

2. Open Source SW: Why and What

Hyunchan, Park

<https://github.com/hyunchan-park/osscourse>

Division of Computer Science and Engineering

Chonbuk National University



강의 일정

주차	월 (100분)	수 (50분)
1 (9/1)	Introduction	오픈소스 SW 개요
2	오픈소스 SW 개발 방법	추석
3	오픈소스 커뮤니티	오픈소스 라이선스
4	버전 관리 도구 Git 1	버전 관리 도구 Git 2
5 (9/30)	버전 관리 도구 Git 3	버전 관리 도구 Git 4
6	GitHub 1	한글날
7	GitHub 2	개인 프로젝트 1: 프로젝트 선정 및 계획
8 (10/21)	중간고사	중간고사
9	코드 분석 1	코드 분석 2
10 (11/4)	코드 분석 3	코드 리뷰
11	개인 프로젝트 2: 대상 프로젝트 코드 분석 결과 발표 및 기능 구현 방향, 커뮤니티 참여 방안 발표 및 피드백	
12	클라우드 기반 개발 환경 구성 1	클라우드 기반 개발 환경 구성 2
13 (11/25)	CI/CD 도구 및 활용	문서화 도구
14	개별 프로젝트 진행	
15 (12/9)	개인 프로젝트 3: 최종 프로젝트 발표, 기능 구현 소개 및 시연, OSS 커뮤니티 활동 소개	



Agenda

- OSS? Why?
- Open Source Definition
- Myths



OSS? Why?



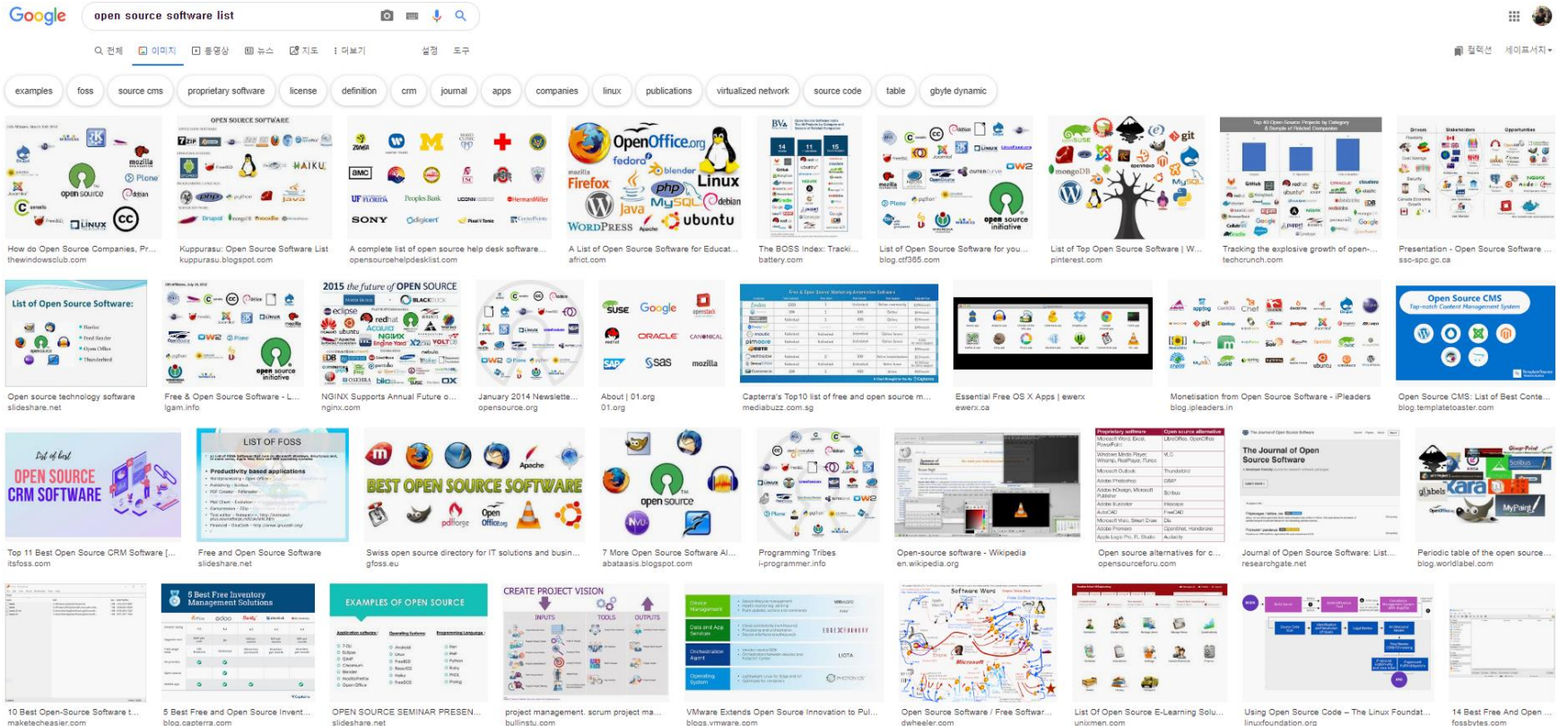
OSS? Why?

- 다양한 영역에서 수많은 Open Source SW가 사용되고 있음



OSS? Why?

- 로고들 나와있는 그림들이 너무 많아서 고르기도 어렵다...



OSS? Why?

- 64개 국가의 1313개 기업 대상 조사 (2015~2016)
- 65% 기업이 OSS를 활용 (increased from 60% @2015)
 - 65% 어플리케이션 개발 과정을 빠르게 하는데 활용
 - 55% 서비스 제공 인프라를 구축하는데 활용
- 67% 개발자들에게 OSS 사용 및 참여를 권장함
- 59% 최첨단 기술 경쟁력을 갖기 위해 OSS를 활용

* Future of Open Source Survey Results, North Bridge + Black Duck, 2016
<https://www.slideshare.net/blackducksoftware/2016-future-of-open-source-survey-results>





THE OPEN SOURCE ADVANTAGE

More and more strategic

2015

2016

Access to source code

Competitive Features

Ability to customize
and fix

Freedom from vendor
lock-in

Freedom from vendor
lock-in

Quality of solutions

TCO

Ability to customize
and fix

* Total Cost of Ownership: 초기 투자+유지비



SECTION 2
DRIVERS



OPEN SOURCE GETS PRACTICAL

Top 3 Technology Areas By Year

2015



Cloud
Computing



Big
Data



Operating
System

2016



Operating
System



Database

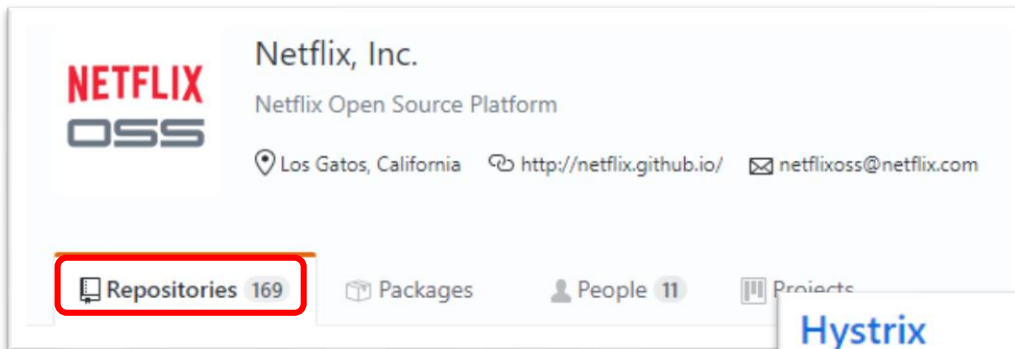


Development
Tools/Software
Development
Lifecycle



SECTION 1
STRATEGY

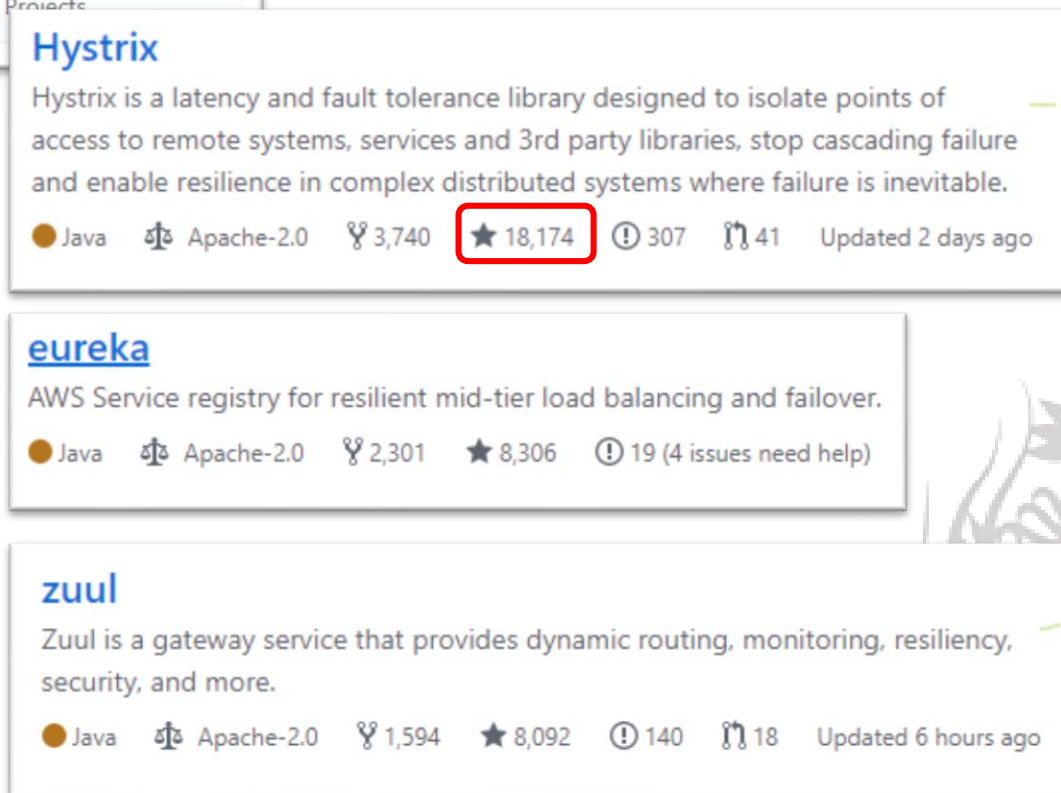
Case study: Netflix OSS



- 169개 Repository on GitHub
- <https://github.com/Netflix>
- <https://netflix.github.io/>



- 서비스의 Reliability 를 검사하기 위해,
- 무차별적, 인위적으로 서비스 컴포넌트들(VM, container)을 다운시킴
 - (참고: Chaos Engineering)



Case study: Netflix가 OSS를 활용하는 이유

- **Paying back**

- 우리도 얻은 게 많으니 은혜를 갚아야지요.

- External community contributions

- 외부 개발자의 참여를 유도할 수 있습니다.

- Improved code and documentation quality

- “Social coding” 의 압력으로 엔지니어들이 체계적이고 깔끔하게 코드와 문서를 작성합니다.

- Durability

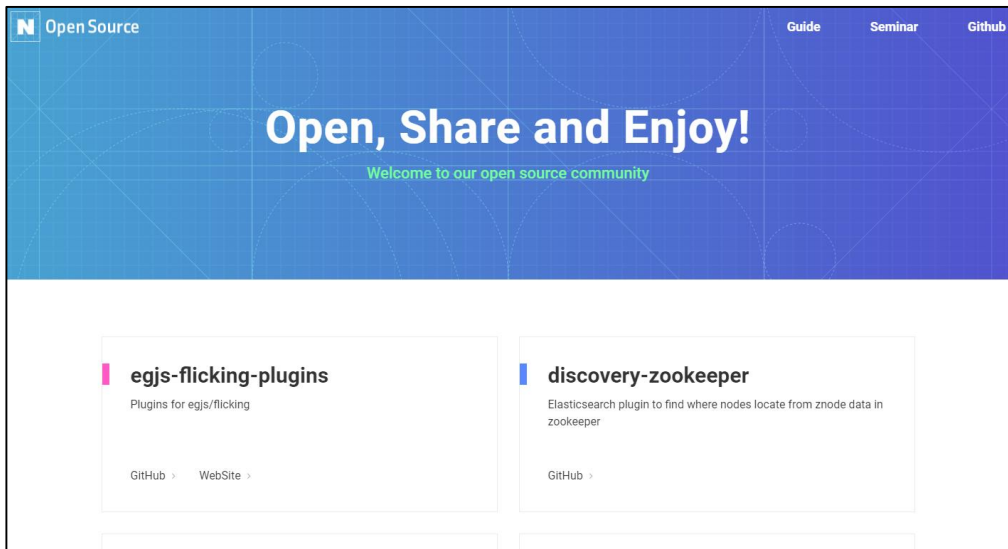
- 다수 개발자의 참여로 시간이 지남에 따라 코드가 더 견고해집니다.

- **Develop the AWS eco-system**

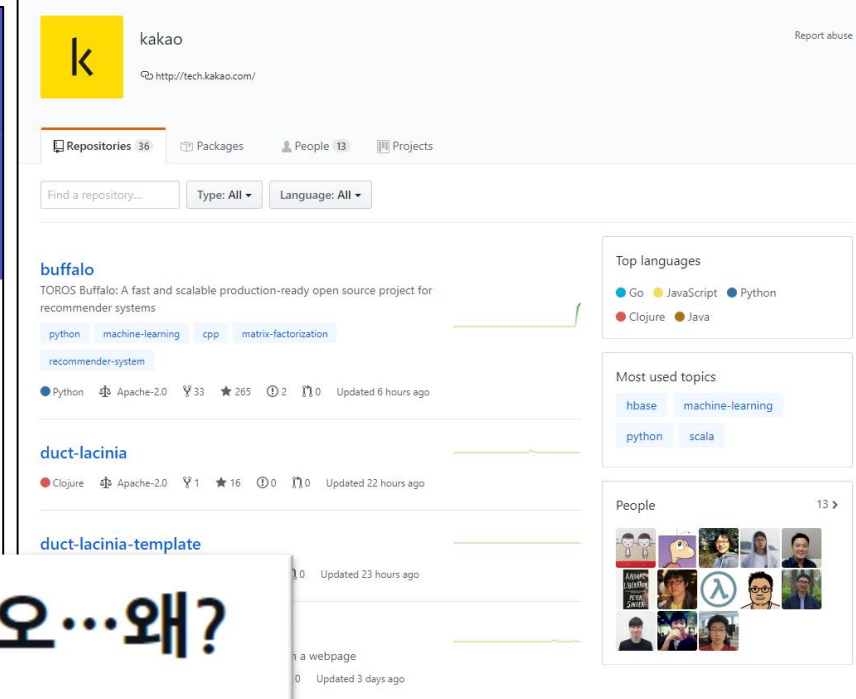
- 처음 AWS 클라우드에 모든 서비스를 이전하기 까지 시행착오가 많았습니다.
- 이런 경험과 해결방안들을 공유하여 AWS 사용자 규모와 커뮤니티를 늘리면, 결국 다시 혜택이 되돌아올 것입니다.

Case study: Naver and Kakao

<https://naver.github.io/>



<https://github.com/kakao>



지도·내비 공간 여는 네이버-카카오...왜?

구글지도 견제...빅데이터 경쟁력 강화 효과도

백봉삼 기자 | 입력: 2016/12/06 09:58

네이버-카카오, AI로 불붙은 'API 공개' 경쟁 플랫폼 확대...국내·글로벌 AI 시장 '각축'

입력 2017.07.04 17:39



Case study: Naver and Kakao

왜 오픈소스 프로젝트로 공개하는가

오픈소스 생태계의 발전을 위해

네이버가 오픈소스로부터 받은 도움에 보답합니다

더 좋은 소프트웨어로 발전하기 위해

네이버를 벗어난 다양한 환경과 요구사항에 대응하면서
공개 전보다 더 견고한 소프트웨어로 성장할 수 있습니다

왜 오픈소스 컨트리뷰션을 하는가

유지 보수를 더 편하게 하기 위해

컨트리뷰션하면 업데이트마다 패치해야하는 수고를 덜어줍니다

잡질을 줄이기 위해

같은 문제로 고생하는 일이 중복되는 것을 막을 수 있습니다

* 박은정 Open Source Manager, Naver, DEVVIEW 2017

<https://www.slideshare.net/devview/111open-share-enjoy-80840971>



그래서 결국 왜?

- 기업
 - Freedom from vendor lock-in
 - 특정 기업의 서비스나 솔루션에 자사 서비스가 의존성을 갖지 않음
 - E.g. Android가 갑자기 유료화되면? 버전업에 돈을 요구하면? 핸드폰 제조사들은?
 - Quality of Solutions
 - 전세계 개발자들이 함께 만들고, 유지하는 고품질의 SW를 즉각 도입할 수 있음
 - OSS 커뮤니티에서 활동하는 검증된 인력을 고용할 수 있음
 - Ability to customize and fix
 - 그럼에도 불구하고 내 필요에 따라 개작하거나 수정할 수 있음
 - (핵심 역량은 “우리 서비스” 를 만드는데 집중)
 - Low TCO
 - 초기 개발 비용, 유지 관리 비용이 적음 (Really? Not yet proved, I think)
- 이미 성공적인 “Business & SW development model” 로 검증됨

그래서 결국 왜?

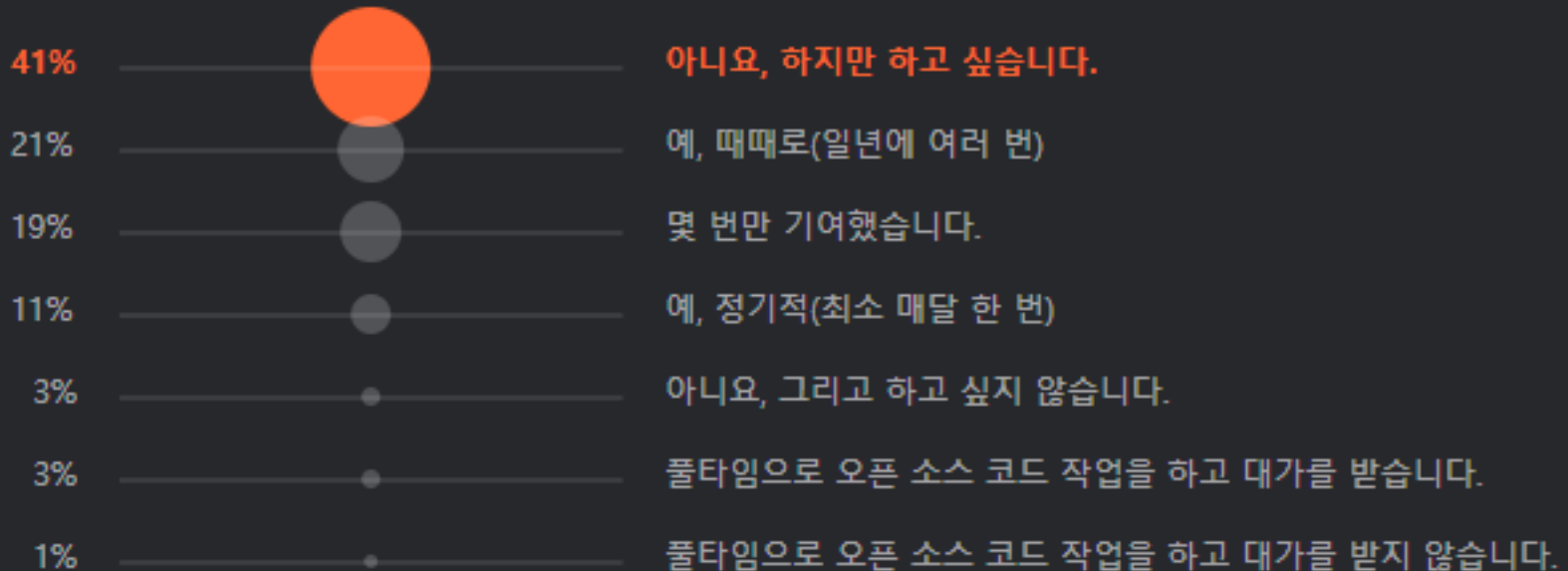
- 개발자 개인
 - 배움
 - 실제 기업에서 사용되는 수준의 코드를 얼마든지 보고 배울 수 있음
 - 어떤 개발자도 누구에게나 배울 점이 있음
 - 협력
 - 어떤 개발자와도 협력할 수 있음
 - 홍보
 - 나의 역량을 인증해주는 society를 가질 수 있고,
 - 또한 다양한 홍보의 기회를 가질 수 있음
- 무엇보다...함께 하면 더 재밌다!! (Online Multi-player Game)

“혁신의 속도를 따라잡자!”

- 너무도 빠르게 변화하는 IT 기술 업계
 - 최근 10년 사이: Mobile, SNS 를 시작으로, 빅데이터, AI 까지
 - 모두가 모든 것을 공부하고 만들어내는 것은 이제 불가능
- 혁신이 이루어지는 공간이 “성당” 에서 “시장” 으로 옮겨가고 있음
 - “성당과 시장”: 에릭 레이먼드가 주장한 2가지 개발 모델 (1997)
 - 성당 모델: 출시 때에만 소스 코드를 공개하고 그 사이에는 제한된 개발자들만 소스 코드에 접근 (당시 많은 GNU 프로그램이 해당)
 - 시장 모델: 소스 코드가 인터넷으로 일반에 공개된 상태로 개발 (Linux)
 - (+) 그리고 곧바로 시장에 내놓을 수 있음!
 - 단순히 결과물의 공개가 아니라, 개발 과정의 공개에 초점을 맞춤
- 시장에서 이루어지는 OSS 개발 과정 자체가 혁신을 만들 것

현실은...

오픈 소스 프로젝트에 기여하십니까?



* 2019년도 개발자 에코시스템의 현황 (7000명 개발자 대상 설문)

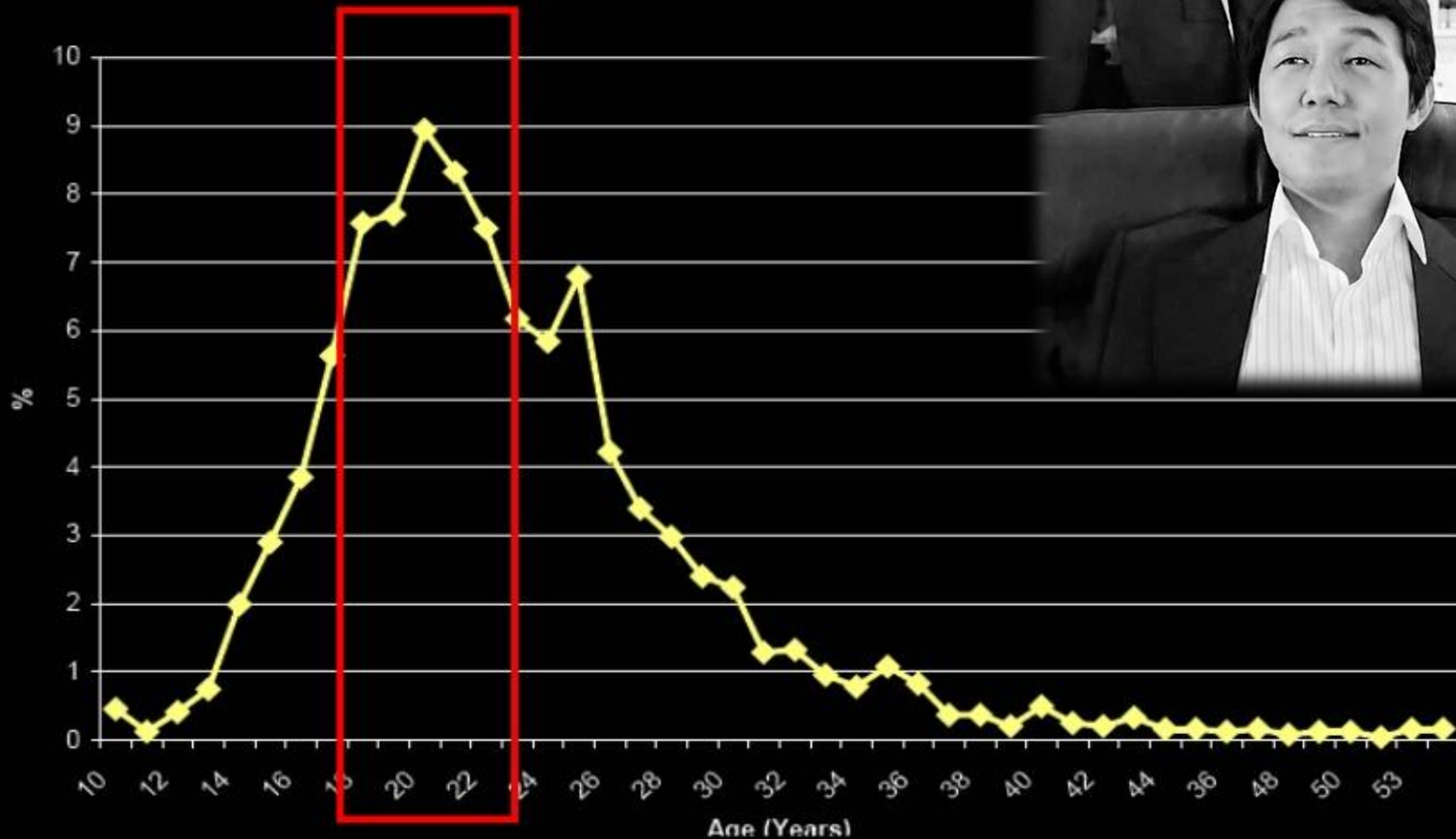
<https://www.jetbrains.com/ko-kr/lp/devecosystem-2019/>



참여 시기 및 나이

“거, 개발하기 딱 좋은 나이네”

Figure 3: Starting Age in OS/FS Community

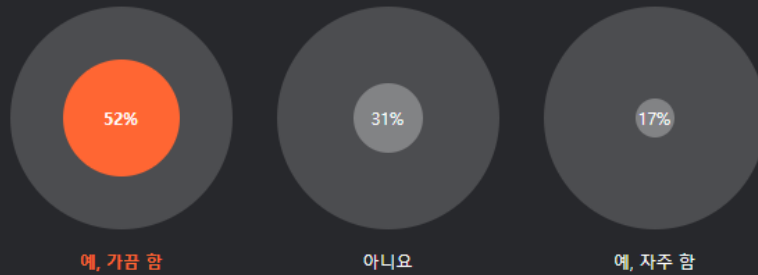


* The four myths of open source (윤석찬, 2013)

https://www.slideshare.net/Channy/the-four-myths-of-open-source?from_action=save

* just for fun

꿈 속에서도 코딩을 하십니까?



직장에서 (기본적인 활동으로) 코딩하는 시간이 많을수록 꿈속에서도 코딩을 하는 경향이 많았습니다.



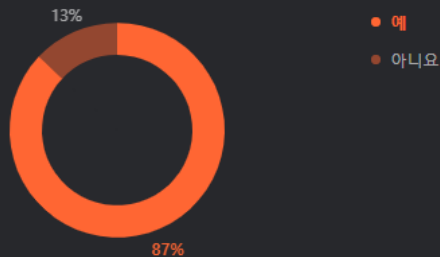
귀하의 차량은 어떤 연료를 사용합니까?



'그 밖의' 연료는 대부분의 경우 에탄올 또는 알콜로 판명되었습니다.



주말에 코딩을 하십니까?



자영업자, 학생 및 교육생은 다른 사람들보다 주말에 좀 더 자주 코딩을 합니다.



미래에 AI(인공지능)가 개발자를 대체할 것이라고 생각하십니까?



전문적으로 머신 러닝을 사용하는 사람들은 이를 취미로 사용하는 사람들보다 AI에 대해 더 회의적입니다. 전자에 속한 사람들 중 27%는 AI가 절대로 개발자를 대체하지 않을 거라고 생각하는 반면, 후자에 속한 사람들 중에는 21%만 이에 동의합니다.

The Open Source Definition





Open Source Initiative

Guaranteeing the 'our' in source...

- Open Source Initiative
 - 1998년 2월, 브루스 페렌스와 에릭 레이먼드가 설립
 - Open Source Definition 발표
- 역할: 라이선스에 대한 OSI 인증
 - 해당 라이선스가 Open Source Definition 을 따르고 있는지 여부를 인증함
 - 현재 인증받은 라이선스는 83개
 - <https://opensource.org/licenses/alphabetical>
- 즉, 사실상 핵심은 라이선스

OSI 인증 예

Wmii



This software is licensed under the [MIT License](#) which has been approved by the OSI as a [valid Open Source License](#) according to the [Open Source Definition](#).

Template:Lowercase wmii (window manager improved 2) is a **dynamic window manager** for **X11**. It supports classic and tiling^[1] window management with extended **keyboard**, **mouse**, and filesystem based remote control. It replaces the workspace paradigm with a new tagging approach.

In its default configuration, wmii's interface is similar to that of the **Vi** text editor; the windows are controlled with Alt+H,J,K,L and can be combined with



The Open Source Definition (v1.9)

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

The Open Source Definition (v1.9)

1. 자유 재배포
2. 소스 코드
3. 2차 저작물
4. 원저작자의 소스시 코드 보존
5. 개인 및 집단에 대한 차별 금지
6. 사용 분야에 대한 차별 금지
7. 이용허락의 배포
8. 제품을 특정한 이용허락 금지
9. 다른 소프트웨어를 제한하는 이용허락 금지
10. 이용허락의 기술 중립 의무

- 오픈 소스 : 오픈 소스 혁명의 목소리 vol. 1,2, 한빛미디어
- 한글 번역과 해석이 있음

OSD 1~4: SW 의 빠른 발전을 위한 규칙들

1. 자유 재배포

- 해당 SW를 여러 SW 와 함께 모아 판매하거나 무상 배포 가능해야 하고, 이때, 비용을 요구할 수 없음

2. 소스 코드

- SW에는 이용하기 용이한 소스 코드를 포함해야만 하며, 배포도 허용해야 함. 무료 다운로드 형태로 제공하는 것도 가능함

3. 2차 저작물

- 개작과 2차 저작물의 창작을 허용해야 하고, 그 결과물이 기존과 동일한 이용허락(라이선스)를 사용해 배포되는 것을 허용해야 함

4. 원저작자의 소스 코드 보존

- 빌드 시에 적용되는 패치 파일과 그 소스 코드를 함께 배포하는 것을 허용하며, 이때는 원저작자의 소스 코드를 수정하는 것을 제한할 수 있음

OSD 5,6: 차별 금지, 7. 간접적 제한 금지

5. 개인 및 집단에 대한 차별 금지

- (해설: 미국 등 몇몇 국가는 특정 종류의 SW 수출이 제한됨. 라이선스에 이러한 사항에 대해 주의를 줄 수 있지만, 라이선스 자체에서 제한하는 것은 금지)

6. 사용 분야에 대한 차별 금지

- (해설: 상업적 이용을 방해하는 조항이 들어가는 것을 금지함)

7. 이용허락의 배포

- 배포가 이루어지면, 자동으로 SW에 부착된 권리가 재배포받은 모든 사람에게 적용되어야 한다
- (해설: 기밀 유지 계약같은 간접적인 수단으로 SW의 폐쇄를 금지함)

OSD 8,9,10: 차별 금지

8. 제품을 특정한 이용허락 금지

- 특정 배포판에 소속된 경우에만 허용하거나, sw의 일부분을 분리해 사용하는 것을 제한하는 행위를 금지함

9. 다른 소프트웨어를 제한하는 이용허락 금지

- 해당 sw와 함께 배포하는 다른 sw에 대해 제한을 포함해서는 안됨.
- (해설: 하나의 배포판으로 묶여 배포되더라도 각각의 sw는 서로 다른 라이선스를 가질 수 있음을 보장함)

10. 이용허락의 기술 중립 의무

- 라이선스는 개별 기술이나 인터페이스 형태를 단정해서는 안됨.
- (해설) Click-wrap 같은 강제 조항을 사용하는 것을 방지
 - 동의할지 여부에 대해 팝업창을 띄워 명시적으로 승낙을 받는 행위 금지

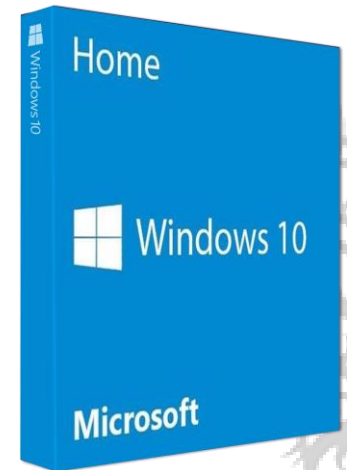
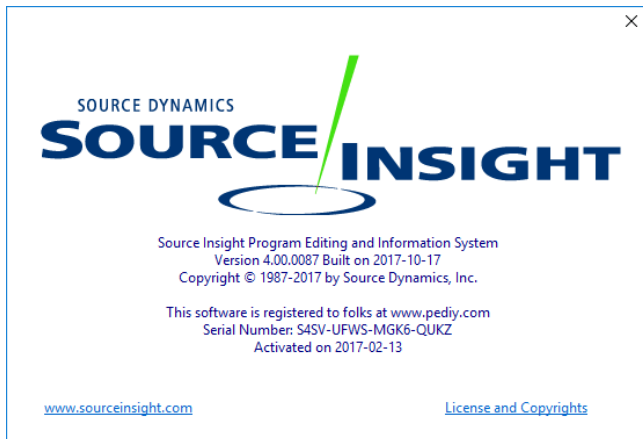
상용 소프트웨어와 오픈소스의 비교

구분	상용 SW	오픈 소스
비용분석	<ul style="list-style-type: none"> 초기 도입비용이 높은 유지보수 비용 및 시스템 개선비용 높음 	<ul style="list-style-type: none"> 초기 도입비용이 낮음 유지보수 비용 및 기능 추가 비용이 낮음
제약사항	<ul style="list-style-type: none"> 라이선스 계약으로 인한 제약사항 (사용료 지급) 	<ul style="list-style-type: none"> 사용상의 제약사항 없음 (무료)
배포형태	<ul style="list-style-type: none"> 라이선스 계약으로 인한 바이너리 제공 소스코드는 기업비밀로 유지함 	<ul style="list-style-type: none"> 소스코드가 공개 라이선스 조건하에 자유롭게 배포,수정
성능분석	<ul style="list-style-type: none"> 비교적 큰 시스템 환경에서의 높은 성능 나타냄 	<ul style="list-style-type: none"> 다양한 환경에서 최적화된 설정으로 높은 성능치를 나타냄 (Intel, PPC, s390등)
기술성	<ul style="list-style-type: none"> 문제점 발생시 폐쇄적인 운영으로 취약점 보유 	<ul style="list-style-type: none"> 소스코드의 공개로 빠른 문제점 해결 유지보수 및 업그레이드 용이, 독점피해 방지
확장성	<ul style="list-style-type: none"> 시스템 환경에 따라 호환성은 보장 높은 적용비용을 지불해야 함 제한된 시스템 운영환경에서 용이 	<ul style="list-style-type: none"> 소프트웨어간 적용비용이 낮음 기능추가 비용이 낮음
공급권	<ul style="list-style-type: none"> 최초도입 개발업체 또는 벤더에게 공급 개발업체 문제시 도입 고객에게도 심각한 영향을 미침 	<ul style="list-style-type: none"> 동일 솔루션에 대한 다수의 업체로부터 지원 및 공급이 가능, 사용자의 선택권이 넓음

클라우드 기반환경을 플랫폼 형태로 만들어 재판매 할 경우 강력한 원가절감, 솔루션 유연성을 가짐(예: 고객 유지보수 비용으로 포함)

상용 SW를 선택하는 것이 유리한 경우

- 상용의 사용 방법이 훨씬 쉬운 경우
- 상용이 사실상의 표준인 경우
- 상용의 지원이 우수하거나 기능이 더 풍부한 경우
- 하드웨어 호환성이 우수한 경우(예: Window XP HW Driver 지원)
- 보증/책임보상이 중요한 경우
- SaaS 기반 서비스가 필요할 때



Open Source SW Foundations

- 왜 재단이 만들어질까?
 - 프로젝트를 운영할 수 있는 기본적 인프라 제공
 - 컨퍼런스, 이벤트, 기부금 등 세금 관련된 일들 처리
 - 라이선스, 특허, 표준 약관, 상표 관리 등 법률적 문제 대응
 - 기술 중심 워크그룹 운영
 - (기본: 각자 관심있는 영역에 따라 모임)
- 참고. Open Software Foundation (OSF)
 - UNIX OS의 Open standard를 작성하기 위해 결성된 재단
 - FSF, OSI와 무관함

주요 OSS Foundations

- Apache Software Foundation
 - 가장 대표적인 오픈소스 재단. 350개 이상의 프로젝트 관리
- Free Software Foundation
 - 무료 운영 시스템 구현에 목표를 둔 GNU 프로젝트만 관리
- Linux Foundation
 - Linux Kernel 관리. 최근 SDx, IoT, Embed, Cloud 등의 다양한 기술 협력



주요 OSS Foundations

- Software Freedom Conservancy
 - BusyBox, Git, Samba, Wine 등 33개 프로젝트 관리
 - GPL 의무화를 위한 GPL Compliance Project를 추진
- OpenStack Foundation
 - 클라우드 시스템 구축 SW인 Open Stack 관련 수십개 프로젝트 관리
- Etc.
 - Python SW Foundation, WikiMedia, Django SW, Eclipse,



성공적인 OSS 프로젝트들

Linux



- GNU에서 관리하고 있는 오픈소스 운영체제
- X86 기반에서 벗어나 임베디드 기기, 모바일 등 다양한 하드웨어에 포팅
- 데스크탑이나 웹서버 위주에서 현재는 클라우드, 빅데이터 등의 분야에서 사실상의 표준 운영체제로 자리 잡음
- 모든 안드로이드 스마트폰 단말(한국 90.1%)
- 전세계 슈퍼 컴퓨터의 93.8%(2013년 11월 기준)
- Google, Twitter, Facebook, Amazon 등
- 개발자 8,000명, 회사 800개, 1,500만줄의 코드
- 3~6개월 주기의 커널 업그레이드

〈서버용 OS 점유율, 2011〉

이름	점유율
Red Hat	28.6%
Debian	12.2%
CentOS	10.6%
Fedora	9.5%
Ubuntu	8.7%
SuSE	11.6%
Others	18.6%



Source : Worldwide Linux Server Operation System Environment by Vendors, 2006-2010, March IDC 2011



성공적인 OSS 프로젝트들

MySQL(MariaDB)



- 전세계적으로 가장 많이 사용되고 있는 오픈소스 데이터베이스의 하나
- 페이스북 등 거의 대부분의 웹 서비스 업체에서 운영 데이터 저장용으로 사용
- 현재는 클러스터링, 분산(샤딩), 복제 및 백업 등 엔터프라이즈 환경에서 필요로 하는 고급 기능까지 지원
- 오라클이 인수(10억 달러) 후 "오픈코어 전략" 표방
- 오픈소스 결과물의 확장 기능이나 엔터프라이즈 기능 공개하지 않고 있음
- 이에 반발한 마이클 몬티가 막내 딸 이름을 붙여 MariaDB 시작

Apache HTTP Server



- Apache 재단의 대표 프로젝트로 전 세계적으로 가장 많이 사용(46% 점유율)하는 웹서버
- NCSA의 HTTPd 코드를 기반으로 Linux에서 사용할 수 있도록 웹 서버 코드를 재 작성
- 커뮤니티 그룹과 미 델라웨어사와 합작하여 Apache Software Foundation 설립 (1999)

성공적인 OSS 프로젝트들

• Hadoop

- 빅데이터 분야의 대표적인 오픈소스 프로젝트
- 거의 모든 빅데이터 업체들이 사용하고 있는 기반 플랫폼
 - (예) Hortonworks, Cloudera, MapR, IBM, Oracle
- 빅데이터 관련 소프트웨어들의 생태계 역할
 - (예) Pig, Hive, HBase
- 모바일, 사물인터넷에 기반한 급속한 데이터 증가에 따라 더욱 중요도 증가



• Openstack

- 클라우드 업체인 Rackspace와 NASA가 협력하여 만든 오픈소스 클라우드 운영 시스템
- 클라우드 시스템 구축에 필요한 컴퓨팅, 네트워크, 스토리지 자원에 대한 가상화 및 운영 관리는 물론이고 보안, 워크로우 등 거의 모든 요구사항 지원하는 사실상의 표준
- 클라우드 업체는 물론 가상화, 하드웨어, 네트워크, 스토리지 등 다양한 벤더들이 참여 중



• MongoDB

- 기존 RDB와는 다른 NoSQL 기반의 데이터베이스
- 빅데이터 시대의 도래와 함께 비정형 데이터를 대량으로 처리할 수 있는 장점 부각
- RDB와 유사한 인터페이스, 폭 넓은 사용자 저변, 높은 성능으로 NoSQL 분야에서 사실상의 업계 표준



성공적인 OSS 프로젝트들

- Pentaho
 - BI(Business Intelligence) 및 데이터 통합(ETL 등) 분야의 오픈소스 강자
 - Hadoop과 NoSQL 지원을 통해 빅데이터 시장 진입
 - 자본 유치와 고객 확보에도 성공
- PostgreSQL(Enterprise DB)
 - MySQL(MariaDB)과 함께 주요한 오픈소스 RDB 프로젝트의 하나
 - Enterprise DB를 통해 상용화된 버전 제공
 - Oracle DB와 뛰어난 호환성
- Wordpress
 - 오픈소스 블로그 플랫폼이자 WCM(Web Contents Management) 플랫폼
 - 전 세계 웹사이트의 22%, WCM 시장의 60% 이상 점유
- WSO2
 - Apache 라이선스 기반으로 한 오픈소스 미들웨어 스위트
 - ESB, BPM, BRM, API Manager, CEP 등의 컴포넌트

성공적인 OSS 프로젝트의 현황

Name	Total contributors	Active contributors	Monthly commits	Active organizations
glibc	370	20-40	100-550	5-25
MongoDB	340	20-45	300-850	5-10
MySQL	360	50-90	350-700	2-15
OpenStack	3800	400-700	2000-6000	100-200
Linux Kernel	15870	600-1100	3800-6600	300-400

[* 2018, Diversity, Productivity, and Growth of Open Source Developer Communities](#)



Myths



Five Myths

1. OSS는 누구나 보안 취약점을 발견하고 공격할 수 있지 않을까?
2. 코드 공유하고 커뮤니티 모임하고, 그게 오픈소스하는 주 목적 아니야?
3. 미국이나 영어권 사람들이나 할 수 있는거 아냐?
4. 그런 건 고급 개발자들이나 하는거 아닌가?



1. OSS의 보안성

기존 금융권 시스템의 구조를 깨버린 카카오뱅크의 리눅스 채택

카카오뱅크의 메인 시스템이 유닉스가 아닌 리눅스를 채택해서 만들어졌다는 얘기는 OS 및 그 OS를 운영하기 위한 하드웨어의 가격이 유닉스 시스템의 그것보다는 상대적으로 저렴하다는 얘기가 된다. OS 자체의 비용은 없을 테고(RHEL을 쓴다면 유료겠지만 그래도 유닉스보다는 쌀 테니) 하드웨어 역시 유닉스 전용 하드웨어보다는 상대적으로 저렴할 테니 말이다(물론 BM이나 HP, 델 등의 유닉스 하드웨어 벤더들도 리눅스를 운영하기 위한 서버를 생산해서 판매한다. 그런데 AIX, HP-UX 등의 유닉스용 하드웨어보다 상대적으로 저렴하다).

여차하면 엔지니어가 직접 부품을 가져다가 조립해서 서버를 만들 수도 있다(실제로 구글의 경우 벤더에서 제공하는 서버를 쓰는 것이 아닌 자체적으로 조립해서 사용한다. 그래서 하드웨어 수급 비용을 낮추는 경우도 있다).

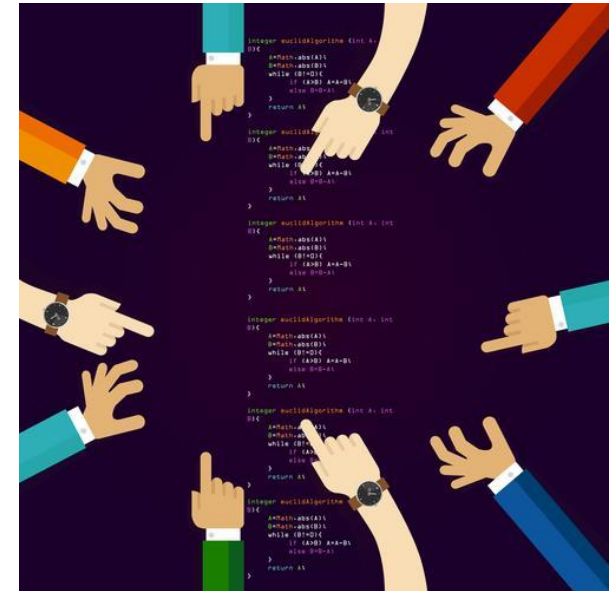
- 카카오뱅크 (2018/1/7 기준)
 - 예적금 총액 5.19조원,대출 총액 4.76조원

- 비트코인 (2019/9/3)
- 시가총액: \$1856억 (약 220조원)
 - KOSPI 의 약 10% 수준



1. OSS의 보안성

- OSS가 높은 보안성을 획득할 수 있는 이유
 - 많은 코드 리뷰어, 많은 사용자가 검증에 참여함
 - 다양한 분야의 개발자가 취약점을 분석할 수 있음
- 그러나 상용 SW보다 더 높다고 할 수는 없음
 - 보안 사고는 다양하게 계속 발생하고 있음
 - 또한 이제 OSS 도입이 활발히 일어나는 시기라, 보안 관리까지 잘 수행하고 있지는 않음
 - 사고가 발생한 경우, 책임을 직접 떠안게 됨



* 오픈소스 보안 현황과 시사점, 김혜영 BSDK 차장

1. OSS의 보안성, 항상 유의해야 함

- 그러나...보안은 항상 의심 또 의심해야 함
 - 아주 많은 리뷰어가 살펴보는 코드는 보안성이 높지만,
 - 소수의 리뷰어만 관심을 갖는 코드, 많이 사용되지만 간과한 부분 등
 - 언제나 보안 취약점의 위험은 존재할 수 있음
- 실제 OSS에 다양한 보안 취약점에 문제가 있을 수 있음
 - 2016년 1071개 SW에 대한 조사 결과,
96%가 OSS였으며, 그 중 60%가 보안 취약점을 갖고 있었음
 - 또한 67% OSS 개발 업체가 보안 취약점에 대한 모니터링을 수행하지 않음

Organizations
are not
effectively
dealing with
open source
security threats



67% OF analyzed applications using open source had vulnerabilities in the components used

On average, vulnerabilities identified in these applications have been publicly **known for over four years**

Financial industries contained 52 vulnerabilities per application, and **60%** of those applications contained **high-risk vulnerabilities**

Retail and E-commerce had the highest proportion of applications with high-risk vulnerabilities, with **83% of audited applications** containing **high-risk vulnerabilities**

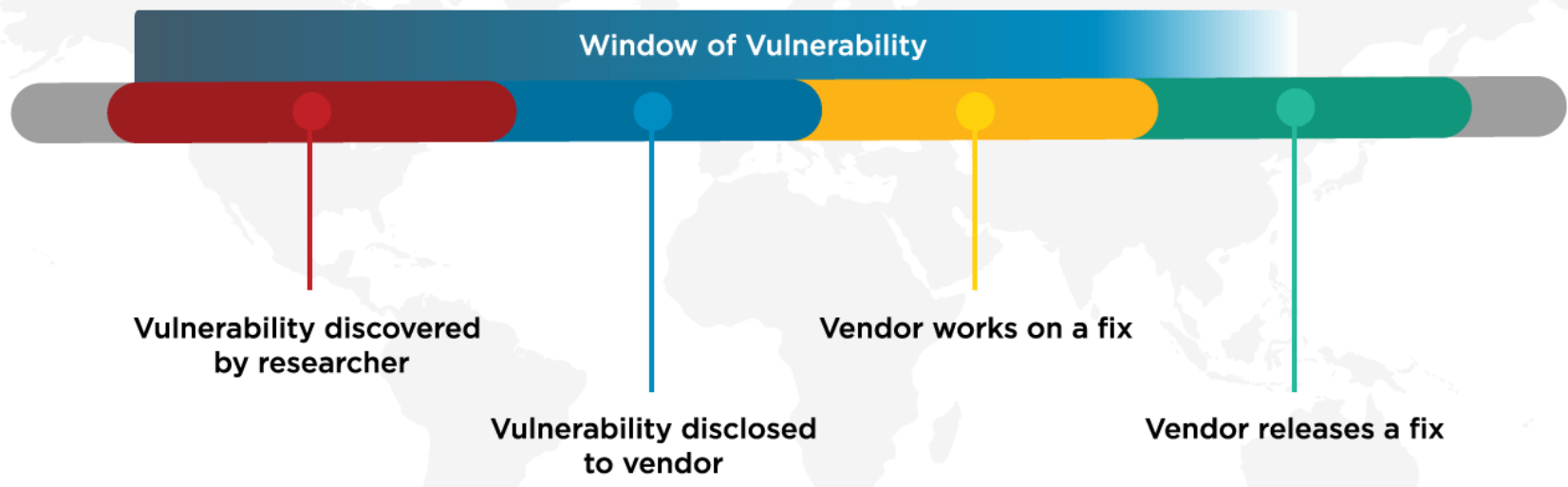
<https://www.helpnetsecurity.com/2017/04/20/open-source-security-threats/>



1. OSS의 보안성, 항상 유의해야 함

- 대표적인 예: Zero-day Attack
 - 배포된 당일, 공격자가 먼저 취약점을 발견하고 공격
 - 패치가 제작, 배포, 설치되기 전까지 무방비
 - 새로운 버전을 너무 빠르게 업데이트하는 것은 일반적으로 좋지 않은 방법

Zero Day Vulnerability Timeline



참고: 5가지 오픈소스 보안 관리방안

- 오픈소스 인벤토리 구축
 - 오픈소스 보안관리의 시작은 어떤 오픈소스가 자사 애플리케이션에서 사용되었는지 아는 것부터 시작함
- 알려진 보안취약점의 파악
 - 기업에서 사용된 오픈소스에서 기존의 알려진 보안취약점을 맵핑하여 보안취약점 현황을 파악함
- 리스크 식별
 - 보안취약점에 대한 리스크를 식별하고 프로젝트 우선순위와 보안취약점의 위험도를 기반으로 조치대상을 우선순위화하여 분류함
- 보안취약점 조치 이행
 - 우선순위로 분류된 보안취약점을 기반으로 조치 이행 및 이력관리를 함
- 새로운 보안취약점 모니터링
 - 기업의 위협요소가 될 수 있는 오픈소스 보안취약점을 지속적으로 모니터링하면 빠른 탐지 및 대응이 가능함

2. OSS의 주 목적은?

- Why do you join Open Source?
 - Intellectually stimulating : 44.9% (지적 호기심)
 - Personal Skill-up : 41.3% (스킬 향상)
 - Work functionality : 33.8%
 - Supporting to be public source code : 33.1% (OSS 지원)
 - Non-work functionality : 29.7%
 - Pleasure on community : 20.3% (커뮤니티 참여)
 - Improving career : 17.5%
 - Hating commerce software : 11.1% (...)
 - Reputation : 11.0%

* 2012 오픈 소스의 4가지 미션, 윤석찬

© 2014 Pearson Education, Inc. or its affiliate(s). All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without prior written permission from the publisher.

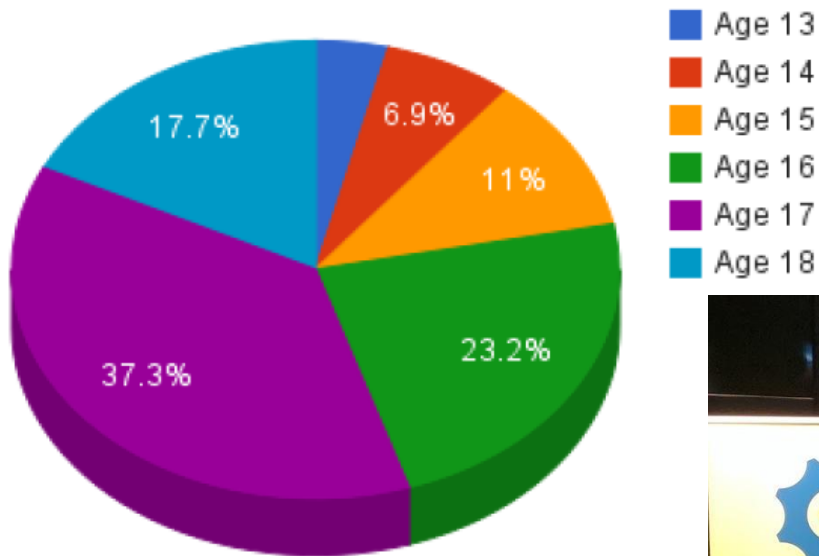


* 2012 The beauty of Open Source


4. 고급 개발자만?

- Google code-in
 - 중고등학생 대상, 1주일 가량 실제 OSS 프로젝트에 참여해 활동

Age Distribution for Google Code-In Students



4. 고급 개발자만?

 **google-highly-open-participation-asf**

The Apache Software Foundation's GHOP Contest Page

[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#)

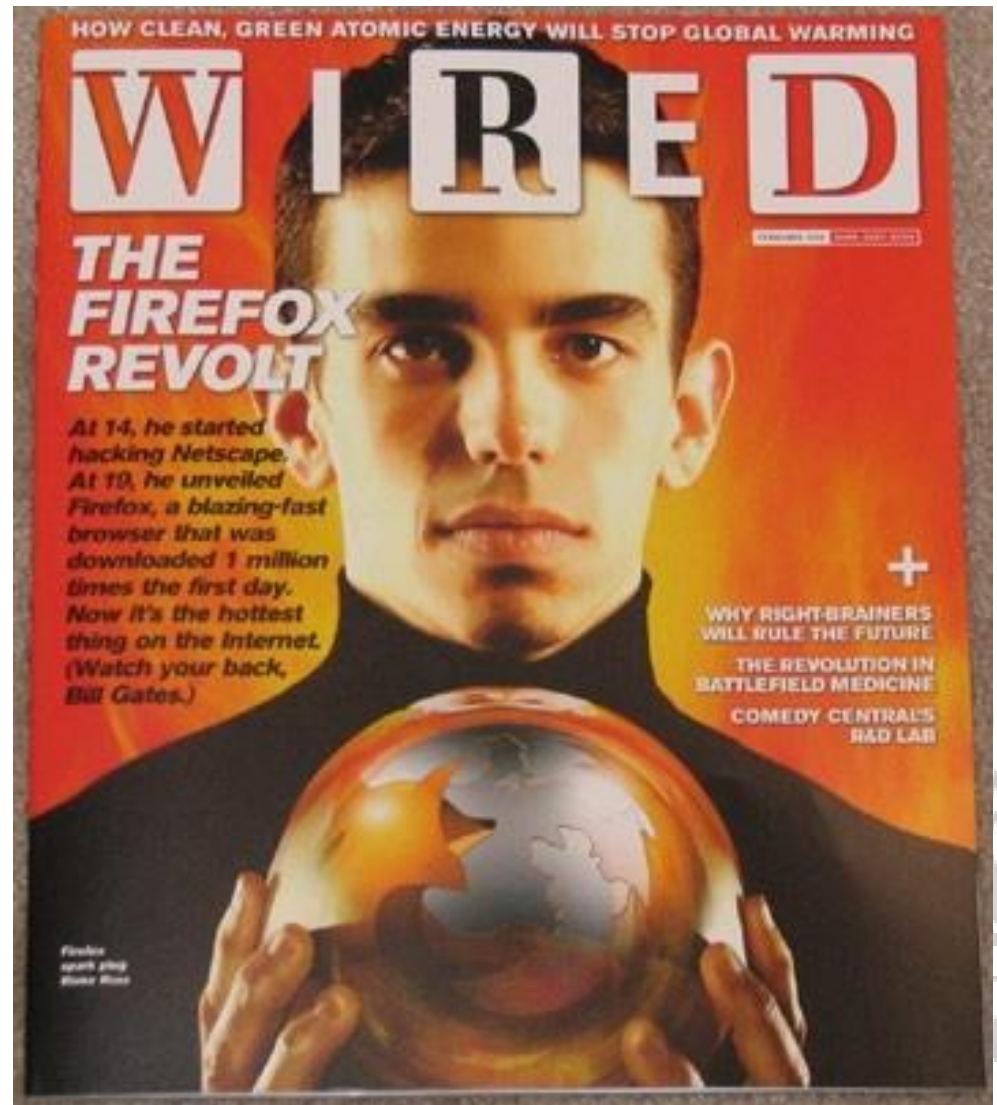
[New issue](#) | **Search** for | [Advanced search](#) | [Search tips](#)

1 - 39 of 39 [List](#) | [Grid](#)

ID	Status	Owner	ClaimedBy	Summary + Labels
5	Claimed	gerolf.seitz	zaicic	Wicket screen cast: GuestBook Training
1	Completed	bdelacretaz	yong...10gic.net	Design a logo for the Tika project tika logo
2	Completed	bdelacretaz	walkerp	Design two related logos for the Sling project sling microslings logo
3	Completed	gerolf.seitz	djwhitlow	Wicket screen cast: Hello World Training
8	Completed	gerolf.seitz	thenewme91	Wicket screen cast: Markup Inheritance Training
12	Completed	gerolf.seitz	higgylm	Wicket: Restlye the Wicket Examples UserInterface
14	Completed	gerolf.seitz	walkerp	Wicket: Create animated launch page for website UserInterface
15	Completed	gerolf.seitz	sarah.ohanesian	Wicket: Create Planet Wicket integrated blog stream UserInterface
16	Completed	bernd.fonderrmann	adelgado1313	JAMES Mail Server: Compile a FAQ list Documentation
17	Completed	bernd.fonderrmann	marty	JAMES Mail Server: Write an installation guide Documentation
18	Completed	bernd.fonderrmann	fr33m2n	JAMES Mail Server: Compare mail server customizability Outreach

4. 고급 개발자만?

- Blake Ross (1985~)
 - Co-founder of Mozilla Firefox
 - 14세 Netscape 코드 분석 (1998년 OSS화 된 이후)
 - 16세 Netscape 인턴
 - 19세 Firefox 출시



4. 고급 개발자만?

오픈소스가 어렵게 느껴지는 이유

내 실력으로 해도 될까..?

비웃지 않을까..?

내 코드가 옳은 코드일까..?

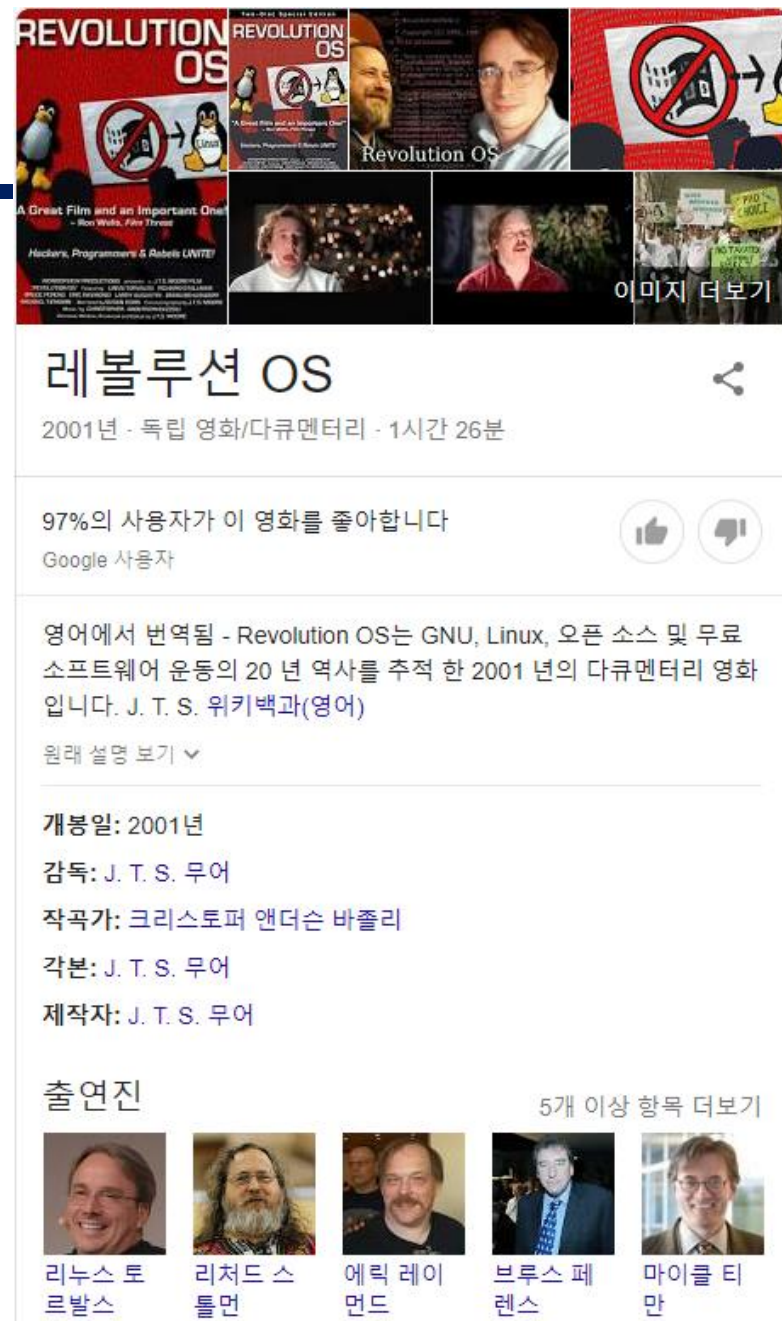
사실 이런 고민들은 하지 않아도 되는 고민입니다.

오히려 오픈소스에 기여하며 코드 리뷰도 받고 다른 사람의 코드를 보며 실력을 향상시킬 수 있습니다.

* 2018 10 오픈소스를 여행하는 히치하이커를 위한 안내서 NHN 이선희

개인 과제 #1 (명절을 맞이하여..)

- 영화보고 후기 쓰기
 - <https://youtu.be/4ZHloJVhcRY>
 - 1.25 배속으로 보면 1시간
- A4 1장 이내
 - 제목, 학번, 이름 간단히 기재
 - 워드 기본 서식 그대로 사용
- 기한: 9/15 (일) 23:59
 - 지각 감점: 5%p / day
 - 3주 내 제출해야 함
- 제출: LMS 시스템
 - “과제 1”



The image shows a YouTube video player for the movie "Revolution OS". The video title is "레볼루션 OS" (Revolution OS). The description mentions it's a 2001 documentary about the history of GNU, Linux, and open source software. The video has 97% likes from Google users. The cast list includes Linus Torvalds, Richard Stallman, Eric Raymond, Bruce Perens, and Michael Tieman. The video is 1 hour and 26 minutes long.

레볼루션 OS

2001년 · 독립 영화/다큐멘터리 · 1시간 26분

97%의 사용자가 이 영화를 좋아합니다
Google 사용자

영어에서 번역됨 - Revolution OS는 GNU, Linux, 오픈 소스 및 무료 소프트웨어 운동의 20 년 역사를 추적 한 2001 년의 다큐멘터리 영화입니다. J. T. S. 위키백과(영어)

원래 설명 보기 ▼

개봉일: 2001년
감독: J. T. S. 무어
작곡가: 크리스토퍼 앤더슨 바졸리
각본: J. T. S. 무어
제작자: J. T. S. 무어


출연진

5개 이상 항목 더보기

리누스 토르발스, 리처드 스티븐슨, 에릭 레이먼드, 브루스 페렌스, 마이클 티먼

(+) GitHub 가입하고 Student Pack 신청하기

- <https://education.github.com/pack>
- 혜택이 많다. 아주 많다.
- 신청하면 시간이 좀 걸릴 수 있으니 미리 해둡시다.

Which best describes your academic status? 

☒ Student ☐ Faculty

The GitHub Student Developer Pack is only available to students aged 13 or older.

What e-mail address do you use for school?

 **Pro-tip:** Selecting a school-issued email address gives you the best chance of a speedy review.

☒ hyunchan.park@chonbuk.ac.kr

✓ Chonbuk National University

+ Add an email address

How do you plan to use GitHub?

Please note, your request cannot be edited once it has been submitted, so please verify your details for accuracy before sending them to us.

Submit your information