

# 개방형 연구개발을 위한 공개소프트웨어 커뮤니티 거버넌스

김형채 인베숨 대표

## 1. 머리말

전통적으로 기술혁신은 우수한 인적자원을 자체적으로 확보하고 내부자원을 효율적으로 활용하는 방식을 연구하며 이루어져 왔다. 이런 방식은 아이디어를 발굴하는 것에서부터 기초연구, 제품개발, 사업화에 이르는 모든 기술혁신 과정을 기업 내부에서 독자적으로 수행하는 것을 의미한다. 그러나 기술이 복잡해지고 제품 수요가 다양해지며 시장경제의 글로벌화가 가속화되자 기술혁신의 비용이 급증하고 있다. 이에 조직의 연구개발 생산성을 제고하고자 기술혁신 과정에서 외부와 협력하는 현상이 확대[2]되고 있다.

공개 소프트웨어는 이러한 개방형 혁신의 핵심 기술로 인공지능(AI), 사물인터넷(IoT), 클라우드 빅데이터 같은 다양한 분야에서 4차 산업혁명과 지능정보사회의 핵심으로 자리 잡았다. 구글, 페이스북 같은 글로벌 대기업도 개방형 혁신 활동을 중요하게 여기며 이를 위하여 자사의 기술을 외부에 공개한다. 또한 공개 소프트웨어 커뮤니티를 중심으로 핵심 기술을 주도하기 위해 노력한다. 본고에서는 이러한 공개 소프트웨어 커뮤니티의 성장과정을 살펴보고 이상적인 공개 소프트웨어 커뮤니티를 구축하는 데 필요한 요소를 알아보고자 한다.

## 2. 공개 소프트웨어 커뮤니티 현황

공개 소프트웨어 커뮤니티[5]란 누구에게나 프로그램의 소스코드에 접근할 수 있는 권리를 보장하고 책임과 권한을 공유하는 커뮤니티를 뜻한다. 이와 더불어 개발자와 사용자가 지속적으로 기여하며 여러 프로젝트를 수행하는 커뮤니티이기도 하다.

정보통신산업진흥원의 최근 조사 자료에 의하면 세계적인 공개 소프트웨어 소스코드 저장소인 깃허브(Github)에서는 약 38만 건의 프로젝트 개발이 활발히 진행 중이며, 약 2,000만 명의 개발자 및 사용자가 16만여 개의 공개 소프트웨어 커뮤니티에서 소프트웨어를 개발하거나 특정 프로젝트에 참여한다고 한다[10].

이에 비해 국내의 경우 총 248개의 공개 소프트웨어 커뮤니티가 운영 중인 것으로 파악되어 글로벌 대비 0.1% 수준의 규모이다. 그중에서도 사용자 커뮤니티는 214개(86%), 개발자 커뮤니티는 34개(14%)로 국내 공개 소프트웨어 커뮤니티의 다수는 오픈소스 사용자 커뮤니티였다. 공개 소프트웨어 커뮤니티를 기반으로 형성되는 소프트웨어 개발 모델은 크게 두 가지로 나뉜다. 하나는 소프트웨어 릴리스를 위한 활동을 중심으로 형성되는 개발자 커뮤니티이다. 다른 하나는 공개된 소프트웨어에 대한 테스트, 버그제출, 의견 제시 등을 중심으로 형성되는 사용

자 커뮤니티이다.

조사 결과에 따르면 해외에서는 여러 재단을 중심으로 커뮤니티에서 진행되는 오픈소스 프로젝트 개발 참여도가 활발하다. 그러나 국내에서는 개발에 참여하기보다는 공개 소프트웨어 커뮤니티가 제공하는 결과물을 활용하거나 사용법을 묻는 것이 대부분으로 생산자 관점의 공개 소프트웨어 커뮤니티 활용이 매우 저조하다.

### 3. 공개 소프트웨어 커뮤니티의 거버넌스 모델

공개 소프트웨어 커뮤니티를 통한 개방형 혁신은 특히 내부의 자원과 역량이 부족한 기술집약형 벤처기업이나 중소기업의 경우 사업화 성공률을 높일 수 있는 중요한 수단이다. 이는 연구 개발, 기술 사업화의 일련의 과정에서 자사가 필요한 자원과 역량을 외부에서 조달할 수 있기 때문이다. 더욱이 기업 간 신제품 개발 경쟁의 가속화로 시장에서 성공한 제품도 그 수명이 갈수록 짧아지며, 기술개발 비용은 천문학적 규모로 증가했다. 따라서 개방형 혁신전략은 선택이 아닌 필수로 인식된다. 하지만 이러한 공개 소프트웨어 커뮤니티 기반의 개방형 혁신전략은 커뮤니티가 성장함에 따라 복잡성이 증가한다. 그렇기 때문에 의사결정이 어렵고 협력관계 관리에 비용이 발생할 수 있다. 또한 보유 기술의 유출을 조심하고 조직 관리의 변화도 필요하므로 올바른 방향설정과 관리가 중요하다.

공개 소프트웨어 커뮤니티는 공개된 소프트웨어의 소스코드에 대한 바른 사용 권한을 제시한다. 이는 커뮤니티 참여자의 비즈니스 전략에 영향을 미치는 라이선스 관리와 어떻게 커뮤니티의 참여자들과 협력할지 정의하고 조직 전략에 영향을 미치는 거버넌스 관리로 구분할 수 있다. 좋은 공개 소프트웨어 커뮤니티를 구축하기 위해서는 공개 소프트웨어의 라이선스 영역과 공개 소프트웨어 커뮤니티의 거버넌스 영역에서 각각의 관리요소를 고려해야 한다.

프로젝트의 라이선스 관리는 OSI(Open Source Initiative)에서 오픈소스 프로젝트에서 사용할 수 있는 10가지 오픈소스 정의를 제시해 놓았다. 이런 정의를 만족하는 GPL, EPL, LGPL, CPL, ApacheL, BSDL, MITL 등 프로젝트의 전략과 목적에 맞게 선택할 수 있는 다양한 공개 소프트웨어 라이선스가 준비돼 있다. 소프트웨어 커뮤니티 참여자에게 공개한 소프트웨어 소스코드에 대한 바른 사용 권한을 제시하는 것은 비교적 쉬운 일이다. 하지만 공개 소프트웨어 커뮤니티의 거버넌스 영역은 대부분의 국내기업에게 아직 생소한 용어이다.

공개 소프트웨어 커뮤니티는 자발적인 참여와 기여가 핵심 원동력이다. 이에 커뮤니티가 잘 유지되려면 커뮤니티 참여자를 어떻게 효과적으로 조직해낼 것인가의 문제를 해결해야 한다. 강제적 규칙이 없는 느슨한 협의로 의사결정이 이루어지는 조직을 어떻게 구성하고 운영해야 하는지, 커뮤니티에서 공개한 소프트웨어를 사용하는 사용자를 어떻게 지원할 수 있는지도 고려해야 한다. 공개 소프트웨어 커뮤니티 거버넌스란 커뮤니티 참여자가 수행할 수 있는 역할과 커뮤니티 참여 방법, 프로젝트 내 의사결정 프로세스를 설명하여 커뮤니티 참여자들이 혼란으로 빠져나가는 것을 방지하는 사회적 프레임워크를 의미한다.

공개 소프트웨어 커뮤니티 거버넌스 모델은 의사결정의 관점에서 보면 한 개인이나 집단이 중앙집권적인 권한을 갖는 선의의 독재형(Benevolent dictatorship) 모델과 참여자들이 분권적 권한을 갖는 능력중시형(Meritocratic) 모델로 구분할 수 있다. 기여의 개방성 관점에서 보면 소수

의 전문가가 폐쇄적으로 개발한 후 결과물을 공개하는 성당형(Cathedral style)모델과 많은 사람의 아이디어가 자유롭게 교류되면서 소프트웨어가 개발되는 시장형(Bazaar style) 모델[1]로 구분할 수 있다.

Linux Kernel 프로젝트는 누구나 기여에 참여하는 것이 가능해 2005년부터 2011년까지 800개 단체에서 7,800 이상의 개발자가 기여했다[7]. 이 많은 기여 중 어느 것을 수용할 것인지 결정하는 최종 권한은 최초 개발자인 리누스 토르발스(Linus Torvalds)에게 있다. GNU Emacs 프로젝트는 소규모 전담자에 의해 개발되며 그만큼 릴리스 빈도가 낮다. 한편 중요한 의사결정은 리처드 스톨먼(Richard Stallman)과 소수의 개발자에 의해 이루어진다. Apache HTTPD 프로젝트는 모든 사람의 기여를 받아들이는 Apache 재단의 능력중시 모델을 채택하고 있어 능력에 따라 의사결정 과정에 참여하는 구조이다. Apache OODT 프로젝트는 Apache 재단의 능력중시 모델을 따르지만 소수의 핵심 개발자에 의해 개발되고 그들이 릴리스 여부를 결정한다. Ubuntu 프로젝트는 선의의 독재자인 설립자가 있지만 능력 위주로 구성된 위원회와 기술 이사회에 상당 부분의 의사결정이 위임되어 있고 일부 기능을 개발할 때는 시장형으로 운영된다. 또한 일부 핵심 기능은 소수의 초청 개발자가 개발하는 방식으로 운영된다.

초기의 공개 소프트웨어 커뮤니티는 아직 다수의 참여자가 없기 때문에 거버넌스 영역을 관리하지 않아도 혼란스럽지 않지만, 공개 소프트웨어 커뮤니티가 성장하게 되면 반드시 해당 프로젝트의 정책과 전략을 커뮤니티 참여자들에게 명확하게 전달해야만 한다. Apache Foundation[8], Eclipse Foundation, OpenStack Foundation, Linux Foundation[9] 등의 성공적인 공개 소프트웨어 커뮤니티들의 공통점을 살펴보면 커뮤니티의 참여자들을 위하여 해당 프로젝트에 어떤 방식으로 참여할 수 있으며 참여를 통해 기대하는 바가 무엇이며 자신들의 기여가 어떤 식으로 사용되고 보호될 것인지를 알리기 위하여 잘 구성된 거버넌스 모델을 관리하고 있는 것을 알 수 있다.

공개 소프트웨어 커뮤니티는 다양한 목적과 동기를 가진 자발적인 참여자들로 구성되지 때문에 커뮤니티의 거버넌스 모델이 갖추어야 할 핵심은 커뮤니티 참여자의 역할과 책임이 명확히 규정되어야 하며 커뮤니티의 의사결정 구조가 투명하게 공개되어야 한다. 따라서 잘 구성된 커뮤니티 거버넌스 모델은 해당 커뮤니티에 참여하려는 후보자들에게 프로젝트의 구조를 쉽게 알 수 있도록 해야 하며, 내 활동이 프로젝트와 어떻게 연루되어 있으며, 프로젝트 내 의사결정이 어떻게 이루어지며, 프로젝트의 전략적 방향에 어떻게 영향을 미칠 수 있는지를 알 수 있도록 해야 한다.

#### 4. 공개 소프트웨어 커뮤니티 거버넌스 관리

직접 공개 소프트웨어 커뮤니티를 구축하기 전에 먼저 목적하는 소프트웨어에 관심이 있는 커뮤니티가 국내외에 이미 존재하는지 검색하는 것이 중요하다. 만약 동일한 관심사를 가진 커뮤니티가 이미 있는데 새로운 커뮤니티를 형성하려고 하면 왜 커뮤니티를 신규로 생성하는지 참여자들에게 설득하기 어렵다. 기존에 있는 다른 공개소프트웨어 커뮤니티의 지지를 얻기도 힘들다. 따라서 만약 유사한 관심사를 다루는 공개 소프트웨어 커뮤니티가 이미 있다면 별도의 커뮤니티를 생성하여 새로운 기반을 형성하기보다 해당커뮤니티와 협업하는 것이 가장 좋다.

이 경우에는 이미 공개 소프트웨어 커뮤니티가 활성화되었기 때문에 공개하려는 소프트웨어를 해당 커뮤니티의 사용자들이 사용할 수 있도록 기능을 추가로 제안하여, 다음과 같이 기존 소프트웨어 기능을 강화하는 방식의 접근이 가능하다.

기존 커뮤니티 중 공개하려는 소프트웨어와 유사한 관심사를 가진 커뮤니티가 없는 경우에는 새로운 공개 소프트웨어 커뮤니티는 생성해도 좋다. 이때 공개 소프트웨어 커뮤니티의 성장 단계는 기술 단계(Technical stage), 오픈소스 단계(Open source stage), 에코시스템 단계(Ecosystem stage)라는 세 단계로 구분할 수 있다.

Technical Stage는 소프트웨어 커뮤니티의 참여자 중 개발자보다는 사용자를 대상으로 공개한 소프트웨어의 바른 사용 방법과 권한을 제공하는 것에 중점을 둔다. 이 단계에서는 공개한 소프트웨어의 라이선스를 컴포넌트별로 식별한 소프트웨어 라이선스 프레임워크를 통해 프로젝트의 라이선스 정책을 사용자들에게 배포해야 한다. 또한 공개한 소프트웨어의 기능을 확인할 수 있는 프로그램 데모도 제공해야 한다. 이 단계에서는 프로젝트를 소개할 수 있는 웹사이트, 제공하는 소스코드를 다운로드 할 수 있는 저장소, 의사소통을 위한 메일링 리스트와 커뮤니케이션 채널, 프로그램의 버그를 추적관리 할 수 있는 버그 트래킹 도구, 각종 문서를 쉽게 작성해 공유할 수 있는 문서화 도구가 필요하다. 이와 더불어 사용자들이 공개된 소프트웨어를 활용할 수 있는 범위와 사용에 따르는 책임을 명확하게 인지할 수 있도록 준비해야 한다.

Open Source Stage는 공개 소프트웨어 커뮤니티에 참여하는 사용자와 개발자를 대상으로 잘 구성된 공개 소프트웨어 커뮤니티 거버넌스 모델을 구축하는 데 중점을 둔다. 이 단계에서는 커뮤니티 참여자들에게 프로젝트의 구조를 쉽게 설명하는 문서를 준비하고, 프로젝트의 로드맵을 제시한다. 또한, 커뮤니티 참여자가 어떻게 참여할 수 있는지를 개발자와 사용자로 구분하여 자세히 알려주고, 커뮤니티 내 분쟁이 일어나거나 의사결정이 필요할 때 어떤 방식의 의사결정 과정을 따르는지 투명하게 공개해야 한다.

공개 소프트웨어 커뮤니티의 전략적 방향에 따라 수많은 유형의 거버넌스 문서가 필요할 수 있다. 다만 일반적인 공개 소프트웨어 커뮤니티 거버넌스 문서에 포함되어야 하는 공통적인 요소는 다음과 같다.

- 개요 (overview)
- 역할과 책임 (roles and responsibilities)
- 지원 (support)
- 기여 과정 (contribution process)
- 의사결정 과정 (decision making process)

Open Source Stage 단계에서는 커뮤니티 참여자들의 역할과 책임에 따른 운영 조직이 구성되고 커뮤니티 운영 조직과 커뮤니티 참여자 간 투명한 합의를 기반으로 커뮤니티 운영이 이루어져야 한다. 이를 위해서 프로젝트 마일스톤과 릴리스를 관리할 수 있는 프로젝트 관리 도구, 개발자 및 사용자를 위한 포럼, 이슈관리, 자동화 배포 환경, 소프트웨어 품질 가시화, 문서 협업 도구가 필요하다. 이에 더해 외부의 참여자들이 프로젝트에 어떻게 기여할 수 있는지 자세히 안내하

는 문서를 준비하고, 커뮤니티의 참여자들과 지속적으로 소통을 유지하는 것이 핵심이다[11].

공개한 소프트웨어 사용자의 수가 늘어나고 공개 소프트웨어 커뮤니티가 성장하면 커뮤니티 참여자들에 의한 사용자 지원으로는 기업 사용자들이 요구하는 서비스 수준을 제공하지 못하는 문제가 발생할 수 있다. Ecosystem Stage는 공개한 소프트웨어를 비즈니스에 활용하는 기업멤버들 및 다른 공개 소프트웨어 커뮤니티와 함께 상생 협력하는 단계이다. 이는 공개 소프트웨어 커뮤니티의 확산과 지속 가능성을 보장하기 위해서이다. 따라서 이 단계에서는 프로젝트를 지원할 수 있는 기업으로 구성된 비즈니스 협의체가 조직되어야 한다. 또한 공식 기술지원 파트너 기업이 커뮤니티에서 홍보되어 기업에서 공개된 소프트웨어를 사용하기 위한 신뢰성을 제공할 수 있도록 커뮤니티에 참여하는 기업을 긴밀하게 관리해야 한다.

따라서 이 단계에서는 파트너 기업의 제품을 소개하는 마켓플레이스를 직접 운영하며, 활용 사례를 소개하고, 공식적인 기술지원이 가능함을 홍보하여 프로젝트의 신뢰성을 확보하는 것이 중요하다.

공개 소프트웨어 커뮤니티가 원활히 운영되기 위해서는 공개한 소프트웨어를 기반으로 기술지원을 제공하는 파트너 기업을 발굴해야 한다. 또한 기업 멤버를 커뮤니티에 흡수하여 커뮤니티의 발전 방향이 커뮤니티 참여 기업의 비즈니스 전략에 영향을 미치는 관계가 형성되어야 한다. 그래야 자연스럽게 커뮤니티가 지속하기 위한 자원 확보가 이루어지고 향후 공개한 공개 소프트웨어 프로젝트를 중심으로 성장하는 것을 기대할 수 있다.

공개 소프트웨어 프로젝트 및 커뮤니티 운영이 성공하기 위하여 가장 중요한 요소는 투명성을 확보하는 것이다. 투명성은 프로젝트에 참여하는 사람과 외부 참여자에 맞추어 프로젝트 관련 도메인 및 기능을 어떻게 유지할 것인가를 명시하는 것이다. 공개 소프트웨어의 특성상 투명성의 유지는 대단히 중요하다. 특히 기업에 의해서 주도되는 공개 소프트웨어라면 투명성의 확보는 해당 공개 소프트웨어 커뮤니티의 운명을 쥐고 있는 열쇠라고 해도 과언이 아니다. 어떤 공개 소프트웨어 개발자도 보이지 않는 장막 뒤에서 펼쳐지는 소프트웨어에 참여하고 싶지 않을 것이다. 기업이 공개 소프트웨어 전환을 시도할 경우, 그동안 기업 내부의 습관에 의해 투명성을 소홀히 하고 내부 프로세스를 공개하지 않는 경우가 있다. 그런 경우 커뮤니티를 활성화하기란 매우 어려우며 공개 소프트웨어 커뮤니티로의 전환이 실패로 이어질 확률 또한 매우 높다. 공개 소프트웨어 커뮤니티에 투명성이 필요한 요소는 공개한 소프트웨어의 비전, 로드맵, 릴리스 계획, 형상관리 계획, 커미터 자격 조건, 새 기능 추가, 소프트웨어 패치의 제출 과정 등이다.

공개 소프트웨어 커뮤니티를 운영할 때 다른 한 가지 중요한 요소는 문서화이다. 실제로 공개 소프트웨어는 자발적으로 참여하는 사람과 개발자 위주로 돌아가는 생태계이기 때문에 문서화에 소홀하다. 여기서 문서화는 공개한 소프트웨어 자체에 대한 문서화와 공개 소프트웨어 커뮤니티의 구조 및 프로세스에 대한 문서화도 포함한다. 문서를 통해서 커뮤니티 거버넌스 모델과 커뮤니티 구조 및 프로세스를 명시할 뿐만 아니라 해당 소스코드에 대한 문서화도 명확하게 해두어야 공개 소프트웨어 커뮤니티를 활성화할 수 있다. 따라서 커뮤니티에 참여하는 개발자 및 사용자가 쉽게 이해하고 도움을 받을 수 있으며 편집도 쉽게 할 수 있는 수준으로 문서화해야 한다.

## 5. 맺음말

본고에서는 최근 중요시 되고 있는 개방형 혁신의 핵심 활동인 공개 소프트웨어 커뮤니티 구축 및 운영을 살펴보았다. 해외의 경우 Apache, Eclipse, Openstack, Linux 등 다양한 공개 소프트웨어 관련 재단이 설립되어 클라우드, 빅데이터, 인공지능 같은 분야에서 핵심 기술을 주도하고 있다. 그러나 국내에서는 공개된 소프트웨어를 가져다 쓰는 사용자 커뮤니티를 중심으로 커뮤니티가 형성되어 있으며 글로벌 영향력이 있는 개발자 커뮤니티는 찾아보기 힘들다.

최근 우리 정부는 국가의 ICT R&D 경쟁력을 강화하기 위해 지속적으로 정책적 노력과 집중투자를 하고 있다. 또한 창의적·선도적 혁신역량을 제고하고 성과를 확산하는 데 있는 한계를 잘 알고 있다. 이에 개방형 혁신의 장점을 반영한 핵심 원천기술의 오픈R&D를 추진하여 미래 유망기술을 글로벌 시장에서 선도하고 연구개발로 나온 결과물의 활용도를 제고하기 위하여 노력[10]하고 있다. 그러나 외부의 참여자와 협력은 공개 소프트웨어 커뮤니티 기반의 개방형 혁신에 익숙하지 않은 연구기관들은 체계적인 관리모델의 부재로 혼란스러워 한다.

정부는 공개 소프트웨어 생태계를 이해하기 위해 다양한 소프트웨어 연구개발을 공개 소프트웨어 프로젝트로 전환해야 할 것이다. 또한 공개 소프트웨어 커뮤니티를 통한 사용자 저변이 확산될 수 있도록 지속적으로 노력해야 할 것이다.

기업은 공개 소프트웨어 커뮤니티에 참여하는 내부 개발자에 대하여 비판적인 시각에서 벗어나 글로벌 공개 소프트웨어 프로젝트의 참여와 기여는 기업의 우수한 기술력을 홍보하는 효과적 수단임을 인식해야 한다. 마지막으로 내부 개발자가 공개 소프트웨어 커뮤니티 참여하는 것을 적극 권장하고 기업의 비즈니스 전략에 공개 소프트웨어 커뮤니티와 연계한 개방형 혁신 방안을 마련해야 할 것이다.

### [참고문헌]

- [1] Eric S. Raymond, The Cathedral and the Bazaar, Knowledge, Technology, & Policy, Fall 1999, Vol. 12, No. 3, pp.23-49, 1997
- [2] Chesbrough, H., Open innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business School Press, 2003
- [3] 김석관 : Chesbrough의 개방형 혁신 이론. 과학기술정책, 2008
- [4] Neary Consulting : Community Building Barriers to Entry. <http://www.neary-consulting.com>, 2009
- [5] OW2 : open source software the governance makes the difference. <https://www.slideshare.net/OW2/open-sourcesoftware-the-governance-makes-the-difference>, 2015
- [6] North bridge & Blackduck, 2016 the future of open source, p.12, 16, North bridge & Blackduck, 2016
- [7] 조인성 : 느슨한 학회로서의 오픈소스 공동체. 한국텍학회, 2017
- [8] The Apache Software Foundation, Annual Report FY2017, p.1-2, The Apache Software Foundation, 2017
- [9] linux foundation project directory,

<https://www.linuxfoundation.org/projects/directory/#.WZ9pV3IUmrE>

[10] 조재홍 : 공개SW 소비국을 넘어 기여국으로 성장을 위한 제언. NIPA, 2018

[11] Federal government of the United States : Open Source Maintainer Guidelines.

[https://18f.github.io/open-sourceprogram/pages/maintainer\\_guidelines/](https://18f.github.io/open-sourceprogram/pages/maintainer_guidelines/), 2018

※ 출처: TTA 저널 제191호

(코로나 이슈로 각 표준화기구의 표준화회의가 연기·취소됨에 따라 TTA 저널로 대체합니다)