

## 기록시스템의 오픈소스화 전략 연구

A Study on Open Source Transition Strategy of Record System

안대진(An, Dae-jin)\* · 임진희(Yim, Jin-hee)\*\*

1. 머리말
  - 1) 연구 목적
  - 2) 연구 배경
  - 3) 연구 내용과 방법
2. 오픈소스화의 절차
  - 1) 소스 '공개'의 의미와 전략적 이점
  - 2) 소스 공개를 위한 절차와 도구
  - 3) 소프트웨어 개선을 위한 협업과정
  - 4) 시사점
3. 기록시스템 오픈소스화 사례
  - 1) ICA-AtoM : 글로벌 거버넌스 주도 개발
  - 2) Omeka : 대학 주도 개발
  - 3) Archivematica : 기업 주도 개발
  - 4) DPLA : 비영리단체 주도 개발
  - 5) veraPDF : 국제협력체계 주도 개발
  - 6) DPSP : 내셔널 아카이브 주도 개발
4. 기록시스템 오픈소스화의 핵심 성공요소
  - 1) 마이크로서비스 아키텍처와 MVC 패턴
  - 2) 공공기록관리 법제의 정비
  - 3) 저작권과 라이선스 전략
  - 4) 협력과 개선을 위한 공동체의 조건
5. 맺음말

\* 명지대학교 기록정보과학전문대학원 박사과정, (주)아카이브랩 대표(주저자).

\*\* 정보인권연구소 연구위원(yimjnhkr@empas.com)(교신저자).

■ 투고일 : 2017년 4월 5일 ■ 최종심사일 : 2017년 4월 5일 ■ 게재확정일 : 2017년 4월 18일.

## 〈초록〉

이 연구는 국가기록원과 서울특별시 기록시스템을 오픈소스화하기 위한 환경을 분석하여 위험요소를 식별하고, 국내에서 이 전략이 성공하기 위해 어떤 요건이 갖춰져야 하는지를 살펴보는 것을 목표로 한다. 2장에서는 소스코드 공개가 수요기관이나 개발업체, 그리고 이용자 모두에게 비용 및 효율성 측면의 전략적 이점을 제공한다는 것을 밝혔다. 그리고 협력적으로 개발하여 소스코드를 공개하는 절차와 이를 지원하는 기술 인프라에 대해 알아보았다. 3장에서는 기록관리 분야의 대표적인 오픈소스 프로젝트 6종을 선정하여 사례연구를 수행했다. 종합적인 시사점을 얻기 위해 오픈소스 프로젝트의 개발 주체를 국제기구, 국제협력체계, 국립아카이브, 개발업체 등으로 구분하였다. 각 프로젝트별로 추진배경과 목적, 개발과 펀딩의 주체, 거버넌스 모델, 개발기간과 비용, 비즈니스 모델과 소프트웨어 아키텍처, 커뮤니티 구성과 라이선스 전략 등을 분석하였다. 사례를 통해 네 가지 핵심 성공요소를 도출했다. 기술적 측면에서는 컴포넌트 기반의 설계가 필요했다. 이를 위해 마이크로서비스 아키텍처와 모델-뷰-컨트롤러 패턴을 제안했다. 다음으로는 기록관과 영구기록물관리기관의 시스템 필수요건 재정립이 필요했다. 그리고 개발업체가 오픈소스 프로젝트에 쉽게 참여하도록 하기 위해 듀얼 라이선스 전략을 제안했다. 마지막으로 지속 가능한 커뮤니티를 만들기 위해 강력한 거버넌스 조직과 효율적인 협력체계를 강조했다.

수요기관 중심의 시장 환경에서 기록시스템의 오픈소스화 전략이 성공하기 위해서는 공동체 안에서 개발업체와 이용자의 역할이 더 발휘되어야 한다. 그리고 이들이 효과적으로 의사소통하기 위한 협력도구와 개발 인프라를 기획 단계부터 중앙집중식으로 구축하는 것이 필수적이다.

**주제어 : 기록시스템, 오픈소스, 오픈소스화**

## 〈Abstract〉

This study aims to analyze the environment for the open-source records system and to identify the risk and requirements for the success

of the strategy in Korea. For this, Chapter 2 presented a review of the strategic benefits of open source to public organizations, developers, and users. It also discussed the process of cooperatively developing and releasing the source code and the technology infrastructure supporting open source. In Chapter 3, six representative open-source projects in the field of records management were selected, and case studies were conducted. To derive comprehensive implications, we have divided the main development body of open-source projects into international organizations, international cooperation systems, national archives, and software development companies. We also analyzed the background and purpose of each project, the agents of development and funding, the governance model, the development period and cost, the business model and software architecture, the community composition, and the licensing strategy. Through this, we have derived four critical success factors. In terms of technology, a component-based design was required; therefore, we proposed a microservice architecture and a model-view-controller design pattern. Next, it was necessary to reestablish system requirements of records center and archives. Moreover, we also proposed a dual licensing strategy to allow developers to easily participate in open-source projects. Lastly, we emphasized a strong governance structure and an effective cooperation framework to create a sustainable community.

For a record system to be open-source successfully in an organization-centered market, the roles of software developers and end users should be exercised more in the community. To achieve this, it is important to build various collaborative tools and development infrastructure from a planning stage to a centralized one.

**Keywords : Records System, Open Source, Open Source Software**

## 1. 머리말

### 1) 연구 목적

최근 서울기록원 ISP에서 서울특별시가 아카이브시스템을 개발하면서 소스코드를 공개하는 전략이 논의되었다(서울특별시 2016a, 646~657). 서울특별시는 본격적인 시스템 구축사업을 앞두고 이를 현실화하기 위한 방안을 구체적으로 검토하고 있다. 이는 기록시스템을 오픈소스화하는 국내 첫 사례가 될 것이다. 이를 계기로 오픈소스 전략이 기록관리의 성장을 견인하기를 바라는 분위기 또한 형성되어 있다.

한편으로는 공개소프트웨어, 즉 오픈소스 소프트웨어에 대한 정의나 라이선스에 대한 오해를 바로잡아야 한다는 문제가 제기되었다. 또한 벤더들의 수익성을 우려하는 의견도 나왔다. 따라서, 이 연구에서는 국가기록원과 서울기록원 등이 기록시스템의 오픈소스화 전략을 실행하기 위한 환경을 분석하여 위험요소를 식별하고, 국내에서 이 전략이 성공하기 위해 어떤 요건이 갖춰져야 하는지를 살펴보는 것을 목표로 한다.

### 2) 연구 배경

한국의 공공기관은 법령과 제도, 표준화된 시스템을 통해 기록관리를 하고 있다. 한국정부는 지속적인 전자정부 추진을 통해 국제적 ICT 평가지표인 UN의 전자정부발전지수(e-Government Development Index)와 온라인참여지수(e-Participation Index), 국제전기통신연합(ITU)의 ICT발전지수(IDI: ICT Development Index) 등에서 지속적으로 세계 1위를 유지하는 전자정부 선진국이 되었다(한국정보화진흥원 2014, 13). 따라서 공공기록관리 분야도 고도의 디지털 환경에 조응하는 시스템을 개발하는 것이 핵심성공요소가 된다.

그러나, 박종연(2013), 이경남(2013), 현문수(2013), 박민영(2013), 정상희(2013), 이보람(2013) 등의 연구에 따르면 표준기록관리시스템은 인수, 검색, 전자기록 보존, 접근관리, 기준관리, 평가.폐기 등 여러 기능상의 한계로 인해 오히려 기록관리의 제약요소로 인식되고 있는 상황이다. 이소연(2015)에 따르면 표준기록관리시스템의 240개 세부기능의 활용현황 조사 결과 응답자의 반인 31곳 이상이 사용한다고 응답한 기능은 생산현황통보 등 26개에 그쳤다. 활용하지 않는 이유는 기능이 작동하지 않아서, 시스템이 너무 복잡해서, 기준정보 정비 등 초기작업이 완료되지 않아서 등이다(이소연 2015b). 표준기록관리시스템을 도입한 825개 기관의 도입비가 평균 3억 원 내외 수준이므로 적어도 2,500억 이상의 비용을 투자하여 10%의 기능만을 활용하고 있는 실정이다.

이러한 문제의 원인은 개발 과정의 미숙한 설계(이소연 2015)나 기록관리자들의 기능요건 표준에 대한 이해수준(임진희 2008) 때문이기도 하지만, 중앙에서 개발하여 배포하는 방식의 태생적 한계로 바라봐야 한다(안대진 외 2016). 표준기록관리시스템은 행정자치부 산하 국가기록원이 개발하여 중앙부처와 지방자치단체 등에 무상으로 배포하고 있다. 소프트웨어의 소스코드는 공개되지 않는다. 따라서 각 기관의 특수한 요구사항이나 기능개선 요구가 쉽게 반영되기 어렵다. 새로운 기능을 추가하기도 어렵다. 예를 들어 RFID와 같은 부가기능을 추가하고자 하여도 별도의 서버를 구축하여 표준기록관리시스템과 연계규격을 만들어 기록정보를 복사하여 이중 관리하는 방식을 취할 수 밖에 없다(오진관 외 2016). 현재와 같이 내부를 볼 수 없는 블랙박스 시스템인 상태에서는 기록공동체의 시스템에서 대한 연구와 개선을 꾀할 수 없다.

한편, 소프트웨어 분야에서는 90년대 후반부터 오픈소스 운동이 시작되어 상당한 성과를 거두고 있다(Raymond 1999). 소스코드를 개방하여 자유롭게 사용하도록 하는 독특한 인센티브 제도가 개발자들의 참여를 이끌어 내며 독점소프트웨어와 경쟁할 수 있는 새로운 기술혁신 방법론으로 인정

받게 된 것이다(송위진 2002). 그 결과 오픈소스 진영은 IBM, Sun, HP, 마이크로소프트와 같은 민간기업들을 참여시키는 데 성공했다. 최근에는 빅데이터나 클라우드, 인공지능, 사물인터넷(IoT) 등 빠른 혁신이 요구되는 분야일 수록 오픈소스 방식의 개발 프로젝트 비중이 높아지고 있다. 2017년 현재 클라우드, 빅데이터, 모바일, IoT(사물인터넷)에 걸쳐 100만 개 이상의 오픈소스 프로젝트가 진행 중이다(MICROSOFTWARE 2017). 최근 주목받는 아이디어들은 오픈소스 커뮤니티 기반으로 탄생하고 있다.

이러한 이유로 유럽연합, 영국, 프랑스, 독일, 미국 등은 정부 차원에서 오픈소스 소프트웨어를 강력하게 지지해 왔으며, 오픈소스의 개발과 구매 정책을 별도로 두고 있다(이철남 2002a). 국내에서는 2003년부터 정보통신부 주도로 공개소프트웨어 활성화 정책이 본격적으로 논의되기 시작했다(이철남 2002b). 문화체육관광부는 한국저작권위원회와 함께 2009년부터 ‘오픈소스 소프트웨어 콘퍼런스’를 개최하여 오픈소스 소프트웨어의 활용을 촉진하고 라이선스에 대한 올바른 인식을 확산하고자 노력하고 있다. 정보통신산업진흥원은 2009년 ‘공개SW 역량프라자’라는 조직을 신설하여 라이선스 검증 및 컨설팅 서비스를 제공하고 있다. 한국소프트웨어산업협회와 정보처리학회 등에서도 오픈소스의 도입과 한국형 오픈소스 전략에 대한 연구를 활발하게 수행하고 있다.

기록관리 분야에서는 2012년 인간과기억아카이브(hmarchives.com)가 설립되면서 오픈소스 기록시스템을 사용하고 기록공동체에 보급하는 활동이 시작되었다(안대진 외 2015). 그 결과 기획재정부, 동대문구청, 서울특별시 북부교육지원청, 국립문화재연구소, 부산대학교 로컬리티아카이브, 대한마이크로노동조합, 416기억저장소, 광명시평생학습원, 김세진이재호기념사업회, (사)한국여성의전화 등에서 에이투엠(AtOM)과 오메카(Omeka), 디스페이스(Dspace) 등 오픈소스를 사용하여 아카이브를 구축하였다.

국가기록원은 올해 전자기록관리 체계의 재설계를 위한 연구사업을 통해 전자기록관리 영역을 확대하고 신기술을 적용하기 위한 방안을 모색하고

있다(국가기록원 2017). 정부 업무환경이 최신 IT기술을 반영한 클라우드, 빅데이터 환경으로 급속히 전환되었기 때문이다. 클라우드, 인공지능 등 이른바 4차 산업혁명의 핵심기술들의 공통 키워드는 ‘분산’과 ‘연결’의 극대화라 할 수 있다. 최근 소프트웨어 분야의 가장 강력한 키워드인 마이크로서비스<sup>1)</sup> 역시 기능을 잘게 나누고 분산된 소프트웨어 컨테이너를 연결하여 시스템을 만드는 방식의 효율성을 강조한다(Wolf 2016). 이러한 개념은 모렉(MoReq)에서 제시하는 모듈화된 시스템을 만들 수 있는 진화된 방법론으로 보인다. 국가기록원이 이번 사업을 통해 정제된 표준기록관리시스템과 영구기록관리시스템을 오픈소스 컨테이너 기반의 유연한 도구로 탈바꿈시키는 기회로 삼았으면 한다. 서울특별시 또한 최초의 지방기록물관리기관으로서 보다 혁신적인 오픈소스 기록시스템을 만들고 확산시키는 계기가 되길 바란다.

### 3) 연구 내용과 방법

2014년 이후 국내에서는 오픈소스 소프트웨어를 이용하여 아카이브를 만드는 방법에 대한 연구가 활발히 이어졌다. 이보람 외(2014), 송정숙 외(2014), 설문원 외(2014), 현문수 외(2014), 최윤진 외(2014), 이동현 외(2014), 육혜인 외(2015), 심갑용 외(2015), 안대진 외(2015), 박태연 외(2016), 안대진 외(2016), 방기영 외(2016) 등이다. 대부분 특정 컬렉션을 디지털 아카이브로 구축하기 위해 에이투엠(AtoM)과 오메카(Omeka)를 어떻게 활용했는지를 다루고 있다. 국내 연구자들이 주로 관심을 둔 에이투엠과 오메카는 웹 기반의 소장기록 퍼블리싱 소프트웨어로 국내 공공기록관리 분야에서 말하는 기록관리시스템이나 영구기록물관리시스템과는 다른 카테고리에 해당한다.

---

1) 마이크로서비스(Microservices)는 소프트웨어의 기능을 독립적인 최소 단위로 쪼개 후 필요한 곳에 API로 전달하여 전체 시스템을 구성하는 소프트웨어 아키텍처이다.(출처: <https://martinfowler.com/articles/microservices.html>)

따라서 오픈소스 기반의 기록시스템<sup>2)</sup>을 설계하고 구축하기 위한 시사점을 제공하는 연구는 거의 진행되지 않았다.

이 연구에서는 국가기록원과 서울기록원이 좀 더 유연하고 확장 가능한 기록시스템을 구축하기 위해 기획자와 개발자 모두가 참고할 만한 오픈소스 전략을 제시하고자 한다. 오픈소스 프로젝트 경험과 이해가 부족한 국내 상황을 고려할 때, 당위성 설명을 위한 철학서와 실제 구축을 위한 실용서 모두가 필요하기 때문이다. 따라서 내용적 범위는 오픈소스화의 절차와 도구, 대표적 오픈소스 프로젝트의 개발 사례, 오픈소스화의 핵심 성공요소 등으로 설정했다.

2장에서는 오픈소스화 경험이 부족한 개발업체들이 고려할 사항들을 주로 알아보았다. 또한 오픈소스 운동이 시작된 90년대 후반에서 현재에 이르기까지 오픈소스가 표방하는 가치와 도구들이 어떻게 변화하며 성공해 왔는지를 함께 검토하였다. 이를 통해 기획자가 기관 내 오픈소스 전략 도입의 당위성을 설득하기 위한 논리를 제시하고자 하였다. 3장에서는 ICA-Atom, 오메카(Omeka), 아카이브매티카(Archivematica), 미국디지털공공도서관(DPLA) 플랫폼, veraPDF, DPSP 등 다양한 층위의 기관들이 주도한 오픈소스 프로젝트 사례를 조사하여 프로젝트가 탄생한 배경과 예산규모, 개발기간, 협력체계 등의 시사점을 도출했다. 4장에서는 프로젝트의 핵심 성공요소를 다루었다. 첫째, 프로젝트 성공에 가장 중요한 소프트웨어 개발방법론, 특히 최근 각광받고 있는 마이크로서비스(Microservices) 아키텍처의 개념과 특징을 다루었다. 둘째로, 현행 공공기록관리 체제에서 오픈소스 수요기관과 개발업체가 고려해야 할 법규와 요건을 검토하였다. 셋째, 오픈소스의 저작권과 라이선스 제약조건, 공공기관과 업체와의 계약조건 현황을 알아보았다. 그리고 마지막으로 커뮤니티 구성과 협력방식의 모형, 협력도구와 활용가

---

2) 이 연구에서 지칭하는 '기록시스템'이란 모렉(MoReq)의 'record system'을 말하며, 현용 기록관리(Record Management)와 보존기록관리(Archival Management)를 포괄하는 개념이다.



능성을 제시하였다.

연구방법은 문헌조사를 통한 사례연구이다. 오픈소스 프로젝트 전반에 대한 선행연구와 각 오픈소스 프로젝트의 도큐멘테이션, 깃허브에 공개된 소스코드와 개발문서 등을 분석하여 특징과 시사점을 정리했다.

이 연구의 한계는 사례로 선정된 6종의 오픈소스 프로젝트에 대한 조사 항목을 모두 충실하게 반영하지 못한 점이다. 개발에 투입된 예산의 규모나 참여한 업체의 정보 등이 공개되지 않은 경우 전체 프로젝트에 대한 문헌을 참고하여 작성하였다. 향후 기록관리 분야 이외의 오픈소스 프로젝트를 조사 대상에 포함한다면 보다 다양한 라이선스 전략이나 협력방식 등의 시사점을 얻을 수 있을 것이다.

## 2. 오픈소스화의 절차

### 1) 소스 '공개'의 의미와 전략적 이점

서울기록원 ISP에서는 오픈소스화의 기대효과로 네 가지를 꼽았다. 최신의 IT기술 습득, 비용절감, 특정 소프트웨어 독점체제(lock-in)의 견제, 오픈스탠다드의 준수 등이다(서울특별시 2016a). 하지만 추진 가능성을 타진하기 위한 비공식 논의에서 개발업체들은 우려를 표시했다. 자사 제품을 오픈소스화하거나 새로운 오픈소스 프로젝트를 추진할 경우의 수익모델을 확신할 수 없었기 때문이다. 그리고 오픈소스화한다는 것의 의미나 라이선스에 대한 이해의 수준도 달랐다. 이는 오픈소스화를 추진할 때 중요한 걸림돌이 된다. 오픈소스 프로젝트의 성공을 위해서는 정책을 실행하는 기관과 개발업체, 그리고 이용자들이 오픈소스 커뮤니티의 핵심 주체로 활동해야 하기 때문이다. 따라서 오픈소스 전략이나 비즈니스 모델에 대한 선행연구와 성공사례를 통해 이러한 우려를 어떻게 해소할 수 있는지 면밀히 살펴

볼 필요가 있다.

우선 소스코드 공개의 의미를 명확히 이해해야 한다. 오픈소스의 ‘오픈’은 개방성(Openness)과 투명성(Transparency)을 의미한다. 원본 소스코드를 공개하여 외부의 개발자와 이용자들을 참여시킴으로써 비공개 독점 소프트웨어보다 더 빨리 더 좋은 소프트웨어를 만들기 위한 것이다. 또 중요한 점은 소스코드를 공개하는 행위가 지적재산권인 저작권과 특허권의 포기를 의미하지는 않는다는 것이다. 오픈소스에도 저작권이 있다. 라이선스 조건이 다를 뿐이다. 오픈소스 소프트웨어 라이선스는 공공 도메인에 가까운 비-카피레프트, 약한 카피레프트, 그리고 상업적 이용을 제한하는 제한적 카피레프트로 구분할 수 있다.<sup>3)</sup> 오픈소스 라이선스들은 사용에 대한 권리뿐만 아니라 마음대로 복제 및 배포를 할 수 있도록 하고, 소스코드까지 제공하여 마음대로 수정할 수 있도록 허락한다. 하지만 상용 소프트웨어처럼 그에 따르는 로열티를 요구하지는 않는다. 대신 몇 가지 지켜야 할 의무사항을 요구한다. 의무사항의 대표적인 사례로는 1)저작권, 개발자, 기여자 정보의 표시, 2)코드를 수정한 경우 수정한 정보의 표시, 3)적용된 라이선스 정보의 제공, 4)동일한 라이선스로 재배포(카피레프트), 5)소스코드의 제공 등이 있다(한국저작권위원회 2016).

소프트웨어 개발업체들이 소스코드 공개를 통해 얻는 이점은 새로운 비즈니스 모델을 창출하는 것이다. 제품의 개발에 외부 자원을 끌어들이며 내부 개발자들만으로는 불가능했던 빠른 기능개선을 도모하고 폭넓은 이용자를 확보하는 것이다. 오픈소스화할 파트와 유료 부가서비스로 제공할 항목을 구분하여 전략적으로 접근하는 것이 중요하다.

---

3) 카피레프트(Copyleft)는 개인의 지적재산권을 중시하는 기존의 카피라이트(Copyright)에 대해 사회적 공유를 강조하는 정신이자 운동이다. 대표적인 비-카피레프트 라이선스는 BSD 계열의 BSD, MIT, Apache 라이선스가 있으며, 약한 카피레프트로는 LGPL과 MPL, 제한적 카피레프트로는 GPL이 있다. (출처: 한국저작권위원회).

〈그림 1〉 오픈소스 라이선스의 유형과 특징(Androutsellis-Theotokis 2011, 63)

		Zero cost	Distribution allowed	No usage restrictions	Source code available	Source code modifications	Linking with proprietary work	Derivative work can be proprietary	Can be relicensed by anyone	OSS license examples		
										GPL compatible	Not GPL compatible (reason)	
Rights abandoned ↑	Public domain	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes			
	OSS	Non-copyleft (permissive)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	BSD mod MIT/X11 Apache v2 AL v2 L-GPL	BSD orig (advertising) AL V1(patent termination)
		Weak copyleft	Yes	Yes	Yes	Yes	Yes	Yes	No	No		MPL (additional restrictions**) NPL (use of code in Netscape) SISL (minor details) SPL (like MPL) IBM CPL (choice of law) EPL (patent lawsuit language)
		Copyleft (restrictive)	Yes	Yes	Yes	Yes	Yes	No	No	No		<b>GPL</b>
	Freeware	Yes	Yes	Yes	No *	No	N/A	N/A	No			
Proprietary	No	No	No	No *	No	N/A	N/A	No				

\* Except under special licensing conditions - \*\* Provision in v1.1 to allow alternative license choice

## 2) 소스 공개를 위한 절차와 도구

오픈소스 라이선스 조항에 따라 소스코드 제공의무가 있을 경우 이를 공개해야 한다. 소스코드 공개의 범위는 대체로 오픈소스 소프트웨어 라이선스를 적용한 모든 영역의 실행 가능한 소스코드에 해당하며, 이용하는 데 필요한 정보 또한 반드시 포함되어야 한다(GitHub 2017). 소스코드 공개의 절차는 소스코드와 관련문서를 준비한 뒤 라이선스를 검증하여 웹사이트 등에 게시하는 순으로 진행된다.

우선 첫째, 준비단계이다. 이 단계에서는 소스코드와 관련문서, 로드맵, 라이선스 문서 등을 정리한다. 가장 먼저 빌드(build)해서 실행 가능한 핵심 기술의 전체 소스코드를 준비해야 한다. GPL 2.0<sup>4)</sup>에서는 ‘변경하기 좋은 형태(preferred form of the work for making modification)’의 소스코드

4) 일반 공중 사용 허가서(GPL, General Public License)는 자유소프트웨어재단에서 만든 자유소프트웨어 라이선스이다. GPL은 가장 널리 알려진 강한 카피레프트 사용 허가이며, 이 허가를 가진 프로그램을 사용하여 새로운 프로그램을 만들게 되면 파생된 프로그램 역시 같은 카피레프트를 가져야 한다(출처: <https://www.gnu.org/licenses/licenses.html#GPL>).

여야 한다고 명시하고 있다(GNU 2007). 일반적으로 다른 개발자가 쉽게 소스코드를 읽고 이해할 수 있도록 들여쓰기(indentation)를 정리하고 주석(comments)을 충실히 달아 두는 것이 좋다(GitHub 2017). 고의로 난독화시킨 경우 라이선스 상의 소스코드 공개 의무를 위반한 것으로 간주될 수 있다(정보통신산업진흥원 2014). GPL의 경우 원본저작물 뿐만 아니라 파생저작물(derivative work)의 소스코드까지도 공개하도록 하고 있다.

소스코드가 준비되면 적용할 오픈소스 라이선스와 프로젝트 설명문서를 작성한다 어떤 라이선스를 적용할 것인지는 여러 관점으로 고려해야 한다. 선행연구들은 의사결정에 필요한 다양한 고려사항을 제시하고 있다. 핵심 이용자(숙달된 이용자를 대상으로 하는 경우 허용적 라이선스가 참여를 촉진함), 기존 오픈소스와의 호환성(기존 GPL 소프트웨어에 대한 컴파일 의존성이 높을 경우 파생 저작물에도 GPL을 적용해야 함), 운영체제와 개발 환경(상용 OS 기반 프로젝트는 제한적인 라이선스를 채택하게 됨), 기업 참여여부와 상업화 계획(기업 참여가 많을 경우 GPL 등 제한적 카피레프트 라이선스를 꺼림), 동기 부여(보다 허용적인 라이선스는 더 많은 개발자들의 참여를 촉진시킴) 등이다(Androutsellis-Theotokis 2011). 가급적 새로운 라이선스를 만들기보다는 기존 오픈소스 라이선스 중 하나를 선택하는 것이 좋다(Fogel 2005). 잘 알려져 있고 신뢰받는 라이선스를 적용함으로써, 사용자들에게 사용허가 측면의 신뢰성과 명확성을 제공하고, 법적 분쟁의 위험을 줄일 수 있기 때문이다. 라이선스의 의무사항이 적으면 많은 참여를 유도할 수 있으나 강력한 표준체계를 유지하기 어려울 수 있다. 예를 들어 오디오/비디오 코덱인 Ogg Vorbis 프로젝트의 경우 BSD 라이선스 적용 후 여러 회사가 조금씩 다른 코덱을 배포하였고, 전 세계적인 업계 표준으로 확산되지 못했다(Engelfriet 2010).

둘째, 라이선스 검증 단계이다. 이 단계에서는 제품의 구현 과정에서 사용된 오픈소스 컴포넌트를 식별하고 각 라이선스에서 준수해야 할 사항들을 검토한다. 특허 등 민감한 지적재산권이 포함될 경우에는 적용되는 오

오픈소스 라이선스 의무사항과 활용형태(커널, 애플리케이션, 장치 드라이버)에 따라 다양한 경우가 발생하므로 이의 식별과 조치를 위해 라이선스 및 법률 전문가에게 의뢰하여 정확한 판단을 받아야 한다. 미래창조과학부 산하 “공개SW역량프라자”에서 제공하는 “공개SW 라이선스 검증 서비스”<sup>5)</sup>나 Protex, FOSSology 등의 오픈소스 라이선스 검증도구를 이용하면 보다 확실하게 라이선스를 검증할 수 있다. 이러한 검증 툴들은 바이너리 코드, 이미지, 폰트, 주석의 문자열 등을 분석하여 적용된 오픈소스 코드를 식별한다. 적용된 오픈소스 소프트웨어 파일과 준법성 위반 목록은 결과보고서로 제공된다. 컨설팅을 통해 충돌되는 컴포넌트의 소스코드를 변경, 삭제, 대체, 분리하는 등의 조치를 취해야 한다.

셋째, 배포 단계이다. 일반적인 소스코드 제공 방법은 웹사이트에 게시하는 것이다. 최근에는 오픈소스 프로젝트의 소스코드 저장소인 깃허브(GitHub)를 이용하는 경우가 가장 많다. 가장 많은 이용자를 보유한 사이트이므로 프로젝트를 홍보하기 쉽고, 버전관리나 코드작성도구를 함께 제공하여 협업이 쉽기 때문이다. 깃허브에서는 해당 프로젝트의 리포지토리를 생성한 후 소스코드와 라이선스 문서(LICENSE.md 또는 LICENSE.txt), 프로젝트를 소개하는 README.md 파일, 소스코드 기여자를 위한 가이드라인, 커뮤니티의 행동강령(Code of Conduct) 등을 같이 공개하도록 하고 있다.

### 3) 소프트웨어 개선을 위한 협업과정

성공적인 오픈소스 프로젝트들은 대부분 다수의 이용자 기반 커뮤니티가 구축되어 있다. 또한 효과적이고 빠른 의사결정 체제나 협업도구를 제공한다. 오픈소스 프로젝트의 참여자들은 지리적으로 분산되어 있고 느슨하게 연결되어 있다. 따라서 다양한 배경과 관심사를 가진 이들을 효과적으로

---

5) 공개SW 라이선스 검증 서비스. 검색일자: 2017.3.30. [http://www.oss.kr/oss\\_business9\\_2](http://www.oss.kr/oss_business9_2)

참여시킬 수 있는 동기부여 요인이나 커뮤니티 구조, 그리고 기술인프라와 협업도구들이 프로젝트 성공의 핵심 요소가 된다.

우선 커뮤니티는 다양한 이해당사자들로 구성되어야 한다. 개발을 주도하고 의사결정을 하는 핵심멤버, 개발자, 이용자 등이다. 그 외에도 번역자, 버그 리포터 등이 있을 수 있다. 의사결정 구조는 프로젝트 성격에 따라 초기 주도자나 소유자의 비전에 따르는 형태, 핵심 멤버들이 투표하여 민주적으로 방향을 결정하는 형태 등이 있다. 오픈소스 프로젝트의 조직 구조는 수평적이며, 자유롭고, 무정부주의적인 특성 뿐만 아니라, 커뮤니티 구성원 간에 합의된 명확한 계층적, 체계적 리더십 모델의 결합으로 이루어진다(Androutsellis-Theotokis 2011). 따라서 참여자의 기여도나 기술적 능력, 지식, 헌신, 평판 등에 근거한 능력 중심의 프로세스를 통해 리더십이나 의사결정의 공정성과 신뢰성을 확보하는 것이 중요하다. 이러한 민주적인 의사결정 체계와 규칙은 외부 참여자들이 해당 프로젝트를 신뢰하고 참여하는 기반이 될 것이다.

오픈소스 프로젝트의 개발 프로세스는 일반적으로 요구사항 정의, 새로운 기능의 결합, 코드 통합, 릴리스 등으로 이루어진다. 요구사항은 전체 커뮤니티 멤버들의 상호작용을 통해 도출된다. 보통 요구사항이 공식적으로 문서화되지는 않으며, 포럼, 이메일 문서, TODO 리스트, 버그리포트, 이슈 트래킹 시스템 등에 요구사항이 숨겨져 있다(Mockus, Fielding & Herbsleb 2002). 그 중 개발할 기능을 선택하고 우선순위를 결정하는 것은 핵심그룹의 역할이다. 일반적으로 새로운 기능을 결합하는 과정은 간략한 설명이나 컨셉에 대한 프로토타입의 생성, 후보 기능 중 선택을 위한 투표, 특정 요구사항의 수집, 그리고 최종적으로 이를 설계하고 구현하는 절차를 따른다(Androutsellis-Theotokis 2011).

그 이후에는 소스코드 저장소에 코드를 통합하게 된다. 코드 통합의 일반적인 과정은 코드 리딩, 개발 및 테스트, 패치 제출, 검토 및 사전 커밋 테스트(Commit Testing), 코드 커밋(Code Commitment)<sup>6)</sup> 등으로 이루어진

다. 개발자의 코드 변경은 동료 검토나 승인, 커밋 테스트 과정을 거쳐 소스코드 리포지터리에 반영된다. 큰 프로젝트의 경우 여러 단계를 두기도 한다. 코드 커밋 단계에는 코드 품질점검을 위한 자동화된 테스트나 식별된 이슈의 자동화된 등록, 관심 사용자에게 대한 자동 통지 등이 포함된다.

이렇게 협력적 방식으로 커밋이 축적되고, 충분히 중요한 숫자의 버그나 기능이 개선되면 새로운 버전의 배포가 이루어진다. 배포를 위해서는 모든 새로운 코드 패치들을 코드 스트림으로 병합하고, 압축된 소스코드나 바이너리 패키지, 또는 이전 버전 사용자를 위한 패치파일, 이들을 모두 포함한 하나의 설치파일 등의 여러 가지 포맷으로 패키징해야 한다. 새로운 릴리스는 일반적으로 주 배포 서버에서 배포되지만 미리 역할의 보조 서버, P2P 파일공유 네트워크 등을 통해서도 배포될 수 있다.

오픈소스 프로젝트는 검증된 기술 인프라에 많은 것을 의존한다. 분업을 위한 기술인프라와 협업도구로는 소스코드관리시스템(SCM, Source Code Management Systems), 버전관리시스템(VCS, Version Control Systems), 이슈 추적시스템(Issue Tracking Systems), 소스코드 저장 및 호스팅 인프라(GitHub 등), 위키, 메일링리스트, IRC(동기식 커뮤니케이션), 번역 플랫폼 등이 있다.

#### 4) 시사점

첫째, 프로젝트 참여자들이 오픈소스화의 비전과 전략적 이점을 공유해야 한다. 먼저 국가기록원이나 서울특별시 등은 소스코드 ‘오픈’이 우리 정부가 지향하는 개방성이나 투명성과 맞닿아 있음을 인식하고 홍보 이미지로 적극 활용해야 한다. 그리고 개발업체가 참여할 수 있는 라이선스 전략

---

6) 커밋(Commit)은 버전관리시스템에서 사용되는 기능으로, 자신이 작업한 파일의 업데이트를 저장소에 반영시키는 것을 말한다. 버전관리시스템은 기존 정보와 업데이트된 파일과의 차이를 분석하여 업데이트된 데이터만 저장소에 업로드한다(출처: GitHub, 검색 일자: 2017.3.30. <https://help.github.com/articles/github-glossary/#commit>)

을 고민하고, 수요기관들로 확산시키기 위한 판을 만들어 줘야 한다. 서울 기록원 ISP에서 국내 업체들이 우려했던 수익성 문제는 적절한 라이선스의 적용으로 어느 정도 해결 가능하다. 개발업체들 역시 기존 제품의 경쟁력을 향상시켜 새로운 수익원을 만드는 기회로 인식할 필요가 있다. 오픈소스화를 통해 기록관리에서 타 분야로 고객이 확장될 수도 있다. 소프트웨어 비즈니스의 흐름은 오픈소스 방식으로 전환되고 있다. 수요기관들은 특정 업체에 종속되지 않으려 하고, 저렴한 비용으로 빠르게 기능이 개선되길 바라기 때문이다. 오픈소스의 효율성을 외면하고 비즈니스 모델을 전환하지 않으면 시장에서 도태될 공산이 크다. 개발업체 입장에서는 이러한 리스크를 없애는 것 자체가 전략적 이점이 된다. 따라서 보다 능동적으로 오픈소스의 여러 비즈니스 모델과 라이선스 전략을 검토할 필요가 있다. 기록연구사 등 이용자들은 기존 기록시스템의 불편함을 스스로 해결하기 위해 적극적 테스터가 되어 핵심적인 요구사항과 아이디어를 모아야 한다.

둘째, 오픈소스는 새로운 시장을 형성할 수 있다. 현재 아카이브나 기록관리 시스템을 설계하고 개발하는 시장은 거의 정체되어 있다. 2007년 이전에 자료관시스템 규격을 만들어 10개 업체들이 참여하였지만 하나둘 빠져나가며 시장 형성에는 실패했다. 특정 업체의 가격 덤핑과 국가기록원의 중앙 공급정책이 원인이었다. 표준기록관리시스템의 경우 수요기관이 특정 기능을 변경하려 해도 국가기록원이 소스코드를 쉽게 제공하지 않아 커스터마이징이 제한될 수밖에 없다. 표준 프로세스를 통제하기 위해서라면 각 기능에 대한 필수 규격을 두되 나머지 기능은 각 기관의 상황에 따라 변경이 가능하도록 하면 된다. 오픈소스화는 정체된 시장을 새로 만들기 위한 유일한 방법이다. 또한 개발업체의 기록관리 시장 진입장벽을 낮추고 산발적 개발을 통해 지속적인 기능개선을 가능하게 한다.

셋째, 정보화 시대의 부가가치 패키징이 필요하다. 제4차산업혁명의 도래로 기술.산업적 측면에서는 융합을 통해 새로운 스마트 비즈니스 모델과 산업생태계가 만들어지고 있다. 사물인터넷(IoT), 클라우드, 빅데이터, 인공



지능 등 초연결성에 기반을 둔 플랫폼 기술들은 대부분 오픈소스 기반이다. 발주기관과 기업 모두 이러한 기술들을 활용하여 새로운 부가 가치를 만들어내야 할 시점이다. 예를 들어 빅데이터 분석을 이용한 기록 내용 및 이용자 요구사항 분석, 머신러닝 기반의 온톨로지 구축이나 기록관리 업무 자동화 등이다. 이러한 혁신을 통해 기록시스템에 새로운 가치를 부여할 수 있을 것이다. 혁신의 동인은 내외부 모두에 있다. 외부의 인력(오픈소스 개발자)과 자원(오픈소스 컴포넌트)을 적극적으로 활용하지 못하면 시장이나 직군 자체가 없어질지도 모른다.

넷째, 개발 주체가 누구냐가 중요하다. 프로젝트의 지속과 확산을 위해서는 강력한 거버넌스와 효과적인 협력체계가 조화를 이루어야 한다. 해당 분야를 대표하는 국제 수준의 거버넌스 조직이 주도하는 경우 영향력과 자금력을 바탕으로 개발업체와 이용자를 커뮤니티에 참여시키기 수월할 것이다. 지속적인 자금공급이 가능한 조직의 경우 소프트웨어의 장기적인 발전을 도모할 수 있을 것이다. 또한 공동의 요구사항을 가진 기관들이 협력하거나 특정 업체의 주도로 개발되는 경우도 있을 것이다. 이러한 여러 경우 별 특징을 3장의 사례를 통해 알아보았다.

### 3. 기록시스템 오픈소스화 사례

3장에서는 기록관리 분야의 대표적인 오픈소스 프로젝트의 진행 과정을 분석하여 시사점을 도출하였다. 종합적인 시사점을 얻기 위해 프로젝트의 개발 주체를 국제기구(ICA), 국제협력체계(EU, PREFORMA), 국립 아카이브(NAA), 대학 및 학술단체(RRCHNM, DPLA), 개발업체(Artefactual Systems, Inc.) 등으로 구분하였다. 오픈소스 프로젝트는 기록시스템이나 디지털 보존과 관련된 제품으로 한정하였다. 분석대상 항목은 프로젝트의 추진배경과 목적, 개발과 펀딩의 주체, 거버넌스 모델, 개발기간과 비용, 프로젝트의 가

차와 지향, 비즈니스 모델과 아키텍처, 커뮤니티와 라이선스 전략 등이다.

〈표 1〉 기록관리 분야 오픈소스 프로젝트 비교

	ICA- AtoM	Omeka	Archivematica	DPLA	veraPDF	DPSP
	ICA	RRCHNM	Artefactual	DPLA	OPF, PDF Association	NAA
	Artefactual	RRCHNM	Artefactual	DPLA	OPF, PDF Association	NAA
( )	UAE , LAC, , ,	, IMS, ,		NEH	PREFORMA	NAA
	692,500 (est)	495,000 US	730,000 US	4,510,000 US	4,758,014	2000,000 AUD
7)	5 (2006- 2010)	2 (2007- 2009)	2 (2009- 2010)	2 (2011- 2013)	2 (2015- 2016)	2 (2001- 2003)
8)	Tools, Metadata Processing	Tools, Metadata Processing	Tools, Preservation System, File Format Migration	Tools, Metadata Processing	Tools, PDF format, Validation, Document	Tools, File Format Migration, Preservation System
/				GLAM	PDF/A	
		GLAM	GLAM	GLAM	GLAM	GLAM
	ICA- AtoM	RRCHNM		DPLA	VeraPDF	NAA
( )	MC(Symfony, PHP)	MC(Zend, PHP)	MVC(Django, Python), Microservices	MC(Ruby on Rails, Ruby)	MC(Struts 2, Java)	
	AGPLv3	GPLv3	AGPLv3	GPLv3	GPLv3+MPLv2	GPLv3
	2.4	2.5	1.6	3.2.2	1.0	6.1

7) 알파와 베타 단계부터 정식 버전이 릴리즈될 때까지를 개발기간으로 잡았다. 소프트웨어의 생명주기(Software Release Life Cycle)에서 알파(Alpha)는 소프트웨어 테스트를 시작하는 첫 단계를 말한다. 모든 기능이 구현되지는 않았지만 주요 요구사항을 만족하는 버전이다. 베타(Beta)는 소프트웨어 기능이 완성되는 단계로 요구사항의 모든 기능을 구현했지만 속도나 성능 문제 등 많은 버그가 존재한다.

## 1) ICA-AtoM : 글로벌 거버넌스 주도 개발

ICA-AtoM은 웹 기반의 소장기록물 기술을 위한 소프트웨어다(ICA-AtoM, n.d.). ICA는 기록물 소장기관들의 소장물을 온라인을 통해 접근 가능하도록 지원하기 위해 무료 오픈소스 소프트웨어를 개발하여 보급하고자 했다. ICA-AtoM 프로젝트는 2005년 ICA의 프로그램위원회(PCOM, Program Commission)의 주도로 시작되었다. ICA는 7명의 위원으로 운영위원회(ICA-Atom Steering Committee)를 구성하여 프로젝트를 진두지휘했고, 캐나다의 아카이브 소프트웨어 전문업체인 아티팩추얼( Systems, Inc.)이 리드 개발사로 참여했다.

ICA-AtoM 프로젝트는 여러 기관으로부터 풀뿌리 자금을 지원받아 지속되었다. 이러한 펀딩의 문제로 베타버전을 공개하기까지 5년이나 걸렸고 개발비는 약 69만 유로가 소요되었다. 개발팀은 2005년 유네스코로부터 45,000 유로를 제공받아 2006년에 프로토타입인 v0.1과 v0.2를 개발했다. 이후 네덜란드의 Archiefschool 등으로부터 추가 자금을 지원받아 2008년 ICA 총회에서 베타 버전을 공개했다. 이 버전은 아카이브 기능을 수행하는 첫번째 소프트웨어로 총 30개 기관이 테스트에 참여하였다. ICA 총회에서 열린 ICA-AtoM 운영위원회에서는 보다 완전한 제품인 v1.2를 2009년 가을까지 개발하길 원했다. 하지만 ICA 관리팀에서는 ICA-AtoM의 프로젝트 비용이 기존의 ICA 프로그램이나 프로젝트보다 높은 데 대한 우려를 표명했다. 그러나 디스페이스(DSpace) 개발비 2백만 US달러(2002), 페도라(Fedora) 개발비 4백만 US달러(2007)에 비하면 ICA-AtoM의 프로젝트 비용은 합리적인 수준이었다. 결국 ICA-AtoM의 운영위원회는 유럽연합의 자금조달 모델(EU funding request)을 제안했고, ICA-AtoM 프로젝트의 비전과 목표를 공유하는 주요 기관들로부터 보다 적극적인 자발적 기부를 요청했다. v1.0 출시 이후 ICA는 캐나다도서관기록관협회(Library and Archives Canada) 등으로

---

8) 아카이브 소프트웨어 레지스트리인 COPTR([http://coptr.digipres.org/Main\\_Page](http://coptr.digipres.org/Main_Page))의 기능 카테고리(Functional Category)를 참고하여 작성하였다

부터 추가 자금을 지원받아 2010년 11월에 첫번째 공식 릴리즈인 v1.1을 공개했다. 그러나 ICA가 거버넌스 모델을 재정의하며 v1.2의 개발 로드맵은 축소되었다. 이후 개발사인 아티팩추얼은 ICA-AtoM의 리드 개발사에서 공식적인 써드파티 계약자로 변경되었고, ICA의 공식적인 자금 지원은 v1.3.1까지만 제공되었다. 이후 개발사인 아티팩추얼은 ICA의 직접적 관여 없이 소프트웨어 개발을 지속하여 AtoM<sup>9)</sup>이라는 별도의 제품명으로 2.3.1 버전(2017년 기준)까지 발전시켜 왔다(Garderen 2009). ICA-AtoM의 소프트웨어 아키텍처는 모듈화를 기반으로 하며 PHP 기반의 웹 어플리케이션 프레임워크인 심포니(Symfony)가 적용되었다.

ICA-AtoM이 추구하는 가치는 기술적 전문성과 실무경험을 전 세계 기록 소장기관에 공유하고, 사용자와 개발자들의 활발한 커뮤니티를 양성하며, 표준 웹 기술을 수용하는 것 등이다(ICA, n.d.). 그 중에서도 ICA의 브랜드를 이용해 충분한 재정과 기술을 보유한 조직을 참여시키고, 이러한 비즈니스 모델을 통해 소프트웨어를 확산시키려 한 점은 주목할 만하다. ICA의 비즈니스 모델은 커뮤니티 기반의 바운티 모델(Bounty model)이다. 이는 커뮤니티 내 자금력을 보유한 기관의 커스텀 개발 계약에서 얻은 자금을 후속 버전의 개발을 위해 쓰는 방식이다. 이 방식은 현재까지도 유지되고 있지만 ICA의 공식 자금지원이 종료된 이후부터 문제가 생기기 시작했다. 소규모 회사인 아티팩추얼은 커뮤니티의 자금지원 없이는 광범위한 개발이나 유지보수 작업을 전담하기 어려웠다. 또한 영리 목적의 개인회사로서는 주로 비영리단체나 기관을 대상으로 하는 문화유산 프로그램의 자금을 지원받을 수도 없었다. 기술적으로는 보다 현대적인 소프트웨어 구조와 협력 체계가 필요했다. 한 기관이 개발한 새로운 모듈을 향후 타 기관에서 이용하고자 할 때 요구되는 리소스를 최소화할 수 있는 기술적 구조가 필요했다. 구체적으로는 ICA-AtoM의 라이브러리나 프레임워크 종속성을 최소화하

---

9) AtoM(Access to Memory). Retrieved March 30, 2017 from <https://www.accesstomemory.org/>

는 핵심적인 작업이 요구되었다. ICA-AtOM은 UNESCO, 월드뱅크그룹(World Bank Group) 등 전 세계 수백 여 기관이 이용하고 있다. ICA 기술표준 4종 뿐만 아니라 국가별 기술표준인 RAD, DACS, 범용 표준인 더블린 코어(DC) 등을 지원하여 주로 북미 지역의 도서관과 아카이브들이 커뮤니티가 개발에 참여하고 있다.

ICA-AtOM은 카피레프트 중 가장 강력한 라이선스 중 하나인 AGPLv3<sup>10)</sup>를 적용하고 있다. ICA-AtOM이 추구하는 전 세계인의 자유로운 접근을 위해 소스코드 뿐만 아니라 프로젝트 개발 과정의 모든 문서, 웹 세미나, 위키, 프리젠테이션 등도 공개하고 있다. ICA-AtOM의 근간이 되는 여러 라이브러리와 큐빗 툴킷(Qubit Toolkit), 심포니 웹 프레임워크(Symfony Web Application Framework)는 모두 AGPLv3 라이선스와 호환된다. 따라서 ICA-AtOM의 소스코드에 대한 저작권은 모두 원 저작자에게 귀속된다. 이러한 전략은 밴쿠버 시 아카이브, 브리티시컬럼비아주 아카이브의 자금으로 만들어진 개발 결과물을 ICA-AtOM의 기능으로 축적시키며 지난 10여 년 동안 프로젝트를 지속시켜 온 원동력이 되었다. ICA-AtOM의 현재 버전은 v2.3.1(2017년 2월 배포)이며 소스코드는 홈페이지(<http://ica-atom.org>, <http://accesstomemory.org>)와 깃허브 리포지토리를 통해 배포된다. AtOM 프로젝트는 아티팩추얼이라는 전문업체가 호스팅, 유지보수, 교육 등의 유료 추가서비스를 제공하는 전문 오픈소스(Professional Open Source)에 해당한다.

## 2) Omeka : 대학 주도 개발

오메카(Omeka)는 소규모 도서관, 박물관, 아카이브를 위한 표준 기반의 기록물 웹 퍼블리싱 소프트웨어이다(Omeka, n.d.). 이 프로젝트는 IT 비전 공자인 GLAM기관(Gallery, Library, Archives, Museum)의 직원이 소장물을

---

10) GNU Affero General Public License. Retrieved March 30, 2017 from <https://www.gnu.org/licenses/agpl.html>

쉽게 온라인 콘텐츠로 작성하고 많은 이용자들을 쉽게 참여시키는 것을 목표로 한다. 따라서 관리자 페이지와 서비스 페이지 모두 아주 직관적인 이용자 인터페이스를 제공하며 어노테이션(annotation) 등 다양한 웹 2.0 기술을 적용하였다.

오메카 프로젝트는 2008년 조지 메이슨 대학교의 역사뉴미디어센터(RRCHNM, Roy Rosenzweig Center for History and New Media)의 주도로 시작되었다. RRCHNM은 1994년 설립된 이래 150여 개에 이르는 오픈소스 프로젝트와 디지털 콘텐츠 사이트를 직접 개발하며 미국 내 역사교육 및 디지털 인문학 분야에서 주도적인 역할을 수행하고 있다. 1998년부터 2014년까지 67개의 다양한 기금으로부터 총 2,760만 US달러를 지원받아 149개 프로젝트를 직접 개발했다(RRCHNM, n.d.). 오메카 프로젝트는 현재 관리자, 연구자, 개발자, 코디네이터 등 9명의 RRCHNM 직원이 관여하고 있으며, 알파 단계에서는 미네소타 히스토리컬 소사이어티(MHS, Minnesota Historical Society)의 연구자들이 파트너로 참여하였다.

오메카 프로젝트는 앤드류 멜론 재단(Andrew W. Mellon Foundation), IMLS(Institute of Museum and Library Services), 슬로언 재단(Alfred P. Sloan Foundation), 크레스 재단(Samuel H. Kress Foundation) 등으로부터 약 50만 US달러의 기금을 지원받아 2007년부터 약 2년 간 개발되었다. RRCHNM은 여러 오픈소스 프로젝트를 성공시킨 경험을 바탕으로 초기 3년간 18개의 버전을 릴리즈했으며 오메카의 호스팅 버전인 Omeka.net의 개발도 완료했다. 2009년 6월에 정식 버전인 v1.0이 공개되었다. 2010년 10월 Omeka.net 베타 버전이 공개되며 본격적으로 서비스가 시작되었고, 2년 후인 2012년에는 3,800개의 사이트로 이용자 커뮤니티가 확산되었다.

오메카 프로젝트는 초기 개발 단계 이후에도 지속적으로 공공 및 비영리 민간단체로부터 자금을 지원받아 기능을 발전시켰다. 공간 기반 콘텐츠를 작성할 수 있는 니트라인(Netline) 플러그인의 경우 NEH(National Endowment of Humanities)로부터 66만 US달러를 지원받아 개발되었다. 현재 홈페이지

에 공개되어 있는 플러그인의 갯수가 93개임을 감안하면 전체 소프트웨어 개발비는 상당한 규모임을 짐작할 수 있다. 오메카의 소프트웨어 구조는 다양한 기능이 쉽게 추가될 수 있도록 모듈화되어 있으며, 이를 위해 PHP 기반의 웹 프레임워크인 젠드(Zend)가 적용되었다.

오메카 프로젝트는 일반 이용자들이 쉽게 소장물을 이용한 웹 콘텐츠를 만들도록 하여 비용을 절감하게 하고 역사 교육에 크게 이바지했다. 오메카 프로젝트의 기술적 포지션은 웹 콘텐츠관리시스템, 컬렉션 관리시스템, 그리고 박물관관리시스템에 두루 걸쳐 있다. 따라서 이용자 커뮤니티도 문화유산 및 학술콘텐츠 분야의 연구자, 교사, 학생, 도서관의 사서나 아키비스트, 박물관의 큐레이터나 웹 디자이너, 교육자, 솔루션 공급자 등으로 다양하게 구성되어 있다. 오메카 프로젝트의 비즈니스 모델은 ICA-AtoM과 마찬가지로 전문업체의 부가 서비스이다. 2010년부터 RRCHNM은 오메카의 커스텀 디자인과 개발, 그리고 호스팅 서비스를 제공하며 수익창출 기반을 마련했다. 하지만 전문 서비스를 통한 수익은 소프트웨어의 개발 및 유지비용에 비하면 극히 일부에 지나지 않는다. 오메카 프로젝트가 전 세계적으로 확산되며 큰 성공을 거둔 이유는 공공과 민간의 지속적인 개발자금을 확보할 수 있었기 때문이다. 이는 미국의 역사교육이나 디지털 휴머니티, 오픈소스에 대한 지원정책 때문이다. 일례로 미국 교육부에서 추진하는 TAH(Teaching American History) 프로그램의 경우 2002년부터 5년간 역사 교육 콘텐츠와 오픈소스 개발에 7억 US달러 이상의 자금을 지원했다. 이러한 자금지원을 바탕으로 RRCHNM은 현재 45명의 상근직원과 8명의 협력 직원으로 구성된 거대한 전문가 조직이 되었다. RRCHNM 뿐만 아니라 버지니아대학도서관의 Scholars' Lab, 클리블랜드대학교의 CPHDH 등에서 개발한 오픈소스 프로젝트들이 지속적인 자금지원을 통해 유지되고 있다. 또한 프로젝트끼리 소스코드를 재사용하고 연계하며 선순환하는 생태계가 형성되고 있다. 오메카 프로젝트 역시 큐레이트스케이프(Curatescape), Omeka.net, Omeka S 등 별도의 프로젝트로 확산되고 있으며, DPLA(dp.la)에서도 기록

물 전시에 오메카를 이용하고 있다.

오메카 프로젝트는 GPLv3 라이선스를 적용하고 있다. GPLv3는 가장 대표적인 카피레프트 라이선스로, 파생 저작물의 소스코드 공개와 라이선스 유지를 의무화하고 있다. 오메카는 공식 홈페이지와 RRCHNM의 깃허브 리포지토리를 통해 소스코드와 도큐멘테이션을 배포하고 있다. 현재 버전은 2.5이다.

### 3) Archivematica : 기업 주도 개발

아카이브매티카(Archivematica)는 오픈소스 기반의 디지털 보존 소프트웨어이다(Archivematica, n.d.). 아카이브매티카 프로젝트의 목적은 OAIS(ISO 14721) 참조모델과 PREMIS 등 디지털 보존 표준을 충족하는 통합 솔루션을 전 세계의 커뮤니티에 보급하는 것이다. 솔루션에는 오픈소스 소프트웨어 뿐만 아니라 기능명세서, 스토리지 서비스(Storage Service)와 포맷정책레지스트리(FPR, Format Policy Registry) 등이 포함된다. 2007년 UNESCO는 “오픈소스 리포지터리와 보존시스템의 방향(Towards an Open Source Repository and Preservation System)”이라는 보고서에서 OAIS를 준수하는 오픈소스 기반의 디지털 보존시스템이 필요함을 제안했다. 디지털 객체를 처리하는 단순한 툴들이 많으므로 약간의 작업을 통해 오픈소스 보존 시스템이 만들어질 수 있으며, 이 툴은 전 세계의 커뮤니티와 기관에 큰 혜택을 줄 것이라는 것이다(Bradley, Lei & Blackall 2007). 이 보고서에 따라 유네스코는 여러 기능을 한 시스템에서 구현할 수 있는 시스템인 아카이브매티카의 개발 자금을 아티팩추얼(Artifactual Systems, Inc.)에 지원하기로 했다. 비슷한 시점에 디지털 보존 시스템을 개발하려 했던 캐나다의 밴쿠버시아카이브와 아카이브 프로토타입을 구상하던 국제통화기금(IMF) 아카이브 역시 아카이브매티카의 개발을 위해 아티팩추얼과 파트너 계약을 체결했다. ICA-AtOM과 마찬가지로 바운티 모델로 오픈소스 프로젝트가 시작된 것이다. 파트너



기관들은 개발비용을 지원하기도 하지만 내부 개발인력이나 인프라 등의 자원을 제공하고 프로젝트의 방향을 논의하는 거버넌스의 주체로도 활동한다.

아카이브매티카 프로젝트는 유네스코의 기금과 개발사인 아티팩추얼이 체결한 두 건의 계약을 통해 초기 개발자금과 인력을 마련했다. 전체 개발 자금의 액수는 파악할 수 없으나 밴쿠버시아카이브 프로젝트의 경우 2008년 11월부터 2010년 12월까지 2년간 73만 US달러가 소요되었다. 밴쿠버시아카이브는 시의 전자문서시스템(ERDMS)과 기록관리시스템(VanDocs), 산하기관의 다양한 시스템에 보관되어 있는 디지털 기록, 그리고 아카이브가 통제할 수 없는 민간의 개인기록들을 영구 보존하고 접근을 제공해야 하는 과제에 직면해 있었다. 프로젝트팀은 공공기록과 개인기록 보존으로 프로젝트 개발 단계를 나누고 각 사례별로 OAIS 기능 모델을 유스케이스 시나리오로 작성한 후 프로토타입으로 구현하는 데 성공했다. 예를 들어 무결성 체크, 파일포맷 식별, 바이러스 체크, 메타데이터 추출과 처리, 포맷 변환, 접근 제공 등의 순으로 처리의 단계를 구분하고 각 단계를 구현할 수 있는 오픈소스 컴포넌트 23종을 하나의 소프트웨어로 묶는 식이었다. 밴쿠버시아카이브의 사례는 InterPARES3 프로젝트 보고서의 케이스스터디 16에 상세히 소개되어 있다(The InterPARES 3 Project 2014).

2010년에는 프로젝트의 리더가 개발 성과를 논문으로 발표하고 국제 학술 세미나에 참석하여 아카이브매티카를 전 세계의 보존 커뮤니티에 소개했다. 이러한 노력을 바탕으로 더 많은 기관 파트너가 기능 구현을 위한 자금을 제공했다. 뿐만 아니라 학술 세미나는 디지털 큐레이션 커뮤니티로부터 시스템 개선을 위한 다양한 조언을 얻었다. 2011년에는 v1.0이 공식 릴리즈되었고 이후 IMF 아카이브, 뉴욕현대미술관(MoMA) 등의 수많은 개별 프로젝트가 이어지며 발전하고 있다. 아카이브매티카의 소프트웨어 구조는 마이크로서비스 기반으로 철저하게 모듈화되어 있으며 파이썬 기반의 장고(Django) 프레임워크가 적용되었다.

아카이브매티카의 가치는 디지털 보존 분야의 모범실무를 오픈소스 소프

트웨어로 구현하고 확산시킨 점이다. EAD, METS, MODS, PRONOM, DROID, PREMIS, TRAC, OAI, BagIt, Planets 등 90년대부터 축적된 디지털 보존 분야의 국제적 연구 성과와 소스코드를 재사용하여 하나의 소프트웨어에 집결시킨 것은 대단한 성과이다. 실제로 아카이브매티카에는 수십 종의 오픈소스 컴포넌트들이 재사용되었다. 모범실무와 최신기술을 기민하게 반영하기 위한 소프트웨어 아키텍처와 개발방식 또한 적용했다. 시장에 존재하는 다양한 오픈소스 툴들을 컴포넌트로 느슨하게 연결하기 위해 파이썬 기반의 마이크로서비스 아키텍처를 구현하였고, 반복하여 테스트하고 자주 배포하는 애자일 개발방식을 도입하였다.

아카이브매티카의 비즈니스모델은 AtoM과 마찬가지로 커뮤니티 기반의 바운티 모델이다. 파트너로 참여한 전 세계의 기관들은 개별적인 파일럿 프로젝트를 통해 아카이브매티카를 커스텀 개발하였고 그 결과물의 소스코드가 아카이브매티카로 축적되며 발전하는 형태이다. 개발 파트너가 후원하는 기능과 개선사항이 우선 개발된다.

아카이브매티카는 강력한 카피레프트 라이선스인 AGPLv3를 적용하고 있다. 여러 기관의 개별 프로젝트 산출물이 지속적으로 공개되어 축적될 필요가 있기 때문이다. 소스코드 및 문서화 페이지는 홈페이지(<http://archivematica.org>)와 깃허브 사이트(<https://github.com/artefactual/archivematica>)를 통해 공개하고 있다. 현재 아카이브매티카의 최신버전은 v1.6, 스토리지 서비스는 v0.10, 포맷정책레지스트리는 v1.1.0이다.

#### 4) DPLA : 비영리단체 주도 개발

DPLA(Digital Public Library of America)는 미국의 도서관, 아카이브, 박물관 등이 소장한 디지털 자료의 메타데이터와 섬네일 이미지를 통합하여 제공하는 디지털 자원의 허브이다. 2015년 기준 1,400여 협력기관들이 제공한 약 1,110만 개의 자원정보를 제공한다. DPLA 프로젝트의 목적은 미국 내 여러 문화유산기관의 메타데이터를 통합하고 효과적으로 서비스하는 것이

다. 이를 위해 독자적인 데이터 모델(DPLA MAP, DPLA Metadata Application Profile)을 수립하고, 입수 시스템, 검색 플랫폼, 디스커버리 툴(전시, 지도, 타임라인), 그리고 통합목록 API 등을 개발하여 오픈소스로 공개했다. DPLA는 메사추세츠주에 도서관으로 등록된 비영리단체(501(c)(3))이며 의장과 이사회, 내외부의 여러 위원회, 그리고 13명의 상근직원으로 구성된 거대 프로젝트이다. DPLA의 조직 형태는 기획 단계에서 운영된 거버넌스 워킹 그룹(Governance Workstream)을 통해 수립되었다. DPLA의 플랫폼과 오픈소스 프로젝트는 4명의 상근 개발자로 구성된 DPLA 테크놀로지 팀이 주도하며 기술위원회의 자문을 받는다.

DPLA 프로젝트는 하버드대학교의 버크만 센터로부터 출발하여 슬로언 재단, NEH, 아카디아기금(Arcadia Fund) 등 공공과 민간의 다수 기금으로 시작되었다. DPLA 프로젝트의 기획 단계인 2011년부터 2015년까지 전체 프로젝트에 제공된 자금은 약 153억 원이며, 이 중 DPLA 플랫폼과 디스커버리 포털의 개발에는 약 50억 원이 쓰였다. 개발비에는 컬렉션 구매와 콘텐츠 개발, 디지털화, 저작권 컨설팅 비용, SNS 운영 등 마케팅과 홍보금액도 포함되어 있다. 개발기간은 2년이다. 2011년에 기획이 시작되어 2013년 4월에 포털 및 오픈플랫폼의 베타버전이 공개되며 240만 건의 기록 목록이 제공되었다. DPLA 플랫폼의 아키텍처는 모듈화를 기반으로 하며 루비(Ruby) 기반의 루비온레일스(Ruby-On-Rails) 프레임워크가 적용되었다.

DPLA 프로젝트는 오픈소스와 오픈데이터 전략, 그리고 효과적인 거버넌스 체계 측면에서 큰 시사점을 준다. 우선 DPLA는 오픈소스가 아니었다면 존재할 수 없었다. 허브로부터 다양한 형식의 데이터를 입수하기 위해 이용된 Akara 프레임워크와 파이썬 프로그래밍 언어, 오픈소스 데이터베이스인 CouchDB, DPLA 플랫폼의 데이터를 API로 내보내기 위해 이용된 루비온레일스(Ruby on Rails)와 Elasticsearch 검색엔진, DPLA 포털의 인터페이스에 이용된 루비온레일스와 PostgreSQL 데이터베이스, 그리고 다양한 자

바스크립트 라이브러리, DPLA 전시 페이지에 이용된 오메카와 이미지 뷰어인 OpenSeadragon 자바스크립트 라이브러리 등 타 오픈소스 프로젝트의 소스코드를 프로젝트 전반에 활용하고 있다. 그리고 직접 개발한 DPLA 입수 코드와 플랫폼, 포털, 전시, 데이터 등을 모두 오픈소스와 오픈데이터로 공개하여 누구나 이용하도록 하였다. 또한 DPLA는 명확한 전략 수립과 지속적인 펀딩을 위해 프로젝트 기획 단계부터 분과별 워킹그룹을 운영했다. 이 워킹그룹을 통해 개발된 매트릭스 도구는 치밀한 전략설계를 가능하도록 했다.

DPLA의 비즈니스 모델은 커뮤니티가 주도하는 개발과 허브 모델이라 할 수 있다. DPLA 플랫폼의 경우 스탠포드대학교 도서관과 하버드대학교 버크만센터의 개발자, 전문 오픈소스 개발자 등 커뮤니티 내의 여러 참여자가 소스코드를 기여하고 있다. DPLA의 허브 모델은 효과적으로 미국 전역의 컬렉션을 수집하고 제공함으로써 컬렉션 제공 기관에게 인센티브를 제공한다. 허브 모델은 콘텐츠 허브와 서비스 허브라는 거점기관을 통해 수집과 서비스 전략을 확산시키는 방법이다. 20만 건 이상의 디지털 객체와 메타데이터를 제공하는 대규모 기관은 콘텐츠 허브로 등록되어 파트너 컬렉션으로 홍보된다. 여러 협력기관을 가진 거점기관은 서비스 허브로 지정되어 해당 지역의 전문서비스, 디지털화, 호스팅 등의 서비스를 제공하며 메타데이터를 수집하여 DPLA에 전달한다. DPLA에 소장 컬렉션을 제공하는 기관에게는 구매비용을 지급하거나 쉽게 제출하기 위한 다양한 기술적 기반과 가이드를 제공한다. 무엇보다 다수의 이용자에게 콘텐츠를 노출하고 홍보하는 효과는 기록소장기관들이 컬렉션을 DPLA에 제공하는 주요한 인센티브가 된다. 또한 DPLA 팀은 DPLA의 문화적 가치를 적극적으로 홍보하고 로비하여 지지기반을 형성함으로써 여러 기금으로부터 지속적인 펀딩을 제공받는다.

DPLA 프로젝트의 커뮤니티는 미국 전역의 1,400여 개 도서관, 박물관, 아카이브, 학술단체에 종사하는 사서, 큐레이터, 아키비스트, 연구자, 오픈

소스 개발자 등으로 구성된다. 오메카를 개발한 RRCHNM은 2016년 11월에 DPLA의 메타데이터 프로파일과 LOD 발행기능을 내장한 오메카 에스(Omeka S)를 오픈소스로 공개하여 미국의 문화유산 커뮤니티들이 보다 쉽게 DPLA에 컬렉션을 제공하도록 지원하고 있다.

DPLA의 깃허브 사이트에는 49종의 어플리케이션과 데이터모델 등이 공개되어 있다. DPLA 플랫폼과 포털, 입수 툴 등 직접 개발한 어플리케이션은 커뮤니티 기반 개발을 위해 AGPLv3 라이선스가 적용되었다. 전시(Exhibition) 등 타 오픈소스 프로젝트의 파생 작업물은 기존 라이선스를 그대로 적용하였으며, DPLA 데이터모델은 제한없는 이용과 확산을 위해 MIT 라이선스가 적용되었다. DPLA 플랫폼의 현재 버전은 v3.2.2이며 포털은 v21.6.10, 입수 툴의 현재 버전은 v33.10.2 이다. DPLA가 개발한 대부분의 오픈소스 프로젝트들은 한 해에 수 차례 이상의 잦은 배포가 이루어진다.

## 5) veraPDF : 국제협력체계 주도 개발

veraPDF는 ISO 19005(PDF/A)의 모든 파트와 적합성 수준(conformance level)을 검증하기 위한 툴이다(veraPDF 2015). 프리포마(PREFORMA, Preservation FORMats for culture information/e-archives) 프로젝트에 참여한 유럽연합의 여러 기록물 소장기관들은 국제적으로 통용되는 문서보존포맷인 PDF/A가 인코딩 소프트웨어에 따라 제각각으로 변환되는 문제를 해결하기 위해 오픈소스 검증 툴인 veraPDF 개발 프로젝트를 시작했다. 이 프로젝트는 veraPDF 컨소시엄(verapdf.org)이 주도하고 있다. 컨소시엄은 오픈보존재단(OPF, Open Preservation Foundation)과 PDF협회(PDF Association)로 구성되어 있다. 소프트웨어 개발에는 듀얼랩(duaLab), 킵 솔루션(KEEPSOLUTIONS) 등 상용 소프트웨어 개발업체가 참여하고, 디지털보존연합(DPC, Digital Preservation Coalition)이 후원하고 있다.

veraPDF는 PREFORMA 프로젝트의 1단계 산출물로 개발되었다. 산출물

에는 오픈소스 소프트웨어 뿐만 아니라 상세한 유즈케이스와 기능 명세서가 포함되어 있다. PREFORMA는 유럽연합의 14개 기관이 참여하고 있는 디지털 보존 커뮤니티이자 프로젝트이다. 이 프로젝트는 개발자, 표준기관, 문화유산 기관 등 유럽연합의 광범위한 디지털 보존 커뮤니티를 대상으로 기관들의 전문지식과 소프트웨어 공급업체의 개발 및 판매 기술을 연결하는 것을 목표로 한다. 무엇보다 디지털 파일을 다루는 오픈소스 적합성 도구를 개발하여 건강한 오픈소스 생태계를 구축하고자 한다. veraPDF는 PREFORMA 프로젝트로부터 개발자금을 지원받았다. 개발 비용은 PREFORMA의 전체 펀딩 금액인 47만 유로의 일부가 이용되었다. PREFORMA에서 총 3개의 오픈소스 소프트웨어를 결과물로 만들어 냈으므로, 그 중 하나인 veraPDF에는 최대 20억 원 정도의 비용이 투입되었으리라 예상된다. 개발은 총 네 단계로 진행되었다. 2015년 3월부터 10월까지 초기 프로토타입을 만들어 첫 번째 릴리즈를 공개했고, 이후 4개월 동안 재설계했다. 2016년 3월부터 12월까지 10개월 동안 두 번째 프로토타입을 만들어 2017년 12월 20일에 후보 릴리즈(RC) v1.0를 공개했다. 이 버전에서는 아카이브매터카, DSpace, JHOVE 등의 보존 소프트웨어에서도 veraPDF의 PDF/A 검증 기능을 이용할 수 있도록 기능을 통합하였다. 2017년부터는 6개월 간의 최종 테스트 단계에 접어들어 완성도를 높이고 있다. veraPDF의 아키텍처 역시 모듈화되어 있으며 자바(Java) 기반의 프레임워크인 스트럿츠 2(Struts 2)가 적용되었다.

veraPDF의 가치는 공공기관과 개발업체의 윈윈전략, 그리고 이를 통한 장기적인 비즈니스모델을 수립한 점이라 할 수 있다. veraPDF의 커뮤니티, 즉 PREFORMA의 커뮤니티에는 도서관, 아카이브 등 문화유산기관 뿐만 아니라 개발업체와 표준기관, 솔루션 구매자 등이 포함되어 있다. 이는 PCP(Pre-Commercial Procurement)와 PPI(Product Procurement of Innovation)로 이어지는 유럽연합의 혁신공공조달 정책에 따른 것이다. 유럽연합은 이 정책을 통해 회원국들의 공공분야 기술적 문제를 공동으로 해결하려 한다. FP7-ICT 프로그

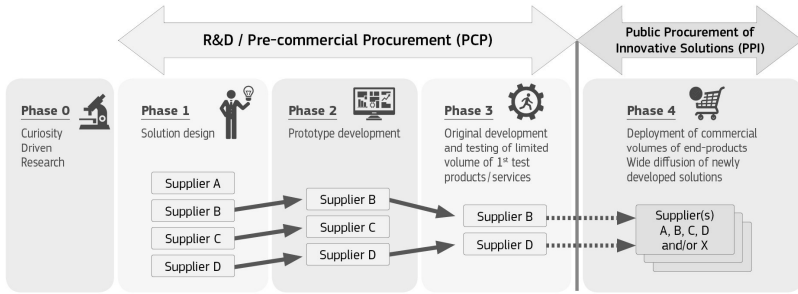
램은 개발업체의 오픈소스 소프트웨어 개발을 초기부터 지원하고 공공조달 형태로 판매를 보장하며, 나아가 공공 표준이 되도록 장기간 지원하고 있다. PREFORMA 프로젝트는 유럽연합과 FP7-ICT 프로그램의 규칙에 따라 공동설계와 프로토타이핑, 테스트 등의 정해진 단계를 거치며, 공동 요구사항 도출 및 업체와의 합의를 조정하고 라이선스의 법적 이슈를 검토하기 위해 FP7-ICT 프로그램의 코디네이터가 배정된다. veraPDF에는 소프트웨어 개발의 기간과 투입공수 등을 정확히 견적하기 위해 코코모 모델(COCOMO, Constructive Cost Model)<sup>11)</sup>이 적용되었다.

FP7-ICT 프로그램은 프로젝트 결과물이 커뮤니티 내에서 자유롭게 재사용되고 개선될 수 있도록 하기 위해 GPLv3 등의 강력한 오픈소스 라이선스를 적용하고 있다. 그 외에 PREFORMA의 도큐멘테이션과 기능명세서 등의 디지털 자원은 크리에이티브 커먼스(CC-BY v4.0) 라이선스와 오픈파일포맷(SOU 2009:86)으로 공개된다. veraPDF의 오픈소스 소프트웨어 라이선스는 GPLv3와 MPLv2(Mozilla Public License)가 적용되었다. 이는 기존 오픈소스 프로젝트의 소스코드를 재사용하여 발전시키고, 다양한 기업들의 소프트웨어에서도 veraPDF가 이용될 수 있도록 하기 위해서이다. MPL 라이선스는 주로 기업들이 주도하는 오픈소스 프로젝트에 사용된다. veraPDF 검증모델과 테스트도구 등의 25개 소스코드와 도큐멘테이션은 홈페이지와 깃허브(<https://github.com/veraPDF>)에 공개되어 있다. veraPDF 라이브러리의 현재 버전은 v1.2이다.

---

11) 코코모 모델(COCOMO, COConstructive COst MOdel)은 1981년 Boehm에 의해 개발된 소프트웨어 비용 추정 모델로 소프트웨어 시스템의 코드 규모(size)를 비용 추정의 근간으로 하여 개발에 필요한 공수(effort)를 산출하기 위해 사용된다(조시호 외, 2010). 코코모 모델은 대기업 및 정부 소프트웨어 프로젝트에 가장 적합하며, 국내에서 주로 사용되는 기능점수모델(Function Points Model)은 모든 규모, 특히 데스크탑 OS 기반의 플랫폼 개발에 적합하다(출처: ProjectCodeMeter, [http://www.projectcodemeter.com/cost\\_estimation/help/GL\\_cocomodiff.htm](http://www.projectcodemeter.com/cost_estimation/help/GL_cocomodiff.htm)).

(그림 2) 유럽연합의 PCP와 PPI 프로세스



\* : <http://www.smart-pcp.eu/pre-commercial-procurement>

## 6) DPSP : 내셔널 아카이브 주도 개발

DPSP(Digital Preservation Software Platform)는 4종의 오픈소스 툴로 구성된 디지털 보존 소프트웨어 플랫폼이다. 호주국립기록보존소(NAA)에서 본 디지털 기록물의 장기보존과 접근을 목표로 개발하여 오픈소스로 배포하고 있다(Cunliffe 2011). 2000년 호주국립기록보존소는 디지털 보존 방법론과 디지털 아카이브 구축을 위한 보존 소프트웨어 연구를 시작했고 2003년 디지털 객체의 보존포맷 변환기인 제나(Xena)를 시작으로 워크플로우 툴인 DPR(Digital Preservation Recorder), 데이터 무결성 검증도구인 체크섬 체커(Checksum Checker), 그리고 이관 툴인 매니페스트 메이커(Manifest Maker) 등 4종의 오픈소스 소프트웨어를 개발하여 공개했다. DPSP 프로젝트는 호주국립기록보존소의 디지털 보존 팀(Digital Preservation team)이 주도하고 있으며 각 소프트웨어는 독자적인 커뮤니티를 구축하고 있다.

DPSP를 구성하는 4종의 오픈소스 프로젝트는 호주국립기록보존소의 주도로 단기간에 집중적으로 개발되었다. 이 프로젝트의 전체 예산은 약 2백만 AUD이며 2001년부터 2003년까지 30개월 동안 진행되었다(Wilson, A. n.d.). Xena의 경우 아카이브의 씹크탱크가 12개월 동안 프로토타입을 만



들어 2013년에 첫 트라이얼 버전을 공개하였다. DPSP 프로젝트는 호주국립 기록보존소의 디지털 보존 전략으로부터 시작되었다. 개발 프로젝트 역시 이에 따라 OAIS 프레임워크를 따를 것, 오픈소스 솔루션을 제공할 것, XML 기반 데이터 형식에 기반할 것 등의 세 가지 원칙을 정하고 있다. 프로젝트 팀은 2013년 4종의 오픈소스 소프트웨어의 초기 버전을 공개하고 현재까지 지속적으로 업데이트를 수행하고 있다. 소스코드의 방향과 기능에 대한 거버넌스는 호주국립기록보존소가 주도하지만 10년 이상 지속되며 성숙해진 이용자와 개발자 커뮤니티를 통해 소스코드 기여가 이루어지고 있다. 최근 개발된 오픈소스 소프트웨어들은 MVC를 지원하는 어플리케이션 프레임워크를 이용하여 소프트웨어의 구조를 모듈화하고 있으나, veraPDF는 프레임워크 적용이나 모듈화 설계 여부를 확인할 수 없었다.

DPSP 프로젝트의 가치는 OAIS 기반의 디지털 보존 솔루션에 대한 초기 모델을 제시한 것과 지속적인 소스코드 재사용이라 할 수 있다. DPSP를 구성하는 오픈소스 소프트웨어 4종은 격리(Quarantine), 보존(Preservation), 저장(Storage)라는 3개의 컴포넌트로 구분된다. 이는 OAIS 프레임워크에 제시된 디지털 객체의 입수(Ingest), 보존계획(Preservation Planning), 저장(Storage) 개념을 따른 것이며, 이 중 격리검사, 이관, 저장 등의 핵심적인 프로세스를 개별적인 오픈소스 소프트웨어로 구현한 것이다. 이는 아카이브매티카 등 OAIS의 전반적 기능을 구현한 솔루션의 프로토타입과 같은 역할을 했다. DPSP의 개별 프로젝트는 전 세계 국공립과 민간 아카이브, 도서관 등 디지털 보존 분야의 이용자와 개발자로 커뮤니티가 확장되었다.

호주국립기록보존소는 DPSP 4종에 GPLv3 라이선스를 적용하고 있다. 강력한 카피레프트 라이선스를 통해 커뮤니티를 지속시키기 위한 것이라 볼 수 있다. DPSP 웹사이트 역시 GPLv3 라이선스로 공개하고 XHTML과 CSS 등의 웹 표준을 따르고 있다. DPSP 로고 등의 아트웍은 보다 자유롭게 이용될 수 있도록 LGPL(Lesser GPL) 라이선스를 적용하고 있다. DPSP의 제품을 상업용 소프트웨어에 이용하고 로고를 붙일 수 있지만, 소스코드는 GPLv3의

의무사항에 따라 공개해야 한다. 4종의 소프트웨어 소스코드와 도큐멘테이션은 호주국립기록보존소의 소스포지 페이지(<http://dpsp.sourceforge.net/>)에 공개되어 있다. 이용자의 편의를 위해 4종의 툴 외에 오픈소스 오피스 도구인 리브레오피스(Libreoffice), 포스트그레스큐엘(PostgreSQL) 데이터베이스, 바이러스 검사기(ClamAV) 등의 오픈소스 소프트웨어가 포함된 설치버전을 제공한다. 인스톨러가 포함된 DPSP의 최신버전은 2013년 공개된 v3.0이다.

#### 4. 기록시스템 오픈소스화의 핵심 성공요소

4장에서는 앞에서 살펴본 선행연구와 오픈소스 프로젝트의 개발 사례를 바탕으로 국내 환경에 적합한 기록시스템의 오픈소스화 전략과 프로젝트 성공을 위한 핵심 요소를 제안하였다. 오픈소스화를 위한 기술, 제도, 전략, 거버넌스 등 네 가지 측면으로 정리하였다.

##### 1) 마이크로서비스 아키텍처와 MVC 패턴

기록시스템의 오픈소스화를 위한 가장 큰 과제는 기술적 기반을 마련하는 것이다. 그 중 오픈소스에 적합한 소프트웨어의 구조나 개발방식은 가장 중요하다. 구체적으로는 모듈화를 가능하게 하는 마이크로서비스 아키텍처, 그리고 소프트웨어의 가독성과 재사용성을 보장하기 위한 모델-뷰-컨트롤러(MVC, Model, View, Controller, 이하 MVC)<sup>12)</sup> 패턴의 설계 등이다.

---

12) 모델-뷰-컨트롤러(Model-View-Controller, MVC)는 소프트웨어 공학에서 사용되는 소프트웨어 디자인 패턴이다. 이 패턴을 성공적으로 사용하면, 사용자 인터페이스로부터 비즈니스 로직을 분리하여 애플리케이션의 시각적 요소나 그 이면에서 실행되는 비즈니스 로직을 서로 영향 없이 쉽게 고칠 수 있는 애플리케이션을 만들 수 있다. 최근에는 MVP와 MVVM 등 Model과 View의 의존성을 다른 방식으로 제어하는 패턴 또한 등장했다. 이러한 패턴들은 코드의 재사용성을 높이고 중복을 방지하기 위한 것이다. 출처: 위키백과. Retrieved <https://ko.wikipedia.org/wiki/Model-view-controller>.

우선 첫째, 마이크로서비스 아키텍처가 유용하다. 소프트웨어 공학은 분석, 설계, 구현, 테스트의 점진적인 개발방식을 취하는 폭포수모델부터 스파이럴, 객체지향, 컴포넌트 기반 개발(CBD), 애자일(Agile)<sup>13)</sup> 등으로 발전되어 왔다. 정보시스템의 복잡도와 규모가 커짐에 따라 신속한 구축과 변경, 그리고 확장이 가능한 시스템을 지향해 왔다고 볼 수 있다. 오픈소스를 이용한 개발 방식은 컴포넌트 기반 개발방식을 취하고 있다. 특히 최근에는 마이크로서비스 방식이 각광받고 있다.

마이크로서비스는 컴포넌트 기반 개발의 발전된 형태이다. 하나를 바꾸려면 전체를 바꿔야 하는 모놀리스(Monolith) 시스템의 비효율성과 서비스 지향 아키텍처(이하 SOA)의 고비용 문제를 해결하기 위해 만들어졌다. 마이크로서비스란 어플리케이션 구성요소를 특정 목적별로 쪼개 뒤 독립적으로 작동하도록 만들고, 구성요소를 조합해 완성된 애플리케이션으로 조립하는 아키텍처를 말한다. SOA의 핵심개념인 ESB(Enterprise Service Bus)는 외부의 온갖 언어로 만들어진 시스템과 서비스들을 중재하느라 복잡도가 증가할 수밖에 없었다. 마이크로서비스는 보다 작고 자율적으로 협업하는 서비스를 목표로 설정함으로써 이를 발전시키고 있다(Newman 2015). 마이크로서비스의 핵심인 컨테이너 가상화 기술과 REST API와 같은 경량 웹 표준 인터페이스의 발전은 마이크로서비스의 기술적 기반을 제공했다. 컨테이너 기술의 대명사격인 도커(Docker)<sup>14)</sup> 등 빠르게 발전하는 오픈소스 컨

---

13) 애자일 소프트웨어 개발(Agile software development) 혹은 애자일 개발 프로세스는 소프트웨어 엔지니어링에 대한 개념적인 일개로, 프로젝트의 생명주기동안 반복적인 개발을 촉진한다. 최근에는 소프트웨어 엔지니어링 뿐 아니라 다양한 전문 분야에서 실용주의적 사고를 가진 사람들이 애자일 방법론을 적용하려는 시도를 하고 있다(출처: 위키백과, Retrieved [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)).

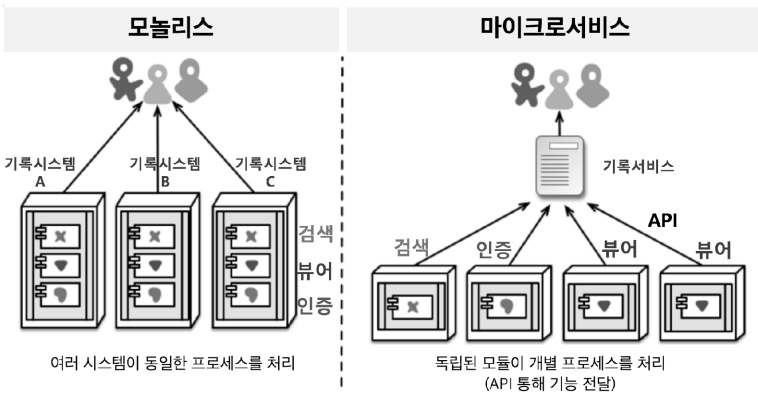
14) 도커(Docker)는 리눅스의 응용 프로그램들을 소프트웨어 컨테이너 안에 배치시키는 일을 자동화하는 오픈 소스 프로젝트이다. 도커 컨테이너는 일종의 소프트웨어를 소프트웨어의 실행에 필요한 모든 것을 포함하는 완전한 파일 시스템 안에 감싼다. 여기에는 코드, 런타임, 시스템 도구, 시스템 라이브러리 등 서버에 설치되는 무엇이든 아우른다. 이는 실행 중인 환경에 관계 없이 언제나 동일하게 실행될 것을 보증한다(출처: 위키백과, Retrieved [https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))).

테이너 기술로 인해 수 년 안에 마이크로서비스의 효율성 또한 비례하여 증가하리라 예상된다(Microsoft 2017). SOA와 구별되는 마이크로서비스의 핵심적인 특징은 컴포넌트 간의 경계가 명확하여 독립적이며 자율적인 서비스가 가능하다는 것이다. 마이크로서비스로 구현된 컴포넌트의 서비스 경계는 비즈니스 경계와 일치한다. 즉 디지털 보존 업무를 바이러스 체크, 파일포맷 식별 등 완전히 독립적으로 작동할 수 있는 최소한의 단위 프로세스로 구분한 뒤 각 프로세스를 지원하는 개별적인 소프트웨어를 하나의 마이크로서비스 컴포넌트로 설정하여 경계를 일치시키는 것이다. 이러한 명확한 경계 구분은 마이크로서비스가 추구하는 작고 일 잘하는 시스템을 만들기 위한 핵심요소이다. 또한 마이크로서비스 아키텍처는 이용자의 참여를 쉽게 하고 소스코드 재사용의 효과를 극대화한다. 예를 들어 아카이브메타카에서는 OAIS의 입수 프로세스를 구현하기 위해 체크섬 확인(Xena), 바이러스 체크(ClamAV), 파일포맷 식별(DROID), 메타데이터 추출(EasyExtract), 식별자 부여(UUID) 등으로 기능을 구분하고 각 기능을 구현하기 위해 기존 오픈소스 프로젝트의 소스코드를 재사용하고 있다. 이 방식대로라면 현용 단계의 기록관리시스템과 영구기록물관리기관의 아카이브 시스템에 공통으로 필요한 기능을 추출하여 모듈화하고 이를 필요로 하는 기관에서 가져다 쓸 수 있게 된다. 중복 개발의 비효율성을 개선하고, 기관별로 필요한 기능만을 선별하여 사용함으로써 시스템의 활용도를 높일 수 있는 구조인 셈이다. 이는 또한 기록시스템의 기능요건을 모듈화하여 제시한 모렉2010(MoReq, Modular Requirements for Records Systems)의 개념을 구현하는 이상적인 방식이기도 하다. 오픈소스 커뮤니티의 측면에서 기능별로 분할된 소프트웨어 구조는 전 세계의 개발자들이 자신이 원하는 모듈의 개발에 참여하도록 동기를 부여하여 소스코드를 개선하도록 한다.

마이크로서비스를 성공적으로 운영하기 위해서는 데브옵스 등의 애자일 프랙티스가 함께 운영되어야 한다. 애자일 방식의 부상은 소프트웨어의 개발이 계획에 기반한 접근 방법에서 끊임없이 반복적으로 개선하는 방식으

로 이동하고 있음을 의미한다. 3장에서 살펴본 대부분의 오픈소스 소프트웨어들은 이용자들의 피드백을 바탕으로 적어도 연간 수 차례의 수정된 버전을 배포하고 있다. ICA-AtoM의 경우 개발의 우선 순위가 이용자들의 피드백에 따라 수시로 조정된다. 애자일 소프트웨어 개발 선언문에는 “공정과 도구보다 개인과 상호작용을, 포괄적인 문서보다 작동하는 소프트웨어를, 계약 협상보다 고객과의 협력을, 계획을 따르기보다 변화에 대응하기를 가치 있게 여긴다”고 명시하고 있다(Beck 2001). 마이크로서비스는 이러한 애자일의 정신을 가능하게 해 주는 민첩성을 제공한다.

〈그림 3〉 모놀리스와 마이크로서비스 아키텍처 비교



\* : <https://martinfowler.com/articles/microservices.html>

셋째, MVC 설계 패턴이 필요하다. MVC의 목표는 사용자 인터페이스와 비즈니스 로직을 서로 영향 없이 고칠 수 있도록 하는 것이다. MVC의 모델 (Model)은 데이터 구조를 표현하고, 뷰(View)는 사용자에게 보여질 부분을 표현하며, 컨트롤러(Controller)는 모델과 뷰 사이에서 동작한다. MVC는 1990년대부터 사용된 설계 패턴이지만 국내에서 이 패턴을 적용한 사례는 많지 않다. 오픈소스 프로젝트에서는 소스코드의 이해와 재사용을 위해 제

계적인 소스코드 건축방식이 중요해진다. 3장에서 조사한 모든 오픈소스 프로젝트들이 MVC 방식으로 설계된 것을 보면 그 중요성을 알 수 있다.

## 2) 공공기록관리 법제의 정비

공공기록물관리법, 기타 법령과 소프트웨어 개발 계약으로 구분하여 제도 측면의 고려사항을 정리하였다. 우선 첫째, 공공기록물 관리에 관한 법률에 따른 고려사항이다. 현재 표준기록관리시스템 수요기관은 825개 정도이다(이소연 2015). 공공기록물관리법에 따라 자체 아카이브를 설치할 수 있는 기관은 국가기록원과 대통령기록관 외에도 총 21개 기관들이 있다(안대진 외 2016). 또한 동법 시행령 30조 2항 1호에 따라 보존기간 30년 이상인 기록물이라 할지라도 사료적 가치가 높지 않은 기록물은 영구기록물관리기관에 이관하지 않고 기록관이나 특수기록관에 보존할 수 있다. 국회기록보존소가 기존 기록관리시스템에 영구기록물관리 기능을 도입한 사례에서 보듯이 특수기록관 또한 아카이브 성격의 시스템을 필요로 하게 되었다(김장환 외 2015). 이러한 수요에 대해 국가기록원은 2016년 상반기에 표준기록관리시스템과 같은 방식의 개발과 배포를 염두에 두고 “표준영구기록관리시스템(AMS) 구축” 계획을 세웠던 바 있다. 그러나, 표준기록관리시스템 사례를 통해 반성이 필요하며 이제는 비용효율성과 지속성을 담보할 수 있는 방식, 즉 오픈소스를 적극적으로 활용하여 시스템을 개발하고 지속적인 발전을 도모해야 할 때이다.

기록시스템의 오픈소스화를 위해서는 우선 기록관과 영구기록관별로 필수적인 시스템 요건을 재정립해야 한다. 공통 기능에 해당하는 요건을 우선적으로 오픈소스화하고 기관별 상황에 따라 특수한 요건에 대한 별도의 오픈소스화 또는 상용 솔루션 사용을 고려해야 한다. 영구기록물 관리를 위한 소프트웨어의 경우 OASIS 참조모델의 기능요건을 필수요건으로 정하여 충실히 구현하는 것이 바람직하다. 3장에서 살펴본 아카이브메타카,

veraPDF, DPSP 등의 오픈소스 프로젝트에서 보듯이 모든 디지털 아카이브 소프트웨어들은 OAIIS 요건 충족(Compatible)을 목표로 개발되고 있다. 오픈소스화의 중요한 원칙은 소스코드를 가급적 그대로 재활용하는 것이다. 소스코드의 재활용 비율이 높을 수록 투입되는 개발 자원이 줄어들고 신뢰도가 높아지기 때문이다. 마이크로서비스 방식을 전제로 할 경우 밴쿠버시 아카이브의 사례처럼 기록관과 아카이브의 업무 프로세스를 독립적인 가장 작은 단위로 구분하여 마이크로서비스의 오픈소스 컴포넌트 단위와 일치시키는 것이 중요하다.

둘째, 기타 관련법령과 소프트웨어 개발 계약 측면의 고려사항이다. 공공기관이 오픈소스 프로젝트를 시작하거나 개발업체가 오픈소스 소프트웨어 개발 계약을 추진할 때 기존의 여러 법령이나 제도와 부딪히게 된다. 예산타당성 심사를 위한 개발비용 산정, 하드웨어 및 소프트웨어, 보안성 검토, 폐쇄망과 인터넷망의 구분으로 인한 제약, 전자정부 프레임워크, 공공누리 정책 등이다.

우선 개발비용 산출 양식의 정비가 필요하다. 우리나라 공공기관에서는 기능점수방식(Function Point)으로 소프트웨어 개발비용을 산정한다. 이 방식은 신규로 개발할 기능을 점수로 환산하여 응용 소프트웨어의 규모를 산정한 후 기능점수별 단가 및 보정계수, 이윤 등을 적용하여 개발비를 계산한다. 서울특별시 기준에 따라 오픈소스 소프트웨어 개발비를 계산하면 실제보다 훨씬 큰 금액이 산출된다(서울특별시 2016b). 이를 해결하기 위해서는 기능점수 도출 양식에 오픈소스를 재사용하는 경우에 대한 항목을 추가해야 한다. 신규개발비 또는 재개발비 산정양식에 “오픈소스 재사용” 항목을 추가하고 세부 항목으로 “재사용 유형(전체사용, 라이브러리만 사용 등)”과 “재사용율(%)”을 넣는 것이 바람직하다.

다음으로는 하드웨어 및 소프트웨어 측면의 이슈이다. 통합전산센터의 인프라를 이용하는 중앙부처의 경우 클라우드 인프라가 정비되어 있어 오픈소스 소프트웨어를 구동하기 위한 리눅스 서버나 Apache 등 소프트웨어

지원에 큰 문제가 없다. 반면 나머지 기관에서는 유닉스나 윈도우 기반의 서버를 운용하는 경우가 많다. 따라서 호환성 문제로 리눅스 기반의 하드웨어 도입부터 쉽지 않다. 또한 아주 단순한 기능을 수행하는 시스템인 경우에도 무리한 수준의 보안 및 이중화 조치를 요구받는다. 예를 들어 보안성 검토를 위한 시큐어 코딩, 그리고 이중화나 백업을 위해 여러 대의 하드웨어를 추가 구매하라는 요청 등이다. 전자정부법 시행령 제71조제1항에 따른 정보시스템 감리의 대상은 1억 원 미만의 소규모 사업이거나 행정업무, 민원업무 처리용이거나, 여러 중앙행정기관이 공동으로 구축하는 경우이다. 1억 원 이하의 오픈소스 프로젝트에 시큐어 코딩의 적용을 검토하라는 요청은 오픈소스 소프트웨어가 보안에 취약할 것이라는 인식 때문에 제기된 것으로 판단된다. 법령대로 시스템의 규모에 따라 이러한 조치들이 선별적으로 취해져야 할 것이다.

전자정부 표준프레임워크 측면에서는 다양한 프로그래밍 언어와 프레임워크를 지원해야 한다. 전자정부 프레임워크는 자바(Java) 기반의 스프링(Spring) 프레임워크로 되어 있다. 따라서 PHP, Python, Ruby 등의 언어로 개발된 ICA-AtoM, 오메카 등 오픈소스 아카이브 소프트웨어들을 여기에 맞추려면 상당한 추가 작업이 필요하다. 현행 체계는 기록시스템 뿐만 아니라 다양한 오픈소스와 상용 소프트웨어와의 호환성 문제를 안고 있다. 웹 기반으로 변화되어 가는 어플리케이션 환경에 지속적으로 대응해야 할 것이다. 장기적으로는 기록시스템의 필수요건을 기록관리 기관의 공통업무 컴포넌트로 등록하여 클라우드 환경에서 자유롭게 구성하여 이용할 수 있도록 정부통합전산센터나 서울특별시 등 지방자치단체의 데이터센터와 호환성 측면의 개선이 이루어져야 한다.

오픈소스 소프트웨어를 납품하며 공공누리 라이선스(KOGL, Korea Open Government License)를 적용하라는 요구를 받기도 한다. 공공누리는 문화체육관광부에서 공공저작물 배포를 위해 개발한 라이선스로 OSI에서 인증한 오픈소스 라이선스에 해당하지 않는다. 따라서 납품되는 소프트웨어는



본래의 오픈소스 라이선스를 따라야 한다.

### 3) 저작권과 라이선스 전략

오픈소스 이니셔티브에서 인증한 오픈소스 라이선스의 종류는 현재까지 약 50여 종 이상이다. 크게 제한적 카피레프트인 GPL 계열, 약한 카피레프트인 MPL 계열, 비카피레프트인 BSD 계열로 구분할 수 있다. 이 중 3장에서 살펴본 6종의 오픈소스 프로젝트들은 대부분 강한 카피레프트인 GPLv3와 AGPLv3를 적용하고 있다. veraPDF의 경우만 GPLv3와 MPLv2 라이선스를 병행하고 있다. 6종의 프로젝트를 세 가지 유형의 라이선스 전략으로 세분화할 수 있다. 각 사례는 국가기록원이나 서울특별시가 기록시스템을 오픈소스화하기 위한 다양한 시사점을 제공한다.

첫째, GPLv3 라이선스 기반의 커뮤니티 모델이다. 오메카, DPLA, DPSP 등 3개 프로젝트가 여기에 해당한다. GPLv3 라이선스는 공공기관이나 비영리단체가 주도하는 비즈니스 모델에 적합하다. 소프트웨어의 주요 이용자인 문화유산기관들이 개별적인 커스터마이징을 통해 소프트웨어를 개선하고 결과물의 소스코드를 다시 커뮤니티 내에 기증하도록 하기 때문이다. 개발 과정에 재사용된 오픈소스 역시 GPLv3 라이선스이거나 GPLv3와 호환되는 라이선스인 경우가 많다. 기업들이 주로 적용하는 BSD계열의 라이선스와 비교할 때 GPLv3는 상대적으로 공공성(General Public)을 중시하는 경향이 있으므로 재사용하거나 플러그인으로 추가되는 오픈소스 프로젝트 역시 커뮤니티 내부로부터 시작된 것들이 많다. 또한 이들은 공공과 민간의 다양한 기금을 비교적 손쉽게 지원받아 해당 소프트웨어를 커스터마이징할 수 있는 여력이 있어 상업적 이용가능성을 덜 중시하는 경향이 있다. 따라서 GPLv3 라이선스는 주로 공공의 이익에 봉사하는 사명을 가진 다수의 기관들이 협력적으로 커뮤니티를 지속해 나가는 데 적합한 라이선스라 할 수 있다.

둘째, AGPLv3 라이선스 기반의 커뮤니티 바운티 모델이다. ICA-AtoM과 아카이브매티카가 여기에 해당한다. 커뮤니티 바운티는 커뮤니티 내 기관들의 자발적 기부를 통해 소프트웨어를 지속시키는 방식이다. 이 방식은 ICA와 UNESCO의 국제적 명성을 통해 실현될 수 있었다. ICA-AtoM과 아카이브매티카는 오메카나 DPLA처럼 대규모 커뮤니티를 이루지는 못 했다. ICA-AtoM은 자금 확보 문제로 공개 버전을 배포하기까지 5년이 걸렸다. ICA의 공식 자금지원이 종료되면서부터는 개발업체인 아티팩추얼이 여러 기관의 개별 커스텀 계약을 통해 근근이 소프트웨어를 발전시키고 있다. 이처럼 커뮤니티 기반의 바운티 모델은 영향력 있는 조직이나 기술력을 가진 전문 개발업체가 주도하지 않는 한 쉽게 유지되기 어렵다. 이러한 이유로 ICA-AtoM과 아카이브매티카에는 GPLv3보다 강력한 AGPLv3 라이선스가 적용되었다. AGPL 라이선스는 클라이언트 소프트웨어 뿐만 아니라 서버의 소스코드까지 공개하도록 하는 독특한 조항을 담고 있다.

셋째, 듀얼 라이선스 전략이다. 이는 veraPDF 뿐만 아니라 PREFORMA 프로젝트의 모든 오픈소스 소프트웨어에 적용되는 전략이다. 듀얼 라이선스란 2개의 라이선스 중 이용자가 하나를 선택하도록 하는 방식이다. PREFORMA 프로젝트는 소프트웨어 업체의 오픈소스 소프트웨어 개발을 초기부터 지원하여 공공기관의 구매로 연결해 주는 유럽연합의 공공조달정책(PCP & PPI)로부터 시작되었다. PREFORMA는 이 정책에 따라 FP7-ICT 프로그램의 기금을 받았고, 커뮤니티에는 문화유산기관 뿐만 아니라 여러 개발업체와 표준기관들도 포함되어 있었다. 이러한 다양한 이해관계를 고려하여 엄격한 소스코드 공개 의무가 있는 GPLv3와 상업적 이용이 가능한 MPLv2를 듀얼 라이선스로 적용하였다. 듀얼 라이선스를 적용할 때는 사용 목적에 따라 치밀한 호환성 검토가 요구된다.

위에서 살펴본 것과 같이 커뮤니티 모델, 커뮤니티 기반 바운티 모델, 그리고 듀얼 라이선스 전략은 우리나라 기록 커뮤니티에도 적용할 수 있는 대표적인 오픈소스 라이선스 전략이다. 개발업체들의 참여를 위해서는 듀

열 라이선스 전략이 가장 적합하다. 이러한 관점으로 국가기록원과 서울특별시  
시가 적용할 수 있는 보다 구체적인 라이선스 전략을 이슈별로 정리하였다.

우선 첫째, 적용되는 라이선스별로 업체와의 계약 조건이 달라진다. 아  
웃소싱을 통해 소프트웨어를 개발하고 이를 오픈소스화할 경우 저작권은  
실제 창작자인 해당 업체나 개발자에게 귀속된다. 개발업체가 제작한 소프  
트웨어의 저작권을 위탁기관이 가지려면 계약서를 작성할 때 결과물의 저  
작권은 위탁자에게 귀속된다는 조항을 넣어야 한다. 오픈소스 소프트웨어  
의 경우 저작권은 명시적인 것일 뿐 그리 중요하지 않다. 대부분의 오픈소  
스 라이선스는 이용과 복제, 수정이 자유롭기 때문에 저작권이 누구에게 있  
건 활용하는 데는 문제가 없기 때문이다. 다만 배포할 때의 조건에 따라 특  
징이 생기므로 이에 대한 사항을 계약서에 명시해야 한다. 적용된 오픈소  
스 라이선스 문서를 그대로 인용하는 것이 가장 명확할 것이다.

GPL 계열의 라이선스를 적용할 경우 개발업체는 개발 결과물을 상업적  
으로 이용할 수 있지만 기술을 공개해야 하는 부담을 안게 된다. 따라서 이  
를 보전하기 위한 조치가 필요할 수 있다. 일반적으로 리드 개발사가 해당  
오픈소스의 전문업체로서 교육과 컨설팅, 유지보수 등을 전담하도록 하고  
있다. 다만 이 경우는 도입할 기관이 많을 경우에 가능하다.

veraPDF의 사례처럼 듀얼 라이선스를 적용할 경우에는 업체의 영업비밀  
이나 핵심 기술이 포함된 패키지에 비커퍼레프트 계열이나 독점 라이선스  
를 적용할 수 있다. veraPDF에서는 이러한 용도로 MPLv2를 선택했지만 일  
반적으로는 Apache License 2.0이나 MIT License가 많이 쓰인다. 이 두 라  
이선스는 GPL과 호환되고 2차 저작물의 소스코드 공개의무가 없어 기업들  
의 참여를 유도하기 위해 이용된다. 듀얼 라이선스는 오픈소스 커뮤니티가  
활성화되지 않은 국내 상황에 적합한 전략이다. 기록관리 분야의 많은 소  
프트웨어들이 GPL 라이선스로 개발되었으므로 이를 재사용하고 개발업체  
들의 참여를 촉진하기 위해서는 GPLv3와 Apache License 2.0 또는 GPLv3  
와 MIT License를 듀얼 라이선스로 적용하는 것이 바람직하다. 계약서에는

GPLv3를 적용할 패키지나 컴포넌트, Apache License 2.0을 적용할 패키지나 컴포넌트를 구분하여 명시해야 한다.

둘째, 단계적이고 부분적인 오픈소스화 전략이 효과적이다. 신규시스템을 개발할 경우보다는 이미 개발해 놓은 시스템을 오픈소스화하거나, 개발업체의 기존 솔루션을 재사용하려 할 때 적합한 전략이다. 이러한 경우 클로즈드 소스(closed-source)로 최신버전을 판매하면서 이전 버전만 오픈소스화하는 단계적 접근이 가능하다. 또는 시스템 이용자들에게만 오픈소스로 제공하여 가치를 제공하면서 경쟁사들이 소스코드를 직접 이용하지 못하도록 대안을 강구해 볼 수도 있을 것이다. 흔한 경우는 아니지만 오픈소스화 초기에는 이러한 대안이 필요하다. 영업비밀이나 상표, 특허 등이 포함된 경우 이를 보호하거나 공개하기 위한 별도의 라이선스 적용을 검토한 후 이에 대한 사항을 계약서에 명시해야 한다. 일정 기간 Preservica 등 아주 잘 설계된 해외의 상용 패키지를 도입하여 내외부 역량을 키우는 것 또한 시스템을 처음 도입하는 기관에게 효율적일 것이다. 기록연구사는 소프트웨어에 대한 이해를 높일 수 있고, 개발업체는 SDK(Software Development Kit)를 제공받아 상용패키지의 특성과 구조를 익히며 역량있는 아키텍트를 키울 수 있기 때문이다. 다만 이 경우에는 해당 솔루션의 아키텍처나 프레임워크에 따라 커스터마이징의 제한을 받게 된다.

셋째, 공공자금 지원을 전제로 오픈소스화하는 전략이다. 새로운 기록시스템을 개발하는 경우, 우수한 기술력을 보유한 업체의 솔루션을 오픈소스화하는 조건으로 인센티브를 제공하거나 해당 시스템의 전문업체 자격을 부여할 수 있다. 이 경우 기록관리 전체 업무를 지원하는 통합 솔루션이 아닌 작은 기능을 수행하는 소프트웨어를 지원하는 것도 가능하다. 예를 들어 A 업체의 메타데이터 추출 솔루션, B 업체의 파일포맷 변환 솔루션 등 기록시스템을 구성하는 보다 작은 단위의 소프트웨어를 지원하여 점차적으로 오픈소스 생태계를 구축하기 위한 기반을 마련해야 한다. 기록시스템이 아니라도 국가기록원이나 서울특별시 차원에서 공공기금이 지원되는 사

업의 결과물을 오픈소스로 제출하도록 하여 오픈소스 개발 인력을 점차 확보해 나가야 한다.

#### 4) 협력과 개선을 위한 공동체의 조건

3장의 사례에서 보듯이 오픈소스 프로젝트의 성공을 위해서는 강력한 리더십을 가진 거버넌스 체계와 커뮤니티 구성원들의 협력적인 개발이 필요하다. 기록시스템의 오픈소스 기록시스템 프로젝트의 성공을 위한 커뮤니티 관점의 고려사항을 제안하고자 한다.

우선 첫째, 주관기관의 강한 리더십과 홍보전략이 필요하다. 기록시스템을 오픈소스화한다는 것은 IT 운영방식이 중앙 집중형에서 장기적이고 협력적인 비즈니스 모델로 전환됨을 뜻한다. 따라서 국가기록원과 서울특별시 등 개발의 주체에서 오픈소스 프로젝트의 거버넌스 기구로 변화되어야 한다. 거버넌스 기구는 우선 프로젝트의 지속과 성공을 위한 기반을 마련해야 한다. 내외부 기관을 가리지 않고 공동으로 운영하기 위한 체계, 비용 분담 및 자금 조달방법, 각 참여기관의 역할과 책임의 문서화 등이다. 이 기구 안에는 커뮤니티 내부의 핵심적인 개발자 그룹과 연구자, 행정가, 기록연구사 등이 포함되어야 한다. 오픈소스 프로젝트들은 협력과 의사소통의 어려움으로 인해, 조정 매커니즘이 느슨해질 경우 새로운 프로젝트로 분기(forking)될 위험이 있다. 또한 ICA-AtoM과 AtoM이 분리된 사례에서 보듯이 거버넌스 조직은 그들을 믿고 소프트웨어를 도입한 커뮤니티의 기관들에게 지속적 의무감을 가져야 한다(ICA 2015). 특히 강한 리더십과 홍보를 통해 커뮤니티의 폭넓은 참여를 촉진하고, 프로젝트의 장기 개발 일정을 완수하기 위한 기금 마련 등에 신경써야 한다. 예를 들어 서울특별시가 결재 문서를 깃허브에 공개한 사례처럼 개발 초기부터 선제적으로 오픈소스화 계획을 공표하고 알파 단계의 릴리즈부터 다양한 테스트 그룹을 참여시켜 관심을 집중시킬 수 있을 것이다.

둘째, 커뮤니티를 넓게 봐야 한다. 크게 개발자 그룹과 적극적 이용자 그룹으로 구분할 수 있다. 개발자 그룹에는 기록시스템 개발업체 뿐만 아니라 업무관리시스템, 전자문서시스템 등 생산시스템 관련업체, 도서관리 및 박물관리시스템 개발업체 등이 포함된다. 그 외에도 오픈소스나 MVC 패턴에 능숙한 웹 개발업체, 큐브리드 등의 전문 오픈소스 제품 개발업체, 오픈소스 툴을 개발하고 보급하는 비영리단체 등도 포함해야 한다. 각 기관에 개발인력을 두고 개별 오픈소스 소프트웨어 프로젝트를 진행하는 미국의 도서관이나 아카이브와 달리 국내에는 협력적 개발에 참여할 개발자 그룹이 기관 내에 없는 경우가 많기 때문이다. 또한 빅데이터, 인공지능 등의 기술력을 갖춘 기업들을 참여시켜 프로젝트의 확산을 도모해야 한다. 연구를 위해 인터뷰한 국내 오픈소스 개발업체의 대표는 기록시스템 분야의 사업성이 매우 안 좋으며, 그 이유로 다수의 이용자가 쓸 수 있는 범용의 기능이 아니라는 점을 꼽았다. 또한 이를 극복하기 위해서는 보존 시스템보다는 상대적으로 범용적인 서비스 시스템 쪽에 다양한 최신기술을 접목해서 타 분야로 확장시키는 전략이 유용할 것이라 제안했다. 기록 빅데이터 분석, 인공지능을 활용한 온톨로지 구축 등은 기록관리의 전문성을 타 분야로 확산시키는 좋은 사례가 될 수 있을 것이다. 적극적 이용자 그룹은 기록관리 뿐만 아니라 도서관, 박물관 등 문화유산기관 전체로 봐야 한다. 국내 각급 기관에 배치된 기록물관리전문요원, 사서, 큐레이터, 학예연구사, 연구기관과 학회, 협회, 개인 연구자, 비영리단체의 활동가, 기록대학원의 학생 등이 있을 것이다. 이들은 경험이나 연구를 기반으로 현장의 요구사항을 만들고 개발 결과물에 대한 피드백을 제공하는 역할을 맡게 된다.

셋째, 커뮤니티의 공동작업과 비용분담이 필요하다. 그 동안 국가기록원과 서울기록원은 자체 예산으로 시스템 개발 비용을 마련할 수 있었다. 하지만 3장의 사례에서 보듯이 오픈소스 프로젝트는 새로운 버전의 배포 주기가 빠르고, 장기간 동안 지속적인 자금지원이 필요하다. 이러한 자금을 한 기관이 모두 분담하는 것은 쉽지 않다. 국가기록원이나 서울특별시

ICA-AtoM이나 아카이브메티카의 사례처럼 독자적인 개발 프로젝트를 수행할 여력이 있는 기관들과 접촉하여 인력과 자금을 분담하는 방식을 논의해야 한다. 연구기록물관리가 필요한 국회, 검찰청 등 특수기록관 설치기관, 경상남도 등 지방기록물관리기관 설립을 준비 중인 지방자치단체, 그리고 국립중앙도서관, 국립중앙박물관, 국립민속박물관, 민주화운동기념사업회 등 다양한 국내 문화유산기관들은 공통적인 기능을 수행하면서도 모두 독자적으로 시스템을 구축하고 있다. 비단 소프트웨어 개발이 아니더라도 교육, 홍보, 인프라 제공, 매뉴얼 제작, 번역 등 커뮤니티 내에서 분담할 개발 관련 작업은 상당히 많다. 또한 한 편으로는 공공 분야의 연구자금을 활용하기 위한 방안을 강구해야 한다. 예를 들어 한국연구재단이나 한국학중앙연구원 등 인문학 분야 연구기금, 오픈소스 프로젝트에 투자하고자 하는 관련 기업 및 투자사 등을 통해 활용 가능한 기금을 안정적으로 조달해야 한다. 한국연구재단의 2017년 예산은 4조8천 억 수준이다. 커뮤니티 소속 기관들이 이러한 기금을 이용하여 디지털 인문학 등의 개별 프로젝트를 수행하고 오픈소스로 개발된 결과물 소스코드를 커뮤니티에 기부하는 방식으로 협력할 필요가 있다.

넷째, 중앙집중식 협력도구 구축이 필요하다. 2장에서 검토했듯이 커뮤니티 내 협력을 위해서는 다양한 협력도구와 개발 인프라가 필요하다. 성공적인 오픈소스 프로젝트에는 효과적이고 빠른 의사결정을 지원하는 투표나 메신저 툴, 기술토론을 위한 포럼이나 블로그, 결정사항을 공유하기 위한 메일링시스템, 이를 기록하기 위한 위키와 자동화된 문서화 도구, 소스코드 저장, 버전관리, 호스팅을 위한 깃허브 등의 인프라, 버그와 기능추가 관리를 위한 이슈추적시스템, 프로젝트 테스트를 위한 개발서버, 오픈소스 검증 툴 등이 필수적이다. 그리고 이러한 도구들은 프로젝트 기획 단계에서 거버넌스 조직의 주도 하에 중앙집중식으로 구축되어야 한다. 이들 중 대부분은 오픈소스 또는 무료 서비스로 공개되어 있다. 다양한 오픈소스 프로젝트에 적용된 사례를 조사하여 적합한 툴을 선별해야 할 것이다. 또

한 3장의 사례들로부터 프로젝트의 성공을 위한 정책이나 도구 또한 필요함을 알 수 있다. DPLA의 경우 정확한 개발비 산정을 위한 전략 매트릭스, 프로젝트의 자금과 지지기반을 확보하기 위한 홍보 채널 등을 개발하여 운영했다. veraPDF 프로젝트에서는 공동의 요구사항을 도출하고, 업체와의 합의사항을 조정하며, 오픈소스 라이선스의 법적 이슈를 검토하기 위한 코디네이터가 배정되었다. 이러한 추가 조치들은 내부의 계획과 커뮤니케이션을 강화하고 외부의 이해당사자들을 커뮤니티 안으로 유입시키는 주요한 동기를 제공할 것이다.

## 5. 맺음말

에릭 레이먼드는 오픈소스의 효율성을 성당과 시장 모델로 설명한다. 성당을 건축하듯이 몇 명의 도사 프로그래머가 만드는 방식보다 서로 다른 의견이 난무하는 소란스러운 시장 방식이 훌륭한 소프트웨어를 만드는 데 훨씬 효과적이라는 것이다(Raymond, E. 1999). ‘자아를 내세우지 않는 프로그래밍’의 저자인 제럴드 와인버그는 개발자들이 자신의 코드에 텃세를 부리지 않고 다른 사람으로 하여금 버그를 찾고 개선 가능성을 찾아내도록 하면 극적으로 빠른 개선이 일어난다고 이야기했다. 이렇듯 오픈소스 소프트웨어는 소스코드의 재사용이나 다수가 참여하는 개발의 효율성에 방점을 두며 가치를 입증해 왔다. 이제는 많은 기업들이 오픈소스에 투자하고 있다. 하지만 그럼에도 불구하고 오픈소스에 대한 오해와 우려는 지속되어 왔다. 이러한 문제의 근본 원인은 오픈소스의 철학적 기반, 특히 소유권의 분산 때문이다. 우리는 개방하고 협력하는 방식보다 사유하고 독점하는 방식에 익숙하다. 이러한 익숙함은 사소하게 볼 것이 아니다. 우리는 공공기록물관리법 제정 이후 17년이 지나도록 검색조차 제대로 안 되는 기록시스



템을 방치해 왔다.

이 연구는 국가기록원과 서울기록원이 현행 기록시스템의 문제를 오픈소스로 풀기 위한 방안을 제안했다. 오픈소스화를 위한 절차와 도구, 라이선스 전략, 최근의 개발방법론과 아키텍처, 그리고 프로젝트를 누가 주도하고 어떻게 협력하는지의 관점으로 시사점을 정리했다. 마이크로서비스 아키텍처나 MVC 패턴, 개발도구, 라이선스 검증도구 등 개발업체를 위한 기술적 제안도 일부 포함되었다. 해외의 오픈소스 프로젝트 개발과정을 정리하며 국내 법제도 환경과 개발 생태계의 부재를 과연 극복할 수 있을지 의문들이 수밖에 없었다. 공공기관이나 비영리단체의 오픈소스 프로젝트들은 대부분 투자자본수익률(ROI, Return On Investment)을 고려하고 있지 않기 때문이다. 이는 공공과 민간의 기금으로 지속되는 모델이다. 이는 오픈소스를 활성화하려는 정부의 정책과 맞물려 기금이 집행되어야 가능해진다. 미국의 인문학기금과 민간의 다양한 기금, 유럽연합의 공공조달 정책 등이 좋은 사례이다. 우리는 커뮤니티의 테두리를 최대한 넓게 잡아야 한다. 기록관리 분야와 인접한 도서관, 박물관, 그리고 민간 분야의 단체들로부터 공동체를 확장해야만 가능한 일일 것이다.

마이크로서비스 아키텍처나 MVC 패턴의 디자인은 국내 상황과 개발업체의 역량을 고려할 때 요원한 일이라 생각할 수 있다. 하지만 이미 가속화된 클라우드 컴퓨팅으로의 전환이나 도커 등 오픈소스 컨테이너 기술의 성장은 위기이자 기회가 될 수 있다. 작지만 일 잘하는 시스템을 만들기 위해 기록시스템의 오픈소스화 전략을 진지하게 고민해 볼 시점이다.

### 〈참고문헌〉

- 국가기록원. 2017. 차세대 기록관리 모델 재설계 제안요청서 .  
김장환, 이은별. 2015. 국회기록정보 통합관리시스템 개발 방향에 관한 연구. 한국기록관리학회지 , 15(2), 103-136.

- 박민수, 서은경. 2012. 표준 기록관리시스템에 대한 사용자 만족도 분석. 한국기록관리학회지, 12(1), 189-207.
- 박민영. 2013. 표준기록관리시스템 기능 평가. 기록학연구, 38, 3-35.
- 박종연. 2013. 표준 기록관리시스템의 인수 기능 평가. 기록학연구, 37, 239-271.
- 박태연, 신동희. 2016. 참여형 아카이브 구축을 위한 오픈소스 소프트웨어 평가. 한국기록관리학회지, 16(1), 121-150.
- 방기영, 김진, 황창주, 김용. 2016. 축제기록관리시스템 구축을 위한 공개용 소프트웨어 비교연구. 정보관리학회지, 33(3), 31-61.
- 서울특별시. 2016a. 서울기록원 정보화전략계획수립용역 완료보고서. 검색일: 2017.3.1. <https://drive.google.com/file/d/0Bxyip0OKGsR5NGxiVkZBNmUzQW8/view>.
- 서울특별시. 2016b. 정보화사업 유형별 예산산정 세부 적용기준.
- 설문원, 최이랑, 김슬기. 2014. 오픈소스를 활용한 사건 아카이브 구축에 관한 연구. 한국기록관리학회지, 14(4), 7-36.
- 송위진. 2002. 오픈소스 소프트웨어의 기술혁신 특성: 리뷰. 기술혁신학회지, 5(2), 212-227.
- 송정숙, 허정숙, 이예린. 2014. 오픈소스를 이용한 부산항 사진 아카이브의 구축 방안. 한국기록관리학회지, 14(3), 127-151.
- 심갑용, 유현경, 문상훈, 이운용, 이정현, 김용. 2015. 한류문화콘텐츠의 기록화를 위한 AtoM 활용 방안에 관한 연구. 기록학연구, 43, 5-42.
- 안대진, 김익한. 2015. AtoM 시스템의 구축과 커스터마이징 방법에 관한 연구. 기록학연구, 45, 5-50.
- 안대진, 임진희. 2016. 디지털 아카이브 시스템 구축을 위한 공개 소프트웨어 활용방안 연구. 정보관리학회지, 33(3), 345-370.
- 오진관, 최광훈, 이연희, 최성호, 임진희. 2016. 기록관리기관의 RFID 시스템 기능 개선 연구. 한국기록관리학회지, 16(4), 55-77.
- 육혜인, 김용, 장준갑. 2015. AtoM을 활용한 민간기록물 관리방안. 한국비블리아학회지, 26(2), 79-105.
- 이경남. 2013. 표준 기록관리시스템 검색 기능 평가. 기록학연구, 37, 273-305.
- 이동현, 전보배, 현문수. 2014. OMEKA를 활용한 노동 아카이브 구축. 한국기록관리학회 학술발표논문집, 217-222.
- 이보람, 황진현, 박민영, 김형희, 최동운, 최윤진, 임진희. 2014. 공개 소프트웨어를 이용한 기록시스템 구축가능성 연구. 기록학연구, 39, 193-228.
- 이보람. 2013. 표준기록관리시스템 평가·폐기 기능 평가. 기록학연구, 38, 37-73.
- 이소연. 2015. 표준기록관리시스템의 활용현황 연구. 기록학연구, 43, 71-102.

- 이철남. 2002a. 오픈소스 소프트웨어 지원정책 : 영국의 정책변화와 그 시사점을 중심으로. *정보통신정책*, 14(20), 13-31
- 이철남. 2002b. 공개소프트웨어 활성화정책의 현황과 방향, *정보통신정책*, 15(5).
- 임진희. 2008. 기록관리시스템 기능요건 표준의 실무적 해석. *기록학연구*, 18, 139-178.
- 정보통신산업진흥원. (2014). *공개SW 라이선스 가이드*.
- 정상희. 2013. 표준 기록관리시스템의 '기준관리' 기능 및 이용 평가. *기록학연구*, 37, 189-237.
- 조시호, 이준하, 박수용. 2010. 비용을 고려하고 아키텍처 평가를 지원하는 소프트웨어 아키텍처 비용 추정 기법. *정보과학회논문지: 소프트웨어 및 응용*, 37(2), 95-103.
- 최운진, 최동운, 김형희, 임진희. 2014. 공개 소프트웨어 OMEKA를 이용한 기록 웹 전시 방안 연구. *기록학연구*, (42), 135-183.
- 한국정보화진흥원. 2014. 국가정보화 20년의 성과.
- 현문수, 전보배, 이동현. 2014. 참여형 디지털 아카이브 구축 실행 방안. *기록학연구*, 42, 245-285.
- 현문수. 2013. 표준 기록관리시스템의 전자기록 보존 기능 평가 연구. *한국기록관리학회지*, 13(2), 115-147.
- Androutsellis-Theotokis, S., Spinellis, D., Kechagia, M., & Gousios, G. 2011. Open Source Software: A Survey from 10, 000 Feet. *Foundations and Trends in Technology. Information and Operations Management*, 4(3-4), 187-347.
- Archivematica. n.d. Preserving memory since 2009. Retrieved 2017.3.18. <http://archivematica.org/>
- Beck, K., et al. 2001. Manifesto for agile software development. Retrieved March 30, 2017 <http://agilemanifesto.org/>
- Bradley, K., Lei, J., & Blackall, C. 2007. Towards an Open Source Repository and Preservation System: Recommendations on the Implementation of an Open Source Digital Archival and Preservation System and on Related Software Development. UNESCO.
- Cunliffe, A. 2011. Dissecting the digital preservation software platform.
- DPLA. 2012. Appendix B: Strategic Matrix (XLSX). Retrieved March 26, 2017 from <http://blogs.harvard.edu/dplaalpha/get-involved/events/mar2012fin/>
- DPSP. n.d. Digital Preservation Software Platform. Retrieved March 29, 2017 from <http://dpsp.sourceforge.net/>
- Engelfriet, A. 2010. Choosing an open source license. *IEEE software*, 27(1).
- Eric S. Raymond, 1999. *The Cathedral and the Bazaar*, O'Reilly Media. Retrieved, March 3, 2017 from <http://www.catb.org/~esr/writings/cathedral-bazaar/>

- Fogel, K. 2005. Producing open source software: How to run a successful free software project. O'Reilly Media, Inc.
- Garderen, Van, P. 2009. The ICA-AtoM project and technology. In *linhal*. in Third meeting on Archival Information Databases, Rio de Janeiro, Brazil, 16, 17.
- GitHub. 2017. Open Source Guides. Retrieved March 3, 2017 from <https://opensource.guide/>
- ICA. (n.d). About the ICA-AtoM Project. Retrieved March 14, 2017 from <https://www.ica-atom.org/about>
- ICA. 2015. ICA Statement on Access to Memory (AtoM). Retrieved March 14, 2017 from <http://www.ica.org/en/ica-statement-access-memory-atom-0>
- ICA-AtoM. n.d. Open Source Archival Description Software. Retrieved March 15, 2017 from <https://www.ica-atom.org/>
- MICROSOFTWARE. 2017. “[Special Report] 혁신을 원한다면 오픈소스를 선택하라” 검색일: 2017.3.3. <http://www.imaso.co.kr/?p=16052>
- Mockus, A., Fielding, R. T., &Herbsleb, J. D. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology(TOSEM)*, 11(3), 309-346.
- Newman, S. 2015. Building microservices. “O'Reilly Media, Inc.”.
- Omeka. n.d. Serious Web Publishing. Retrieved March 15, 2017 from <http://archivemata.org/>
- Raymond, E. 1999. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3), 23-49.
- RRCHNM. n.d. History. Retrieved March 14, 2017 from. <https://rrchnm.org/our-story/history/>
- The InterPARES 3 Project, Team Canada (2014). Case Study 16: City of Vancouver Archives: Requirements Analysis for a Digital Archives System.
- veraPDF. 2015. Industry Supported PDF/A Validation. Retrieved March 26, 2017 from <http://verapdf.org/>
- Wilson, A. n.d. Access Across Time: How the NAA Preserves Digital Records. Retrieved March 29, 2017 from [http://www.erpanet.org/events/2003/rome/presentations/Wilson\\_presentation.pdf](http://www.erpanet.org/events/2003/rome/presentations/Wilson_presentation.pdf)
- Wolf, Eberhard, 2016. Microservices : flexible software architecture , Addison-Wesley Professional. (김영기 역. 2016. 마이크로서비스 : 유연하고 확장 가능한 소프트웨어 아키텍처 , 서울: 에이콘).