* **오픈소스 소프트웨어 정의와 의의**
  + 소프트웨어의 소스코드가 공개되어, 누구나 소스코드를 자유롭게 사용하고 수정하여 재배포할 수 있는 소프트웨어이다. OSS, FOSS, 오픈소스, 공개 소프트웨어라는 용어가 사용되나 통상 ‘오픈소스’라고 말한다. 오픈소스의 대표적인 예로는 리눅스가 있다.
  + 오픈소스는 자유 소프트웨어와 비슷하지만, 자유소프트웨어는 자유 소프트웨어 재단과 GNU 프로젝트와 관련된 소프트웨어에서 자유를 중시하는 의미에서 사용하고, 오픈소스 소프트웨어는 소스의 형태 자체를 중시한다.
  + Open Source Initiative(OSI)가 인증하는 공개 소스 소프트웨어 인증 마크를 통해 소프트웨어가 실제로 공개 소스라는 것을 증명하고, 공개 소스의 복제도 가능하게 한다.
  + 2008년, Standish Group에 의한 리포트에 따르면 오픈 소스 소프트웨어의 채택은 소비자에게 연간 600억 달러의 절약을 가능하게 한다.
* **오픈소스의 역사와 현재, 미래**
  + 1950년대

이 시기에 만들어진 소프트웨어는 연구나 학술적인 목적으로 만들었기 때문에 대부분 퍼블릭 도메인이었다. 또한 하드웨어 간의 호환성이 없었기 때문에 사용자 간에 소프트웨어를 자유롭게 배포하여 마음대로 수정하는 관습은 흔히 있는 일이었다.

* + 1960년대

컴퓨터 프로그래머와 개발자는 컴퓨터 분야를 발전시키기 위해 소프트웨어를 공유했으며 소프트웨어 공유문화가 퍼져 있었다.

* + 1970-1980년

소프트웨어 개념이 상용화 측면으로 옮겨 갔다. 그러나, 학자들은 여전히 소프트웨어를 공동으로 개발하였다. 1980년대 HP/UX, AIX, Solaris 등의 상용 버전 Unix가 개발되었고 상용화 되었다. UNIX 운영체제 소스 코드의 공유와 이에 기반을 둔 다양한 수정본의 개발은 오픈소스 운동에 지대한 영향을 미쳤다.

* + 1976년

빌게이츠가 컴퓨터 애호가들에게 공개 편지 ‘Open Letter to Hobbyists’를 보냈다. 이 편지는 소프트웨어 저작권에 대해 인지를 하는 계기가 되었고, 상업용 소프트웨어 시대가 시작되었다.

* + 1980년대 초

AT&T가 유닉스와 관련된 지적재산권을 독점하면서 본격적인 소스 코드 독점에 대한 문제가 부각 되었다.

* + 1983-1986년

리처드 스톨만이 자유 소프트웨어 운동을 주도하였다. 상용 소프트웨어에 대항한 자유로운 대안을 만들기 위한 의지를 표출하며, GNU is Not a Unix(GNU) 프로젝트를 시작하였다.

* + 1984년

자유 소프트웨어 운동 시작 후 지적 재산권이 기업의 독점적 지위를 높이고 시장을 독차지하며 기술혁신을 가로막을 것이라는 우려에 자유 소프트웨어 재단이 설립되었다.

* + 1989년

오픈소스 라이선스의 대표격인 General Public License가 발표되었다. 이것은 오늘날 지적 재산권을 상징하는 의미이 Copy Right에 반대적인 입장인 Copy Left 운동의 핵심이다.

* + 1991년

리눅스 토발즈가 리눅스를 만들었다.

* + 1997년

Eric Raymond가 자유 소프트웨어 철학을 대변하는 The Cathedral and the Bazzar을 발행하였다. 이 소논문은 폐쇄형 소프트웨어 개발 모델을 성당에, 개방형 소프트웨어 개발 모델을 시장에 비유하여 개방향 소프트웨어 개발 모델이 더 우수한 개발 모델이라고 주장하였다. 후에 1998년 초, 엄청난 주목을 받으며 Netscape Communicator 인터넷 제품군을 free software로 출시하도록 동기를 부여하는 한가지 요소가 되었다.

* + 1998년

Eric Raymond가 자유라는 표현이 갖는 무비용의 인식을 바꾸기 위해 오픈소스라는 표현을 사용하였다.

* + 1998년 2월 3일

미국 캘리포니아 마운틴 뷰의 VA 리눅스 시스템즈 사무실에서 열린 전략회의에서 오픈소스라는 용어가 처음 사용되었다.

* + 1998년 2월 말

Open Source Initiative, OSI가 새로운 용어(open source)의 사용을 장려하고 오픈소스 원칙을 전도하는 목적으로 설립되었다. OSI는 폐쇄형 소프트웨어 개발 이력과 인터넷 개발자 커뮤니티가 이미 제공한 오픈 개발 사례에서 최소 20년의 증거를 확보하여 넷스케이프 같은 상업적 사업에 오픈소스 사례를 제시하였다.

* + 1999년

에릭 레이먼드가 전체 소프트웨어 95% 이상이 상업적 목적이 아닌 실용적 목적으로 개발되고 있으므로 오픈소스가 되어야 한다고 주장하였다.

* + 2000년대 중반

오픈소스에 대한 인기가 높아지면서 라이선스의 무분별한 확산이 문제가 되었다. 2004-2006년 OSI는 이 문제를 해결하기 위해 공공의견 수렴 과정을 진행하였다.

* + 2009년

유럽연합 집행 위원회 Flossmetrics의 연구 결과에 따르면 (상업적 목적이든 아니든 간에) 전체 코드의 35%가 오픈소스였다.

* + 현재

현재 오픈소스 시장의 대부분은 2005년에 리눅스 토발즈가 만든 git이 차지하고 있다. git에는 8500만개 가량의 프로젝트와 2800만명 이상의 개발자를 가진 거대한 오픈소스 코드 시장이다. sw개발자에게 자신의 깃허브 계정은 커리어 그 자체가 될 정도로 깃 허브는 sw개발시장에서 거대한 부분을 차지하고 있다.

github가 오픈소스 시장을 차지하기 전 오픈소스 시장을 장악했던 소스포지가 있었다. 그러나 소스포지는 서비스 형태를 재구축하면서 개발자들에게 금전적 부담을 지웠고, 이로 인해 개발자들이 github로 대거 이주하게 되면서 소스포지는 하락세를, github는 상승세를 타게 되었다. 소스포지 뿐만 아니라 ms에서 운영하던 코드 플렉스 역시 git허브에 밀려 폐쇄하게 되었다.

깃허브의 오픈소스 시장장악은 it모바일 시장의 성장과 함께 했다는 점에 있다. 또한 더욱 다양한 1인, 소규모 개발자가 sw를 개발하게 되면서 깃허브의 영향력 역시 커지게 되었다.

2017년 깃허브는 자사의 오픈소스 커뮤니티인 코드플렉스를 폐쇄한 ms에게 700억달러에 인수되었다. 그러나 개발자들은 깃허브가 거대 it기업인 ms에 종속되었다는 그 자체를 불안해하며 깃랩등으로 이전하기 시작했다.

아직까지 오픈소스 시장은github가 장악하고 있지만, 과거 오픈소스 시장을 독점하다 몰락한 소스포지 처럼 언제 든지 github도 무너질 가능성이 있다. ms가 깃허브를 인수한 700억 달러를 어떤 방식으로 회수하느냐에 따라 오픈소스 시장의 흐름이 바뀌게 될 것이다.

* + 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 장점**
  + 융통성

라이선스 비용이나 예산에 제한을 받지 않고, 다양한 오픈 소스들을 테스트 한 뒤 최선의 것을 선택할 수 있다.

* + 기술 지원

신속한 문제 해결, 빨라진 성능 개선 프로세스, 기술의 공동 습득이 가능핟.

* + 재활용

소스 코드 접근이 가능하기 때문에 재활용이 증가한다.

* + 품질

이미 검증된 소스를 사용함에 따라 개발이 빨라지고 유연해진다.

* + 이용에 비용이 들지 않거나 적다

기업들은 필요한 인력과 시간을 절약하여 시장적시성과 비용 절감을 위해 검증된 오픈소스 소프트웨어를 활용하게 된다.

* + 다양성

오픈소스는 개인의 필요나 지적유희의 목적으로 개발되는 경우가 상당부분이기에 전 세계에 흩어져 있는 여러 개발자들에 의해 개발되는 오픈소스 소프트웨어의 다양성은 상용소프트웨어를 능가한다.

* **오픈소스의 단점**
  + 기능의 불확실성

무엇이 OSS를 구성하고 있는지, 상대적인 장단점이 무엇인지 불명확하다.

* + 기술지원의 불확실성

제품의 지원이 분산되거나 못 받는 것에 대한 불안감이 존재한다.

* + 탐색의 어려움

특정한 비즈니스 문제들에 대한 적절한 OSS 응용제품을 확인하는 것에 대한 어려움이 존재한다.

* + 로드맵 미비

상용프로그램에서 볼 수 있는 로드맵을 기대하기 힘들다.

* + 하드웨어 지원

독점 소프트웨어보다 새로운 하드웨어를 지원하는 것이 늦다. (호환성 문제)

* **오픈소스의 수용정책의 확산과 해외정책 동향**
  + 독일의 오픈소스 소프트웨어 정책

독일 연방의회는 연방정부 행정기관이 오픈소스 소프트웨어의 사용과 기반을 조성하도록 2001년에 권고하였다. 또한 독일 연방경제기술부와 독일연방정보기술보안청에서는 BerliOS프로젝트, SPHINX프로젝트, GNUPG프로젝트등 오픈소스 소프트웨어 사용 활성화를 위한 노력을 해왔다.

특히 독일연방내무부 산하에 전자정부관련 업무의 조정 및 자문역할을 수행하는 기관인 KBSt를 설치하여 공공부분에서 오픈소스 소프트웨어 활용을 지원해주도록 하고있다. 2002년에는 독일연방내무부는 IBM과 협약을 맺고 모든 공공기관이 IBM을 통해 오픈소스 소프트웨어를 저렴한 가격으로 조달할 수 있도록 했다.

뮌헨시의 경우 2003년 약 1만 4천대의 컴퓨터 소프트웨어를 리눅스로 전환하였으며, 시의회의 경우 2009년까지 클라이언트 컴퓨터에 오픈소스 소프트웨어를 사용하기로 하는 등 오픈소스 소프트웨어 정책을 적극적으로 추진하였다.

* + 프랑스의 오픈소스 소프트웨어 정책

프랑스는 오픈소스 소프트웨어 도입에 적극 추진했던 나라이다. 2002년 상원은 모든 공공행정기관에 오픈소스 소프트웨어 사용을 의무화하고, 자유소프트웨어기구를 설립하는 발안을 발의하였으나 입법화 시키지는 못했다. 그 대신 오픈소스 소프트웨어 활용 가이드 등을 통해 활성화를 유도해 오고있다.

중앙 정부수준에서는 내무부, 관세청, 재정경제산업부, 프랑스 국립경찰청등에서 오픈소스 소프트웨어인 오픈오피스(OpenOffice)를 도입하였으며, 2004년 수립된 전자정부 계획의 일환으로 정부기관의 데스크톱에 오픈소스 소프트웨어를 설치하기로 하였다. 프랑스는 ATICA(정보 및 커뮤니케이션 기술관리 기구), NIRIA(국립 컴퓨터 사이언스와 컨트롤 연구기관)등 다양한 기구를 통해 오픈소스 소프트웨어 정책을 적극적으로 추진해오고 있다는 것이 특징이다.

지방 정부수준에서의 오픈소스 소프트웨어 추진상황을 보면 아를의 경우 2005년 시정부의 모든 지적재산권 응용서버들을 FOSS로 전환하는 작업을 진행하였고, 파리는 같은 해 395개 서버에 리눅스 운영체제를 도입하였다.

* + 미국의 오픈소스 소프트웨어 정책

미국의 경우 연방정부 차원보다는 주정부 차원에서의 오픈소스 소프트웨어 활용정책이 활발히 추진되고 있다. 그러나 2003년 미 국방부의 오픈소스 사용에 대한 규정 제정과 OMB의 소프트웨어 구입에 있어 오픈소스 소프트웨어를 상용소프트웨어와 차별하지 않는다는 조달 중립적 정책등으로 오픈소스 소프트웨어 도입을 활성화하고 있으며, 세계 오픈소스 소프트웨어 시장을 주도해 가고있다.

주정부 차원에는 오픈소스 소프트웨어 구매와 관련하여 법제화를 추진했던 사례가 다수 있는데, 2003년 오리건 주에서는 새로운 모든 소프트웨어를 획득할 경우 오픈소스 소프트웨어의 사용을 고려할 것을 요구하는 법안이 제출되었고, 캘리포니아 주정부도 공공기관의 오픈소스 소프트웨어 사용의무를 지우는 ‘Digital Software Security Act’를 상정하였다.

* + 남아프리카 공화국의 오픈소스 소프트웨어 정책

2000년대 초 남아프리카공화국의 GITOC 정부가 추진하는 전자정부 정책에서 공개 소프트웨어의 역할이 명확하게 인정되어야 한다는 결론을 내리고 이를 정부에 건의함. 이는 전자정부 등 정부의 IT기술도입에 대한 비용이 증가하고 있는 상황에서 이미 제공되고 있는 공개소프트웨어가 제공할 수 있는 편익을 최대한 이용할 수 있도록 하기 위함.

남아프리카 공화국 정부의 OSS 정책은 크게 3가지이다. 첫째, 가치면에서 공개소프트웨어의 구현은 궁극적으로 가치를 창출하여야 한다. 경제적인 면에서는 외화 절감, 조달비용 절감이 있고 사회적인 면에서는 정보에 대한 접근권, IT 교육에 대한 수단 제공이 있다. 둘째, 공개소프트웨어를 구현하고 유지하기위한 적절한 능력이 필요하다. 공개소프트웨어를 구현하고 지원하기 위한 능력을 확대하기 위한 action steps와 정보의 제공, 전문가 조언, 개발지원 등을 정책으로 내세웠다. 셋째, 모든 주요 참여자들이 정책에 대해 충분한 지원을 할 것이다. 세부 정책으로는 컨퍼런스, 워크샵 등을 통해 공개소프트웨어 관련 정보의 확산, 공개소프트웨어와 관련된 토론을 촉진하고 정보를 확산하기 위한 웹 사이트 구축 등이 있다.

* + 대한민국의 오픈소스 소프트웨어 정책

우리나라는 2003년부터 정부주도로 오픈소스 소프트웨어 활성화가 본격적으로 시작되었다. 그 이전 1990년대 후반에는 세계적으로 웹서버 등의 용도로 빠르게 확산되는 리눅스를 국내 기술로 전환하는데 집중하는 리눅스 육성정책이 주를 이루었다. 그러나 2002년 소프트웨어 산업 육성 중장기 계획에서 리눅스 육성정책이 제외되고 2003년 시행계획에서 ‘공개소프트웨어 활성화 추진’ 정책이 포함되어 오늘에 이르고 있다.

한편, 정부의 ‘공개 소프트웨어 활성화’ 정책과 관련되는 기관으로 2004년 설립된 ‘공개소프트웨어 기술지원센터’, ‘공개소프트웨어 기술지원단’이 있다. 2007년에는 직접적인 기술지원업무는 민간시장 영역이라는 판단하에 센터는 역할을 종료한다. 그러나 공공영역에서 지원해야 할 ‘ 공개소프트웨어 기업 및 솔루션 현황, 정보제공 역할’등은 필요했으므로 2009년에 ‘공개 소프트웨어 역량프라자’를 개소하여 현재에 이르고 있다.

정부의 오픈소스 소프트웨어 활성화 정책은 조달부문의 제도개선으로 이어져 ‘공개 소프트웨어 관련기업의 공공부문 사업 참여 기회 확대, 입찰과정에서 기회 봉쇄 장벽제거’ 등이 이루어졌다. 특히 업무 시스템 고도화, 사이버 인프라 구축, 보건 의료 정보화 사업등 오픈소스 소프트웨어 도입효과가 큰 사업을 통해 상용소프트웨어 보다 약 50%의 비용절감 효과를 얻을 수 있다는 결과를 얻었다.

또 오픈소스 소프트웨어에 대한 저작권 분쟁 등에 대비하기 오픈소스 소프트웨어 라이선스에 대한 분석지원 사업도 추진하였는데, 오픈소스 소프트웨어 라이선스에 대한 올바른 사용이 이루어지도록 하는 기반을 조성해오고있다..

* 오픈소스의 기본이념
  + 소프트웨어의 소스 코드를 자유롭게 읽고, 재배포 및 개조를 가능하게 함으로써 소프트웨어가 향상되고, 한 사람이 느린 속도로 소프트웨어를 개발하는 것보다 여러 사람들이 고치고 쓰고 버그를 개선하는 것이 보다 빠를 수 있다.
  + 이미 있는 오픈소스를 fork하여 새 프로젝트가 만들어지는 경우, 이를 공개해야한다.
  + 소프트웨어를 더 좋게 만들기 위해, 외부 개발자들이 참여하도록 하면 더 다양한 환경에서 소프트웨어를 시험해보고 품질을 높일 수 있다.
* **오픈소스 관련 재단**
  + 자유 소프트웨어 재단

자유 소프트웨어의 생산과 보급을 장려하기위해 리처드 스톨만이 세운 재단으로 주로 컴퓨터 소프트웨어를 배포하고 수정하는 보편적인 자유를 재고한다. 1990년대 중반 이후로 이 재단의 직원들과 자발적인 기여자들은 대개 자유 소프트웨어 운동과 자유 소프트웨어 커뮤니티를 위한 법적, 구조적 문제에 대한 작업을 처리하고 있다.

* + 아파치 소프트웨어 재단

1999년 6월, 아파치 소프트웨어 재단은 아파치 그룹으로 구성되어 미국 델라웨어에서 설립되었다. 그들이 개발하고 있는 소프트웨어는 아파치 라이선스 조항 아래 배포되고 자유 소프트웨어/오픈소스 소프트웨어이어야 한다. 아파치 프로젝트는 공동제작과 합의에 기반한 개발 프로세스와 오픈되고 실용적인 소프트웨어 라이선스라는 특징으로 규정된다.

* + 모질라

1998년 넷스케이프 출신 개발자들이 창립한 자유 소프트웨어 커뮤니티이다. 파이어 폭스 웹 브라우저, 썬더버드 이메일, 버그질라 오류 추적기 등을 개발했다. 2003년 모질라 커뮤니티가 중심이 되어 비영리 재단인 모질라 재단을 설립했다.

* + 레드햇

1993년에 창립된 리눅스 배포판 제작회사로, 레드햇 리눅스를 제작했다. 레드햇은 오픈 소스를 비즈니스로 전개하면서 소프트웨어를 무료로 공급하고 지원을 유료로 하는 비즈니스 모델을 기본으로 하였지만, 최근에는 고도화된 소프트웨어를 유료로 판매하는 비즈니스도 시작하고 있다.

* + OSI

오픈소스 소프트웨어 사용을 장려하기 위해 만들어진 단체, 1998년 2월에 브루스 페렌스와 에릭 레이먼드가 넷스케이프 커뮤니케이터에 대한 소스코드를 공개한 것에 대해 고무되어 설립하였다.

* **오픈소스 소프트웨어의 조건**

오픈소스 소프트웨어와 다른 소프트웨어의 차이는 라이선스의 차이에서 비롯된다. 오픈소스 소프트웨어가 되기 위해서는 소프트웨어의 배포 조건이 OSI에서 정한 다음의 9가지 기준을 만족하여야 한다.

* + 자유로운 배포와 재배포(Free Redistribuition)

특정한 소프트웨어의 라이선스에는 해당 소프트웨어의 일부나 전부가 다수의 프로그램으로 구성되는 배포판의 일부로 포함되어 재배포되지 못하도록 배포나 판매상의 제한을 설정 할 수 없다. 또한 이러한 종류의 배포판에 대한 판매나 양도에 있어서 별도의 라이선스 비용을 징수 할 수 없다.

* + 소스 코드의 공개

프로그램 저작물에는 반드시 소스코드가 포함되어야 하며, 컴파일된 형태 뿐만 아니라 소스코드의 배포 또한 허용되어야 한다. 만약, 소스 코드를 제외한 상태로 배포하고자 한다면 일반적으로 통용되는 매체를 이용해서 제작 실비에 준하는 비용으로 소스코드를 제공해야만 한다. 소스 코드는 프로그래머들이 개작하기에 용이한 형태로 제공되어야 하며 고의로 복잡하고 혼란스럽게 만들어진 형태와 선행처리기나 번역기에 의해서 생성된 중간 형태의 코드는 허용되지 않는다.

* + 2차적 저작물

라이선스에는 프로그램 원저작물의 개작이나 이를 이용한 2차적 프로그램의 창작이 허용되어야 하며, 이러한 파생적 프로그램들은 최초의 프로그램이 갖고 있던 라이선스의 규정과 동일한 조건하에서 재배포될 수 있어야 한다.

* + 소스코드 수정의 제한(Integrity of Author’s Source Code)

빌드 과정을 통해서 프로그램을 개작할 목적으로 소스 코드와 패치 파일을 함께 배포할 경우에는, 정상적인 빌드를 보장하기 위해서 라이선스 안에 소스 코드의 수정을 제한하는 항목을 추가할 수 있다. 그러나 이러한 경우에도 수정된 소스 코드를 이용해서 만들어진 소프트웨어에 대한 자유로운 배포를 허용해야 하며, 수정된 소스 코드를 통해서 만들어진 2차적 프로그램을 원래의 프로그램과 구별하기 위해서 별도의 이름과 버전을 사용할 것을 요구하는 항목을 추가할 수 있다.

* + 사람이나 그룹에 대한 차별 금지(No Discrimination Against Persons or Group)

모든 개인이나 단체에 대하여 동일하게 적용되어야 한다.

* + 라이선스의 배포(Distribution of License)

프로그램에 대한 권리는 반복되는 배포에 따른 별도의 라이선스 승인이나 양도 과정 없이도 프로그램을 배포받은 모든 사람에게 동일하게 적용된다.

* + 라이선스 적용 상의 동일성 유지(License Must No Be Specific To a Product)

로그램에 대한 권리는 반복되는 배포 과정에서 특정한 배포판에 포함되어 있는 상태로만 유효하지 않고, 모든 배포 단계에서 동일한 효력을 갖는다. 만약, 특정한 배포판에 포함되어 있던 프로그램을 독립적으로 사용하거나 재배포한다면 해당 프로그램을 배포받은 사람은 프로그램이 포함되어 있던 최초의 배포판 상태에서 발생된 권리와 동일한 권리를 갖는다.

* + 다른 라이선스의 포괄적 수용

라이선스에 오픈 소스 소프트웨어와 함께 배포되는 소프트웨어에 대한 제한을 설정해서는 안된다.