





공개SW 입문 가이드

본 가이드는 공개SW를 처음 접하는 개발자들의 역량 향상과 교수자를 위한 정보 제공을 목적으로 한다.

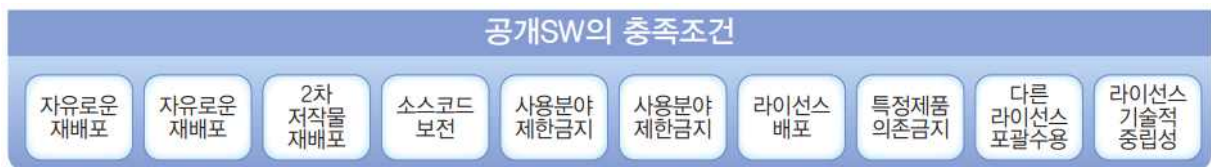
◆ 공개SW(Open Source Software)의 개념

◆ 공개SW 정의

- "오픈소스 소프트웨어" 또는 "오픈 소스" 등 그 명칭과 상관없이 소프트웨어(SW)의 저작권자가 해당 소스코드를 공개하여 이를 사용, 복제, 수정, 배포할 수 있는 권한을 부여한 SW를 말한다.

◆ 공개SW 조건

- 공개SW 활성화 및 인증을 관장하는 비영리 기관인 OSI(Open Source Initiative)에서 제시한 다음의 10가지 공개SW조건을 충족해야 한다.



◆ 공개SW 라이선스

- 공개SW의 사용, 복제, 수정, 배포와 관련하여 허용되는 권한 범위를 명시한 이용허락 조건을 말한다.

◆ 공개SW 개발방식

- SW의 소스코드를 공개하고 SW를 개발 및 유지 관리하는 전 과정에 최초 개발한 자 외에도 누구나 자유롭게 참여할 수 있도록 하는 개발방식을 말한다.

◆ 공개SW 커뮤니티

- 공개SW 개발 사업 또는 과제를 수행하기 위한 온오프라인 협업 공간으로서, 과제 개발 과정에서의 소스코드 공유, 의사소통 등을 위한 공간을 말한다.

공개SW(Open Source Software)

저작권은 존재하지만 저작권자가 소스코드를 공개하여 누구나 자유롭게 사용, 복제, 배포, 수정, 활용할 수 있는 SW를 지칭

공개SW 반대 개념은 상용SW가 아닌(소스코드)비공개SW



*OSI(Open Source Initiative) : 공개SW 이용 장려를 위해 만들어진 비영리단체

[그림 1] 공개SW의 정의 및 특징

개발효율성

- 공개SW는 SW개발을 위한 핵심 자산
- 상용SW의 96%가 공개SW 활용을 통해 개발기간 단축 및 비용절감 실현

기업 저변확보

- 소스코드 개방을 통해 SW저변 확산 용이



인재양성

- 공개SW 통한 SW교육의 효과적 활용 및 실무 SW기술 학습

[그림 2] 공개SW의 중요성

최신기술 도입

- 소스코드 원천기술에 대한 접근으로 선진SW기술습득 용이
- 개방적인 SW개발문화에 따른 기술혁신과 아이디어 수용 가능

벤더 종속성 탈피

- 국내 시장에서의 독과점 SW 대체재 역할

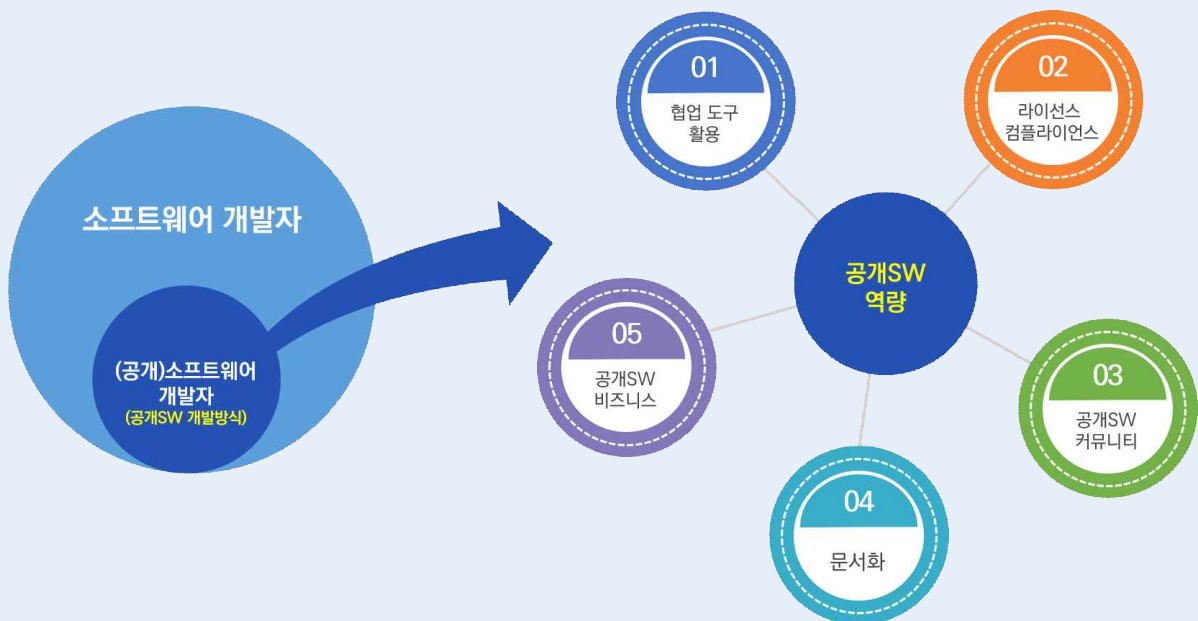
[출처 : 공개소프트웨어 연구개발 수행 가이드라인, 과학기술정보통신부(2018)]

공개소프트웨어 역량 (Open Source Software Competency, OSSC)

◆ 공개SW 역량이란?

공개소프트웨어는 소프트웨어의 한 종류가 아니라 소프트웨어 그 자체이며, **소프트웨어 개발자**라 함은 공개소프트웨어를 포함한 포괄적 개념이다.

따라서 별도의 공개SW 개발자를 구분하고 있지는 않으며, **공개SW 개발 방식**으로 개발하는 모든 소프트웨어 개발자를 '**공개SW 개발자**'로 정의하고자 한다. 이를 바탕으로 소스코드를 공개하고 누구나 자유롭게 참여할 수 있도록 하는 '**공개SW 개발방식**'의 특징에 따라 예비 개발자가 이해하고 갖추어야 할 역량(지식&스킬(Skill))을 '**공개SW 역량**'으로 정의하고 소개하고자 한다.



◆ 공개SW 역량 요소

1 (지식&기술) 협업 도구 활용

(공개SW 개발방식) SW의 소스코드를 공개하고 SW를 개발 및 유지 관리하는 전 과정에 최초 개발한 자 외에도 누구나 자유롭게 참여할 수 있도록 하는 개발방식이며, (협업도구) 소스코드 공개 방법은 일반적인 공개SW라면 누구나 쉽게 접근 가능하고, 기본적인 소스코드 관리가 용이한 공개된 저장소(Repository) 중 한 곳을 선택하여 공개한다. 전 세계적으로 가장 많은 공개SW 프로젝트가 관리되고 있는 협업도구는 깃허브(Github)가 대표적이며 이러한 협업방식에 대한 이해와 사용능력을 필요로 한다.

2 (지식) 라이선스 컴플라이언스

공개SW는 소스코드를 공개하여 누구나 자유롭게 사용, 복제, 수정, 재배포가 가능하지만, 지적재산권으로 보호되는 저작물이다. 따라서 소프트웨어 개발자와 사용자 간의 사용방법 및 조건의 범위를 명시한 라이선스 컴플라이언스에 대한 이해를 필요로 한다.

3 (지식&활동) 커뮤니티

공개SW는 그 특성상 소스코드를 공개하고, 많은 사람들의 동참이 필수적이며 활발한 개발자 및 사용자의 참여인 커뮤니티를 통해서 개선되고 증진되고 있다. 커뮤니티는 새로운 기술을 배울 수 있는 공간이자 새로운 사람을 만날 수 있는 장이지만, 막 프로그래밍을 접하는 사람들에게 공개SW 커뮤니티는 낯선 공간이다. 따라서 공개SW 커뮤니티에 대한 이해가 필요하며 개발자로 지속적인 성장하기 위해 커뮤니티 활동이 요구된다.

4 (지식&기술) 문서화

공개SW 활용에 가장 큰 난관은 불충분한 문서이며, 좋은 문서가 인기 공개SW 프로젝트나 커뮤니티를 만들며 보다 다양한 사람들이 참여할 수 있도록 하기 위해서는 문서를 잘 만들어 놓는 것이 중요하다.

5 (지식&활동) 비즈니스

과거 무료로 인식되었던 자유 소프트웨어(Free Software, 오픈소스의 초기 형태)와 달리 공개SW(Open Source)는 상용화를 위한 다양한 비즈니스 모델들이 등장하였다. 공개SW를 활용하여 수익 창출이 가능한 방안들이 등장하면서 공개SW 확산이 가속화되고 이를 활용하는 기업들이 증가하면서, 공개SW 비즈니스 방식에 대한 이해가 필요하다.

1. 협업 도구 활용

◆ 대표적인 협업 도구 _ Github

깃허브(Github)는 깃(Git*) 원격 저장소이자 웹 그래픽 기반 환경에서 깃 도구를 편하게 이용할 수 있도록 만든 호스팅 서비스로, 전 세계 개발자들이 함께 공개SW 프로젝트의 소스코드를 수정하고 참여하며 공개SW 프로젝트 확산에 기여하고 있다.

* 깃(Git)은 2005년에 개발된 공개SW로 분산형 버전관리 시스템(DVCS)으로, 리누스 토발즈와 주니오 하마노가 개발

깃허브(Github)는 프로그램의 소스코드 뿐만 아니라 설치 관련 문서, 활용 방법, 번역 페이지도 등록가능하고, 깃북(gitbook)을 통해서 전자 출판으로 활용할 수도 있으며 소스코드 검색 및 관련 통계도 제공된다. 또한, 깃허브(Github)에는 다양한 공개SW 프로젝트들이 있다.

【 깃허브(Github) 흐름의 이해 및 주요 용어 】



- Pull Request 과정 -

- Watch : 해당 프로젝트의 관찰 기능, 각 이력들에 대한 알림제공
- Star : 해당 프로젝트에 대한 관심, 북마크와 같은 기능
- Fork : 내 Branch로 메인 프로젝트 코드 저장소 복사본 생성
- Issue : 기능을 논하거나 버그 추적하는 대화 공간
- Pull request : Fork한 Branch의 작업완료 후 Master Branch로 병합 요청
- Commit : 변경 내용 저장
- Branch : 테스트를 하거나 새로운 기능을 개발하기 위한 독립된 저장소
- Contributor : 오픈소스 프로젝트 기여자

[출처 : 공개소프트웨어 연구개발 수행 가이드라인 & 깃허브]

Key Point

깃허브(Github)의 주요 용어 및 흐름을 이해하고 사용할 줄 알아야 한다.

2. 라이선스 컴플라이언스

◆ 라이선스 컴플라이언스

일반 상용 소프트웨어와 마찬가지로 공개SW에도 저작권 등 지적재산권이 있다. 그래서 권리자의 허락 없이 함부로 사용하면 소송을 당할 수 있다. 그런데 공개SW의 권리자들은 되도록 많은 사람이 자유롭게 사용할 수 있도록 광범위한 라이선스를 부여하고 있다. 예를 들어 사용자들에게 사용에 대한 권리뿐만 아니라 마음대로 복제 및 배포를 할 수 있도록 하고, 소스 코드까지 제공하여 마음대로 수정할 수 있도록 허락한다. 하지만 상용 소프트웨어처럼 그에 따르는 로열티를 요구하지는 않으며, 대신 몇 가지 지켜야 할 의무사항을 요구한다.

따라서 공개SW를 이용하려면 공개SW 개발자가 만들어놓은 조건의 범위에 따라 해당 소프트웨어를 사용해야 하며, 이를 위반할 경우에는 라이선스 위반 및 저작권 침해로 이에 대한 법적 책임을 져야한다.

【 소스코드 공개 여부에 따른 라이선스 비교 및 주요 라이선스 특징 】

구분	무료 라이선스	유료 라이선스
공개	대부분 OSS (Apache, Tomcat, JBoss)	일부 배포판 OSS (Red Hat, MySQL)
비공개	Freeware(Winzip, RealAudio) Shareware(초기에만 무료) Adware(JetAudio)	상용 비공개 소프트웨어 (MS Office, Oracle, Shareware)

구분	무료 이용가능	배포 허용가능	소스코드 취득가능	소스코드 수정가능	2차적 저작물 재공개 의무	독점SW와 결합가능
GPL	●	●	●	●	●	X
LGPL	●	●	●	●	●	●
MPL	●	●	●	●	●	●
BSD license	●	●	●	●	X	●
Apache llicense	●	●	●	●	X	●

[출처 : 공개SW포털(oss.kr) 공개SW라이선스(정보통신산업진흥원)]

Key Point

공개SW 라이선스를 이해하고 각 라이선스의 주요 특징을 알아야 한다.

3. 커뮤니티

◆ 커뮤니티

공개SW 커뮤니티 참여자에는 개발자, 관리자, 사용자 등이 있으며, 개발자 간 소통을 위해서는 커뮤니티 참여자의 유형에 따라 편리하게 사용할 수 있는 메일링 리스트, 포럼, 버그리포트, 프로젝트 관련 문서, FAQ 등을 준비해야 한다. 이러한 대응 자료가 준비되고 커뮤니티 참여자들과 소통을 강화해야 커뮤니티 조직이 구성되며, 버그 리포트에 대한 신속한 대응과 사용자 및 참여 개발자의 문제점을 안정적으로 해결함으로써 커뮤니티는 활성화된다. 커뮤니티는 해당 프로젝트의 문제 및 요구사항을 스스로 해결할 수 있는 핵심 주체이다.

커뮤니티는 새로운 개발자와 사용자가 외부에서 유입되기 때문에 그들이 본 프로젝트에 어떻게 기여해야 하는지에 대한 가이드라인과 행동강령에 따라 운영되고 있으며, 온·오프라인을 통해 참여할 수 있다.(주로 온라인(페이스북 등)을 통해 참여)

【 (예)기여자 행동 강령 규약(Contributor Covenant) 한국어 버전 】

[서약] 개방적이고 친근한 환경 조성을 위해, 기여자와 유지자는 프로젝트와 커뮤니티에서 연령, 신체 크기, 장애, 민족성, 성 정체성과 표현, 경력, 국적, 외모, 인종, 종교 또는 성적 정체성과 지향에 관계없이 모두에게 차별없이 참여할 것을 서약합니다.

[표준]

긍정적인 환경을 조성하기 위해 기여자가 해야 할 행동은 다음과 같습니다.

- 소외하지 않고 배려하는 언어 사용
- 서로 다른 경험과 관점 존중
- 열린 마음으로 건설적인 비판을 수용
- 커뮤니티에 가장 최선이 무엇인지에 주력
- 다른 커뮤니티 구성원들에 대한 공감 표현

하지 말아야 할 행동은 다음과 같습니다.

- 성적인 언어와 이미지 사용, 원치 않는 성적 관심이나 접근
- 소모적인 논쟁, 모욕적이거나 비하하는 댓글과 개인적 또는 정치적인 공격
- 공개적이거나 개인적인 괴롭힘
- 동의 없는 집주소 또는 전자주소 등의 개인 정보의 공개
- 부적절한 것으로 간주될 수 있는 다른 행위

[출처 : 공개소프트웨어 연구개발 수행 가이드라인(과학기술정보통신부)]

Key Point

공개SW 커뮤니티에 대한 이해와 운영 정책을 알아야 하며 커뮤니티 참여를 위해 행동 강령(규범)을 잘 준수해야 한다.

4. 문서화

◆ 문서화

공개SW 프로젝트를 할 때 만나는 문제와 관련된 질문에서 '미완성 혹은 친절하지 않게 작성된 문서'가 가장 큰 문제로 꼽혔다. 문서화는 해당 프로젝트에 대한 사용 방법은 물론, 기여하는 방법, 사용 및 기여 조건, 커뮤니티 행동 기준 등에 대한 지침을 제공한다. 따라서 문서화가 잘 되어 있어야 프로젝트를 활용하고, 기여하는 사람이 늘어날 수 있다. 개발문서의 기본이 되는 README 문서부터 잘 정리하는 것이 중요하다.

【 README 문서 작성 시 고려할 사항 】

README 문서는 다른 사람들이 프로젝트를 발견했을 때 제일 처음 접하는 문서이다. 대부분의 사용자는 README 문서를 읽는 것만으로 이 프로젝트를 좀 더 탐색할지, 다른 프로젝트를 찾아볼지 결정한다.

README 문서를 작성할 때 특별히 준수해야 하는 규칙은 없지만 사용자의 이해를 돕기 위해 다음과 같은 내용은 필수로 포함하는 것이 좋다.

- **프로젝트의 목적 및 용도** : 프로젝트에 대해 간단하게 설명하는 내용을 포함한다.
 - 이 프로젝트는 무엇을 위한 것인가?
 - 어떤 문제를 해결할 수 있는가?
 - 왜 이 프로젝트가 유용한가?
 - 어떤 사람들이 이 프로젝트를 사용하면 좋은가?
 - 이 프로젝트는 어떻게 작동하는가?
- **프로젝트를 시작하는 방법** : 프로젝트를 처음 사용하기 위해 필요한 내용을 포함한다.
 - 오픈소스 소프트웨어를 설치, 사용하기 위해 필요한 전제조건이 있는가?
 - 어떻게 설치, 사용, 테스트하는가?
 - 설치 가이드 문서는 어디에 있는가?
- **저작권, 라이선스 정보** : 프로젝트의 사용 범위 및 조건을 설명하는 내용을 포함한다.
 - 어떤 라이선스로 배포되는가?
 - 상세한 라이선스 정보는 어디에서 확인할 수 있는가?
 - 오픈소스 소프트웨어를 사용함에 있어 제약 조건(특허, 상업적 사용에 대한 제약)이 있는가?
- **외부 리소스 정보** : 프로젝트 내에 포함된 외부의 코드나 리소스의 정보를 포함한다.
 - 각각의 출처 및 배포 라이선스는 무엇인가?

[출처 : 네이버 오픈소스 가이드]

Key Point

좋은 문서 작성을 위해 프로젝트 목적 및 용도, 시작 방법, 라이선스 정보 등을 바탕으로 누구나 쉽게 이해할 수 있는 문서를 작성해야 한다.

5. 비즈니스

◆ 비즈니스

공개SW 모델의 적용은 기업 비즈니스에 많은 전략적 이점들을 제공할 수 있으며, 다양한 방법으로 회사 또는 조직 운영에 영향을 줄 수 있다.

일반적으로 3가지 '개발모델', '서비스 제공 모델', 개발과 서비스를 병행하는 '하이브리드 모델'로 분류 할 수 있다. 다만, 여기에서의 구분은 모델들 간의 경계를 의미하는 것이 아니며 실제 많은 기업들은 몇 개의 모델을 조합하거나 연속적인 모델로 활용하고 있다.

【 공개SW기반 주요 비즈니스 모델】

1. 개발모델

회사가 특정한 프로젝트의 기획자 또는 리더로서 오픈소스 소프트웨어를 개발하는 모델이다. 이러한 기업들의 경우 커뮤니티를 이끌고 관리하며, 그 과정에서 보유 기술의 확산, 시장의 확대뿐만 아니라 커뮤니티 멤버들의 자발적 기여를 통한 기술의 지속적 혁신이라는 기회 요인을 획득하는 것이 주요 전략이다. 그러나 기업의 수익 또한 중요하기 때문에 대개는 커뮤니티를 위한 오픈소스 제품과 수익 창출을 위한 상업적 제품을 균형 있게 유지한다. 이러한 모델은 관련 제품의 매출을 촉진하는 것 외에도 해당 기술을 사실상의 표준으로 자리 잡게 하거나, 제품의 잠재적인 고객을 포함한 청중들의 공감대를 형성하거나, 유지보수 원가를 절감하는 등의 장점이 있다.(사례 : MySQL, 시스코, 샌드메일 등)

2. 서비스 제공 모델

단순히 소프트웨어가 무료라고 해서 그것이 바로 그 소프트웨어가 모든 사람들에게(예를 들어, 기술적인 측면에서) 접근 가능하다는 것을 의미하지는 않는다. 기업의 입장에서는 적절한 오픈소스 소프트웨어의 선정, 설치, 훈련 및 지원이라고 하는 과업이 항상 필요하기 마련이고, 따라서 만일 라이선스 구매 예산을 서비스의 개선에 지출할 수 있게 된다면 보다 더 나은 서비스를 기대할 수 있기 때문에 고객의 입장에서 훨씬 매력적일 수 있다. 서비스 제공 모델은 규모 및 솔루션의 세분화, 산업분야의 세분화, 맞춤형 개발/선정/컨설팅/통합/교육/훈련/기술지원/유지보수 등 특정 서비스에 대한 전문화를 통한 여러 가지의 세부적인 모델들 존재할 수 있는데, 대부분의 경우 기업들마다 가능한 서비스들을 조합하여 비즈니스를 영위하는 경우가 많다.(사례 : 소스랩스, 와이드 오픈소스, 스파이크, 레드햇 등)

3. 개발·서비스 혼합(하이브리드) 모델

개발한 제품을 오픈소스 소프트웨어로 공개하고 수익은 서비스를 통해 창출하는 유형이다. (사례 : 알프레스코(CMS 솔루션), 콩피에르(ERP, CRM 솔루션), V타이거 및 오픈브라보 등)

[출처 : 오픈소스 소프트웨어 비즈니스모델(김중배)]

Key Point

비즈니스 패러다임 변화를 이끌고 있는 공개SW 비즈니스 모델을 이해해야 한다.

📖 학습 정보 제공

◆ 정보 제공

구분	웹사이트 정보
공개SW 개요	• 공개SW포털 : https://www.oss.kr/oss_intro
Github 사용법	• 깃허브 : https://guides.github.com/
공개SW 라이선스	• 공개SW포털 : https://www.oss.kr/oss_license
공개SW 커뮤니티	• 공개SW포털 : https://www.oss.kr/community_support
문서화	• 네이버 : https://naver.github.io/OpenSourceGuide/book/
공개SW 비즈니스	• 공개SW협회 : http://kossa.kr/

◆ 온라인 교육

구분	웹사이트 정보
공개SW 개요	• OLC센터 : https://olc.kr/course/course_online_view.jsp?id=246&cid=518
Github 사용법	• OLC센터 : https://olc.kr/ 📖 Git/Github 입문
공개SW 라이선스	• OLC센터 : https://olc.kr/ 📖 오픈소스 라이선스 이해와 관리
공개SW 비즈니스	• OLC센터 : https://olc.kr/ 📖 공개SW 비즈니스

📖 공개SW 포털

- 공개SW 정보 제공 및 교육, 개발자 지원, 커뮤니티 지원, 라이선스 검증 및 창업 컨설팅 등 다양한 정보 제공

※ 자세한 정보(<https://www.oss.kr>)

📖 공개SW 개발자대회

- 창의적 예비 개발자 양성을 위해 매년 개최되고 있으며, SW비전공 대학생·일반인 및 공개SW를 처음 접하는 개발자 등을 위해 온라인 교육을 제공하며, 맞춤형 멘토링, 라이선스 검증 등 대회 전 과정을 통해 공개SW 개발과정을 경험하고 프로젝트 경쟁력을 제고할 수 있는 대회

※ 자세한 정보(https://www.oss.kr/dev_competition)

참고문헌 및 자료 출처

- 기업이 알아야 하는 공개SW 라이선스 및 거버넌스의 이해(2017.8/NIPA 공개SW역량프라자박준석 수석)
- 공개소프트웨어 표준 해설서(2015.11/한국정보통신기술협회(TTA))
- 오픈소스 소프트웨어 활성화를 위한 성숙도 및 적용성 평가 모델(OSM AAM)의 설계 및 구현에 관한 연구(2013.봄호/정보화정책)
- 공개소프트웨어 연구개발 수행 가이드라인(2018.6/과학기술정보통신부 &정보통신산업진흥원)
- 공공 공개SW 거버넌스 가이드(2020.1/정보통신산업진흥원)
- 오픈소스 소프트웨어 라이선스 가이드 3.0(2016.11/문화체육관광부&한국저작권위원회)
- 오픈소스 소프트웨어 비즈니스모델(2017, 김종배)
- 네이버 오픈소스 가이드(<https://naver.github.io/OpenSourceGuide>)
- 오픈AIP 및 문서화 방안(네이버(2017) 옥상훈)
- 공개SW 포털(<https://www.oss.kr/>)
- 깃허브(<https://guides.github.com/>)

「공개SW 입문 가이드」는 공개SW 역량 향상 및 개발자 교육을 위한 효율성 제고를 위한 목적으로 작성되었습니다.

- 발행일: 2020년 12월 10일
- 발행처: (사)한국공개소프트웨어협회 / 정보통신산업진흥원

본 자료를 인용하거나 활용 할 때에는
출처를 명시하여 주시기 바랍니다.

※ 기타문의 : kossa@kossa.kr / 02-599-7917