

경제학이 오픈소스 소프트웨어에 대해 알고 있는 것: 역사, 산업 조직, 정책 함의

이 남 형* · 김 준 일**

논문 초록

오픈소스 소프트웨어는 자발적인 개발자들이 공개된 소스코드를 이용하여 개발하고, 라이선스와 함께 자유롭게 배포하는 소프트웨어이다. 오픈소스 소프트웨어는 지적재산권을 배타적 사용이 아니라 공공의 사용을 위해 활용한다는 점, 생산에 참여한 사람이 직접적인 이득을 볼 수 있는 방법을 제한하였지만 그 가능성은 열어 놓았다는 점, 분산된 형태의 생산을 이끌었다는 점에서 새로운 시대의 새로운 생산방법으로서 주목받을 만하다. 본 논문은 두 가지를 목적으로 한다. 먼저 오픈소스 소프트웨어 개발의 역사가 컴퓨터 발달의 초창기부터 시작하고 있음을 밝히고자 한다. 더불어 Lerner and Tirole(2002)의 선도적인 연구 이후 축적된 경제학 연구를 참여 동기, 개발 과정 및 관리, 시장 경쟁, 라이선스, 정책, 다른 산업에 주는 함의 등으로 정리했다. 이로써 이론 및 정책적으로 중요성이 높아지고 있는 이 분야의 후속 연구에 도움을 주고자 한다.

핵심 주제어: 소프트웨어 산업 역사, 오픈소스 소프트웨어, 공개 소프트웨어

경제학문헌목록 주제분류: L1, L5, L8, N7, O2

투고 일자: 2015. 2. 12. 심사 및 수정 일자: 2015. 10. 31. 게재 확정 일자: 2016. 1. 25.

* 제1저자, 고려대학교 대학원 경제학과 박사과정, e-mail: marginalman@korea.ac.kr

** 교신저자, 고려대학교 경제학과 연구교수, e-mail: kjoonil@korea.ac.kr

I. 서론: 오픈소스 소프트웨어는 왜 중요한가

오픈소스 소프트웨어(이하 OSS: Open Source Software)¹⁾는 자발적인 개발자들이 공개된 소스코드²⁾를 자유롭게 이용하여 개발하고 라이선스와 함께 다시 배포하는 소프트웨어이다. 윈도우즈, 인터넷 익스플로러, 포토샵 등의 사유소프트웨어(이하 PS: Proprietary Software)의 소스코드는 공개되지 않으며 소비자에게 사용 비용을 부과할 수 있다. 무료 소프트웨어(freeware)는 무료로 배포된다는 점에서 OSS와 유사하지만 소스코드가 공개되지 않는다는 점에서 다르다.

소스코드의 공개와 이를 규율하는 라이선스로 인해 OSS는 상업용 소프트웨어와 다른 특징을 갖는다. 요리법을 소유한다고 하여 그 요리를 먹을 수 있는 것은 아니지만, 어떤 소프트웨어 소스코드의 소유는 곧 소프트웨어의 소유와 사용이 가능함을 의미한다. 그렇다고 OSS 사용의 제약이 없는 것은 아니다. OSS의 배포는 라이선스를 따라야만 한다. 라이선스에는 소스코드의 작성자가 사용자에게 배포 비용 이상의 비용이나 로열티 등을 요구하지 못하는 규제뿐만 아니라 소스코드의 수정과 파생 작업(derived works)의 허용, 최초 소프트웨어와 동일한 라이선스 하에서의 재배포의 허용 등의 조건이 포함되어 있어야 한다(Open Source Initiative, 1999).

이러한 기술적 특성과 제도적 규정으로 인해 OSS 소스코드의 작성자는 사용자에게 소프트웨어의 배타적 사용에 대한 비용을 부과할 수 없다. 이로 인해 OSS는 등장한 시점부터 정치적 양극단의 입장에서 비난과 찬사를 동시에 받았다. 무료 배포라는 특징 때문에 공산주의 또는 지적재산권에 대한 심대한 위협으로 비난받거나, 사적 소유와 생산에 반하는 혁명이라고 찬양받기도 했다. 다른 한편, 자유주의자들은 OSS와 함께 광범위한 의미에서 오픈소스로 분류되는 카피레프트와 위키피디아 등을 독점에 대한 대안이자 “자유주의적 아이디어의 힘을 보여주는 훌륭한 사례”로 부르기도 했다(Lee, 2007; Moglen, 2003).

1) 이후에서는 내용에 따라 OSS 또는 F/LOSS로 약칭한다. F/LOSS의 F와 L은 각각 Free와 Libre의 약어이다. 오픈소스 소프트웨어(OSS)와 자유 소프트웨어(F/LOSS)는 많은 공통점을 갖고 있음에도, 두 개발자 커뮤니티의 분명히 구별되는 차이점을 부각시키기 위해 구분되어 사용되기도 한다. 하지만 이 둘의 차이를 보이는 것은 이 글의 범위를 벗어난다.

2) 소스코드는 컴퓨터의 기계어를 인간이 읽을 수 있도록 표현한 것으로 이를 컴파일하면 컴퓨터 등의 기계에서 명령을 실행할 수 있게 전환된다. 따라서 소스코드를 컴퓨터에서 작동시키기 위해서는 이를 기계어로 변환시킬 수 있는 컴파일러가 필요한데, 이것도 공개되어 있다.

마이크로소프트사(Microsoft Inc.)는 OSS가 소프트웨어 산업에서 수용되는 과정에서 가장 극적인 입장 변화를 보였다. 마이크로소프트사의 창업자인 빌 게이츠(Bill Gates)는 자신의 프로그램이 공용으로 사용되는 것에 비판적인 입장을 보이면서, “프로그래밍, 오류 찾기, 결과물의 문서화 등을 무료로 하는 취미가(hobbyists)는 있을 수 없다”고 말했다. 그 뒤를 이은 스티브 발머(Steve Ballmer)는 2001년의 인터뷰에서 OSS는 마치 “지적재산권에 달라붙은 암”과 같다고 했다.³⁾ 하지만 2014년 현재 최고 경영자인 사티아 나델라(Satya Nadella)는 기술 개발 발표회에서 “마이크로소프트사는 리눅스(Linux)를 사랑한다”고 말했다. 실제로도 마이크로소프트사는 서버 메시지 블록 시스템인 삼바(Samba)와 클라우드 시스템인 하둡(Hadoop) 등 최근 주목 받는 OSS 프로젝트의 주요 기여자이고, 자사의 개발자용 프로그램인 닷넷(.Net)을 오픈소스화 했다.

직접적인 이윤 동기를 갖기 어려움에도 불구하고 사용자인 동시에 개발자인 사람들의 자발적인 참여를 토대로 빠르게 발전할 수 있었던 것은 OSS의 실용적인 성격 때문이었다. 이는 F/LOSS의 주창자들이 직접 내세운 자유와 효율성이라는 가치와 맞닿아 있다. 리처드 스톨먼(Richard Stallman)은 소프트웨어의 배포에 너무 많은 권력을 행사한다는 이유로 미국의 소프트웨어 저작권 체계를 반대했으며, 사람들이 자신의 주변 사람을 도울 수 있도록 일상생활에서 사용하는 도구를 바꾸고 개선할 수 있는 행동을 자유롭게 결정할 수 있는 시스템을 만들어야 한다고 했다. 그는 1985년 자유 소프트웨어 재단(Free Software Foundation)과 그누 일반 공중 사용 허가(이하 GPL: GNU General Public License) 체계를 만들고, 자신의 주장을 활발하게 알려나갔다. 하지만 소스코드의 공개가 자유의 문제일 뿐만 아니라 효율성을 높인다는 점도 지적했다. 소프트웨어 생산의 한계비용은 0에 가깝기 때문에 가격이 0에 가까운 것이 정상이고, 소프트웨어 라이선스 비용은 소프트웨어 사용의 부정적 유인이 된다. 따라서 소프트웨어 사용에는 비용을 부과하지 않는 것이 사회적으로 더 바람직하다. 그리고 PS는 소프트웨어의 사용자가 문제를 파악하고 이를 소스 코드의 작성자에게 전달한 후 수정이 이루어지고 그 뒤 다시 배포과정을 거쳐야만 수정된 버전을 사용할 수 있는데 반해, OSS는 사용자가 동시에 개발자이므로 문제를

3) 이러한 발언의 근거는 소프트웨어가 다양한 코드의 집합이기 때문이다. OSS 라이선스의 적용을 받지 않는 코드를 개발하더라도, 소프트웨어의 다른 코드가 OSS 라이선스의 적용을 받는다면 새로 개발한 코드는 동일한 라이선스를 적용하여 공개해야 한다.

파악하자마자 직접 수정하여 바로 사용할 수 있는 장점이 있다고 강조했다 (Stallman, 2010, pp. 46-47).

이와 같은 오픈소스 방식의 효율성을 설명하는 가장 유명한 표현은 “보는 눈이 많으면 프로그램 오류는 줄어든다”이다. 에릭 레이먼드(Eric Raymond)는 PS가 중앙 집중화된 방식으로 소수의 개발자가 프로그램을 만드는 “성당”인데 반해, OSS는 소스코드가 공개되어 많은 수의 사용자가 직접 문제를 해결하고 더 나은 코드를 만들 수 있는 “대시장”이라고 비유했다(Raymond, 1998).

오픈소스 개발 방식의 확산은 최근의 현상처럼 보이지만, 컴퓨터가 등장한 시점부터 소프트웨어 개발자들은 코드에 대한 배타적 소유권을 주장하지 않고 매우 개방적인 태도를 취했다. II절에서는 1950년대부터 1990년대까지 소프트웨어 개발자 집단에 존재한 오픈소스 개발 방식의 역사를 서술했다.

OSS의 짧은 역사와 더불어 본 논문에서는 지난 15년 동안의 OSS에 관한 경제학 연구를 정리하고자 한다. Lerner and Tirole(2002)의 선도적인 분석 이후로 경제학에서는 이 분야에 대한 많은 연구가 나왔다. 상징적인 사건으로서 2005년부터 미국 경제학문헌연보 분류기호 엘17 오픈소스 상품과 시장(JEL Code L17 Open Source Products and Markets)이 등장한다.⁴⁾ 새로운 코드는 오픈소스 소프트웨어를 염두에 두고 만들어졌으나, 이후 유사한 개념의 상품이나 시장에 대한 연구로 그 범위가 확장되었다. 일반적으로 소프트웨어 산업에 대한 분석은 분류기호 엘86(L86)에 포함되는데, 오픈소스에 대한 새로운 분류 코드가 만들어진 것은 이 분야에 연구가 증가하였고 이러한 연구 관심이 인정받았음을 시사한다.

오픈소스에 관한 경제학의 전통적인 연구 대상은 ‘참여 동기, 개발 과정 및 관리, 시장 경쟁’의 3가지이다. 이중 가장 먼저 활발하게 연구된 것은 개발자의 참여 동기였다. 왜 개발자가 직접적이거나 명시적인 보상이 없는데도 자발적으로 코드를 만들고 이를 공개하는지에 관한 많은 연구가 이루어졌다. 개발 과정 및 관리에 관한 연구는 오픈소스 개발 방식에 관한 문제를 다루는데, 특히 전 세계에 흩어져 있는

4) EconLit.org 검색 결과로 엘17 코드의 경제학 문헌이 본격적으로 나타나는 것은 2005년이다. 전미경제학회의 담당자인 찰스 맥캔(Charles McCann)과의 이메일 교류에 따르면, 코드를 도입한 정확한 시점을 기록으로 남기고 있지 않지만, 이 코드를 도입한 사람은 아사토시 마에시로(Asatoshi Maeshiro)로 확인되었다. 그는 36년간 미국경제학문헌연보의 분류 작업을 담당했으며 2005년 12월에 퇴임했다. 이후 분류기호 씨(Code C)에 대한 컨설팅트로 2014년 12월까지 업무를 맡아왔다.

개발자가 수백만 줄의 코드를 효과적으로 생산하도록 어떻게 조정하는가의 문제에 연구 초점이 맞춰졌다. 시장 경쟁을 주제로 한 연구는 PS와의 경쟁 환경에서 OSS가 시장을 장악할 수 있을 것인가 아니면 오히려 개발의 부진으로 시장에서 사라질 것인가를 주로 분석했다(Bonaccorsi and Rossi, 2003; Johnson, 2012).

소스코드가 공개되어 있으므로 OSS는 공공재의 성격을 갖는다. 표준적인 경제학의 설명대로라면 개발이 지속되지 않거나 과소공급이 발생할 것이다. Ostrom의 선구적인 연구 이후 우리는 시장이 아니라 잘 조직된 커뮤니티와 커뮤니티에서 합의된 규칙과 같은 제도가 공공재 또는 공공자원의 문제를 해결할 수 있음을 알고 있다. 그런데 이러한 연구의 주요 분석대상인 숲이나 어족 등의 공유자원과 달리, 소프트웨어는 사용에 따라 소모되지 않는 특징이 있다. 또 소프트웨어 또는 확장된 개념으로서 지식은 공유될수록 그 가치가 커질 수 있다. 그러므로 현재의 공유 자원을 어떻게 고갈시키지 않을 것인가의 문제가 아니라 어떻게 꾸준히 공동의 것으로 유지하며 새로운 내용을 더해갈 것인가의 관점으로 OSS를 접근할 필요가 있다.

현재 OSS 개발에서 공공재의 과소 공급과 무임승차 문제를 해결하고, 개발자의 참여를 유도하는 대표적인 제도는 라이선스라 할 수 있다. 제Ⅲ절에서는 라이선스의 역할과 함께, OSS 개발자의 사회경제적 특징 및 행동 동기, 개발 과정 및 시장 경쟁에 초점을 맞춘 연구를 정리했다.

그리고 제Ⅳ절에서는 OSS 개발과 관련된 정책 사례와 성과에 관한 분석을 소개했다. 마지막으로 결론에서는 소프트웨어를 넘어서 오픈소스 방식이 적용된 사례를 간단히 언급하고, 향후 OSS 연구의 방향을 제기하고자 한다.

Ⅱ. 오픈소스 소프트웨어의 짧은 역사

리눅스나 안드로이드(Android) 운영체제의 성공으로 인해 오픈소스 방식의 개발이 최근에 나타난 것처럼 보인다. 하지만 소스코드를 무료로 공유하는 문화는 컴퓨터가 등장한 시점부터 존재했다.

컴퓨터는 보험, 은행, 유통업, 항공 산업 등에서 1950년대부터 임금 계산, 비용 산출, 주가 계산, 기체 구조 계산 등을 위해 도입되었다. 컴퓨터 제조사는 고객에 특화된 소프트웨어를 하드웨어와 함께 판매하고, 소프트웨어에 대한 별도의 가격은 책정하지 않았다. 이들은 대부분 소프트웨어를 컴퓨터를 판매하기 위한 수단으로

여겼다. 소프트웨어를 만든 다른 집단은 사용자들 자신이었다. 유니박사(UNIVAC Inc.)의 컴퓨터 사용자 집단인 유스(USE)와 아이비엠사(IBM Inc.)의 컴퓨터 사용자 집단인 셰어(SHARE)가 대표적이다(Campbell-Kelly and Aspray, 1996, pp. 192-193).

1952년 아이비엠사는 폰 노이만 구조(von Neumann Architecture)⁵⁾를 최초로 적용한 701 시스템을 선보였다. 아이비엠사는 관련 소프트웨어를 같이 제공했지만, 많은 사용자가 불만을 갖고 있었으며, 이들 스스로 자신의 용도를 위한 소프트웨어를 작성하기 시작했다. 1954년 더 빠른 중앙 처리 장치를 가진 704 시스템 제작 계획이 발표되었다. 704 시스템은 기존의 701 시스템과 호환되지 않았기 때문에 새로운 소프트웨어가 필요했다. 폴 아머(Paul Armer)와 3명의 엔지니어가 개별적인 노력을 뛰어넘어 공통의 문제를 해결하기 위해 새로운 협력체를 만들기로 결정했다. 1955년 8월, 컴퓨터를 보유한 18개 기관의 대표자가 랜드 연구소(Rand Corporation)에서 셰어라는 사용자 그룹을 탄생시켰다. 참여자는 62명으로 빠르게 늘어났고, 약 300여개의 소프트웨어가 작성되었다(Armer, 1980; Mapstone and Bernstein, 1980).

셰어는 불필요한 코드를 쌓지 않으면서 소프트웨어 개발에 협력할 수 있는 토대를 만들었다는 점에서 중요하다. 용어의 정리, 표준적인 코드의 확립, 프로그램의 상호 설치를 위한 하드웨어 통일, 원하는 프로그램의 목록 작성, 자발적인 참여자에게 코딩 업무의 할당, 배포 체계의 확립 등 현재의 OSS 개발 과정에서 이루어지는 일과 거의 동일한 소프트웨어 개발 방식이 이 때 나타났다고 할 수 있다.

하지만 셰어는 반독점에 대한 우려와 컴퓨터 기술의 발달로 힘을 잃게 된다. 1956년 중반 제너럴 일렉트릭사(General Electric Inc.)와 웨스팅하우스사(Westinghouse Inc.)는 셰어가 반독점법을 위반했을 가능성을 제기했다. 셰어는 회원의 자발성을 강조하는 방식으로 대응했다. 하지만 셰어 회원의 대부분이 자신이 속한 회사의 컴퓨터 관련 업무로 임금을 받고 있었으므로 이들의 활동이 완전히 자발적인 것으로 보기는 어려웠다. 다른 대응책은 응용 소프트웨어의 작성 및 배포를

5) 프린스턴 구조(Princeton architecture)라고도 하는 폰 노이만 구조는 중앙 처리 장치(CPU: central processing unit), 기억 장치(Memory), 입출력 장치(I/O: input/output interfaces)의 세 개의 하위 구조를 갖고 있다. 이후에 나온 컴퓨터는 모두 이 구조를 따르고 있으며, 프로그램을 저장할 수 있다는 가장 큰 장점이자 성공 요인을 갖고 있었다.

금지시키는 것이었다. 이들은 경쟁 친화적인 것으로 간주된, 시스템 수준의 소프트웨어 개선에 노력을 집중하기 시작했고, 프로그래밍 언어인 팩트(PACT)와 운영체제인 에스오에스(SOS)를 만드는 데까지 나가게 되었다. 하지만 1957년 아이비엠사가 프로그래밍 언어인 포트란(Fortran)을 공개함에 따라 쉘어의 노력은 쓸모없게 되었다(Akera, 2001, pp.727-728).

1960년대 초부터 진공관 컴퓨터는 트랜지스터 컴퓨터로 대체되기 시작했다. 1961년 여름, 2,700개의 트랜지스터를 사용한 테크사의 피디피-1(PDP-1)이 엠아이티 대학(MIT)에 기증되었다. 이 대학의 모형기차 동호회(TMRC: Tech Model Railroad Club) 회원들은 이 컴퓨터를 마치 장난감처럼 사용해보기 시작했다. 이후 이들은 스스로를 해커(hackers)라고 불렀으며, 정보의 자유가 컴퓨터 사용의 다양성을 열어주고, 자유로운 컴퓨터 활용이 새로운 미래를 가져올 것이라고 확신했다. 이들은 컴퓨터와 관련된 여러 가지 최초의 것을 만들었다. 문서 편집기, 쌍방향 디버거(interactive debugger), 체스 프로그램, 컴퓨터 음악, 컴퓨터 비디오 게임이 여기에 포함된다. 각각의 프로젝트에 참여한 사람들은 다른 사람들이 자신의 프로그램을 사용하는 것을 자랑스러워했고, 사용료를 요구하지 않았다(Levy, 1984, Ch.3)⁶⁾.

1960년 중반부터 나타난 기술적 변화와 반독점 정책은 소프트웨어 개발에 큰 변화를 가져왔다. 1964년 아이비엠사는 에스/360(S/360) 시스템을 공개했다. 컴퓨터 하드웨어를 교체하면 사용자가 소프트웨어를 다시 작성해야만 했던 이전 시스템과 달리, 새로운 시스템은 서로 다른 하드웨어에서 동일한 소프트웨어를 구동할 수 있게 했다. 이로 인해 기업의 컴퓨터 구입 또는 임대 비용이 증가하여 아이비엠사의 판매량이 크게 늘어나게 되었다. 곧 알씨에이사(RCA Inc.)가 에스/360 소프트웨어를

6) 최초의 컴퓨터 게임인 스페이스워(Spacewar)에 관한 흥미로운 일화가 있다. TMRC의 회원인 스티브 러셀(Steve Russell)은 스페이스워를 처음 만들었는데, 로켓 모양의 점(우주선) 두 개를 시계방향 또는 시계반대 방향으로 회전시킬 수 있도록 하고, 무작위로 스크린에 떠있는 점(별)을 피하거나, 서로에게 어뢰를 발사할 수 있도록 했다. 그는 이 프로그램을 책상 서랍에 넣어두고 누구나 개선시킬 수 있도록 했다. 다른 회원인 피터 샘슨(Peter Samson)은 별의 위치가 실제의 항성도를 반영하도록 수정했다. 그는 이 게임을 엘 카시오페아 사냥(Shootout-at-El-Cassiopeia)이라고 불렀다. 댄 에드워즈(Dan Edwards)는 스크린 중앙에 태양을 집어넣고 태양의 중력을 반영하여, 태양에 너무 가까이 다가간 우주선은 불타버리게 했다. 최초의 컴퓨터 게임이 끊임없이 변용될 수 있었던 것은 소스 코드가 공개되었기 때문이었다. 스페이스워는 테크사의 마케팅에 자유롭게 사용되기도 했다.

구동할 수 있는 스펙트라(Spectra 70) 시스템을 공개했다. 아이비엠사는 경쟁사가 소프트웨어의 개발 없이 하드웨어를 판매할 수 있다면 자신들에게 큰 위협이 될 것으로 여겼다. 1964년 여름, 아이비엠사는 경쟁을 막을 수 있는 궁극적인 방법은 자사의 소프트웨어를 묶음 판매에서 제외시키는 것이라는 결론을 내렸다(Humphrey, 2002).

그러나 하드웨어와 소프트웨어의 분리 판매는 기술적 변화보다 반독점법 피소에 기인한다고 봐야한다. 1968년 12월, 씨디씨사(CDC Inc.)가 아이비엠사의 독점 행위로 자사의 성장과 수입이 피해를 입었다며 고소를 제기했다. 아이비엠사의 독점 행위에 대해 미국 법무부 차원의 고소도 예상되고 있었다. 아이비엠사는 미국 법무부에 대한 선제적 대응으로 1968년 12월 6일 하드웨어와 함께 무료로 제공되는 주요 서비스를 판매 묶음에서 제외시키겠다고 발표했다.⁷⁾ 이 발표 이후, 즉각적으로 소프트웨어 산업이 나타난 것은 아니지만, 소프트웨어가 별도의 상품으로서 구매 대상이 되어야 한다는 인식을 사용자에게 심어줌으로써 시장을 형성시킨 중요한 계기가 되었다.

아이비엠사의 에스/360 시스템은 시장에서 큰 성공을 거두어 사실상의 표준으로 자리 잡았는데, 이는 1970년대 소프트웨어의 협력적 개발에 영향을 주었다. 연구소와 대학에서는 여러 종류의 플랫폼에서 구동하는 운영체제를 만들고자 노력을 시작하게 되었고, 이 중 가장 성공적인 운영체제라 할 수 있는 유닉스(UNIX)의 최초 버전이 1969년 등장했다. 동시에 유닉스에서 구동되는 소프트웨어를 작성할 수 있는 씨 언어(C Language)도 공개되었다. 에이티앤티사(AT&T Inc.)의 벨 연구소(Bell Laboratories)에서 작성된 유닉스는 대학이나 정부 기관 등에 무료로 배포되었다.

1980년대 초, 에이티앤티사는 유닉스와 관련된 지적재산권을 이용하여 사용료를 받기 시작했다. 이로 인해 소스 코드를 공유하던 기존의 연구자와 학생들이 위협을 받게 되었고, 이들은 새로운 운영체제 개발을 위한 체계화된 노력을 시작했다. 1984년 엠아이티 대학의 인공지능 연구소(Artificial Intelligence Laboratory) 연구원인 리처드 스톨먼이 유닉스와 유사한 운영체제를 만들자는 그누(GNU: GNU is not a UNIX) 프로젝트를 시작했다. 그는 1985년 자유소프트웨어 재단을 창설하고 자유 소프트웨어는 네 가지의 기본 권리를 담아야 한다고 주장했다. 자유⁰⁸⁾은 어떠한

7) 그럼에도 불구하고, 1969년 1월 17일 미국 법무부 장관 램지 클라크(Ramsey Clark)는 아이비엠사에 대한 반독점 소송을 제기했다.

목적으로도 프로그램을 구동할 수 있는 자유이다. 자유 1은 프로그램이 어떻게 작동하는지 학습하고, 원하는 대로 이를 수정할 수 있는 자유이다. 자유 2는 주변 사람을 도울 수 있도록 복제본을 재배포할 수 있는 자유이다. 자유 3은 수정한 버전의 복제본을 다른 사람에게 배포할 수 있는 자유이다. 더불어 네 가지의 기본 권리를 제한하는 재배포를 금지시키는 카피레프트(copyleft) 개념이 제시되었다(Free Software Foundation, 2010; Stallman, 2010, p. 12).

하지만 자유 소프트웨어 운동은 대중적으로 큰 성공을 거두지 못했다. 소프트웨어 산업의 형성 이후, 이득을 받지 않는 자발적 참여자에 의해서만 소프트웨어가 개발되기는 어려웠다. 자유 소프트웨어의 지지자들은 자신들의 운동이 공짜 맥주가 아니라 발언의 자유를 위한 것⁹⁾이라고 옹호했다. 더 많은 개발자의 참여를 이끌어내기 위해서는 경제적 동기를 보장할 수 있는 방법이 있거나 기술적으로 충분한 장점이 있어야 했는데, 자유 소프트웨어 운동은 이 둘을 충족시키지는 못했다.

1991년 리누스 토르발스(Linus Torvalds)가 유닉스의 새로운 변형 버전 0.02를 공개했다. 그는 GPL 라이선스를 채택했고, 더 많은 개발자들이 참여할 수 있도록 유도했다. 최초로 공개된 버전은 인텔 80386 칩에서만 구동되었지만, 곧 다양한 종류의 플랫폼에서 구동될 수 있게 되었다. 그는 새로운 운영체제를 리눅스로 명명했다.

1990년대 중반부터 월드와이드웹이 활성화되기 시작했고, OSS는 웹을 구성하는 가장 인기 있는 소프트웨어가 되었다. 리눅스 운영체제(Linux), 아파치 웹 서버(Apache), 미에스큐엘 데이터베이스(MySQL), 파이치피 프로그래밍 언어(PHP)가 주로 쓰였고, 머리글자를 따서 통상 램프 시스템(LAMP)으로 불렸다.

1997년 에릭 레이먼드와 팀 오라일리(Tim O'Reilly), 래리 웰(Larry Well) 등이 '자유' 소프트웨어라는 용어가 혼란을 가중시킨다고 보고, 기업의 상업적 활동을 가능하게 하고 소스코드 공유의 장점을 누릴 수 있다는 점을 부각시킬 수 있는 '오픈소스'라는 명칭을 쓸 것을 제안했다. 브루스 페렌스(Bruce Perence)는 '오픈소스 정의(Open Source Definition)' 초안을 작성했다. 오픈소스는 배포 조건을 완화시킨 점에서 자유 소프트웨어와 차이를 보였다. 자유 소프트웨어 운동에서는 카피레프트

8) 컴퓨터 프로그래머의 관습대로 0부터 시작한다.

9) "(Thus), 'free software' is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech,' not as in 'free beer' (Free Software Foundation, 2010)."

원칙에 따라 수정된 내용은 원래의 라이선스와 동일한 조건에서 재배포 되어야 했지만, 오픈소스에서는 수정된 내용은 수정자 자신이 정한 라이선스에 따라 배포할 수 있도록 했다. 이로 인해 수정자가 자신의 수정 사항에 대해 배타적인 라이선스를 적용하고, 그 사용료를 받을 수 있게 되었다. 이는 많은 기업이 OSS 개발에 참여하도록 하는 요인이 되었다.

OSS가 소프트웨어 산업에 자리를 잡게 된 것은 1990년대 후반부터라고 봐야 한다. 그런데, 앞에서 본 것처럼 오픈소스 개발방식은 컴퓨터의 초창기에서부터 연원을 찾을 수 있다. 이 당시의 역사는 다소 일화적으로만 정리되어 있지만, 1980년대 이후 자유 소프트웨어 운동과 OSS의 등장에 문화적 토대를 제공하였으며 무엇보다 협력적인 개발 방식의 경험을 남긴 것으로 평가될 수 있다. 하지만, 1990년대 이후 OSS의 발전은 이전 시대보다 더 복잡한 양상을 띠고 있다. 다음 절에서 최근의 특징과 함께 개발자의 참여 동기를 비롯하여 시장 경쟁과 라이선스의 역할 등, 오픈소스 소프트웨어 개발에 관한 경제학 연구의 성과를 살펴보고자 한다.

Ⅲ. 오픈소스 소프트웨어의 경제학

1. OSS 개발자와 참여 이유

OSS 개발이 성공적으로 이루어지기 위해서는 개발자의 지속적인 참여가 필수적이다. 2014년 10월 현재, OSS 개발의 가장 오래된 포털 서비스 중의 하나인 소스포지(<http://sourceforge.net>)에 등록된 프로젝트는 약 192만개, 등록된 개발자는 약 446만명이고, 최근 널리 사용되고 있는 깃허브(<http://github.com>)에 등록된 프로젝트는 약 3,110만개, 등록된 개발자는 약 1,220만명이다.

〈Table 1〉 The number of registered OSS projects and developers in the web based source code management services

Web based source code management services	Registered Projects (projects)	Registered Developers (persons)
Sourceforge.net*	1, 923, 137	4, 467, 160
Github.com**	31, 100, 000	12, 200, 000

* Sources: SourceForge Research Data Archive: A Repository of FLOSS Research Data, sf1014.

** Sources: <https://github.com/about/press>.

왜 개발자들이 자발적으로 직접적인 보상이 없는 프로젝트에 참여하느냐는 경제학의 큰 의문이었다. 가장 먼저 참여 동기로 제시된 경제학의 설명은 개발자가 직접 자신의 문제를 해결하는 것 자체가 스스로에게 이득이 되는 “가려운 곳을 긁는(scratching an itch)” 일이고, OSS 개발자로서의 명성은 노동 시장에서 신호 효과를 일으켜 자신의 임금을 상승시킨다는 것이었다(Lerner and Tirole, 2002).

하지만 이러한 설명은 개발 결과의 공개라는 OSS의 중요한 특징을 간과하고 있다. 자신의 문제를 해결하는 것은 분명 스스로에게 이득이지만, 그것을 다른 사람에게도 제공하는 것도 이득이 된다는 보장은 없으므로 굳이 자신의 해결책을 공개할 필요가 없다. 신호 효과는 소스 코드 공개를 유도할 수 있지만, 신호 효과가 있기 위해서는 프로젝트가 어느 정도 명성을 얻었을 때 그리고 자신의 기여가 중요한 평가를 받을 때에만 가능하므로, 처음부터 신호 효과를 목표로 공개를 했다는 설명은 과도한 주장일 수 있다. 실제로도 OSS 개발에서 도움말 파일 제작이나 현지 언어로의 전환 등 상대적으로 부수적인 작업이 많이 이루어지고 있는데, 이런 작업은 프로그래머로서의 능력을 입증하는 것은 아니므로 임금을 상승시킬 가능성이 낮다. 또 명성을 위해 현재 프로젝트의 관리자에게 도전하거나 기존의 프로젝트에서 독립하여 자신의 프로젝트를 만드는 일(forking)이 빈번하게 나타나야 하지만, 그런 일은 드물게 관찰될 뿐이다(Hertel, Niedner and Herrmann, 2003).

개발자에게는 금전적인 보상이나 명성과 같은 외재적 동기(extrinsic motivations)가 아니라 내재적 동기(intrinsic motivations)가 더 중요하다는 반론도 제기되었다. 외재적 동기는 어떤 행위와 연결된 직간접적인 보상이 주는 원인에서 행위의 원인을 찾는 것이고, 내재적 동기는 행위 그 자체가 주는 즐거움이 동기가 된다는 것이다. 내재적 동기는 크게 두 가지, 첫째 즐거움 또는 둘째 의무나 공동체 활동에서 비롯된다고 본다. 취미 생활로서의 즐거움, 자신의 수정 사항을 공유하거나 선물하는 데서 오는 즐거움, 내가 도운 대로 상대방도 누군가를 도울 것이라는 일반화된 호혜성이 OSS 개발 참여에 중요한 역할을 한다는 것이다(Bergquist and Ljungberg, 2001; Krishnamurthy, 2006).

개발자의 참여 동기에 대한 실증 연구는 이 두 가지가 혼재되어 있음을 보여준다. 많은 개발자들이 외재적 동기, 즉 커뮤니티에서의 명성이나 상호 인증, 자기 홍보 효과, 프로그래밍 능력의 향상 등이 참여에 영향을 미쳤다고 응답했다. 동시에 이러한 외재적 동기보다 내재적 동기를 더 중요하게 평가하고 있는 것도 확인된

다. 개발의 즐거움, 지적인 자극, 창의성의 표현 등과 같은 내재적 동기를 중요 요인으로 응답한 비율이 외재적 동기보다 높게 나타났기 때문이다(Baytiyeh and Pfaffman, 2010; Hertel, Niedner, and Herrmann, 2003).

개발 참여 동기에 대한 실증 연구는 연구자가 중요하다고 생각하는 행동동기를 제시하고, 이 중 개발자가 중요하다고 생각하는 것을 선택하도록 하는 방식의 설문 조사라는 한계가 있다. 이를 극복하기 위해 개발자가 응답한 행동 동기에 따라 실제 얼마나 많은 시간을 투입하는가가 조사되었는데, 더 많은 작업 시간을 투입하도록 하는 것은 외재적 동기였다(Hars and Ou, 2002).

OSS 개발이 계속됨에 따라 개발 참여 동기와 지속 동기를 구분한 연구도 가능해졌는데, 개발자의 참여가 지속되는 과정에서 내재적 동기와 외재적 동기의 경계가 흐려지고 참여 이유도 지속적으로 변화하는 것으로 나타났다. 대부분의 개발자가 자신의 필요를 채우면 참여한 OSS 프로젝트에 남아 있지 않고 프로젝트를 떠나는 것으로 확인되었다. 참여 지속을 결정하는 데 영향을 주는 최초 참여 동기는 뚜렷하게 나타나지 않았다. 개발에 계속 참여하거나 다른 프로젝트의 개발에 새로이 참여하도록 하는 이유는 학습이나 취미가 가장 큰 요인으로 보인다. 기술을 습득하고 능력을 배양하기 위해 프로젝트에 참여하지만, 자신의 지식과 능력을 다른 사람과 나누고 싶어서 프로젝트에 계속 남아 있음을 밝힌 연구도 있는데, 한국의 OSS 개발자를 대상으로 한 인터뷰에서도 동일한 사실이 확인된다(송우일, 2013; Fang and Neufeld, 2009; Freeman, 2007).

개발자 개인의 사회경제적 특성도 참여 동기와 지속 동기에 영향을 주는 것으로 나타났다. 설문조사 또는 커뮤니티 회원 등록 자료를 이용하여 개발자의 사회경제적 조건을 확인한 연구에 따르면, OSS 개발자는 남성이 많고, 북미와 유럽(주로 독일, 영국, 프랑스)에 집중되어 있다. 설문조사를 이용한 한 연구에 따르면, OSS 개발자 집단을 ‘전문가’, ‘열정적인 해커’, ‘사회적 학습자’, ‘사회적 프로그래머’, ‘사용자 혁신가’의 다섯 개로 나누었을 때¹⁰⁾, 전문가와 해커는 대규모 프로젝트에 참여하는 경우가 많지만, 다른 세 집단은 소규모 프로젝트에 참여하는 경우가 많은 것으로 나타났다. 그리고 학생이나 취미 활동으로 개발에 참여하는 경우 전문가에 비해 내재적 동기가 더 크게 작동한다는 차이도 확인되었다(David and Shapiro, 2008;

10) 분류 기준은 F/LOSS 철학에 동조하는가, 관련 직종의 임금을 받고 있는가, 기술 수준이 높은가 등이다.

Gonzalez-Barahona, Robles, Andradas-Izquierdo and Ghosh, 2008; Hars and Ou, 2002).

개발자가 속한 사회의 특징이 참여에 미치는 영향에 관한 연구도 제한적이거나 진행되었다. 독일의 컴퓨터 잡지에서 8,000여명의 정보통신기술 종사자를 대상으로 수행한 조사를 이용하여 OSS 개발 참여에 영향을 주는 요소를 분석한 연구에 따르면 공식적인 교육 수준과 OSS 개발 참여가 양의 상관관계를 보이지 않았지만, 직장에서 OSS개발로 임금을 받는 것과 여가 시간은 OSS 개발 참여와 양의 상관관계를 보이는 것으로 나타났다. 거시적인 사회경제적 조건과 OSS 개발 참여의 상관관계를 분석한 연구도 있다. 소스포지 자료를 이용하여 데이터베이스에 등록된 개발자의 이메일 주소, IP 주소, 선호하는 시간대를 조합하여 국적을 추정하고, 이를 세계 은행(World Bank) 등 국제기구와 세계 가치 조사(World Value Survey)의 자료와 비교하였다. 이 연구에 따르면 자기 결정권, 상호 신뢰, 지적재산권 보호 정도가 높은 사회에 속해 있을수록 OSS 참여도가 높은 것으로 나타났지만, 새로운 아이디어에 대한 선호, 경쟁에 대한 선호는 유의미한 영향을 주지 않았다(Bitzer and Geishecker, 2010; von Engelhardt and Freytag, 2013).

개인의 참여와 더불어 기업의 활발한 기여도 흥미로운 분석 대상이다. 기업은 자신의 개발 인력을 OSS 프로젝트에 직접 참여시키거나, OSS 프로젝트 운영에 관한 비용을 지원하는 간접 방식으로 개발에 참여한다. IBM은 1999년 리눅스에 대한 지원을 공표했고, 2001년 10억 달러의 재정지원과 함께 자사 인력의 근무 중 오픈소스 프로젝트 기여를 용인했다. 2011년 7월에 발표된 리눅스 커널 3.0 개발에 참여한 기업의 수는 191개였으며, 2013년 6월에 발표된 커널 3.10 개발에서는 참여 기업의 수가 243개로 증가했다(Corbet, Kroah-Hartman and McPherson, 2013).

기업은 오픈소스 소프트웨어 개발이 기업 외부의 “많은 눈”을 활용함으로써, 자신의 자원을 사용하지 않더라도 직면한 문제를 해결하는 이점을 누릴 수 있다. 그리고 기업의 코드 공개는 호혜적인 행동(reciprocal behavior)이 되어, 개발자 커뮤니티의 다른 기업 또는 다른 개발자들의 기여를 유인할 수 있다. 그렇더라도 기업은 여전히 자신이 개발한 코드 중 일부만을 공개할 유인은 남아 있다.

OSS 커뮤니티가 개발한 코드가 네트워크 효과를 갖는 경우에도 기업이 OSS를 지원할 유인이 있다. 공개된 소스코드는 기업이 스스로 개발하는 코드를 보충할 가능성이 높고, 공개된 코드에 대한 기업의 지원은 소프트웨어 간의 호환성을 높일

수 있다. 한편 OSS 커뮤니티의 기술 개발로 인해 소프트웨어 기능이나 성능이 개선된다면, PS 기업이 생산하는 보완재에 대한 소비자의 지불가능의사가 높아질 수 있다. 또 사용자 혁신의 장점으로 인해 OSS가 더욱 확산되어 사용된다면, 기업은 OSS에 대한 투자를 더 늘리게 될 것이다(Bonaccorsi and Rossi, 2006; Mustonen, 2005; Reisinger, Ressner, Schmidtke and Thomes, 2014).

기업은 OSS를 이용하여 어떻게 이윤을 얻을 수 있을까? 하드웨어와 소프트웨어가 보완재로 판매되는 시장에서 OSS와 PS가 과점 경쟁을 벌인다고 가정해보자. OSS는 개발 비용을 외부화하거나 초기 개발 비용을 감소시킬 수 있으므로 하드웨어 기업이 오픈소스 개발에 기여하는 것은 합리적일 수 있다. 이러한 논리를 확장해보면, 전자 기기나 서버 같은 하드웨어 생산자가 OSS에 참여하는 현상뿐만 아니라, 기술 교육, 기술 컨설팅, 웹 호스팅 등의 서비스를 제공하는 기업이 OSS에 참여하는 것도 설명 가능하다. 실제로도 리눅스를 토대로 성공한 레드햇(Red Hat Inc.)의 사업 모델은 개별 소비자의 용도에 따른 설치 또는 교육 등의 관련 보완재를 판매하는 것이다(Di Gaetano, 2015).

정리해보면 개인의 OSS 개발 참여를 설명하려는 이론은 외재적인 동기와 내재적인 동기 중 어느 하나에만 치우쳐 있지만, 실증 연구는 이 둘이 혼재되어 있음을 보여주고 있다. 개발자는 커뮤니티에서의 평판을 높이거나 신호 효과로 소득을 늘리고자 하는 목적도 갖고 있지만, 순수하게 공유의 즐거움이나 자기 만족의 취미 생활을 위해서도 OSS 개발에 참여하기도 한다. OSS의 생산은 개발자 커뮤니티에서 이루어지고, 기업은 이를 판매하여 직접적인 이윤을 얻을 수는 없지만 관련된 서비스나 기기를 보완재로 판매함으로써 이윤을 얻을 수 있다. 또 기업의 코드 공개가 다른 개발자의 코드 공개를 유도할 수 있으므로, 기업은 직간접적으로 OSS 개발에 참여할 유인이 있다.

최근의 연구는 다양한 참여 동기의 존재를 인정하고, 다양한 목적이 있는 행위자가 어떻게 조정되는지, 즉 누가 무엇을 개발할 것인가, 무엇을 먼저 개발할 것인가의 문제가 어떻게 해결되는지와 그 결과가 시장에서 어떻게 다른 소프트웨어와 경쟁하는지에 더 많은 관심을 기울이고 있다. 다음 소절에서는 이러한 거버넌스와 시장 경쟁에 관한 연구를 정리했다.

2. OSS 개발 과정과 시장 경쟁

1990년대 후반 이후 성공한 OSS 프로젝트는 모두 개발자 커뮤니티를 기반으로 한다. 사용자이자 개발자인 사람들이 커뮤니티를 만들어서 소프트웨어의 사용 목적, 개발 일정, 기능 우선순위 등을 스스로 결정하는 자유와 코드 공개의 효율성을 누릴 수 있었다. 이러한 생산 방식은 사용자 혁신(user innovation) 또는 공유 기반의 상호 생산(common based peer production)으로 설명되기도 한다(Benkler, 2002; von Hippel and von Krogh, 2003).

경제학에서 OSS 개발의 조정 문제를 다루는 방향은 생산에서의 무임승차를 어떻게 해결할 것인가와 실제 생산과정을 어떻게 조직할 것인가의 세부적인 문제에 초점을 맞추고 있다. 모듈화라는 OSS의 기술적 특징¹¹⁾은 개발자의 무임승차를 줄이고 공공재의 사적 공급을 가능하게 할 수 있다. 개발자는 자신의 비용-편익 비율이 문턱점을 넘어야 개발에 참여한다고 하자. 개발자는 자신의 비용을 들여 소프트웨어의 일부분을 수정할 것이다. 그리고 수정된 부분이 반영되어 소프트웨어가 구동될 때 사용 이득이 발생할 것이다. 사용 이득에는 자신의 수정으로 인해 소프트웨어 전체가 제대로 작동하는 지 그리고 다른 사람도 그들의 수정을 공개할 지의 두 가지 불확실성이 존재한다. 모듈화는 어떤 한 부분의 수정이 전체의 작동 여부에 영향을 미치지 않는다는 의미이므로 기술적인 불확실성을 제거하게 된다. 다른 한편 개발자 모두에게 자신이 수정한 부분을 공개하는 것이 전체적인 이득을 높이고 중복 수정의 가능성을 낮추는 것이 확실해지므로, 개발자는 각자의 모듈을 공개하는 선택을 하게 된다(Baldwin and Clark, 2006).

OSS 개발자는 온라인 커뮤니티를 통해 연결되므로, 특정한 기능을 하는 소프트웨어를 만들고자 하는 공통의 목적을 위해 이들을 조직하는 방식은 어려운 문제이다. OSS 커뮤니티의 조직 원칙에 관해 Markus(2007)는 가장 좋은 개념 연구이며 관련 사례 연구도 잘 정리하고 있다. OSS 커뮤니티는 집단 행동, 조정 문제의 해결, 개발자들이 특정 프로젝트에 기여할 수 있도록 하는 환경 조성의 세 가지 문제

11) 컴퓨터 프로그래밍의 모듈화는 프로그램의 기능을 독립적이고 호환 가능한 부분(모듈)으로 분리하고, 각 부분은 프로그램에서 필요로 하는 단 하나의 기능만 수행할 수 있도록 하는 소프트웨어 디자인 방식이다. 모듈화는 물리적인 상품 생산에서도 많이 이용되는 데, 자동차 산업이 대표적인 예이다.

를 풀어야 한다. 이러한 목표를 달성하기 위해 소유 원칙, 허용 원칙, 커뮤니티 원칙, 소프트웨어 개발 원칙, 갈등 해결 및 원칙에 관한 원칙, 정보 및 도구 원칙 등 6개의 장치가 사용된다. de Laat (2007)는 자생적인 거버넌스가 나타나고, 이후 내부 거버넌스로, 마지막으로 외부자에 대한 거버넌스 구조로 발전해나간다고 설명하고 있다. 중요한 내부 거버넌스 장치는 6개로 모듈화, 역할의 분담, 의사 결정의 위임, 훈련과 계발, 형식화, 독재와 민주주의이다. OSS 커뮤니티의 조직 형태에 관해서는 네트워크형, 클럽형, 양파형 위계 등¹²⁾ 다양한 설명이 있다. 커뮤니티는 회원 간의 이메일, 홈페이지 게시판, CVS (Concurrent Versioning Systems)를 이용한 업데이트 관리를 의사 소통의 수단으로 사용하고 있다. 이를 통해 소프트웨어에 필요한 기능, 개발 일정 및 진행 과정, 개선 사항 및 기타 다양한 문제에 관한 정보가 유통된다.

한편 이윤을 추구하는 기업과 소프트웨어 개발을 추구하는 커뮤니티는 의사결정 과정에서 갈등을 빚을 수 있다. 이런 맥락에서 기업이 OSS 커뮤니티에서 어떤 역할을 할 것인가에 대한 연구도 진행되었다. Dahlander and Magnusson (2005)는 기업의 전략을 기업과 커뮤니티 모두 이득을 보는 공생, 상호 무차별한 공존, 기업은 이득을 보지만 커뮤니티는 손해를 보는 기생의 세 가지로 나누고, 데이터베이스 소프트웨어인 미에스큐엘을 사용하는 기업의 사례를 통해 세 가지 전략을 어떻게 실제로 구사하는지 설명했다. West and O'Mahony (2008)는 기업이 시작한 12개의 OSS 프로젝트를 대상으로 기업이 커뮤니티의 생산과정, 의사결정 과정, 지적재산권 3개의 영역에서 어떤 역할을 하는지 정리했다. 기업은 자신이 시작한 OSS 프로젝트에 대한 투자로부터 이윤을 얻는 데 목표를 두지만, 참여한 개발자는 공유된 기술에 대한 능력을 키우는 데 초점을 맞추었다. 또한 기업이 의사결정에 절대적인

12) 네트워크형은 비중앙집중적, 비위계적 의사소통에 토대를 둔 의사결정 구조를 지칭한다. 클럽형은 OSS 개발자나 사용자가 프로젝트 구성원으로 자발적으로 참여함과 동시에 일정 정도의 의무를 지키면서 소프트웨어를 사용한다는 점을 강조하는 입장이다. 이 둘은 의사결정이 수평적으로 이루어진다는 점에서는 유사하다. 그리고 이러한 입장은 레이몬드와 같이 OSS를 지지한 실천가가 묘사한 커뮤니티 구조이다. 하지만, 현실적으로 거의 모든 OSS 프로젝트에서 활발히 활동하는 개발자는 극소수로 국한되어 있으며, 의사결정도 이들에게 집중된 특징을 보이고 있다. 양파형 위계는 프로젝트를 시작한 극소수의 사람을 중심에 두고 그 외부에 소수의 핵심 개발자-다수의 공동 개발자-상당한 수의 적극적 사용자-대다수의 소극적 사용자로 커뮤니티 인원이 점차 증가하면서, 중심부로 갈수록 더 많은 의사결정권을 갖는 구조를 지칭한다(Crowston and Howison, 2006).

영향력을 행사하는 특징을 보였다.

자발적인 참여가 지속되고 효율적으로 운영된다면 OSS는 사용 비용이 없으므로, PS에 비해 시장경쟁에서 유리할 수 있다. 실제로도 OSS는 시장에서 PS를 몰아내고 있을까? 인터넷 인프라 소프트웨어인 샌드메일(sendmail)과 바인드(bind), 웹 브라우저인 모질라 파이어폭스(Mozilla Firefox), 사무용 프로그램인 오픈오피스(OpenOffice), 그래픽 프로그램인 김프(Gimp), 프로그래밍 언어 인터프리터와 컴파일러인 펄(Perl)과 파이썬(Python), 프로그래밍 환경인 이클립스(Eclipse), 기업 시스템인 이그룹웨어(eGroupware)와 오픈씨알엑스(openCRX), 콘텐츠 관리 시스템인 워드프레스(wordpress), 통계 패키지인 알(R) 등, 사실상 거의 모든 소프트웨어 부문에서 PS에 대응하는 OSS를 찾을 수 있다.

시장점유율 측면에서 OSS와 PS의 시장점유율을 정확히 비교평가하기는 어렵다. 다만 일부 소프트웨어의 경우 사용자를 파악할 방법이 있는데, <Table 2>는 선정된 웹사이트의 방문자 기록을 수집하여 데스크톱 운영체제, 웹 브라우저, 웹 서버의 사용 현황을 조사한 자료이다. 소수의 사례이기는 하지만 이 분야의 OSS는 개발자 규모, 성공, 영향력 면에서 성공적이라고 꼽히는 리눅스, 파이어폭스, 아파치가 조사대상이었어서 비교의 의미가 있다.

<Table 2> Usage Share of Operating Systems and Browsers

Categories Ranks	Desktop Operating Systems		Desktop Browsers		Web Servers	
1	Windows	(91.39%)	Internet Explorer	(50.03%)	Apache	(37.00%)
2	Mac	(6.99%)	Chrome	(31.41%)	Microsoft Server	(27.12%)
3	Linux	(1.62%)	Firefox	(12.24%)	nginx	(16.61%)
4			Safari	(4.33%)	Google	(2.17%)
5			Opera	(1.52%)	Others	(17.10%)
6			Others	(0.47%)		

Notes: As of November, 2015; Italics are OSS.
Sources: marketshare.hitslink.com (Desktop Operating Systems and Browsers), netcraft.com (Web servers), last accessed December 29, 2015.

데스크톱 운영체제에서 OSS의 점유율은 2%에도 미치지 못한다. 하지만, 웹 서버 시장은 약 54%를 차지하고 있다. 웹 브라우저로는 세번째로 많이 사용되고 있으며, 웹킷(webkit)이라는 OSS를 토대로 개발되는 사파리(Safari)와 역시 블링크(Blink)라는 OSS를 토대로 하는 크롬(Chrome)과 오페라(Opera)를 포함시키면 인터넷 익스플로러(Internet Explorer)를 제외한 주요 브라우저는 모두 오픈소스에 기반하고 있는 셈이다. 일부 소프트웨어의 사례이기는 하지만, OSS는 웹 서버 같은 개발자 중심의 인터넷 인프라 프로그램에서는 강점이 있지만, 운영체제나 웹 브라우저와 같은 일반 사용자 중심의 프로그램에서는 약점이 있음을 유추할 수 있다. 또한, PS나 OSS 어느 하나가 시장을 전부 차지하기 보다는 일정하게 나눠 갖고 있다고 할 수 있다. 소프트웨어 각각의 시장 점유율뿐만 아니라, 기업 내부에서도 OSS와 동시에 PS를 사용하는 경우가 많은 것으로 나타나고 있다(Lerner and Satterthwaite, 2010).

그렇다면, OSS와 PS는 왜 공존할까? 하나의 이유는 전환비용, 네트워크 효과, 모듈화 같은 기술적 특징이다. 만약 OSS와 PS의 기술 수준이 동일하다면, 사용자 입장에서는 비용이 전혀 들지 않는 OSS가 가장 유리한 선택일 것이다. 하지만, 소프트웨어 사용의 외부성이나 전환 비용이 발생한다면, PS가 이미 시장 점유율이 높을 조건에서 OSS가 시장에 신규 진입하기 어려울 수 있다. Casadesus-Masanell and Ghemawat (2006)은 기존 사용자 수로 인해 네트워크 효과가 발생하여, 윈도우즈와 리눅스 사용자 증가가 누적적인 영향을 받음을 보여주었고, 기존 사용자 효과가 클수록 이 둘이 시장에 공존할 수 있음을 보였다. 소프트웨어의 모듈화 정도도 OSS의 확산에 영향을 줄 수 있다. Landini (2012)는 소프트웨어로부터의 수익과 비용이 모듈화와 투입되는 노동량에 영향을 받는다고 가정할 때, OSS와 PS 중 어느 방식을 채택하는 것이 유리한지 모형화 했다. 이 모형에 따르면 모듈화 비용이 낮다면 두 시스템이 공존할 수 있으며, 소프트웨어로부터의 기대 수익이 크다면 PS가 채택될 가능성이 높아진다.

두 소프트웨어가 시장에서 공존하는 또 다른 이유는 PS가 연구개발과 가격 설정에서 OSS에 비해 장점이 있기 때문이다. OSS가 기술 발전의 상당 부분을 자발적 활동에 의존하는 데 반해, PS는 연구개발 투자를 집행할 수 있으므로, 기술 발전 면에서 PS가 유리할 수도 있다. Llanes and de Elejalde (2013)는 OSS와 PS의 공존이 비대칭적 시장 구조 때문임을 보였다. PS는 연구개발에 더 많이 투자할 수 있

고 따라서 시장 점유율을 높일 수 있다. 반면 OSS의 장점은 낮은 개발 비용에 있다. 따라서 연구개발 투자의 유용성이 높고 OSS의 보완성이 강하고 OSS의 차별화 가능성이 높으면, OSS의 성공 가능성도 높아진다. 연구개발뿐만 아니라 PS는 OSS보다 많은 상품에 가격을 설정할 수 있으므로 이윤을 높일 수 있는 장점이 있다. Economides and Katsamakas (2006)는 양면 시장 모형을 응용해 소프트웨어를 플랫폼과 애플리케이션으로 나누어 분석했다. PS 기업은 플랫폼과 애플리케이션을 수직 결합해 가격을 부과하고 이윤 극대화 전략을 구사할 수 있지만, OSS 프로젝트는 플랫폼에 가격을 부과하지 못하고 애플리케이션에만 가격을 부과할 수 있다. 이 모형은 수직결합된 OSS 프로젝트에 대한 수요가 수직결합된 PS 기업의 시스템에 대한 수요보다 크지 않다면, 시장 점유율과 이윤 모두에서 PS 기업이 OSS 프로젝트를 압도할 수 있음을 보여주었다.

앞에서 언급된 기술적 요인과 경제적 요인을 모두 고려한 연구도 있다. Bonaccorsi and Rossi (2003)는 내재적 가치에 대한 평가, 소프트웨어 사용자 수에 따른 네트워크 효과, 개발자 집단의 수가 증가함으로써 발생하는 네트워크 효과, 오픈소스 소프트웨어 확산에 따른 소프트웨어 산업 내 이익 추구 기업의 연구개발 투자 증가와 품질 개선 노력을 주요 변수로 하여 OSS 채택 함수를 구성하였고, 이를 토대로 행위자 기반 시뮬레이션을 구현했다. 주요 변수 값을 변화시킨 시뮬레이션 결과에 따르면 PS가 지배적인 소프트웨어가 될 가능성이 높으며, 나아가 OSS가 시장에 자리 잡더라도 일반적인 상황은 두 소프트웨어가 공존하는 것이지 OSS가 일방적으로 시장을 점유하는 것은 아닌 것으로 나타났다.

그러면 OSS와 PS의 공존이 사회적으로 바람직할까? Bitzer and Schröder (2007)는 OSS 기업이 독점 시장에 진입하여 복점 시장으로 전환되고 기업이 가격이나 수량이 아니라 기술 경쟁을 한다는 가정을 전제하고 모형 분석을 통해 OSS 간의 경쟁이나 OSS와 PS의 경쟁이 혁신에 긍정적 효과가 있음을 보여주고 있다. 독점에서 복점으로 전환됨으로써 기업은 더 많은 혁신 노력을 기울이게 되고, 따라서 산업 전체의 기술 수준이 높아지게 된다. Haruvy, Prasad and Sethi (2008)은 시장의 복점 경쟁을 가정하고 1단계에서 기업이 OSS와 PS 중의 개발 방식을 선택하고, 2단계에서 공개 기술과 전유 기술 중 어느 한 쪽에 투자할 것인지 선택할 수 있는 최적 선택 모형을 고안했다. 시장점유율, 소프트웨어 가격, 소프트웨어 중 공통 부분의 개발 비용과 품질, 소프트웨어 중 전유 부분의 개발 비용과 품질 등에 대한 수치

해석(numerical analysis)은 경쟁 기업이 있을 때 오픈소스 전략이 기업 이윤과 사회적 잉여를 극대화 할 수 있음을 보여주었다. 이와 달리 Llanes and de Elejalde (2013)는 둘이 공존하는 균형 상태는 사회적 최적이지 아닐 수 있다고 보았다. PS 기업의 협력이 너무 적거나 OSS 프로젝트의 연구개발 투자가 너무 적게 되기 때문이다.

OSS가 기업 활동에 어떤 영향을 미치고 있는가에 대한 실증 연구도 최근 이루어지고 있다. Campbell-Kelly and Garcia-Swartz(2010)는 사례 연구를 통해 OSS 기업과 PS 기업의 생산 방식, 사업 모델, 전략적 행동이 유사해지고 있음을 보여주고 있다. 예를 들어 레드햇, 미에스큐엘, 콤피에르(Compiere) 같은 OSS 기업은 PS 기업과 마찬가지로 이중 라이선스 전략(double licensing strategy)을 사용하고 있다. 또 OSS 기업이나 PS 기업 모두 하드웨어 판매가 아니라 서비스 부문에서 주로 수입을 얻고 있다. Lamastra(2009)는 134개의 이탈리아 중소기업에 대한 설문조사와 전문가 평가를 토대로 OSS가 기업의 혁신에 미치는 영향을 분석했다. 조사 대상 기업 중 소프트웨어 솔루션으로 PS를 사용하는 기업은 109개였으며, OSS를 사용하는 기업은 27개였다. 조사 대상 기업이 만드는 소프트웨어 상품, 모듈, 플랫폼이 기업 내부와 시장 모두에서 얼마나 새로운 것인가를 3명의 전문가가 평가했는데, 모든 영역에서 OSS를 사용하는 기업의 혁신성이 전반적으로 높은 것으로 나타났다. Colombo, Piva and Lamastra(2014)는 중소기업 관리자와의 인터뷰와 기업 자료를 이용하여 OSS 커뮤니티에 기여하는 EU 중소기업의 소프트웨어 산업 내 다각화를 연구했다. 기업 규모는 산업 내 다각화에 음의 효과를 주었지만, 많은 OSS 프로젝트에 기여하고 있는 기업은 더 다양한 소프트웨어 상품군을 생산하고 있었다. 나아가 중소기업이 충분한 내부 기술 자원을 갖고 있다면 고용된 인력이 근무 시간 중 OSS 프로젝트에 자발적으로 참여할 수 있도록 하는 것이 산업 내 다각화에 긍정적 효과를 줄 수 있는 것으로 나타났다.

지금까지의 연구를 정리하면, OSS 커뮤니티는 소프트웨어 개발을 위해 체계적인 의사결정 구조를 갖고 있으며, 다양한 종류의 거버넌스 장치를 활용하고 있다. 거의 모든 소프트웨어 분야에서 OSS 프로젝트를 찾을 수 있다. 하지만, 프로그래머 수준을 넘어 많은 사람이 OSS를 대규모로 사용한다는 증거를 찾기 어려운 것이 사실이다. OSS의 구입비용이 무료라는 특징에도 불구하고 프로그래밍 지식이 없는 일반 사용자의 OSS 사용이 낮은 이유를 해명하는 것은 앞으로의 중요한 연구과제

가 될 것이다. 한편 시장에서 OSS는 PS와 시장에서 공존하고 있다고 봐야 한다. PS 사용에는 비용을 지불해야 함에도 PS가 시장에 남아 있을 수 있는 이유는 이미 갖고 있는 네트워크 효과와 더 많은 기술 개발 투자가 가능하기 때문이다. 다음 절에서는 OSS 개발을 가능하게 하는 제도적 장치로서 라이선스의 역할에 대한 연구를 정리하고자 한다.

3. OSS 라이선스의 역할

소프트웨어는 그림, 출판물, 음악 등의 상품과 마찬가지로 복제가 쉬워서 최초의 창작자에게 비용을 지불하지 않더라도 이용 가능하기 때문에 복제를 막기 위한 보호 장치가 필요하다. 다른 한편으로 복제의 비용, 즉 창작의 한계비용을 지불할 의사가 있는 사람이 이용할 수 있을 수 있도록 복제를 허용하는 장치가 필요하다. 따라서 이러한 상품의 거래는 완전한 권리의 이전이 아니라 계약을 통해 사용의 범위를 결정하는 것이 일반적이다. 이 계약은 창작자의 이득과 다수의 이용이라는, 다소 상충하는 이해관계를 모두 보장할 수 있어야 한다. OSS도 특정 라이선스에 따라 배포된다. OSS 라이선스의 중요한 특징은 OSS 개발에 대한 참여와 배제를 조정하는 법적 도구로서, 소프트웨어의 충분한 개발을 보장하기 위해 의도적으로 고안된 제도적 장치라는 점이다.

스톡먼이 설립한 자유소프트웨어 재단은 현재의 지적재산권법 테두리 안에서 소프트웨어에 대한 자유로운 접근을 허용할 수 있는 법적 장치를 만들고자 했다. 그 결과가 현재까지 가장 널리 사용되는 라이선스인 GPL이다. 이외에도 LGPL (Lesser GPL), MPL (Mozilla Public License) 등 다양한 종류¹³⁾의 OSS 라이선스가 존재한다. O'Mahony (2003)는 주요 OSS 프로젝트가 라이선스를 어떻게 활용하고 있는지를 사용의 제한, 법적 제재, 상표권 사용 및 브랜드의 보호 등 기능별로 나누어 구체적으로 설명하고 있다. 특히 재단에 라이선스를 이관하고 관리를 맡기는 방식의 장점을 보여주고 있다.

다양한 종류의 OSS 라이선스가 있지만, 경제적으로 가장 중요한 내용은 소스코드의 재배포에 관한 것이다. OSS가 PS에 대해 우위를 갖기 위해서는 지속적인 개

13) 이들의 개별적인 특성을 정리하는 것은 이 글의 범위를 벗어난다. 라이선스에 대한 정리는 Open Source Initiative 홈페이지에서 확인할 수 있다.

발과 혁신이 전제되어야 하는데, OSS의 공공재적 성격으로 인해 이를 보장할 수는 없다. 라이선스는 공공재의 과소 공급과 무임승차 문제를 해결하고, 개발자의 참여를 유도하는 제도로서 기능해야 한다.

먼저 왜 소스코드를 공개하는 라이선스가 채택되는지 살펴보자. 소스코드의 공개는 소프트웨어 개발과 관련된 거래비용을 줄일 수 있다. Bessen (2006)은 소프트웨어 사용에서 발생하는 모든 문제를 염두에 두고 소프트웨어 코드를 작성할 수 없고 개발된 소프트웨어가 사용자의 필요성을 모두 만족시킨다는 보장이 없다는 점에 주목했다. 이로 인해 불완전 계약과 홀드 업 문제가 발생하며 거래비용이 높아질 수 있다. 소스코드가 공개된다면, 사용자는 자신의 요구에 맞는 소프트웨어를 만들 수 있고, 기업은 더 많은 기능이 포함된 복잡한 소프트웨어를 더 효율적으로 만들 수 있는 장점이 있다. 특히 전문적인 요구를 수용해야 하면서 자체적인 개발 능력을 가진 기업일수록 소스코드의 공개를 선호할 수 있다. Scotchmer (2010)는 순차적인 혁신 모형을 활용하여 어느 한 쪽이 선도자가 될지 모르는 경우에 소스코드가 공개될 수 있는 이유를 보여주하고자 했다. 두 기업이 있는 경우, 어느 한 기업이 혁신 선도자라는 것이 알려져 있다면, 소스코드를 공개하지 않는 것이 가장 유리한 전략이 될 것이다. 그렇지만 오픈소스 라이선스를 사용하는 것이 산업 전체의 이득을 극대화할 수 있을 것이 알려져 있고, 누가 첫번째 혁신 기업인지 모르는 상황이라면, 두 기업은 “무지의 베일”에 있는 상황이라 할 수 있고, 그렇다면 두 기업 모두 오픈소스 라이선스를 선택할 수 있다.

소스코드 공개에 대한 실증 연구로서, Harison and Koski (2010)는 핀란드의 소프트웨어 170개 기업에 대한 설문조사를 통해, 기업의 소스 코드 공개 여부에 미치는 요소를 분석했다. 종업원의 교육 정도로 측정된 인적 자본은 OSS 채택에 양의 영향을 미쳤으며, 고용자 중 개발자의 비중이나 지적재산권의 보유 정도는 통계적으로 유의미한 영향을 미치지 않았다. 기업 규모가 작고, 서비스 중심이며, 신규 진입한 기업일수록 OSS 채택에 적극적이었지만, 기업의 설립 시점은 유의미한 영향을 미치지 않았다. 가족이나 개인이 소유한 기업일수록 PS를 채택하는 경우가 많았으며, 소유가 분산되어 있는 기업일수록 OSS를 사용했다.

소스코드를 공개한다면 어떤 라이선스로 공개를 할까? OSS 라이선스는 재배포에 대한 재량권을 “더” 또는 “덜” 부여하느냐에 따라 구별될 수 있는데, 이러한 차이는 경제적인 행동 유인에 영향을 줄 수 있다. 개발자가 지대추구자라면 공개된 소스코

드에 자신의 소스코드를 추가하고 이를 공개하지 않고 배타적으로 사용할 유인을 갖는다. 또한 지대추구 행위는 직접적 대가없이 소스코드를 공개하는 기여자를 몰아낼 가능성이 있으므로, 라이선스는 이를 막을 수 있어야 한다. 다른 한편, 라이선스가 너무 엄격하게 공개를 강조하게 되면, 기업 등 사적 이익을 고려하는 행위자의 참여를 막는 걸림돌이 될 수도 있다. 이러한 입장에서 보면 더 많은 개발자가 참여할 수 있는 여지를 만드는 것이 중요할 수 있다. GPL은 가장 엄격하고 카피레프트의 성격이 강한 라이선스로서, GPL로 배포된 코드를 사용한 소프트웨어는 GPL과 동일한 라이선스로 재배포 되어야 한다. 관용도가 높은 라이선스는 버클리 소프트웨어 개발 라이선스(이하 BSD: Berkeley Software Development)로 이 라이선스를 사용하는 개발자는 최초 저작권자를 명기하기만 하면 되고, 자신이 추가한 코드를 공개할 필요는 없다.

Franck and Jungwirth(2003)는 GPL이 지대추구자와 순수한 기여자 모두에게 유인이 될 수 있고, 이와 동시에 지대추구자가 기여자를 몰아내지 않는 조건이 된다는 주장을 펴고 있다. GPL이 금지시키고 있는 것은 추가적인 개발을 재배포하지 않는 것인데, 이는 공공재의 생산에 기여하지 않고 사적 이익만을 추구하는 것을 막겠다는 뜻이지, 사적 이득 자체를 금지하는 것은 아니다. OSS를 이용한 이윤 추구를 배제시키고 있지 않기 때문에 공개된 소스코드로부터 이익을 얻고자 하는 행위자를 유인할 수 있으며, 이와 동시에 무임승차자를 배제시킴으로써 지속적으로 코드의 기여가 이뤄질 수 있도록 한다는 것이다. D'Antoni and Rossi(2007)는 소프트웨어가 사회적 이득을 발생시킬 수 있고, 이러한 이득은 소프트웨어 개발자의 기여 정도와 개발자의 능력 수준에 의해 결정된다고 가정했다. 이와 더불어 GPL 체계에서는 비용 없이 코드가 공개되지만, BSD 체계에서는 코드 공유를 위한 협상 비용이 발생한다고 보았다. 협상 비용으로 인해 다수의 개발자로부터 협력을 이끌어내고 이를 조정하기 위해서는 GPL이 우월하다. BSD는 피드백이 필요하지 않을 때 다른 개발자로의 확산 효과를 극대화하는데 더 적합할 수 있다.

더 제한적인 라이선스와 상대적으로 덜 제한적인 라이선스의 사용에 관한 실증 연구로서, Hofmann, Riehle, Kolassa and Maurer(2013)는 오토(Ohloh.net)의 데이터베이스를 이용해, 2008년을 기준으로 GPL 3,248개를 비롯하여 더 제한적인 라이선스를 사용하는 프로젝트는 총 3,272개이고, BSD 730개를 포함하여 덜 제한적인 라이선스를 사용하는 프로젝트는 총 1,883개임을 보여주었다. 나아가 2001년

까지 더 제한적인 라이선스가 상대적으로 빠르게 성장했지만, 이후에는 덜 제한적인 라이선스가 빠르게 성장하고 있음을 보였다. Sen, Subramaniam and Nelson (2008)은 어떤 성향의 개발자가 어떤 라이선스의 프로젝트를 선택하는지 확인하기 위해 온라인 설문조사 자료를 이용했다. 이 연구에 따르면 문제 해결이라는 동기를 가진 개발자는 중간 정도의 제약이 있는 라이선스를, 동료의 인식을 통해 지위를 높이하고자 하는 개발자는 거의 제약이 없는 라이선스를 선호하는 것으로 나타났다.

그런데 과연 라이선스의 성격이 개발 성과에 영향을 줄까? 이를 확인하기 위해 Lerner and Tirole (2005a)은 소스포지의 자료를 사용했다. GPL처럼 제한적인 성격이 강한 라이선스를 채택한 프로젝트의 규모가 더 크다는 점, 데스크톱 도구나 게임과 같은 최종 사용자 중심이고 기업에서 개발한 소프트웨어일수록 카피레프트의 성격이 강한 반면, 개발자가 주로 사용하거나 상업적 활용도가 높은 소프트웨어일수록 재량권이 더 크다는 점, 그리고 성숙기에 접어드는 프로젝트일수록 카피레프트 라이선스가 덜 발견된다는 점을 보여주었다. 유사하게 Stewart, Ammeter and Maruping (2006)은 프레쉬미트(Freshmeat.net)의 자료로 제한적인 라이선스를 사용하는 프로젝트보다 덜 제한적인 라이선스를 사용하는 프로젝트가 더 오랫동안 사용자가 많음을 보였고, Fershtman and Gandal (2007)은 소스포지의 자료에서 덜 제한적인 라이선스의 개발자가 상대적으로 2.9배 더 많은 코드를 공개함을 확인했다. 이와 반대로 Colazo, Fang and Neufeld (2005)는 소스포지에서 진행 중인 62개 프로젝트의 평균 3년간 개발 현황을 관찰한 결과, 제한적인 라이선스를 채택한 프로젝트일수록 개발자의 참여도와 생산성이 높다는 점을 보여주었다.

마지막으로 OSS 라이선스는 새로운 방식으로 혁신에 기여할 수 있을까? 여기서 중요한 요소는 기술 혁신 또는 지식 생산이 누적적이고 순차적(cumulative and sequential)인 성격을 갖는다는 전제이다. 만약 기술이 그러한 특징을 갖고 있다면 각각의 혁신이 이전의 혁신을 토대로 할 수 밖에 없기 때문에 각 단계에서 비용을 더 높여가는 방식은 그렇지 않은 방식 보다 상대적으로 불리하게 된다. Polanski (2007)는 기술적으로 개발 단계가 늘어날수록 그리고 단계가 늘어날수록 보수가 증가한다면, PS 방식에 비해 OSS 방식이 더 적합할 수 있음을 보이고 있다. OSS 방식의 행위자는 비용을 들이지 않고 커뮤니티에서 집단적으로 만든 지식을 사용할 수 있는 장점을 누리지만 각 단계에서 자신의 개발 비용을 지불해야 하고, PS 방식의 행위자는 이전 단계에서 다른 행위자가 개발한 지식을 활용하는 비용과 자신의

개발 비용을 지불해야 하는 대신 자신의 개발 성과를 판매할 수 있는 이득을 누린다. 따라서 지식 생산자의 협상력이 떨어진다면, PS 방식에서는 생산 단계가 늘어날수록 또 단계에 대한 보수 증가가 있다면, 협상 비용으로 인한 손해가 더 커지기 때문이다. 한편, Maurer and Scotchmer (2006)는 OSS가 지적재산권의 새로운 패러다임이 될 수 있다는 주장을 폈다. 오픈소스 방식은 자신뿐만 아니라 다른 개발자들이 코드를 공개할 것이라는 기대에 의존하는데, GPL은 로열티가 없는 상태에서 지식 교환을 가능하게 한다. GPL은 최초 개발자 이후의 무임승차 욕구를 제한할 수 있고, 만약 이를 어기는 경우 로열티를 부과할 수 있는, 즉 처벌이 가능한 제도이다. 따라서 GPL을 회피하고자 한다면 처음부터 코드를 다시 작성해야할 수 있고, 그러한 비용이 높으면 GPL을 받아들이는 것이 유리하게 된다. Bessen and Maskin (2009)은 순차적인 성격에 보충적 (complimentary)인 특징을 더했다. 어떤 기술이 보충성을 갖고 있다는 것은 잠재적인 혁신가가 새로운 방향을 제시할 가능성이 항상 열려 있음을 의미한다. 특허는 이러한 보충성을 가로막는 장애가 될 수 있으므로, 특허에 의한 지식의 보호가 사회의 후생을 증진시킨다는 보장은 없게 된다. 나아가 발명가의 기대 이윤 또한 경쟁과 모방을 통해 증대될 수 있음을 지적하고 있다.

라이선스의 특징, 특히 소스코드의 재배포에 관한 규제는 추가 개발자의 배타적 이득, 추가 개발자의 참여, 나아가 최초 개발자의 공개 여부 결정 등에 영향을 줄 수 있다. 최초의 개발자가 소스코드를 공개한 후, 이를 토대로 이후의 다른 개발자가 기능을 개선하거나 추가했을 때, 라이선스는 이를 공개할 범위와 의무를 정한다. 가장 엄격한 라이선스는 추가한 코드를 동일한 라이선스 하에서 모두 공개하도록 강제한다. 반면 관용적인 라이선스는 추가 개발자가 최초 저작권자를 명기할 의무는 있지만 자신이 추가한 코드를 공개할지 여부는 직접 결정할 수 있도록 한다. 관용적인 라이선스는 추가 개발자가 자신의 코드로부터 얻을 수 있는 배타적 이득을 보장하여 추가 개발자의 참여를 유도할 수 있지만, 모든 추가 개발자가 이렇게 행동한다면 최초 개발자의 노력에 무임승차하는 셈이다. 이것이 예상된다면 최초 개발자는 오히려 처음부터 소스코드를 공개하지 않을 수도 있다.

이런 면에서 우리의 관심사는 어떤 조건에서 개발자는 소스코드의 공개와 비공개 중 공개를 선택할 것인가, 소스코드를 공개한다면 어떤 조건을 붙일 것인가, 그리고 오픈소스 라이선스가 새로운 방식으로 혁신에 기여할 수 있는가 등이다. 앞에서

본 것처럼, OSS가 갖는 기술적 특성, 즉 모듈화, 순차성, 보충성이 소스코드의 공개여부와 라이선스의 선택 및 그 결과에 영향을 줄 수 있다.

현재까지 OSS 라이선스는 새로 덧붙여진 소스 코드의 공개를 유인하는 기능을 효과적으로 수행해왔다. 하지만, 기업의 참여가 활발해지고 덜 제한적인 라이선스가 확산되는 현상을 감안할 때, 앞으로도 현재의 라이선스 구조가 효과적이라라는 보장은 없다. 그렇다면 현재의 현상 변화에 따라 소스 코드의 공개를 유지할 수 있도록 하는 라이선스의 조건을 연구하는 것은 흥미로운 연구 방향이 될 것이다.

나아가 오픈소스 방식이 소프트웨어에 국한되지 않고 다른 분야에서도 활용되기 위해서는 일반화된 관점으로서 지속적인 지식 생산을 가능하게 하는 제도적 기능을 할 수 있는 라이선스의 내용이 무엇인가로 연구 방향을 확장시킬 필요가 있다. 그리고 라이선스 외의 저작권 제도도 OSS 개발에 영향을 줄 수 있음도 고려되어야 한다. 추가된 코드 또는 원래의 코드가 라이선스를 위반하여 배타적 이득을 위해 사용되었는지 확인하는 가장 일반적인 과정은 역설계(reverse engineering)이다. 그런데 저작권법 등 관련 제도가 역설계를 허용하지 않는다면, 이는 오픈소스 발전에 제약요인으로 작용하게 될 것이다.

IV. 오픈소스 소프트웨어와 정부 정책

OSS는 사용자가 필요에 따라 자유롭게 소프트웨어를 변경하고 적응시킬 수 있고, 최초의 구매 비용이 무료일 수 있으므로 정부 또는 공공기관의 정보통신서비스 구축에 매력적인 대안이다. 정부의 직간접적인 OSS 지원 목적은 경제, 기술, 정치적 측면으로 나눌 수 있다. 정부 차원의 OSS 관련 정책은 경제적으로 행정 기관의 소프트웨어 관련 지출 절감, 공공재 생산 촉진, 시장 경쟁 활성화, 소프트웨어 산업 진흥의 목적을 갖는다. 기술적으로는 호환성 확대, 보안 강화, 사용 편리성 증대, 응용 가능성 제고가 중요한 목표가 된다. 정치적으로는 민주주의 발전, 미국 의존 약화, 지적 재산권 침해에 대한 대안적인 접근의 의미가 있다(Dravis, 2003).

행정 기관은 소프트웨어의 직접 구매자로서 OSS를 채택하여 비용을 줄일 수 있다. PS와 달리 OSS 사용의 라이선스 비용을 직접 지불하지 않아도 되기 때문이다. 하지만 실제 운영 과정 전체에 소요되는 비용과 기존의 PS에서 신규 OSS로 전환하기 위한 비용을 포함한 총보유비용(Total Cost of Ownership)을 고려할 필요가 있

다. 전환 비용에는 기술적 호환성과 사용자 교육이 반드시 포함되어야 한다. 정부의 OSS에 대한 지원은 경쟁 정책 및 산업 정책이 될 수 있다. OSS의 도입으로 인해 소프트웨어 산업에서 선택의 기회가 많아지고 경쟁이 촉진될 수 있고, 현지화 과정을 통해 소프트웨어 엔지니어의 능력을 증진시켜 이들은 인적 자원이 될 수 있기 때문이다. 또 OSS는 기업이 소프트웨어를 처음부터 제작해야하는 비용을 줄일 수 있도록 함으로써, 새로운 기업이 나타날 수 있는 기회를 늘릴 수 있다. 또 불법 복제를 막기 위한 감시 및 처벌의 비용을 들이는 대신, OSS 사용을 장려함으로써 불법복제를 회피할 수도 있다.

한편 OSS는 정부 서비스의 지속성과 보안 문제에 대한 해결책이 될 수 있다. 정부가 특정 기업의 PS를 채택하면 해당 기업의 프로그램을 지속적으로 사용해야만 하는 위험이 있다. 만약 소프트웨어를 공급하는 기업이나 소프트웨어 자체를 교체하고자 할 때 전환 비용이 크게 발생한다면, 행정 기관은 소프트웨어 사용의 고착 문제를 겪게 된다. OSS는 코드가 공개되어 있기 때문에 정부의 기술적 자립성을 높일 수 있는 대안이다. 보안 또한 중요한 문제이다. 사용자는 PS의 코드를 확인할 수 없으므로 소프트웨어에서 어떤 일이 벌어지고 있는지 정확히 확인할 방법이 없다. OSS는 소스코드를 공개함으로써 작동 구조를 파악할 수 있다. OSS를 수정할 자유도 중요한 의미가 있다. 특히, 소프트웨어의 현지화를 직접 할 수 있다는 것이 큰 장점이 된다. 영어 이외의 언어를 사용하는 국가 또는 지역에서는 소프트웨어의 인터페이스를 해당 지역의 언어로 구현할 수 있어야 하는데, PS 기업은 상업적 이득이 떨어지는 지역을 위한 현지화 작업을 지원하지 않을 수 있다. 이런 경우 OSS는 좋은 대안이 된다.

OSS 정책은 정부의 정보통신기술 활용이나 소프트웨어 산업 진흥의 효과뿐만 아니라, 지적 재산을 사회에서 사용하고 공유하는 제도를 설계하는 문제와도 연결된다. 지적재산권의 범주에 포함되는 소프트웨어 특허와 저작권 관련 제도는 OSS 활용과 밀접한 관계를 갖는다. 소프트웨어의 특허를 인정하는 경우 OSS의 채택을 막거나 특정 부류의 OSS 사용을 금지시킴으로써 OSS의 효과를 제한할 수 있다. 소비자보호법 역시 동일한 영향을 미칠 수 있다.

전 세계의 많은 중앙 또는 지방 정부는 앞에서 열거한 다양한 정책적 목적을 달성하기 위해 OSS를 활용하고자 했다. 중앙 및 지역 정부의 구체적인 OSS 정책은 Dravis(2003)와 Lewis(2010)가 정리한 목록을 참고할 만하다. 이후에서는 정부의

정책 수단을 직접 구매 또는 간접 지원, 제도적 환경 조성, 저개발국의 발전 전략이라는 세 가지 기준으로 정리하고자 한다.

정부의 기본적인 정책 수단은 공공기관, 학교, 대학 등의 OSS 구입을 의무화하도록 하는 것이다. 특히 저개발국가는 2000년대 초부터 적극적으로 정부 차원의 OSS 구입에 나섰다. 가장 적극적인 지역은 남아메리카로 브라질의 경우 2000년 페르남부코(Pernambuco) 지방 정부가 OSS 구입을 의무화시켰고, 2003년 중앙정부는 모든 개인용 컴퓨터의 80%를 리눅스로 전환시킴으로써 개인용 컴퓨터 구입 비용을 낮추겠다는 계획을 발표했다. 베네수엘라는 2006년 소프트웨어 자유법(Software Libre Law)을 통과시켜 모든 정부기관의 소프트웨어를 OSS로 의무 전환시키고자 했다. 이 과정에는 OSS 개발 협동조합을 활성화 시키는 방안도 포함되었다. 2008년부터 2010년까지 에콰도르, 도미니카, 쿠바, 파라과이, 트리니다드 토바고, 볼리비아, 콜롬비아가 OSS의 의무 사용 또는 지원 법안을 통과시켰다. 아시아에서는 2001년 말레이시아가 전자 조달 시스템, 재무부 및 연구 기관 등의 OSS 사용에 관한 정책을 시작하였으며, 2002년부터 민간 부문과 공동으로 가정에서 사용할 수 있는 OSS 기반의 컴퓨터 보급을 시작했다. 2010년 현재 말레이시아 정부 기관의 OSS 채택률은 97%로 공표되었다.

선진국은 2010년대 이후 정부 차원의 OSS 도입에 관한 정책을 제시하고 있다. 프랑스는 광범위한 검토를 거쳐 2012년 9월 정부 차원의 OSS 이용에 관한 가이드라인 “아이로 회람(Ayrault Circular)”을 공표하고, 2013년 8월부터 중앙 정부 부처별로 구체적인 OSS 도입 계획을 밝혔다. 프랑스 교육부는 23,000 대의 서버가 우분투 리눅스를 기반으로 하고 있으며, 오픈오피스도 광범위하게 사용 중이라고 밝혔다. 영국과 호주 정부도 정부 구매 원칙을 공표했고, EU 차원의 가이드라인도 제안된 상황이다. 미국의 경우 국방부, 제대군인부와 같은 정부 기관과 나사 외 주요 연구 기관의 OSS 이용 사례가 알려져 있다. 한국 정부의 OSS 관련 정책 현황은 정보통신산업진흥원에서 발간하는 공개소프트웨어 백서에서 확인 가능하다.

OSS를 사용하는 경우 세금 혜택을 주거나 하드웨어를 상대적으로 낮은 가격에 구입할 수 있도록 하는 보조금 정책도 있다. 싱가포르의 리눅스를 사용하는 기업에 세금을 감면하고 있으며, 독일은 리눅스가 설치된 아이비엠 장비를 할인된 가격으로 구입할 수 있도록 했다. 이 보다 더 간접적인 방식은 정부가 OSS 사용 캠페인을 벌이는 것이다. 미국의 경우 2002년 북미시시피(North Mississippi)의 지역 교육 컨

소기업이 “배움의 자유(Freedom to Learn)”라는 시범 사업에서 공립 학교의 OSS 사용을 권장했다.

정부는 직접 구매 또는 간접적인 지원 이외에 지적재산권 제도를 통해서 OSS에 영향을 미칠 수도 있다. 상업적 소프트웨어의 사용을 위해 로열티를 지불해야 하는 것이 일반적이다. 앞서 말한 것처럼 OSS는 배포 비용 이상의 비용을 받을 수 없다는 원칙을 갖고 있으므로, 이러한 로열티 지급을 피할 수 있는 방법이 된다. OSS는 집단적 개발이라는 속성상 소프트웨어에 대한 특허를 신청하거나 인정받기 어렵다. 반대로 상업적 소프트웨어를 개발하는 기업은 자사의 특허를 이용하여 OSS에 대한 소송을 제기하기도 한다. 다른 한편, III.3절에서 본 것처럼 OSS는 배타적인 경제적 이득은 제한하지만, 소스코드의 공개를 유지하기 위해 라이선스를 적극적으로 활용한다. 라이선스와 관련된 가장 큰 지적재산권 문제는 파생 저작물의 권리에 관한 것이다. GPL 라이선스는 2차 저작물에 대해 동일 라이선스를 적용할 것을 요구하고 있다. 그런데, 국가마다 파생 저작물의 범위가 다르기 때문에 문제가 될 수 있다. OSS는 기술적으로 역설계 방법을 통해 자신의 소스코드에 대한 무단 사용 여부를 확인한다. 대부분의 저작권법은 소프트웨어의 호환성과 상호운영성을 위해 역설계를 허용하고 있다. 하지만, 저작권법에서 역설계를 금지시킨다면 이는 PS에만 유리한 조건이 될 것이므로 OSS 개발에 큰 문제가 될 수 있다.

정부가 OSS 개발에 영향을 줄 수 있는 또 다른 제도는 개방형 표준(open standards)을 정하는 것이다. 이는 OSS 사용의 장벽을 제거함으로써 OSS 사용을 확대시킬 수 있는 정책이다. 예를 들어, 정부는 정부의 전자 문서가 중앙 정부와 지방 정부 간, 정부 부처 간, 공무원과 일반 국민 간의 국가 내 모든 행위자에서 동일한 비용 또는 아무런 비용이 들지 않고서도 열람 및 작성이 가능하도록 만들어야 하는 의무를 갖고 있다. 전자화된 공문서의 열람과 작성이 특정 소프트웨어만 사용하도록 제한을 받아서는 안 된다. 따라서 정부는 전자 공문서의 공개 표준을 수립하는 것이 바람직하다. 공문서의 공개 표준이 만들어지면, 전자 문서의 호환성이 보장되므로, 표준을 준수하면서 그 외의 기능으로 경쟁하는 다양한 소프트웨어가 생산될 수 있다. 그럼으로써 정부뿐만 아니라 가장 많은 수의 국민이 가장 낮은 가격으로 사용할 수 있는 전자 문서용 소프트웨어를 보급할 수 있게 될 것이다. 2014년 7월 영국 정부는 열람용으로 PDF/A 또는 HTML, 공유 및 협업용으로 ODF를 공개문서 표준으로 발표했다. 동년 8월 우리나라는 일부 공문서에 한해

ODF 지원을 발표했다. 공개 표준은 문서에만 한정되는 것은 아니다. 인터넷 서비스의 통신 규약 TCP/IP, HTTP 등과 문자 표현과 관련된 UNICODE는 이미 사용 중인 대표적인 개방형 표준이다.

정부의 OSS 직접 사용 또는 사용 강요, OSS 사용에 대한 보조금 지급, OSS 사용의 홍보와 같은 정책의 효과에 대해 이론적으로 접근한 연구는 조건에 따라 상이한 결과를 보이고 있다. 정책이 사회 후생에 미치는 영향은 네트워크 효과의 유무나 그 정도에 따라 다르게 나타났는데, 네트워크 효과를 고려하지 않은 경우 공공기관의 OSS 사용 강요는 PS의 가격을 인상시키는 결과를 가져와 소비자 후생을 감소시킨다. 네트워크 효과가 강하게 있으면 모든 정책은 소비자 후생을 증대시킨다(Comino and Manenti, 2005; 안일태, 2005).

중앙 또는 지방 정부의 OSS 도입 정책의 결과를 확인할 수 있는 몇 개의 사례 연구도 있다. 2004년 13개 유럽 국가의 955개 공공기관을 대상으로 한 조사에서는 유의미한 OSS 보급률이 나타나지 않았는데, 이는 OSS 사용을 기피해서가 아니라 기관에서 OSS 사용을 인지하지 못하는 경우가 많았기 때문이다. 상당수의 응답이 리눅스, 아파치, 미에스큐엘은 사용 중이지만 OSS는 사용하고 있지 않다고 했다. 브라질 연방 정부, 프랑스 의회, 독일 의회, 매사추세츠 주, 밴쿠버 시에 대한 사례 연구에 따르면, 개방성을 측정하기 어려웠고 정보통신기술 관련 의사 결정에서 개방 또는 공개는 매우 다양한 의미로 모호하게 사용되었다. 또, OSS와 PS를 혼용하는 경우가 일반적이었으며, 상호호환성과 표준이 성공적인 정보통신기술 생태계를 구축하는데 핵심적인 역할을 한 것으로 나타났다. 이는 OSS로의 전환에 성공하기 위해서는 정부의 소프트웨어 구매 과정이 투명해지는 것보다 정보의 사용 가능성과 정보 그 자체의 투명한 공개에 더 신경을 써야함을 의미한다. 다른 연구에 따르면, 캐나다는 직접 구매보다는 개방형 표준을 정하는 정책을 택했는데, 예상과 다르게 소프트웨어 조달에서 소프트웨어 브랜드가 여전히 큰 영향을 행사하고 있는 것으로 나타났다. 그리고 캐나다 정부의 조달 과정은 간접적으로 PS에 유리했다. 캐나다 정부 기관은 소프트웨어 계약 입찰을 2단계로 하는데, 1단계에서는 라이선스 비용을 제시하고 2단계에서는 사후 지원 비용을 제시하는 것이 일반적이다. 그런데 대부분의 OSS 관련 기업은 소프트웨어 라이선스를 판매하지 않고 사후 지원 서비스에서 수익을 얻는다. 따라서 이 기업은 1단계 입찰에 참여조차 할 수 없는 문제가 있었다(Henkel and Lakhani, 2011; Thakur, 2012).

앞서 열거한 비용, 인력 양성 등의 장점으로 인해 OSS는 저개발국이 유용하게 사용할 수 있는 정보통신 기술 발전 전략이 될 수 있기도 하다. 상대적으로 선진국에 비해 저개발국의 인프라가 부족하다는 점을 고려하면, 저개발국의 OSS 도입은 기술 영역에 국한되지 않고, 근본적인 차원에서 사회 전반의 정보 생성, 유통, 유지에 관한 대안이 될 수 있다. 저개발국의 OSS 사용은 세계 수준에서 경쟁력이 있는 전자 정부 구축, 정보통신 산업 육성, 교육 부문 활용, 인적 자원 양성 등의 다양한 차원에서 접근할 필요가 있다(UNCTAD, 2012).

저개발국가 정부의 OSS 이용에 관한 사례는 발전 전략으로서의 성격을 잘 보여준다. 1999년 마드리드 폴리테크닉 대학(Polytechnic University of Madrid)과 국경 없는 엔지니어회(Engineers without Borders)를 중심으로 설립된 라틴 아메리카 의료 연합(Enlace Hispano Americano de Salud)은 페루, 콜롬비아, 에콰도르, 쿠바의 일부 병원과 협력하여 정보통신 기술을 이용한 원격지 공공의료 지원 사업을 시작했다. 원격지의 진료 시설을 연결하기 위한 통신 네트워크 구축에 리눅스를 활용하였으며, 의료 정보 관리를 위한 웹과 자바 기반의 어플리케이션인 오픈 엠알에스(OpenMRS)가 개발되었다. 스리랑카는 2004년 쓰나미 피해 이후 재해관리 시스템을 OSS 기반으로 구축하기 시작했다. 사하나(Sahana)로 명명된 이 시스템은 웹 기반의 관리 어플리케이션으로 실종자 및 피해자 관리, 구호 조직의 관리 및 행정 업무, 피난민 캠프 관리, 구호 물자 배분 관리를 목적으로 하며, 파키스탄, 필리핀, 인도네시아 등에도 보급되었다. 6명의 상근 개발자와 동시에 커뮤니티의 지원을 받았으며, 인도적 목적을 갖고 있는 다른 OSS 프로젝트의 시발점이 되었다. 브라질은 정부 지원으로 지리 정보 시스템(Geographical Information Systems)을 OSS로 개발했다. 테라리브(TerraLib)로 명명된 이 프로젝트는 오픈소스 라이브러리와 관련 어플리케이션 개발을 목표로 했다. 국립 우주 연구소와 리우 데 자네이로 가톨릭 대학교가 주축인 프로젝트였다. 이 데이터를 사용하는 민간 기업은 2006년 초 기준으로 10개였으며, 시장점유율은 약 15%로 추정되었다. OSS를 이용한 정보통신기술 인력 양성을 위한 프로젝트는 정보통신기술 혁신(ict@innovation) 사업이 대표적이다. 독일 연방정부의 지원을 받는 기즈(GIZ)와 FOSSFA 재단(Free Software and Open Source Foundation for Africa)이 시작한 이 프로그램은 2012년 현재 사하라 이남의 15개 국가에서 운영 중이고, 200명 이상의 교수진이 OSS를 활용한 사업 모델을 600명 이상의 기업가, 학생 등에게 강의하고 있다(Sowe, Parayil and

Sunami, 2012; UNCTAD, 2012).

앞서 살펴보았듯이 비용 절감이라는 미시적인 목표부터 경제 발전이라는 거시적인 것까지 다양한 방향으로 OSS 정책이 수립되었다. 그러나 현재까지 정부의 OSS 관련 정책을 소개 또는 정리하는 수준의 연구가 진행되었다고 볼 수 있으며, 경제에 미치는 효과에 대해서는 자세히 연구되지 않았다. 개발자의 참여를 촉진시키고 이들이 공유된 지식을 활용하여 새로운 사업을 시작할 수 있도록 지원하는 제도에 관한 연구와 정부의 OSS 도입이 소프트웨어 산업이나 경제 성장에 미치는 영향에 대한 연구가 특히 부족하다고 할 수 있다. 라이선스에 대한 연구와 함께 이러한 방향의 연구는 앞으로 오픈소스 방식의 기술 개발이 소프트웨어 이외의 다른 분야에 어떻게 응용될 수 있을가에 중요한 디딤돌이 될 것이다.

V. 결론을 대신하여: 소프트웨어를 넘어선 오픈소스는 가능한가

OSS는 광범위하게 사용되고 있다. 소프트웨어 산업으로만 보면 리눅스는 셋톱박스, 공유기, 슈퍼컴퓨터 등의 임베디드 OS 분야에서 시장 점유율이 높다. 영국 런던과 남아프리카공화국 요하네스버그의 주식시장은 리눅스 기반의 시스템을 채택하기도 했다. 최근에 OSS는 산업적으로 중요한 역할을 차지할 뿐만 아니라 핵심 개념을 다른 분야에도 적용하는 단계로 나가고 있다.

OSS를 자발적인 협력에 의한 지식 생산이라고 생각한다면, 위키피디아(Wikipedia)는 좋은 사례가 될 것이다. 자유롭게 항목을 수정하고 편집하는 온라인 백과사전으로, 지미 웨일스(Jimmy Wales)가 2001년 시작한 이래로 2014년 현재 257개 언어의 1,700만개 이상의 기사를 담고 있으며, 접속자는 3억 6,500만 명에 달한다. 이외에도 커넥티드 카, 건축 설계 및 시공, 브라질, 스위스, 스페인, 에스토니아, 인도, 캐나다, 호주의 전자 투표 시스템에 이르기까지 오픈소스 방식은 전통적인 생산 영역과 다양한 영역에서 시도되고 있다.

OSS는 분산된 형태의 생산 방식과 더불어, 지적재산권을 배타적 사용이 아니라 공공의 사용을 위해 활용한다는 점, 생산에 참여한 사람이 직접적인 이득을 볼 수 있는 방법을 제한하였지만 동시에 그 가능성은 열어 놓았다는 점 등에서 소프트웨어에 국한되지 않고, 새로운 시대의 새로운 유인 체계로 주목 받을 만하다. 특히 앞으로 오픈소스 방식을 활용할 수 있는 가장 흥미로운 분야는 생명공학과 3D 프린

팅일 것이다.

생명공학에서는 지적재산권에 의한 보호가 강화되면서 새로운 기술의 발달이 저해되고 연구예산의 제약이 커짐에 따라, 저작권의 제약을 받지 않으며 많은 사람의 참여를 이끌어 낼 수 있는 공개 방식으로의 전환이 시도되고 있다. 대표적인 분야로 암 연구, 씨앗 개발 등이 있다. 생명공학 기술이 주요 대기업의 투자에 의해 지배되는 경향이 있어, 산업 내 경쟁을 촉진시키기 위해서는 오픈소스 방식이 적용될 필요가 있다는 주장도 제기되고 있다(Hope, 2008).

3D 프린터는 소프트웨어의 지시에 따라 재료를 적층하거나 절삭함으로써 원하는 형태의 모형을 만들 수 있다. 3D 프린터는 더 작은 수의 상품을 경제적으로 생산할 수 있고, 더 적은 노동을 투입하고 더 탄력적으로 생산할 수 있으며, 온라인과 결합된 제조업이 가능하다는 면에서 “제3의 산업혁명(a third industrial revolution)”이 될 수도 있다. 최근에는 3D 프린터 하드웨어와 결과물을 제작할 수 있는 소프트웨어 모두 오픈소스로 배포하는 시도가 나타나고 있다(Pearce, 2012).

이 두 분야에서 오픈소스 방식이 작동한다면, 새로운 방식으로 대규모의 협력적인 지식 생산과 물리적 상품의 생산이 가능함을 의미한다. 하지만 Lerner and Tirole (2005b)은 OSS 모델이 생명공학을 비롯하여 다른 산업에서 응용되기 어렵다고 평가했다. OSS가 성공한 기술적 이유는 모듈화에 있으며, 따라서 소프트웨어 이외의 분야에서도 성과를 얻기 위해서도 독립적인 분할 생산이 가능한 상품이어야 한다. 이와 더불어 이들을 결합하는 비용도 낮아야만 한다. 그런데 이러한 기술적 특성을 만족시킬 수 있는 산업은 소프트웨어 외에 찾아보기 힘들다. 또 산업이 발전하기 위해서는 개인의 기여와 비자본투자가 아닌 대규모 집단과 상당 수준의 자본 투자를 필요로 하며, 대중을 상대로 하는 시장일수록 상대방부터의 인정과 자기 만족을 달성하기 어렵다는 문제가 있다.

그렇지만 이러한 예상은 모듈화라는 기술적 특성을 고정적인 것으로 바라보고 있다. 위키피디아의 예에서처럼 모듈화가 가능한 분야가 우리의 예상보다 더 많을 수 있으며, 다른 한편으로 자동차 산업이나 건축의 경우에서처럼 기술 발전에 의해 이전에 모듈화가 되지 않았던 분야에서 모듈화가 나타날 가능성을 배제할 수 없다. 또 OSS의 사례에서 보듯 인정과 자기 만족이라는 내재적 가치뿐만 아니라 외재적 가치도 OSS 개발에 참여하도록 하는 중요한 요소이다. 오픈소스 방식이 다른 산업에 줄 수 있는 의미는 자기 만족을 누리하고자하는 취미가의 자발적인 참여보다, 특

허와 저작권으로 소프트웨어 복제를 제한하여 기술 혁신의 이득을 보장하는 전통적인 방법과 다른 유인체계를 통해 다양한 이해관계자를 혁신 과정에 끌어들이는 방식에 있다.

Johnson (2012)의 지적대로 “OSS를 미스터리라고 주장하는 것은 잘못되었을 뿐만 아니라 이 운동의 더 중요한 역할과 의미로부터 주의를 분산”시킬 뿐이다. 오픈 소스 방식의 생산 양식이 다른 산업에 줄 수 있는 의미를 찾는 것이 앞으로 OSS에 대한 연구의 생산적인 방향이 될 것이다. 여기에는 혁신을 추동하고 커뮤니티를 조직하는데 대한 직접적인 금전적 보상의 역할에 강하게 집착하지 않고, 기업을 포함하여 생산을 조직하는 새로운 방법, 특히 지적재산권이 혁신에 기여할 수 있는 새로운 방법을 연구하는 것이 포함되어야 한다.

■ 참 고 문 헌

1. 송우일, 『꾸준히, 자유롭게, 즐겁게 : 한국 오픈 소스 개발자들 이야기』, 인사이트, 2013.
(Translated in English) Song, W., *Steadily, Freely, Enjoyably: Interview of Korean Open Source Developers*, Insight, 2013.
2. 안일태, “오픈소스 소프트웨어에 대한 정부지원정책의 경제적 효과,” 『계량경제학보』, 제16권 제3호, 2005, pp. 51-76.
(Translated in English) Ahn, I., “Economic Effects of Government’s Supports for Open Source Software,” *Journal of Economic Theory and Econometrics*, Vol. 16, No. 3, 2005, pp. 51-76.
3. Aker, A., “Voluntarism and the Fruits of Collaboration: The IBM User Group, Share,” *Technology and Culture*, Vol. 42, No. 4, 2001, pp. 710-736.
4. Arner, P., “SHARE-A Eulogy to Cooperative Effort,” *IEEE Annals of the History of Computing*, Vol. 2, No. 2, 1980, pp. 122-129.
5. Baldwin, C. Y. and K. B. Clark, “The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model?,” *Management Science*, Vol. 52, No. 7, 2006, pp. 1116-1127.
6. Baytiyeh, H. and J. Pfaffman, “Open Source Software: A Community of Altruists,” *Computers in Human Behavior*, Vol. 26, No. 6, 2010, pp. 1345-1354.

7. Benkler, Y., "Coase's Penguin, or, Linux and The Nature of the Firm," *The Yale Law Journal*, Vol. 112, No. 3, 2002, pp.1-42.
8. Bergquist, M. and J. Ljungberg, "The Power of Gifts: Organizing Social Relationships in Open Source Communities," *Information Systems Journal*, Vol. 11, No. 4, 2001, pp. 305-320.
9. Bessen, J., "Open Source Software: Free Provision of Complex Public Goods," in J. Bitzer and P. J. H. Schröder, eds., *The Economics of Open Source Software Development*, Elsevier B. V., 2006, pp.54-81.
10. _____ and E. Maskin, "Sequential Innovation, Patents, and Imitation," *RAND Journal of Economics*, Vol. 40, No. 4, 2009, pp.611-635.
11. Bitzer, J. and I. Geishecker, "Who Contributes Voluntarily to OSS? An Investigation Among German IT Employees," *Research Policy*, Vol. 39, No. 1, 2010, pp.165-172.
12. _____, and P. J. H. Schröder, "Open Source Software, Competition and Innovation," *Industry and Innovation*, Vol. 14, No. 5, 2007, pp.461-476.
13. Bonaccorsi, A. and C. Rossi, "Why Open Source Software can Succeed," *Research Policy*, Vol. 32, No. 7, 2003, pp.1243-1258.
14. _____ and _____, "Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business," *Knowledge, Technology & Policy*, Vol. 18, No. 4, 2006, pp.40-64.
15. Campbell-Kelly, M. and W. Aspray, *Computer: A History of the Information Machine*, N. Y. : Basic Books, 1996.
16. _____ and D. Garcia-Swartz, "The Move to the Middle: Convergence of the Open-Source and Proprietary Software Industries," *International Journal of the Economics of Business*, Vol. 17, No. 2, 2010, pp.223-252.
17. Casadesus-Masanell, R. and P. Ghemawat, "Dynamic Mixed Duopoly: A Model Motivated by Linux vs. Windows," *Management Science*, Vol. 52, No. 7, 2006, pp.1072-1084.
18. Colazo, J. A., Y. Fang, and D. J. Neufeld, "Development Success in Open Source Software Projects: Exploring the Impact of Copylefted Licenses," *AMCIS 2005 Proceedings*, 2005, Paper 432.
19. Colombo, M. G., E. Piva, and C. R. Lamastra, "Open Innovation and Within-industry Diversification in Small and Medium Enterprises: The Case of Open Source Software Firms," *Research Policy*, Vol. 43, No. 5, 2014, pp.891-902.
20. Comino, S. and F. M. Manenti, "Government Policies Supporting Open Source Software for the Mass Market," *Review of Industrial Organization*, Vol. 26 No. 2, 2005, pp. 217-240.
21. Corbet, J., G. Kroah-Hartman, and A. McPherson, *Linux Kernel Development: How Fast It is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It*, Linux Foundation, 2013.
22. Crowston, K. and J. Howison, "Hierarchy and Centralization in Free and Open Source Software Team Communications," *Knowledge, Technology & Policy*, Vol. 18, No. 4,

- 2006, pp. 65-85.
23. Dahlander, L., "Penguin in a New Suit: A Tale of how de Novo Entrants Emerged to Harness Free and Open Source Software Communities," *Industrial and Corporate Change*, Vol. 16, No. 5, 2007, pp. 913-943.
24. D'Antoni, M. and M. A. Rossi, "Copyright vs. Copyleft Licensing and Software Development," Working Paper Series 510, Università Degli Studi Di Siena, 2007.
25. David, P. A. and J. S. Shapiro, "Community-based Production of Open-source Software: What do we Know about the Developers who Participate?," *Information Economics and Policy*, Vol. 20, No. 4, 2008, pp. 364-398.
26. de Laat, P. B., "Governance of Open Source Software: State of the Art," *Journal of Management and Governance*, Vol. 11, No. 2, 2007, pp. 165-177.
27. Di Gaetano, L., "A Model of Corporate Donations to Open Source Under Hardware-software Complementarity," *Industrial and Corporate Change*, Vol. 24, No. 1, 2015, pp. 163-190.
28. Dravis, P., "Open Source Software : Perspectives for Development," Working Paper 27680, World Bank, 2003.
29. Economides, N. and E. Katsamakas, "Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry," *Management Science*, Vol. 52, No. 7, 2006, pp. 1057-1071.
30. Fang, Y. and D. Neufeld, "Understanding Sustained Participation in Open Source Software Projects," *Journal of Management Information Systems*, Vol. 25, No. 4, 2009, pp. 9-50.
31. Fershtman, C. and N. Gandal, "Open Source Software: Motivation and Restrictive Licensing," *International Economics and Economic Policy*, Vol. 4, No. 2, 2007, pp. 209-225.
32. Franck, E. and C. Jungwirth, "Reconciling Rent-Seekers and Donators - The Governance Structure of Open Source," *Journal of Management and Governance*, Vol. 7, No. 4, 2003, pp. 401-421.
33. Free Software Foundation, "The Free Software Definition," <http://www.gnu.org/philosophy/free-sw.html>, 2010, last accessed on Nov. 4, 2014.
34. Freeman, S., "The Material and Social Dynamics of Motivation: Contributions to Open Source Language Technology Development," *Science Studies*, Vol. 20, No. 2, 2007, pp. 55-77.
35. Gonzalez-Barahona, J. M., G. Robles, R. Andradras-Izquierdo, and R. A. Ghosh, "Geographic Origin of Libre Software Developers," *Information Economics and Policy*, Vol. 20, No. 4s, 2008, pp. 356-363.
36. Harison, E. and H. Koski, "Applying Open Innovation in Business Strategies: Evidence from Finnish Software Firms," *Research Policy*, Vol. 39, No. 3, 2010, pp. 351-359.
37. Hars, A. and S. Ou, "Working for Free? Motivations of Participating in Open Source Projects," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences 2001*, IEEE, 2002, pp. 1-9.
38. Haruvy, E., A. Prasad, and S. Sethi, "Competition with Open Source as a Public Good,"

- Journal of Industrial and Management Optimization*, Vol. 4, No. 1, 2008, pp.199-211.
39. Henkel, J. and K. R. Lakhani, "The Trend Toward "Open" IT in the Public Sector: Motives, Choices, and Outcomes," Technical Report, SSRN, 2011.
 40. Hertel, G., S. Niedner, and S. Herrmann, "Motivation of Software Developers in Open Source Projects: An Internet-based Survey of Contributors to the Linux Kernel," *Research Policy*, Vol. 32, No. 7, 2003, pp.1159-1177.
 41. Hofmann, G., D. Riehle, C. Kolassa, and W. Mauerer, "A Dual Model of Open Source License Growth," in E. Petrinja, G. Succi, N. El Ioini and A. Sillitti, eds., *Open Source Software: Quality Verification: 9th IFIP WG 2.13 International Conference*, Springer Berlin Heidelberg, 2013, pp.245-256.
 42. Hope, J., *Biobazaar: The Open Source Revolution and Biotechnology*, Harvard University Press, 2008.
 43. Humphrey, W. S., "Software Unbundling: A Personal Perspective," *IEEE Annals of the History of Computing*, Vol. 24, No. 1, 2002, pp.59-63.
 44. Johnson, J. P., "Open Source Software," in M. Peitz and J. Waldfogel, eds., *The Oxford Handbook of The Digital Economy*, Oxford University Press, 2012, pp.460-485.
 45. Krishnamurthy, S., "On the Intrinsic and Extrinsic Motivation of Free/libre/open Source (FLOSS) Developers," *Knowledge, Technology & Policy*, Vol. 18, No. 4, 2006, pp.17-39.
 46. Lamastra, C. R., "Software Innovativeness. A Comparison between Proprietary and Free/Open Source Solutions Offered by Italian SMEs," *R&D Management*, Vol. 39, No. 2, 2009, pp.153-169.
 47. Landini, F., "Technology, Property Rights and Organizational Diversity in the Software Industry," *Structural Change and Economic Dynamics*, Vol. 23, No. 2, 2012, pp.137-150.
 48. Lee, T. B., "Why Libertarians Should Celebrate Free Software," *Techknowledge*, No. 105, Cato Institute, 2007.
 49. Lerner, J. and J. Tirole, "The Simple Economics of Open Source," *The Journal of Industrial Economics*, Vol. 50, No. 2, 2002, pp.197-234.
 50. _____ and _____, "The Economics of Technology Sharing: Open Source and Beyond," *Journal of Economic Perspectives*, Vol. 19, No. 2, 2005a, pp.99-120.
 51. _____ and _____, "The Scope of Open Source Licensing," *Journal of Law, Economics and Organization*, Vol. 21, No. 1, 2005b, pp.20-56.
 52. _____ and M. A. Satterthwaite, *The Comingled Code: Open Source and Economic Development*, MIT Press, 2010.
 53. Levy, S., *Hackers: Heroes of the Computer Revolution*, Penguin Books, 1984.
 54. Lewis, J. A., *Government Open Source Policies*, Center For Strategic and International Studies, 2010.
 55. Llanes, G. and R. de Elejalde, "Industry Equilibrium with Open-source and Proprietary Firms," *International Journal of Industrial Organization*, Vol. 31, No. 1, 2013, pp.36-49.
 56. Mapstone, R. and M. Bernstein, "The Founding of SHARE," *IEEE Annals of the History*

- of Computing*, Vol. 2, No. 2, 1980, pp.363-372.
57. Markus, M. L., "The Governance of Free/open Source Software Projects: Monolithic, Multidimensional, or Configurational?," *Journal of Management and Governance*, Vol. 11, No. 2, 2007, pp.151-163.
58. Maurer, S. M. and S. Scotchmer, "Open Source Software: The New Intellectual Property Paradigm," Working Paper 12148, National Bureau of Economic Research, 2006.
59. Moglen, E., "The dotCommunist Manifesto," <http://moglen.law.columbia.edu/publications/dcm.html>, 2003, last accessed on Jun. 17, 2014.
60. Mustonen, M., "When Does a Firm Support Substitute Open Source Programming?," *Journal of Economics & Management Strategy*, Vol. 14, No. 1, 2005, pp.121-139.
61. O'Mahony, S., "Guarding the Commons: How Community Managed Software Projects Protect Their Work," *Research Policy*, Vol. 32, No. 7, 2003, pp.1179-1198.
62. Open Source Initiative, "The Open Source Definition," <http://opensource.org/osd>, 1999, last accessed on Oct. 25, 2014.
63. Pearce, J. M., "Building Research Equipment with Free, Open-Source Hardware," *Science*, Vol. 337, No. 6100, 2012, pp.1303-1304.
64. Polanski, A., "Is The General Public Licence a Rational Choice?," *The Journal of Industrial Economics*, Vol. 55, No. 4, 2007, pp.691-714.
65. Raymond, E. S., "The Cathedral and the Bazaar," <http://catb.org/~esr/writings/cathedral-bazaar/>, 1998, last accessed on Nov. 4, 2014.
66. Reisinger, M., L. Ressler, R. Schmidtke, and T. P. Thomes, "Crowding-in of Complementary Contributions to Public Goods: Firm Investment into Open Source Software," *Journal of Economic Behavior & Organization*, Vol. 106, 2014, pp.78-94.
67. Scotchmer, S., "Openness, Open Source, and the Veil of Ignorance," *American Economic Review*, Vol. 100, No. 2, 2010, pp.165-171.
68. Sen, R., C. Subramaniam, and M. L. Nelson, "Determinants of the Choice of Open Source Software License," *Journal of Management Information Systems*, Vol. 25, No. 3, 2008, pp.207-239.
69. Sowe, S. K., G. Parayil, and A. Sunami eds., *Free and Open Source Software and Technology for Sustainable Development*, Tokyo: United Nations University, 2012.
70. Stallman, R., *Free Software, Free Society: Selected Essays of Richard M. Stallman*, 2nd ed., Free Software Foundation, 2010.
71. Stewart, K. J., A. P. Ammeter, and L. M. Maruping, "Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects," *Information Systems Research*, Vol. 17, No. 2, 2006, pp.126-144.
72. Thakur, D., "A Limited Revolution - The Distributional Consequences of Open Source Software in North America," *Technological Forecasting & Social Change*, Vol. 79, No. 2, 2012, pp.244-251.
73. UNCTAD, *Information Economy Report 2012 - The Software Industry and Developing Countries*, United Nations, 2012.
74. von Engelhardt, S. and A. Freytag, "Institutions, Culture, and Open Source," *Journal of*

Economic Behavior & Organization, Vol. 95, 2013, pp.90-110.

75. von Hippel, E and G. von Krogh, "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science," *Organization Science*, Vol. 14, No. 2, 2003, pp.209-223.
76. West, J. and S. O'Mahony, "The Role of Participation Architecture in Growing Sponsored Open Source Communities," *Industry & Innovation*, Vol. 15, No. 2, 2008, pp.145-168.

The Knowns and the Unknowns about Open Source Software: History, Industrial Organization and Policy Implications

Namhyung Lee* · Joonil Kim**

Abstract

Open source software is developed by voluntary developers who use publicly available source codes and distributed freely with licenses. Open source software can be regarded as a new way of production in terms that: it exploits intellectual property rights not for exclusive benefits but for public usage; it limits direct private profits but opens possibilities for indirect ones; and it takes advantage of dispersed production networks. This paper has two main objectives: we try to explain that open source way has started from the beginning of computers and survey related researches in economics after the renowned work of Lerner and Tirole (2002). In this paper, mainly we deal with developers' motivations, development processes and managements, market competition, licenses, policies and implications to other industries.

Key Words: history of software industry, open source software, surveys

JEL Classification: L1, L5, L8, N7, O2

Received: Feb. 12, 2015. Revised: Oct. 31, 2015. Accepted: Jan. 25, 2016.

* First Author, PhD candidate, Department of Economics, Korea University, 145 Anamro, Seongbuk-gu, Seoul 02481, Korea, Phone: +82-2-3290-2200, e-mail: marginalman@korea.ac.kr

** Corresponding Author, Research Professor, Department of Economics, Korea University, 145 Anamro, Seongbuk-gu, Seoul 02841, Korea, Phone: +82-2-3290-2717, e-mail: kjoonil@korea.ac.kr