

오픈 소스 프로젝트 문서화 가이드라인 연구

박치남*, 류성열**

*승실대학교 컴퓨터학과

**승실대학교 컴퓨터학과 교수

e-mail: hbsnow@korea.com

A Documentation Guide to Open Source Software Project

Chi-Nam Park*, Sung-Yul Rhew**

*Dept of Computer Science, Soong-Sil University

**Dept of Computer Science Prof, Soong-Sil University

요 약

오픈소스 소프트웨어를 효율적으로 활용하기 위한 다양한 연구와 정책이 이루어지고 있으나 OGC에서 제기한 가이드에서 볼 수 있듯이 많은 단점을 가지고 있다. 특히 오픈 소스를 이용하여 새로운 제품을 개발하고자 하는 사람들이 제품의 선정과 구체적인 소스 코드를 분석하기 위한 문서화의 표준이 없다.

본 연구에서는 이러한 문제점을 극복하기 위한 기본적인 방안을 작성하기 위하여 오픈소스의 개발모형을 조사 분석하고, 분석된 모형에 근거한 오픈소스의 선정과 소스코드를 효율적으로 분석하기 위한 문서화 표준의 가이드라인을 작성하고자 한다. 또 이를 기반으로 오픈소스 커뮤니티 관리적 차원에서의 해결방안으로 관리항목의 기능을 분류하고 특성분류에 따른 문서화 작성지침을 제안 한다

1. 서론

[오픈소스 소프트웨어(OSS: Open Source Software)는 누구나도 소스코드를 읽을 수 있고 사용자가 능력이 있다면 각종 버그의 수정은 물론이고 그것을 개조하여 기능을 추가할 수 있으며, 누구나 그 소프트웨어의 개발에 참여할 수 있다. 따라서 오픈소스 소프트웨어는 프로그램을 복제하여 배포할 수 있는 자유, 소프트웨어의 소스코드에 접근할 수 있는 자유, 프로그램을 개선할 수 있는 자유를 개발자에게 보장한다.

이처럼 소스코드의 공개를 통해 사용, 복제, 수정, 배포가 자유로운 OSS가 사용소프트웨어의 독과점에 의한 폐해를 방지하고, 유지보수 및 업그레이드에 투여되는 예산과 절차를 감축할 수 있다는 장점을 바탕으로 전 세계적으로 그 영역이 확대되고 있다.

구분		소스코드공개	
		예	아니오
비용	무	비상업용	프리웨어
	유	오픈소스 소프트웨어	웨어웨어
비용	상	상업용	사적목적
	유	오픈소스 소프트웨어	상업적 소프트웨어

(표 1) 소프트웨어의 라이선싱과 배포관련유형

구분	내용
융통성	라이선스 비용이나 예산에 제한을 받지 않고, 다양한 오픈소스들을 테스트 한 뒤 최선의 것을 선택할 수 있음
기술지원	신속한 문제해결, 빨라진 성능개선 프로세스, 기술의 공동 습득이 가능함
기술혁신	유료일 경우 사용치 않았을 기술의 실험 적용이 가능함
재활용	소스코드 접근이 가능함으로 재활용이 증가함
품질	이미 검증된 소스를 사용함에 따라 개발이 빨라지고 유연해짐
표준	표준에 충실하고 상호 유연성이 뛰어나

(표 2) 오픈소스활용의 장점

* 본 연구는 승실대학교 교내연구비 지원으로 이루어졌음.

하지만 많은 장점을 가진 오픈소스 소프트웨어의 활용이 기대만큼 확산되지 못하며 성공적인 활용사례가 적은 실정이며 그 요인은 오픈소스 프로젝트의 활용상에 문제점들로 인해 회사와 개발자는 오픈소스를 채택하기에 주저하고 있기 때문이다.

구분	내용
기능의 불확실성	무엇이 OSS를 구성하고 있는지와 상대적인 장단점이 무엇인지가 불명확하다
기술지원의 불확실성	제품의 지원이 분산되거나 지원을 못받는 것에 대한 불안감이 존재한다.
탐색의 어려움	특정한 비즈니스 문제들에 대한 적절한 OSS응용제품을 확인하는데 대한 어려움이 존재한다.
문서화 미비	문서화가 특정개인의 취향에 따라 편협하게 이루어졌거나 아예 없는 경우도 존재한다.
로드맵 미비	상용 프로그램에서 볼수 있는 Roadmap을 기대하기 힘들다
하드웨어 지원	특정 소프트웨어보다 새로운 하드웨어를 지원하는 면에서 뒤쳐진다.

(표 3) 오픈소스 활용상의 문제점

오픈소스 소프트웨어의 기능의 불확실성과 탐색의 어려움으로 인해 기업의 올바른 오픈소스 분석/선정을 기대하기 힘들며 문서화미비, 기술지원의 불확실성, 로드맵미비, 하드웨어지원 미비로 오픈소스 소프트웨어 도입 후 문제가 발생할 경우 책임소재를 명확히 할 수 없기에 기업이 오픈소스를 채택하지 않은 것은 어쩔 수 없는 결과일 것이다.

이러한 문제를 해결하기 위하여 본 논문에서는 오픈소스 커뮤니티 관리적 차원에서의 해결방안으로 기존에 오픈소스 커뮤니티의 관리항목의 기능분류와 표준 도큐멘테이션을 기반으로 상용소프트웨어 도큐멘테이션과 오픈소스소프트웨어의 특성분류에 따른 도큐멘테이션 산정과 작성지침을 제안하였다

2. 관련연구 및 조사

2.1 Martin Fink이 오픈소스를 활용한 개발 모델

Martin Fink는 오픈소스를 활용한 소프트웨어 개발을 독점라이선스에 판매되는 소프트웨어를 오픈소스 라이선스로 전환하는 '아웃바운드'와 현존하는 OSS를 사용하거나 오픈소스를 제품의 부분으로 혹은 IT프로젝트안에서 사용할수 있게 통합하는 '인바운드'로 구분하고 있다

여기서 아웃바운드의 장점은 상용 소프트웨어로서의 정의된 기능성을 통하여 기능의 불확실성을 해결할 수 있으며 탐색과 문서화의 미비는 극복할 수 있게 될 것이다. 기술지원의 불확실성 또한 회사형 오픈소스 프로젝트로 인해 지속적인 기술지원이 향상될

것이며 로드맵과 하드웨어지원여부 또한 가시화될것으로 분석된다.

이같은 아웃바운드형 오픈소스 소프트웨어의 편입은 독점화를 위해 그 수요가 많지 않을것이며 프로젝트화후 프로젝트에 대한 일반 개발자들의 진입장벽은 높을 수밖에 없을 것이다.

2.2 Ramel Sophie의 오픈소스 활용 유형

Ramel Sophie는 사적 소프트웨어에 대한 소프트웨어 재사용 유형을 OSS 재사용에 적용하여 각각의 경우에 대한 고려사항을 제시하고 있다.

'컴포넌트 재사용'은 자체 소프트웨어에 OSS 컴포넌트만을 재사용하는 경우이다, '프레임워크 재사용'은 컴포넌트 재사용과 유사하지만 단일 프레임워크에 다른 기능들을 적용시킨다는 점에서 차이가 있다.

'코드의 재사용'은 OSS 코드의 일부(일부 라인, 메소드 또는 함수, 클래스 또는 전체프로그램)를 복사하는 과정으로 구성된다.

OSS에 일부만을 복사/재사용하는 방법을 통하여 오픈소스 활용상의 문제점은 그 범위가 축소됨으로 인해 불확실성은 낮아질 것이며 선별을 위한 노력과 문제발생 확률은 적어질 것이다.

이는 오픈소스 소프트웨어의 장점이 축소되는 문제가 있으며 문서화의 미비에서 오는 재사용범위설정상의 어려움은 여전히 존재할 것이다.

2.3 Diomidis Spinellis의 오픈소스 코드 분석 기법

Diomidis Spinellis은 오픈소스 코드의 다양한 분석기법을 다음과 같이 제시하고 있다. 첫째, 프로젝트의 소스코드트리, 즉 프로젝트의 소스코드를 담은 계층적인 디렉토리 구조의 분석을 통해 프로젝트의 기반 구조와 소프트웨어 공정구조, 배치구조를 파악한다. 이를 통하여 소스트리에는 소스코드 이외에도 명세서, 최종 사용자용 문서와 개발자용 문서, 테스트 스크립트, 멀티미디어지원, 빌드도구, 예제, 지역화파일, 리비전 내역, 설치절차, 사용권 허가정보 등을 포함한다. 둘째 프로그램의 디버깅 출력을 프로그램 제어 흐름과 자료요소들의 핵심적인 부분을 이해하는데 활용한다.

Diomidis Spinellis의 분석기법은 사용자/개발자가 오픈소스 소프트웨어의 활용상의 문제점을 직접 해결하는 노력과 그 방법을 제시함에 있어 주목할만한 연구로 생각된다.

이런 노력은 오픈소스 프로젝트의 문제점을 극복하는데 중요한 근본적인 방법이 되어줄 수 있으나 많은 노력이 수반되어야함에 있어서의 우려와 그 문서 산출물에 있어 구체적인 가이드의 제시가 없다

2.2 오픈소스 소프트웨어 활용 문제점 해결

독점라이선스에 판매되는 소프트웨어를 오픈소스 라이선스로 전환하는 ‘아웃바운드’ 방식은 그 수요가 많지 않고 일반 개발자의 진입장벽이 높으며, OSS의 일부만을 복사/사용하는 재사용방법은 OSS의 장점을 축소하며 문서화의 미비로 인한 재사용법 위설정상에 어려움이 있으며, 사용자/개발자가 문제점을 직접 해결하는 노력과 방법은 많은 노력이 수반되어야 하며 그 문서 산출물에 대한 구체적인 가이드라인이 부족하다.

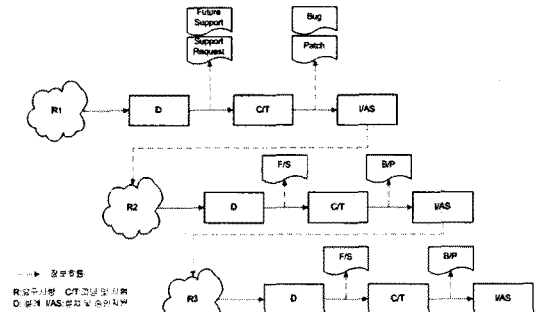
본 논문에서는 오픈소스 소프트웨어의 활용 문제점을 해결하는 방법으로 Diomidis Spinellis의 분석기법에서 착안한 오픈소스 소프트웨어 프로젝트도큐멘테이션 가이드라인을 제시하여 문제항목 각각에 대한 도큐멘테이션 산정과 작성지침을 제안하였다.

3. 오픈 소스 도큐멘테이션 가이드 구성

3.1 오픈소스 프로젝트 개발 모형

오픈소스 소프트웨어 개발모형은 전 세계에 있는 수많은 우수한 개발자들이 직접개발과 Debugging과정에 참여를 통한 개발과정의 활동과 작업은 대개 모든 구조물에 대해서 동일한 순서로 반복적으로 사용되는 진화적 모형형태를 취하고 있으며 유지보수와 운영공정이 개발공정과 함께 병행적으로 사용될 수 있으며 획득, 공급, 지원 및 조정공정이 대개 개발공정과 병행하여 사용된다. 이같은 모형은 초기능력의 필요와 장기간 제한된 자원위임을 오픈소스 광범위한 참여그룹으로 극복하며 빠른 개발 사이클과 실 테스트를 거친 버그 리포팅 및 요구/패치를 통해 기술변화의 감지와 사용자피드백, 신속한 문제해결, 빨라진 성능개선 프로세스표, 상호 유연성이 뛰어난 오픈소스만의 장점을 가질수 있게 된다.

빠른 개발 사이클과 실 테스트를 거친 버그 리포팅 및 요구/패치를 위해 Bug Report, Patch, Support Request, Future Request과 각 단계별 도큐멘테이션의 구성을 통해 오픈소스 프로젝트 개발모형을 완성하게 된다.



3.2 오픈소스 프로젝트 도큐멘테이션 특징

일반적 소프트웨어 표준 도큐멘테이션 기능중 오픈소스 소프트웨어 특성분류는 아래와 같다.

기능	표준 설명	내용	오픈소스 특성
산출물 정의의 수단	단개별 산출물 정의와 보고용	각 단계별 산출물, 타당성조사 및 초기요구조건, 설계, 구현, 테스트 및 구현에 대한 계획	짧은 개발사이클을 위한 과도한 문서화는 지양, 실 개발지향의 문서화가 필요, 프로젝트 개요와 라이선트, 오픈소스 조건의 명기와 개발 계획 단계, 기본설계를 위주로 구성
진척 관리의 수단	관리자가 계획의 진행과정을 검토할 수 있도록 관련문서를 규정	개발계획, 각 단계에 대한 계획과 계획 변경에 대한 기록	일정을 위주로 한 로드맵은 참여자의 수와 능력 검증에 어려움으로 기능을 위주로 한 로드맵을 제시
업무 통신 수단	조직의 실무자와 시스템 개발자 사이에 상호 통신을 위한 수단	요구사항과 기능적인 명세, 데이터 저장과 프로그램 명세를 설명하는 설계 명세서	업무통신수단은 오픈소스 개발에 가장 중요한 요건으로 개요, 기능, 모듈에 대한 명세를 통해 커뮤니티 통신수단을 지원
책임 소재, 역할 분담	개발팅의 책임에 대한 명확한 구분과 역할을 기술	보안과 통제에 대한 정보책임에 대한 정의	기능위주에 역할 분담
품질 보증, 유지 보수	품질 보증계획과 기준	품질 보증계획, 기준, 시스템 기능의 규정	시스템 기능과 요구/버그 리포팅 업무통신수단에서 포함
교육 및 참고문헌	시스템 개발기능적인 명세	요구사항과 기능적인 명세	산출물 정의의 수단에서 포함

정의한 6가지 기능 중 오픈소스 프로젝트는 산출물정의 수단과 진척관리수단, 업무통신 수단으로써

의 문서화 작업이 다른 기능보다 중요성을 가지며 이로 인해 기능별 문서화와 단계별 문서화는 아래와 같이 정의한다.

기능	오픈소스 도큐멘테이션 기능 특징	산출물	중요도
산출물 정의 수단	1. 빠른 개발 사이클을 위한 과도한 문서화는 지양 2. 실 개발지향의 문서화 3. 프로젝트 개요와 라이선스, 오픈소스 조건의 명기 4. 개발 계획단계, 기본설계를 위주로 구성	프로젝트 개요서 요구사항기술서 화면 항목 설명서 테이블정의서 기능 규격서 프로시저정의서 모듈 명세서 소스코드명세서	상
진척 관리의 수단	1. 기능을 위주로 한 로드맵 편성	프로젝트개요서	중
업무 통신 수단	1. 오픈소스 개발에 가장 중요한 요건 2. 개요, 기능, 모듈에 대한 명세 3. 커뮤티티 통신수단을 지원	프로젝트개요서 화면레이아웃구성도 업무배경도 응용시스템구조도 기능규격서 모듈명세서 소스코드명세서	상
책임소재, 역할 분담	1. 기능위주에 역할 분담	프로젝트개요서 업무배경도 요구사항기술서	중
품질보증, 유지보수	업무통신수단에서 포함	프로젝트개요서 기능규격서	하
교육 및 참고문헌	산출물 정의 수단에서 포함	프로젝트개요서 기능규격서 모듈명세서	하

오픈소스 기능별 특성분류를 토대로 개발공정별 도큐멘테이션 산정은 다음과 같다.

공정	설명	산출물	중요도
계획	오픈소스 프로젝트의 목적, 범위, 제한 사항을 명세	프로젝트 개요서 라이선스 정의서	필수 필수
분석	개발영역에 대한 수행업무와 요구사항을 분석을 통한 업무절차 및 필요기능을 정의	업무배경도 요구사항기술서 화면레이아웃구성도 화면레이아웃설명서	선택 필수 선택 선택
기본 설계	분석을 통한 사항을 시스템 구현관점에서 어떻게 해결할 것인가를 결정, 기능별 기술	응용시스템구조도 테이블구조도 테이블정의서 프로시저정의서 프로시저명세서 기능규격서	필수 선택 선택 선택 선택 필수
상세 설계	물리적 설계안을 도출하고 프로시저를 프로그램으로	모듈명세서	필수

	전환하여 개별 프로그램의 모듈을 명세		
구현	프로그램을 원시코드로 구현	소스코드명세서	필수
완료			

4. 결론

소스코드에 자유로운 접근으로 경제성, 보안성, 확장성, 유지보수가 뛰어나므로 인해 세계각국에서 새로운 패러다임으로서 오픈소스 소프트웨어를 활성화하려는 정책이 추진되고 있다. 하지만 오픈소스가 지니고 있는 활용상에 문제들(기능의 불확실성, 기술지원의 불확실성, 탐색의 어려움, 문서화 미비, 로드맵 미비, 하드웨어 지원여부)로 기업은 오픈소스 채택을 주저하고 있으며 활성화 및 성공사례는 많지 않다.

본 논문에서는 오픈소스 활용상의 문제점을 해결하기 위해 오픈소스 소프트웨어 프로젝트 도큐멘테이션 가이드라인을 제시하며 문제항목 각각에 대한 도큐멘테이션 산정과 작성지침을 제안하였다.

오픈소스 개발모형을 분석 및 구성하고 소프트웨어 표준 도큐멘테이션 기능 중 짧은 개발 사이클과 실 테스트를 거친 버그 리포팅 및 요구/패치를 중심으로 오픈소스 소프트웨어 특성분류를 분석하고 특성분류를 통한 각 기능별, 각 단계별로 도큐멘테이션을 산정하였다.

참여자에게는 업무통신수단으로 관리자에게는 평가기준을 사용자에게는 오픈소스를 선택하는데 판단 기준이 되어줄 것이다.

참고문헌

- [1] 오픈소스 소프트웨어 연구보고서 2002.12
- [2] 기업의 오픈소스 소프트웨어 활용방안 2004.10
- [3] 한국형 오픈소스 소프트웨어 기술 개발 전략 2002.8
- [4] 정보통신분야의 소프트웨어 개발절차 표준화에 관한 연구 - SI업체를 중심으로 2005.8
- [5] TTA.KO-11.0001~ 0006 소프트웨어 문서 작성 지침 표준
- [6] TTA KO-11.0009 정보시스템 문서화 지침서 표준
- [7] TTA.IS-6592 컴퓨터기반 응용시스템문서화지침
- [8] "Guidance on implementin Open Source Software" Office of Government Commerce, 2002.9