

공개 소프트웨어와 소프트웨어 산업

한국과학기술원 천명권

1. 머리말

1950년대 후반부터 소프트웨어 산업이 시작되었지만, 소프트웨어를 상품으로 인정한 것은 60년대에 들어서이고, 70년대에 이르러서야 하드웨어 공급업체는 소프트웨어의 가격을 별도로 책정을 하였다. 하지만 이때까지 소프트웨어의 구매자는 대부분 전문 직종의 사람들이었고, 일반인에게 소프트웨어가 상품으로 인치된 것은 70년대 중반 이후 PC용 소프트웨어에 의해서이다. PC용 소프트웨어가 독자적인 상품으로 판매된 것과는 달리, 중대형 컴퓨터의 소프트웨어는 회사나 기관의 특수한 목적에 부합되는 전산 시스템(하드웨어 + 소프트웨어 + 어플리케이션) 구축의 부품으로 사용이 되었다. 80년대에 들어, 몇 개의 소프트웨어들이 표준화에 성공을 하여 거대 기업으로 성장을 하였고, 이에 많은 국가와 기업들은 소프트웨어 산업에 관심을 집중하기 시작하였다.

하지만 소프트웨어의 산업화는, 과거에 소프트웨어를 공유 발전시키던 관행을 저해하고, 소프트웨어의 독점을 위한 라이선스의 적용을 야기시켰다. 이에 1983년 MIT의 연구원이던 스톨만은 소프트웨어는 자유롭게 사용할 수 있어야 한다는 기치하에 자유 소프트웨어(Free Software) 운동을 시작하였다. 1990년대의 인터넷의 발전과 리눅스의 개발은 이 운동을 전세계로 확산시키는 계기가 되었다. 하지만 자유 소프트웨어에 적용되는 GPL(General Public License)의 엄격함은 기업체들의 참여를 저해하는 요인이 되었다. 이에 1998년 공개 소프트웨어(Open source software)의 개념이 정립되고, 적용 라이선스의 폭이 넓어짐에 따라, 많은 기업이 이를 지지하게 되었다. 이를 통해 공개 소프트웨어는 단지 순수한 사회운동의 차원에서 소프트웨어 산업의 구조를 바꾸는 중요한 역할을 하기 시작하였다. 결과적으로 기업뿐만이

아니라 국가들도 공개 소프트웨어의 중요성을 인식하고, 공개 소프트웨어의 사용 활성화와 공개 소프트웨어 개발 활성화를 위한 정책들을 채택하기 시작하였다.

이 글은 공개 소프트웨어가 소프트웨어 산업에 어떤 영향을 주고 있는가를 알아 보기 위해 2절에서는 공개 소프트웨어의 정의에 대해 알아 보고, 3절에서는 소프트웨어의 특성과 가치 사슬에 대해 알아 본다. 그리고 4절에서는 공개 소프트웨어가 소프트웨어 시장에 어떤 영향을 주는가를 알아보고, 5절에서는 공개 소프트웨어를 활용한 사업 모델들을 알아본다. 그리고 마지막 6절에서는 이와 같은 상황에서 우리가 해야 할 일에 대한 몇 가지 제안을 하도록 한다.

2. 공개 소프트웨어 정의

2.1 공개 소프트웨어(Open Source Software)

공개 소프트웨어의 기본 개념은 소스를 공개하여 사용자가 소스를 읽을 수 있고, 변경할 수 있으며, 변경된 소프트웨어를 배포할 수 있도록 하는 것이다. 이와 같은 개념은 라이선스를 통해 이루어진다. 자유 소프트웨어(Free Software)가 GPL(GNU General Public License)과 LGPL(Lesser General Public License)만을 사용하는 것과는 달리, 공개 소프트웨어는 여러 가지의 라이선스를 허용하고 있다. 대표적인 라이선스로는 GPL, LGPL, BSD(Berkeley software distribution) 라이선스, MIT 라이선스, 그리고 MPL(mozilla Public License)이 있다. 현재 MPL이 가장 많이 사용되는 라이선스다.

2.2 공개 소프트웨어 개발(Open Source Project)

리눅스의 개발 방식은 소프트웨어 개발에 새로운

방식이 가능하다는 것을 보여 주었다. 이를 Raymond는 시장 방식(Bazaar-principle)이라고 표현하였다[1]. 이 방식은 분산된 계획과 수행에 기반을 두고 있으며, 많은 공개 소프트웨어 개발 방식을 설명하는데 사용된다. 이 개발은 네트워크 방식으로 조직된다. 즉 전 세계에 흩어져 있는 개발자 집단은 각자 소프트웨어의 흥미 있는 부분들에 대해 일을 하고 개선을 시키며, 이것들이 모여 좋은 소프트웨어가 만들어지는 것이다.

3. 소프트웨어 산업의 특징과 가치 사슬

3.1 소프트웨어 산업의 특징

컴퓨터 산업은 모든 기술적 제품과 마찬가지로 맞춤 산업으로 시작하였다. 즉 필요한 기관의 요구에 따라, 하드웨어와 소프트웨어 그리고 관련된 서비스가 하나의 프로젝트로 수행된 것이다. 이후 표준화에 따라 하드웨어와 소프트웨어의 분리가 일어났고, 다시 소프트웨어와 서비스의 분리가 일어났다. 하지만 아직도 기업의 경우 맞춤 서비스를 요구하는 경우가 대부분이다. 결과적으로 소프트웨어 산업은 다음과 같이 분리하여 생각할 수 있다.

1. 표준화된 소프트웨어 제품 생산
2. 표준화된 소프트웨어 관련 서비스
3. 맞춤 제품 생산과 관련 서비스

소프트웨어 산업에 있어서, 제품 개발은 많은 비용이 들지만 복제하는데 드는 비용은 아주 낮다. 결과적으로 표준화된 제품의 경우 대량 판매를 할수록 이익이 많이 남는다. 하지만 맞춤 제품의 경우는 그렇지 않으므로, 가능한 한 많은 부분을 표준화하려는 노력이 계속되고 있다. 또한 소프트웨어의 가치는 물질적인 제품과는 달리 측정하기가 어렵다. 결과적으로 소프트웨어의 가치는 계약 당사자간의 이해 관계에 따라 결정된다. 이는 표준화된 제품의 경우 제품의 가치보다는 시장을 확대하려는 이해 관계에 따라 제품의 가격이 결정된다는 것을 의미한다.

소프트웨어 관련 서비스는 자문, 설치, 지원, 그리고 교육에서 어플리케이션 관리까지 다양하지만, 주로 인력 위주의 사업이므로 규모의 확장에 따른 직접적인 이득은 별로 없다. 하지만 간접적으로 관리 비용의 축소, 기술 축적의 용이, 거대 규모의 프로젝트 수주 등의 장점이 있다.

3.2 소프트웨어의 가치 사슬

소프트웨어의 가치 사슬은 표준화된 제품과 맞춤 제품(Individual Software)에 차이가 있다. 다음 그림은 각각의 가치 사슬을 보이고 있다.

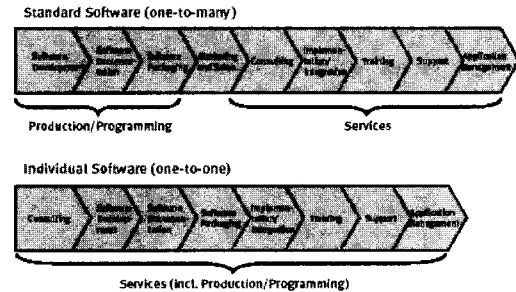


그림 1 소프트웨어 가치 사슬

맞춤 소프트웨어(Individual software)의 경우 자문(consulting)이 앞에 있고, 마케팅/유통 부분이 없다. 표준화된 제품의 경우, 생산/프로그래밍, 마케팅/유통, 그리고 서비스의 3가지 단계로 나눌 수 있다.

표준화된 제품의 경우 초기에는 생산/프로그래밍에 비용이 집중되고, 이후에는 마케팅/유통 그리고 서비스에 비용이 집중된다. 표준화된 제품 시장에서 자주 인용되는 성공 요인은 무어의 법칙이라 불린다. 이 법칙에 따르면, 제품이 특정한 틈새 시장에 출시되어 12개월에서 18개월 사이에 시장 점유율이 40%에 다다라야 한다는 것이다. 경쟁자의 시장 점유율이 훨씬 적은 경우, 사람들은 이 회사가 시장을 선도한다는 말을 퍼트리기 시작한다. 이와 같은 일이 일어나면, 다음 12개월 안에 시장 점유율은 50%를 넘을 가능성이 높다. 이와 같은 상태에 이를 때까지는 제품의 갱신을 위한 생산/프로그래밍, 마케팅/유통 그리고 서비스 비용은 점점 증가한다. 하지만 업계의 표준으로 자리를 잡은 이후의 마케팅/유통 비용은 점점 감소하고, 서비스 비용은 표준화된 서비스 업체들이 늘어남에 따라 부담이 줄어들게 된다. 결과적으로 이후에 소요되는 비용은 선도 자리를 지키기 위한 비용이라고 보면 된다.

반면에 사용자의 요구에 따라 맞춤 생산하는 소프트웨어의 경우는 전체를 하나의 서비스 산업으로 보아야 한다. McKinsey의 내부 조사에 따르면, ERP (Enterprise Resource Planning) 시스템을 구축하는

경우, 소프트웨어 제품 관련 비용이 30%에 해당하고, 나머지 70%가 서비스와 관련된 비용으로 나타났다 [2].

일반적으로 공개 소프트웨어의 개발은 표준화된 제품의 경우에 속하고, 결과적으로 관련 산업은 대부분 표준화된 제품의 가치 사슬에 따른다. 하지만 점차적으로 공개 소프트웨어가 맞춤형 소프트웨어의 부품으로 사용되는 경우가 증가함에 따라 이 가치 사슬 역시 중요한 역할을 한다. 그러면 표준화된 제품의 3가지 단계가 독점 소프트웨어의 경우와 공개 소프트웨어의 경우 어떻게 다른가 알아 보도록 한다.

3.3 생산/프로그래밍

이 단계에는 소프트웨어 개발, 문서화, 그리고 패키징이 속한다. 각각의 경우를 보도록 한다.

3.3.1 소프트웨어 개발

소프트웨어 개발의 추세는 다음과 같다[3].

- 하드웨어에 비해 소프트웨어의 중요성이 증가하고 있다.
 - 소프트웨어 제품에 비해 소프트웨어 관련 서비스들의 중요성이 증가하고 있다.
 - 복잡성의 증가
 - 품질 요구의 증가
 - 맞춤형 제품이 표준화된 제품으로 변하는 추세
 - 이미 존재하는 소프트웨어에 대한 부담 증가(구 버전의 소프트웨어를 새로운 컴포넌트에 적용하는 것은 소프트웨어 개발의 3분의 2에 해당하고, 이는 점점 증가하고 있다.)
- 이에 따라 소프트웨어 개발의 성공 요인은 다음과 같다.

- 기능 요구가 충족되어야 한다.
- 고품질이어야 한다.
- 시장에 내 보내는 시간이 짧아야 한다.
- 개발 비용은 최소화되어야 한다.

공개 소프트웨어의 경우 다음과 같은 장점이 이야기 되고 있다.

첫째, 공개 소프트웨어는 전체 집단이 문제들을 해결하므로 보다 빨리 그리고 보다 잘 오류 처리를 한다. 결과적으로 개발자 집단이 큰 경우, 고품질이다. 둘째, 공개 소프트웨어는 다양한 사용자를 대상으로 하므로 안정성과 기능을 강조한다. 셋째, 공개 소프트웨어의 경우는 소스가 공개 되어 있으므로, 보

다 좋은 성능을 위한 특별한 요구조건들에 맞게 소프트웨어가 개선될 가능성이 높다.

3.3.2 소프트웨어 문서화

소프트웨어 문서는 3종류로 분류할 수 있다.

첫째, 소프트웨어 자체가 문서이다. 소스 코드를 볼 수 있으면, 개발자는 코드를 사용하여 제품에 대한 정보를 얻을 수 있다. 둘째, 소프트웨어에 내장되어 소프트웨어 코드를 설명하는 전자 문서가 있다. 셋째, 소프트웨어의 기능을 설명하는 종이로 된 문서가 있다.

독점 소프트웨어는 보통 종이 문서와 같이 판매된다. 문서는 개발 과정의 마지막 단계에서 작성된다. 반면에 공개 소프트웨어는 일반적으로 계속 발전되는 소프트웨어이므로 사용자를 위한 문서가 불충분하다. 공개 소프트웨어와 관련된 기업들, Red Hat(전자 문서)과 O'Reilly publishing company(종이 문서)의 사업 모델의 일부가 문서를 만들고 판매하는 것이다. 즉 공개 소프트웨어의 경우는 종이로 된 문서가 부족한 반면, 소스 코드가 보안을 해 준다.

3.3.3 소프트웨어 패키징

소프트웨어 패키징은 소프트웨어 제품들과 부품을 모으고 통합하며 최적화하는 것이다. 단순히 여러 소프트웨어 부품을 모으는 것(예를 들면, CD-ROM이나 웹 사이트)은 생산 개념에 있어서 가치를 추가하지 않으므로 소프트웨어 패키징이 아니다. 소프트웨어 패키징은 패키지에 가치를 추가하는 것이고, 패키징된 소프트웨어 제품의 가치는 각 부분의 가치를 더한 것과는 다르다.

독점 소프트웨어는 보통 패키지 방식으로 판매된다. 패키징은 개발의 마지막 단계에 속한다. 공개 소프트웨어를 기반으로 한 사업에서 패키징은 중요한 사업모델이다. 공개 소프트웨어는 웹 상에 각각 개발 프로젝트의 형태로 존재하므로, 유통업체들은 각 소프트웨어의 현재 버전들을 모아, 최적화한 후 패키지로 판매한다.

3.4 마케팅/유통

마케팅은 사용자에게 제품을 알리고 구매하도록 하는 것을 목표로 한다. 독점 소프트웨어 제품의 마케팅은 이를 개발한 회사에 의해 실행된다. 하지만 공개 소프트웨어의 경우는 일반적으로 개별 제품

대한 마케팅이 없다. 단지 패키지들의 유통, 보완 제품들, 그리고 공개 소프트웨어 서비스들에 의해서 이루어진다.

유통의 목적은 물리적으로 제품을 고객이 접근하여, 사용하고, 살 수 있도록 적절한 장소로 옮기는 것이다. 독점 소프트웨어의 유통은 주로 소프트웨어 제품을 개발한 소프트웨어 회사에 의해 행해진다. 또는 별도의 유통업자나 부가가치 재판매업자에 의해 행해진다. 부가가치 재판매업자는 독점적으로 또는 독자적으로 하드웨어와 소프트웨어의 조합을 판매하고, 소프트웨어 제조업자들을 위한 판매망의 역할을 한다. 즉 취합은 유통의 한 부분이며, 소프트웨어 제품을 수정하지 않고 단지 부품들을 함께 유통하는 것이다. 부가 기능이 추가되지는 않는다. 독점 소프트웨어에 있어서 취합은 개발 과정의 한 부분이나, 공개 소프트웨어에 있어서는 유통업자에 의해 수행되고, 이것 역시 별도의 공개 소프트웨어 사업 모델의 공간을 이룬다.

3.5 서비스

서비스 단계에는 자문, 설치 및 통합, 교육, 지원, 그리고 어플리케이션 운영이 포함된다. 자문의 경우 특별히 공개 소프트웨어 제품들과 이 분야에 대한 특별한 지식을 제공하는 많은 회사들이 있지만, 그들의 자문 과정과 독점 소프트웨어에 집중하는 회사들의 자문 과정 사이에 차이는 없다. 설치 및 통합의 단계에서 공개 소프트웨어를 선호하는 이유는 독점 소프트웨어로는 할 수 없는 맞춤 작업을 공개 소프트웨어로는 할 수 있다는 점이다. 즉 공개 소프트웨어는 소프트웨어 제품을 제한 없이 필요에 따라 수정하여 사용할 수 있는 기회를 제공한다.

교육은 두 종류의 회사에 의해 행해진다. 고전적인 교육 기관은 공개 소프트웨어를 포함한 모든 종류의 소프트웨어에 대한 교육을 제공한다. 그들에게 교육은 핵심 경쟁력이다. 소프트웨어 유통업자와 서비스 회사들도 특별히 그들이 제공하는 소프트웨어 제품과 솔루션에 대한 교육을 제공한다. 하지만 그들에게 교육은 제공하는 하나의 서비스일 뿐, 핵심 경쟁력은 아니다. 공개 소프트웨어의 경우, 교육은 유통업자나 통합회사에 의해 제공된다.

독점 소프트웨어의 지원은 특별한 서비스 회사나 소프트웨어 제조업자에 의해 제공된다. 전형적으로 개인 사용자와 기업 사용자의 구분이 있다. 개인 사

용자의 경우 프로그램 안에 포함된 기본적인 도움 기능과 단순한 전화 지원이 있고, 기업 사용자의 경우는 운영자를 위한 좀 더 강력한 지원과 경우에 따라 사용자 요구에 따라 수정을 돕는다. 반면 공개 소프트웨어에 대한 지원은 우선 여러 가지 포럼을 통해 공개 소프트웨어 집단에 의해 이루어진다. 하지만 기업 사용자의 경우 이와 같은 지원은 문제가 해결되는 것을 보증하지 않으므로 충분하지 않다고 생각한다. 이를 해결하기 위해, 유통업자와 독립적인 공개 소프트웨어 서비스업자 그리고 통합 회사들은 지원을 제시한다. 유통업자의 경우 소프트웨어의 특정한 문제나 부분에 대해 전문가인 공개 소프트웨어 개발자들을 고용하기도 한다. 계약에 의해, 이와 같은 공개 소프트웨어 개발자들은 유통업자의 고객들에게 전문적인 지원을 한다.

4. 공개 소프트웨어와 소프트웨어 시장

소프트웨어 시장은 소프트웨어 제품 시장, 소프트웨어 서비스 시장, 그리고 임베디드 소프트웨어 시장의 3부분으로 나누어 질 수 있다.

4.1 소프트웨어 제품 시장

1999년에 메킨지는 소프트웨어 시장을 분석하여 제품 사업에 있어서 일반적인 사업 역학을 규명하였다[2]. 분석은 고전적인 소프트웨어 사업(독점 소프트웨어의 개발)에 초점을 맞추었다. 다음의 결론을 제시하였다.

- 소프트웨어 제품 사업은 진입 장벽이 낮다. 시장은 지식에 의해 움직이고, 적은 투자비를 필요로 한다. 낮은 금융 진입 장벽은 혁신의 속도를 빠르게 하고, 이는 결국 기술 진입 장벽도 낮게 만든다. 결과적으로 이 시장은 항상 새로운 회사들의 지속적인 위협을 받는다.
- 한계 비용이 낮으므로 전세계를 시장으로 보아야 하고, 연구와 개발을 위한 높은 고정 비용으로 인해 많은 양이 팔려야만 한다. 하지만 공개 소프트웨어의 경우, 특별한 프로젝트에 투입된 시간의 개념으로 보면 높은 고정 비용이지만, 이것이 꼭 금전상으로 그런 것은 아니다. 그러므로 공개 소프트웨어는 이론상으로는 많은 노력과 적은 양의 사용이 가능한 것이다.
- 다음과 같은 특징 때문에 선도 제품이 되려는 경주

와 집중하려는 경향이 있다. 첫째, 프로그램의 호환성에 의한 network 효과가 존재한다. 즉 하나의 프로그램이 더 많은 사용자를 확보하면, 사용자들은 더 많은 이익을 그 프로그램에서 얻을 수 있다. 둘째, 사람들은 하나의 프로그램을 사용하여 일을 하도록 교육을 받으면, 다른 프로그램을 사용하지 않는 장벽이 있다. 셋째, 프로그램의 인지도는 구매 결정의 중요한 요인이다. 이와 같은 집중의 경향은 일반 소비자 시장에서 훨씬 강력한 것 같다. 특별한 IT 지식을 가진 사람들에게 의해 구매 결정이 내려지는 시장에서는 교육 장벽과 인지도 장벽이 일반 소비자 시장만큼 높지 않다.

· 하지만 소프트웨어 제품 시장에서의 시장 점유율은 기술의 변화 때문에 항상 불안하다.

소프트웨어 제품은 서버 시장, 데스크 탑 시장, 그리고 대기업용 솔루션 시장으로 분류할 수 있다. 각 시장에서 공개 소프트웨어의 영향을 알아 본다.

4.1.1 서버 시장

리눅스는 서버의 운영체제로서 다음과 같은 여러 가지 장점을 제공한다.

- 높은 안정성과 신뢰성
- 라이선스 비용이 낮거나 없다.
- 자유롭게 수정할 수 있다.
- 커널이 상대적으로 작다.

뿐만 아니라 많은 유닉스 어플리케이션들을 쉽게 수정하여 리눅스 하에서 수행시킬 수 있다. 또한 많은 소프트웨어 어플리케이션 벤더들은 리눅스와 유닉스에서 동시에 수행시킬 수 있는 어플리케이션을 개발하기 위해 노력하고 있다. 결과적으로 서버 시장에서 리눅스의 시장 점유율은 높아질 것으로 예상된다. 운영체제인 리눅스뿐만 아니라 어플리케이션 분야에서도 공개 소프트웨어는 점점 높은 시장 점유율을 보이고 있다. 대표적인 것으로는 웹 서버인 아파치를 비롯하여 파일 서버인 삼바를 들 수 있다.

4.1.2 데스크 탑 시장

리눅스와 OpenOffice는 오랫동안 마이크로 소프트사의 윈도우와 오피스의 경쟁자로 인식되어 왔다. 하지만 아직 사용자의 편의성과 친밀성의 면에서 상당히 약하다. 더군다나 대부분의 PC가 마이크로 소프트사의 제품을 사전 설치하여 판매하는 상황에서 리눅스가 강력한 경쟁자가 되기는 어려울 것 같다. 하지만 어플리케이션의 경우, Mozilla나 OpenOffice

는 1, 2년 안에 마이크로 소프트사의 IE(Internet Explorer)나 Office의 강력한 경쟁자가 될 수도 있을 것이다.

4.1.3 대기업용 솔루션

대기업용 솔루션의 주요 제품은 ERP(Enterprise Resource Planning)시스템이다. 하지만 이외에 CRM(customer relationship management), SCM(supply chain management), KM(knowledge management), 그룹웨어, e-learning 소프트웨어 등을 들 수 있다. 현재 많은 제품들이 리눅스 운영체제에서 사용할 수 있다.

4.2 소프트웨어 관련 서비스 시장

이 시장의 특징은 서비스 특성에 따라 소프트웨어 제품 시장과는 다르다[2].

- 지식 기반 사업이므로 회사를 시작하는데 필요한 비용은 낮으나, 지식이 가장 큰 진입 장벽이다.
- 소프트웨어 제품 시장의 혁신 속도가 빠르므로 서비스 사업의 혁신 속도도 같은 속도를 유지할 수 있어야 한다. 새로운 기술이 나오면, 이와 관련된 서비스를 제공할 수 있어야 한다.
- 소프트웨어 서비스는 고정비용과 임계비용이 높은 고전적인 인건비 사업이다. 즉 비슷한 프로젝트를 두 번째 수행한다 해도, 소요되는 비용은 첫 번째 프로젝트와 거의 같은 수준이 든다.
- 인적 자원이 가장 중요한 자산이다. 결과적으로 능력 있는 직원을 채용하고, 이들에게 투자하는 것이 중요하다.

· 제품 사업과는 달리, 서비스 시장은 규모의 경제에 의해 지배되지 않는다. 결과적으로 이 시장은 여러 회사가 나누어 가진다. 결과적으로 이 시장에는 작은 회사들이 각각의 지역을 담당하는 형태를 나타낸다. 전 세계를 대상으로 하는 회사는 아주 적다.

대기업을 대상으로 하는 세계적 서비스에 공개 소프트웨어가 영향을 준 것은, 대기업 솔루션 제품 회사들이 그들의 제품을 리눅스 하에서 수행되도록 작업을 한 것이다. 반면에 소규모 서비스 회사에게 있어 공개 소프트웨어는 커다란 영향을 주었다. 첫째, 리눅스와 공개 소프트웨어만을 집중적으로 서비스하는 회사가 생겼다. 둘째, 공개 소프트웨어 유통업체가 그들의 제품에 대한 서비스와 지원을 하고 있다. 셋째, 특별한 서비스 회사들이 그들의 서비스에 공개

소프트웨어를 포함하고 있다. 이 3가지 그룹의 회사들간의 경쟁은 앞으로 계속 될 것이다.

4.3 임베디드 소프트웨어 제품과 관련 서비스 시장

임베디드 소프트웨어 사업의 특성은 다음과 같이 요약할 수 있다.

- 호환성과 이식성 그리고 개발자의 능력 때문에 간접 네트워크 효과가 강하다. 임베디드 시스템 틀이 보편적이 아니면, 관련 부품이 적게 사용될 것이고, 결과적으로 적은 사람이 관련 소프트웨어를 다룰 수 있을 것이다. 특히 운영체제는 다른 어플리케이션과 소프트웨어 개발 틀의 기반이 되므로 네트워크 효과가 강력하다.
- 지식 기반 사업이고 초기 비용은 많이 들지 않는다. 결과적으로 빠른 혁신이 일어난다. 하지만 네트워크 효과가 강한 진입 장벽을 형성한다.
- 임계비용이 낮으므로 전세계 시장을 대상으로 해야 한다. 공개 소프트웨어가 아닌 경우, 높은 초기 개발 비용을 회수하기 위해서는 대량 판매를 해야 이윤을 남길 수 있다.
- 장기적으로 보면, 기술의 변화에 따라 안정적인 시장 지위를 보장할 수 없다.

리눅스는 이 시장에서 다음과 같은 장점이 있다 [4].

- 표준화가 되어 있고 공개 소프트웨어이다. 모듈 구조로 되어 있어 불필요한 부분을 쉽게 제거할 수 있다. 결과적으로 필요에 따라 리눅스의 기능을 자유롭게 변경할 수 있다.
- 가격이 낮다.
- 능숙한 프로그래머들이 많이 있다. 반면에 다음과 같은 단점이 있다.
- 리눅스는 실시간 시스템으로 디자인 되지 않았다. 리눅스를 다양한 환경에서 실시간 시스템으로 만들기 위한 프로젝트들이 진행되고 있지만, 실시간 시스템으로 디자인된 운영체제와 비교하여 아직 성능이 떨어진다.
- 임베디드 표준에서 보면 리눅스는 비교적 크다. 하지만 이를 줄이는 작업은 계속되고 있으며, 경쟁이 되는 독점 소프트웨어의 커널 역시 크다.
- 임베디드 시스템은 일반적으로 사업에서 중요한 부분이다. 전문적으로 24시간 지원을 하는 곳이 없

는 것이 리눅스의 문제이다.

결과적으로 리눅스는 실시간 처리를 필요로 하지 않거나, 중요한 부분이 아닌 임베디드 시스템의 좋은 대안이다. 기대할 수 있는 리눅스 임베디드 시스템은 셀탑 박스, 케이블 TV, 장난감, 자동차 부품, 제조용 부품, 그리고 장기적으로 가전제품을 들 수 있다[5].

5. 공개 소프트웨어 사업 모델

공개 소프트웨어에 기반을 둔 사업 모델은 다음의 두 가지로 분류할 수 있다.

- 유통업체와 소매업체
- 공개 소프트웨어 관련 서비스

유통업체와 소매업체는 다시 리눅스 유통업체, 틈새 특수 공개 소프트웨어 유통업체, 그리고 소매업체로 분류할 수 있다. 공개 소프트웨어 관련 서비스는 공개 소프트웨어 개발 및 공동체 활성화 사업과 서비스와 지원 사업으로 나눌 수 있다.

5.1 유통업체와 소매업체

5.1.1 리눅스 유통업체

리눅스 유통업체는 리눅스 운영체제를 자신들의 버전으로 패키징하여 판매한다. 초창기의 리눅스 유통업자는 Red Hat, SuSE, MandrakeSoft, Caldera, Turbolinux, and Slackware 등을 들 수 있다. 이 회사들은 리눅스 운영체제를 다양한 하드웨어를 지원하는 여러 종류의 서버 어플리케이션, 데스크 탑 어플리케이션, ecommerce suits 등과 같은 패키지로 판매한다. 생산 과정에 있어 유통업자에 의해 추가되는 가치는 주로 패키징에 있다.

리눅스 유통 사업의 중요 성공 요인은 인지도를 쌓는 것이다. 그러므로 유통업자들은 마케팅에 많은 돈을 투자한다(광고, 박람회, 세미나 활동).

5.1.2 틈새 전문 공개 소프트웨어 유통업체

틈새나 전문 유통업체는 운영체제가 아닌 다른 공개 소프트웨어를 개발하고 배포한다. 그들의 제품은 어플리케이션, 개발, 그리고 관리 툴들을 포함한다. 일반적으로, 그들의 제품은 리눅스에서 사용되도록 개발되지만, 일부 제품은 윈도우나 다른 운영체제에서도 사용할 수 있다. Zope(formerly Digital Creations), Sendmail.com, Covalent Technologies, Cygnus(acquired by Red Hat), Precision Insight

(acquired by VA Linux), MySQL, ActiveState, and CollabNet 등이 좋은 예이다.

이 사업 모델에서 회사는 공개 소프트웨어 프로젝트와 공생을 한다. 이와 같은 회사의 주요 기능은 지정된 제품의 일정을 조정하고 배달과 지원을 책임진다. 이들은 특정한 제품의 주요 개발자들을 고용하여 개발자 집단과의 관계를 유지한다.

대부분의 회사들은 그들의 제품에 부가적인 서비스를 판매하고 있다(자문과 지원). 이와는 별도로 MySQL의 경우, 일반인에게는 GPL 라이선스를 사용하고, 기업인 경우는 상용 라이선스를 적용한다. Precision Insight는 바로 전 버전의 소스까지만 공개한다. Sendmail.com은 기본적인 sendmail 기능 위에 독점 상용 소프트웨어를 만들어 판매한다.

5.1.3 공개 소프트웨어 소매업체와 보완 상품들

소매업자는 유통업자의 중요한 판매망이다. 그들은 유통업자의 소프트웨어 제품을 팔거나 공개 소프트웨어 제품이나 판매촉진 상품들에 대한 추가적인 문서나 정보를 제공하고 판매한다. 소매업자는 소프트웨어 개발 과정에는 포함되어 있지 않다. 대신, 그들의 주요 경쟁력은 유통과 출판이다. 소매업자는 단지 공개 소프트웨어 하나에만 매달리지는 않는다. Lehmanns Fachbuchhandlung나 mitp의 경우, 공개 소프트웨어 관련 교육이나 책은 그들의 소매업이나 출판 사업의 한 부분일 뿐이다. O'Reilly만이 공개 소프트웨어에만 기반을 둔 유일한 기업으로 알려져 있다. O'Reilly는 문서와 교육용 책들을 팔고 있다.

CNET, ZDNet, 그리고 Fona(Denmark)는 출판과 소매업 사슬에 속하는 다른 예이다. 이외에 Crazy Penguin(UK), Linuxland(Netherlands), Linux Central(U.S., part of INT Media Group), 그리고 Linuxbutikken(Sweden, Norway) 등은 특화된 리눅스 상점들이다.

5.2 서비스 업체

5.2.1 공개 소프트웨어 개발 및 공동체 활성화 사업

이 분야는 두 가지 다른 종류의 사업이 존재한다. 첫째는 SourceXchange, Cosource.com, intraDAT(vshop.org)과 같이 시장터를 제공하는 사업이고, 두 번째는 LogOn Technology Transfer나 Linux New Media 등과 같이 컨퍼런스나 박람회를 개최하는 사

업이다.

소프트웨어 개발을 위한 시장은 소프트웨어 개발 요구와 개발자 집단을 연결하여 주는 사업이다. 이 경우 단지 소프트웨어 개발 요구들만을 모아 놓은 것만도 서비스가 될 수 있다. 같은 문제를 가지고 있는 여러 고객이 시장을 통해 자금을 모아 소프트웨어 솔루션을 얻는다. 하지만 알려진 바로는, 이 사업으로 이익을 남긴 기업은 없다. SourceXchange는 2001년 4월에 문을 닫았다. 구매자와 판매자를 연결하는 기능은 서비스 회사의 범주에서 추가 서비스로 사용되는 것이 나올 것이다. MandrakeSoft와 같은 유통업자는 이 모델의 일부를 지원 업무를 제공하는데 적용하고 있다. 그들은 특별한 문제를 풀기 위해 공개 소프트웨어 개발자나 전문가에게 어느 정도의 돈을 지불한다.

이 그룹의 다른 사업 모델인 컨퍼런스 주최 사업은 공개 소프트웨어 프로젝트에 관한 관심을 증가시켰고, 공개 소프트웨어 집단과 사업 파트너가 만날 수 있는 기회를 제공했다. 공개 소프트웨어나 리눅스에 특화된 주최자나 일반적인 컨퍼런스 주최자는 공개 소프트웨어에 그들의 역량을 집중시켜 어느 정도 수익을 올릴 수 있었다. 하지만 공개 소프트웨어 컨퍼런스나 박람회는 일반적으로 입장료가 싸서, 이의 주최자가 이윤을 남기는지는 의문이다.

5.2.2 공개 소프트웨어 관련 서비스와 지원

공개 소프트웨어 관련 서비스와 지원에는 자문, 시스템 통합, 지원, 유지보수, 원격관리, 교육, 그리고 어플리케이션 관리 등이 포함된다. 공개 소프트웨어 관련 서비스 시장의 회사들은 그들의 경력에 따라 다른 형태를 취한다. 첫째, 리눅스나 다른 공개 소프트웨어에 대한 경력을 가지고 있는 회사들이 있다. 그들은 자신들의 제품 지식을 기반으로 서비스 사업을 만들려고 노력한다. 그러므로 이들의 핵심 경쟁력은 기술관련 지식과 제품에 대한 지식이다. 대부분은 모든 서비스를 제공한다. 리눅스 유통업체, 틈새 특수 유통업체, 그리고 독립적인 공개 소프트웨어 서비스 회사들이 이 범주에 속한다.

두 번째, 일반적인 IT와 관련된 서비스를 제공하던 회사로서 특별한 프로세스 지식을 가지고 있다. 이는 IT 자문, 시스템 통합, IT 교육, IT 채용 등에 있어서 유용한 지식이 될 수 있다. 이들은 자신들의 서비스 영역을 공개 소프트웨어 관련 서비스로 확장

할 수 있다.

둘 중 어느 그룹의 회사가 성공할 것인가는 서비스 시장에서 제품에 대한 지식과 프로세스에 대한 지식 중 어느 것이 중요한가에 달려 있다.

6. 맺음말

소프트웨어 시장은 진입 장벽이 낮은 반면 경쟁이 심하고, 표준화 제품이 되기 위해서는 제품의 생산뿐 아니라 마케팅/유통과 서비스를 모두 충족 시킬 수 있어야 한다. 현재 대부분의 표준화 제품이 미국의 회사에 의해 선도되고 있는 것은, 전 세계를 장악하고 있는 하드웨어 제조업체의 마케팅/유통망을 충분히 활용한 결과라고 할 수 있다. 이와 같은 상황에서 공개 소프트웨어의 출현은 미국이 소프트웨어 시장까지 독점하는 것을 어느 정도 제어해 줄 것이다. 하지만 공개 소프트웨어의 경우 역시 대표적인 업체들을 보면, 대부분 선진국이 주도 하고 있는 것을 볼 수 있다. 이와 같은 현상은 공개 소프트웨어 프로젝트에 참여하는 소프트웨어 엔지니어들의 분포를 보면 이해 하기가 쉬울 것이다. 즉 공개 소프트웨어가 소스를 공개한다고 하여 아무 노력 없이 이를 활용하고 발전 시킬 수 있는 것은 아니다. 이를 사업적으로, 또는 개인적으로 활용하기 위해서는 많은 노력이 필요하다. 한국의 소프트웨어 산업 발전을 위해서는 공개 소프트웨어의 사용 활성화와 공개 소프트웨어 개발의 활성화는 필수적이다. 이를 위해 공개 소프트웨어 개발 및 공동체 활성화 사업을 정부 주도로 하는 것이 필요하다. 대학의 경우는 많은 학생이 공개 소프트웨어 개발에 참여할 수 있도록 교과 과정을 수정하는 것이 필요할 것이다. 기업체와 연구소의 경우는 선도 제품이 되기 어려운 소프트웨어들은 과감하게 공개하여 소프트웨어 엔지니어의 질 향상과 공개 소프트웨어 집단에서 한국의 위상을 세우는 것이 필요하다.

이 글은 유럽 연합회의 자금으로 Berlecon Research와 International Institute of Infonomics at

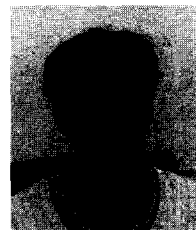
the University of Maastricht에 의해 수행된 FLOSS 프로젝트의 최종 보고서의 Part III: Basics of Open Source Software Markets and Business Models의 내용을 축약한 것이다. 좀 더 자세한 내용을 원하는 사람은 다음 사이트를 참조하면 된다.

<http://www.infonomics.nl/FLOSS/report/>

참고문헌

- [1] Raymond, E., "The Cathedral and Bazaar," <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar.html>, 1997.
- [2] Hoch, Detlev J. et al. (1999): Secrets of Software Success: Management Insights from 100 Software Firms around the World. Boston, Massachusetts, 1999.
- [3] Balzert, Helmut (1998): Lehrbuch der Software-Technik : Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung, Heidelberg.
- [4] Cook, Rick (2000): Embedded Linux Basics. LinuxWorld.com, May 2000, <http://www.linuxworld.com/linuxworld/lw-2000-05/lw-05-embedded.html>.
- [5] IDC (2000a): Linux: What s the use? Western Europe 1999-2004.

천 명 권



1981 한양대학교 전자학과
 1983 한국과학기술원 산업전자과(석사)
 1988 한국과학기술원 전산학과(박사)
 1983~1999 데이콤
 1999~2001 Georgia Tech. Exchange Visitor
 2001~현재 한국과학기술원 초빙교수
 관심분야: 컴퓨터 구조, 소프트웨어 공학, 공개 소프트웨어
 E-mail : tcheun@camars.kaist.ac.kr