**출처:** [**https://blog.naver.com/thl0818/221085219318**](https://blog.naver.com/thl0818/221085219318)

[**https://blog.naver.com/leeks219/221204070253**](https://blog.naver.com/leeks219/221204070253)

[**http://www.ciokorea.com/news/39723**](http://www.ciokorea.com/news/39723) **(IBM오픈소스)**

<http://www.itworld.co.kr/news/107891#csidx3658019b0b216939b2b3195f42e8bb3> (오픈소스 관련뉴스들 모아져있는 곳)

사례연구를 통한 오픈소스 소프트웨어 비용 산정방법에 대한 연구

신성진 이후재 류성열

한국정보처리학회 2008년 춘계학술대회

VOL. 15 NO. 01 pp. 0339 ~ 0342 2008 . 05

[출처] 오픈 소스의 장단점|작성자 정일동

**오픈소스 학위논문 볼수있는곳:** [**http://www.riss.kr/search/Search.do?detailSearch=false&searchGubun=true&strQuery=%EB%8F%99%EB%AC%BC%EC%9B%90&queryText=&exQuery=&colName=bib\_t&query=%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4&x=0&y=0&keyword1=&op1=AND&keyword2=&op2=AND&keyword3=&keywordOption=0**](http://www.riss.kr/search/Search.do?detailSearch=false&searchGubun=true&strQuery=%EB%8F%99%EB%AC%BC%EC%9B%90&queryText=&exQuery=&colName=bib_t&query=%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4&x=0&y=0&keyword1=&op1=AND&keyword2=&op2=AND&keyword3=&keywordOption=0)

**오픈소스 법적 측면:** [**https://opensource.guide/ko/legal/#why-do-people-care-so-much-about-the-legal-side-of-open-source**](https://opensource.guide/ko/legal/#why-do-people-care-so-much-about-the-legal-side-of-open-source)

**오픈소스 소프트웨어:** <http://www.itworld.co.kr/t/54649/%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0/106673#csidxcbc8741df782cab9f7160c91f69f39b>

**- 오픈소스의 정의**

오픈소스 소프트웨어(Open Source Software)는 소프트웨어의 소스코드가 공개되어, 누구나 소스코드를 자유롭게 사용하고 수정하여 재배포할 수 있는 소프트웨어이다. OSS, FOSS, 오픈소스, 공개 소프트웨어라는 용어가 사용되나 통상 '오픈 소스'라고 말한다.

Open Source Initiative에서는 10가지 조건으로 오픈 소스를 상세하게 정의하고있다.

**-오픈소스의 역사**

소프트웨어가 하드웨어인 컴퓨터와 분리되어 상업화되어 가면서 소스코드가 비공개로 전환되는 것에 대한 반발로 자유소프트웨어 운동이 일었는데 리처드 스톨만의 주도로 1983년 GNU 프로젝트가 시작되었다. OSI의 공동 설립자인 브루스 페렌스는 "기존에 있던 자유 소프트웨어 개념을 기업에 홍보하고 또 라이센스를 인증하기 위해 오픈소스 운동에 적합한 이름이다." 라고 설명했다.

지적재산권이 기업의 독점적 지위를 높이고 시장을 독차지하며 기술혁신을 가로막을 것이라는 우려로 1984년에는 자유소프트웨어재단(FSF, Free Software Foundation)이 설립되었다. 자유라는 의미는 비용을 지불하지 않는다는 의미보다는 소스코드에 대한 변경등의 접근권한이 최초 제작자와 동일하게 누구에게나 주어진다는 의미가 더 강하다.

오늘날 지적재산권을 상징하는 의미의 '카피 라이트(Copy Right)'에 반대적인 입장인 '카피 레프트(Copy Left)'운동의 핵심이면서 오픈소스 라이센스의 대표격인 일반 공중사용허가서(GPL, General Public Licese)는 1989년 발표되었다.

GNU 프로젝트는 컴파일러, 편집기, 디버거, 빌드 도구등 프로그래밍과 컴퓨터 운영에 필요한 다양한 도구를 개발하면서 진행되다가 실질적인 결실은 리누스 토발즈의 리눅스 커널과의 결합에 의해 완성되었다.

1997년 에릭 레이몬드는 '성당과 시장 모델'에서 개방형 개발모델의 강점을 주장했다. 성당 모델은 폐쇄형 소프트웨어 개발 모델, 시장모델은 개방형 소프트웨어 개발 모델에 비유했다. 다수의 사람에 의해 버그를 잡아낼 가능성이 높기 때문에 시장 모델이 우수한 개발 모델이라고 주장했다.

1998년 에릭 레이몬드는 자유라는 표현이 갖는 무비용의 인식을 바꾸기 위해 '오픈소스'라는 표현을 사용했다. 이후 넷스케이프는 넷스케이프 네비게이터의 개방형 개발을 스스로 주도하지 않고 흘러가는 대로 내버려뒀다. 이것이 모질라 프로젝트로 이어졌다. 이는 이념적인 접근보다는 실용적인 목적이 있다면 많은 기업이 자유소프트웨어 운동에 동참한다는 것에 의의이다.(?) 이후 '오픈소스'라는 용어가 처음 소프트웨어에 적용된 것은 1998년 2월 3일 미국 캘리포니아 마운틴 뷰의 VA 리눅스 시스템즈 사무실에서 열린 전략회의에서였다. 오픈소스라는 용어 자체는 포어인사이트 인스티튜트의 사장 크리스틴 페터슨이 적용했다. OSI는 1998년 2월말 새로 등장한 오픈소스의 집사 역할을 위해 설립됐다. 에릭레이몬드가 초대사장을 맡았고, 브루스 페렌스는 부사장을, 그리고 초대 이사회에는 브라이언 베렌도르프, 이안 머독, 러스 넬슨, 칩 살젠버그가 참여했다. OSI로고는 콜린 비브록이 만들었는데, O는 개방성을, 열쇠 구멍은 소스코드의 잠금 해제를 암시한다.(OSI의 로고 첨부)

1999년 데비안 프리 소프트웨어 지침(Debian Free Software Guideline)의 작성자인 브루스 페렌스는 이 지침을 소프트웨어 라이센스가 어떻게 오픈소스로 인식될 수 있는지에 대한 객관적인 정의로 개작했다. 바로 OSD(Open Source Definition)이다.

금전적인 대가가 없어도 일을 즐기고, 자기 전문분야 일에 대한 열정이 있으며, 새롭게 혁신되고 확산되며 나아지는 사회적 생산물의 결과에 스스로 만족하기 때문에 개발자들이 오픈소스 커뮤니티에 활발하게 참여한다. 끊임업이 의사를 나누고 서로 협력하는 사람들과 그들의 행동이 내용물의 수준과 질을 신뢰성 있게 만든다.

2000년대 중반, 오픈소스에 대한 인기가 높아지면서 라이센스의 무분별한 확산이 문제가 됐다. 2004~2006년 기간에 OSI는 이 문제를 처리하기 위해 공공 의견 수렴과정을 진행했다.

오픈소스 프로젝트는 2011년 이후 급격히 증가하고 있으며, 오픈소스 프로젝트 수가 100만개를 넘었고, 특히 모바일, 클라우드, 빅데이터, 사물 인터넷, 인공지능 영역에서 급증하며 산업계 전반에 적용되어 혁신을 이끌고있다. 이런 흐름에 따라 IBM은 AI의 블랙박스화를 막겠다며 오픈소스로 알고리즘 공개를 선언했다. 하지만 1999년, 에릭 레이먼드는 전체 소프트웨어의 95% 이상이 상업적 목적이 아닌 실용적 목적으로 개발되고 있으므로 오픈소스가 되어야 한다고 주장하지만 아직까지도 거의 모든 코드가 오픈되지 않고 있다. 레드 햇(Red Hat) CEO 짐 화이트허스트는 2008년 레드 햇 서밋에서 기업 소프트웨어의 낭비를 성토하며 다음과 같이 말했다.

“오늘날 소프트웨어의 절대 다수는 기업에서 개발되며 판매 용도가 아니다. 그리고 이렇게 개발된 소프트웨어조차도 실제로 사용되지 않는 경우가 태반이다. 막대한 자원이 IT 소프트웨어 개발에 낭비되고 있다. ... 결국 오픈소스가 전 세계 모든 고객에게 가치를 전달하기 위해서는 고객이 오픈소스 제품을 사용하면서 오픈소스 및 개발 커뮤니티의 일환이 되어 적극적으로 참여토록 해야 한다.”

물론 아주 발전이 없었던 것은 아니다. 유럽연합 집행위원회 플로스메트릭스(European Commission Flossmetrics)의 2009년 연구 결과에 따르면 (상업적 목적이든 아니든 간에) 전체 코드의 35%가 오픈소스였다. 이는 다른 집계에 비해 아주 후한 수준이다. 클라우데라 공동 창립자 마이크 올슨은 2013년 오픈소스가 기업 인프라를 지배하게 되었다고 말했다. 기업 인프라에 거스를 수 없는 트렌드가 형성되고 있다. 데이터센터를 운영하는 기업은 열이면 열 오픈소스 운영 체제와 데이터베이스, 그리고 미들웨어를 사용한다. 지난 10년간 폐쇄적이고 사유적 형태의 플랫폼 수준 소프트웨어 인프라가 등장하여 지배적 위치를 차지한 적은 단 한 번도 없다.”

올슨의 말은 옳았다. 기업 인프라스트럭쳐 혁신의 상당 부분이 오픈소스 라이선스에 의해 지배된다. 아직까지 포화 상태는 아니지만, 컨테이너 혁명의 원동력이라 할 수 있는 것도 오픈소스인 도커(Docker)와 쿠버네티스(Kubernetes)이다. 빅 데이터는 어떠한가? 하둡(Hadoop), 카프카(Kafka), 그리고 그 외에도 다양한 오픈소스 테크놀로지가 존재한다. 또 상대적으로 신기술인 기계 학습이나 AI도 마찬가지다. 텐서플로우(TensorFlow), MXNet과 같은 오픈소스는 시장을 주도하고 있다. 즉 일반적으로 사용되는 애플리케이션은 폐쇄적이고 사유성을 띨지라도 플랫폼만큼은 점차 오픈소스화되어 가고 있다고 할 수 있다. 그렇다면 대부분 코드가 사적 라이선싱에 묶여 있는 상황에서 미래가 오픈소스에 달려 있다고 말할 수 있을까? 그 둘은 양립 가능한 개념인가?

오픈소스화되는 코드가 늘어나면 더 빠르게 수가 증가하지 않을까? ARM의 존 마크 워커는 필자에게 “최근의 메이저급 혁신은 거의 모두가 오픈소스 플랫폼에서 일어나고 있다. 아직도 많은 이가 새로운 시도를 한다”고 설명했다.

**-오픈소스의 장점**

1. 융통성: 라이센스 비용이나 예산에 제한을 받지 않고, 다양한 오픈 소스들을 테스트한 뒤 최선의 것을 선택할 수 있음.

2. 기술 지원: 신속한 문제 해결, 빨라진 성능 개선 프로세스, 기술ㅡ의 공동 습득이 가능함

3. 기술혁신: 유료일 경우 사용치 않았을 기술의 실험 적용이 가능함

4. 재활용: 소스 코드 접근이 가능함으로 재활용이 증가함

5. 품질: 이미 검증된 소스를 사용함에 따라 개발이 빨라지고 유연해짐

6. 표준: 표준에 충실하고 상호 유연성이 뛰어남

**-오픈소스의 단점**

1. 기능의 불확실성: 무엇이 OSS를 구성하고 있는지와 상대적인 장단점이 무엇인지가 불명확하다.

2. 기술지원의 불확실성: 제품의 지원이 분산되거나 지원을 못받는 것에 대한 불안감이 존재한다.

3. 탐색의 어려움: 특정한 비지니스 문제들에 대한 적절한 OSS 응용제품을 확인하는데 대한 어려움이 존재한다.

4. 문서화 미비: 문서화가 특정 ㄱ인의 취향에 따라 편협하게 이루어졌거나 아예 없는 경우도 존재한다.

5. 로드맵 미비: 상용 프로그램에서 볼 수 있는 로드맵을 기대하기 힘들다

6. 하드웨어 지원: 독점 소프트웨어보다 새로운 하드웨어를 지원하는 면에서 늦음.

**-오픈소스의 미래**

비록 지난 20년 간 오픈소스 코드를 ‘사용’하는 기업들은 늘었지만, 결국 오픈소스의 성공 여부를 판가름 하는 것은 과연 그것이 소프트웨어에서의 혁신 방식을 바꿔놓을 수 있는지, 어떻게 바꿔 놓을 것인지에 달려 있다는 사실을 잊지 말자. 우리는 이제서야 조금씩 최고의 소프트웨어는 오픈소스 소프트웨어라는 사실을 깨달아가고 있는 중이다.

물론, 모든 소프트웨어에 오픈소스가 적합한 것은 아니다. 아파치 소프트웨어 재단 디렉터이자 어도비 수석 과학자 베르트랑 델라크레테즈의 말처럼, “오픈소스는 인프라 소프트웨어에 최적”이다. 반대로 애플리케이션 소프트웨어에는 오픈소스가 크게 확산되기 어려운데, 왜냐하면 “(소프트웨어 스택을 헤치고 올라 갈수록) 합의가 어려워지기 때문”이다. 또한, 특정 소프트웨어에 관심과 적합한 역량을 지닌 개발자 수 역시 스택 레이어가 올라 갈수록 줄어들 것이다.

그러나 기반형 소프트웨어(foundational software)의 경우 오픈소스가 혁신을 주도해 나가는 경향이 있다. 기업이 오픈소스에 맞춰 “소프트웨어의 패러다임을 바꿀 정도”라고 워커는 말한다. 그러나 앞으로의 20년 동안은 점차 이러한 경향이 사그라들고 오픈소스 커뮤니티 활동 및 참여가 더욱 활발해질 것으로 예상된다.

**- 오픈소스의 법적 측면**

오픈소스는 다른 사람들이 작업을 사용, 수정 및 공유하기를 기대하기 때문에 다른 일반적인 경우들과는 다르다. 그러나 법적 기본값은 독점적인 저작권이므로 명시적으로 이러한 사용 권한을 명시한 사용권이 필요하다.

오픈소스 라이선스를 신청하지 않으면, 프로젝트에 기여한 모든 사람도 자신의 저작물의 독점적인 저작권자가 된다. 즉, 아무도 자신의 기여를 사용, 복사, 배포 또는 수정할 수 없으며 “아무도”에서는 귀하를 포함하지 않는다는 의미이다.

프로젝트는 사용자가 알지 못하는 라이선스 요구 사항과의 종속성을 가질 수 있다. 프로젝트의 커뮤니티 또는 고용주의 정책에 따라 프로젝트에서 특정 오픈소스 라이선스를 사용해야 할 수도 있다.

Git hub에서 프로젝트를 공개하는 것은 프로젝트 라이센싱과 동일하지 않다. 공개 프로젝트는 GitHub의 서비스 약관에 명시되어 있으며, 다른 사람들이 프로젝트를 포크화할 수는 있지만, 그렇지 않은 경우에는 권한이 없다.

다른 사람들이 프로젝트를 사용, 복사, 수정 또는 다시 사용할 수 있게하려면, 오픈소스 라이선스를 포함해야한다. 예를 들어, 권한을 부여하지 않는다는 조건에서는 공개적으로 GitHub 프로젝트의 일부를 코드에 명시적으로 사용할 수는 없다.

**- 2017년 분야별 최고의 오픈소스 소프트웨어**

오픈소스는 외로운 개발자와 밤낮을 가리지 않는 작업, 한 줄 한 줄 개인의 비전을 실현하기 위한 코딩 등의 이미지를 떠오르게 하는 용어로 많이 사용됐다. 하지만 해가 갈수록, 그리고 InfoWorld가 최고의 오픈소스 소프트웨어 대상을 진행하고있다.

오픈소스는 여전히 개인의 비전을 실현하는 공간이다. 올해 최고의 오픈소스 소프트웨어로 꼽힌 프로젝트는 많은 수가 개인 개발자의 열정의 산물이라는 이야기의 반증이다.

예년과 다르게, 2017년에 선정된 최고의 오픈소스는 분산 시스템 기술을 완성하고 머신러닝으로 가는 길을 열고 더 나은 애플리케이션 구축 방법을 발명하는 거대 클라우드 업체들이 만든 것이다.

오픈소스 지형도의 다른 영역에서는 귀중한 개발 자원이 다른 방식으로 합쳐지는데, 바로 상용 소프트웨어 업체와 이익을 공유하는 방식이다. 아파치 소프트웨어 재단에서 진행하는 일군의 빅데이터 프로젝트가 대표적인 예로, 이들 프로젝트는 지난 몇 년 동안 계속 최고의 오픈소스 소프트웨어로 선정되고 있다. 이는 클라우드 파생 프로젝트들과는 다른 모델이지만, 결과는 동일하다. 최고의 엔지니어 팀이 첨단 소프트웨어를 만들고, 그 결과물은 우리 모두가 무료로 사용한다.

오픈소스 소프트웨어의 본질과 기원은 계속 진화할 것이라는 데는 의심의 여지가 없다. 최고의 프로젝트가 독립 개발자에게서 나오든 오픈소스 업계의 복합적인 조직에서 나오든 Infoworld는 놓치지 않고 발굴해 낼 것이다. 2017년 최고의 분야별 오픈소스 소프트웨어를 살펴보자.

2017년 최고의 오픈소스 소프트웨어 : 소프트웨어 개발 툴

2017년 최고의 오픈소스 소프트웨어 : 머신러닝 툴

2017년 최고의 오픈소스 소프트웨어 : 클라우드 컴퓨팅 소프트웨어

2017년 최고의 오픈소스 소프트웨어 : DB 및 분석 툴

2017년 최고의 오픈소스 소프트웨어 : 네트워킹 및 보안 소프트웨어