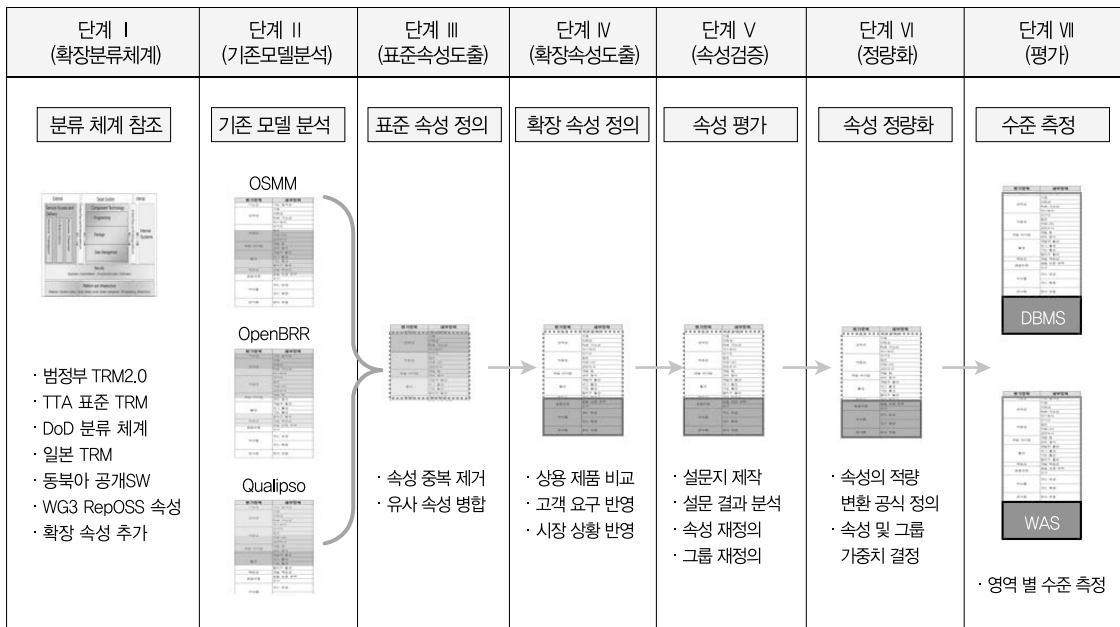
n = number of attributes in depth $d-1$

여기서 $M(d)$ 은 d 단계에 있는 속성의 성숙도(maturity)를 나타내며, 바로 하위 단계에는 n 개의 하부 속성으로 세분화되어 있다. 이러한 OSS 성숙도 모델에 대한 연구를 정리하면 <표 1>과 같다.

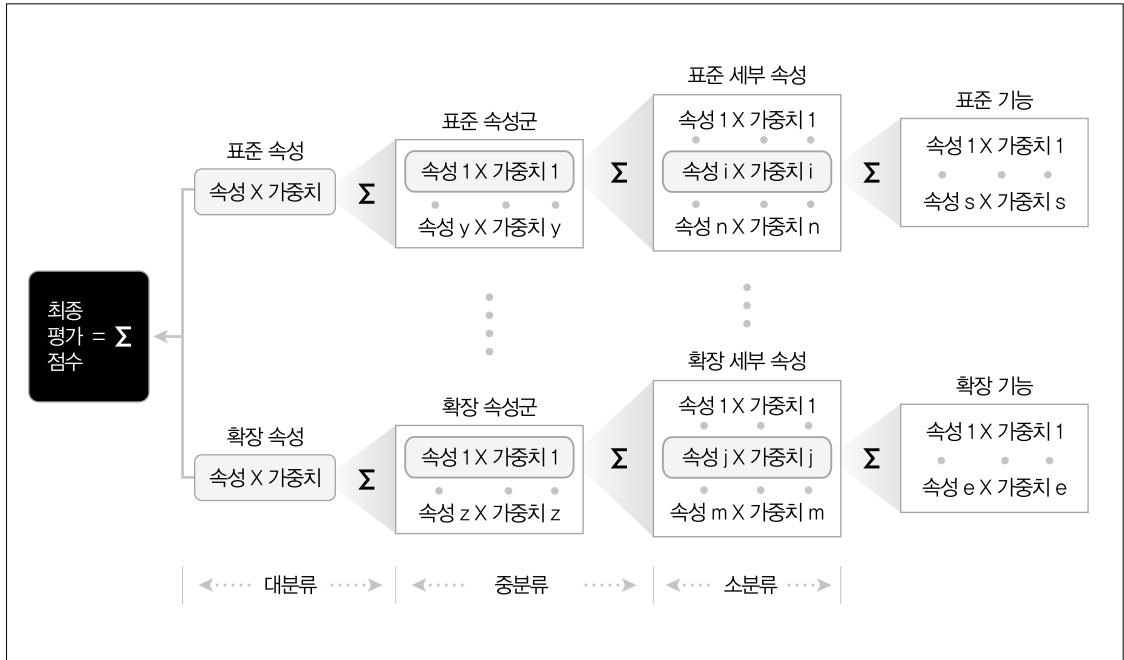
그러나 현재의 OSS 성숙도 평가 모델은 OSS 자체적인 수준 평가에 충실하기 때문에 상업적인 적용 차원에서는 적합성을 제공하지 못한 경우가 발생한다. 따라서 본 연구는 이러한 실용성을 개선하기 위하여 상업적 적용의 척도를 제공할 수 있는 효과적인 평가 모델을 설계하고 구현하였다.

<그림 1>은 구체적인 평가 과정을 7단계로 나누어 설명하고 있다. 특히 속성을 선정하는 기존모델 분석단계에서는 평가에 기여도가 있는 속성만을 중복 없이 채택하여 입력 오류에 따른 정확도 손실을 줄였다. 다음 단계에서는 속성을 표준 속성과 확장 속성으로 구분하였고, 이들을 하부 속성으로 세분화시켜 속성 구조를 대·중·소 3단계로 계층화시켰다. 최하위 세부 속성 마다 측정에 필요한 변수와 지표를 사용하여 정량화 규칙을 정의하고, 설문 결과에 따라 가중치를 할당하여 성숙도와 적용성을 평가하였다.

성숙도 모델의 정확성을 높이기 위하여 표준 세부 속성과 확장 세부 속성을 구분하고 가중치를 두어 중



<그림 1> OSS 성숙도 및 적용성 평가 모델의 설계 및 구현 방법



〈그림 2〉 3단계 속성 분류에 의한 평가 체계

요도를 반영하였다. 이후 설문을 통해 속성 분류를 검증하고 가중치에 대한 근거를 확보하였다.

또한 단순하고 명확한 변환 공식을 제시함으로써 정량화 방법을 공개하여 공정성을 기하였다. 실질적인 측정 방법의 예를 보여주기 위하여 OSS 분류체계 중 WAS(Web Application Server)를 선정하여 평가하고 이 과정에서 본 연구의 평가모델을 통해 평가하는 방법을 구체적으로 설명하였다. 전반적인 평가 방법은 〈그림 2〉에 설명되었다.

이렇게 최종적으로 얻어진 평가 점수는 OSS를 적용하는데 자체적인 성숙도 평가뿐만 아니라 CSS 대비 어느 정도 적합한지를 판가름할 수 있는 실용적인 척도를 제공할 수 있게 되었다. 위에서 설명한 7단계의 평가 과정을 요약하면 다음과 같다.

1. 확장 분류 체계 정의

OSS를 분류하고 OSS 제품과 CSS를 비교하기

위해서 OSS 분류 체계를 정의한다. 이를 위해 선진 분류 체계와 표준화된 범정부 OSS 분류체계를 참조한다.

2. 기존 성숙도 평가 모델 분석

국내·외에서 인지도가 있는 OSS 성숙도 평가 모델을 참조하여 모델 별로 속성 특성을 분석한다. 또한 총체적으로 속성을 취합하고 분류하기 위해서 모든 속성과 용어를 일관성 있게 정리한다.

3. 표준 속성 도출 및 정의

기존의 평가모델에서 취합한 속성으로부터 OSS라면 필수적으로 소유하여야 하는 속성을 정의하고 분류한다. 이렇게 분류되고 정리된 속성은 공통적이고 필수적인 특성이 있으므로 표준 속성으로 정의되며 성숙도 평가에 사용한다.

4. 확장 속성 도출 및 정의

OSS 성숙도 평가모델에 표준 속성뿐 아니라 사업적 관점에서 연관성이 있는 속성을 확장 속성으로 도출하였다. 여기에 속하는 속성들은 제품 이외 속성이 많으며, 사업 관련 속성, 회사 정책 및 라이선스⁵⁾에 관련된 속성들이 포함된다.

5. 속성 검증

이전 단계에서 도출한 표준 속성과 확장 속성의 신뢰성을 검증하기 위해 설문 실시하는 단계이다. 설문 결과에 따라 표준 속성과 확장 속성의 신뢰성 및 타당성을 검증하며 속성 및 속성군의 가중치를 할당한다.

6. 속성 정량화

평가 모델의 목적인 정량적인 평가 점수를 제공하기 위해서 모든 속성을 정량적인 숫자로 변환한다. 단순하고 명확한 정량화에 편리하도록 5점 척도로 변환 공식을 제공한다. 이 공식이 산출하는 값에 영향을 주는 변수와 지표도 함께 제공된다.

7. 평가

모든 속성을 정량화시킨 이후에 속성별 가중치를 대입하여 전체 점수를 계산한다. 자료 부족이나 상이한 데이터로 인해 널(null) 또는 해당사항 없음(N/A)이 할당된 경우에는 최하위 1점 척도를 적용한다. 속성 가중치는 설문결과에 근거하여 할당하였다.

Ⅲ. 새로운 모델 정립

1. 확장 분류 체계 정의

이 단계는 OSS 성숙도 평가 준비를 위한 가장 기초적인 작업으로써, 기능 또는 용도에 따른 OSS 영역의 분류를 수행한다. 기존 OSS 성숙도 모델은 OSS 사이의 비교 목적으로는 충분하였기 때문에 CSS를 포함한 비교는 고려하지 않았다.

그러나 OSS를 도입하거나 대체하는 상황에서는 유사한 기능의 CSS와 비교하지 않고서는 적용성에 대한 판단이 사실상 쉽지 않다. 따라서 이 단계에서는 적용성 평가를 목적으로 기존 OSS 분류체계(한국정보통신협회, 2011)를 활용하여 OSS와 CSS를 분류하였다. 이 분류 체계의 전반적인 구조는 대분류,

〈표 2〉 OSS 확장 분류 체계

대분류	중분류	소분류	OSS 예	CSS 예
Service Access and Delivery	Access Channel 포함 3개 항목	Web Browser 등 18개 항목	Firefox	Internet Explorer
Component Technology	Package 포함 3개 항목	CMS 등 22개 항목	SugarCRM	Microsoft Dynamic CRM
Security	Application Security 포함 4개 항목	Anti-Virus 등 10개 항목	Clam AntiVirus	Norton AntiVirus
Interface and Integration	Integration 포함 2개 항목	Web Service 등 6개 항목	Apache Axis2	Microsoft .NET F/W
Platform and Infrastructure	OS 포함 6개 항목	Linux 등 40개	Ubuntu	AIX

5) 라이선스는 OSS의 공통적인 속성이지만 사업적인 정책에 주요한 영향을 미치므로 표준 속성이 아닌 확장 속성으로 분류함

중분류, 소분류로 이루어진 3중 계층이다. 추후 적용성 평가를 위해 최하위 카테고리인 소분류에 OSS와 CSS가 할당되었다. 원래 모든 OSS를 할당하는 것이 원칙이나, 시장에서 가시적이고 사용자 또는 고객의 선호도가 높은 OSS를 우선적으로 선택하였다.⁶⁾ CSS는 이미 할당되어 있는 OSS를 기준으로 기능이

〈표 3〉 ISO 9126 소프트웨어 제품 품질 특성

소프트웨어 제품 품질					
기능성	신뢰성	사용성	효율성	유지보수성	이식성
적합성 정확성 상호운영성 보안성 준수성	성숙성 결합허용성 회복성 준수성	이해성 학습성 운영성 친밀성 준수성	시간반응성 자원효율성 준수성	분석성 변경성 안정성 시험성	적용성 설치성 공존성 대체성 준수성

〈표 4〉 기존 연구의 속성과 표준 속성의 매핑

속성군	표준 속성	〈표 1〉에 근거한 기존연구												
		1	2	3	4	5	6	7	8	9	10	11	12	13
기능성	기능 적합성	●	●		●		●	●	●	●	●	●	●	●
	지원성		●	●	●	●		●	●	●	●			●
	상호운영성	●	●		●	●		●	●	●	●	●	●	●
신뢰성	가용성	●	●		●		●		●	●		●	●	●
	회복성	●	●	●	●	●	●							
	최신성			●	●	●	●	●						●
	성숙성	●			●		●	●	●	●		●	●	●
사용성	이해성		●	●	●	●		●	●	●	●		●	●
	학습성		●		●	●		●					●	
	운영성			●	●		●	●	●	●	●	●	●	●
유지보수성	분석성	●	●		●	●			●	●	●	●		
	전문기술	●	●	●	●	●	●	●	●	●				●
	시험성		●		●		●	●	●		●		●	
이식성	대체성	●	●	●	●	●		●	●	●	●	●	●	
	대체후기능성	●		●	●					●				
	설치성	●		●	●				●	●	●	●		●
커뮤니티	나이 및 규모	●	●	●	●	●			●	●	●			
	주체	●			●						●			
	접근성	●	●	●		●	●				●			
	관리체계		●		●		●	●	●	●	●			

6) 현재 등록되어 있는 OSS 제품은 동북아 공개SW 활성화 포럼 WG3에서 운영하고 있는 RepOSS 웹 사이트(www.reposs.org)에서 확인 가능함

유사하여 상호 대체가 가능한 대상을 선정하였다.

이렇게 함으로써, 단순히 OSS를 영역별로 구분하는 수준을 넘어 OSS와 CSS가 상호 비교될 수 있는 확장 분류체계⁷⁾를 구성하였다. 세부적인 구성은 <표 2>과 같으며, 본 연구에서 사용한 버전에서는 343개의 OSS 제품과 126개의 상용 소프트웨어 제품이 등록되어 있다.

2. 기존 성숙도 평가 모델 분석

이 단계에서는 국내외 성숙도 모델들을 분석하여 평가에 영향을 미치는 속성들을 추출하였다. 이를 위해 <표 1>에서 정리한 기존 OSS성숙도 평가모델과 소프트웨어 제품 품질 표준인 <표 3> ISO 9126(한국정보통신기술협회, 1988)을 기반으로 모집합이 되는 OSS의 속성들을 추출하였다. 다만 ISO 9126에서 제시한 효율성 부분은 전문가들의 협의를 거쳐 OSS를 평가하기에 적합하지 않다는 의견을 반영하여 제외하였고 대신 OSS에서 중요한 요소로 간주되고 있는 커뮤니티 속성을 추가하였다.

아래 <표 4>는 기존 연구 분석을 통해 중복적으로 언급되는 속성들을 취합하여 정리한 표이다.

3. 표준 속성 도출 및 정의

표준 속성은 OSS의 공통 속성으로 OSS 자체적인 성숙도를 측정하는데 사용된다. 이전 단계에서 추출한 모집단 속성을 6개의 속성군과 20개의 표준 속성으로 정의하였다. 각 표준 속성에 대한 정의는 다음 <표 5>와 같다.

4. 확장 속성 도출 및 정의

이 단계에서는 OSS를 상업적 용도로 사용하고자 할 경우에 평가 척도가 되는 속성을 도출한다. OSS 범위 내에서 성숙도를 평가하는데 사용되는 표준 속성과는 달리 확장 속성은 상용을 목적으로 하는 적용성 평가를 위한 속성이며 이 속성들은 기술지원이나 문제해결과 같이 제품 외적인 부분, SLA(Service Level Agreement)나 고객 선호도와 같이 사업 관련 부분, 라이선스와 면책과 같이 회사 정책 관련 부분들도 포함된다. 따라서 확장 속성은 CSS와의 비교에 관련된 부분들을 포함하도록 정의하였으며 특히, 확장 속성 중 하나인 대체 후 기능성은 CSS와의 기능 비교를 통해서 적용성을 평가하는데 유용하게 사용된다.

일반적으로 기능은 카테고리 별로 상이한 특징이 있으므로 세부적인 기능 속성 또한 카테고리 별로 특수하게 정의되어야 한다. 일례로 WAS 카테고리에는 HA(High Availability)가 세부 기능이지만 프로그래밍 언어 영역에서는 해당되지 않는 속성이기 때문이다. 따라서 확장 속성은 OSS 전반적인 카테고리에 공통적으로 적용되는 속성이 아니기 때문에 하나의 카테고리 내에서 CSS와의 비교를 통해서 적용성을 평가하는데 유용하게 사용된다. 이 속성의 도출을 위해 OSS의 관련 전문가⁸⁾를 통한 델파이기법이 사용되었으며 자세한 내용은 <표 6>과 같다.

5. 속성 검증

속성 검증 단계에서는 지금까지 도출한 표준 속성과 확장 속성의 적합성 및 분류의 타당성을 검증한다. 이를 위해 OSS와 CSS를 모두 경험했던 사용자를 대상으로 설문 실시하였으며, 그 결과에 따라 속성, 속성군, 속성 가중치 등을 결정하였다. 설문을

7) 하나의 제품이 복합적인 기능을 수행할 경우에는 완전한 MECE(Mutually Exclusive Completely Exhaustive) 규정을 충족하기 어렵기 때문에 본 분류체계는 중복 할당을 허용함

8) OSS 관련 전문업체인 락플레이스, 블랙덕 소프트웨어, 삼성SDS, 아비도스, 한글과 컴퓨터와 OSS 정책을 담당하고 있는 정보통신산업진흥원(NIPA), 정보화진흥원(NIA) 등의 기업·기관의 전문가가 참여하여 온·오프라인 회의를 진행

〈표 5〉 표준 속성의 분류와 정의

속성군	표준 속성	설명
기능성	기능 적합성	1 분류 체계의 해당 카테고리에서 마땅히 제공해야 하는 목표 기능을 충실히 수행하는 수준
	지원성	2 설치 툴, 패치, 관리, 모니터링 등 목표 기능을 최상의 조건으로 수행하는데 필요한 보조 기능을 다양하게 제공하는 수준
	상호운용성	3 다양한 운영체제(Linux, Unix, Windows)에서 설치 및 작동이 가능한 수준
이식성	대체성	4 동일한 기능의 다른 OSS 제품에서 전환 및 대체(Migration)를 용이하게 수행할 수 있는 수준 (표준 수용성)
	대체후기능성	5 유사 OSS 제품으로 대체한 이후에도 동일한 기능을 수행할 수 있는 수준
	설치성	6 다양한 플랫폼에 이식될 수 있도록 구성 파라미터(Configuration Parameter)의 조작이 용이하고 설치가 간단하고 편리한 수준
신뢰성	가용성	7 에러, 버그, 정지, 종료 등 비정상적인 동작이 없이 정상적으로 운영되는 정도
	회복성	8 문제 및 장애 발생 시 복구 및 대응이 잘 되는 정도
	최신성	9 최근 일정 기간 동안 신속하게 발전하는 정도
	성숙성	10 커뮤니티의 인력 구성, 역할 분배, 운영 및 관리 체계가 얼마나 안정적이고 체계적인지 나타내는 수준
사용성	이해성	11 매뉴얼, 가이드, 튜토리얼 등 제품 사용 및 이용에 필요한 문서 및 자료의 제공 수준
	학습성	12 제품 구성, 설치, 운영에 필요한 자문, 컨설팅, 교육, 인증(자격증) 등에 관련된 서비스를 제공하는 수준
	운용성	13 사용, 운영, 관리에 편리한 기능 수준 (예 GUI 환경)
유지보수성	분석성	14 에러 또는 문제를 해결하는데 도움이 되도록 원인과 상태를 상세히 분석할 수 있는 메일링, 버그 리포팅, 이슈 트래킹 등 소통 수준
	전문기술	15 해당 OSS에 대해서 전문 업체 또는 커뮤니티의 기술 지원 서비스가 가능한 수준
	시험성	16 패치 또는 업그레이드 버전에 대한 품질 측정 수준
커뮤니티	나이 및 규모	17 오랫동안 활동이 왕성하게 지속되고 최근에도 활동이 활발하여 발전하고 있는 수준
	주체	18 커뮤니티가 쇠약하지 않고 발전할 수 있도록 기업 및 단체로부터 지속적인 경제적, 인력, 사업적 지원이 있는 정도
	접근성	19 양질의 인터넷 정보검색이 가능하고, 커뮤니티 참여와 지적 자산의 공유에 편리한 인터페이스를 제공하고 있는 수준 (이메일, 게시판, 페이스북)
	관리체계	20 커뮤니티 내에서 프로그램 개발, 소스 코드 기여, 수용 여부 심사, 품질 테스트, 로드 맵 수립 등 개발과 품질에 관련된 활동이 체계적으로 진행되고 있는 수준

실시한 결과 106명이 응답하였고 응답자의 직위 및 업무 성격에 따른 응답 빈도와 비율을 살펴보면 〈표 7〉과 같다.

정확한 결과 분석을 위해서 특정 값 하나로 치우치거나 극과 극의 상반된 값으로 불성실하게 응답한 4개를 제외하고, 정상적인 응답으로 판단되는 102개를 가지고 분석을 시도하였다. 위에서 요약한 결과 분석 내용을 자세히 설명하면 다음과 같다.

1) 신뢰성 및 타당성 분석

신뢰성 분석은 측정도구의 신뢰성(Reliability)을 확인하기 위해서 수행하는 방법으로서 본 연구에서는 내적 일치성을 평가하는 크롬바 알파(Cronbach's alpha) 계수를 사용하였다. 분석 결과인 〈표 8〉에서 확인할 수 있는 바와 같이 크롬바 알파 값이 표준 속성과 확장 속성 모두에 대해서 0.6 이상을 보여 줌으로써 속성 응답의 신뢰성이 있는 것으로 판단되었다(장원경 외, 2005).

한편, 속성 구분의 타당성을 위해 요인분석을 실시하였다. 그 결과 표준 속성에 대한 요인분석 결과 공

〈표 6〉 적응성 평가를 위한 확장 속성의 정의

속성군	확장 속성		설명
타제품이식성	상용 대체 용이성	1	CSS으로 대체(신규 도입)하거나 전환하기 용이한 수준
	대체 후 기능성	2	전환 이후 CSS를 대체해서 목표 기능을 수행하는 수준
	대체 방법론	3	상용제품으로 전환하기 위한 방법론 또는 프로세스가 제공되는 수준
라이선스이슈	라이선스 충돌	4	배포된 버전에 대해서 라이선스 문제가 있는 정도
	면책 서비스	5	저작권 문제 발생 시 법적 위험으로부터 고객을 보호할 수 있는 수준
	특허 침해성	6	특허 침해 문제가 있는 정도
고객 충실도	SLA 충실도	7	고객 서비스가 수준 별로 정의되어 있고 지원되는 정도
	문제해결 능력 만족도	8	문제가 발생했을 시 신속한 기술지원을 제공할 수 있는 수준
	기술보증	9	기술 지원을 통해 제품의 품질이 보증되는 수준
	선호도	10	고객이 선호하는 정도

〈표 7〉 설문 응답자의 직위와 업무성격에 따른 응답 빈도 및 비율 분포

구분		빈도	비율(%)
직위	사원급	28	26.4
	대리급	29	27.4
	부장급	29	27.4
	임원급 이상	20	18.8
	합계	106	100
업무경력	1년미만	12	11.3
	1~3년	23	21.7
	4~6년	15	14.2
	7~9년	14	13.2
	10년 이상	42	39.6
	합계	106	100
업무특성	정보화기획	22	20.8
	정보화운영	23	21.7
	프로그램 개발	31	29.2
	기타	30	28.3
	합계	106	100

유치가 0.5 이하로 측정된 최신성을 제외하였다. 최신성이 비록 표준 속성에 포함되지는 않았지만 급격한 성장 속도를 반영해 줄 수 있으므로 첨단 기술 또는 유망 커뮤니티를 발굴하는 지표로 충분히 활용 가능하다. 따라서 최신성을 표준이나 확장 이외의 옵션

속성으로 고려해 볼 수 있을 것이다. 요인분석 결과는 아래 〈표 9〉과 같다.

2) 속성 재분류

문헌연구, 국제표준 및 전문가들에 의해 속성들을

〈표 8〉 신뢰성 분석 결과

변수		항목수	크롬바알파
표준속성	기능성	3	0.601
	이식성	3	0.793
	신뢰성	4	0.770
	사용성	3	0.845
	유지보수성	3	0.806
	커뮤니티활동	4	0.718
확장속성	타제품 이식성	3	0.820
	라이선스 이슈	3	0.822
	고객 충실도	4	0.813

〈표 9〉 표준 속성의 타당성 분석 결과

	성분			
	1	2	3	4
사용성1	.831			
사용성3	.764			
사용성2	.761			
유지보수성1	.749			
유지보수성2	.656			
유지보수성3	.641			
커뮤니티활동4	.526			
이식성2		.735		
이식성1		.725		
기능성3		.642		
이식성3		.530		
기능성2		.517		
커뮤니티활동2			.814	
커뮤니티활동3			.707	
커뮤니티활동1			.667	
신뢰성4			.532	
기능성1				.805
신뢰성1				.708
신뢰성2				.540

배치하여도 속성 들이 속성군에 포함되는 요인인지 타당성은 부족할 수 있다. 이를 보완하기 위해 요인

분석 결과를 활용하여 〈표 10〉과 같이 속성의 재분류를 수행하였다. 재분류된 속성의 신뢰성을 분석한

〈표 10〉 표준 속성의 재정의

변경 전 표준 속성		변경 후 표준 속성		
속성군	속성	속성군	속성	
기능성	1. 기능 적합성	사용지원	이해성 (사용성1)	
	2. 지원성		학습성 (사용성2)	
	3. 상호운용성		운용성 (사용성3)	
이식성	1. 대체성		분석성 (유지보수성1)	
	2. 대체후기능성		전문기술 (유지보수성2)	
	3. 설치성		시험성 (유지보수성3)	
신뢰성	1. 가용성		관리체계 (커뮤니티4)	
	2. 회복성		기능지원	대체성 (이식성1)
	3. 최신성(신뢰성3)			대체후기능성 (이식성2)
	4. 성숙성			설치성 (이식성3)
사용성	1. 이해성	지원성 (기능성2)		
	2. 학습성	상호운용성 (기능성3)		
	3. 운용성	커뮤니티		나이 및 규모 (커뮤니티1)
유지보수성	1. 분석성		주제 (커뮤니티2)	
	2. 전문기술		접근성 (커뮤니티3)	
	3. 시험성		성숙성 (신뢰성4)	
커뮤니티	1. 나이 및 규모	신뢰성	적합성 (기능성1)	
	2. 주제		가용성 (신뢰성1)	
	3. 접근성		회복성 (신뢰성2)	
	4. 관리체계			

〈표 11〉 확장 속성의 신뢰도 분석

	성분		
	1	2	3
고객충실도2	.888		
고객충실도1	.789		
고객충실도3	.770		
고객충실도4	.571		
라이선스 이슈1		.881	
라이선스 이슈3		.855	
라이선스 이슈2		.761	
타제품이식성1			.845
타제품이식성2			.799
타제품이식성3			.743

결과 모든 속성이 크롬바 알파가 0.6 이상으로 측정되었다. 이것은 재분류된 표준 속성에도 신뢰도에는 문제가 없는 것으로 말하는 것이다. 따라서 <표 10> 우측에서 보는 바와 같이 최종적인 표준 속성은 4개의 속성군과 19개의 속성으로 재구성되었다.

확장속성에 대해서도 신뢰성과 타당성을 분석하였다. 그 결과 크롬바 알파값이 0.6 이상 도출되었고 요인분석도 아래의 표에서 보는바와 같이 분석되어 기준과 변동 없이 활용하였다.

이로써 표준 속성에는 19개, 확장 속성에는 10개의 평가 항목이 최종적으로 도출되었다. 한편, 속성의 가중치 결정에 객관성을 높이기 위해 속성의 가중치를 해당 항목을 선택한 응답자의 비율을 근거로 산정하였다. 분석 결과 전반적으로 현격한 차이를 보이는 속성은 발견되지 않았으나 속성군 중에서 신뢰성과 이식성이 상대적으로 높은 비중을 나타내고 있는 점은 가중치 결정에 유의할 만하다.

6. 속성 정량화

여기서는 이전 단계에서 도출한 29개의 속성을 정량적인 값으로 변환시키는 작업이 수행된다. 이 정량화 작업은 정성적인 특성을 숫자로 변환시켜야 하기 때문에 객관적으로 정확한 기준을 제시하기가 쉽지 않다. 그래서 지금까지 발표된 수많은 OSS 성숙도 평가 모델들은 구체적인 정량화 방법을 공개하지 않고 주관적인 결정에 의존하고 있는 것이 사실이다. 또한 정량화 방법을 제시하는 경우에도 사용자 관점에서 신뢰성을 검증하기는 쉽지 않은 상황이다(김정숙, 2009).

그러나 객관성과 신뢰성에 문제가 있다고 해서 모델만 설계하고 측정 방법을 제시하지 못한다면 평가 모델 자체의 실용적 가치도 사라지게 된다. 그러므로 이 단계에서는 다년간 OSS뿐만 아니라 CSS를 동시

에 사용하고 적용해 온 전문가들의 경험을 바탕으로 실용적인 전환 방법을 제시하였다. 이를 위해 각 속성 별로 평가에 영향을 미치는 변수와 평가 지표를 설정하였고 어떻게 점수가 측정될 수 있는지 정량화 공식을 제안하였다.

이 변환 공식은 현실적으로 조사 가능하고 정확한 값을 사용하기 위해서 변수 값의 개수보다는 종류를 세는 것이 특징이다. 예를 들어 OSS의 개념, 사용법, 구조, 알고리즘 등 지식 습득에 필요한 정보를 얼마나 잘 제공하는지 측정하는 지표인 이해성이라는 속성을 정량화시킬 때 관련 문서의 개수 대신에 문서의 종류를 사용하였다. 일반적으로 문서 개수를 정확히 조사하기란 쉽지가 않고 개수를 세는 것보다 종류를 파악하는 것이 더욱 용이하기 때문이다. 따라서 여기에 소개된 정량화 공식들은 모두 정확한 입력이 결국 정확한 평가로 이어질 것이라는 가정 하에 정의되었다.⁹⁾ 붙임 자료에 속성 별로 정량화에 필요한 변수와 지표를 제시하고 있으며, 이들을 사용해서 정량적인 속성 값을 구하는 방법을 구체적으로 설명하고 있다.

일반적으로 품질 평가에 있어서 기능은 중요한 부분을 차지한다. 이렇게 비중이 있는 기능을 정확히 측정하기 위해서 표준 속성 중 하나인 기능 적합성과 확장 속성 중 하나인 대체 후 기능성 모두 기능 속성 별로 가중치를 산정하였다. 이는 기능 속성 별로 기여도 차이가 없는 기존 공식¹⁰⁾과는 달리, 기능에 속하는 항목들에 대해서 차별적인 가중치를 부여함으로써 모든 세부 기능을 동일하게 취급해서 생기는 부정확성을 감소시켰다. 또한 제품 기능은 동일한 카테고리 내에서만 평가되기 때문에, 설문으로 기능 항목의 가중치를 산정하기에는 한계가 있는 관계로 설문 대신에 평가자의 자체적인 판단에 의해서 결정하는 방법을 택하였다.

9) 변환 공식은 절대 불변의 규칙이 아니기 때문에 각기 다른 사용자 입장에서는 자체적인 기준이나 정책에 근거하여 수정할 수 있음

10) 속성 값 = 지원하는 기능 수 / 전체 기능 수

〈표 12〉 속성 분류 및 가중치

분류	속성군	속성군 가중치	속성	속성 가중치
표준속성	사용지원	25%	1. 이해성	0.15
			2. 학습성	0.14
			3. 운용성	0.14
			4. 분석성	0.15
			5. 전문기술	0.14
			6. 시험성	0.15
			7. 관리체계	0.14
	기능지원	26%	8. 대체성	0.19
			9. 대체 후 기능성	0.20
			10. 설치성	0.21
			11. 지원성	0.20
			12. 상호운용성	0.20
	커뮤니티	22%	13. 나이 및 규모	0.23
			14. 주체	0.22
			15. 접근성	0.27
			16. 성숙성	0.28
	신뢰성	27%	17. 기능 적합성	0.33
			18. 가용성	0.34
			19. 회복성	0.33
확장속성	타제품이식성	36%	20. 상용 대체 용이성	0.33
			21. 대체 후 기능성	0.33
			22. 대체 방법론	0.34
	라이선스이슈	31%	23. 라이선스 충돌	0.33
			24. 면책 서비스	0.33
			25. 특허 침해성	0.33
	고객 충실도	33%	26. SLA 충실도	0.24
			27. 문제해결 능력만족도	0.26
			28. 기술보증	0.25
29. 선호도			0.25	

7. 평가

이 단계에서는 지금까지 설계한 평가 모델을 사용하여 OSS의 성숙도와 적용성을 측정한다. 실질적인 평가 방법의 실례를 보여주기 위해서 가장 사용 빈도가 높은 분야인 운영 체제(OS, Operating System),

웹 서버(Web Server), WAS, 데이터베이스 관리시스템(DBMS, Database Management System) 카테고리를 평가 대상으로 선정하였다. 그러나 지면의 제약으로 인해 구체적인 사례는 WAS 분야에 대해서만 설명하고 나머지 분야는 결과만 제공하도록 한다.

〈표 13〉 표준 속성 평가를 위한 표준 기능 속성의 정의 및 평가 기준

표준기능 속성군	표준기능 속성군 가중치	표준기능 속성	표준기능 속성 가중치	속성 정의	평가 기준
표준 지원성	0.2	웹 서비스 표준준수	0.6	SOAP 1.1, WSDL 1.1, UDDI 2.0 등 웹서비스 표준을 준수 하는 정도	1점: 지원 없음 3점: 중간 수준 지원 5점: 최신의 모든 웹 서비스 표준을 지원
		Java API 표준 지원	0.4	JAX-RPC, JAXM등의 Java API 표준을 지원하는 수준	1점: 지원 없음 3점: 중간 수준 지원 5점: 최신의 모든 API 표준을 지원
접속력	0.4	Multiple Data Source 접근 기능	0.5	다양한 DBMS DRIVER 지원 - DB2, Informix, Oracle, SQL Server, Sybase Adaptive Server 등 모든 Relational Database 지원	1점: 1개 RDBMS 지원 2점: 2개의 RDBMS 지원 3점: MySQL, Postgres, 1개 이상 상용 RDBMS 지원 4점: MySQL, Postgres, 2개 이상 상용 RDBMS 지원 5점: MySQL, Postgres, 3개 이상 상용 RDBMS 지원
		웹 서버 접속력	0.5	특정 제품에 종속되지 않고 OSS 뿐만 아니라 다양한 상용 웹 서버에 접속됨: Apache, Tomcat, IIS, SunOne, Web2B 등	1점: 1개 웹 서버 지원 2점: 2개의 웹 서버 지원 3점: 3개의 웹 서버 지원 4점: 4개의 웹 서버 지원 5점: 5개 이상의 웹 서버 지원
웹 시스템 개방성	0.1	아키텍처 개방성	0.1	개발 툴에 종속적이지 않은 독립적인 개발 환경을 보장함	1점: 개발 툴에 종속 5점: 개발 툴에 독립
		개방형 모델링	0.5	EJB Container를 지원하여 서버 컴포넌트 모델을 구축할 수 있음	1점: 지원 불가 3점: EJB 컨테이너 지원 OR 서버 추가가 가능함 5점: EJB 컨테이너 지원하고 서버 추가가 가능함
		Axis 지원	0.3	Web Services Engine Axis2 지원	1점: 지원 불가 5점: Portlet 지원
		포틀렛 지원	0.1	Portlet Support *5점: Axis2 최신 버전 지원	1점: 지원 불가 5점: Axis2 버전 지원
보안성	0.3	검증 기능	0.2	File Realm/ DB Realm 등 다양한 보안 검증 기능 제공	1점: 지원 불가 2점: 1개 지원 3점: 2개 지원 4점: 3개 지원 5점: DB Realm, File Realm, JDBC Realm 모두 지원
		보안 패치 용이성	0.3	패치 작업에 대한 가이드를 제시하거나 온라인을 통한 상시 다운로드를 제공하여 보안 패치 작업을 수월하게 함	1점: 지원 불가 3점: 1개 지원 5점: 패치 가이드 및 온라인 다운로드모 두 지원
		보안 체계 구성	0.5	SSO(Single Sign On), CA(Certificate Agent), SSL(Secure Socket Layer)등을 제공하여 사용자 인증 관리 및 기존 보안 정보를 기반으로 보안체계를 용이하게 구성함	1점: 지원 불가 3점: 1개 지원 4점: 2개 지원 5점: SSO, CA, SSL 모두 지원

〈표 14〉 확장 속성 평가를 위한 확장 기능 속성의 정의 및 평가 기준

확장기능 속성군	확장기능 속성군 가중치	확장기능 속성	확장기능 속성 가중치	속성 정의	평가 기준
개발 및 관리 편리성	0.4	플러그 인	0.1	이클립스 기반으로 플러그인 기능을 제공함	1점: 지원 불가 5점: 이클립스 플러그 인 지원
		개발 툴 제공	0.4	자체 통합개발환경 툴을 제공함	1점: 지원 불가 5점: 자체 통합개발환경 툴을 제공
		DML 테스트 용이성	0.2	CRUD를 실행시키는 테스트를 간편하게 실행시키는 환경 구성	1점: 지원 불가 5점: DML 테스트 환경 제공
		메모리 용량 제어	0.1	Heap memory 사용량 모니터링 및 자동 대응 알림 기능 제시	1점: 지원 불가 5점: 메모리 용량 제어 가능
		로그 관리	0.2	IP, ACCESS, USER, 예러, 디버그 모드 등을 설정하여 분류 별로 로그를 처리함	1점: 1개 이하 모드로 설정 가능 2점: 2개 모든 모드로 설정 가능 3점: 3개 모든 모드로 설정 가능 4점: 4개 모드로 설정 가능 5점: 5개 모든 모드로 설정 가능
네트워크 통신 기능	0.4	통신 프로토콜	0.7	HTTP, RMI, IOP, JMS, JNLP, COM BRIDGE, DCOME 지원	1점: 1개 이하 지원 2점: 2개 지원 3점: 3개 지원 4점: 4개 지원 5점: 주요 통신 프로토콜 5개 이상 지원
		메시징 엔진	0.3	Built-in Messaging Engine과 Java-Based Messaging Engine	1점: 지원 불가 3점: 1개만 지원 5점: Built-in 과 Jaba Based Message Engine 모두 지원
고가용성	0.2	클러스터링	0.6	가용성 증대를 위한 복수 서버 동시 운영 기능	1점: 지원 불가 5점: 클러스터링 지원
		부하 균형	0.4	고속 처리와 병복 제거를 위한 부하 분산 능력	1점: 지원 불가 5점: 로드 밸런싱 지원

우선 WAS 분야에서 평가 대상으로 Geronimo, Glassfish, JBoss, JOnAS, Resin, Tomcat 등을 대표적인 OSS로 선정하였고 CSS 비교를 위하여 WebLogic, WebSphere, JEUS 등 국내외 시장에서 지명도가 높은 제품을 선정하였다. 그리고 데이터 수집 결과에 따라 평가의 정확성이 좌우되므로 측정 과정에서 최대한 모든 속성을 입력하였으며 만일 해당 정보를 찾을 수 없거나 자료 간 불일치로 인하여 정확한 값을 입력할 수 없는 경우에는 널(null) 값으로 처리하였다. 이러한 널 값이 평가에 기여하는 영향이 작을수록 정확도는 향상될 것이라는 가정 하에

최하위 점수인 1점을 부여하여 가장 보수적인 방법으로 정량화시켰다. 이렇게 함으로써 모르는 값을 임의대로 추측하여 평가하는 것과 동일한 측정 오류를 최소화시키는 효과를 기대할 수 있었다.

가중치 할당에 있어서는 이전 단계에서 설명한 바와 같이 설문 결과에 따라서 속성군과 속성 별 설문 결과를 사용하였다. 그리고 설문으로 가중치를 결정하기 어려운 기능 속성에 대해서는 평가자의 전문적인 주관에 따라 가중치를 부여하였다. 이를 구체적으로 설명하기 위해서 WAS 평가에 사용된 표준 기능 속성과 확장 기능 속성은 각각 〈표 13〉과 〈표14〉에

정리하였다.¹¹⁾

속성 점수, 속성 가중치, 속성군 점수, 속성군 가중치가 결정되었으므로, 전체적인 평가 점수를 계산할 수 있게 되었다. 다만, 마지막 단계로 성숙도와 적용성에 대한 가중치는 사업적 활용에 더 큰 비중을 두어 차별적인 비율을 배정하였다.¹²⁾ 그러므로 최종적인 평가 점수는 다음과 같은 산출방법으로 평가되었다.

- 29개 속성 점수 = 속성 별 정량화 결과 점수
- 표준 속성 중 기능 적합성 = 표준 기능 속성 점수
- 확장 속성 중 대체 후 기능성 = 확장 기능 속성 점수
- 표준 속성군 점수 = Sum(표준 속성 점수 × 표준

속성 가중치)

- 확장 속성군 점수 = Sum(확장 속성 점수 × 확장 속성 가중치)
- 평가점수 = 표준 속성군 점수 × 표준 속성군 가중치 + 확장 속성군 점수 × 확장 속성군 가중치

위의 산출 방법에 따라 구해진 결과는 다음 <표 15>와 같다.

<표 15>의 결과는 OSS 관련 단체¹³⁾에서 실험적으로 수행한 결과이다. 그리고 자체적인 검증에 따라 카테고리 별로 평가된 OSS의 순위는 통상적으로 인식하고 있는 순서와 크게 차이가 나지 않는 것으로 확인되었다.

<표 15> 성숙도와 적용성이 반영된 최종 평가 점수의 예시

분류	CSS	OSS	평가 점수
Web Server	IIS, SungOne, Web2B	Apache	2.88
		Nginx	2.47
		Jetty	2.29
		Lighttpd	2.23
Web Application Server	JEUS WebLogic, Websphere	JBoss	4.95
		Tomcat	4.32
		Glassfish	4.09
		Geronimo	3.69
		JOnAS	2.99
		Resin	2.97
RDBMS	IBM DB2, MS SQL Server, Oracle DBMS,	MySQL	4.50
		PostgreSQL	4.25
		CUBRID	3.34
		Firebird	2.71
OS	HP-UX, IBM AIX, Oracle Solaris	Red Hat Linux	4.88
		CentOS	3.99
		Fedora	3.87
		Ubuntu	3.84

11) WAS 카테고리의 기능 평가의 항목은 Red Hat과 Wikipedia에서 제공하는 자료를 참조함

12) 본 사례에서는 OSS의 성숙도 보다 상용 적용성에 비중을 두어 각각 40 대 60의 비율을 할당함

13) 공개SW활성화 포럼의 WG3 참여 단체들로서 약 6개의 정부 및 민관 업체로 구성됨

〈표 16〉 오픈소스 적용 예시(모바일 홈쇼핑 서비스의 DBMS)

오픈소스 적용단계	카테고리 선정	표준속성반영	고객요구기능반영	확장속성반영
오픈소스 적용예시	데이터베이스	제품추천	RFP	제품제안
	CUBRID Derby MySQL PostgreSQL	MySQL PostgreSQL	신규도입 확장성 증시 로깅 변경	MySQL

여기에서 특이한 점은 카테고리가 다른 OSS 간에는 평가 점수에 차이가 있다는 점이다. 그러나 실질적인 적용에 있어서는 하나의 특정 카테고리 내에서 평가 점수를 비교하기 때문에 카테고리 별 차이는 큰 문제가 되지 않는다. 즉 OS 분류 내에 속하는 Linux들은 Web Application Server에 속하는 WAS 제품들과 다른 통계적 데이터를 보여주고 있지만, Linux는 OS 분류 안에서만 그리고 Web Application Server 제품은 Web Application Server 분류 안에서 비교해야 의미가 있으므로 측정 결과는 정상적이라고 말할 수 있다.¹⁴⁾

IV. 사례적용

여기서 제시한 평가 모델의 평가 점수는 OSS의 성능도와 적용성을 판단하는데 기반 자료로 사용이 가능하다. 다만, 평가 주체의 가치관과 여건에 따라 평가 점수는 다소 상이할 수 있다. 이 점수가 비록 모든 사용자에게 절대적인 지표를 제공하지 못한다 할지라도 OSS 적용에 참조할 수 있는 의미있는 정보를 제공하기 때문에 이 모델은 활용 차원에서 큰 도움이 될 것이 분명하다.

실질적인 상황에서 본 모델의 적용 방법을 구체적으로 설명하기 위해서, 〈표 16〉과 같이 고객의 정보 시스템 구축 시 공급사가 OSS를 선정하는 가상 시

나리오를 예시할 수 있다.

- 고객 서비스 분야가 모바일 홈쇼핑이며 적용 카테고리 DBMS를 선정함
- DBMS 카테고리 내에 등록된 모든 OSS에 대해서 표준 속성만으로 성능도를 평가함
- 성능도 평가 결과에 따라 상위 OSS인 MySQL과 PostgreSQL을 사업 제안에 적합한 후보군으로 고객에게 소개함
- 기존 DBMS를 전환하지 않는 신규 도입이고 insert 트랜잭션을 중심으로 서비스하며 향후 사용자의 지속적인 증가가 예상되는 사업 특성을 속성 정량화에 반영함
- 성능도와 적용성에 가중치를 할당해서 얻은 최종 평가 점수의 결과에 따라 최종 제안 제품으로 MySQL을 선정함

V. 결론 및 연구의 한계

본 연구에서 제시한 모델은 OSS 평가 방법의 프레임워크로서 실용적인 활용을 목적으로 OSS의 성능도와 적용성을 평가하는 메카니즘이다. 이 모델은 지금까지 발표된 여러 평가 모델의 한계를 극복하기 위해서 기존의 평가 속성을 축약적으로 재 정의하였고 정량적인 평가 지표를 제시하고 있다. 무엇보다도

14) 본 논문에서 샘플로 채택한 각 카테고리는 그 분야에서 전문가로 인정된 인력에 의해서 개별적으로 평가 되었으므로 절대적인 평가 점수 관점(최고 점수, 최하 점수, 평균 등)에서는 상이할 수 있음. 즉, 극히 보수적인 성향의 평가자에 의해 채점된 웹 서버 카테고리는 다른 카테고리에 비해서 절대적인 점수가 낮게 산출되었음

OSS 활용 차원에서의 성숙도 뿐만 아니라 상용 비교를 통한 적용성 수준을 평가함으로써 모델의 실질적인 사용 가치를 배가시켰다.

무엇보다도 본 논문은 성숙도 뿐만 아니라 적용성에 대한 척도를 제공하기 위해서 실용성, 정확성, 공정성에 근거하여 평가 체계를 구축하고 있다. 실용성 관점에서는 OSS를 선택하고 적용하는데 상용 제품과의 비교를 통해서 실질적인 적용성을 평가하였다. 실제로 OSS를 단순히 개인적인 목적이거나 단체 내부적인 용도를 넘어서 사업 제안과 프로젝트 실행 단계에서 OSS를 대체하거나 전환하는데 사례가 많기 때문에 실용적 지표의 제공은 매우 유용하다. 정확성 관점에서는 측정의 오류를 낮출 수 있도록 평가 기어도에 따라 속성을 정의하고, 한 명이 여러 차례 측정하거나 여러 명이 측정하더라도 동일한 결과가 나오도록 최대한 정량적인 방법으로 평가 공식을 수립하여 정확도를 높였다. 특히 현실적으로는 속성 값을 찾을 수 없거나, 여러 개의 속성 값이 존재할 경우에도 정량화 방법을 채택하여 unknown 속성에 따른 부정확한 측정을 보완하였다. 공정성 관점에서도 속성의 정량화 방법과 가중치 할당을 투명하고 객관적으로 처리함으로써, 정성적인 속성을 애매하게 정량화시키거나 속성의 중요도인 가중치를 할당하는데 있어서도 주관성에 의존하는 근본적인 실수를 제거하였다.

그리고 이 모델이 어떻게 현실적으로 사용될 수 있는지를 설명하기 위하여 OS, DBMS, WAS, WS등 인프라 카테고리에서 평가 대상을 선정하여 각 OSS 별로 성숙도와 적용성을 채점하는 사례를 구체적으로 기술하였다. 특별히 DBMS 분야에서 사업 제안의 예를 들어 실제 현장에서 응용될 수 있는 가상의 시나리오를 소개하였다.

따라서 본 연구는 OSS 활성화를 위한 정책적 입장에서 중요한 가치가 있고 OSS 활용성 증대에 크게 기여할 것으로 판단된다. 이를 위해 OSS 정책을 추진하는 기관 또는 단체에서 모든 OSS 분류체계별

평가를 통해 활용가능 한 OSS의 제시와 검증 체계가 필요하다고 생각된다. 그러나, 제품 속성의 정량화를 통해 최대한 객관적이고 공정성이 있는 OSS 평가 프레임워크를 제시하려고 노력하였지만 여전히 변환 공식의 변수 선정과 지표 산정 방식에서 주관적 모호성을 완전히 배제할 수는 없었다. 그리고 현실 적용을 위해 몇 개의 샘플을 활용한 결과를 제시하였으나 대상 개수와 공수의 문제로 모든 OSS 분류체계에는 적용하지 못하였다.

향후에는 이러한 문제를 해결하기 위해 OSS 사용을 고려하고 있는 사업에서 OSS 평가 모델을 지속적으로 활용하여 문제점을 파악하고 지표와 정량화 공식을 개선이 필요할 것으로 생각된다. 그러면서 다양한 OSS의 평가에 대한 활용과 선순환적인 개선 활동을 통해서 더욱 실용적인 모델로 발전시켜 나가야 할 것이다.

■ 참고문헌

- 권문주·김태웅·김민하 (2008). “국내 공개소프트웨어의 도입 실태 및 활성화 장애요인에 관한 탐색적 연구.” 『정보화정책』, 15(4): 3-21.
- 김정숙 (2009). 「IFCS에 의한 오픈소스 소프트웨어 선정 기법」. 숭실대학교 대학원 석사학위논문.
- 김지혁·류성열 (2009). “응용 오픈소스 소프트웨어 특성에 적합한 논리적 품질평가 모델에 관한 연구.” 『정보처리학회논문지』, 16(1): 73-81.
- 송지훈·이시진 (2008). “오픈소스 소프트웨어 특성을 고려한 소프트웨어 평가모델.” 한국인터넷정보학회 춘계학술발표대회 발표논문.
- 심화영 (2012). “공공정보화 공개SW 확대 실효성 논란.” 『디지털타임즈』. 7월 9일.
- 이영민·류성열·김종배 (2008). “오픈소스 소프트웨어의 선정 절차에 관한 연구.” 『정보처리학회논문지』, 15(6) : 793-802.
- 장원경·김태균 (2005). 「한글 SPSS 12을 이용한 자료 분석의 이해와 응용」. 서울: 대경출판사.
- 정보통신산업진흥원 (2010). 「공개SW Governance V2.0」. 서울: 정보통신산업진흥원.

- 정의정·강동재·정성인 (2011). “공개SW기반 클라우드 컴퓨팅 기술 현황.” 『전자통신동향분석』, 26(5): 43-54.
- 한국소프트웨어진흥원 (2007a). 「소프트웨어 산업백서 2007」. 서울: 한국소프트웨어진흥원.
- 한국소프트웨어진흥원 (2007b). 「전 세계 오픈소스 SW 분류 및 성숙도 평가 - Optaros' Open Source Catalogue 2007 보고서 중심」. 서울: 한국소프트웨어진흥원.
- 한국정보통신기술협회 (1988). 「소프트웨어 품질 특성 및 메트릭 - 품질 특성 및 부특성」. 서울: 한국정보통신기술협회.
- 한국정보통신기술협회 (2011). 「공개소프트웨어 분류 체계 및 프로파일」. 서울: 한국정보통신기술협회.
- 한국정보화진흥원 (2009). 「법정부 기술참조모델 v2.0」. 서울: 한국정보화진흥원.
- Atos Origin (2006). “Method for Qualification and Selection of Open Source software (QSOS) version 1.6.” <http://www.qsos.org>. (Retrieved on Oct 10, 2012).
- Bennett, Jeremy (2011). “Open Source in Embedded System Development.” <http://www.embecosm.com>. (Retrieved on Oct 18, 2012).
- Campbell, Piers R. J. & Faheem Ahmed (2011). “An Assessment of Mobile OS-Centric Ecosystems.” *Journal of Theoretical and Applied Electronic Commerce Research*, 6(2): 56-60.
- çetin, Gürkem & M. Gokturk (2008). “A measurement based framework for assessment of usability-centricness of open source software projects.” *Signal Image Technology and Internet Based Systems*, IEEE International Conference on.
- del Bianco, V. & Lavazza, L. & Morasca, S. & Taibi, D. (2009). “Quality of Open Source Software: The QualiPSo Trustworthiness Model.” *International Federation for Information Processing*, 199-212.
- Frans-Willem Duijnhouwer & Chris Widdows (2003). “Open Source Maturity Model.” <http://bolsa.info.unlp.edu.ar>. (Retrieved on Oct 10, 2012).
- Fuggetta, Alfonso (2003). “Open source software - an evaluation.” *The Journal of Systems and Software*, 66(1), 77-90.
- Information-Technology Promotion Agency of Japan (2011). “Technical Reference Model for the Government Procurement of Information Systems (TRM) 2010.” <http://www.ipa.go.jp>. (Retrieved on Oct 12, 2012).
- Karin van den Berg (2005). “Finding Open Options.” Master Degree Thesis. Tilburg University.
- Lurie Marty (2012). “Open Source Big Data for the Impatient.” <http://www.ibm.com>. (Retrieved on Oct 10, 2012).
- MAMPU (2008). “OPEN SOURCE SOFTWARE (OSS) IMPLEMENTATION GUIDELINES.” <http://www.oscc.org.my>. (Retrieved on Oct 12, 2012).
- Navica Inc. (2009). “The Open Source Maturity Model is a vital tool for planning open source success.” <http://www.navicasoft.com/pages/osmm.htm>. (Retrieved on Oct 12, 2012).
- Petrinja, Etiel & Sillitti, Alberto & Succi, Giancarlo (2010). “Comparing OpenBRR, QSOS, and OMM Assessment Models.” 6th International IFIP WG 2.13 Conference on Open Source Systems, OSS 2010, Notre Dame, IN, USA, May 30 - June 2, 2010. Proceedings.
- Rashid, Tariq (2011). “Open Source Software for Government Assessment Model.” <http://www.computerweekly.com>. (Retrieved on Oct 12, 2012).
- Raza, Arif (2011). “A Usability Matutity Model for Open source Software.” A partial fulfillment of the requirements for the degree of Doctor of Philosophy. The School of Graduate and Postdoctoral Studies, The University of Western Ontario.

- SpikeSource, Carnegie Mellon West, and Intel Corporation (2005). "Business Readiness Rating for Open Source." <http://www.openbrr.org> (Retrieved on Oct 15, 2012).
- Taibi, Davide & Lavazza, Luigi & Morasca, Sandro (2007). "OpenBQR: a framework for the assessment of OSS." *IFIP Working Group 2.13 on Open Source Software*, 173-186.
- Wheeler, David A. (2011). "How to Evaluate Open Source Software / Free Software (OSS/FS) Programs." http://www.dwheeler.com/oss_fs_eval.html. (Retrieved on Oct 12, 2012).
- Wittmann, Marion & Nambakam, Ranga (2009). "QualiPSo, Quality Platform for Open Source Software." <http://www.qualipso.org>. (Retrieved on Oct 15, 2012).
- Wong, Ronnie Anthony (2011). "Assessment of Open Source Business Strategies in the Domain of Embedded Systems." *University of Oulu Department of Information Processing Science*, 13-17.

〈붙임〉 속성 별 정량화 공식과 적용 방법

속성군	속성	변환 공식	적용 방법
사용지원	이해성	변수 = {제품 기술 자료} 지표 = 제공하는 기술 문서 종류 / 전체 기술 문서의 종류 1 점: 0.0 ≤ 지표 < 0.2 2 점: 2.0 ≤ 지표 < 0.4 3 점: 4.0 ≤ 지표 < 0.6 4 점: 6.0 ≤ 지표 < 0.8 5 점: 0.8 ≤ 지표 ≤ 1.0	전체 기술 문서의 종류 = 5 1. 매뉴얼: 사용자, 기술자, 관리자 2. 데이터 시트: 테스트 결과 3. 교육 자료: 이론, 실습 4. 기술 보고서: 논문, 백서, 제안서 5. 가이드: 프로그램 샘플, 패턴
	학습성	변수 = {학습 지원 서비스} 지표 = 제공하는 기술 서비스의 종류 / 전체 기술 서비스의 종류 1 점: 0.0 ≤ 지표 < 0.2 2 점: 2.0 ≤ 지표 < 0.4 3 점: 4.0 ≤ 지표 < 0.6 4 점: 6.0 ≤ 지표 < 0.8 5 점: 0.8 ≤ 지표 ≤ 1.0	전체 기술 서비스의 종류 = 5 1. 컨설팅 서비스 2. 교육 서비스 3. 인증서 발급 서비스 4. 콜 센터 지원 서비스 5. 원격 지원 서비스
	운용성	변수 = {운용 지원 유틸리티} 지표 = 제공하는 유틸리티의 종류 / 전체 유틸리티의 종류 1 점: 0.0 ≤ 지표 < 0.2 2 점: 2.0 ≤ 지표 < 0.4 3 점: 4.0 ≤ 지표 < 0.6 4 점: 6.0 ≤ 지표 < 0.8 5 점: 0.8 ≤ 지표 ≤ 1.0	전체 유틸리티의 종류 = 5 1. 웹 기반 관리 콘솔 2. 서버 설정 3. 디플로이 4. 환경 설정 5. 모니터링
	분석성	변수 = {이메일, 게시판, 이슈 트래킹, FAQ, Q&A, 위키, 블로그} 지표 = 제공하는 분석 지원 종류 / 전체 분석 지원 종류의 수 1 점: 0.0 ≤ 지표 < 0.2 2 점: 2.0 ≤ 지표 < 0.4 3 점: 4.0 ≤ 지표 < 0.6 4 점: 6.0 ≤ 지표 < 0.8 5 점: 0.8 ≤ 지표 ≤ 1.0	전체 분석 지원 종류의 수 = 5 1. 이-메일 연락 2. 게시판 운영 3. 이슈 트래킹 지원 4. FAQ 또는 Q&A 제공 5. 위키 또는 블로그 운영
	전문기술	변수 = {지원 방식} 1 점: 기술 지원 불가 2 점: 커뮤니티 기술 지원 3 점: 해외 업체가 직접 지원 4 점: 국내 업체를 통한 간접 지원 5 점: 국내 업체의 직업 지원	상위 점수는 하위 점수의 배점기준을 충족시킬 수 있음 4점: 국내업체는 지사 형태로 해외업체의 기술 지원을 연결함
	시험성	변수 = {버전 종류, 테스트} 1 점: 품질 측정 불가 2 점: 안정화된 버전이 1.0 미만임 3 점: 안정화된 버전만 배포 4 점: 안정화된 버전 이전에 1개 이상의 테스트 버전 배포 5 점: 안정화된 테스트 수준에 따라 복수개의 다양한 버전 배포	릴리즈 버전이 1.0 미만인 경우에는 일반적으로 상용 적용이 어려움 테스트 버전으로는 Alpha, Beta, Gamma 등이 있음

	관리체계	<p>변수 = {로드맵, 커미터, 프로세스, 테스트, 운영자}</p> <p>지표 = 진행되는 관리활동 종류의 수 / 전체 관리활동 종류의 수</p> <p>1 점: 0.0 (<= 지표 < 0.2)</p> <p>2 점: 2.0 (<= 지표 < 0.4)</p> <p>3 점: 4.0 (<= 지표 < 0.6)</p> <p>4 점: 6.0 (<= 지표 < 0.8)</p> <p>5 점: 0.8 (<= 지표 <= 1.0)</p>	<p>전체 관리활동 종류의 수 = 5</p> <ol style="list-style-type: none"> 로드맵 발표 복수 커미터 활동 (단독이 아님) 심사 프로세스 운영 품질 테스트 수행 복수 유지 운영자(maintenanc operator) 활동
기능지원	대체성	<p>변수 = {대체 기간}</p> <p>1 점: 4주일 (<= 대체 기간)</p> <p>2 점: 3주일 (<= 대체 기간 < 4주일)</p> <p>3 점: 2주일 (<= 대체 기간 < 3주일)</p> <p>4 점: 1주일 (<= 대체 기간 < 2주일)</p> <p>5 점: 0주일 (<= 대체 기간 < 1주일)</p>	OSS에서 OSS로 전환하는데 필요한 시간으로 측정
	대체 후 기능성	<p>변수 = {표준 기능}</p> <p>지표 = 제공 표준 기능 / 표준 기능 수</p> <p>1 점: 지표 <= 60%</p> <p>2 점: 60% < 지표 <= 70%</p> <p>3 점: 70% < 지표 <= 80%</p> <p>4 점: 80% < 지표 <= 90%</p> <p>5 점: 90% < 지표 <= 100%</p>	대체 전 OSS의 표준 기능을 수행할 수 있는 수준으로 측정
	설치성	<p>변수 = {설치 파라미터}</p> <p>1 점: 이기종 플랫폼으로 설치 불가</p> <p>2 점: 일부 리눅스 기반 플랫폼</p> <p>3 점: 모든 리눅스 기반 플랫폼</p> <p>4 점: 모든 리눅스 및 2개 이하 상용 OS 기반 플랫폼</p> <p>5 점: 모든 리눅스 및 대부분의 상용 플랫폼</p>	<p>플랫폼 전환의 용이성</p> <p>OS 변경에 따른 환경 변수 조작에 대한 용이성</p> <p>상용 플랫폼의 운영 체제로는 Unix, Windows, Mac OS 등이 있음</p>
	지원성	<p>변수 = {툴, 패치, 가이드, 모니터, 알람, 콘솔}</p> <p>지표 = 제공하는 지원 종류의 수 / 전체 지원 종류의 수</p> <p>1 점: 0.0 (<= 지표 < 0.2)</p> <p>2 점: 2.0 (<= 지표 < 0.4)</p> <p>3 점: 4.0 (<= 지표 < 0.6)</p> <p>4 점: 6.0 (<= 지표 < 0.8)</p> <p>5 점: 0.8 (<= 지표 <= 1.0)</p>	<p>전체 지원 종류의 수 = 5</p> <ol style="list-style-type: none"> 별도의 설치 툴 제공 패치 및 가이드의 다운로드 제공 제품 모니터링 툴 제공 장애 알람 제공 (SMS, Push e-Mail) 웹 콘솔 제공
	상호 운용성	<p>변수 = {호환성}</p> <p>1 점: 단일 리눅스에서만 이식 가능</p> <p>2 점: 일부 리눅스에서 이식 가능</p> <p>3 점: 모든 리눅스에서 이식 가능</p> <p>4 점: 모든 리눅스 및 2개 이하 상용 OS에서 이식 가능</p> <p>5 점: 모든 리눅스 및 대부분의 상용 OS에서 이식 가능</p>	설치성과는 달리 인터페이스가 있는 프로그램과의 연동성 및 최적의 운용 상태 유지를 위한 파라미터 설정 용이성 등을 측정함
커뮤니티	나이 및 규모	<p>변수 = {버전 번호, 연령}</p> <p>지표 = 최종 버전 번호 × 나이</p> <p>1 점: 0 (<= 지표 < 12)</p> <p>2 점: 12 (<= 지표 < 24)</p> <p>3 점: 24 (<= 지표 < 72)</p> <p>4 점: 72 (<= 지표 < 180)</p> <p>5 점: 180 (<= 지표)</p>	<p>지표는 최종 버전 번호와 월 단위의 커뮤니티 나이를 곱해서 산출함</p> <p>버전 번호가 1.0 이상이고 커뮤니티 나이도 12개월 이상이 되어야 자생력이 있는 커뮤니티로 인정함</p> <p>버전이 3.0 이상이고 연수가 5이상이면 최상위 수준으로 인정함</p>

	주체	<p>변수 = { 후원 단체 유무}</p> <p>1 점: 지원 없음 2 점: 하나의 중소기업 지원 3 점: 복수의 중소기업 지원 4 점: 하나의 대기업의 지원 5 점: 복수의 대기업의 지원</p>	인력 및 자금에 대한 후원 단체의 유무로 측정함
	접근성	<p>변수 = {계시판, 포럼, 위키, 검색성, 인터넷}</p> <p>지표 = 제공하는 접근 방법의 종류 / 전체 접근 방법의 종류 개수</p> <p>1 점: 0.0 <= 지표 < 0.2 2 점: 2.0 <= 지표 < 0.4 3 점: 4.0 <= 지표 < 0.6 4 점: 6.0 <= 지표 < 0.8 5 점: 0.8 <= 지표 <= 1.0</p>	<p>전체 접근 방법의 종류 개수 = 5</p> <ol style="list-style-type: none"> 1. 게시판 운영 2. 포럼 운영 3. 위키 운영 4. 인터넷 검색 시 첫 페이지 출력 5. 인터넷 사이트에서 정보 제공 <p>외부에서 커뮤니티로 연락하거나 관련 정보를 얻을 수 있는 용이성</p> <p>OSS 커뮤니티에 대해 전문 정보를 제공하는 인터넷 사이트로는 ohloh.net, wikipedia.org 등이 있음</p>
	성숙성	<p>변수 = {기간, 버전 출시, 관리 체제, 평가 방법, 위원회 운영}</p> <p>지표 = 충족하는 성숙지표의 종류 / 전체 성숙 지표의 종류 개수</p> <p>1 점: 0.0 <= 지표 < 0.2 2 점: 2.0 <= 지표 < 0.4 3 점: 4.0 <= 지표 < 0.6 4 점: 6.0 <= 지표 < 0.8 5 점: 0.8 <= 지표 <= 1.0</p>	<p>전체 성숙 지표의 종류 개수 = 5</p> <ol style="list-style-type: none"> 1. 최초 버전 출시 이후 3년 이상 지속적으로 신규 버전 출시 2. 최근 배포한 안정된 버전의 넘버가 1.0 이상 3. 관리 운영자(maintenance operator), 커뮤니티(심의회), 개발자 등의 운영 체제 확립 4. 기여도 및 참여도에 따른 개발자의 등급 체제 확립 5. 이사회 운영 - 개인의 독단적 판단이 아닌 위원회에 의한 의사 결정 방식
신뢰성	기능 적합성	<p>변수 = {표준기능, 기능 별 가중치}</p> <p>기능 적합성 = SUM(표준 기능 × 표준 기능 가중치)</p>	<p>표준 기능은 OSS가 속하는 소분류 카테고리 안에서 표준적으로 제공하는 기능임</p> <p>가중치는 평가자에 의해 결정됨</p>
	가용성	<p>변수 = {연간 가동률}</p> <p>지표 = 연간 실행 시간 / 1년</p> <p>1점: server - 99% 이하 application - 90% 이하 2점: server - 99.9% 이하 application - 94% 이하 3점: server - 99.99% 이하 application - 96% 이하 4점: server - 99.999% 이하 application - 98% 이하 5점: server - 100% application - 100%</p>	<p>연간 가동률은 제품의 추천 사양 환경에서 운영이 중단되지 않고 실행되는 수준임</p> <p>일반적으로 1년 동안 초단위의 가용한 시간 비율이 사용됨</p>
	회복성	<p>변수 = {회복기간}</p> <p>1점: 1일 < 회복기간 2점: 1시간 < 회복기간 <= 1일 3점: 10분 < 회복기간 <= 1시간 4점: 1분 < 회복기간 <= 10분 5점: 0초 < 회복기간 <= 1분</p>	<p>1일: 참을 수 없음 1시간: 참을 수 있으나 불만이 큼 10분: 불만 없이 참을 수 있음 1분: 지연을 느끼지 못함</p>

타제품 이식성	상용 대체 용이성	변수 = {대체기간} 1점: 12주 <= 대체기간 2점: 8주 <= 대체기간 < 12주 3점: 4주 <= 대체기간 < 8주 4점: 2주 <= 대체기간 < 4주 5점: 0주 <= 대체기간 < 2주	CSS를 대체하는데 걸리는 시간 = 데이터 이전을 포함하여 테스트까지 완료하는데 소요되는 주단위 기간임
	대체 후 기능성	변수 = {확장기능, 기능 별 가중치} SUM(확장 기능 × 확장 기능 가중치)	확장기능 = OSS가 속하는 소분류 카테고리 안에서 표준 이외에 제공하는 기능 가중치는 평가자에 의해 결정됨
	대체 방법론	변수 = {대체 방법론} 1점: 대체 방법론 없음 3점: 개념적 방법론만 있음 5점: 실용적 방법론도 있음	개념적 방법론: 유즈 케이스, 다이어그램, 흐름도, 문장 서술, 활동 단계(phase) 수준의 전환 방법론 실용적 방법론: Data type, parameter, procedure, API, 활동 단위(activity) 수준의 전환 방법론
라이선스 이슈	라이선스 충돌	변수 = {라이선스 충돌 개수} 1점: 충돌 존재 5점: 충돌 부재	충돌: 배포하는 제품과 그 내부에 사용된 OSS 간의 라이선스 의무 사항이 불일치함
	면책 서비스	변수 = {면책 서비스 조건} 1점: 면책 서비스 불가 5점: 면책 서비스 제공	면책 서비스: 사용자가 법적인 문제에 연루되지 않도록 공급자가 모든 책임을 지는 서비스
	특허 침해성	변수 = {특허 침해 개수} 1점: 특허 침해 있음 5점: 특허 침해 없음	특허 소송 보복 조항이 있는 라이선스라 할지라도 제3자의 특허에 대해서는 침해가 발생한 경우 보호받을 수 없음
고객 충실도	SLA 충실도	변수 = {기술 서비스 정책 충실도} 지표 = 제공하는 기술서비스의 종류 / 전체 기술 서비스의 종류 개수 1 점: 0.0 <= 지표 < 0.2 2 점: 2.0 <= 지표 < 0.4 3 점: 4.0 <= 지표 < 0.6 4 점: 6.0 <= 지표 < 0.8 5 점: 0.8 <= 지표 <= 1.0	전체 기술 서비스의 종류 개수 = 5 1. 기술지원 서비스의 가격 정책이 수립되어 있음 2. 최초 지원 요청 후 응답 시간이 지정되어 있음 3. 상시 패치 및 업데이트 서비스가 지정되어 있음 4. 고객 창구가 개설되어 있음 5. 교육 또는 컨설팅 서비스가 제공되고 있음
	문제해결 능력 만족도	변수 = {해결기간} 1 점: 2주 <= 해결기간 2 점: 1주 <= 해결기간 < 2주 3 점: 2일 <= 해결기간 < 1주 4 점: 1일 <= 해결기간 < 2일 5 점: 해결기간 < 1일	2주 이상은 참을 수 없는 기간으로 설정함 당일 접수된 문제를 당일 해결할 수 있으면 최고 수준으로 설정함
	기술보증	변수 = {보증 조건, 보상 조건} 1점: 제품 보증 및 보상 없음 3점: 제품 보증 제공 및 보상 없음 5점: 제품 보증 제공 및 보상 제시	OSS 자체는 보증(warranty)가 없으나 사업적인 목적으로 문제 발생 시 즉각적인 해결 서비스를 제공함 제품 자체의 하자로 인하여 사용자의 손해가 발생할 경우 보상을 약속할 수 있음
	선호도	변수 = {점유율, 레퍼런스 개수} 1 점: 점유율 = 0%, 레퍼런스 0개 2 점: 점유율 < 2%, 레퍼런스 < 2개 3 점: 점유율 < 5%, 레퍼런스 < 5개 4 점: 점유율 < 8%, 레퍼런스 < 8개 5 점: 점유율 > 10%, 레퍼런스 > 10개	점유율은 상용을 포함한 시장에서 copy수 기준으로 산정함 레퍼런스는 상용 서비스를 제공하는 시스템으로 사용되는 사이트 개수임