

# 오픈소스 연구개발의 성공을 위한 전략 : Next Level 성장 가이드라인

---

Kevin Kim

[hckim@invesume.com](mailto:hckim@invesume.com)



김형채

- 개방형OS 하모니카 프로젝트 리더
- 오픈소스 전문기업 인베숨 대표
- 한국정보통신기술협회 공개소프트웨어 분과 위원

<https://hamonikr.org>

### HamoniKR 6.0 Taebaek

ISO 다운로드

서버 #1 : HamoniKR 6.0 Taebaek Lite [↓](#)

서버 #1 : HamoniKR 6.0 Taebaek [↓](#)

서버 #2 (aws) : HamoniKR 6.0 Taebaek Lite [↓](#)

서버 #2 (aws) : HamoniKR 6.0 Taebaek [↓](#)


SHA256(Lite) : 5668dbf16505a72194daf9531b7f4db262b751d4bb20d48d4d4bd911e4599bff

SHA256 : 9b684c2ee2fa61285a56adc4f85ca84ec549de5267a92efcf0e7b63687618d11

#### 최소사양

- CPU(x64) : 하모니카는 모든 최신 컴퓨터에 권장됩니다
- 램 : 2GB RAM 이상
- 하드디스크 : 10G 이상(Lite) / 20G 이상
- 해상도 : 800x600

[새로운 기능](#) [릴리즈 노트](#)



하모니카 OS 누적 다운로드 수  
[↓](#) **200,757**

The screenshot shows the GitHub profile page for the user 'hamonikr'. At the top, there is a navigation bar with a search box and a menu icon. Below the navigation bar, the user's profile information is displayed, including their profile picture, name 'hamonikr', and bio 'Hamonikr Project - managed by invesome'. The profile also shows 27 followers, location 'Korea, Seoul', and contact information for website, Twitter, and email. Below the profile information, there is a 'Pinned' section with a 'Customize pins' link. The pinned repositories are listed in a grid:

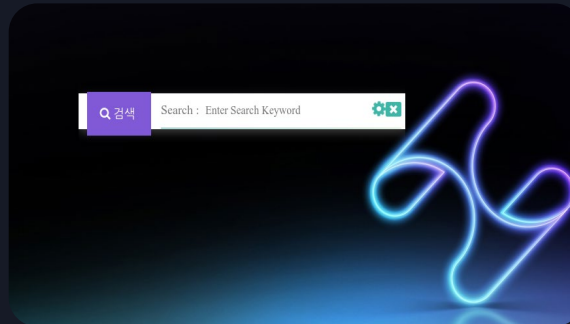
- nimf** (Public): 다국어 입력기 프레임워크 Nimf is a lightweight, fast and extensible input method framework. Language: C. Stars: 124, Forks: 18.
- hamonize** (Public): 비대면 시스템 관리 솔루션 Remote pc management system. Language: JavaScript. Stars: 31, Forks: 23.
- democratization-expertise** (Public): 개방형OS 기술지원 사이트 Democratization of Expertise. Language: JavaScript. Stars: 19, Forks: 12.
- systemback** (Public): Easy to use backup and recovery on modern Ubuntu-based systems. Language: C++. Stars: 5, Forks: 5.
- hamonikr-apphonia** (Public): 화상회의 프로그램. Language: Vue. Stars: 2, Forks: 59.
- hamonikr-onboard-layout-ko** (Public): Korean Keyboard Layout for onboard. Stars: 5.

# 주요 프로젝트

개방형OS 하모니카, 화상회의 SW Hamonia, 클라우드 문서검색 프로그램 hamonikr-finder, 원격 데스크탑 접속 프로그램 hamonikr-xrdp, 다국어 입력기 nimf, 사용자 기술지원 hamonikr-support 등 다수의 프로젝트 운영 중.



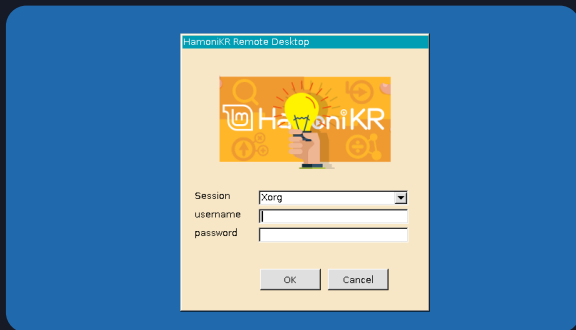
개방형 OS 하모니카



클라우드 검색



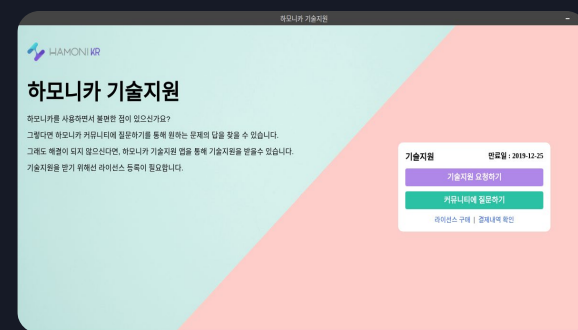
플러그인 필요 없는 화상회의



원격 데스크탑 접속



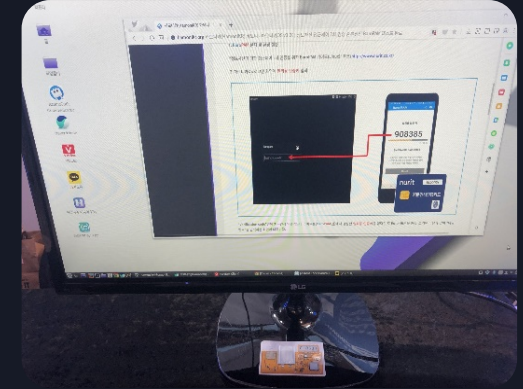
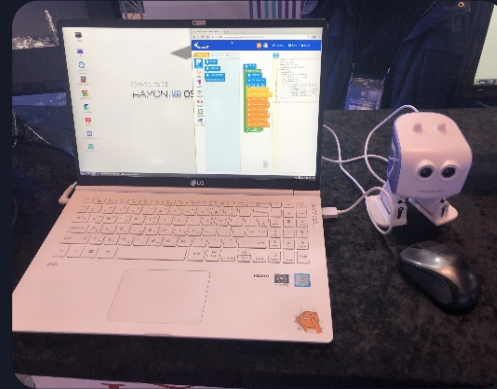
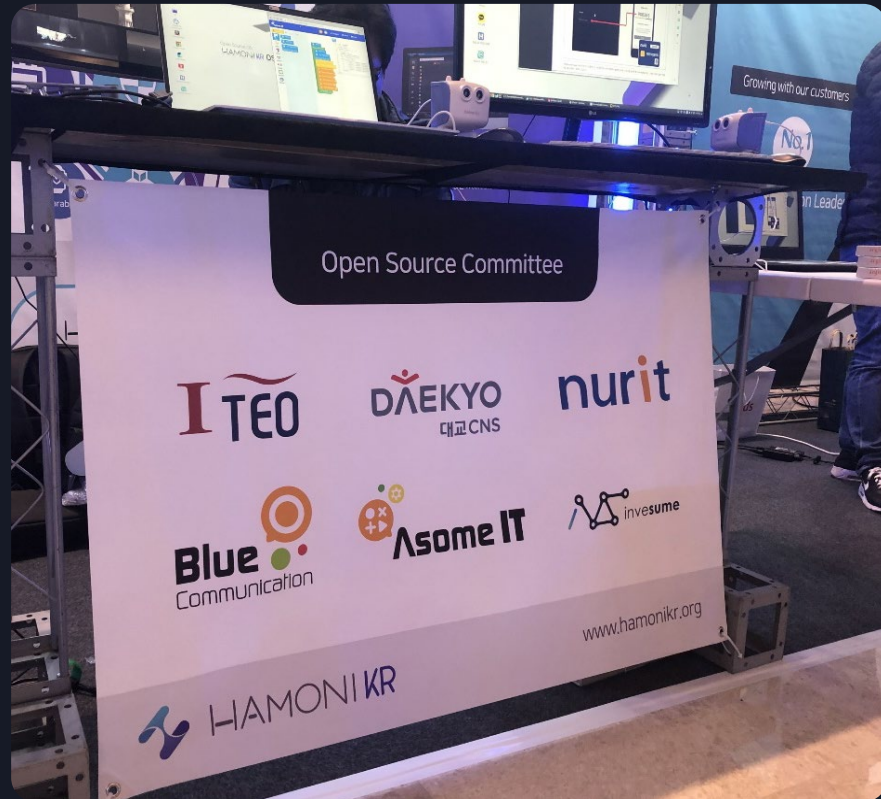
시스템 모니터링



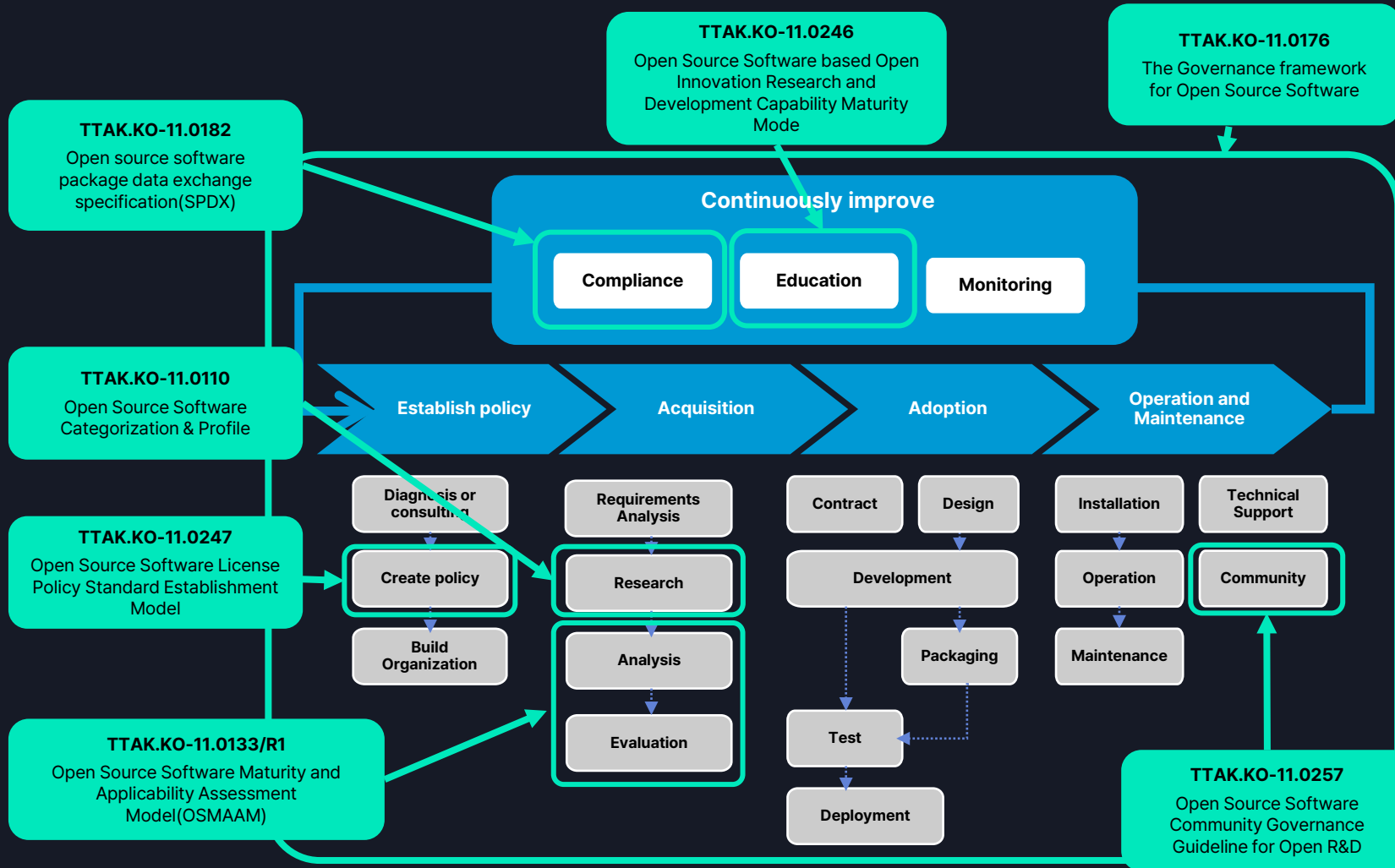
사용자 기술지원

# HamoniKR Community

하모니카 커뮤니티는 오픈소스를 활용하여 제품을 개발하거나, 오픈소스를 보급 확산하고 사용자 지원을 제공하는 많은 기업들과 협력하여 국내 오픈소스 기반의 소프트웨어들이 확산되도록 노력하고 있습니다.



# Standardization



## 한국정보통신기술협회(TTA)

- 공개소프트웨어 보안취약점 관리 지침
- 개방형 데스크톱 운영체제 규격 1.0
- 개방형 연구개발을 위한 공개소프트웨어 커뮤니티 거버넌스 지침
- 공개소프트웨어 기반 개방형 혁신 연구개발 역량 성숙도 모델
- 공개소프트웨어 라이선스 정책 수립 모델
- 공공 공개소프트웨어 거버넌스 지침
- 공개소프트웨어 성숙도 및 적용성 평가 지침
- 오픈소스 라이선스 분류 기준
- 공개소프트웨어 거버넌스 프레임워크
- 공개소프트웨어 정보교환명세
- 공개소프트웨어 분류체계 및 프로파일



오픈소스SW 연구개발 프로젝트

# 오픈소스에 진심인 글로벌 기업들

## finviz S&P 500 map



## Open Source Contributor Index

RANK	ORGANIZATION	ACTIVE CONTRIBUTORS	TOTAL COMMUNITY
1	Google	4733 +297	8745 +468
2	Microsoft	3202 +282	6312 +487
3	Red Hat	2616 +162	3857 +161
4	Intel	1471 +122	3135 +202
5	Amazon	1294 +113	2725 +202
6	IBM	1073 +107	2314 +148
7	VMware	660 +51	1055 +47
8	Oracle	619 +167	1154 +194
9	SAP	600 +40	1032 +58
10	GitHub	541 +43	1143 +69

Source : <https://opensourceindex.io/>



# WHY

## 혁신 촉진

- 다양한 개발자들의 참여로 새로운 아이디어와 접근 방식을 얻을 수 있다.
- 개방된 환경에서 빠르게 변화하고 성장하는 기술 트렌드를 따라잡을 수 있다.

## 개발 비용 절감

- 커뮤니티의 기여로 인해 개발 리소스와 시간을 절약할 수 있다.
- 기존의 오픈소스 라이브러리나 도구를 활용하면 개발 과정을 가속화할 수 있다.

## 품질 향상

- 전 세계 수많은 개발자들이 코드를 검토하고 테스트하므로 버그를 더 빨리 찾고 수정할 수 있다.
- 코드의 투명성이 높아지므로, 품질 관리와 표준 준수가 강화된다.

## 커뮤니티 구축 및 브랜드 강화

- 오픈소스 프로젝트는 기업의 브랜드 인지도와 신뢰도를 높일 수 있다.
- 커뮤니티와 협력하면서 고객과 시장과의 연결고리를 강화할 수 있다.

## 기술 표준 활용

- 오픈소스 프로젝트를 통해 업계 표준을 주도하고 영향력을 확장할 수 있다.
- 다른 기업과 협력하여 공통된 기술 기반을 만들 수 있다.

## 재사용 및 상호 운용성

- 오픈소스 코드는 재사용이 용이하며, 다양한 시스템과의 통합이 쉽다.
- 표준화된 접근 방식으로 인해 다른 오픈소스 솔루션과의 호환성이 향상된다.

# 국내 오픈소스SW 방식의 연구개발 프로젝트

국내에서는 2014년부터 SW연구과제 대상 오픈R&D 방식 적용 중이며, 2020년 소프트웨어진흥법이 개정되어 많은 국가연구개발과제가 오픈R&D 방식으로 진행.

- 스타랩 과제
- SW기초연구센터 과제
- 차세대정보통신기술개발 과제
- 인공지능 국가프로젝트
- 공개SW 기술개발 지원사업
- 개방형OS 환경개발 및 보급확산 사업

▲국가연구개발사업 결과물 공개(3장25조2항): SW 연구 및 기술개발 촉진을 위해 국가 연구개발 사업의 결과물을 공개SW로 배포하게 했다. 공개SW는 저작권자가 원시코드를 공개해 활용, 복제, 수정, 재배포가 자유로운 소프트웨어를 말한다.

# 전통적 연구개발 프로젝트와 차이점

## 라이선스와 지적재산권

- OSS 프로젝트: 오픈소스 라이선스 하에 공개되며, 코드는 공개적으로 이용, 수정, 배포될 수 있다.
- 일반 프로젝트: 특정 기업이나 조직이 독점적으로 소유하고 관리하며, 지적 재산권이 엄격히 보호된다.

## 개발과 협업 모델

- OSS 프로젝트: 종종 분산된 개발자들과 협업하며, 커뮤니티 기반의 개발이 일반적이다.
- 일반 프로젝트: 보통 내부 팀 또는 특정 파트너와의 협업으로 이루어진다.

## 투명성 및 접근성

- OSS 프로젝트: 코드, 개발 프로세스, 문서 등이 대개 투명하고 공개적으로 접근할 수 있다.
- 일반 프로젝트: 해당 정보는 내부적으로 보호되며, 외부에 공개되지 않는 경우가 많다.

## 지속성 및 유지보수

- OSS 프로젝트: 커뮤니티 참여로 인해 프로젝트의 지속성이 보장될 수 있으며, 다양한 기여자들이 유지보수에 참여한다.
- 일반 프로젝트: 특정 기업이나 조직이 지속성과 유지보수를 책임진다.

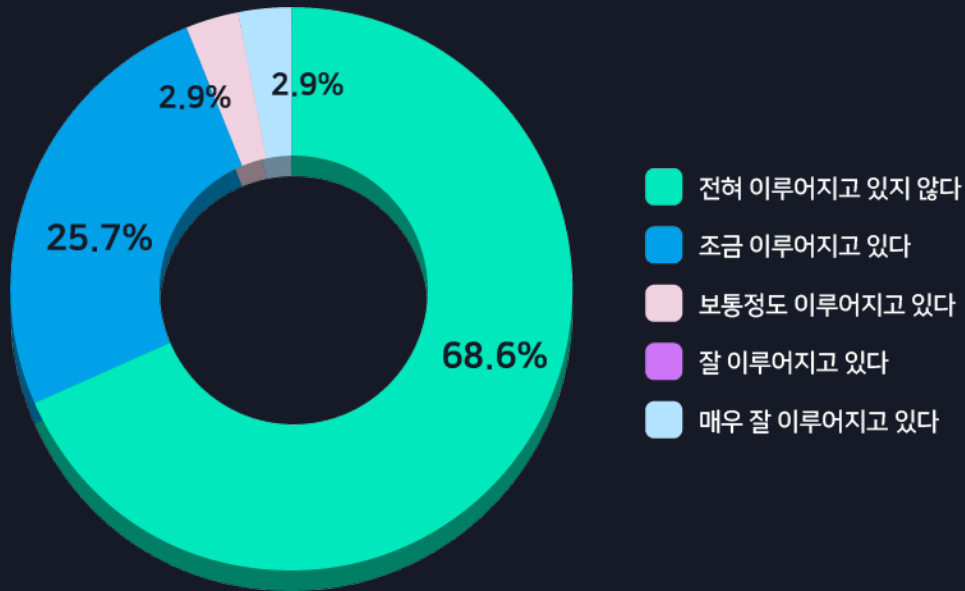
## 비즈니스 모델

- OSS 프로젝트: 수익 모델은 서비스, 컨설팅, 특정 기능의 상용 버전 판매 등 다양하게 구성될 수 있다.
- 일반 프로젝트: 제품 판매, 라이선스 비용, 구독 모델 등을 통해 수익을 창출한다.

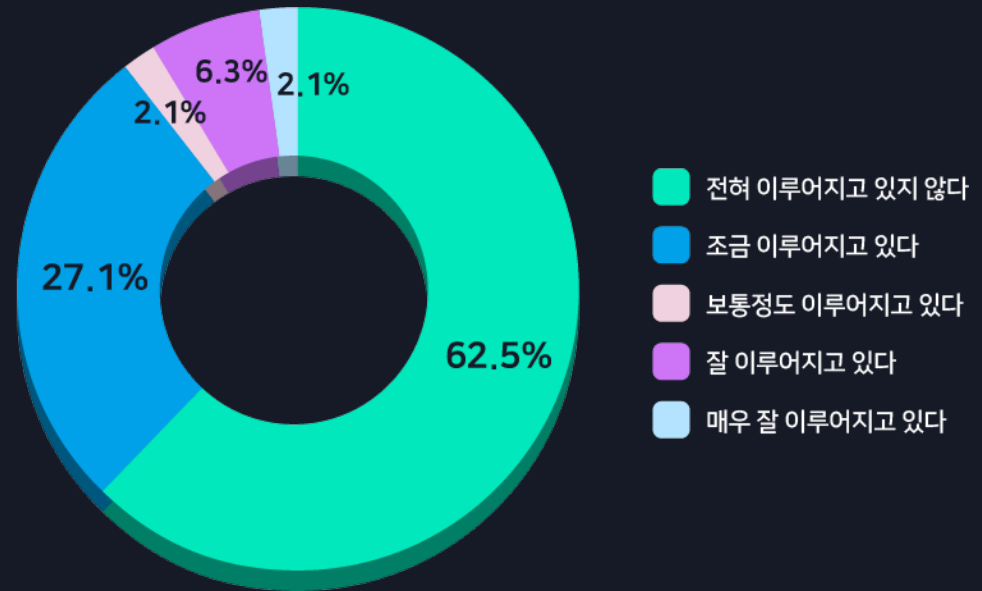
# 국내 공개SW 방식 연구개발 애로사항

2018년 정보통신기획평가원의 오픈R&D 수행기관 34개를 대상으로 국내 오픈R&D 현황을 조사한 결과를 보면 공개SW 담당자의 94.3%는 오픈소스 연구개발 관리 정책이 부족하다고 답변.

또한 89.6%는 공개SW 방식에 적합한 연구개발 프로세스가 적용되고 있지 않다고 답변.



공개SW 연구개발 과제의 관리 정책



공개SW 연구개발 과제의 연구개발 방식

# GOAL

기업에서 오픈소스SW 연구개발 프로젝트의 연구 책임자와 참여하는 연구원들에게 프로젝트를 성공적으로 수행하기 위해 필요한 전략과 가이드라인을 제시.

- 프로젝트 수행을 위한 사업계획서를 작성할 때 고려할 점
- 프로젝트 수행에 필요한 개발환경
- 오픈소스SW의 공급망 관리
- 프로젝트의 성과 측정 및 관리
- 프로젝트 상태를 진단해 볼 수 있는 자가 점검 방법



URL : <https://github.com/iitp-rnd/oss-guideline>

# 2

오픈소스 연구개발 프로젝트  
성숙도 관리 방안

# Open R&D Capability Maturity Model

정보통신단체표준(TTAK.KO-11.0246) :

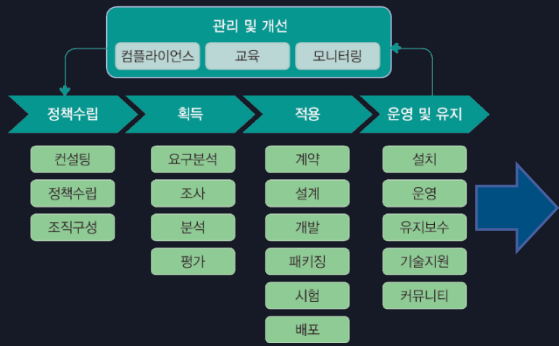
본 표준의 주요 내용은 개방형 혁신 연구개발의 역량 성숙도를 평가하고 개방형 혁신 연구개발 활동의 능력 수준을 개선하기 위해 필요한 성숙도 모델을 정의한다. 이를 위해, 역량 성숙도 등급(Capability Maturity Level), 성숙도 모델 프로세스가 적용되는 도메인(Domain), 도메인별 세부 등급 기준 등을 다룬다.



# Open R&D Capability Maturity Model

공개SW 거버넌스 프레임워크에서 오픈소스 SW방식의 연구개발 과제에서 필요한 활동요소 선정

## 25개 오픈R&D 활동요소



공개SW 거버넌스 프레임워크

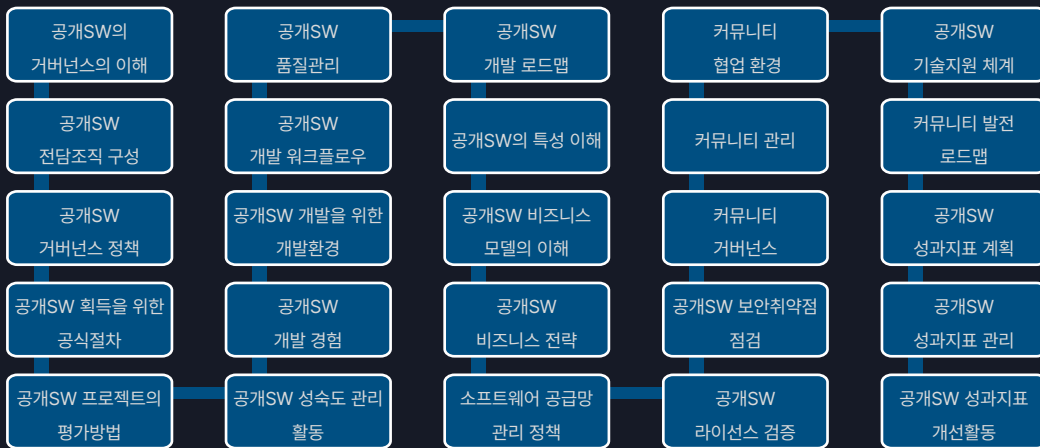




# Open R&D Capability Maturity Model

오픈소스 SW 연구개발에 필요한 활동 요소들을 도메인 별로 재정의하고 해당 도메인의 평가방법을 정의

25개 오픈R&D 활동요소



<b>비즈니스 전략</b>	오픈소스SW 기반의 비즈니스 전략
<b>정책 및 조직</b>	오픈R&D 거버넌스 정책과 조직의 구성
<b>프로젝트 평가</b>	오픈소스SW 프로젝트의 성숙도 평가
<b>공급망 관리</b>	오픈소스SW가 포함되는 소프트웨어 공급망 관리
<b>커뮤니티</b>	오픈소스 커뮤니티 거버넌스
<b>개발환경</b>	오픈소스SW 개발을 위한 개발 환경
<b>성과관리</b>	오픈R&D에 적합한 성과지표

# Business Strategy

연구개발 결과물을 활용한 사업화가 가능한 비즈니스 모델을 선택하고 검토

- 비즈니스 모델**
- 구독 모델 (Subscription Model)
  - 상용 및 오픈소스SW 제품 (Offering Commercial and Open Source Products)
  - 지원 및 교육 모델 (Support and Training Model)
  - 컨설팅 모델 (Consulting and Integration Strategy)
  - 듀얼 라이선스 모델 (Dual License Model)
  - 호스팅 서비스 (Hosted Service)
  - 광고 모델 (Advertising Model)
  - 강화된 상용SW (Commercial Enhancement)
  - 시스템 모델 (Embedding Model)
  - 후원 모델 (Patronage Model)
  - 패키지 SW (Packaging Model)



사업 모델	상세 설명
구독 모델	사용자들이 해당 오픈소스SW를 도입 및 사용할 경우 패치 업데이트 등 보다 전문적인 기술지원과 문제해결 등에 대한 지원을 받고 싶을 경우 지원수준 별 책정된 서비스 가격으로 연간 계약을 통해 서비스 요금을 부과하는 사업모델
상용 및 오픈소스SW 제품	오픈소스SW R&D 결과물 중 일부 기능을 구현하는 소스코드를 오픈소스SW 라이선스가 적용된 오픈소스SW로 공개하고 기능이나 성능, 활용범위 등이 보다 향상된 SW사용을 원하는 사용자들을 위해 공개하지 않은 소스코드를 상용 라이선스가 적용된 상용SW로 판매하는 사업모델
듀얼 라이선스 모델	오픈소스SW R&D결과물을 오픈소스SW 라이선스가 적용된 오픈소스SW로 공개하고 동일한 SW를 상용 라이선스가 적용된 상용SW로 판매하는 사업모델

# Business Strategy

## 오픈소스SW R&D 적용 오픈소스SW 라이선스 분류기준

- 오픈소스SW 라이선스를 소스코드 공개를 기준으로 퍼미시브 라이선스(Permissive License) 계열, 위크 카피레프트 라이선스(Weak Copyleft License) 계열, 스트롱 카피레프트 라이선스(Strong Copyleft License) 계열로 분류할 수 있다.
- 카피레프트(Copyleft)는 일반적인 상용/독점 저작권을 의미하는 카피라이트(Copyright)의 반대개념을 내포하는 용어로서 공유와 협업을 목적으로 제정된 라이선스들에 적용되고 있다.



# Business Strategy

## 선택한 비즈니스 모델에 적합한 라이선스 정책 수립

사업 모델	오픈소스SW 라이선스 정책
구독 모델 (Subscription Model)	<ul style="list-style-type: none"> <li>공개된 소스코드를 가지고 특정 기업이나 개발자들이 유사한 SW를 개발하여 독점/상용 라이선스로 판매하는 것을 제한하기 위해서 모든 소스코드를 반드시 공개해야만 하는 <b>스트롱 카피레프트 라이선스 정책이 권장</b> 됨.</li> <li>스트롱 카피레프트 라이선스를 적용함에 있어 라이선스 양립성에 대한 검토가 필요함</li> </ul>
상용 및 오픈소스SW 제품 (Offering Commercial and OpenSource Products)	<ul style="list-style-type: none"> <li>특정 기업 혹은 개발자들이 공개된 소스코드를 수정하여 독점/상용 라이선스로 판매하는 것을 제한하기 위해서 공개된 소스코드에는 해당 소스코드에만 라이선스가 적용되는 <b>위크 카피레프트 라이선스 사용이 권장</b> 됨,</li> <li>만약 R&amp;D결과물에 보다 강력한 스트롱 카피레프트 라이선스를 적용하기 위해서는 공개된 코드와 공개하지 않은 코드가 분리된 저작물 형태로 결합되어야 함</li> </ul>
듀얼 라이선스 모델 (Dual License Model)	<ul style="list-style-type: none"> <li>오픈소스SW를 사용하는 사용자들은 무료로 사용할 수 있지만 해당 오픈소스SW에 적용된 라이선스 의무사항을 준수해야 하고 상용SW를 구매하는 사용자들은 일반 상용SW와 동일하게 독점적으로 사용할 수 있음.</li> <li>특정 기업 혹은 개발자들이 오픈소스SW로 공개된 소스코드 전체 혹은 일부를 가지고 유사한 상용 독점 SW를 개발하여 판매하는 것을 제한하기 위해 공개된 소스코드에는 <b>스트롱 카피레프트 라이선스 적용이 권장</b> 됨.</li> <li>스트롱 카피레프트 라이선스는 해당 소스코드 전체 혹은 일부를 복제, 수정 사용할 경우 해당 코드와 연결된 모든 소스코드를 동일한 라이선스를 적용하여 누구나 사용할 수 있도록 공개해야 함에 따라 유사 경쟁SW 개발 및 판매를 예방할 수 있음.</li> <li>다만 동일한 소스코드에 대해 오픈소스SW 라이선스와 상용 라이선스를 동시에 적용하기 위해서는 해당 소스코드에 대한 저작권리를 100%로 확보하고 있어야 함.</li> </ul>

# Business Strategy

## 비영리 목적의 오픈소스SW에 적합한 라이선스 정책

추진전략	오픈소스SW 라이선스 정책
상용SW대체 전략	<ul style="list-style-type: none"> <li>• 기존 상용SW 기능을 대체할 수 있는 오픈소스SW를 무료로 제공함으로써 많은 사용자에게 비용절감을 가능하게 하기 위해서는 특정 기업 및 개발자들이 해당 오픈소스SW를 활용하여 상용/독점 판매를 제한하는 것이 바람직함</li> <li>• 해당 오픈소스SW를 일부 혹은 전체를 복제, 수정하여 2차 저작물을 개발 및 배포할 경우 모든 소스코드를 공개해야 하는 <b>스트롱 카피레프트 계열의 라이선스 적용이 권장</b> 됨</li> </ul>
원천기술 및 신기술 기반 시장 확대전략	<ul style="list-style-type: none"> <li>• 기존에 없는 원천기술이나 신기술이 적용된 오픈소스SW를 제공함으로써 많은 사용자들을 통한 사용 및 기타 융합기술 적용을 통해 시장을 확대 시키기 위해서는 누구나 자유롭게 별다른 제약조건 없이 사용할 수 있도록 허용하는 것이 바람직함.</li> <li>• 개인 및 기업들이 자유롭게 사용하고 해당 오픈소스SW를 다양한 기술 및 제품에 적용하여 부가가치를 얻게 하기 위해서는 <b>사용 및 저작권 고지 의무만을 요구하고 있는 퍼미시브 라이선스 계열의 라이선스 적용이 권장</b> 됨</li> </ul>
브랜드 가치 및 명성	<ul style="list-style-type: none"> <li>• 유용한 오픈소스SW를 지속적으로 제공함으로써 브랜드 및 명성을 높이기 위해서는 누구나 자유롭게 별다른 제약조건 없이 사용할 수 있도록 허용하는 것이 좋음.</li> <li>• 개인 및 기업들이 자유롭게 사용하고 해당 오픈소스SW를 통해 다양한 부가가치를 얻게 하기 위해서는 <b>사용 및 저작권 고지 의무만을 요구하고 있는 퍼미시브 라이선스 계열의 라이선스 적용이 권장</b> 됨</li> </ul>
인재육성 전략	<ul style="list-style-type: none"> <li>• 대학 및 연구소 개발자들을 오픈소스SW 개발에 적극 참여시키고 글로벌 커뮤니티 및 커미터들과의 협업을 통해 보다 유능한 인력으로 성장하고 유능한 인력을 유인할 수 있게 하기 위해서는 누구나 자유롭게 별다른 제약조건 없이 사용할 수 있도록 허용하는 것이 바람직 함.</li> <li>• 개인 및 기업들이 자유롭게 사용하고 해당 오픈소스SW를 통해 다양한 부가가치를 얻게 하기 위해서는 <b>사용 및 저작권 고지 의무만을 요구하고 있는 퍼미시브 라이선스 계열의 라이선스 적용이 권장</b> 됨</li> </ul>

# Business Strategy

## 라이선스 정책에 따른 의무사항

오픈소스SW 라이선스 분류	R&D 결과물 공개시 발생 의무사항
퍼미시스 라이선스 계열 (MIT, BSD 등)	<ul style="list-style-type: none"> <li>• 사용 오픈소스SW명, 저작권 명, 라이선스 명, 라이선스 사본, 출처에 대한 고지</li> <li>• 경우에 따라, Apache 등과 같이 무상 특허 허용 의무사항을 가진 라이선스 검토 필요</li> </ul>
위크 카피레프트 라이선스 계열 (LGPL, EPL, MPL 등)	<ul style="list-style-type: none"> <li>• 사용 오픈소스SW명, 저작권 명, 라이선스 명, 라이선스 사본, 출처에 대한 고지</li> <li>• 오픈소스SW를 수정 사용할 경우 수정 사용 소스코드 공개 의무 (예외 조항 있음)</li> <li>• 경우에 따라, EPL, MPL 등과 같이 무상 특허 허용 의무사항을 가진 라이선스 검토 필요</li> </ul>
스트롱 카피레프트 라이선스 계열 (I) (GPL 2.0, AGPL 2.0 등)	<ul style="list-style-type: none"> <li>• 사용 오픈소스SW명, 저작권 명, 라이선스 명, 라이선스 사본, 출처에 대한 고지</li> <li>• 사용한 오픈소스SW 라이선스와 동일한 조건으로 전체 소스코드 공개 의무</li> </ul>
스트롱 카피레프트 라이선스 계열 (II) (GPL 3.0, AGPL 3.0 등)	<ul style="list-style-type: none"> <li>• 사용 오픈소스SW명, 저작권 명, 라이선스 명, 라이선스 사본, 출처에 대한 고지</li> <li>• 사용한 오픈소스SW 라이선스와 동일한 조건으로 전체 소스코드 공개 의무</li> <li>• 경우에 따라, GPL 3.0, AGPL 3.0 등과 같이 무상 특허 허용 의무사항을 가진 라이선스 검토 필요</li> <li>• 추가적으로 기술적 보호조치의 보호금지, 설치정보 제공의무가 발생하는 GPL 3.0, AGPL 3.0 등과 같은 오픈소스SW 사용일 경우</li> </ul>

# Policy and Organization

연구개발과제 수행을 위한 오픈소스 관리정책 구비하고 참여연구원 모두 수시로 확인할 수 있도록 공유

오픈소스를 사용하여 소프트웨어  
제품과 서비스를 만들어서  
배포하기 위한 정책

- 오픈소스 SW의 식별
- 라이선스 의무사항 검토
- 오픈소스 SW의 결합 방식 정의
- 컴플라이언스 산출물(고지문, SBOM 등)
- 컴플라이언스 프로세스

외부 오픈소스 커뮤니티에  
기여하기 위한 정책

- 기여할 소스코드의 범위
- 기여할 소스코드의 저작권 표시방법
- 민감한 지식자산의 처리방법
- CLA 서명 관리
- 기여물의 리뷰 및 승인 프로세스

기업의 소프트웨어를 오픈소스로  
공개하기 위한 정책

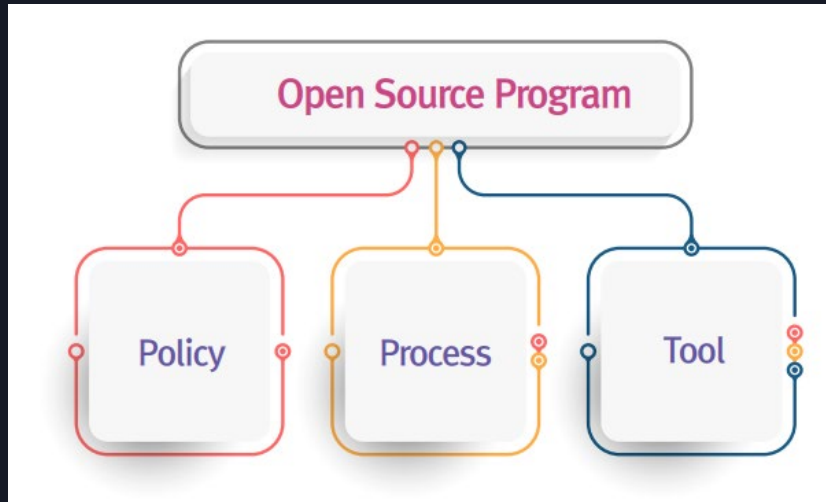
- 공개 전 리뷰 및 승인 프로세스
- 소스코드 라이선스 검증
- 민감한 지식자산의 검증방안
- 공개 후 관리 방안 및 조직 구성

정책 예시 : [https://haksungjang.github.io/docs/governance\\_iso5230/appendix/1-policy-template/](https://haksungjang.github.io/docs/governance_iso5230/appendix/1-policy-template/)

Generic FOSS Policy Example : [https://github.com/todogroup/policies/blob/master/linuxfoundation/lf\\_compliance\\_generic\\_policy.pdf](https://github.com/todogroup/policies/blob/master/linuxfoundation/lf_compliance_generic_policy.pdf) (Linux Foundation)

# Policy and Organization

조직의 오픈소스 거버넌스 체계를 구축하는 핵심 조직을 중심으로 오픈소스 전략을 수립하고 실행하는데 필요한 정책, 프로세스 및 도구를 제공(OSPO, Open Source Program Office)



오픈소스 프로그램

- 사내외 오픈소스 전략을 명확하게 전달
- 전략 실행을 소유 및 감독
- 상용 제품 및 서비스에서 오픈 소스의 효과적인 사용 촉진
- 오픈 소스 커뮤니티에 고품질의 빈번한 코드 릴리스 보장
- 개발자 커뮤니티에 참여하고 회사가 다른 프로젝트에 효과적으로 기여하는지 확인
- 조직 내 오픈소스 문화 조성
- 오픈 소스 라이선스 준수 검토 및 감독 유지

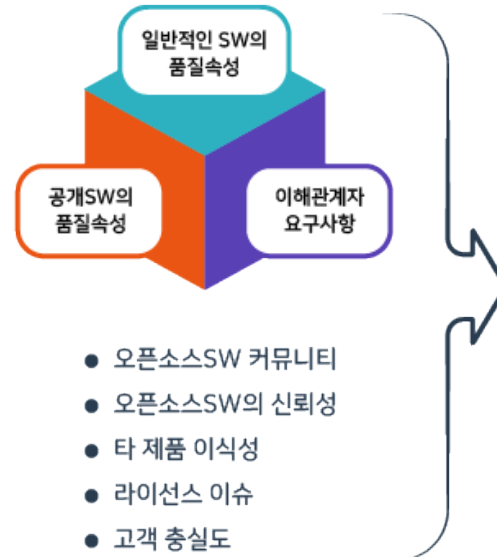
참고 : [https://www.oss.kr/oss\\_guide/show/22012cba-8614-442b-bfcf-62eab9ec888a](https://www.oss.kr/oss_guide/show/22012cba-8614-442b-bfcf-62eab9ec888a)



# Project Evaluation

조사-분석-평가-계약 과정을 통해 과제에 적합한 외부 오픈소스 SW 프로젝트를 선정

TTA Standard	정보통신단체표준(국문표준) TTAK.KO-10.3000x 제정일: 2012년 xx월 xx일	중분류	세부항목
	공개SW 성숙도 및 적용성 평가 표준 The standard of maturity and applicability assessment for open source software	사용지원	이해성
			학습성
			운용성
			분석성
			전문기술
	기능지원	시험성	
		관리체계	
		대체성	
		대체후기능성	
설치성			
커뮤니티	지원성		
	상호운용성		
	나이 및 규모		
신뢰성	주체		
	검근성		
	성숙성		
	적합성		
			가용성
			회복성



오픈소스 명 / 평가 항목	jitsi	easyrtc	skylink	peerJS	kurento
사용기술	10	10	10	10	10
비즈니스 적합성	9	9	0	9	9
활성도	7	8	7	5	8
성숙도	9	9	5	7	8
학습 난이도	8	10	10	8	3
보안성	9	8	*	7	9
라이선스	8	9	0	9	9
기술지원여부(무료)	10	10	10	0	5
결과 합(80점 만점)	70	73	42	55	61

오픈소스SW 성숙도 및 적용성 평가 표준(TTA)

외부 오픈소스SW 활용을 위한 프로젝트 평가 예



# Supply Chain Management

## SBOM 기반 오픈소스SW 공급망 관리

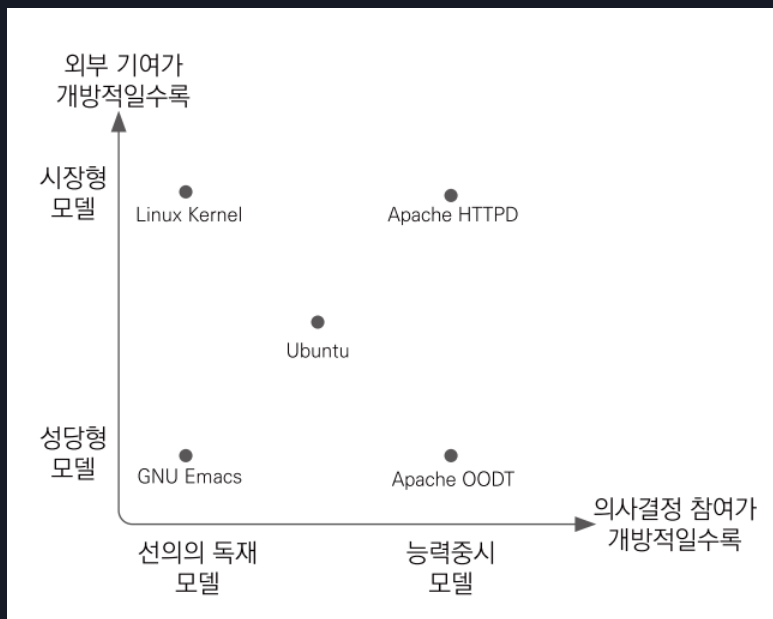
연구 주관기관은 참여기관 혹은 외부 개발자로부터 오픈소스SW R&D 결과물에 포함되는 소프트웨어를 공급받을 경우 해당 오픈소스SW R&D를 위해 수립된 오픈소스SW 사용 정책이 준수 될 수 있도록 계약 시에 오픈소스SW 사용 정책을 분명히 고지하고 SW 공급시에 정책준수 여부를 확인하기 위한 조항이 필요.

컴포넌트명	라이선스	라이선스	버전	수정	결합 형태	공개	고지	특허	기준
		LGPL 2.1	1.11	○	Dyanmic Link	○	○	X	준수
컴포넌트명	보안취약점	데이터베이스	ID		릴리즈 날짜		Base Score	등급	기준
		NVD	CVE-2017-3586		2017-04-24		5.5	Medium	준수

# Community Building

## 오픈소스 커뮤니티 유형 및 관리활동

- 지속적인 가치 있는 기여를 통해 영향력을 얻는 능력주의 위주의 조직
- 한 개인이나 집단이 중앙 집권적인 권한을 갖는 선의의 독재형(benevolent dictatorship) 모델과 참여자들이 분권적인 권한을 가지는 능력 중시형(meritocratic) 모델로 구분



오픈소스 커뮤니티 거버넌스 모델 유형

구분	커뮤니티 관리 활동의 예
프로젝트 관리	로드맵, 제품 배포, 요구사항 관리, 추가 기능 개발 및 리뷰
커뮤니티 관리	참여자 관리, 의사소통 채널 관리, 포럼, 메일링 리스트, 이벤트 관리 등
라이선스 관리	라이선스 관련 법률적 분쟁 협의 및 해결
소프트웨어 개발	개발, 소스 코드 관리(커밋, PR), 테스트, 할당된 이슈 및 버그 처리, 데모 제공 등
시스템 운영	소스 코드 저장소, 자동화 배포 시스템, 이슈 및 버그 관리 시스템, 문서화 도구, 품질 관리 시스템, 공식 웹사이트 등

# Community Building

## 오픈소스 커뮤니티 구성요소

- 웹사이트 및 포럼
- 기여자 가이드라인
- 커뮤니티 거버넌스
- 마일스톤 및 로드맵
- 협력기업 관리
- 지적 재산권 관리
- 모니터링
- 홍보

### 컨트리뷰션 가이드라인

- 1. Fork**
  - Upstream Repository를 자신의 GitHub 계정으로 Fork 합니다.
- 2. Clone**
  - Fork한 Repository를 자신의 Local working directory로 Clone 합니다.

```
$ mkdir -p $working_dir
$ cd $working_dir
$ git clone https://github.com/hamon1kr/hamonize.git
```
- 3. Create a branch**
  - 개발용 branch를 생성하여 해당 branch에서 작업 및 테스트를 수행합니다.

```
$ cd hamonize
$ git checkout -b myfeature
```
- 4. Commit**
  - 수정 사항을 commit합니다.

```
$ git commit -a -m '[commit message]'
```
- 5. Push**
  - 수정 사항을 자신의 GitHub Repository에 Push 합니다.

```
git push origin myfeature
```
- 6. 빌드 테스트**
  - travis-ci를 통해 수정한 소스코드가 정상적으로 빌드 되는 것을 확인해 주세요.
  - 1) travis-ci.com에 접속, github 계정으로 로그인
  - 2) 포크한 hamonize 저장소 팔로우
  - 3) 환경변수 GITHUB\_TOKEN 등록
  - 4) release 브랜치를 작업한 브랜치의 상태로 업데이트
  - 5) travis-ci에서 이력진 빌드가 'passed' 인 것을 확인
- 7. Pull Request 생성하기**

기여자 가이드라인 예시

### Decision making

Decisions in Apache projects result from:

- action
- consensus in the community
- (if needed) votes within the community

We have, over the years, developed a very simple and effective approach to consensus building and decision making. Projects, and the ASF generally, make the vast majority of decisions using [lazy consensus](#). If the "lazy" approach to consensus building seems unsuitable for a particular decision, we seek to [build consensus](#) within the community. Very occasionally, usually for formal reasons relating to legal responsibilities, it is necessary to call a [vote](#).

Full descriptions of each of these approaches are available on various foundation and project sites. However, we've provided an initial introduction to [decision making](#) here on the community development site.

### Project independence

While not all ASF projects practice all aspects of the Apache Way in the same way, there are a number of rules that all projects should follow when complying with PMC [release voting](#), [legal policy](#), [brand policy](#), using [mailing lists](#), etc., which vary slightly between projects.

One of these invariant rules is that PMCs manage projects independently of any commercial interests. The goal is to ensure involvement in most all participants are equal and thus have an equal opportunity to contribute to and benefit from our software, regardless of occupation or financial objectives. There is more discussion of this principle in our document [Project Independence](#).

### Development processes

Each project in Apache is free to define its own development processes as long as it respects our [decision-making](#) requirements. However, over the many years that the ASF has existed, our projects have evolved and documented many best practices for development. Some of the most popular of these practices. Most Apache projects operate in ways that are largely similar to these:

- [Crediting contributors](#) is a critical part of community building
- [ASF Licensing Policy](#)
- [Releases](#)
  - [Releases FAQ](#)
  - [Release audit](#) is a process each release candidate must go through to ensure it is managing intellectual property (IP) correctly
    - [RAT/Creduir](#) is a tool to assist this process
  - [Release management](#) is how the project prepares and publishes its releases
  - [Handling cryptography](#) in an Apache Release
  - [Source Headers and NOTICE](#) Files
  - [Distribution best practices](#)
- [ASF Developer section sitemap](#)

### Assisting with project management and marketing

There are many visible contributions of value in an Apache projects (documentation, testing, coding user support, design etc.), but there are also plenty of "back office" tasks that are necessary to maintain a healthy project. This section describes some of the activities that you can help with. In most cases you will need to be a committer to carry out these activities, although non-committers can often act in supporting roles.

- [Write informational blogs](#)
- [Assessing and approving new committers](#)
- [Board Reports](#)
- [Apache Project Branding/Trademark Resources](#)

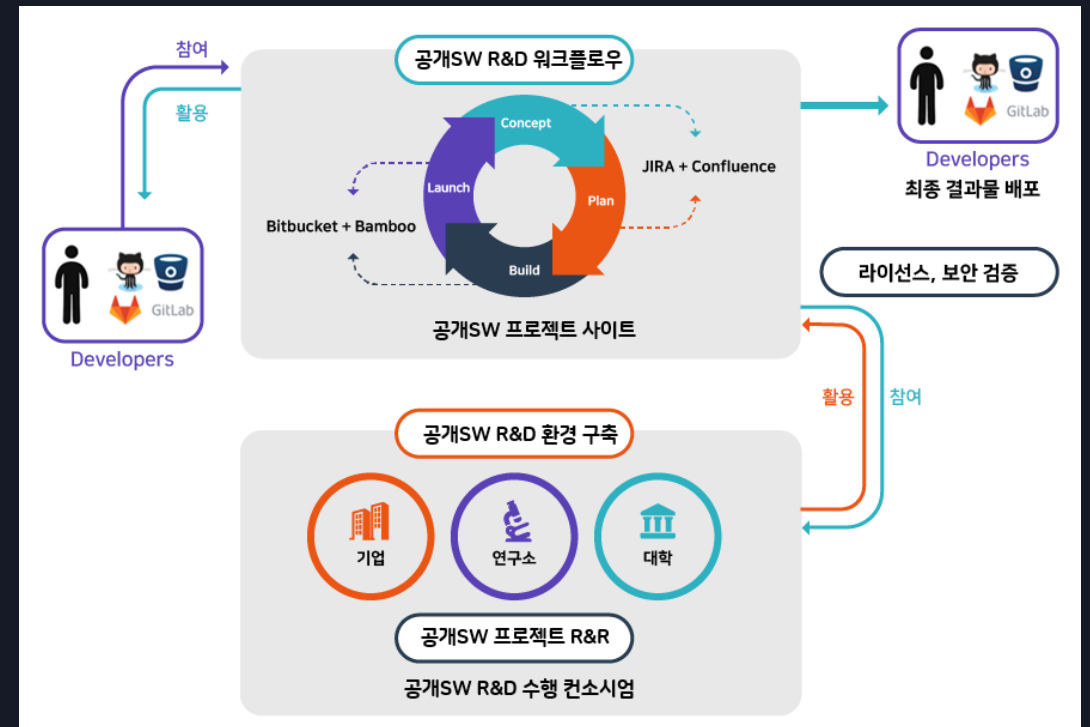
- 역할과 책임
- 지원
- 기여과정
- 의사결정 과정

커뮤니티 거버넌스 문서 예 (Apache Foundation)

# Development Environment

외부 기여자들의 과제 참여가 가능하도록 개발환경 구축

- 연구개발과제 워크플로우(git-flow, github-flow 등)
- 소프트웨어 소스 코드 저장소(Github, Gitlab, Bitbucket 등)
- 소프트웨어 형상 관리 도구(Mantis, Jira, Github 등)
- 소프트웨어 테스트 자동화 도구(Catch, Junit, unittest 등)
- 소프트웨어 보안 취약점 점검 도구(Veracode Greenlight, Synopsys Code Sight 등)
- 소프트웨어 릴리즈 관리 도구(Travis, Hudson, Bamboo 등)
- 라이선스 검증 도구(올리브, FOSSology, CodeEye, Clarity 등)
- 라이선스 컴플라이언스 도구(Fossilight, Fossa, SW360, Black Duck, SPDX 등)
- 연구개발과제 이슈 및 버그 관리 도구(Github, Gitlab, Jira 등)
- 참여형 문서 협업 도구(Wiki, Confluence 등)



오픈소스 SW 방식의 연구개발 과제 개발환경 예

# Performance Management

연구과제에 적합한 성과지표를 정의하고 프로젝트의 메타데이터를 가시화

- 공개한 소스 코드 저장소의 활성화 정도를 표시하는 스타, 커밋(commit), 포크(fork), 이슈(Issue), 풀리퀘스트 (Pull Request) 등의 지표
- 커뮤니티의 활동성을 대변하는 기여자 수, 커뮤니티 사용자, 홍보 채널, 각종 행사 참여 수, 기술 문서 등의 지표
- 결과물의 활용성을 의미하는 기술이전, 적용사례, 파트너 참여기업 수 등의 지표



성과지표	배지 이미지 생성 URL	보기
Stars	/github/stars/:org	Stars 7k
Commit	/github/commit-activity/:interval/:user/:repo	commit activity 79/month
Fork	/github/forks/:user/:repo?label=Fork	Fork 150
Pull Request	/github/issues-pr/:user/:repo	pull requests 136 open
해결된 이슈	/github/issues-closed-raw/:user/:repo	closed issues 899
해결된 PR	/github/issues-pr-closed/:user/:repo	closed pull requests 7k
컨트리뷰터	/github/all-contributors/:user/:repo/:branch*	all contributors 66
다운로드 수	/github/downloads/:user/:repo/total	downloads 857k

오픈소스SW 프로젝트 성과지표 가시화 예

# Open R&D Capability Maturity Model

## 도메인별 평가방법

도메인	평가항목	평가방법
비즈니스 전략	<ul style="list-style-type: none"> <li>공개SW기반의 비즈니스 모델에 대한 이해 정도</li> <li>공개SW기반의 비즈니스 모델에 대한 전략 보유 현황</li> <li>공개SW 프로젝트에 대한 기술지원 체계의 보유 현황</li> </ul>	<ul style="list-style-type: none"> <li>공개SW기반의 비즈니스 모델을 정의한 문서</li> <li>공개SW기반의 비즈니스 모델에 대한 진입 전략 확인</li> <li>공개SW 프로젝트에 대한 기술지원 체계의 구축 및 관리 현황</li> </ul>
정책 및 조직	<ul style="list-style-type: none"> <li>오픈R&amp;D에 필요한 거버넌스 필요성 및 구성요소에 대한 이해 정도</li> <li>공개SW에 대한 별도의 전담자 또는 조직의 현황</li> <li>오픈R&amp;D 거버넌스 정책의 구비 현황</li> </ul>	<ul style="list-style-type: none"> <li>오픈R&amp;D 수행에 필요한 활동의 이해 정도</li> <li>공개SW 정책 및 각 활동에 대한 별도의 전담조직 또는 담당자</li> <li>오픈R&amp;D 수행에 필요한 라이선스, 개발방법, 관리 등 정책의 구비</li> </ul>
프로젝트 평가	<ul style="list-style-type: none"> <li>공개SW 프로젝트를 평가하는 방법의 보유 현황</li> <li>프로젝트의 성숙도를 관리할 수 있는 역량</li> <li>프로젝트의 성숙도를 관리할 수 있는 관리활동의 현황</li> </ul>	<ul style="list-style-type: none"> <li>오픈소스 프로젝트의 성숙도에 대한 이해 정도</li> <li>직접 개발 또는 외부 유입 오픈소스 프로젝트에 대한 평가방법 보유</li> <li>프로젝트 성숙도를 측정할 수 있는 담당자 및 도구</li> </ul>
공급망 관리	<ul style="list-style-type: none"> <li>외부에서 획득되는 공개SW의 획득에 필요한 절차의 보유 현황</li> <li>공개SW가 포함되는 소프트웨어 공급망에 대한 관리 정책 및 활동</li> <li>공개SW가 포함되는 소프트웨어 공급망에 대한 라이선스 관리 활동 현황</li> </ul>	<ul style="list-style-type: none"> <li>프로젝트에서 사용되는 외부 컴포넌트의 식별 상태</li> <li>포함되는 외부 프로젝트에 대한 관리 정책의 보유</li> <li>포함되는 외부 프로젝트에 대한 라이선스 및 보안 관리 활동</li> </ul>
커뮤니티	<ul style="list-style-type: none"> <li>공개SW 커뮤니티를 위한 거버넌스에 대한 이해 현황</li> <li>커뮤니티를 관리할 책임과 권한의 정의 및 활동 현황</li> <li>공개SW 커뮤니티 활동을 위한 협업에 필요한 환경의 구축 현황</li> </ul>	<ul style="list-style-type: none"> <li>커뮤니티 발전방향에 대한 로드맵의 보유 및 관리</li> <li>외부 참여자를 관리하는 담당자 및 활동 정의</li> <li>커뮤니케이션 채널 및 커뮤니티 거버넌스 관련 문서</li> </ul>
개발환경	<ul style="list-style-type: none"> <li>공개SW 개발을 위한 개발환경의 보유상태</li> <li>오픈R&amp;D를 위한 개발 워크플로우의 보유현황 및 관리활동</li> <li>공개SW로 운영될 프로젝트의 품질관리체계 보유현황</li> <li>배포되는 공개SW에 대한 관리방안 현황</li> </ul>	<ul style="list-style-type: none"> <li>공개SW 개발 프로젝트의 소스코드 형상관리 체계</li> <li>공개SW 개발 프로젝트의 이슈 및 버그 관리 체계</li> <li>공개SW 개발 프로젝트의 품질가시화 방안</li> <li>공개SW 개발 프로젝트의 문서화(예제, 가이드, 로드맵 등)</li> </ul>
성과관리	<ul style="list-style-type: none"> <li>오픈R&amp;D에 적합한 성과지표의 정의 현황</li> <li>오픈R&amp;D에 적합한 성과지표의 관리 현황</li> <li>오픈R&amp;D에 적합한 성과지표의 지속적 개선 현황</li> </ul>	<ul style="list-style-type: none"> <li>오픈R&amp;D에 적합한 성과지표가 정의된 문서</li> <li>오픈R&amp;D에 적합한 성과지표를 관리하는 활동 여부 및 담당자</li> <li>오픈R&amp;D에 적합한 성과지표의 지속적 개선 활동</li> </ul>



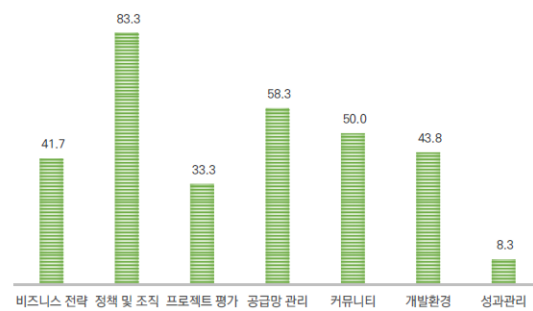
# Open R&D Capability Maturity Model

## 평가 결과 예시

도메인별 중요도(가중치)

도메인	가중치	도메인평점	가중평균
비즈니스 전략	20%	41.7	46.25
정책 및 조직	10%	83.3	
프로젝트 평가	10%	33.3	
공급망 관리	20%	58.3	
커뮤니티	10%	50.0	
개발환경	20%	43.8	
성과관리	10%	8.3	

도메인별 성숙도

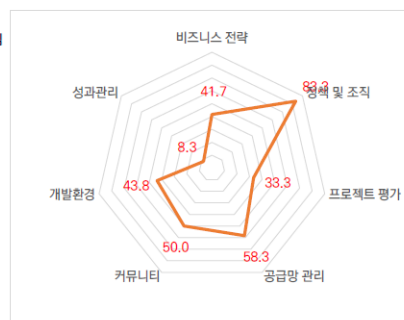


오픈소스 SW 연구개발 역량 성숙도 세부 평가 항목

도메인	평가항목	평가방법	평가결과	항목점수 (100점 변환)	평균
비즈니스 전략	공개SW기반의 비즈니스 모델에 대한 이해 정도	공개SW기반의 비즈니스 모델을 정의한 문서	3	50	41.7
비즈니스 전략	공개SW기반의 비즈니스 모델에 대한 전략 보유 현황	공개SW기반의 비즈니스 모델에 대한 진입 전략 확인	3	50	
비즈니스 전략	공개SW 프로젝트에 대한 기술지원 체계의 보유 현황	공개SW 프로젝트에 대한 기술지원 체계의 구축 및 관리 현황	2	25	
정책 및 조직	오픈R&D에 필요한 거버넌스 필요성 및 구성요소에 대한 이해 정도	오픈R&D 수행에 필요한 활동의 이해 정도	5	100	83.3
정책 및 조직	공개SW에 대한 별도의 전담자 또는 조직의 현황	공개SW 정책 및 각 활동에 대한 별도의 전담조직 또는 담당자	5	100	
정책 및 조직	오픈R&D 거버넌스 정책의 구비 현황	오픈R&D 수행에 필요한 라이선스, 개발방법, 관리 등 정책의 구비	3	50	
프로젝트 평가	공개SW 프로젝트를 평가하는 방법의 보유 현황	오픈소스 프로젝트의 성숙도에 대한 이해 정도	2	25	33.3
프로젝트 평가	프로젝트의 성숙도를 관리할 수 있는 역량	직접 개발 또는 외부 유입 오픈소스 프로젝트에 대한 평가방법 보유	2	25	
프로젝트 평가	프로젝트의 성숙도를 관리할 수 있는 관리활동의 현황	프로젝트 성숙도를 측정할 수 있는 담당자 및 도구	3	50	
공급망 관리	외부에서 획득된 공개SW의 획득에 필요한 절차의 보유 현황	프로젝트에서 사용되는 외부 컴포넌트의 식별 상태	4	75	58.3
공급망 관리	공개SW가 포함되는 소프트웨어 공급망에 대한 관리 정책 및 활동	포함되는 외부 프로젝트에 대한 관리 정책의 보유	2	25	
공급망 관리	공개SW가 포함되는 소프트웨어 공급망에 대한 라이선스 관리 활동 현황	포함되는 외부 프로젝트에 대한 라이선스 및 보안 관리 활동	4	75	
커뮤니티	공개SW 커뮤니티를 위한 거버넌스에 대한 이해 현황	커뮤니티 발전방향에 대한 로드맵의 보유 및 관리	5	100	50.0
커뮤니티	커뮤니티를 관리할 책임과 권한의 정의 및 활동 현황	외부 참여자를 관리하는 담당자 및 활동 정의	2	25	
커뮤니티	커뮤니티 활동을 위한 협업에 필요한 환경의 구축 현황	커뮤니케이션 채널 및 커뮤니티 거버넌스 관련 문서	2	25	
개발환경	공개SW 개발을 위한 개발환경의 보유상태	공개SW 개발 프로젝트의 소스코드 형상관리 체계	2	25	43.8
개발환경	개발 워크플로우의 보유현황 및 관리활동	공개SW 개발 프로젝트의 이슈 및 버그 관리 체계	3	50	
개발환경	공개SW로 운영될 프로젝트의 품질관리체계 보유현황	공개SW 개발 프로젝트의 품질지표화 방안	4	75	
개발환경	배포되는 공개SW에 대한 관리방안 현황	공개SW 개발 프로젝트의 문서화(예제, 가이드, 로드맵 등)	2	25	8.3
성과관리	오픈R&D에 적합한 성과지표의 정의 현황	오픈R&D에 적합한 성과지표가 정의된 문서	1	0	
성과관리	오픈R&D에 적합한 성과지표의 관리 현황	오픈R&D에 적합한 성과지표를 관리하는 활동 여부 및 담당자	2	25	
성과관리	오픈R&D에 적합한 성과지표의 지속적 개선 현황	오픈R&D에 적합한 성과지표의 지속적 개선 활동	1	0	

오픈소스 SW 연구개발 역량 성숙도 수준

Lv3	Lv1	문제의식 자체가 없거나, 표준화된 프로세스 부재
	Lv2	표준화/문서화된 절차가 존재하여 혼란을 통해 진파, 준수여부는 개인의 역량에 맡겨짐
	Lv3	절차의 준수여부에 대한 측정 및 조치가 이루어짐
	Lv4	진사적 표준화된 절차가 관리되고 전사 활동에 대한 측정 및 조치가 이루어짐
	Lv5	지속적 개선이 이루어지고, 조직이 내외부 변화에 신속하게 대응 가능



# Check List

도메인	활동	세부 평가방법 예시	O/X
비즈니스 전략	오픈소스 기반의 비즈니스 전략 설정 및 실행	비즈니스 전략이 명확하게 설정되어 있는가?	
비즈니스 전략	오픈소스 기반의 비즈니스 전략 설정 및 실행	비즈니스 전략이 실행되고 있는가?	
비즈니스 전략	오픈소스 기반의 비즈니스 전략 설정 및 실행	전략에 대한 피드백이 반영되고 있는가?	
비즈니스 전략	오픈소스 기반의 비즈니스 전략 설정 및 실행	전략이 명확하게 커뮤니케이션 되고 있는가?	
비즈니스 전략	오픈소스 기반의 비즈니스 전략 설정 및 실행	전략이 지속적으로 개선되고 있는가?	
정책 및 조직	오픈소스 프로젝트를 위한 조직 내 정책 및 조직 구조 설정	오픈소스 정책이 명확하게 설정되어 있는가?	
정책 및 조직	오픈소스 프로젝트를 위한 조직 내 정책 및 조직 구조 설정	정책이 조직 내에 잘 전파되고 있는가?	
정책 및 조직	오픈소스 프로젝트를 위한 조직 내 정책 및 조직 구조 설정	정책에 대한 이해도가 높은가?	
정책 및 조직	오픈소스 프로젝트를 위한 조직 내 정책 및 조직 구조 설정	정책이 지속적으로 개선되고 있는가?	
정책 및 조직	오픈소스 프로젝트를 위한 조직 내 정책 및 조직 구조 설정	정책 준수 상태가 모니터링되고 있는가?	
프로젝트 평가	프로젝트에 활용하는 포함된 다른 오픈소스 프로젝트의 평가 방안	프로젝트 평가 방안이 명확하게 설정되어 있는가?	
프로젝트 평가	프로젝트에 활용하는 포함된 다른 오픈소스 프로젝트의 평가 방안	평가 방안이 실행되고 있는가?	
프로젝트 평가	프로젝트에 활용하는 포함된 다른 오픈소스 프로젝트의 평가 방안	평가 방안에 대한 피드백이 반영되고 있는가?	
프로젝트 평가	프로젝트에 활용하는 포함된 다른 오픈소스 프로젝트의 평가 방안	평가 방안이 지속적으로 개선되고 있는가?	
프로젝트 평가	프로젝트에 활용하는 포함된 다른 오픈소스 프로젝트의 평가 방안	평가 방안이 커뮤니케이션되고 있는가?	
공급망 관리	오픈소스 소프트웨어의 라이선스, 보안 취약점, 소프트웨어 공급망 등 관리	라이선스 이행 상태가 모니터링되고 있는가?	
공급망 관리	오픈소스 소프트웨어의 라이선스, 보안 취약점, 소프트웨어 공급망 등 관리	보안 취약점 관리가 체계적으로 이루어지는가?	
공급망 관리	오픈소스 소프트웨어의 라이선스, 보안 취약점, 소프트웨어 공급망 등 관리	품질 관리 방안이 실행되고 있는가?	
공급망 관리	오픈소스 소프트웨어의 라이선스, 보안 취약점, 소프트웨어 공급망 등 관리	공급망 관리 방안이 명확하게 설정되어 있는가?	
공급망 관리	오픈소스 소프트웨어의 라이선스, 보안 취약점, 소프트웨어 공급망 등 관리	공급망 관리 방안이 지속적으로 개선되고 있는가?	
커뮤니티	오픈소스 커뮤니티와의 상호작용 및 참여	커뮤니티와의 상호작용이 체계적으로 이루어지는가?	
커뮤니티	오픈소스 커뮤니티와의 상호작용 및 참여	커뮤니티 참여 활동이 체계적으로 이루어지는가?	
커뮤니티	오픈소스 커뮤니티와의 상호작용 및 참여	리더십이 커뮤니티 내에서 발휘되고 있는가?	
커뮤니티	오픈소스 커뮤니티와의 상호작용 및 참여	커뮤니티 내에서의 의사결정 과정이 명확한가?	
커뮤니티	오픈소스 커뮤니티와의 상호작용 및 참여	커뮤니티 참여 활동이 지속적으로 개선되고 있는가?	
개발 환경	오픈소스 프로젝트 개발을 위한 환경 설정 및 관리	개발 환경 설정이 체계적으로 이루어지는가?	
개발 환경	오픈소스 프로젝트 개발을 위한 환경 설정 및 관리	개발 환경 관리가 체계적으로 이루어지는가?	
개발 환경	오픈소스 프로젝트 개발을 위한 환경 설정 및 관리	지속적 통합은 체계적으로 이루어지는가?	
개발 환경	오픈소스 프로젝트 개발을 위한 환경 설정 및 관리	개발 환경이 사용자에게 편리한가?	
개발 환경	오픈소스 프로젝트 개발을 위한 환경 설정 및 관리	개발 환경 설정 및 관리가 지속적으로 개선되고 있는가?	
성과 관리	오픈소스 프로젝트의 성과 측정 및 관리	성과 측정 방안이 명확하게 설정되어 있는가?	
성과 관리	오픈소스 프로젝트의 성과 측정 및 관리	성과 관리가 체계적으로 이루어지는가?	
성과 관리	오픈소스 프로젝트의 성과 측정 및 관리	방향성 재정립이 성과에 기반하여 이루어지는가?	
성과 관리	오픈소스 프로젝트의 성과 측정 및 관리	성과 측정 결과가 명확하게 커뮤니케이션 되고 있는가?	
성과 관리	오픈소스 프로젝트의 성과 측정 및 관리	성과 측정 및 관리 방안이 지속적으로 개선되고 있는가?	

3

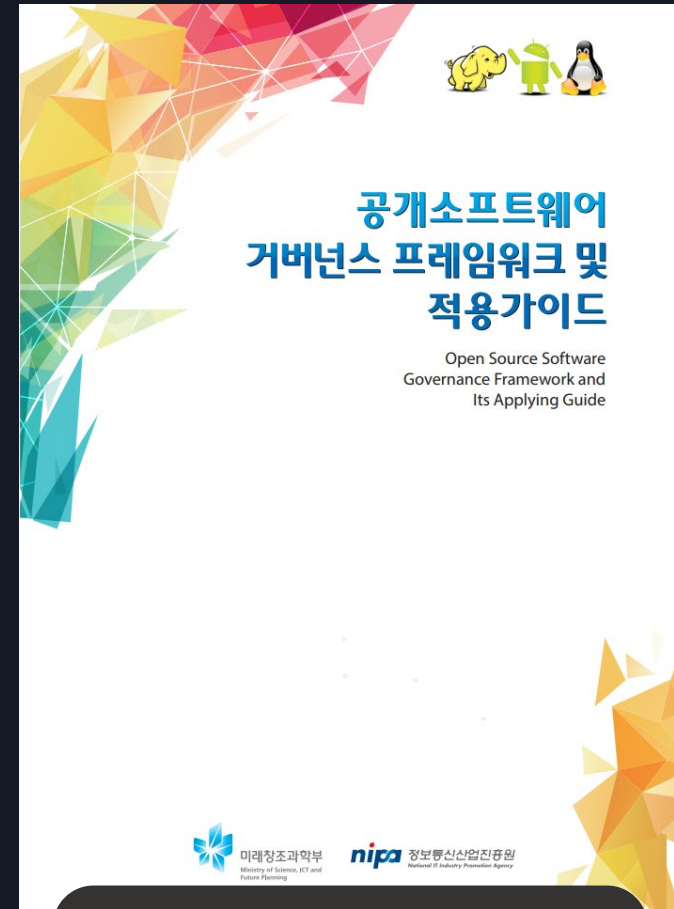
참고자료

# 공개소프트웨어 거버넌스 프레임워크 및 적용가이드

(목적) 오픈소스를 안전하게 사용, 적용 및 배포하기 위해 필요한 사항을 다양한 사용자 관점에서 활용할 수 있도록 소프트웨어 개발 라이프 사이클 단계별로 제시한 틀

(내용) 다양한 유형의 오픈소스 사용자들이 유형별로 제시되는 가이드 라인을 따라가면서 오픈소스 거버넌스 활동요소를 선택적으로 고려할 수 있도록 제공하고 있으므로 자신의 상황에 적합한 활용 케이스를 참조할 수 있다.

[https://www.oss.kr/oss\\_guide/show/18fadea5-be8d-4631-9223-8805fbfd847b](https://www.oss.kr/oss_guide/show/18fadea5-be8d-4631-9223-8805fbfd847b)



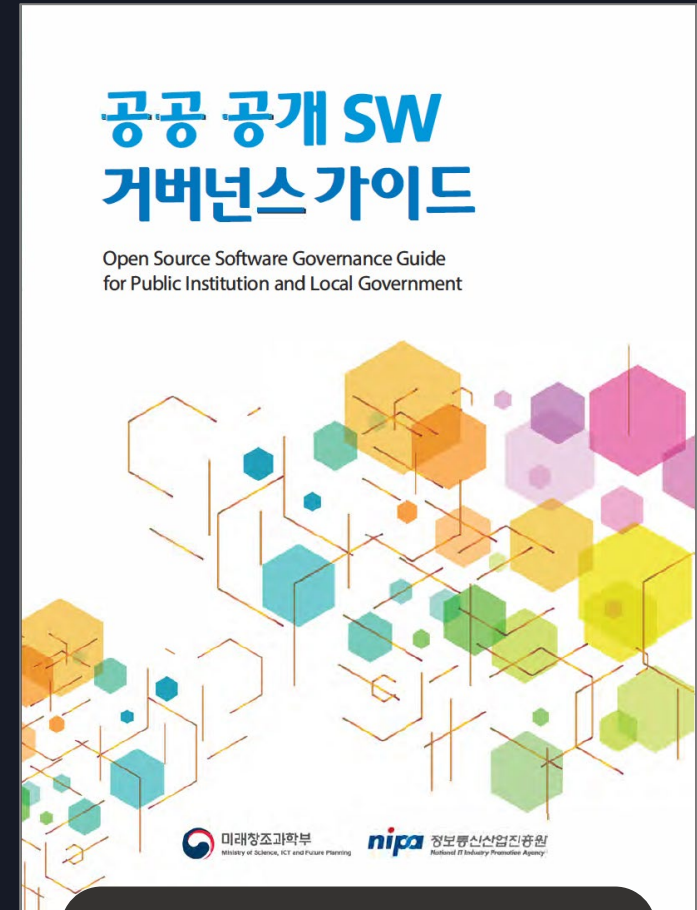
[www.oss.kr](http://www.oss.kr)

# 공공 공개SW 거버넌스 가이드라인

(목적) 공공부문에서 공개SW를 활용해야 하는 경우에 필요한 관련 법령 및 지침을 정보화사업 추진 단계별로 구분하여 제시하여 안전한 공개SW 활용을 돕는 가이드라인 제공.

(내용) 전반적인 추진단계는 정보화사업 단계별 관리점검 가이드 3.0에서 제공하는 관리절차를 따르고 있으며, 각 단계에서 공개SW가 사용되는 사업의 경우 필요한 검토요소를 제시

[https://www.oss.kr/oss\\_guide/show/b9586acc-4cf4-4c54-b849-fc76da229fd3](https://www.oss.kr/oss_guide/show/b9586acc-4cf4-4c54-b849-fc76da229fd3)



[www.oss.kr](http://www.oss.kr)

# 공개소프트웨어 연구개발 실무 가이드라인

(목적) 공개SW 개발 방식의 국가 연구개발 사업을 수행하는 기관에서 프로젝트 수행 단계에서 준수해야 할 절차와 행동을 제공

(내용) 공개SW 연구개발 과제에 요구되는 사업계획서 작성 단계, 공개SW 연구개발 수행 단계, 공개SW 커뮤니티 운영 단계로 구분하여 국가 연구개발 사업의 수행기관에서 참고할 수 있는 가이드 라인 제시

[https://www.oss.kr/oss\\_guide/show/c05f6485-b8bd-49fd-9f35-58a0ec079235](https://www.oss.kr/oss_guide/show/c05f6485-b8bd-49fd-9f35-58a0ec079235)



[www.oss.kr](http://www.oss.kr)

# 기업 공개SW 거버넌스 가이드(OpenChain 2.0 해설서)

(목적) 다양한 오픈소스를 활용하게 되는 기업의 제품 생산과정에서 발생할 수 있는 소프트웨어 공급망의 오픈소스 컴플라이언스를 달성하기 위해서 필요한 활동을 제시

(내용) 2020년 국제표준인 ISO 5230으로 등록된 오픈체인 2.1을 국내 기업이 보다 쉽게 이해하고 스스로 위험을 관리할 수 있도록 오픈소스 컴플라이언스를 달성하기 위해 수행해야 하는 최소한의 요구사항을 설명하고 자세한 준수 방법을 제시.

[https://www.oss.kr/oss\\_guide/show/7050bff0-d06b-43f0-99a6-9975afcd486f](https://www.oss.kr/oss_guide/show/7050bff0-d06b-43f0-99a6-9975afcd486f)



[www.oss.kr](http://www.oss.kr)

# HAMONIKR

함께 오픈소스의 미래를 믿고 달려가고 있는 멋진 팀원들 때문에  
여기까지 오게 되었다고 생각합니다.

앞으로 더욱 좋은 오픈소스 생태계를 만들기 위해서 노력하고 팀  
원들과 함께 오픈소스 개발자로 자부심을 느낄 수 있도록 만들겠  
습니다.

“하모니카OS” 유튜브 검색 후 채널 많이 이용해 주시기요  
구독과 좋아요는 큰 힘이 됩니다.

감사합니다.



인베숨 김형채 대표 / [hckim@invesume.com](mailto:hckim@invesume.com)