부록 B

공개소프트웨어 지원 정책

1. 공개소프트웨어 지원 정책

1.1. 세계 주요국가의 지원 정책

1.1.1. OSS 지원의 목적

오픈소스 소프트웨어(Open Source Software, 이하 OSS)를 지원하는 이유로는 경제적 효율성, 보안안전프라이버시, 특정 벤더에 대한 종속성의 극복, 기술혁신, 시장경쟁의 확보, 국내 소프트웨어 산업의 발전 등 여러 가지를 들고 있다. 특히 OSS 지원의 필요성에 관한 이러한 논리적 이유는 공공분야에서 더욱 강하게 나타나는데, 그 결과 전 세계적으로 정부차원에서 OSS를 지원하는 정책이 다양하게 제시되거나 실현되고 있다. 미국유럽연합 등을 중심으로 OSS의 개발을 지원하는 정책이 있는가 하면, 페루, 아르헨티나, 이탈리아, 등 남미 및 유럽을 중심으로 공공부문에서 OSS의 구매를 강제하거나 권장하는 정책이 제안되고 있다.

OSS에 대한 지원정책의 확산을 막기 위하여 최근 마이크로 소프트의 지원으로 ISC(Initiative for Software Choice)가 만들어졌는데, 각국의 정책입안자들에게 소프트웨어 구매와 관련하여 OSS에 대한 선호 또는 강제구매를 배제하고 소프트웨어 선택의 기회를 넓힌다는 목표를 밝히고 있다. 주요한 활동 중의 하나는 각국의 소프트웨어 구매와 관련된 정책을 평가하여 반대 또는 지지의 목소리를 내는 것이다. 이와 관련하여 아르헨티나, 콜롬비아, 이탈리아, 페루 및 캘리포니아 주 법안에 대해서는 반대의 의견을 제시하고 있으며 영국의 정책에 대해서는 찬성하는 의견을 제시하였다.

OSS에 대한 각국 정부의 지원을 둘러싼 논란이 가열되고 있는 가운데 이러한 동향이 국내의 정책에는 어떠한 시사점을 주는지에 대한 논의가 필요하다고 본다. 여러 국가의 정책 모두가 시사점을 주는 것은 분명하지만,

현재 국내 상황과 비추어볼 때, 특히 영국의 정책이 많은 정책 대안을 제시해 줄 수 있다고 본다.

이하에서는 OSS에 대한 영국 정책의 배경이 되고 있는 유럽연합의 정책을 간단히 살펴보고 영국의 정책, 특히 소프트웨어의 구매와 관련된 정책을 집중적으로 살펴보고자 한다.

1.1.2. 유럽연합의 OSS 지원 프로그램

유럽 위원회(European Commission)는 1998년부터 오픈소소 소프트웨어 관련 정책을 추진해 오고 있다. 현재 직접적으로 관련이 있는 정책으로는 기술개발프로그램인 프레임워크(Framework) 프로그램 중 IST(The Information Society Technologies) 프로그램과 회원국 행정부간의 데이터 교환을 촉진하기 위한 IDA(Interchange of Data Between Administrations) 프로그램이 존재한다. IST 프로그램은 오픈소스 소프트웨어를 위한 기술공급 정책으로서 기술개발을 지원하는 프로젝트들을 운영하고 있다. 대부분 구체적인 기술을 개발 하거나 적용가능성을 탐색하는 프로젝트이지만, FLOSS 프로젝트처럼 사회, 과학적, 정책적 측면에서 오픈소스 소프트웨어를 연구하는 프로젝트도 있다.

<표1>프레임워크 프로그램의 OSS지원 프로젝트

Domains	Projects(number and acronym)
Software technology and engineering, development tools	34717 AMOS(free software package and component indexing and search), 32360 ARCHWARE(software architecture for evolvable software), 29380 GENESIS, 28402 OPHELIA(free/open source software

	development environments)
Platforms	29285 ACEOS(Linux port for Tricore processor), 28503 OPENROUTER(open source software router for SoHo market), 28152 PENGUIN-PC(RTAI-based industrial controller software), 32316 INES(Cluster of Embedded systems)
Middleware and reference implementation of standards	34546 PUBLICVOICEXML(reference implementation of VoiceXML), 37610 MIDAS(under negotiation, includes open source middleware), 37126 ADAPT(under negotiation, open source middleware for composable networked services)
Security, Public Key infrastructures and their applications	34340 EUPKI(free software public key infrastructure), 35174 OPEN-EVIDENCE(open source document authentication and time stamping), 29289 SECRETS(trial of open source security software) Generic applications for administrations 35176 ASWAD(free software workflow tools for administrations)
Generic applications for health	26162 SPIRIT(open source health care initiative), 25429 SMARTIE 37711 OPENECG(Effective Computational Geometry for Curves and Surfaces), 34512 OSMIA(open source medical imaging software), 10345 PICNIC
Generic applications for education	26249 ITCOLE
Generic applications for tourism	13015 FETISH, 20447 E-TOUR, 20147 CRUMPET
Scientific and technical software	31064 OROCOS(robot control software), 26473 ECG and 30078

	EPISEM ACTION (Epicentre Shared Earth Model Activity Collaboration through Meta Data Interoperability over the Net)
Media technology	34879 AGNULA(specialised free software distributions for audio/music)
Socio-economic studies	29565 FLOSS(statistics and economics of free/open source software)

IDA 프로그램은 정보통신기술을 활용하여 회원국 행정부간의 신속한 전자정보교환을 지원하기 위한 프로그램으로서, 공공부문의 소프트웨어의 호환성 확보, 시민참여의 용이성 증대, 소프트웨어 공동 활용 촉진 등을 지향하고 있다. 2002년 6월 IDA 프로그램은 유럽 연합의 공공부문에서 창출된 소프트웨어와 지식들을 공유할 수 있는 방안을 탐색하면서, 각 회원국의 공공분야에서 개발된 소프트웨어를 재활용할 수 있는 기본 조건으로서 오픈소스 방침을 제안하였다.4 여기에는 각국 공공부문에서 개발된 소프트웨어를 공유하기 위한 법제도, 기능, 기술적 설계, 유지보수 방안, 자금조달 등 소프트웨어를 공유할 수 있는 포탈에 대한 구체적인 시스템 설계를 수행하고 있다.

가. 유럽연합의 지원 프로젝트

1) 프레임워크 프로그램

EU차원의 공동연구개발사업인“제6차 프레임워크 프로그램”의 일환으로 공개소프트웨어 기술개발 및 활용촉진 프로젝트, 공공부문에서 공개소프트웨어의 구현 경험을 공유하고 지원하기 위한 프로젝트, 공개소프트웨어 현황과 정책을 연구하는 프로젝트들이 추진되고 있다.

2) 공개소프트웨어 기술개발 프로젝트

현재 5차 프레임워크 프로그램을 뒤이어서 약 30여개의 공개소프트웨어 기술개발프로젝트가 추진되고 있다. 유럽집행위원회가 자금을 지원하는 다양한 프로젝트들에 유럽의 기업과 연구소, 대학들이 참여해서 보안과 e-러닝, 개발 툴과 미들웨어와 같은 공개소프트웨어를 개발하고 있다. 프로젝트 목록은 http://www.cordis.lu/ist/ka4/tesss/impl_free.htm에서 살펴볼 수 있다.

3) COSPA 프로젝트

COSPA(Consortium for Open Source in the Public Administration) 프로젝트는 유럽 공공부문에"공개 데이터 표준(Open Data Standards)"과 공개소프트웨어 도입을 연구하고 지원해서 공공부문의 생산성 향상과 사무자동화를 제고하는 것을 목표로 하고 있다. 이 프로젝트가 추진된 배경은 유럽의 여러 지방정부에서 공개소프트웨어가 활용되고 있지만 그 활용현황, 문제점, 해결방안 등 관련 정보들이 종합 관리되고 있지 않아 그것을 정리하고 체계화할 필요성이 제기 되었기 때문이다.

사업은 유럽지역의 多언어 환경에서 접근할 수 있는 지식과 경험을 축적하고 지역 차원 혹은 유럽차원에서의 워크숍을 개최하여 축적된 경험과 지식을 확산시키는 데 있다. 현재 이 프로젝트에는 이탈리아, 헝가리, 덴마크, 영국, 아일랜드, 벨지움 등 유럽 15개국 소속 기업, 대학, 공공기관 등이 참여하고 있으며 캐나다, 뉴질랜드의 대학, UNESCO 등이 옵서버로 관계하고 있다. 이 프로젝트의 주관기관은 Free University of Bolzano - Bozen, Italy이다. 이 프로젝트는 2004년 1월부터 2005년 12월까지 2년 동안 추진되는 프로젝트로서 총 400만 유로가 소요되는 데, 이중 260만 유로는 유럽집행위원회가 제공하고 있다.

관련 웹사이트는 http://www.cospaproject.org 이다.

4) CALIBRE(Coordination Action for Libre Software) 프로젝트

이 프로젝트는 유럽 각 지역에서 이루어지고 있는 공개소프트웨어 연구개발활동을 지원하고 그 결과를 종합하여 공개소프트웨어의 연구 방향을 제시하는 것을 목표로 하고 있다. 그 동안 유럽의 공개소프트웨어 개발활동은 고립 분산되어 이루어져왔는데, 그것을 연계하고 종합하기 위한 프로젝트로서의 의미를 지니고 있는 것이다. 또한 유럽차원에서 공개소프트웨어의 확산을 통해 그것을 활용하는 산업의 경쟁력을 강화시키는 것도 중요한 목표로 설정하고 있다.

이 프로젝트는 유럽 공개소프트웨어 산업 포럼(European Open Source Industry Forum: 포럼명칭 Calibration)을 구성하여 유럽 차원의 공개소프트웨어 정책을 조정하고 공개소프트웨어의 베스트프랙티스를 유럽 산업계에 확산시키며 유럽 차원의 공개소프트웨어 연구와 활용을 조정하고 통합하는 역할을 수행할 계획이다. 특히 유럽 공개소프트웨어 산업 포럼은 공개소프트웨어와 그 방법론을 활용해서 자동차, 통신, 가전 등 유럽 산업 전반의 경쟁력을 향상시키는 것을 목표로 하고 있다. 프랑스, 이탈리아, 아일랜드, 네덜란드, 폴란드, 스페인, 스웨덴, 영국, 중국의 산업 및 대학의 연구팀이 참여하고 있는 민간주도의 프로젝트로서 2004년 9월부터 2006년 8월까지 추진되는 이 프로젝트에 유럽집행 위원회는 150만 유로를 투자하고 있다. 이 사업의 주관기관은 University of Limerick, Ireland이다. 관련 웹사이트는 http://www.calibre.ie이다.

5) FLOSS-POLS 프로젝트

이 프로젝트는 공개소프트웨어에 대한 정부정책을 종합적으로 연구하는 사회과학 프로젝트로서 공개소프트웨어 정책개발을 지향하는 프로젝트이다.

주관기관은 University of Maastricht – MERIT이다. 이 프로젝트는 5차 프레임워크에서 이루어졌던 공개소프트웨어 현황 조사프로젝트(FLOSS 프로젝트)의 후속 프로젝트로서 정책개발에 초점이 맞추어진 연구이다. 이 프로젝트에서는 공개소프트웨어 정부정책의 평가, 공개소프트웨어 개발에서 성차별 문제, 공개소프트웨어 개발에서 나타나는 협력적 개발 모델에 대한 연구들을 수행하여 공개소프트웨어를 활성화하기 위한 정책대안 개발을 지향하고 있다. 2004년 3월 1일부터 2006년 2월 28일까지 수행되는 동 프로젝트의 웹사이트는 http://www.flosspols.org이다.

나. 영국의 OSS 지원정책

유럽연합은 2000년 6월 오픈소스 소프트웨어의 지원과 관련하여 "2001년 유럽 위원회와 회원국은 IST 및 IDA 프로그램을 통해 유럽연합 전 지역에 걸쳐 경험을 교환함으로써 공공 영역 및 전자정부 베스트 프랙티스(Best Practice)에서 오픈소스 소프트웨어의 사용을 강화할 것"이라고 밝혔었다.1 이에 대하여 영국은 전자정부 상호호환성 프레임워크(e-Government Interoperability Framework)에서 개방 표준 및 명세서(specifications)를 사용할 것을 강제해 왔으며, 이를 지원할 수 있는 시장주도 제품을 허용해 왔다. 최근 OeE(Office of the e-Envoy) 및 OGC(Office of Government Commerce)가 영국 정부 내에서 OSS의 사용에 관한 명확한 정책을 가질 필요가 있다고 보고 이에 관한 정책을 발표하였다. 이하에서는 이러한 발표의 배경 및 정책의 내용을 구체적으로 살펴보고자 한다.

1) QinetiQ 보고서

영국정부는 EU의 OSS관련 정책에 대응하기 위하여 외부 전문기관에

OSS가 영국정부에 미치는 영향에 관한 분석을 의뢰하였으며, QinetiQ의 보고서는 그에 대한 결과이다. 주요한 내용은 다음과 같다.

⊙ 상호호환성 문제의 해결

- 정부가 무료로 제공되는 독점 프로토콜이나 데이터 포맷에 과도하게 의존하는 것은 상호 호환성 측면에서 많은 위험을 내포하게 되는데, 이는 개방 데이터 표준을 선택적으로 선택함으로서 적절히 조절할 수 있다. 예컨대 인터넷 분야에서 수많은 개방 표준들이 존재하고 있다. 데이터 표준에 관한 오픈소스 소프트웨어 참조 구현(Reference Implementation)의 존재는 그러한 표준의 채택을 가속화시킬 수 있으며, 정부도 선택적으로 OSS 참조 구현을 지원할 필요가 있다.

⊙ IT 인프라기술에 대한 영향력 증대

- OSS의 등장은 미국 이외의 시장 참여자들로 하여금 보다 쉽게 IT 인프라기술의 미래를 결정하는데 참여할 수 있도록 한다. 정부지원 소프트웨어의 활용을 위한 수단으로 미국이 OSS를 성공적으로 활용한 사례로 볼 때, 영국 정부도 정부가 지원한 소프트웨어의 이용을 위한 기본적인 방법으로 OSS의 사용을 고려할 필요가 있다.

⊙ 보안성 문제

- 국가 IT 인프라의 취약성과 관련하여 OSS의 장단점에 관한 뜨거운 논쟁이 있어 왔다. 하지만 국가 IT 인프라의 취약성을 증진시키거나 혹은 감소시키는 것과 관련하여 보고서는 OSS와 독점 소프트웨어의 차이는 주요한 요소가 되지 않는다고 결론짓고 있다.

⊙ 소프트웨어에 대한 권리의 취득

- OSS는 소프트웨어의 소스코드에 접근함으로써 소프트웨어의 개선 문

제(Legacy Problem)를 상당히 감소시킨다는 것을 보여주었다. 따라서 정부가 패키지 소프트웨어의 커스터마이징을 포함하여 주문(Bespoke) 소프트웨어를 구매할 때 그에 관한 완전한 권리를 취득할 필요가 있다.

⊙ 소프트웨어에 대한 지원모델의 변화

- 오픈소스 모델은 사용되는 소프트웨어를 지원하는데 새로운 패러다임을 제공해 준다. 다른 나라에서도 이러한 프로젝트들이 추진되고 있는데, 영국 정부도 그러한 소프트웨어에 대한 OSS 접근법의 유용성을 테스트하기 위한 파일럿 프로젝트를 고려할 필요가 있다.

2) 정부의 정책 방향

보고서는 서버 인프라(Server Infrastructure) 분야3의 정부구매 시장에서 OSS의 사용과 관련한 어떠한 정책을 제시하는 문제에 대해 상당히 조심하고 있다. 데스크탑 분야에서는 아직 OSS가 성숙하지 못했기 때문에 OSS를 선호하는 정책을 명확히 할 이유가 없지만, 서버 인프라 분야에서는 OSS가 이미 기술적으로 유용한 대안이 되고 있기 때문이다. 보고서는 정부가 서버 인프라 시장에서 OSS를 일반적으로 선호하는 것을 명확히 함으로써 얻을 수 있는 이익은 없다고 하면서도, 정부가 가이드를 제시하지 않는다면 입찰자의 편견 때문에 OSS가 비용 대비 효과 면에서 최고의 선택사항인데도 선택되지 않는 상황이 있다고 분석하고 있다. 최근 일련의 언론보도는 영국 정부가 마이크로소프트의 솔루션을 선호한다는 인상을 입찰자들에게 주고 있는데, 정부는 마이크로소프트 제품이 선호되는 상황에 대해 입장을 명확히 해야 함을 지적하고 있다. 또한 몇몇 독점 제품은 이후에 다른 공급자의 제품을 구축하는 것을 어렵게 만들고 있는데, 정부는 독점 프로토콜에 종속되는 위험을 어떻게 막을 것인지에 관한 정책을 명확히 해야 함을 밝히고 있다.

⊙ 정책내용

- 영국정부는 IT 구매에 있어서 독점 소프트웨어와 함께 OSS 솔루션을 고려할 것이다. 계약은 가치/비용의 기준에 의해 이루어질 것이다.
- 영국정부는 앞으로의 모든 IT 개발과정에서 개방 표준 및 명세서를 지원하는 상호호환성을 가지는 제품만을 사용할 것이다.
- 영국정부는 독점적인 IT 제품과 서비스에 종속되는(lock-in) 것을 피하기 위해 노력할 것이다.
- 영국정부는 정부가 구매하는 주문형(bespoke) 소프트웨어 코드 또는 COTS(Commercial Off The Shelf) 소프트웨어의 커스터마이징에 관한 완전한 권리를 획득하는 것을 고려할 것이다.
- 영국정부는 OSS를 정부가 투자한 R&D 소프트웨어의 기본적인 활용방법으로 사용하기 위한 가능성을 개발해 갈 것이다.

⊙ 정책의 근거

이상과 같은 정책을 추진하는 이유는 다음과 같다.

- 솔루션을 구매할 때는 항상 가치/비용 기준에 의하여야 한다. 그것은 OSS 솔루션일 수도 있고 독점적인 솔루션일 수도 있으며, 또는 양쪽이 혼합된 형태일 수도 있다. 결정은 개별 사례에 따라 독자적으로 이루어져야 한다.
- 언제나 시스템의 상호호환성이 제공되고 유지되어야 한다. 전자정부 상호호환성 프레임 워크(e-Government Interoperability Framework)는 공공영역 전체를 통해 사용이 강제되고 있으며, 이와 적합성을 유지하는 것은 정부의 전자 서비스 제공에 필수적이다.
- 정부의 시스템에 대한 비용 및 위험을 감소시키기 위해 모든 노력이 이루어져야 하며, 이러한 정책을 채택함으로써 그러한 목적을 달성할 수 있다.
- 정부 시스템의 보안은 필수적이다. 적절하게 짜여진 OSS는 적어도 독점적인 시스템만큼 안전하며, 현재 인터넷 공격도 보다 적다. 보안관리 기술의 유용성과 많은 다양한 시스템의 장점 사이에 적절한 균형점을

가질 필요가 있다. 몇몇 경우에서 주류의 독점적인 제품이 OSS 보다 덜 안정적인 것으로 나타났다.

⊙ 향후 계획

이상의 정책을 구현하기 위해 다음과 같은 조치가 이루어질 것이다.

- OGC는 이러한 정책을 반영하여 구매 가이드라인을 갱신할 것이다.
- 소프트웨어 인프라 및 응용 시장에서 구매활동에 참여하고 있는 모든 사람들에게 OSS의 장점과 단점에 관한 적절한 권고가 이루어질 것이다.
- 구매에 있어서 OSS와 독점적인 솔루션의 장점을 어떻게 평가할 것인지에 관하여 관련분야에서 활동하고 있는 사람들에게 적절한 조언이 이루어질 것이다.

⊙ 지적재산권 문제

정부의 새로운 OSS 정책은 "영국정부는 정부가 구매하는 주문형 소프트웨어 코드 또는 COTS(Commercial Off The Shelf) 소프트웨어의 커스터마이징에 관한 완전한 권리를 획득하는 것을 고려할 것이다"고 명시적으로 밝히고 있다. 이와 관련하여 지적재산권 관련 문제를 정리할 필요가 있다.

어떤 소프트웨어에 대한 완전한 권리를 획득하기로 하는 결정을 내리기 전, 구매자들은 어떠한 권리가 존재하는지, 있다면 어떤 권리인지, 그리고 그러한 권리를 획득했을 때의 이점은 무엇인지를 명확히 할 필요가 있다. 이러한 결정은 다음의 세 단계 과정을 통해 이루어지는 것이 바람직하다.

- 1단계: 기술적인 평가

기술적인 분석은 프로젝트 진행 중 어떠한 것이 실제로 쓰여지거나, 창조되거나, 개발되는지를 확인하는 것과, 그것이 새롭고, 창조적인 것인지를 결정하는 것을 포함한다.

- 2단계: 법적인 평가

어떤 무형의 권리를 "지적재산권"으로 총칭하여 언급하는 것이 편리하기는 하지만, 세부적인 내용을 살펴보면 이것은 많은 관점에서 서로 다른 법률들을 포함하는 집합적인 개념 (collective term)이다. 따라서 실제로 무엇이 쓰여지거나, 창조되거나, 개발되었는지, 그리고 어떤 종류의 지적재산권법이 관련되는지를 개별적으로 파악할 필요가 있다. 주로 저작, 디자인, 데이터베이스, 특허, 영업비밀, 노하우 등과 관련된다. 이러한 평가를 한 후에 다음과 같은 법적인 평가가 필요하다.

- 쓰여지거나, 창조되거나, 개발된 결과물 또는 자산 (assets)에 어떤 지적재산권이 존재하는가?
- 이러한 지적재산권을 어느 쪽이 소유하는가?

이러한 질문들에 대하여 개별 지적재산권법에 따라 다른 평가가 나올 수 있다. 법적인 분석은 세 번째 단계를 위하여 어떤 지적재산권이 관련되는지를 결정하는 것이다.

▪ 3단계: 재정적/경제적 평가

3단계는 지적재산권에 대한 비용을 지불하고 얻게 되는 이익과 지적재산권을 사적인 영역(private sector)에 남겨둘 때 얻게 되는 이익사이의 균형에 대한 적절한 분석을 포함한다. 이러한 평가는 프로젝트 단위별로 이루어져야 하지만, 고려해야 할 사항들은 다음의 내용들을 포함하게 된다.

- 공공영역이 지적재산권을 소유함으로써 얻게 되는 경쟁촉진의 효과
- 라이센스 조건에서 지적재산권이 가지는 상업적인 가치: 지적재산권의 잠재적인 라이센스가 존재하는지, 그들이 라이센스에 얼마를 지불하려고 하는지.
- 소유의 비용
- 등록비용
- 집행비용: 제3자가 침해할 경우 지적재산권을 보호하기 위한 조치
- 방어비용: 지적재산권의 성립에 관한 문제가 제기되는 경우, 또는 제3자의 지적재산권을 침해하였다는 주장이 있는 경우 이를 방어하기 위한 비용
- 관리비용: 지적재산권의 상업적인 이용에 관한 감시 (예컨대 라이센스가 라이센스 조건을 만족하고 있는지 여부)
- 지적재산권을 획득하는 내용 이외의 계약 조항을 통하여 수익을 발생시킬

수 있는 가능성, 예컨대 사적영역의 공급자와 제3자 수익 배분(a third party revenue distribution) 약정을 맺는 것.
- 사적영역이 지적재산권을 공공영역으로 이전하는데 부과하는 프리미엄; 반대로 이전하지 않는데 대한 할인.

구매자들은 구매과정 중 가능한 이른 시기에 구매과정에서 개발된 어떠한 소프트웨어에 대한 지적재산권을 획득하는데 이익이 있는지를 결정하여야 한다. 그러한 권리를 획득하는데 이익이 존재한다는 결정을 한다면 전문가로부터 법적인 또는 계약상의 조언을 구해야 할 것이다.

3) 종속성의 극복

독점제품에 종속되는 것은 여러 가지 원인에서 기인한다.

⊙ 경쟁업체들보다는 어떤 특정 공급업체의 제품이 현재 사용하고 있는 제품과 보다 잘 통합할 수 있을 것이라는 인식, 구매자들은 종종 현재 설치된 소프트웨어와의 호환성 문제를 이유로 무의식적으로 소프트웨어의 선택을 제한한다.

⊙ 할인을 이유로 특정 공급업체로부터 도입하려는 적절하지 못한 계약, 그러한 할인은 이후 종속을 극복하기 위해 지출해야 할 비용보다 낮을 수 있다.

⊙ 특정 브랜드 교육(Brand specific training), 이것은 현재 사용하고 있는 특정 제품 세트에 집중투자를 한 구매조직 내부 및, 자체비용을 감소시키기 위해 특정 제품세트에 표준화되어 있는 아웃소싱 파트너 모두에서 발생할 수 있다.

⊙ 상세한 분석을 하지 않고 전환비용이 상당히 높을 것이라는 가정을 하는 경향

⊙ 대체 제품세트를 확인하는 비용이 존재한다. 특히 시장 지배적인 독점 제품으로부터 OSS로 전환을 시도하는 경우 OSS에 대한 시장의 인식도가 광고에 의해 제공되는 독점적인 제품만큼 성숙하지 못했기 때문에 더욱 그러하다.

⊙ 미래의 구매약속을 바탕으로 공급자들이 유지비용이나 업그레이드 비용을 할인해 주는 경우가 존재한다.
⊙ 현재의 시장 지배적인 제품이 "최고의 선택사항"이라고 받아들이는 경향

구매부서는 현상의 유지(maintenance of the status quo) 또는 브랜드 리더십의 수용(acceptance of brand leadership) 또는 지배(dominance) 등의 문제에 대한 의문을 제기하여야 한다. 특히 구매자들은 종속을 극복하는 비용을 바탕으로 종속의 정도를 측정하여야 한다. 폐쇄적이고 독점적인 제품에 대한 계약이 끝난 후 전환하는 비용과 계약을 유지하는데 얻는 이익 사이에 균형을 맞출 필요가 있다.

4) 기타 OSS의 지원: NHS

영국에서 오픈 소스 소프트웨어에 대한 구현이 가장 활발한 분야는 국가보건시스템(national health care system)이다. 오픈소스 소프트웨어는 National Health Services(NHS)에서 일반 IT 솔루션으로 논의되고 있으며, 오픈소스 정책과 관련하여 NHS가 가장 활발한 활동을 하고 있다. 오픈소스 보건 응용프로그램(open source healthcare applications)은 현재의 폐쇄된 상업용 시장에 보다 많은 경쟁을 촉진시킬 것이며, 적합성 및 상호호환성을 유지하면서 혁신을 가져올 것으로 전망한다. 이것은 결국 적은 비용으로 양질의 프로그램을 구입할 수 있도록 한다.

EU의 공개소프트웨어 관련 프로그램은 크게 공동연구개발사업인'프레임워크 프로그램'과 공공부문 정보화와 관련된'IDA프로그램'으로 나누어서 살펴볼 수 있다.

1.1.3. 미국

미국의 공개소프트웨어 개발 정책과 관련해서 주목할 만한 것은 GOCC(Government Open Code Collaborative)(http://www.gocc.gov)의 설립이다. GOCC는 공공부문과 비영리 학술기관들이 개발한 코드들을 공유하기 위한 자발적 협력체이다. 유럽연합의 IDA가"Pooling Open Source Software"를 표방한 것처럼 미국의 주정부 등과 대학 등이 각자의 문제를 해결하기 위해 개발한 소프트웨어를 공유하는 협력체를 구성한 것이다. GOCC는 2004년 6월 30일에 출범했는데 메사추세츠 주, 로드 아일랜드 주, 펜실바니아 주, 웨스트버지니어 주 등 다양한 주정부와 MIT, 하버드와 같은 대학 연구기관이 참여하고 있다. 소프트웨어를 공유하기 위해 GOCC는 저장소(Repository)를 설치했는데 이는 코드 저장소로서 역할을 할 뿐만 아니라 소프트웨어 개발 베스트 프랙티스, 공공부문 관련 솔루션 등의 공유와 개발을 지원하는 역할을 수행하는 지원센터의 역할을 하고 있다.

1.1.4. 중국

중국은 국가 안보 확립과 기술자립국 실현이라는 정책적 목표하에 공개소프트웨어 개발 및 도입 확산에 적극적으로 정책을 펼치고 있다. 중국의 공개소프트웨어 관련 정책은 공공수요 확산을 위한 구매제도 정비와 표준화 등 연구개발 투자이다.

가. 공공 구매제도 정비

공공 도입 확산 정책은 2002년 6월 29일 개최된 제9기 전국인민대표대

회 상무위원회 제28차 회의에서 통과된 중화인민공화국 정부 구매법을 통하여 공식적으로 정부부처의 리눅스 사용을 장려하고 있다. 이 법은 자국의 물품, 프로젝트 및 서비스를 우선 구매해야 한다고 명시하고 있으며, 나아가 베이징시 과학기술위원회 등 각 성, 시 정부에서 중국 소프트웨어 제품을 우선적으로 구매할 것을 규정하는 후속조치가 뒤따르면서 공개소프트웨어 관련 정책에 가속도가 붙었다.

나. 연구개발 정책

2000년 6월 발표한 국무원 18호 문건에서 중요한 공개소프트웨어와 기초 소프트웨어의 개발을 지지한다고 명시함으로써 공개소프트웨어에 대한 정부지원을 공식적으로 밝혔다. 본 문건을 기초로 중국 소프트웨어 산업구조의 불완전성과 낮은 기술수준을 높이기 위해 선진기업과의 기술협력을 강화하기 위하여 2004년 3월 신식산업부와 HP는 Linux Software Lab을 설치하였다. 또한, 전자정부 건설을 위해 2004 중국정부정무전자화 투자를 위하여 400억위안을 투자할 것을 발표하였으며, 이 중 소프트웨어와 정보서비스 관련 투자는 140억위안이다. 이러한 전자정부화를 위한 IT 표준화에 대한 투자의 일환으로 중국전자기술표준화연구소(CESI) 주도로 전자정보산업발전기금을 사용하여 3년간 연 2500만위안을 투자하기로 발표하였다. 동 연구는 정부투자 및 민간연구소가 공동으로 연구를 진행하고 있으며 현재까지 주요 연구 결과는 'Linux 표준체계연구보고 초안','Linux API 규범 초안','Linux Operation System 기술요구규범 초안', 'Linux 서버시스템 요구규범 초안', 'Linux 사용자 Interface 규범 초안' 등 서버 시스템에서 사용자 인터페이스까지 다양한 표준화 연구를 진행하고 있다.

다. 지방자치단체

지방자치단체 중 북경시는 공개소프트웨어 특히 리눅스 도입 확산을 위

해 가장 적극적인 정책을 펼치고 있다. 북경시가 추진하는 리눅스 4대정책은 다음과 같다.

첫째, 리눅스 특히 데스크탑용 리눅스 도입 확산의 지속적 추진 둘째, 신식산업부와 과학기술부의 전폭적인 지지를 얻어 리눅스 연구성과를 실제 응용할 수 있는 방향으로 상업화지원 셋째, '1+1+1'Training Program을 통하여 Linux 소프트웨어 기업이 필요한 각종 Linux 전문인재를 교육시키고, 이를 통해 리눅스를 보급하기 위해서는 향후 3년내 300여명의 Training전문가, 1,000여명의 기술전문가와 1,000여명의 소프트웨어 운용전문가 양성 넷째, Linux 개발 단체인 OSDL(Open Source Development Labs)과 북경소프트웨어품질검증센터 (BSTC ; Beijing Software Testing Center)가 중국어 Linux 개발 협력을 진행하고 있다.

1.1.5. 일본

일본 정부가 공개소프트웨어에 주목하기 시작한 것은 2002년 후반기부터이다. 경제 산업성이 신성장산업에 대한 연구를 위해 구성한 '신성장정책부회'의 보고서인'창조적 산업구조의 구축'에 정보기술개발 조직의 네트워크 형태로 공개소프트웨어에 대해 기술하였으며, 공개소프트웨어를 소프트웨어 분야의 핵심기술로 인식하고 정부의 정책적 육성을 추진하기 시작하였다.

다음에서는 수요확산과 기술개발 정책을 주도적으로 추진하고 있는 경제산업성, 총무성 등 중앙정부와 지방자치단체의 정책 및 사업을 나누어 살펴보기로 한다.

가. 경제산업성

경제산업성은‘공개소프트웨어 활용기반정비사업', ‘미답(未踏) 소프트웨어 창조사업', ‘공개소프트웨어 데스크탑 도입 실증 실험' 등 공개소프트웨어 도입 및 개발 확산을 위하여 기술개발, 교육, 사용저변확대 등 다양한 분야에 걸쳐 정책사업을 추진하고 있다.

공개소프트웨어 개발 사업으로는 공개소프트웨어를 안심하고 활용할 수 있는 환경을 정비하기 위해 우수 소프트웨어를 공개소프트웨어로 제안 공모를 통해 발굴 지원하는‘공개소프트웨어 활용기반 정비사업', 개인 또는 그룹을 대상으로 차세대 IT시장 창출을 선도할 독창성 있는 연구자를 적극적으로 발굴함과 동시에 그들이 개발에 전념할 수 있는 환경을 정비하고 신시장을 개척할 소프트웨어 개발을 지원하는 ‘미답(未踏) 소프트웨어 창조사업', 효율적인 IT학습, 교육을 지원하기 위한 콘텐츠 및 소프트웨어의 기술개발을 지원하기 위하여‘교육정보화촉진 기반정비 사업', 정보가전산업의 시장선도, 상호운영성의 확보, 오픈 소스에 의한 운영체제, 미들웨어 등의 기술 실용화, 각종 기술과 서비스 조합에 의한 시스템 등의 설계, 제작 및 실증을 위하여‘정보가전협조기반 정비사업'을 추진하고 있다.

수요확산 정책으로는 공개소프트웨어 도입 확산 기반 조성을 위하여 2002년 11월부터 일본의 공개소프트웨어 이용현황 분석을 토대로 공개소프트웨어 도입 검토 지침 및 법적 과제를 정리하고 그 결과를 2003년 8월 공표하였다. 또한 동북아 국가간 공개소프트웨어 기술개발, 인력양성 등 정책공조를 위하여‘동북아시아 공개소프트웨어 포럼'에 적극 참여하고 있으며, 2005년도엔‘아시아 공개소프트웨어기반정비사업'으로 사업명칭을 변경할 예정이다.

나. 총무성

총무성은 e-Japan 중점계획에 근거하여 전자정부, 전자자치체 구축이

진행되고 있으며, 프로젝트 추진을 위한 조사 연구를'조사연구회'를 통하여 추진하고 있다. 동 연구회는 전자정부, 전자자치체 등의 시스템에 공개소프트웨어 도입 위상 검토에 이바지할 것을 목적으로 공개소프트웨어 및 비공개소프트웨어에 관하여 보안, 운영, 비용 등 다양한 관점에서 장단점을 객관적, 중립적 평가를 실시하고 있다. 공동아웃소싱, 전자자치체 추진전략을 2003년부터 시작하여 자치체형 공개소프트웨어의 개발과 보급, 지방공공단체에 있어서 공개소프트웨어 도입 방법에 관한 조사연구를 실시하였다. 자치체형 공개소프트웨어 개발/보급은 개별(또는 협력)자치체에서 개발한 소프트웨어를 모든 자치체가 자유롭게 개선/이용할 수 있도록 개발하는 것이며, 지방공공단체에 있어서 시스템 개발의 현재 상태 파악과 과제의 정리를 행함과 동시에 지방공공단체가 공개소프트웨어 도입에 의한 시스템 개발을 유효하면서 원활히 실현하기 위해 검토해야 할 유의점의 정리와 도입 순서 등에 대한 조사연구를 진행 중에 있다.

다. 지방자치단체

중앙정부 공개소프트웨어 관련 정책 못지않게 지방자치단체들의 자체적인 공개소프트웨어 도입확산, 기술개발, 인력양성을 위하여 다양한 정책을 추진하고 있는데 그 중에서 특히 홋가이도, 나하시, 기후현, 스모토시가 가장 활발한 정책을 펼치고 있다. 홋가이도는 전자도청 추진을 위하여 주로 네트워크 시스템에 사용할 기본 소프트웨어로 공개소프트웨어를 채용하고 있다. 홋가이도 경제 산업국에서는 특정기업의 제약을 받지 않는 공개소프트웨어의 도입, 이용 촉진이 도내 IT기업의 요소기술이나 제품 개발력의 향상 등에 상승효과를 기대할 수 있고, 정보산업 클러스터 형성에 기여하였다고 평가하고 2003년 8월부터'오픈소스에 의한 도내 IT산업의 새로운 발전 시책 조사'를 수행하였다.

나하시는 지역 IT기업을 중심으로 설립된 NPO 법인인 OSPI(Open

Source Promotion Institute)를 중심으로 자치체 등의 공적기관과 제휴하여 공개소프트웨어를 이용한 지역 IT기업의 활성화 운동을 전개하고 있다.

나가사키현은 공개소프트웨어를 활용하여 자치제 시스템을 구축하였으며, 이는 일본에서 가장 선도적으로 공개소프트웨어를 활용한 전자자치제라고 할 수 있다.

기후현이 출자한 (재)소프토피아재팬은 2003년 7월 공개소프트웨어에 의한 소프트웨어 개발을 추진하기 위해 OSPCJ(Open Source Promotion Consocium Japan; http://www.ospcj.org)을 설립하고 컨소시엄 내에 공개소프트웨어 개발 커뮤니티를 운영하여 공개소프트웨어 개발환경의 제공, 오브젝트지향 시스템 디자인 향상, 네트워크형 협력개발에 의한 지시기 공유를 추진하고, 오픈 환경에서의 IT진흥을 목표로 설정하고 있다.

IT특구로 선정된 효우고(兵庫)현 스모토(洲本)시는 2003년 7월 오픈소스 이용 시스템의 개발지원 프로젝트 OSCA(Open Source Community in Awaji ; http://www.city.sumoto.hyogo.jp/tokku)를 개시하고 공개소프트웨어를 중심으로 한 주민지원시스템이나 지역경제 활성화 지원시스템 등의 구현을 추진하고 있다.

1.2. 우리나라의 지원 정책

우리나라는 외국보다 다소 늦어졌으나 정보통신부를 중심으로 공개소프트웨어에 대한 정책적 지원을 추진하고 있다. 정보통신부는 2003년 1월 주요 정책 중의 하나로 공개소프트웨어 이용 활성화를 추진한다고 밝혔다. 주요 내용은 공공기관 정보화 시스템 구축 시 공개소프트웨어 도입을 가로막는 입찰제한을 제거하고, 소프트웨어 구매 가이드라인을 제시해 공개소프트웨어 활성화의 제도적 기반을 만들겠다는 것이다. 이를 위해 한국 소프트웨어진흥원에 공개소프트웨어지원센터를 설립하고 공개소프트웨어에 대한 개발, 보급, 인력 양성 및 대외 협력 등 <표2>와 같이 종합적인 지원 정책을

추진하고 있다.

특히 임베디드 시스템 기술과 공개소프트웨어인 리눅스 기술이 결합하면 우리나라 정보통신산업이 한층 더 발전하는 계기가 될 것으로 기대하고 있다. 이를테면 국내 리눅스 사업체들은 대부분이 리눅스 서버 분야에 진출하고 있다. 리눅스 서버 분야는 많은 사업자가 참여하고 있어 서로 경쟁이 매우 심한 편이다. 이에 반해 임베디드 리눅스 분야는 정보가전, 셋톱박스, 휴대형 전화기, 산업용 장비 등의 임베디드 시스템에 수요가 늘고 있어 수익성 확보를 기대할 수 있다. 정보 가전이나 휴대전화 같은 분야는 대량 생산제품으로서 전통적으로 우리나라가 경쟁력 있는 분야라는 강점이 있다. 그간 정보통신산업의 축적된 기술능력과 리눅스 기술을 결합하면 임베디드 시스템 분야에서 단연 두각을 나타낼 수 있다.

<표2> 우리나라의 공개소프트웨어 지원 정책

구분	내용
공개S/W 기술 개발 지원 사업	▪ 리눅스 PC용 편리성 향상 기술 개발 지원 ▪ 유망 분야 공개S/W 기술 개발 프로젝트 추진 ▪ 공개S/W 개발환경조성 지원
공개S/W 이용 활성화 지원 사업	▪ 공공부문 선도 시범 사이트 구축 지원 ▪ 공공기관의 공개S/W 사용 환경 확보 ▪ 공개S/W 이용 촉진을 위한 홍보 지원 ▪ 공개S/W 활성화 법·제도 기반 조성
인력 양성	▪ 기반S/W 개발 인력 양성 지원 ▪ 선도 개발자 양성 지원 ▪ 개발 저변 인력 양성 지원
대외 협력 지원	▪ 해외 시장 진출 전략 수립 지원 ▪ 동북아 3국간의 공개S/W 표준 제정 및 공동 기술 개발 지원

우리나라의 공개 소프트웨어 확산의 장애요인은 수요 측면에 있어서는 시장 진입 장벽, 성공 사례 부족, 호환성 결여 등을 꼽고 있으며, 공급 측면에 있어서는 기술 지원 체계의 미흡, 전문 인력의 부족 등 을 들 수 있으나, 이러한 제약 요인은 일부는 사실이 나 대부분은 수요자의 잘못된 인식과 편견에서 비롯된 것이라 할 수 있다. 이러한 제약 요인을 해소하기위해 다음과 같은 주요 정책을 추진하고 있다.

첫째, 공개 소프트웨어 확산의 가장 큰 현실적 장애 요인은 특정기술과 제품을 명시하는 시장 진입장벽이라 할 수 있다. 공개 소프트웨어 시장진입 장벽은 특히 공공기관에서 정보시스템 구축을 위한 사업 발주시 제안요청서에 명시되게 되는데, 고의 또는 과실로 인하여 유닉스 등 특정 운영체제를 명시함으로써 리눅스로 대표되는 공개 소프트웨어 진입을 가로막고 있

으며, 나아가 국산 하드웨어에 대해서도 진입장벽이 되고 있다. 이러한 진입장벽을 해소하기 위해 지난 2004년 12월 정부혁신지방분권위원회는 '전자정부사업 공개소프트웨어 도입 권고안'을 통과시킨 바 있고, 정보통신부는 다양한 기술 가이드를 통해 불공정 경쟁 환경을 개선하고자 노력하고 있다.

둘째, 공개 소프트웨어가 성공 사례가 부족하고, 리눅스기반의 우수한 상용 소프트웨어가 부족하다는 인식이다. 이러한 인식은 대표적인 공개 소프트웨어에 대한 편견과 오해라 할 수 있다. 그 이유는 웹 서버, 파일 서버 등에서 이미 기술적 안정성을 입증 받았으며, 중대형 서버 시장에서도 도입이 되고 있고 IHV, ISV들도 주요 응용 솔루션을 개발 제공하고 있기 때문이다. 이러한 편견과 오해를 해소하기 위해 다양한 공개 소프트웨어 도입 성공사례를 발굴 및 전파하고 있으며, 공공부문을 중심으로 한'공개소프트웨어기반 정보시스템 구축 시범사업'을 통하여 기술적 안정성을 입증하고 있다. 또한, 다양한 공개소프트웨어기반 솔루션을 알리기 위해 '공개기반 기업과 제품정보 가이드'를 발간하고, DB화하여 정보를 제공하고 있다.

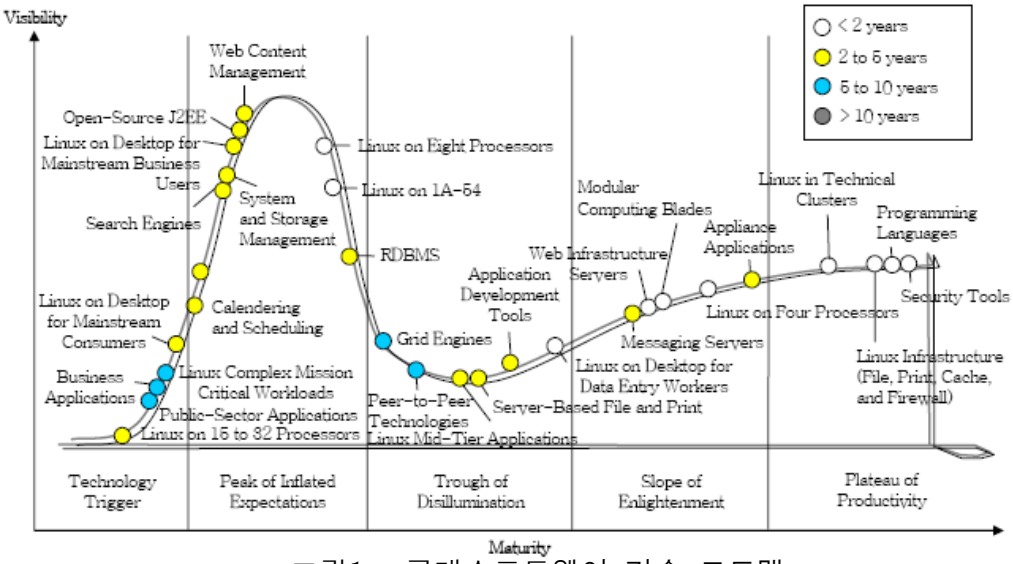
셋째, 공개 소프트웨어는 전문기업 및 전문 인력이 부족하여 제대로 된 기술지원을 받을 수 없다는 인식이다. 이는 공개 소프트웨어 수요 부족에 따른 인력공급 및 진입 기업이 부족하기 때문이다. 따라서 정부는 리눅스를 중심으로 전문인력을 양성하고, 한국소프트웨어진흥원 내에 '공개소프트웨어 기술 지원센터'를 설치하여 리눅스를 도입한 공공기관을 중심으로 온-오프라인 기술지원을 수행하는 동시에, 리눅스기반의 애플리케이션에 대한 안정성 테스트를 수행하고 있다.

넷째, 공개 소프트웨어는 다양한 기업 및 개발자커뮤니티를 통하여 개발되고 있기 때문에 호환성이부족하다는 인식이 있다. 특히 다양한 배포판의 난립으로 인한 호환성 결여를 해소하기 위해 ETRI와 민간기업 컨소시엄을 통해 표준 리눅스 스펙인 '부요(Booyo)'를 개발하고 있다.

1.2.1. 기술 지원정책

가. 공개 소프트웨어 기술 로드맵

2005년 가트너에서 보고된 <그림1>의 공개 소프트웨어 기술 hyper cycle 자료로 두 가지 측면에서 공개 소프트웨어 기술 발전 방향을 분석할 수 있다.



<그림1> 공개소프트웨어 기술 로드맵

첫번째로 시스템 플랫폼 관점이다. 서버 플랫폼의 경우는 지난 몇 년 동안은 1-4 프로세서 환경인 엔트리급(소형) 서버 환경에서 발전해 왔지만, 점차적으로 성능, 기능 개선, 보안 기능 강화로 64비트프로세서, 8 프로세서 이상의 중요업무를 처리하는 엔터프라이즈급(중대형) 서버로 확대되고 있다. 데스크톱 플랫폼의 경우는 제한적인 업무 사용자 환경에서 다양한 계층의 일반 사용자가 편히 사용할 수 있는 환경으로 기술이 발전할 것이다.

두번째 소프트웨어 계층적인 관점으로 리눅스 커널과 같은 최하위 계층의 기술 발전에서 미들웨어의 공개 소프트웨어화를 거쳐 응용 소프트웨어까지 공개 소프트웨어기술로 발전할 것이라고 분석하고 있다. 또한 공개 소프트웨어 기술은 개발자와 개발자들간에 소스를 무료로 공개하며 서로 간의 요구 사항을 반영하면서 기술이 성숙해진다. 즉, 기술 개발이 시작(technology trigger)하여 기술이 시장에 적용되기 전(trough of disillumination)까지는 개발자커뮤니티를 통해서 기술이 개발된다는 것이다. 이개발 과정 동안 기업은 기술개발에 필요한 비용을 투입하는 것은 아니지만, 여러 공개 소프트웨어를 가지고 제품화를 하는 단계인 통합, 시험 및 안정화과정에 소요되는 개발 비용을 투입하여 상용화(plateau of productivity)를 한다. 따라서 공개 소프트웨어기반 제품은 초기 개발 때부터 개발비가 투입된 상용 소프트웨어보다 다소 저렴한 비용으로 구입할 수 있다는 것이지 무료가 아닌 것을 지적하고 싶다.

1) 국내 표준 리눅스 규격, 부요

부요는 표준 규격서인 동시에 규격 구현물인 표준판이다. 사용상의 호환성 및 안정성을 확보하고 국외 배포판에 대등한 수준의 운영체제를 개발하는 것을 목표로 하는 부요를 개발함으로써, 다양한 분야에서 널리 공개 소프트웨어가 활용될 수 있는 신뢰감을 마련하고자 한다. 국내뿐만 아니라, 국외 여러 나라에서도 공개 소프트웨어를 활용하여 자국의 운영체제를 확보함으로써 자국의 소프트웨어 산업을 육성하고 주요 시스템의 보안과 관련 기반 기술을 확보하고자 추진중에 있다. 그 예로 중국의 RPLinux, 필리핀의 Bayanihan, 태국의 TLELinux, 말레시아의 Mimos Linux, 인도의 CDACLinux, 스페인의 bnuLixEX 등이다. 부요라는 단어는 날아오르는 펭귄을 상징한다. 우리나라 민요 가운데 까투리 사냥에서 중간에 까투리를 풀 섶에서 나와 날아오르도록 놀라게 하는 소리가 있다. 남쪽 지방에서는 '훠이여'북쪽 지방에서는 '우여'

그런데 중부 지방에서는 '부요'라고 외친다. 즉 부요라는 이름은 리눅스의 희망과 도전 정신을 나타내는 펭귄을 날아오르게 하자는 의성어이다. 부요를 통해 펭귄을 날아오르게 해 리눅스 산업도 크게 일어날 것이라는 희망을 담고 있다. 그래서 부요의 로고도 펭귄이 날아오르는 모습이다. 또한 부요란 말은 한자로 풍요롭다는 의미인'富饒'와 발음이 같아 소프트웨어 산업을 부유하고 넉넉하게 하겠다는 의도도 담겨 있다.

부요 플랫폼은 부요 데스크톱과 부요 서버로 구성된다. 부요 기술개발은 예전의 국내 배포판 제품처럼 국외 제품을 기반으로 한글화, GUI 변경만의 수준이 아니라, 명실 공히 국내 표준 운영체제로 대표할 수 있게 안정성, 호환성 결여 문제 극복뿐만 아니라 자체적으로 커널 및 미들웨어 부분의 핵심 기술개발에도 초점을 두고 있다. 이렇게 개발되는 핵심기술들은 유명 공개 프로젝트 사이트를 통해서 공개된다. 부요가 국내 표준 운영체제라고 국제표준을 부합하지 않는 것은 아니다. 2005년 12월 부요 데스크톱과 서버 규격 1.0은 한국정보통신기술협의 표준 승인을 거쳤고, 현재 규격 1.0의 시험 절차서를 마련하고 있다. 이와 병행하여 규격 2.0도 제정 중에 있다.

부요 표준판을 기반으로 하는 상용 배포판도 출시되고 있다. 계속해서 출시될 국내 상용 배포판들은 사용자 환경뿐만 아니라, 응용 소프트웨어 수행 환경에서도 완전한 호환성, 안정성을 보장할 수 있도록 부요 규격을 따르기를 희망한다. 그러면 그동안 국내 사용자가 느끼고 있는 공개 소프트웨어에 대한 불안감을 해소할 수 있는 계기가 마련되고, 국내 공개 소프트웨어 시장을 보다 활성화할 수 있기 때문이다. 부요 데스크톱은 사무, 제조, 교육 환경과 같은 제한된 응용만을 사용하는 환경에서의 편의성, 기능 개선을 우선적으로 목표로 한다. 그 다음 게임 등과 같은 일반 사용자가 편리하게 사용할 수 있는 데스크톱 환경까지 꾸준히 개선하고 발전시킬 계획이다. 현재의 부요 데스크톱은 익숙하게 사용되고 있는 데스크톱 환경보다는 부족한 점이 많다. 그러나 리눅스뿐만 아니라, 데스크톱을 구성하는 주요 공개 소프트웨어

기술이 지속적으로 개선되고 발전되고 있기 때문에 훌륭한 데스크톱 환경으로 빠른 시간 내에 부상될 것으로 생각된다. 부요 데스크톱은 다음 사항들을 고려하여 개발되고 있다.

⊙ LSB 등 산업 표준 규격을 기반으로 하는 개방형 구조 제공
⊙ 기존 대중적인 데스크톱 운영체제 대비 부족 기능 해결
⊙ 국내 사용 환경을 위해서 표준 한글 사용 환경과 한글처리 기능을 제공
⊙ 설치가 용이하고, 원격 업데이트 지원체제 제공
⊙ 데스크톱 환경 개선과 주변 장치 지원
⊙ GNOME 기반의 사용자 친화적인 데스크톱 환경 제공
⊙ 서버중심의 리눅스를 데스크톱용으로 가볍고 빠른 데스크톱 환경 제공
⊙ 오피스, 인터넷 뱅킹 등 사무업무용 환경 제공

<그림2> 부요 GUI 화면

<그림2>에서 볼 수 있듯이, 사용자에게 친화적인 환경을 제공하기 위해서 복잡하고 정리가 안 된 메뉴 구조를 개선하고, 아이콘 이름도 사용자가 어떤

응용인지를 쉽게 할 수 있도록 하였으며, 테마 및 용어 통일 등 전체적으로 통일성을 제공하였다. 부요 서버는 국제 산업 표준을 만족하고 고성능, 고가용성, 고신뢰성을 보장하면서 엔트리급, 엔터프라이즈급 서버 환경을 안정적으로 지원하는 것을 목표로 다음 사항을 고려하여 개발되고 있다.

⊙ LSB, FHS 등 국제 산업 표준 만족
⊙ 리눅스 기능 강화를 위한 CGL, DCL 규격 만족
⊙ 풍부한 서버 개발 도구 제공
⊙ 고부하 지속환경 및 서버 성능 벤치마킹에서 국외 제품과 동일 기능, 성능 제공
⊙ 공개 소프트웨어기반의 보안, 시스템관리, 서버 가상화, 웹, DBMS 등 솔루션 제공

부요 플랫폼은 이미 존재하는 공개 소프트웨어들 중에 소스가 공개되어 있고 많은 개발자가 참여하여 진행 중인 소프트웨어를 선정하여 개발한다. 그러나 부요 규격에 적합한 기능의 소프트웨어가 없는 경우는 직접 개발하여 국제 커뮤니티에 공개한다. 다음 웹 사이트는 대표적으로 공개된 리눅스 커널 기능들이다.

2) 커뮤니티 동향

앞서 언급한 공개 소프트웨어 기술들은 개발자 커뮤니티를 통해서 발전하고 성숙되고 있다. 물론 최근에는 국외 대형 기업의 공개 소프트웨어에 대한 관심으로 기술 개발의 투자를 점차 늘리고는 있지만 공개 소프트웨어의 기술 성장의 중심축은 자발적으로 구성된 커뮤니티이다. 이러한 커뮤니티는 북미와 북유럽에 많이 발달되어 있다. 한국을 포함한 아시아 지역은 부족한 부분이 많다. 개발자가 자발적으로 커뮤니티에 참가하여 많은 시간

을 기여할 수 있는 것은 공개 소프트웨어가 가진 철학을 공유하고 타인에 대한 기여를 통한 자기만족, 즉 소스 코드의 개발에 참가함으로써 얻어지는 명성과 기술적인 가치의 공유를 통해 자기 발전 등을 얻을 수 있기 때문이다. IBM에서는 공개 소프트웨어를 소스 코드가 공개 되어 있는 소프트웨어라고 정의하는 것보다 공개 협력(public collaboration)이라고 정의한다. 즉 상용소프트웨어의 개발 방식인 팀 체제인 계층적인 형태의 개발 모델이 아니라 네트워크 형태의 커뮤니티 방식으로 기술이 성숙해지는 것이다. 이 과정에 학력, 성별, 소속에 관계없이 개발 능력을 가지고 있는 어느 누구나 개발에 참여하여 동료 검토를 거치면서 견고하고 안전한 소프트웨어로 발전한다는 것이다.

국내의 공개 소프트웨어 커뮤니티 현주소는 프로그램을 만들고, 개량하고, 매뉴얼 등의 문서를 쓰고 번역하는 활동과 오프라인 모임과 세미나 등을 운영하는 것이다. 즉 국내의 커뮤니티 특성은 아직까지 질의응답을 중심으로 운영되고 있어, 활동이 국내에서만 머무르고 국제적인 활동에 동참하거나 기여하는 부분이 매우 미진하다는 것이다. 또한 국내의 공개 소프트웨어 생산율이 이미 존재하거나 개발 중인 소프트웨어에 대비 1%도 되지 못하는 점이다. 따라서 공개 소프트웨어를 잘 활용하는 정책도 중요하지만, 보다 다양하고 성숙한 기술의 발전을 위해 기술 개발에 기여할 수 있는 국내 개발자 커뮤니티의 육성도 시급하다. 국외의 경우는 대표적인 커뮤니티로 현재 130만 명의 개발자와 11만 개 이상의 개발 프로젝트를 운용중인 sourceforge.net을 들 수 있다. 그뿐만 아니라, 아파치, 파이션, KDE 등 수많은 주요 커뮤니티가 운영되고 있다.

본 단락에서는 국외의 다양한 커뮤니티에서 리눅스에 관련되는 커뮤니티의 동향을 살펴보고자 한다. 리눅스 개발 커뮤니티는 북미와 북유럽을 중심으로 300여 개의 커뮤니티가 존재한다. 그 중에서도 100여 개가 활발히 진행 중에 있다.

<표3> 리눅스 커뮤니티

	ubuntu	mandriva	opensuse	mepis	damn small	debian	slackware	gentoo	freebsd	vector	xandros	puppy	arch	Yellow dog
소스 공개	○	×	○	×	×	○	○	○	○	×	×	×	○	○
기반 시스템	Debian	from the scratch	RPM	Debian	Debian	from the scratch	from the scratch	from the scratch	BSD Unix	Slack ware	Corel	from the scratch	from the scratch	Fedora
의사 결정	기술 위원회	NA	노벨기업	1인	2인	투표	NA	관리자 협의	관리자 결정	관리자	회사내 관련조직	재단	Trusted User 결정	내부적
개발 커뮤니티	○	○	○	×	×	○	○	○	○	×	×	×	○	×
펀드 수단	회사	회사	회사	기부	기부	기부	회사	기부	기부	기부	판매	×	×	회사
규모	NA	NA	NA	1명	2명	1500명 이상	4명	330여명	350여명	7~8명	NA	19명	18명	NA
데스크톱	○	○	○	○	○	○	구분없음	○	○	○	○	○	○	○
서버	○	○	○	×	×	○		○	○	×	×	×	○	○
기본 데스크톱 환경	Gnome	Any	KDE	KDE	Flux Box	Gnome	Any	Any	Gnome	KDE, gnome	KDE	Gnome	Enligh-tement	Any
특성	사용편이	유럽형 리눅스	세계2위 리눅스	쉬운 데스크톱	Liveed	개발 시스템 체계화	Simple/ stable	Flexibili-ty가 높음	진행 프로젝트가 다수	SOHO	MS APP 호환	Liveed	Lightwe-ight and simple	ppc 전용
국가	Isle of Man	프랑스	독일	미국	미국	×	미국	미국	미국	캐나다	캐나다	호주	캐나다	미국

<표3>는 현재 활발히 운영 중인 리눅스 개발 커뮤니티를 정리한 것이다. 국내에 이미 잘 알려져 있는 젠투(gentoo), 데비안(debian), 페도라(fedora) 커뮤니티에는 수천 명의 참여자가 있는 반면에, 멤피스(mepis), 뎀스몰(damn small) 같은 커뮤니티는 운영자 혼자만 존재하는 경우도 있다. 이런 경우에도 커뮤니티 인기도가 상위에 존재하는 이유는 다른 커뮤니티에 비해 차별적인 요인이 있기 때문이다. 예를 들면, 세상에서 가장 사용하기 쉬운 데스크톱 리눅스를 만든다든지, 개발 주기를 짧게 하여 수시로 새로운 버전을 제공한다든지, 특정 플랫폼이나 특정 응용을 잘 지원 한다든지 등이다. 또한 규모가 작은 커뮤니티의 경우는 독자적인 운영 방식을 택하고 있지만, 대다수의 커뮤니티에서는 투표 등의 의사 결정 조직과 체계를 갖추어 운영을 하고 있다.

개발자를 위한 버그관리 및 소스 코드 관리 시스템, 개발자간의 의사소

통 방식도 구축되어 있다. 개발뿐만 아니라 사용자를 위한 공간도 아주 중요하다. 다양한 시험과 사용에 따른 버그 보고 및 요구사항을 제공하는 중요한 주체이기 때문이다. 따라서 소프트웨어의 다운로드 및 업데이트 사이트, 질의응답 및 공지 등에 필요한 공간을 마련하고 있다. 이처럼 공개 소프트웨어 기술 발전에 있어 커뮤니티의 중요성은 아무리 강조해도 지나치지가 않다. 커뮤니티를 통한 많은 참여자로 안정적인 소프트웨어를 생산할 수 있으며, 많은 기능들은 분담하여 개발함으로써 단시간에 다양하고 성숙한 기능을 생산할 수 있다는 것이다. 더불어 커뮤니티 활동은 훌륭한 인력 양성의 장이 될 수 있으며, 기술 확보로 창업의 기회를 마련할 수도 있다. 끝으로 향후의 공개소프트웨어의 비즈니스 모델에 있어 커뮤니티가 없는 제품은 사라질 것이고 존재하더라도 그 제품의 인지도는 떨어질 것이다.

1.2.2. 표준화 지원정책

가. LSB 규격

1998년 5월경 LSB 프로젝트는 리누스 토발즈, 상용 배포판 업체, 리눅스 인터내셔널, 그리고 Free-BSD의 지원 아래 세상에 나오게 되었다. 현재 LSB 프로젝트는 공개 소프트웨어 기술의 활성화를 지원하는 국제 비영리 컨소시엄인 FSG의 워킹그룹으로 진행되고 있으며, 2001년 6월 최초 규격의 발표를 시작으로 현재 3.0까지 발표하였다. LSB의 목적은 리눅스 배포판 간에 바이너리 응용의 호환성을 증대하고, 하드웨어와 소프트웨어의 이질성으로 발생하는 소프트웨어 개발과 이식의 복잡성을 감소시키기 위해서이다. LSB에서는 운영체제의 시스템 호출 인터페이스, 시스템 라이브러리, 패키지 포맷 등을 규정하고 있고, 대부분의 리눅스 배포판들의 기준으로 사용되고 있다. 하지만 바이너리 응용의 호환성을 유지한다면, 리눅스 커널이 사용되지 않아도 상관없다. LSB인증은 FSG의 인증 프로그램을 거쳐야

하며, 제품이 적용되는 인증 시험을 통과했을 때 제품과 관련되는 LSB trademark를 사용하도록 라이선스를 얻게 된다.

나. FHS 규격

FHS는 소프트웨어로 하여금 설치된 프로그램과 디렉토리의 위치를 예견할 수 있게 하고, 사용자로 하여금 설치된 프로그램과 디렉토리의 위치를 예견할 수 있게 한다. 1993년 8월부터 시작되어 대부분의 리눅스 배포판은 이 규격을 따르고 있으며, 현재 최신 규격 버전은 2.3이다.

다. CGL, DCL 규격

CGL, DCL 규격은 OSDL의 산업 요구기능 규격이다. OSDL 기관은 다양한 IT 시스템에 리눅스 기술을 적용시키는 역할을 하는 비영리 기관이다. 2000년에 설립되어 IBM, 인텔, HP, 노키아, ETRI 등 75개 회원사로 구성되어 있다. 리눅스 커널 개발자와 메인테인어인 리누스 토발즈와 엔드류 몰톤 등 45명의 직원으로 구성되어 있다. CGL, DCL 규격뿐만 아니라 OSDL은 DTL, MLI 요구기능 규격도 개발하고 있다. 요구기능 규격 제정은 고객과 벤더의 요구사항과 use case로부터 시작된다. 제안된 요구사항과 use case는 관련되는 내부 워킹그룹에 전달되어, 이미 존재하는 규격과의 중복성, 필요성을 검토한 후 규격으로 추가한다.

다른 규격서와 달리, OSDL은 규격의 구현물을 공개프로젝트로 진행하기 때문에 PoC에 대한 웹 사이트 정보도 함께 제공한다. 따라서 시간이 지나면 OSDL규격 기능에 대한 구현물은 공개 프로젝트로 완성되며, 벤더들은 그 결과를 제품화로 활용할 수가 있다. 하지만 OSDL은 표준화의 주체는 아니다. 산업체의 요구사항에 따른 문제(gap) 분석, 규격화 및 구현을 한다. 즉 OSDL은 기존의 표준들, LSB, POSIX 등을 준수하고 있다. CGL

규격은 리눅스를 통신 장비와 같은 실시간 성, 고가용성, 고신뢰성을 요구하는 시스템에 적용하기 위한 규격이다. 2002년부터 규격 개발을 시작하여 현재 규격 3.2 버전을 완료하였다. 규격 내용으로는 가용성, 서비스성, 표준, 성능, 하드웨어 지원, 보안, 클러스터링 등 7개 분야이며, 140여 개의 요구기능 규격으로 구성되어 있다.

2005년부터는 CGL 등록(registration)으로 유통되고 있는 리눅스 배포판의 CGL 규격 만족 정도를 시험하고 있다. 이것은 통신 사업자가 리눅스 제품을 선택하는 기준으로 활용되도록 하고 있다. DCL 규격은 리눅스를 데이터베이스 시스템과 같은 중요업무 시스템에 활용하기 위한 규격이다. 2002년부터 시작하여 현재 규격 2.1 버전을 완료하였다. 엄밀히 말하면 DCL 경우는 규격이 아니라 capability로 정의하고 있다. DTL은 데스크톱 리눅스 분야이다. 전 세계적으로 3% 정도의 점유율인 리눅스를 더욱더 활성화하고자 한다. 2004년부터 시작되었으나 활동이 미진하였고, 2005년부터 활발한 활동을 하고 있다. MLI은 모바일 시스템에 대한 규격이다. 2005년부터 시작되어 규격 작업이 진행 중에 있다. <표4>는 OSDL 활동을 정리한 것이다.

<표4>OSDL 활동

	Carrier Grade Linux	Data Center Linux	Desktop Linux	Mobile Linux
시작년도	2002	2002	2004	2006
목표시장	통신 인프라	엔터프라이즈 서버	데스크톱	단말기, 무선
목표기술	리눅스 커널, 미들웨어	리눅스 커널, 미들웨어	리눅스 커널, 미들웨어, 응용	커널, 미들웨어, 응용, 서비스 전달
회원 수	23	24	19	13
진행상황	요구사항 규격 3.2	Capability 규격 2.1	진행중	진행중

1.2.3. 법적 이슈 및 입법 정책

가. 지적재산권의 보호 및 제한의 정도

1990년대 중반 선진국들의 주도로 WTO/TRIPS(세계무역기구/무역관련 지적재산권협정, 1994년)와 WIPO(세계지적재산권기구)이 성립되었는데 새로운 지적재산권체제는 정보기술의 발달 및 정보재의 생산을 자본주의적으로 이끌어가는 구실을 한다. 핵심은 두 가지로 요약할 수 있는데 하나는 유통이 극히 쉬워진 디지털 정보의 저작권을 보호하는 것이며, 다른 하나는 인간의 유전 정보를 포함한 생명정보의 특허권을 보호하는 것이다. 지적재산권의 강화와 확대는 정보사회의 핵심인 정보자원 및 인간의 생명정보까지도 정보권리자-실질적으로 글로벌 대기업-의 이익을 더욱 보장하며 모든 것을 의존하게 하는 결과를 가져올 수도 있을 것이다. 이런 점에서 본다면 지적재산권문제는 단순히 법적, 경제적 차원뿐만 아니라 더욱 포괄적인 사회적 차원에서 검토되어야 할 것이라는 주장은 설득력이 있어 보인다. 비판정보 사회론의 입장에서는 정보재(情報財)는 자본주의적 상품으로 간주되며 이를 보호하기 위한 제도적 장치가 바로 지적재산권이라 주장한다.

정보의 상품화를 활성화하기 위하여 국가의 개입을 최소화하고 시장에 모든 것을 맡겨야 한다는 시장자유주의의 입장에서는 “저작권과 특허는 점차 그리고 가차 없이 사이버공간에서 낡은 것으로 될 것이며 앞으로는 암호기술이 개별적인 정보제공자들로 하여금 자신들의 디지털 재산을 보호할 수 있도록 할 것” 이라는 주장도 있다. 최근의 IT대기업들의 '신뢰받는 컴퓨팅'(Trustyworth computing)의 추진 동향과도 그 맥락이 닿아 있는 듯하다.'고도기술의 독점'의 폐해는 IT 분야에서 극명하게 나타난다. 기술의 '잠금효과'(lock-in)와 '네트웍 효과'(nrtwork effect)가 작용하여 하나의 공급자가지배적 위치를 차지하는 사례를 마이크로소프트 등 지금의 컴퓨터 업계의 현황에서 쉽게 찾아볼 수 있다.운영체계들 간에는 기술적 호환성이

없다. 따라서 하나의 운영체계가 지배적 지위를 차지하게 되면, 대다수 응용 프로그램이 그 운영체계에 기반 할 수밖에 없게 된다. 정보화가 진전됨에 따라 지배적인 위치 차지한 운영체계 및 미들웨어 등 기술 기반이 되는 부문의 지배력은 더욱 더 강화된다. 정보사회의 가장 중요한 기술적 기반 자체를 소수의 기업이 장악하게 되는 것이다. 이런 상황에서 사회성이 강한 분야에서는 지적재산권이 제한되어야 하지 않을까 하는 주장도 나온다.

정보사회주의의 입장에서는 정보·지식은 본래 '이용에 배타성이 없는 재화'로서 '공공재'에 해당하는 것이며 지적재산권법은 특정한 사회적 목적을 위해 이러한 공공재로서의 정보·지식을 사적인 재화로 변화시키는 제도적 장치라는 주장을 펴고 있다. 한편 시장 효율성은 정보의 자유로운 흐름을 요구하는 반면에 정보의 생산을 위한 인센티브는 정보의 흐름을 지체시키고 제한하는 일시적 독점을 요구한다. 결국 문제의 본질은 지적재산권에 내재된 재화로서의 정보·지식의 모순된 특성에 있다고 보여 진다.

정보·지식을 단순히 사적 재산으로만 취급하게 되면 당연히 효율성과 인센티브 간의 모순은 심화될 수밖에 없으며 마이크로소프트와 같은 독점의 폐해를 시정하는 것도 사실상 어렵게 된다. 그러므로 경제적 반독점의 견지에서, 더 나아가정보사회의 평등과 정의의 견지에서 정보재의 생산과 분배를 둘러싼 논의를 더욱 활성화할 필요가 있을 것이다. 이러한 논의의 바탕위에서 저작권법, 컴퓨터 프로그램 보호법, 특허법 등 IT의 지적재산권에 관련된 법률에 대한 장기적 관점에서의 개정 논의가 절실하게 요구된다.

지적재산으로 보호되는 알고리즘이나 특허를 사용하여 프로그램을 개발하고 공개한 경우 이를 개발하고 사용하는 다수의 사람들이 졸지에 범법자가 될 가능성이 매우 높다. 따라서 공개소프트웨어의 개발 의욕 고취와 실효성 있는 지적 재산권 보호를 위하여서는 국가에서 공개소프트웨어 개발 시 사전에 지적재산권 침해의 가능성을 최대한 예방할 수 있는 제도적 장치의 마련과 이를 위하여 필요하다면 입증 책임의 전환 등과 같은 법적인 조

치를 취하는 일이다.

한편 소프트웨어의 패러다임이 "제품에서 서비스"(from product to service)로 전환되어가고 있음에 따라 계약에서의 Service Level Agreement(SLA)가 도입되어 일반화 되면 지적재산권의 라이센스 비용과 관련된 문제는 자연적으로 해소가 될 것으로 예상된다. 물론 유틸리티로서의IT 사용에 대한 또 다른 법적인 문제가 발생하겠지만 지적재산권을 통한 정보의 독점을 둘러싼 논란은 어느 정도 불식될 수 있을 것이다.

나. 라이센스 모델의 선택

최근 조사에 의하면 전 세계적으로 24,000이상의 공개소프트웨어 중 GNUGPL이 73%, LGPL이 10%, BSD가 7%의 분포를 보이고 있다고 한다.11)라이선스 모델의 선택에 있어서는 정부가 구매자로서의 입장과 공개소프트웨어의 개발투자자 또는 시장조성자로서의 입장 그 어느 쪽에 서느냐에 따라 달라지게 될 것이다. 구매자로서 또는 공공을 위한 기술개발 투자자의입장에서는 GPL 쪽을 채택해야 할 것이고 경쟁력 있는 공개소프트웨어 사업자의 시장 조성을 위한 차원에서는 BSD에 가까운 라이선스가 정착되도록 지원해야 할 것이다.

다. S/W 특허

1) 알고리즘과 비즈니스모델의 특허

소프트웨어를 공공재로 보는 입장에서는 알고리즘과 비즈니스 모델의 특허의 폐지를 주장한다. 공개소프트웨어의 개발 및 발전에는 S/W의 특허가 걸림돌이 될 가능성이 높다. 따라서 독일에서의 특허법 개정 논의와 같이 비영리 사용자나 프로그램개발자는 해당 소프트웨어를 자유롭게 사용할 수 있도록

장기적인 관점에서 특허법 개정을 검토하는 것이 필요하다고 하겠다. 단, 이 경우 그 범위를 IT 분야의 기반이 되는 인프라 기술에 대하여 한정하는 것이 타당할 것이다.

2) W3C의 특허정책

표준화와 지적재산권은 그 개념과 특성상 갈등관계에 있다. 표준화는 기술의 공유 및 확산과 기술 상용화를 통한 보급 및 활용에 초점을 맞추고 있는 반면, 지적재산권은 기술의 사유·보호 및 창조적 발명과 혁신에 관한 인센티브 제도라 할 수 있다. 이러한 표준화와 지적재산권의 상충관계는 기술개발 및 시장경쟁이 매우 치열한 국제관계에서 극히 중요한 문제로 부각되고 있다. 이와 같은 상충관계는 해당 표준 채택의 필요성이 크면 클수록, 관련지적재산권으로부터의 배타적 이익이 높으면 높을수록 심각해지게 된다.W3C는 2001년도에 기업들이 중요 기술을 이치에 맞고 편파적이지 않을 것(RAND, reasonable and non-discriminatory)을 기초로 해 상용화 하는 것을 허용했다. RAND는 일반인들을 비롯해 프리 소프트웨어나 공개소스 운동에 참여하는 사람들에게 엄청난 비난을 받았으며 이에 대한 수정안을 마련하여2003년 5월 확정하였다.

새롭게 제정된 정책은 기업들이 자사의 특허를 보호할 수 있도록 허용하였으며 W3C가 규약을 제정하는데 꼭 필요한 기술에 대해서만 특허료를 부가하지 않도록 했다. W3C 로열티 프리 라이센스 요건은 일반적으로 인정되고 있는 오픈 소스라이센스 조항과 일관성이 있다. 로열티 프리 방침은 W3C 권고 그 자체가모든 유저 및 구현자에 대하여 진실로 제공된다는 보증을 제공한다. W3C특허 방침의 제 1 목표는 로열티를 자유롭게 사용하고 W3C 권고안의 구현을 가능하게 하는 것으로, 특허 기본 권리를 라이선스 한 데 대해 반드시 동의해야 한다.

3) 개방형 표준(open standard)

최근의 정보화 분야에서의 표준화는 어떤 시스템 운용 환경에서도 서비스 받을 수 있도록 하는 상호운용성과 서로 다른 아키텍처를 쉽게 통합하는 확장형 플랫폼를 확보하는 작업이 중요하게 인식되고 있으며 이를 위하여 개방형 표준의 채택과 실행이 중요한 이슈로 되어있다. 앞서의 오레곤주 법안에서는 공개 표준을 다음과 같이 정의하고 있다.

⊙ 다음은 컴퓨터 자료의 암호화 및 전송에 대한 규격을 의미한다.
모든 사용자가 읽고 실행할 수 있는 자료

- 사용자를 특정 판매자 또는 그룹에 얽매이지 않도록 하는 자료
- 적합성 인증을 위해 표준화 기관에서 요구하는 수수료를 제 외하고는 모든 사용자가 사용료(로얄티) 또는 비용을 전혀 지불하지 않고 구현할 수 있는 자료
- 구현물의 기술 표준 적합성 외에는 어떤 이유로든 특정한 한 구현자에게 상대적으로 유리하게 작용하지 않는 자료
- 확장성 구현을 금지하지 않는 자료. 단, 침탈적 행위를 통한 표준위반을 금지하는 라이센스 규정은 적용할 수 있다.

개방형 표준(open standards)이란 공개적으로 문서화되고 공식적 또는 사실의 프로세스에 의하여 산업계에서 받아들여졌으며 산업계에서 채택시 자유롭게 이용 가능한 인터페이스(interface) 또는 포맷(format)을 기술한 것을 말하며 다음과 같은 특성을 보유하고 있는 것이다.

⊙ 제약 없이 공개되며 비영리 산업기구에 의하여 통제
⊙ 산업계에서 자유롭게 채택 가능
⊙ 표준에 부합하는 구현 제품의 출시

Sun의 Jonathan Schwartz는 개방형 표준(프로토콜)이 오픈 소스보다 더 중요하다는 주장을 펴고 있다. 그 논거로서는 개방형 표준은 테스트 과정과 그 도구를 포함한 호환성을 제공하며 구매자와 이용자들의 선택을 가능하게

해주는 반면 오픈 소스가 반드시 이종 플랫폼간 호환성을 가져다주지는 않으며, 오픈 소스는 라이센스에 따라서 공개의 개념이 다르기 때문에 오픈 소스의 이점이 반드시 개방형 표준보다 낫다고 볼 수도 없다는 것이다. 양자에 있어서 진정으로 중요한 것은 공개와 관련된 커뮤니티가 그 배후에 자리 잡고 있으며 이해관계자와 새로운 아이디어의 제공자가 참여하는 프로세스가 충분히 보장되는가에 있다는 것이다. IBM도 최근에 소프트웨어에 대한 정책의 기조를 개방형 표준에 두고 오픈 소스를 적극적으로 지원하는 소프트웨어 전략을 수립하였다. “공공부문에서의 오픈컴퓨팅, 개방형 표준, 공개소프트웨어의 역할”이라는 제목의 백서는 공개성(openness)의 핵심은 유연성(flexibility)을 가능케 하는데 있다고 주장하며 공개성(openness)의 제반 목표를 다음과 같이 규정한다.

⊙ 유연성 보장
⊙ 상호운용성 보장
⊙ 공급자에의 고착 방지
⊙ 기술적 결정의 시민에의 부담 방지
⊙ 비용 효율성 도모
⊙ 정보 접근의 보장
⊙ 경쟁의 공정한 장 미련
⊙ 행동의 자유 최대화

구매정책과 OSS에 대하여는 OSS는 개발 접근 방법이며 오픈 소스를 기반으로 한 S/W가 상업적 S/W와 같이 취급되도록 하는 구매 정책을 권고하고 특정 방법을 선호하거나 강제하는 정책은 공개성이 추구하는 목표에 오히려 저해요인이 된다고 주장하고 있다. 백서는 기업과 정부는 오픈컴퓨팅, 개방형 표준, 공개소프트웨어를 포용(embrace)하여야 하며 구매정책은 실용적이고, 공개성(openness)을 고수해야하며 공개성의 목표를 고려하여야 한다는 결론을 내라고 있다. 그러나 오픈 소스와 개방형 표준과 밀접하게 연관되어 있

는 개념은 아니다. 즉 폐쇄형 표준과 프로토콜이 오픈 소스를 배제하는 것은 아니다.

대형 벤더들은 포맷과 프로토콜을 통제하고 싶어 한다. 그래서 그들은 MPEG 나 Open Mobile Alliance와 같은 산업체 컨소시엄은 물론이고 IETF 나 W3C 같은 개방형 표준단체에의 전면적인 참여를 통하여 공식적인 "표준" 제정 절차를 주도한다. 표준을 만드는 이유는 대부분 이익이 낮거나 경쟁사의 기술이 크게 위협이 될 때 특히 편리함을 위한 것이었다. 대부분 표준화 작업은 작은 회사들보다는 대기업들에 더 큰 이익을 가져다준다. 표준화로 인해 이득을 보는 것은 중소기업들이 아니라 대기업들이다. 만약에 어떤 소프트웨어나 칩 디자인이 공개되면 결국은 가장 저렴하게 그것을 판매할 수 있는 회사, 즉 대규모 공장과 연구개발 팀을 가진 대기업들이 이익을 챙길 것이다.

개방형 표준은 산업에의 적용 문제이고 오픈소스는 시민운동 차원의 문제로 서로 공개(open)이라는 개념을 보는 관점 자체가 다르다고 볼 수 있다. 공개소프트웨어는 그 자체로서의 의미가 존재한다. 즉 시스템의 투명성에 대한 요구사항의 충족, 비용(TCO) 측면의 이점과 독점적 소프트웨어를 견제할 수 있는 대안으로서의 기능이 그것이다. 소프트웨어 분야에서 그 기술의 근간을 이루는 운영체제와 인터넷 웹 브라우저는 이미 사회화된 상품이며 정보통신 분야의 근저를 이루는 규칙이자 표준이 되었다.

표준은 누군가에 의해 독점적으로 소유되어서는 안 되며 이는 사회 공공의 자산임이 분명하다. 그러나 작금의 글로벌 사회에서는 표준의 지배에 의한 사실상 독점의 해소가 쉽지 않은 것이 현실이다. 정부는 공공의 목적을 위해 사용되도록 표준 기술의 사회화 즉, 개방형 표준의 확산에 힘써야 하며 지적재산권의 보호를 통한 사적 소유의 강화와 아울러 일정한 범위의 소프트웨어 분야에서는 공개소프트웨어의 활성화를 통하여 이를 대체할 수 있는 선택가능성이 자리 잡을 수 있도록 정책의 균형을 기하여야 할 것이다. 또한 개방형 표준이 공개소프트웨어의 기반 위에서 자리 잡을 수 있도록 하기 위하여 필요한 경우

국가적인 기술 관리의 법적인 뒷받침이 필요할 수도 있다.

최근 정보통신부가 추진하는 정보기술관리 입법 노력은 이런 점에서 중요한 의미를 지니고 있다. 이 법안에 공개소프트웨어와 개방형 표준의 정의와 시행 원칙 등을 규정하고, 공공부문에서 개방형 표준의 대상-예를 들어 인프라 기술, 미들웨어 그리고 응용소프트웨어 중-을 공개소프트웨어에 의하여 대체할 수 있는 범위까지로 합리적으로 설정하고 비공개소프트웨어의 공개에 대한 인센티브를 부여하는 기술 정책 등을 시행하는 방안 등을 생각해 볼 수 있다.

라. 안전성 신뢰성 보장

OECD가 제시한 개인정보보호원칙 중 중요한 것으로 시스템공개의 원칙이 있다. 개인의 정보를 취급하는 시스템은 공개를 원칙으로 하여 개인정보침해 의 위해를 사전에 방지하도록 하여야 한다는 취지이다. 공개소프트웨어가 그 자체로서 중요성을 가지는 것은 어쩌면 바로 이점에 있다고 할 수 있으며 특정 시스템에 있어서는 경제적 논리를 떠나서 적용해야할 당위성이 존재한다고 할 수 있다.

'신뢰할 수 있는 컴퓨팅'(Trustworthy computing 또는 Trusted computing: 이하 TC라 약칭)의 문제도 공개소프트웨어의 지지자들은 비판적인 소리를 높이고 있다. Trusted Computing Group(TCG)은 MS, IBM, HP와 AMD 등 IT대기업의 연합(alliance)으로 이용자에게 안전하고 신뢰할 수 있는 컴퓨팅 제품과 서비스를 제공하는 것을 목표 내세우고 있다.

이를 비판하는 측에서는 그 속내는 DRM을 통하여 해적판 S/W를 방어하며 독점 규제를 피하여 교묘한 고착화(lock-in)를 도모하는 전략으로 보고 있다. 이들은 TC가 GPL을 붕괴시킨다고 주장한다. Linux 등을 기반으

로 한 중소S/W업체나 신규업체가 GPL에 근거하여 S/W를 개발할 때 TCG가 수용할 만한 평가 인증서를 획득해야 하는데 소요 비용과 그 코드가 제반 악의적인 공격에도 문제없음을 증명해야 하는 노력 때문에 사실상 GPL 적용이 어렵게 되는 문제가 발생하게 된다는 것이다. TC 자체에 대한 전면적인 부정은 어려겠지만 이러한 지적은 유념해야 할 필요가 있다고 생각된다. TC에 대한 장기적인 관점에서의 국가적 대응방안의 마련은 식량안보와 같은 차원의 IT안보의 측면에서 고려할 만한 일이라 할 수 있다.

마. e-business와 공개소프트웨어

e-business와 관련하여 공개소프트웨어 기반의 응용소프트웨어의 사용시 K4인증이나 공인인증서 규격의 개발 등의 문제나 Mozila, Netscape 등 인터넷 브라우저의 문제도 OSS 활성화 저해 요인으로 지적되고 있다. freebank.org는 마이크로소프트사의 운영체제와 브라우저에서만 사용할 수 있는 우리나라의 인터넷 뱅킹 환경이 개선되기를 희망하는 사람들이 모여 만든 프로젝트 사이트로 금융기관과 Mac이나 Linux PC에서의 금융거래를 가능케 하기 위한 운동을 벌이고 있다. 사회의 정보화가 전면화 되고 e-business가 사람들의 일상이 되었고 ubiquitous 사회로의 이전을 눈앞에 두고 있는 이때에 이러한 문제에 대한 공개소프트웨어의 입장에서의 대응도 필요하다.

바. 정보 접근권 문제

전자정부사업 등 공공기관의 시스템 구축 시 독점화되지 않은 웹 표준으로 시스템, 홈페이지를 구축하기 위한 고려가 필요하다. 정보공유를 위한 사회단체는 정부기관 홈페이지들이 특정 브라우저에 대한 독점적 지위를 인정하여 다른 OS와 다른 브라우저를 사용하는 네티즌에 대한 불평등한 차별을 하고 있으며 전자정부 홈페이지(www.egov.go.kr)는 리눅스, 매킨토시

사용자들은 주민등록번호, 주소, 아이디 중복 등의 확인이 불가능하여 회원, 비회원 가입조차 할 수 없다며 전자정부의 심각한 정보 접근권 문제를 제기하고 있다. 전자거래가 수반되는 전자정부 민원서비스에서의 공인 인증 문제와 함께 정보 접근권 확보를 위한 제도적 검토와 접근에 장애가 없는 시스템 구현을 기술적 노력도 수행되어야 할 것이다.

사. 입법 방향

공개소프트웨어의 활성화를 위한 법제도 개선의 주요 목표는 다음과 같이 정리할 수 있다.

⊙ 독점적 지위의 제품/서비스에의 고착화 방지
⊙ 국내 S/W 산업의 균형 발전
⊙ IT 부문의 창의적 개발 노력 증진
⊙ 투명한 업무처리와 개인 정보의 보호

앞서 검토된 내용으로부터 도출된 공개소프트웨어 활성화와 관련된 입법의 방향을 이러한 목표와 연계하여 정리해 보면 다음과 같다.

1) 독점적 지위의 제품/서비스에의 고착화 방지

⊙ 개방형 표준의 정착, 확산을 위한 국가 차원의 정보기술관리 입법
  ▪ 개방형 표준 원칙의 천명과 공공조달에 있어서의 인프라 분야의 소프트웨어의 소스 공개 유도
⊙ 공개소프트웨어에 조달 과정에서의 대한 공정한 기회 부여
⊙ 공개소프트웨어 라이센스에 의한 담보책임, 제조물책임 등의 합리적 설정 검토
⊙ 정보시스템 도입 관련 계약에서의 SLA 제도 도입
⊙ Trusted computing 등 기술 고착화(lock-in) 방지를 위한 법제도

검토

⊙ 현행 공공구매 프로세스에서의 공개소프트웨어 활성화 저해 제도 및 관행 해소를 위한 조치

- 운영 유지보수 비용 계약 규정 개선
- 소프트웨어의 보증과 제조물 책임 등에 관련 부분의 문제점 해소

2) 국내 S/W 산업의 균형 발전

⊙ IT 부문의 독과점에 대한 새로운 정의

- 기술의 결합, 사실상 표준 등

⊙ S/W 비즈 모델의 변화에 대응한 관련 법령 및 제도의 정비

- 공개소프트웨어의 사업 모델에 적합한 라이센스의 정부 구매에의 수용 및 적용 확산 지원
- 우리나라 공개 S/W산업의 잠재적 가능 분야에 적합한 라이센스 모델의 수용 또는 개발 지원
- 서비스지향의 소프트웨어 전략에 대응한 법령의 개선 정비(SLA 계약의 도입 등)

3) IT 부문의 창의적 개발 노력 증진

⊙ IT 부문의 창의력 증진을 위한 R&D 활성화 입법 정책 수립 시행

- S/W 및 비즈니스 모델의 특허 인정 여부 또는 범위(특허법의 개정, TRIPS의 검토)
- Open standard의 정착 및 대형 벤더 주도의 표준화를 지양(止揚)하는 제반 정책 및 입법에의 반영 검토

⊙ 국가가 투자하거나 개발한 S/W 소스의 공개

⊙ 오픈 소스의 기본 정신과 그 상업적 이용의 조화를 도출해 내는 제반 사항의 장기적 관점에서의 검토

- GPL과 지적재산권의 충돌을 야기할 수 있는 법률적 쟁점
- 공공부문에서의 오픈 소스 라이센스의 사용절차 및 우리 법과 공공 행정의 특수한 목적에 부합하는 여러 라이센스에 대한 최적 안 도출

4) 투명한 업무처리와 개인 정보의 보호

⊙ 시스템 공개의 원칙

- 개인정보보호 관련법에서의 원칙 반영
- u-Korea 관련 법제도에서의 공개소프트웨어 정신 반영 검토

⊙ 정보접근권 차원에서의 불합리한 법제도 개선

- 정보격차 해소에 관한 관련 입법 내용에 다양한 접근 채널 및 기술적 방법도 내용 요소로 포함하여 공개소프트웨어 이용자들의 권리 보호

OSS 활성화를 위한 개괄적인 입법방안을 활성화의 의의와 연관시켜 정리해 보면 <표5>와 같다.

<표5> OSS 활성화의 입법방안

OSS 활성화의 의의	정부 정책 방향	필요성	입법 방법
경제성-TCO 등	구매 프로세스의 개선	필요	국가 계약법 및 조달 관련 개정
시장경쟁의 확보	공정경쟁환경 조성	필요	독점규제 및 공정거래에 관한 법률의 개정
사적 독점 IT 제품과 서비스의 고착화 회피	정부 구매 정책에서 선택의 다양성 확보	불요	-
상호운용성 극대화	개방형 표준 정착	필요	정보기술관리법 제정
보안성, 안전성, 프라이버시	투명한 업무처리와 개인정보보호의 원칙 보장	필요	개인정보보호법 등에 반영
국내 소프트웨어 산업의 발전	전략 분야의 육성 기술개발	필요	소프트웨어 산업 진흥법 등 개정
IT 부문의 창의적 개발노력 증진	지적재산권제도의 합리적 개선	필요	지적재산권 관련법 및 특허법 등의 개정

1.3. 공개소프트웨어 도입의 위험 요소

1.3.1. 도입유형

공개소프트웨어를 도입하는 방법에는 크게 3가지 방법이 있다. 첫째 기관이 직접 선택하여 도입하는 직접선택, 둘째 민간기업 등 외부의 추천에 의한 간접공급, 셋째 기관내부에서 공개소프트웨어를 개발 및 수정하는 내부개발 등이다. 이들 3가지 도입방법들에 대한 자세한 설명을 하기에 앞서 공개소프트웨어를 도입함으로써 얻을 수 있는 기대효과와 장점은 어떤 것들이 있는가를 살펴보면 다음과 같다.

첫째, 소프트웨어의 유지관리의 지속성(영속성)을 보장 받는다. 원공급자가 폐업하여 사라지거나 지원을 중단하더라도 공개된 소스를 보유하고 있으므로 기관은 해당코드를 새로운 공급자에게 양도함으로써 정보시스템 유지관리의 지속성(영속성)을 보장받는다.

둘째, 소프트웨어 도입 및 정보시스템 구축비용을 절감할 수 있다. 독점 소프트웨어는 라이센스 수수료와 함께 서비스 및 기술지원 비용을 발생시키지만 공개소프트웨어는 라이센스 수수료에 대한 비용 대신에 서비스 및 기술지원비용만을 발생시킨다. 따라서 공개소프트웨어를 도입하면 도입비용을 현저하게 낮춘다. 또한, 유지보수 측면으로 보면 공개소프트웨어는 소스코드에 바로 접근하여 수정할 수 있는 권한이 확보되므로 유지보수 및 업그레이드에 투여되는 예산을 줄일 수 있다.

셋째, 높은 호환성을 보장받는다. 공개소프트웨어는 다양한 형태의 소스코드를 배포하고 있으므로 상대적으로 많은 플랫폼에 이식가능하며 또한 이와 같은 플랫폼 독립성(호환성)은 수요자에게 하드웨어 선택의 폭을 넓혀준다.

넷째, 패치와 업그레이드 주기 및 속도가 빠르다. 공개소프트웨어는 소스코드가 공개되어 있고 많은 개발자들이 함께 공동 작업을 하고 있으므로 보안취약성등과 같은 결함을 발견한 후 몇 시간 또는 몇 일내에 재빨리 패치 및 업그레이드가 이루어지므로 보다 안전하게 시스템을 운영할 수 있다.

다섯째, 독점소프트웨어를 공급하는 업체로 부터의 구속을 탈피할 수 있다. 공개소프트웨어는 동일한 제품에 대해서도 공급자가 다수업체가 될 수 있기 때문에 공급자의 불합리성에 보다 적극적으로 대처할 수 있으며 또한 독점소프트웨어에 대한 구속을 회피할 수도 있다. 즉, 독점소프트웨어의 경우에는 선택의 여지없이 사용해야 만하는 불합리성이 존재할 수 있지만 공개소프트웨어는 공급업체와 제품이 다수이기 때문에 이와 같은 구속은 존재하지 않는다.

여섯째, 소스코드의 수정 및 확장권한을 확보할 수 있다. 독점소프트웨어는 컴파일된 실행파일 즉, 이진파일형태로 제공되기 때문에 소스코드를 수정하거나 확장을 위한 변경작업을 할 수가 없으나, 공개소프트웨어는 소스코드가 공개되어 있기 때문에 현재 시스템에 적용되어 있는 소프트웨어의 수정과 추가 확장을 위한 변경작업이 얼마든지 가능하다.

일곱째, 공공부문의 정보시스템 통합이 용이하다. 정부 및 공공부문의 정보시스템 구축 시에 핵심모듈을 공개소프트웨어로 구축할 경우 공공부문의 개방표준화가 용이하게 되어 정보화 예산을 절감하고 호환성 및 이식성을 확보하여 공공부문 전체의 정보시스템 통합을 촉진시킬 수 있다.

여덟째, 수요자 권리를 확보할 수 있다. 공개소프트웨어 도입 확대는 소프트웨어산업이 공급자 위주의 시장으로부터 소프트웨어 수요자(사용자, 정부) 위주의 시장으로 전환하는데 중대한 역할을 할 수 있다. 독점소프트웨어 업체들이 그들의 소프트웨어를 공급하면서 독점지위를 유지하기 위하여 소스코드가 아닌 실행파일 형태로 제공하고 있다. 하지만 이것은 지극히 공

급자 위주의 관점이며 소프트웨어를 구매하는 수요자의 입장에서는 소프트웨어구매와 함께 소스코드를 요구할 수도 있다고 볼 수 있다. 단순히 소스코드만을 요구하는 것이 아니라 공급업체가 없어지거나 사라지더라도 도입한 시스템의 유지보수와 패치 및 업그레이드 등을 보장받으려면 소스코드가 공개되어 있지 아니하고서는 불가능한 일이기 때문이다. 따라서 소비자들이 이러한 유지보수, 패치, 업그레이드 등을 보장 받으려면 독점소프트웨어로는 그 한계가 있기 때문에 공개소프트웨어를 사용하여 그 권한을 보장받을 수 있는 것이다.

공개소프트웨어는 수요자 측면에서의 이러한 장점과 함께 국가 산업적 측면에서는 소프트웨어 개발 원천기술을 확보함으로써 지식기반경제 핵심산업인 소프트웨어산업의 경쟁력을 강화시키고, 최신기술의 공개, 개발자 커뮤니티 등을 통한 기술교육에 의하여 국내 소프트웨어제품의 선진제품과의 기술격차를 줄일 수 있으며, 빠른 신제품 개발 등 시장에 즉시 대응이 용이하며 개발의욕 고취를 통한 양질의 핵심소프트웨어 인력양성이 가능하다는 장점이 있다. 직접선택, 간접공급 내부개발 도입방법 이외에도 다양한 방법들이 존재할 수 있지만 공개소프트웨어를 도입하는 거의 모든 경우는 대부분 이 세 가지 범주 내에 속하기 때문에 이번 절에서는 이들 3가지 방법에 대해서 설명한다.

가. 직접선택

직접선택("내부조달"이라고도 함)이란 공개소프트웨어 리소스 사이트에서 사용자가 특정 공개소프트웨어를 직접 다운로드하여 활용하는 것을 의미한다. 물론 다운로드 행위자체가 기관 내부의 직원에 의해 수행되며 기관 내부에서 공개소프트웨어 관련 정보가 어느 정도 확보되어 있어야 가능한 방법이다. 이 방법은 비용발생이 거의 없다는 장점이 있는 반면 위험부담이

크다는 단점이 있다.

특정 공개소프트웨어를 직접 다운로드 한다는 것은 독점소프트웨어를 공급업체로 부터 구매하는 것에 비한다면 기술검증에 대한 책임을 스스로 부담한다는 단점이 있다. 즉, 다운로드하는 소프트웨어 내에 트로이목마와 같은 바이러스 코드가 들어있다거나 키로거(key-logger)와 같은 악성 소프트웨어에 의한 감염을 포함한 소프트웨어가 될 수 있기 때문이다. 그러나 세계적으로 사용빈도가 높은 공개소프트웨어를 공식 사이트로부터 다운로드받는다면 이러한 부담을 제거할 수 있으며, MD533) 등의 무결성 체크를 선행하는 경우 대부분 해결이 가능하다. 그리고 또 다른 문제점은 다운로드한 공개소프트웨어의 보상과 보증이 보장되지 않는다는 것이다. 즉, 공개소프트웨어 솔루션을 기관내부에서 직접 선택하여 도입하려고하는 경우에는 소프트웨어의 보증과 보상을 보장받지 못하므로 이에 대한 위험감소 대책을 고려해야만 한다.

이러한 위험에 대한 대책으로는 지역 내에 기술지원이 가능한 개발 업체가 하나이상 존재하는 공개소프트웨어를 선택하는 것이다. 이와 같은 대책을 마련해 둔다면 공개소프트웨어를 도입한 기관이 직접 해결하지 못하는 기술적 문제에 대해 안정성을 보장 받을 수 있다. 즉, 도입기관의 시스템 운영자는 대부분 매일의 주기적이고 반복적인 업무는 수행할 수 있다 하더라도 기술지원업체를 미리 확인하고 점검해 두는 것은 2중 3중의 안전장치를 가지는 것이며 시스템 가용성을 더욱 높이게 되어 위험요소를 감소시킬 수 있다.

<표6> 공개소프트웨어 도입 절차

직접선택 워크플로우	
1단계	공개소프트웨어 솔루션 조사
2단계	유력한 공개소프트웨어 솔루션 선택

3단계	솔루션의 목적에 맞는 적합성과 비용적 가치 검토
4단계	솔루션의 품질과 보안성 분석
5단계	예비 프로젝트 수행
6단계	예비 프로젝트 결과를 통하여 기본적 수준의 예측사항 검토
7단계	프로젝트 수행 계획 수립
8단계	프로젝트 수행

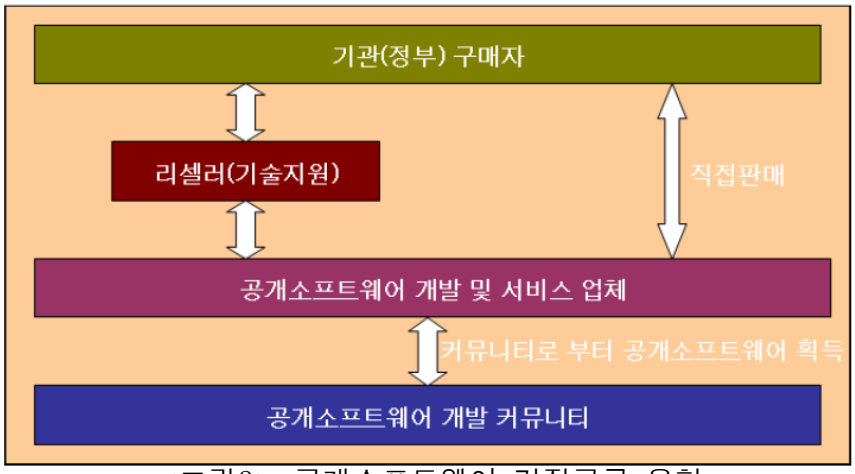
직접선택 방법으로 공개소프트웨어를 도입할 경우 다음의 절차를 따를 것을 권고한다. 특정 공개소프트웨어 제품이 기관의 요구조건을 충족시킬 경우 도입에 앞서 해당 기관에서 보다 상세한 검토를 수행하기로 결정하기로 하였다면 다음과 같은 체크리스트를 작성하고 평가과정을 거치도록 해야 한다.

<표7> 직접선택방법 도입 전 체크리스트

▪ 기관에서 사용되는 운영체제 플랫폼에서 해당 공개소프트웨어가 실행되는가?
▪ 해당 공개소프트웨어가 기존의 운영체제 플랫폼에 대해 획득, 시험, 배치해야 하는 추가적인 시스템 구성요서, 라이브러리 또는 모듈을 필요로 하는가?
▪ 해당 공개소프트웨어의 설치 절차가 이해하기 쉽고 명확하게 정의되어 있는가?
▪ 도입기관이 목적에 대한 적합성 여부를 결정할 수 있도록 공개소프트웨어를 설치, 배치, 시험할 수 있는 내부 전문지식을 갖추고 있는가?
▪ 해당 공개소프트웨어의 설치 및 제거 절차가 명확하게 정의되어 있는가?

나. 간접공급

간접공급("외부조달"이라고도 함)이란 공개소프트웨어를 업체로 부터 공급받는 것을 의미한다. 즉, 공개소프트웨어 공급 전문기업으로부터 특정 공개소프트웨어를 공급받는 것으로 기술지원과 서비스 비용이 발생한다. 비용이 발생하는 대신 직접선택 방법에 비해 위험부담을 줄일 수 있는 안전한 방법이라고 할 수 있다. 간접공급 방식으로 공개소프트웨어를 도입하였다면 라이센스에 관련된 법적인 위험요소에 대한 책임은 원칙적으로 공급업체에 있다. 하지만 이 경우에도 공급업체와의 계약서 내용에 따라서 법적인 책임 소재가 달라질 수 있으므로 이 부분에 대한 확인을 반드시 해야 한다. 즉, 기관은 기술적인 위험요소와 변경관리 위험요소를 감소시키기 위한 적절한 실사조사를 수행해야 한다. 즉, 기관이 공개소프트웨어 솔루션을 위해 간접공급 방식을 선택한 경우, 해당 업체 즉, 공급업체가 모든 기술 지원에 있어 책임을 져야하며 기관은 공급업체가 해당 공개소프트웨어에 대한 적절한 위험경감 절차를 수행함을 확인해야한다는 의미이다. 다음은 간접공급 방식의 몇 가지 공급유형을 나타낸 것이다.



<그림3> 공개소프트웨어 간접공급 유형

이 외의 사항은 비공개소프트웨어를 도입할 때와 고려사항 및 절차가 유사하다. 예를 들어 기관이 네트워크 및 시스템에 도입된 소프트웨어 보안 표준에 대한 정책을 이미 보유하고 있을 경우에 개발업체는 위험분석의 일환으로 그러한 정책을 반드시 준수해야 한다. 또한 모든 절차가 끝난 후 검수작업 시에도 보안 표준과의 일치성 여부를 확인하도록 해야 하며 계약서 내에 이러한 모든 절차들이 포함되어 있어야한다. 또한, 기관은 개발업체에게 모든 소프트웨어 구성요소에 대한 공급을 확인해야 하며 소프트웨어 출처와 라이센스 계약, 그리고 위험평가 문서를 확인해야 한다. 그리고 비공개소프트웨어를 도입할 때와 마찬가지로 도입된 공개소프트웨어의 전체적인 통합문제를 분석하고 이를 파악하여 반드시 문서화해야 하며 개발업체의 수행능력을 확인할 수 있도록 자체평가를 수행할 수도 있다.

비공개소프트웨어를 업체로부터 도입할 때와 마찬가지로 간접공급 방법은 개발업체의 적격성, 확장성, 소프트웨어의 성숙도 등에 관련된 분석이 선행되어야 하며 이러한 문제는 소규모 개발업체와 거래할 때 특히 신중히 검토되어야 한다. 따라서 앞서 설명한 바와 같이 잠재적인 위험요소를 줄이기 위하여 개발업체에 대한 적절한 실사조사 및 평가는 반드시 이루어져야하며 또한 평가 결과에 따른 조치가 이루어져야 한다. 다음은 이러한 평가에 반드시 포함되어야 할 내용들이다.

- 재정적 보장 및 안정성
- 위험관리의 적격성
- 내부 관리의 평가
- 업무 지속성 계획의 검토
- 일반적인 기능 개요
- 서비스 전달 및 관리

▪ 기타 기관의 특성에 맞는 평가요소

실제로 기관들은 공개소프트웨어 제품을 직접선택(내부조달)을 하기 위한 기술적인 능력을 갖추지 못한 경우가 대부분이므로 외부의 서비스 제공업체를 통해 공개소프트웨어 솔루션을 도입 받는 간접공급(외부조달)방법을 선택하는 것이 바람직하다. 그리고 잘 알려져 있고 인기 있는 대부분의 공개소프트웨어 제품들은 상용 솔루션 제공업체들을 통해 공급이 가능하다.

하지만, 외부의 어떤 업체를 통해 공급받더라도 공개소프트웨어와 비공개 소프트웨어(독점소프트웨어)의 공급에는 차이점이 있다. 즉, 기관이 외부 서비스 업체를 통해 공개소프트웨어 솔루션 또는 제품을 공급받는 경우 일반적으로 관련 공개소프트웨어는 무료로 제공받고 이에 대한 서비스를 구입한다. 이것은 라이센스를 구입하고 부가가치 서비스로 지원을 제공받는 비공개 소프트웨어(사유소프트웨어)의 경우와 차이가 있다. 그리고 기관이 외부 공급업체를 선택하기에 앞서 후보 업체들에 대한 실사로서 재정 상황, 안정성, 기술 능력 등을 평가해야 한다. 이러한 평가 작업들은 커뮤니티의 지원이 사라질 수 있는 위험한 제품선택에 대하여 위험을 줄여준다. 그리고 도입 후 어느 시점에 외부 공급업체가 사라지더라도 다른 공급업체에게서 지원을 계속 받을 수 있는 안정성과 유연성을 제공해 준다.

다. 내부개발

내부개발이란 기관내부에서 직접 개발하는 방법을 의미하며 공개되어 있는 수많은 공개소프트웨어들 가운데 도입하고자 하는 용도에 가장 적합한 공개소프트웨어를 선택하고 다운로드하여 기관내부의 개발자들에 의해 수정 개발하는 것을 의미한다. 아무것도 없는 상태에서 처음부터 개발하는 것이 아니라 공개되어 있는 수많은 공개소프트웨어들 가운데 도입하고자 하는 용도에 가장 적합한 공개소프트웨어를 선택하고 다운로드하여 기관내

부의 개발자들에 의해 수정 개발하는 것을 의미하는 것이다.

기관내부 개발 방법으로 공개소프트웨어를 도입할 때에 고려해야 할 사항들을 간단히 정리해 보면 다음과 같다.

- 가장 적합한 공개소프트웨어 선택기준 마련
- 다운로드한 공개소프트웨어의 안정성 검토
- 적용되는 라이센스 정책마련
- 가장 적합한 개발방법론 마련
- 개발 직원의 개발능력 검토
- 사용직원의 사용법 교육 안 마련
- 개발소프트웨어의 문서화 작업
- 개발된 공개소프트웨어의 안정성과 영속성 대책 마련
- 기타 특정업무의 종속되는 대책 마련 등

다음은 실제 내부 개발을 추진한 경우 필요한시에 단계별 개발절차이다.

1) 서비스의 정의 및 적용업무 분석

가장 먼저 해야 할 작업이 어떤 업무에 적용하기 위하여 공개소프트웨어를 도입할 것인가를 정의해야하는 작업이다. 즉, 이러한 작업을 "서비스 정의"작업이라고 할 수 있으며 웹기반으로 개발할 것인지 아니면 기관내부 솔루션으로 사용하기위한 개발인지 아니면 유관기관과의 정보공유 및 협동 네트워크를 위한 개발인지를 정확하게 구분하고 이를 정의해야 한다.

2) 기반 기술 정의

다음 단계는 개발기술에 대한 분석과 정의이며, 공개소프트웨어의

개발에 필요한 기반기술들을 간단히 정리하면 다음과 같다.

⊙ 운영체제, 웹서버, DBMS, 사용 언어 등에 대한 기반기술 분석 및 정의
⊙ 공용 framework 개발 및 업무 서비스 요구 분석 및 설계 기술
⊙ 분산 DB 구축 설계 및 관리 기술
⊙ 그룹웨어의 확장 및 공유 수준 지정에 의한 access 범위 설정 기술
⊙ 유관 기관 간 정보 DB 표준화 (XML 기반)

이와 같이 관련 기반기술들이 정의가 되어 있어야 하며 개발 직원에 대한 교육과도 연계되어야 할 항목이다.

3) 적용 업무에 대한 분석

개발하려는 공개소프트웨어가 적용될 업무에 대한 기본적인 분석이 이루어 져야한다. 즉, 공개소프트웨어를 직접 개발하기 이전에 다음과 같은 사항들이 이미 검토되어야 한다.

⊙ 활용성 : 개발 이후의 당 기관 또는 유관기관에서의 활용성
⊙ 수요의 규모 : 개발한 공개소프트웨어의 수요정도에 대한 검토
⊙ 재활용성 : 개발한 공개소프트웨어가 타 업무에 재활용 가능 여부
⊙ 표준성 및 표준의 준수여부

4) 개발기간 및 투입인력 분석

공개소프트웨어를 직접 개발하기 위해서는 개발기간과 투입인력에 대한 분석과 정의가 이루어져야 한다. 개발기간에 대한 정의는 다음을 고려하여 작성하도록 한다.

⊙ 5단계 개발기간 분석 : 요구분석기간, 분석기간, 개발기간, 수정 및

검토기간, 적용 및 안정화기간에 대한 분석 및 각 기간별 위험관리 대책 마련(개발기간의 준수여부에 따라 기간초과에 따르는 위험관리)

⊙ 개발인력에 대한 대책

- 기반 기술(주로 Language)에 가장 적합한 기술인력 활용
- 개발경험이 있는 내부 개발 직원 활용
- 개발책임자(PM)을 선정하여 프로젝트로 진행
- 개발 분야에 따라 외부 개발 직원을 투입대책 마련
- 개발자 교육 대책 마련

⊙ 개발한 공개소프트웨어의 사후 관리문제

내부개발에 의해 개발 완료된 공개소프트웨어는 일회성 프로젝트로 끝나지 않도록 사후관리가 이루어져야 한다. 사후관리에는 개발 직원에 대한 사후관리를 포함한다. 사후관리는 개발한 공개소프트웨어를 관리대장에 등록하여 지속적인 관리가 이루어지도록 한고, 개발 시에 사용한 기술을 종합 정리하여 이를 문서화하고, 개발한 공개소프트웨어의 설치문서, 사용법문서, 업그레이드 문서 등을 표준화하여 작성한다. 지금까지 설명한 내부개발 방법을 설명하였다. 기관 내부에 공개소프트웨어 개발자를 확보하기 어려울 뿐만 아니라 프로젝트 진행을 위한 관리자 확보가 어렵기 때문에 현실적으로 내부개발 방법으로 개발하는 경우는 거의 없다. 개발자나 프로젝트 실무자를 확보하고 있다 하더라도 실제 진행을 위해서는 보직변경 및 장기간 동일업무 종사여건등과 같은 조직내부적인 환경이 갖추어져야 할 것이다. 결론적으로 내부개발의 방법으로 프로젝트를 진행하기란 현실적인 어려움들이 존재하는 것이 사실이고 이러한 현실적인 어려움을 극복하고 프로젝트를 진행한다 하더라도 개발 및 프로젝트 노하우의 축적이 지속되기란 더욱 어려운 실정이다.

5) 종합비교

지금까지 설명한 직접선택, 간접공급, 내부개발 3가지 방법에 대한 장단점을 종합비교하면 다음 표와 같다. 공공기관에서 3가지 방법 중 하나를 선택하여 공개소프트웨어를 도입할 수는 있지만 국내 공개소프트웨어 업계의 현실과 기관의 현실을 고려할 때에 가장 현실성이고, 일반적으로 사용할 수 있는 방법은 간접공급 방식이다.

<표8>공개소프트웨어 도입 방식에 따른 비교

	직접선택	간접선택	내부개발
내용	▪ 소프트웨어를 직접 다운로드 하여 사용목적에 맞게 활용하는 것	▪ 소프트웨어를 외부업체로부터 공급받는 것을 의미	▪ 내부의 개발자들에 의해 수정 개발
장점	▪ 비용발생이 거의 없다	▪ 직접선택 방법보다는 위험부담을 줄일 수 있는 안전한 방법	▪ 개발 및 프로젝트 노하우의 축적
단점	▪ 소프트웨어의 보상과 보증이 보장되지 않으므로 위험부담이 크다	▪ 기술지원과 서비스 비용을 지불해야하기 때문에 비용적인 부담발생	▪ 개발자 확보 어려움 ▪ 프로젝트 실무자 확보 어려움 ▪ 보직변경 및 장기간 동일업무 종사여건 등 조직내부적인 기반환경필요
주의사항	▪ 지역 내에 기술지원이 가능한 개발업체가 하나 이상 존재하는 공개소프트웨어를 선택함으로써 도입기관의	▪ 개발업체에게 모든 소프트웨어 구성요소에 대한 완전한 감사를 수행하도록 하여 소프트웨어의 출처와 라이센스계약, 그리고	▪ 개발기간과 투입인력에 대한 세심한 분석과 정의가 필요 ▪ 기간별 위험관리대책 필요 ▪ 개발인력 준비방안

	안정성확보	위험평가 문서를 확인 ▪ 개발업체의 수행을 확인할 수 있도록 자체평가를 수행	필요 ▪ 개발한 소프트웨어의 사후관리대책 필요

1.3.2. 위험요소 분석

공개소프트웨어 도입 시에는 여러 가지 위험요소들이 존재한다. 이번 장에서는 이런 공개소프트웨어의 위험요소들에 대하여 몇 가지 분류로 나누어서 살펴보고자 한다.

가. 경쟁 유효성

공개소프트웨어를 도입함에 있어 존재하는 위험요소들과 공개소프트웨어를 도입하기를 망설이게 되는 여러 가지 도입 저해요소들은 상관관계가 존재한다. 먼저 본 가이드의 앞부분에서 다루었던 도입 저해요인들 가운데 경쟁 유효성에 관한 위험요소에 관련된 항목들을 간단히 정리해 보면 다음과 같다.

⊙ 공급업체의 신뢰성과 경쟁력 확보 여부
⊙ 도입하는 공개소프트웨어 자체의 경쟁력 확보 여부
⊙ 기능성 및 사용 편의성 등에 대한 경쟁력 확보여부
⊙ 비용절감의 경쟁력 확보 여부 등

첫째, 공급업체가 자체적인 경쟁력뿐 아니라 대외적인 경쟁력을 확보하

였는가의 여부에 따라서 위험요소가 존재한다. 이 부분은 공급업체의 신뢰성이 우선 바탕이 되어야하며 신뢰성을 기본으로 하여 두 가지의 경쟁 유효성 즉, 공급업체 스스로가 내부와 외부의 경쟁력을 확보하고 있는가에 대한 것과 공급되는 특정 공개소프트웨어를 기술 지원할 수 있는 업체들이 다수 존재하여 이들 업체들이 상호 경쟁력 확보를 하고 있으며 도입기관에서 기술지원 요청을 할 수 있는 다수 기업체가 존재하는가에 대한 것이다. 이중 후자는 공급받는 공개소프트웨어에 대한 영속성 확보를 위한 위험요소 감소가 그 목적이라고 할 수 있다. 둘째, 도입하는 공개소프트웨어 자체 경쟁력에 대한 부분이다. 공개소프트웨어뿐만 아니라 어떤 소프트웨어든 도입되기 위해서는 동일 분야의 타 제품보다 경쟁력이 확보되어야 한다. 만약 도입한 이 후에 보다 나은 소프트웨어를 찾게 된다면 도입자체를 재검토해야하 는 심각한 문제가 발생하게 된다. 따라서 이러한 위험요소를 줄이기 위하여 공개소프트웨어 자체에 대한 경쟁력확보 여부를 반드시 확인해야 한다. 셋째, 도입하는 공개소프트웨어의 사용자측면의 UI(User Interface)와 기능적인 경쟁력이 있는가에 대한 것이다. 즉, 이것은 도입되는 공개 소프트웨어에 사용상의 편리성과 기능상의 우수성에 대한 경쟁력을 확보하고 있는가에 대한 것으로써 훌륭한 소프트웨어라 하더라도 사용이 불편하다면 초기 도입을 위한 교육비용이 높아질 수 있다. 넷째, 공개소프트웨어 도입으로 얻으려고 하는 목적에 가장 높은 경쟁력을 확보한 공개소프트웨어의 선정이다. 공개소프트웨어의 장점은 비용절감 뿐만 아니라 기술 표준성 확보, 호환성 보장, 확장성, 영속성 등 여러 가지가 존재한다. 수행하려는 사업의 특성을 분석하여 가장 우선시 되어야 하는 경쟁력을 선정하고 이에 적합한 공개소프트웨어를 도입하여야 한다. 그리고 기존 시스템에 공개소프트웨어를 도입할 때에는 업체의 인적 기술력과 기술지원 경쟁력을 확보했는지 고려해야 한다. 즉, 기존정보시스템, 기존 소프트웨어와의 호환성 및 기능의 제약 등 기술적 제약 요건을 분석하고 기존 시스템에 공개소프트웨어를 도입함으로써 발생할 수 있는 문제를(예를 들어 기존 비공개소프트웨어와의

호환성문제, 추가 개발이 필요한 부문이 없는지 등) 기술지원을 제공하는 기업체에서 성실하고 적극적인 기술지원 및 교육을 제공하는지 여부를 확인하여야 한다.

나. 보증문제

공개소프트웨어와 독점 공급 되는 비공개 소프트웨어는 보증문제에 있어서도 큰 차이점이 있다. 독점 공급 되는 비공개 소프트웨어는 구매 시에 보증문제가 이미 언급되어 있고 계약서에도 대부분 명시되어 있으나 공개소프트웨어는 보증에 대해서 다소 분명하지 못한 점이 있다. GPL 제12항 및 제13항을 보면 GPL 라이센스로 제공되는 모든 공개소프트웨어의 문제 발생에 대한 책임은 원칙적으로는 수요자에게 있다는 것을 알 수 있다. 물론 공개소프트웨어를 도입하는 기업과 수요자 사이에 보증에 대한 별도의 계약관계가 있다면 보증에 대한 문제를 해결할 수 있을 것이다. 하지만 GPL에서는 공급자와 수요자 사이에 보증에 대한 별도의 계약을 하였다하더라도 근원적인 보증의 결여를 제한할 수는 없다고 명시하고 있다. 그렇다면 공개소프트웨어를 도입하려는 수요자의 입장에서는 다음과 같은 의문점을 가지게 된다. 많은 장점이 있다고는 하지만 공급자가 확실히 보증하지 않는 공개소프트웨어를 어떻게 믿고 도입해야 하는가?

공개소프트웨어의 보증에 대한 문제의 출발점은 여기서 부터 출발한다. 즉, 이번 절에서는 공개소프트웨어를 도입하면서 어떻게 하면 보증의 결여에 대한 대책을 세우고 안정되고 영속성 있는 공개소프트웨어를 도입할 것인가에 대하여 살펴볼 것이다. 이러한 공개소프트웨어의 보증의 결여에 대한 근본적인 원인은 공개소프트웨어가 공급자의 소유가 아닌 프로젝트의 산출물이기 때문이다.

물론 많은 이유가 존재하겠지만 공동의 프로젝트에 의하여 공동 개발된

소프트웨어이기 때문이며 이는 결코 보증의 결여에 대한 문제가 단점으로만 작용하고 있는 것은 아니다. 즉, 많은 개발자들에 의해 자발적인 공동개발 산물이라는 것은 문제 발생 시에 빠른 보완능력을 그 자체로서 가지고 있기 때문이다. 따라서 공개소프트웨어의 보증의 결여에 대한 문제해결의 실마리 또한 여기서 출발한다. 예를 들어 공동프로젝트의 산출물(A), 공개소프트웨어 공급업체(B), 공개소프트웨어 도입 기관(C)이 있는 경우를 가정했을 때, 공공기관C는 공개소프트웨어를 도입하기로 최종 결론을 내리고 공개소프트웨어 공급업체 B와 계약하여 A라는 공개소프트웨어를 GPL 라이센스를 적용하여 공급받기로 하였다. 여기서 B와 C의 계약서 작성 시에 보증문제에 대하여 공공기관C가 주의해야 할 것이 있다. 공개소프트웨어 A의 근원적인 문제에 대해서는 보증을 받을 수 없지만 공개소프트웨어 A를 공동 개발하는 프로젝트 내에서 근원적인 문제가 해결되어 패치버전이 나온다면 B는 즉시 이를 적용하여 도입한 공개소프트웨어를 개선해야한다. 즉, 이는 근원적인 문제를 원천적으로 해결해야한다는 조건이 아니라 근원적인 문제에 대한 해결이 이루어 졌다면 공급업체에서 도입한 공개소프트웨어에 이를 즉시 적용하여 개선해야 한다는 의미이다. 바로 이것이 공개소프트웨어를 도입할 때에 공공기관C가 가장 우선적으로 검토해야 할 보증문제의 기본적인 해결책이다. 앞서 살펴본 특정 공급업체로부터 공급받았을 경우에 보증의 문제가 발생하지만 업체로 부터 공급받지 아니하고 공개소프트웨어를 도입하기 위하여 무료다운로드를 제공하는 사이트에서 직접 다운로드를 하였을 경우에도 다운로드한 소프트웨어의 보상과 보증은 전혀 이루어 지지 않는다. 따라서 이에 대한 위험요소의 대책도 세워야 한다. 즉, 특정한 사이트에서 다운로드를 받은 공개소프트웨어에 대한 가장 기본적인 대책은 배포처 사이트의 주소뿐 아니라 다운로드한 공개소프트웨어에 대한 상세정보를 확보해야 한다. 즉, 공개소프트웨어의 프로젝트 진행자 또는 개발자, 또는 개발 배포회사에 대한 가장 기본적인 정보를 의미한다. 이렇게 확인해 두어야만 배포사이트의 주소가 바뀌거나 프로젝트 진행자가 바뀌거

나 또는 개발자, 배포회사가 사라지더라도 최소한의 문제해결 실마리는 얻을 수 있기 때문이다. 이것은 무료 다운로드한 공개소프트웨어의 위험요소를 줄이기 위한 최소한의 조치인 것이다. 그리고 보증의 문제에 대한 도입기관의 또 다른 대책으로는 지역 내에 가능한 많은 공개소프트웨어 지원업체를 알아두어야 한다는 것이다. 공개소프트웨어를 공급한 업체가 사라지거나 또는 기술지원을 중단하더라도 지속적인 보증과 지원을 받을 수 있는 가능성이 높아지기 때문이다. 이것은 도입기관에서 자체적으로 개발하여 도입하는 경우에도 해당된다. 즉, 기관내부에서 공개소프트웨어를 자체 커스터마이징하여 도입하는 경우에도 외부의 지원업체를 많이 알아 두는 것이 안정성과 영속성을 높이기 때문이다.

이와 같이 보증에 대한 여러 가지 안정장치들은 공개소프트웨어가 핵심적인 업무에 도입되는 경우에 가장 큰 효과를 얻을 수 있다. 지금까지 설명한 공개소프트웨어의 보증의 결여에 대한 위험요소를 감소시키려면 공개소프트웨어의 도입 시 기본적인 조건이 충족되어야 한다. 이는 보증의 결여에 대한 대책이 아니라 지금까지 설명한 보증 결여에 대한 대책 수립을 위한 전제조건이다. 먼저, 도입하는 공개소프트웨어의 라이센스가 명확해야 한다. 물론 거의 대부분 GPL로 도입이 되겠지만 공개소프트웨어에 적용되는 라이센스의 종류가 많기 때문이며 이들은 필요에 의해 수정되거나 새로 만들어진 것이므로 GPL과는 다른 부분들이 존재한다. 따라서 도입하는 공개소프트웨어에 적용되어 있는 라이센스와 그 내용을 꼼꼼히 확인해 보아야 한다. 하지만, 어떤 라이센스를 적용하든 도입하는 공개소프트웨어에 대한 완전한 권리를 획득하기 위해서는 소스코드의 확보와 수정권한 그리고 수정 후 이를 재사용하고 확장할 수 있는 권한을 모두 획득해야 한다. 이는 공개소프트웨어의 보증문제를 해결하기 위한 가장 기본적인 조건이다. 물론 당연한 것이기는 하지만 만약 이 조건이 만족되지 않는다면 공개소프트웨어의 도입자체부터 재검토해야하며 도입된다하더라도 보증문제는 전혀 보장되지 않는다고 보아야 한다. 따라서 소스코드에 대한 권리는 반드시 확인해

보아야 한다. 그리고 보증문제에 대하여 한 가지 더 확인해 두어야 할 것은 간접도입 방식으로 전문 기술지원 업체로부터 공개소프트웨어를 도입하는 경우, 도입하는 공개소프트웨어에 대한 기술문서, 사용자지침서, 온라인 튜토리얼, 교육 훈련 등에 대한 확보이다. 이절의 앞부분에서 말씀드렸듯이 공개소프트웨어의 보증에 대한 보장은 1차적으로 공급업체에서 이루어져야하겠지만 불가피하게 보증문제가 발생하였을 경우에는 기술문서등과 같은 문서들이 확보되어 있어야만 해결이 가능하기 때문이다.

다. 소스코드의 가용성

공개소프트웨어를 도입함에 있어 발생할 수 있는 위험요소들 가운데 경쟁유효성과 보증문제에 대한 위험요소들에 대하여 살펴보았다. 이번에는 공개소프트웨어의 소스코드에 대한 위험요소들이 어떤 것들이 있는가에 대하여 살펴볼 것이다. 즉, 이번 절에서는 소스코드의 가장 핵심적인 요구조건인 가용성에 대하여 살펴볼 것이다. 첫째 소스코드 자체의 확보여부, 소스코드의 수정 및 수정후의 재사용, 확장성보장등과 함께 소스코드 자체로서 경쟁력을 가지고 있는지 그리고 향후 확장성을 보장할 수 있도록 코딩되어 있는지, 소스코드의 수정 시에 어떤 개발자라도 수정작업이 용이하도록 보장되는지, 그리고 여러 루틴들이 상호 유기적으로 표준화되어 코딩되었는가를 검토해 보아야하며 이러한 부분들이 보장이 되어야만 소스코드 자체의 경쟁력과 가용성을 보장받을 수 있다.

⊙ 소스코드의 확보
⊙ 소스코드의 수정 가능성 확보
⊙ 수정한 소스코드의 재사용, 확장성, 재배포 가능성 확보
⊙ 확장 / 수정 용이성
⊙ 소스코드의 표준화

공개소프트웨어로 도입하는 소프트웨어의 소스코드는 많은 측면에서 보장되어야 한다. 만약 독점 공급되는 비공개 소프트웨어라면 소스코드 자체를 확보하기가 거의 어렵기 때문에 이러한 점들을 고려할 필요가 없겠지만 공개소프트웨어는 소프트웨어를 공급받을 때에 소스코드를 함께 공급받기 때문에 이에 대한 가용성 문제를 검토함으로써 공개소프트웨어의 장점을 극대화할 수 있다. 특히 직접선택 방식으로 공개소프트웨어를 사용하는 경우, 소스코드의 재사용에 관련된 가용성 확보는 필수적인 것이다. 다행스럽게도 GPL을 포함한 대부분의 공개소프트웨어 라이센스에는 소스코드를 공개해야 한다는 조건이 있으므로 간단한 확인만으로도 소스코드 확보는 어렵지 않다. 둘째, 소스코드에 대한 재사용, 확장성, 재배포 권한여부를 확인해야 한다. 모든 공개소프트웨어는 소스코드의 제공 및 수정에 대한 권리를 제공하지만 수정한 소스코드의 배포에 대한 권한은 공개 소프트웨어 라이센스의 종류에 따라 틀리다. 예를 들어 수정한 공개소프트웨어를 사업용으로 재배포 가능 또는 불가능한 경우, 재배포할 때에 수정한 소스코드를 함께 공개해야 하는 의무 등을 부가하는 라이센스도 있다. 그러므로 보유한 공개소프트웨어에 대한 라이센스를 확인하여 재배포에 부여되는 조건을 확인해야 한다. 앞에서 언급한 두 가지 조건이 소스코드 가용성의 전제조건이다. 즉, 소스코드의 확보, 수정권한, 재사용권한이 소스코드의 가용성에 대한 기본조건이 된다는 의미이다.

결론적으로 소스코드의 가용성에 대한 위험요소를 줄이기 위해서는 도입 시 이 두 가지가 전제되어야 한다는 것을 알 수 있다. 직접선택 방식으로 공개소프트웨어를 도입하는 경우 소스코드의 수정 및 확장성을 높이기 위하여 소스코드의 표준화를 추가하여 검토할 수 있다. 이 또한 소스코드의 가용성을 높이는 것으로써 표준화되어 있는 소스코드는 가독율이 좋으며 개발자가 달라지더라도 수정이 용이해지며 소스코드를 확장하기에 매우 유

용하기 때문이다. 따라서 소스코드의 표준화는 가용성을 높이는데 매우 큰 역할을 한다.

라. 위험관리 및 완화방법

1) 도입유형별 위험평가

이미 설명하였듯이 공개소프트웨어를 도입하는 방법에는 크게 3가지가 있다. 도입하려고 하는 공개소프트웨어를 기관에서 직접 선택하는 직접선택 방법이 있고, 외부의 공개소프트웨어 개발업체로부터 도입하는 간접공급 방법이 있으며 나머지 하나는 기관내부에서 공개소프트웨어를 직접 개발하여 도입하는 내부개발 방법이 그것이다. 이번 절에서는 이 3가지 도입유형에 따른 도입 시의 위험요소에 대하여 알아보도록 할 것이다.

⊙ 직접선택 방법에 따른 위험 평가

공개소프트웨어를 기관에서 직접 다운로드하여 도입하고자 하는 경우 비교적 많은 위험요소가 내재되어 있다. 가장 먼저 주의해야 할 사항은 다운로드한 공개소프트웨어가 정말깨끗하고 믿을 수 있는 좋은 소프트웨어인가를 확인하기 전까지는 매우 불안한 요소를 내포하고 있을 가능성이 있기 때문이다. 이 말의 의미는 다운로드한 공개소프트웨어 내에 트로이목마와 같은 바이러스 코드나 또는 키로거(key-logger)와 같은 악성 소프트웨어에 의한 감염된 소스등과 같은 악의적인 목적을 가진 소스코드가 내포되어 있을 가능성이 존재하기 때문이다. 그리고 앞 절에서도 살펴보았던 문제이지만 직접 다운로드한 소프트웨어에 대해서는 보상과 보증이 명확하지 않고 보장되지 않는다는 점이다. 즉, 공개소프트웨어 솔루션을 기관내부에서 직접 선택하여 도입하려고하는 경우에는 소프트웨어의 보증과 보상을 보장받지 못하는 위험요소가 존재한다. 또 다른 위험요소는 공개소프트웨어의 배포정보에 대한

부분이다. 즉, 공개소프트웨어를 지속적으로 업그레이드해서 배포해야한다는 의무적인 사항은 그 어디에도 존재하지 않는다. 따라서 오늘 다운로드한 공개소프트웨어가 내일은 배포되지 않을 수도 있고 또한 어떠한 공지사항 없이 배포위치가 바뀔 수 있기 때문이다. 따라서 공개소프트웨어를 직접 다운로드하여 도입하는 기관에서는 배포처나 개발프로젝트에 대한 위험요소 감소대책을 확보해야한다. 그리고 어떤 공개소프트웨어에도 존재할 수 있는 공통적인 위험요소로는 기술지원에 대한 위험요소이다. 특히 공개소프트웨어를 직접 다운로드하여 도입한 기관에서는 자체 기술지원시스템을 확보하고 있거나 유사시에 도움을 요청할 수 있는 공개소프트웨어 커뮤니티 또는 전문 기술지원업체에 대한 다양한 정보를 확보하고 있어야 한다. 즉 직접 다운로드하여 도입하는 공개소프트웨어에 대해서는 더욱 적극적이고 명확한 기술지원에 대한 위험요소 감소대책을 확보해야 한다.

⊙ 간접공급 방법에 따른 위험 평가

간접공급에 의한 공개소프트웨어 도입이란 수요자가 공개소프트웨어를 외부업체를 통해 공급받는 것을 의미하는 것으로서 공공기관에서 직접 선택하는 앞의 방법보다는 위험요소를 감소시킬 수 있는 방법이다. 즉, 외부업체를 선택하여 선택한 업체에서 공개소프트웨어를 공급하는 것으로 기술지원과 서비스 비용을 지불해야하기 때문에 비용적인 부담이 있기는 하지만 위험요소를 줄일 수 있다는 장점이 있다. 하지만 이 경우에도 공개소프트웨어의 도입에 따른 위험요소들이 존재한다. 따라서 업체에서 공급하는 공개소프트웨어에 적용되어 있는 라이센스가 어떤 것인가를 명확하게 확인해야 한다. 계속해서 언급하였듯이 공개소프트웨어에 적용되어 있는 라이센스는 여러가지 종류가 있다. 이들은 필요에 의하여 그 환경과 요구사항에 맞도록 만들어진 것들이며 많은 부분에서 그 내용을 달리하는 경우가 있다. 따라서 간접방식으로 공급되는 공개소프트웨어 공급 계약서 내에 라이센스에 대한 명확한 언급이 있어야만 한다. 물론 도입기관의 담당자는 그 라이센스의 내용에

대하여 명확하게 숙지해야 한다. 그리고 외부 업체에서 공급하는 간접공급방식에서의 또 다른 위험요소로는 하자보증에 대한 부분이다. 1차적으로 공급업체는 하자보증에 대한 지원을 해야 하며 또한 책임을 져야하지만 공개소프트웨어의 GPL 라이센스가 적용된다면 근원적인 문제점에 대해서는 공급업체에서 보증에 대한 책임을 회피할 수 있다는 위험요소가 발생한다. 따라서 공개소프트웨어를 도입하는 기관 담당자는 보증에 대한 위험요소에 대한 대비책을 세워두어야 한다. 그리고 간접방식으로 공급되는 공개소프트웨어에 대한 위험요소에는 일반적인 공급계약방식에서 발생할 수 있는 거의 모든 위험요소가 고려되어야 한다. 즉, 공급업체의 적격성에 대한 위험성 평가, 시스템의 확장성 가능여부에 대한 위험요소 평가, 소스코드의 성숙도 및 신뢰도에 대한 위험요소 평가 등이 모두 고려되어야 한다. 마지막으로 간접방식에서는 외부 공급업체의 기술지원 능력에 대한 위험요소가 존재한다. 즉, 공급업체의 기술지원 능력에 대한 객관적인 검증 없이 공개소프트웨어를 도입하였다면 도입 이후의 기술지원 문제가 발생할 수 있다는 점을 명심해야 한다. 아울러 비공개소프트웨어의 경우와 마찬가지로 공급업체의 파산 등으로 기술지원을 받지 못하는 사태가 발생할 수 있으므로 동일 솔루션의 기술지원업체에 대한 정보를 확보해 두는 것도 공개소프트웨어의 장점을 극대화하여 위험요소를 줄일 수 있는 좋은 방법이다.

⊙ 내부개발 방법에 따른 위험평가

마지막으로 내부개발 방법에 의한 공개소프트웨어 도입에 따른 위험요소는 다음과 같다.

- 다운로드한 공개소프트웨어의 안정성 검토
- 개발된 공개소프트웨어의 라이센스 문제
- 개발된 공개소프트웨어의 안정성과 영속성 문제
- 내부 개발자의 개발능력 문제
- 개발 후 문서화 작업 문제 등

앞 절에서 이미 설명하였듯이 기관 내부에서 공개소프트웨어를 개발하기 위해서는 이미 배포되어 있는 공개소프트웨어들 가운데 가장 적합한 공개소프트웨어를 우선 선택하는 작업을 해야 한다. 이때 발생할 수 있는 위험요소는 선택하는 공개소프트웨어의 안정성과 적합성을 검토해야 한다. 즉, 다운로드하는 공개소프트웨어 소스내부에 악의적인 코드가 들어있지 않는가를 확인해야하며 또한 개발하고자 하는 소프트웨어의 목적에 가장 적합한 소프트웨어인가를 검토하는 작업을 해야 한다. 특히 공공기관의 업무에 적용하기 위한 경우 소프트웨어의 기술적 안정성 및 신뢰성 검토에 신중을 기하여야 한다. 다음은 다운로드하는 공개소프트웨어에 적용되어 있는 라인센스, 그리고 개발완료 후에 적용할 라이센스 문제를 검토해야 한다. 개발 전에 공개된 공개소프트웨어를 다운로드할 때에 소스확보가 가능한지, 소스수정이 가능한지, 그리고 소스 수정 후에 업무에 적용가능한지, 유관기관에 이를 재배포할 수 있는 재배포 권한이 있는가에 대한 면밀한 검토가 도입 전에 이루어져야만 한다. 만약 이에 대한 정확한 검토 작업 없이 추진하였다면 개발 완료되었을 때에 업무적용 및 활용, 배포에 심각한 문제가 발생할 수 있다는 위험요소를 가지고 있으므로 주의해야 한다. 다음은 개발된 공개소프트웨어의 안정성과 영속성에 대한 문제이다. 바로 앞에서 언급한 라이센스 문제와 일부분 결부되는 문제이지만 소프트웨어의 안정성문제는 소스코드의 성숙도측면에서 고려되어야 한다. 그리고 소프트웨어의 영속성 문제는 시스템증설이나 향후 업그레이드 시에 재활용 및 재사용이 가능한가라는 확장측면에서 고려되어야 할 문제이다. 따라서 소프트웨어의 안정성부분에 따르는 위험요소와 향후 확장 및 증설측면에서의 영속성에 따르는 위험요소도 고려되어야 한다. 다음은 공개소프트웨어의 소스코드를 도입사업에 적합하도록 수정한 후, 해당 소프트웨어의 버전관리에 대한 방안이다. 공개

소프트웨어는 사업에 적합하도록 수정한 이후에는 해당 공개소프트웨어의 업체에서 새로 출시된 버전과 호환성에 문제가 발생할 수 있으며 특히 수정한 부분에서 기술적인 문제가 발생하였을 때에 외부의 공개소프트웨어 커뮤니티에서 해결책을 구하기가 어려울 수도 있다. 그리고 개발 완료된 이후에 개발된 공개소프트웨어에 대한 문서화작업을 해야 한다. 문서화 작업은 크게 두 가지로 나누어서 이루어져야 하는데 첫 번째는 소스코드에 대한 문서화 작업, 두 번째는 소프트웨어 사용에 대한 문서화 작업이 그것이다. 첫 번째 소스코드에 대한 문서화 작업은 소스코드의 재사용, 업그레이드 및 패치, 향후증설작업등을 위하여 소스코드에 대한 자세한 설명과 함께 수정 방법 등에 대한 문서를 의미한다. 이 문서화 작업은 처음 개발했던 개발자를 위해서이기도 하지만 다른 개발자가 보았을 때에 도움이 되도록 작성되어야 한다. 두번째 사용법에 대한 문서화 작업으로써 이는 개발완료 이후에 업무에 적용하여 사용자들이 이를 원활하게 사용할 수 있도록 사용법문서등을 의미하는 것이다. 즉, 설치문서, 사용법문서등을 의미한다. 이와 같은 문서화 작업이 제대로 이루어져 한다. 왜냐하면 개발된 소프트웨어의 확장성과 활용성에 따르는 위험요소가 발생할 수도 있기 때문이다.

⊙ 기술로드맵 평가

기관에서 소프트웨어를 도입할 때에는 비공개 소프트웨어 뿐 아니라 공개소프트웨어를 도입할 때에도 기술적인 로드맵을 확보해야 한다. 기술 로드맵이란 도입하는 소프트웨어에 향후 추가할 기능 개선점과 전달 시간계획에 대한 개요를 제공해 주는 문서이다. 이 문서의 목적은 소프트웨어를 도입하는 고객이 해당 소프트웨어의 발전방향을 미리 파악하여 향후 시스템 구축 계획에 반영할 수 있도록 하기 위한 것이다. 물론 기술 로드맵의 일정과 기능이

반드시 지켜진다고 보장이 되는 것은 아니나 이것은 독점 공급되는 비공개 소프트웨어에서도 마찬가지이다. 그러나 이러한 객관적인 기술 로드맵을 통하여 기관은 해당 소프트웨어의 발전 방향을 예측할 수 있으며 특정 기술의 단종에 따른 위험요소를 평가하고 대책을 수립하는데 큰 도움이 된다. 또한 이에 못지않게 중요한 것은 미래 비전에 대한 공급업체의 수행 능력이다. 만약 공급업체가 제품 로드맵, 또는 일반 기술 플랫폼을 수정하거나 변경한다면 이는 위험요소가 매우 높은 업체로 인식해야 한다. 따라서 도입기관은 개발업체에서 공급하는 제품이 원래 의도했던 대로 미래지향적이지 못하거나 훌륭하지 못하다고 판단하게 될 것 이다. 또한 공급받은 제품이 몇 년 후에 발생할지 모르는 기술지원에 대한 보증도 할 수 없을 것이다. 또한 기관은 미래의 위험요소를 보다 정확하고 적절하게 평가할 수 있도록 제품 로드맵을 검토해야 한다. 이러한 로드맵에 대한 정기적인 검토는 기관이 공급업체의 신뢰성과 능력을 파악하는데 큰 도움이 된다. 그리고 이러한 작업들은 미래의 계획 수립 시에 위험요소를 더욱 줄이는 훌륭한 방법이 다. 로드맵 평가 및 검토 과정은 다음사항을 고려한 절차대로 이루어지는 것이 좋다.

- 공급받은 제품의 개발업체가 제공한 로드맵을 배치하고 기록한다.
- 개발업체(또는 커뮤니티)가 발표한 최신 개발정보 및 제품 배포일 및 버전에 대한 정보를 비롯하여 사용자 커뮤니티에 대한 정보를 배치하고 기록한다.
- 현재의 타임라인(timeline)을 점검한다.
- 가능한 경우 개발업체 또는 대표 개발자에게 직접 연락하여 (개발업체의 대표자에게 연락한다.) 로드맵 및 배포 일정에 대한 업데이트의 주기 등에 대한 정보를 확인한다.
- 제품 로드맵과 배포 일정을 모두 포함하는 웹 페이지의 개별 사본을 배치하고 기록한다.
- 변경내용을 확인할 수 있는 이전의 로드맵과 최신 로드맵을 비교한다. 특히 추가 예정되었던 사항이 제거된 항목이 있다면 이를 확인한다.

- 추가된 항목을 확인할 수 있는 이전버전과 소프트웨어 개발자의 최신 뉴스 항목을 비교 한다.
- 최초로 제안되었던 이정표의 내용들이 실제 실행되었는지 확인할 수 있도록 현재 뉴스 항목 웹 페이지에서 확인된 타임라인과 날짜를 비교 한다.
- 분석 기간이 요구된다면 이 주기를 비교한다.

이러한 정보들은 공급되는 공개소프트웨어 제품 뿐 아니라 공급업체(개발업체)에 대한 위험요소 평가의 일부분으로 구성되어야 한다. 이러한 과정은 소프트웨어를 생산하고 유지 관리하는 주요 부분에서 발생할 수 있는 문제점들을 도입기관에게 알려주는 매우 유용한 역할을 하게 된다. 따라서 문제 발생에 대한 조기 경고를 함으로써 도입되는 공개소프트웨어의 위험요소를 줄일 수 있고 또한 효율적인 관리를 할 수 있다. 기술로드맵의 변경에 따른 위험요소 관리 방법은 다음과 같다. 공개소프트웨어를 개발하는 어떤 개발업체들은 품질 및 세부사항에 대한 기술 로드맵을 충분히 제공하지 못할 수도 있다. 더욱이 이미 마련되어 있는 자체 로드맵 또한 일관성 있게 지키지 못하는 경우도 있다. 이러한 개발업체들로 인하여 도입기관은 장기적인 미래전략 수립을 하는 것이 매우 힘들게 된다. 결론적으로 이러한 로드맵에도 위험요소가 존재하고 있다. 이러한 로드맵에 대한 위험요소를 줄이는 방법은 아주 간단한 원칙에서 찾을 수 있다. 기관이 타 기관 또는 사용자들이 광범위하게 사용하고 있는 공개소프트웨어 제품을 미리 선택하고 공급업체에게 이미 선택한 제품을 공급하도록 한다면 위험요소의 상당부분을 줄일 수 있다. 일반적으로 많은 사람들로 부터 광범위하게 사용되어지고 있는 소프트웨어에 대한 위험요소는 매우 낮을 뿐 아니라 기관 및 사용자들이 요구하는 거의 모든 기능과 특징들을 이미 갖추고 있기 때문이다. 이상과 같이 공개소프트웨어의 도입에 따른 로드맵 확인과 로드맵의 평가, 그리고 로드맵의 관리방법과 위험요소들에 대하여

알아보았다. 결론적으로 공개소프트웨어를 선택하고 도입할 때에는 기술 로드맵이 있는가를 확인하고 이를 평가하는 과정을 거쳐야 한다. 이러한 작업들로써 우리는 미래에 있을 위험요소 즉, 미래 위험요소를 줄일 수 있다.

⊙ 위험완화를 위한 검토사항

지금까지 공개소프트웨어의 도입에 따르는 위험요소들에 대한 분석과 관리방법에 대하여 알아보았다. 이제 이러한 위험요소를 줄이기 위한 방법들을 살펴보겠다. 공개소프트웨어를 도입하기로 결정하기 전에, 도입하고자하는 제품의 기술, 이전 기록, 사용 패턴, 인기도, 기존 사용자의 피드백을 신중하게 검토할 필요가 있다. 다음은 공개소프트웨어를 도입할 때에 검토할 항목들이다.

- 도입하는 공개소프트웨어가 배포사이트에서의 알림사항과 제공문서에 기술되어 있는바와 같이 운영되는가를 확인해야 한다.
- 도입기관의 필수 요구항목과 요구 성능이 일치하는지 또는 그 이상의 성능을 보유하고 있는가를 확인한다.
- 도입하는 공개소프트웨어 프로젝트가 지속적으로 진행되고 있는가를 확인한다. 즉, 도입하는 공개소프트웨어의 프로젝트 사이트에 대하여 다음 사항을 검토해야 한다.
  - 배포되어 있는 소프트웨어의 보안패치 및 업데이트
  - 해당 프로젝트에 대한 공지사항 및 발표 사항 등의 유무
  - 프로젝트 내에서의 포럼, 메일링, 문서 등의 활용유무
  - 기술방향을 세부적으로 기술해 둔 프로젝트 로드맵
  - 프로젝트 참여 개발자와 사용자의 지속적인 확대 여부

- 제품의 기술지원을 위한 지역 기술지원업체의 가용성을 확인한다.
- 최신 발표에 열거된 공개된 웹사이트에서 해당 소프트웨어가 지원되는

가?

- 해당 소프트웨어의 웹사이트가 이전 버전에 대한 기록을 보유하고 있는가? 그리고 보유기록에 의해 지금까지의 배포횟수와 빈도를 점검할 수 있는가?
- 배포 사이트가 보안 수정 및 특징 업데이트에 대한 정보를 제공하는가?
- 해당 공개소프트웨어를 위해 공개적으로 사용할 수 있고 독립적인 사용자 포럼이나 메일링 리스트를 보유하고 있는가?
- 포럼이 사이트에서 또는 공공의 검색 엔진을 통해 검색될 수 있는 문서보관소를 보유하고 있는가?
- 소프트웨어가 프로젝트 웹사이트에서 이용 가능한 로드맵을 발행하여 보유하고 있는가? 그리고 이 로드맵이 기관의 요구조건을 충족시키는가?
- 프로젝트가 기관의 요구조건을 충족시키기 위해 추가 기능을 필요로 하는 경우, 개발팀에 새로운 기능이나 개선품을 요청할 수 있는 프로세스를 보유하고 있는가?
- 프로젝트 웹사이트에 지역 기술지원 제공업체가 열거되어 있는가? 그리고 이들 지역 기술지원업체들에 대한 위험요소 평가를 수행하였는가?
- 해당 공개소프트웨어를 적절히 운용하기 위해 필요한 보조 구성 요소, 모듈, 라이브러리 또는 시스템 요구조건을 분석하였는가?
- 이러한 구성요소들 각각에 대한 위험요소를 줄이기 위한 체크리스트를 작성하였는가?

또한 공개소프트웨어를 도입하는 기관은 대상 소프트웨어에 대한 사용자의 평가를 직접 확인할 수 있다. 즉, 해당 프로젝트 메일링리스트 또는 포럼은 해당 공개소프트웨어가 보유하고 있는 기술의 적합성, 안전성, 품질의 우수성을 보다 객관적으로 검증할 수 있는 유용한 방법이다. 따라서 사용자들이 참가하고 있는 포럼과 메일링을 통하여 사용자들의 만족도에 대하여 평가

할 수 있고, 큰 문제없이 오랜 시간 동안 해당 공개소프트웨어를 사용해오고 있는가를 평가할 수 있다.

마지막으로 도입하고자하는 공개소프트웨어의 위험요소를 완화하기 위하여 도입제품의 장기적인 생존가능성을 평가해야 한다. 공개소프트웨어를 개발하는 공개소프트웨어 프로젝트는 사용자 커뮤니티가 활성화되어 있을수록 안정적인 기능과 지속적인 업그레이드를 보장할 수 있다. 반대로 해당 공개소프트웨어 프로젝트가 소수의 사용자에 의해 소극적으로 운용되고 있다면 외부 커뮤니티를 통한 기술지원이나 해당 공개소프트웨어의 장기적인 업그레이드는 원활하게 이루어지지 않을 가능성이 많다. 따라서 공개소프트웨어의 위험요소를 평가할 때에는 이러한 사항도 고려하여야 한다.

참고문헌

[송위진02]송위진, "한국형 오픈소스 소프트웨어 기술개발 전략", 과학기술정책연구원, 2002. 8.

[이철남02]이철남, "오픈소스 소프트웨어를 위한 법률적 기초", 정보통신정책 306호, 정보통신정책연구원, 2002. 8.

[CIPR02]CIPR, "Integrating Intellectual Property Rights", Commission on Intellectual Property Rights, 2002. 9.

[Eric01]Eric S. Raymond, "The Cathedral & Bazaar", O'Reilly, 2001

[Law99]Lawrence Lessig, Code and Other Laws of Cyberspace, 1999

[6]Patrice-Emmanuel SCHMITZSebastien CASTIAUX, "Polling Open Source Software, An IDA Feasibility Stud", European Commission, DG Enterprise, June 2002.

[7]Rishab Ghosh, Bernhard Krieger, Ruediger Glott, Gregorio Bobles, "Free/Libre and Open Source Software: Survey and Study", FLOSS Final Report, 2002. 6., at http://www.infornomics.nl/FLOSS/report

[8]"Guidance on implementing Open Source Software", Office of Government Commerce, 2002. 9.

[9]"Open Source software and The NHS: WHITE PAPER", NHS Information Authority, 2002. 1. at http://www.nhsia.nhs.uk/def/pages/features/i_250202.asp

[이도규06]이도규, “공개 소프트웨어 정책성과 발전 방향,” 한국정보과학회 정보과학회지, 2006년 6월, pp.5-8.

[손덕주06]손덕주, “리눅스 데스크톱 규격 및 서버 규격의 단체 표준 제정,” 한국정보과학회 정보과학회지, 2006년 6월, pp.30-33.

[안백송06]안백송, 고광원, “공개 소프트웨어 핵심기술 기반 리눅스 시스템, 부요,” 한국정보과학회 정보과학회지, 2006년 6월, pp.34-41.

[IBM04]IBM white paper,“The role of open computing, open standards

and open source in the public sector", 미발표 자료, 2004.5

[Jon03]“Open Source vs. Open Standards”, Jonathan Schwartz, CNet news.com, April 10, 2003
<http://news.com.com/2010-1071-995823.html>

[JILT02]O'Sullivan M, 'Making Copyleft Ambidextrous: An Expose of Copyleft', The Journal of Information, Law and Technology(JILT) 2002.3

[PFF96]Keyworth, G(1996), People and Society in Cyberspace, <http://www.pff.org:80/tsos-1.html>

[17] Boyle, J.(1996), Shamans, Software and Spleens: Law and the construction of the Information Society, Harvard Unv. Press

[홍성태00]‘지적재산권과 ’현실정보사회‘의 모순’, 홍성태, 2000.2.18 http://www.ipleft.or.kr/digital/ipleft01.htm

[진흥원03a]김영문 외, 공개소프트웨어 도입가이드라인 연구, 한국소프트웨어진흥원, 2003

[진흥원02]김영문 외, 공개소프트웨어 라이센스 연구, 한국소프트웨어진흥원, 2002

[진흥원03b]김영문 외, 공개소프트웨어활성화를 위한 법제도 개선방안 연구, 한국소프트웨어진흥원, 2003

[전산원00]최성모 외, 공공기관을 위한 리눅스 도입 방안 연구, 한국전산원, 2000

[진흥원0405]한국소프트웨어진흥원, 공개소프트웨어 도입 성공사례집, 2004, 2005