

# 오픈소스 중요성과 시사점

이진휘([jhlee@nipa.kr](mailto:jhlee@nipa.kr))

공개SW팀 소프트웨어산업본부

2019. 07. 24

## 목 차

### I. 오픈소스

1. 개념
2. 라이선스
3. 커뮤니티

### II. 오픈소스 중요성

1. 경제·사회 패러다임과 오픈소스
2. 국가혁신체제와 오픈소스
3. 4차산업혁명과 오픈소스
4. AI산업육성과 오픈소스
5. 오픈소스 활용역량

### III. 우리의 현실

### IV. 시사점

# I. 오픈소스

## 1 개념

- (정의) SW의 내용을 프로그래밍 언어로 나타낸 설계도인 소스코드가 공개되어, 특정 라이선스 방식을 통해 배포되고 수정, 복제, 사용, 재배포가 자유로운 SW를 지칭
  - 저작권자가 SW개발하여 특정 라이선스 조건으로 소스코드<sup>1)</sup>를 공개(Open 혹은 reveal)한 것을, 다른 사람이 관련 정보에 접근(access)할 수 있고, 2차 저작물을 자유롭게 생성(create), 수정(modify), 배포(distribute)가 가능한 혁명적인 SW개발 모델
    - \* OSI(Open Source Initiative)는 10개의 엄격한 오픈소스 정의 조건을 제시(<https://opensource.org/osd>)
  - (History) 애초에 SW 기술의 사유화를 반대하는 개발자들의 사회운동에서 출발하여 개인 개발자들이 자발적으로 커뮤니티에 참여하여 SW를 개발하고, 그 결과물을 공개하여 공유하는 행동이 SW산업에서 관행으로 정착
    - SW생산에서 개방형 개발이 독점적인 대안보다 더 효과적이라는 아이디어에 기초
      - \* 통상 지식이 협업(연구자)으로 생산(논문)되어 평가·공유(학회)되어 지속적으로 발전되는 것과 동일한 논리
  - (상용SW와 오픈소스) 모두 저작권은 존재하지만 저작권 행사 방식에서 차이
    - 상용SW 저작권자는 저작물 코드를 공개하지 않고 적절한 사용료를 요구
    - 반면, 오픈소스 저작권자는 특정 라이선스로 소스코드를 공개하고 누구나 복제, 설치, 사용, 변경, 재배포가 자유롭도록 허용
    - MIT대 Cusumano 교수<sup>1)</sup>는 기존 상용SW기업이 서비스 기업으로 변모하는 주된 이유도 기업의 전략적인 오픈소스 활용 때문이라고 주장
      - \* 미국 상용SW기업이 설립 후 20년이 경과하면 제품(라이선스) 매출을 서비스 매출이 초과

<표1> 私的독점(상용)과 오픈소스 비교<sup>2)</sup>

구분	私的 독점SW	오픈소스 SW
코드 판독	-코드에 대한 정보를 바이너리 코드로 제공하기 때문에 코드 판독 불가능	-코드에 대한 정보를 오픈소스로 제공하기 때문에 판독 가능
라이선스료	-사용 시 라이선스료 부과(라이선스+서비스료)	-라이선스료 부과 없음(서비스료)
SW 업그레이드 주체	-개발업체만 버그 수정과 업그레이드가 가능	-사용자 혹은 커뮤니티 참가자
보안상 허점 인식주체	-개발업체만 SW 보안상 허점을 알 수 있음	-사용자들의 엄격한 검토로 보안성이 높음
신제품 출시 시간	-신제품 출시에 통상 2-3년의 장시간 소요	-개발주기가 3개월로 짧음
IPR	-개발업체	-여러 단체가 소유

- (SW는 오픈소스) 최근 SW는 오픈소스를 의미하고 유능한 SW개발자는 오픈소스에 익숙하고 활용능력을 보유하고 있다는 가정이 당연히 되는 시대에 진입

1) 소스코드는 사람이 읽을 수 있는 형식의 프로그램 코드를 말하며, 실행코드 또는 바이너리 코드는 컴퓨터가 인식하고 실행할 수 있는 형식의 코드를 지칭한다. 소스코드를 컴퓨터가 읽을 수 있도록 바이너리 코드로 변화하는 것을 컴파일(compile)이라고 함

□ (오픈소스 생산·공유) 오픈소스는 인터넷을 이용하여 세분화된 노동 분업으로 생산되는 디지털 시대 이전의 생산 방식과는 완전히 다른 **혁명적 SW생산 방식**

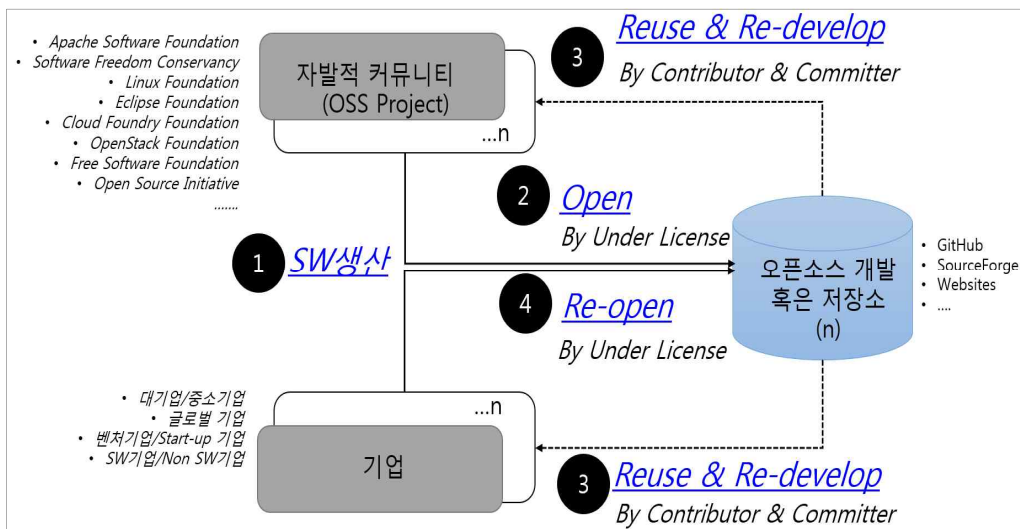
○ 최근 오픈소스 생산은 **자발적 커뮤니티**뿐만 아니라 **기업의 적극적인 참여와 지원**으로 발전되며, 특정 **정보기술의 발전 방향을 결정**

\* 최근 상용SW사는 소스코드를 공개하고 커뮤니티를 운영 혹은 후원하며 SW생산, 네트워크 및 판매 채널로 활용<sup>3)</sup>하고, 비즈니스 모델을 컨설팅, 기술지원 등 서비스 사용료를 청구하는 서비스 모델로 전환

\* 이론적 설명(경제학)은 기업 투자(Private)와 집단적(Collective)투자로 개발된 공공재(Public goods)<sup>4)</sup>

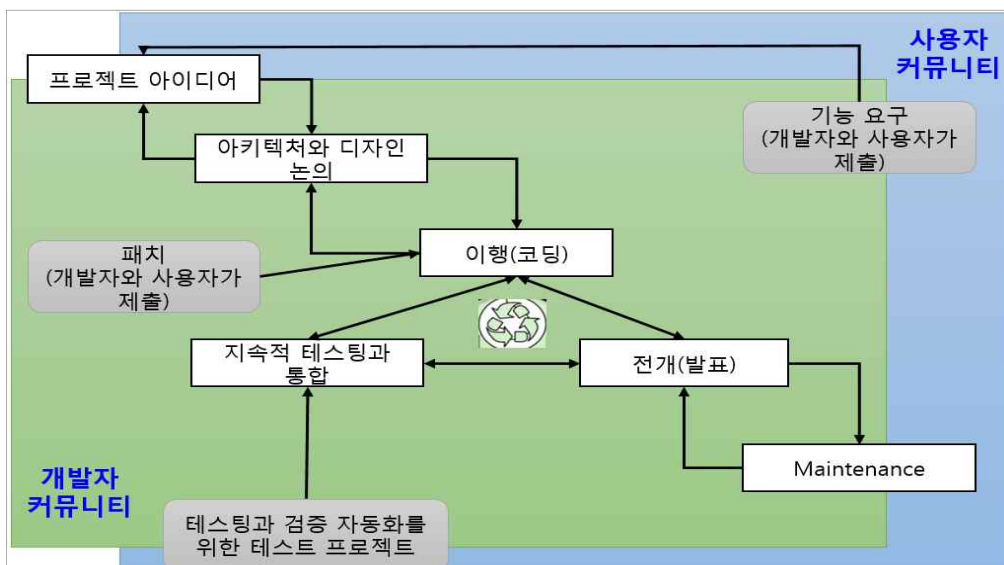
○ 오픈소스 개발 생태계는 기업을 중심으로 고객(사용자), 오픈소스 커뮤니티, 재단·연합, 자발적인 개발자, 투자자 등으로 구성

<그림1> 오픈소스 생산·공유 개념도



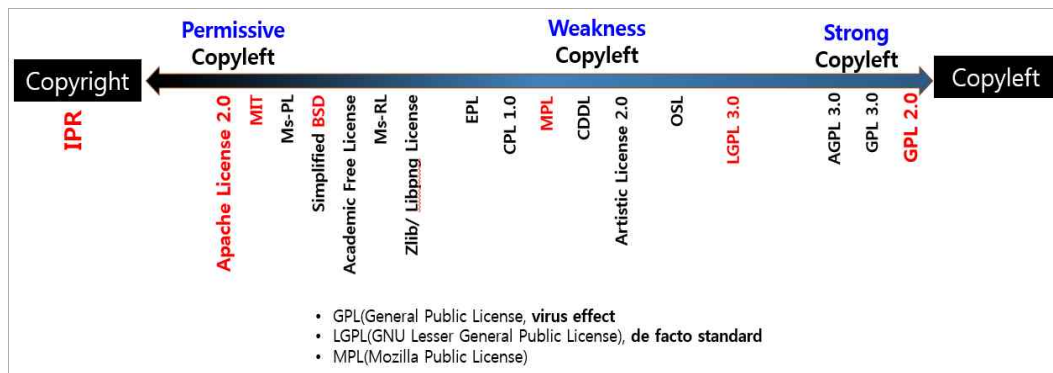
□ (오픈소스 개발모델) 오픈소스 SW 개발모델은 자발적인 개발자들이 특정 SW를 생산하는 **오픈소스 프로젝트 커뮤니티**와 공개된 SW에 대해 버그 확인, 테스트, 새로운 기능 향상, 기타 문서화 등 의견 제시 등을 수행하는 **사용자 커뮤니티**가 상호작용으로 지속적으로 발전

<그림2> 오픈소스 개발 모델<sup>5)</sup>



- (오픈소스 라이선스) 오픈소스 라이선스는 기업 혹은 커뮤니티가 소스코드를 공개할 때 제시하는 제반(소스코드의 사용, 생산, 수정 및 배포 등) 규정
  - (제정 목적) 소스코드 공개를 강제하기 보다는 오픈소스의 사유화 방지 목적이 강함
  - (라이선스 유형) 약 1,000여종 존재, 소스코드가 공개되고 추가 수정 및 배포할 수 있는 권리, 기타 조건을 포함하는 라이선스 모델은 크게 GPL, LGPL, MPL, BSD로 구분
    - OSI(Open Source Initiative) 승인 득한 라이선스는 87종 중에서 60여종이 사용(2018.3 현재)
    - 아파치나 MIT계열은 오픈소스를 활용할 때 제약이 거의 없으나, GPL은 2차 저작물에 대한 소스코드 공개 의무 등이 보다 강력함)

<그림3> 라이선스 유형<sup>7)</sup>



- 법적관점에서 라이선스가 이용허락 여부 또는 2차적 저작물의 문제에 대해서 어떻게 표현하고 있는가와 관련하여 permissive & Strong copyleft로 분류
  - \* 이용허락(permissive)은 사용함에 있어서 별다른 요구사항을 부여하지 않고 광범위한 권한을 부여
  - \* copyleft는 소스코드가 사용되어지게 함, 라이선스는 원래 저작물과 그에 따른 파생 저작물에 적용되어야 함
  - \* virus effects : GPL하에서 오픈소스를 사용하거나 그것을 기반으로 하여 제작된 모든 SW는 다시 오픈소스가 될 수밖에 없는 순환적인 구조를 갖게 됨
- (라이선스 유형선택) 기업, 커뮤니티, 개인 사용자 및 경쟁업체 간의 협업을 위한 기반을 제공하고 SW개발 활동 수준에 결정적인 영향
  - \* 전 세계 수 백만개의 오픈소스 프로젝트들은 모두 저작권이 존재하므로 해당 코드에 부여된 저작권, 라이선스 의무사항을 면밀히 검토하여 사용하는 것이 가장 중요
  - \* 최종 사용자를 대상으로 하는 프로젝트는 상대적으로 제한적인 라이선스, 반면 개발자, 인터넷 또는 상용OS를 지향하는 프로젝트는 제한이 적은 라이선스를 사용하는 경향을 보임<sup>8)</sup>
  - \* 서비스 제공에 중점을 둔 회사는 오픈소스 라이선스하에 제품을 공급하는 경향이 있는 반면, 가족이나 개인 소유의 회사는 독점 SW에 의존하는 경향<sup>9)</sup>
- (라이선스 준수 의무) 상업용 SW와 같이 오픈소스 이용 시 해당라이선스를 준수해야 하며, 위반할 때에는 이용 권리가 박탈되고, 제품화 한 경우에는 더 이상 제품 판매가 불가
  - 최근 저작권법 위반, 라이선스 위반 등에 관한 분쟁 사례가 급증<sup>10)</sup>
    - \* 미국의 Jacobsen사건, 오라클과 구글의 저작권 분쟁 소송, 국내H사의 오픈소스 라이선스 위반사건 등

□ (오픈소스 공개범위 및 전략) 기업이나 커뮤니티는 개발된 소스코드를 어느 정도 (물리적 공개범위) 공개하고 어떻게 혹은 누구에게 공개(공개전략)하는 전략을 구사

○ (공개범위) 통상 전부(All)를 공개하지 않고, 선택적(selectively)으로 일부를 공개

\* Github 설립자인 프레스톤-베너는 핵심 비즈니스 가치가 있는 모듈(Rails)은 절대 공개하지 않고 IPR로 판매하고, 범용도구(Grit, Rescue 등)만 공개<sup>11)</sup>

○ (공개수준) 대부분 선택적으로 부분을 공개(opening part)하든지 부분적으로 공개(partly open)하는 2가지 전략을 채택

- 부분을 공개하는 것은 전체 SW에서 선택적으로 일부의 소스코드를 공개하는 전략으로 이중(Hybrid) 라이선스가 대표적

\* 예를 들면 기본 버전(Community Edition)은 공개하고, 특정 기능이나 추가 기능(엔터프라이즈 기능)은 상용(라이선스)으로 판매, 혹은 코어(핵심) 모듈은 비공개, 주변 모듈은 공개

- 부분적으로 공개하는 것은 고객관점에서 소스코드에 대한 접근 및 사용 등을 특정 그룹에게만 부여하는 것으로 이해관계자 그룹별로 차별되게 공개

\* 듀얼 라이선스(Dual Licence)는 어떤 고객 군(학생)에게는 공개하고, 타 그룹 군은 오픈소스 활용을 제한 혹은 영리목적 vs. 비영리 목적 사용에 제한

□ (오픈소스 공개이유) 기업이 소스코드를 공개하는 이유는 기업이 처한 상황에 따라 다양한 이유들을 제시하고 있으나, 가장 일반적인 이유는

○ 모든 것이 연결되는 4차 산업혁명 플랫폼에서 타 기업과 협력(개방형 혁신)하지 않으면 연결할 수 없고, 또한 자사가 모든 것에 대해 표준을 주도할 수 없기 때문

\* Github의 프레스톤-베너는 회사광고, 코드를 공개하면 유능한 개발자(기여자)를 끌어들이 SW개발에 힘의 승수효과(Force multiplier) 창출, 이미 자사의 SW에 경험을 가진 유능한 개발자의 채용이 쉬움을 제시

□ (오픈소스 공개방법) 기업이나 커뮤니티는 오픈소스 저장소(Github, Sourceforge 등)에 공개하거나 자사가 구축·운영하는 오픈소스 포털에 공개

○ GitHub는 오픈소스 개발에 필요한 SW 저장과 버전관리, 개발자 커뮤니티의 협업과 온라인 교류를 위한 다양한 기능을 제공하는 소셜 코딩의 대표적인 사이트

\* 사용자는 저장소를 탐색하고 소스코드 등을 다운로드 할 수 있고, 토론, 저장소 관리, 다른 저장소로의 기여 제출, 코드의 변경사항 검토 등 기능 활용이 가능

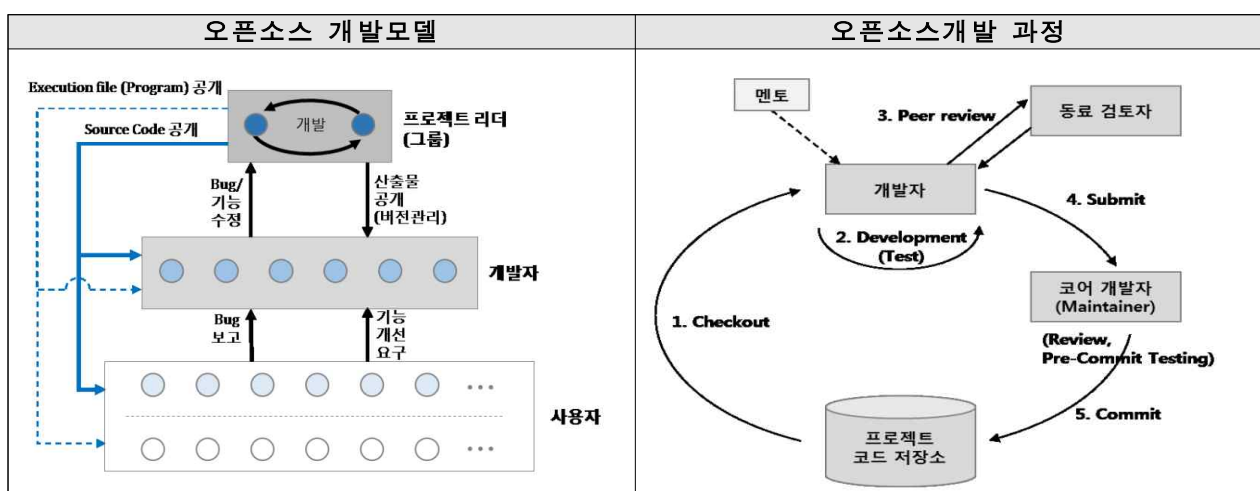
\* 깃(Git)은 2006년경 리누스 토발즈가 직접 개발한 분산버전관리 시스템

<표2> 주요 오픈소스 저장소와 제공 기능<sup>12)</sup>

오픈소스 저장소	운영조직	코드 리뷰	버그 추적	웹 호스팅	번역 시스템	메일 기능	포럼	SW개발	팀	바이너리 배포
Azure DevOps	Microsoft	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Buddy	Buddy, LLC.	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
GitHub	GitHub, Inc	Yes	Yes	Yes	No	No	No	3rd-party	Yes	Yes
GitLab	GitLab Inc.	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
GNU Savannah	Savannah	Yes	Yes	Yes	No	Yes	No	No	Yes	Unknown
SourceForge	BizX LLC	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes

- (정의) 온라인에서 오픈소스 프로젝트를 생성되면 자발적으로 참여하는 사용자와 개발자로 구성되어<sup>13)</sup> SW 개발을 위해 온·오프라인으로 소통하고 협력하는 집단
- 누구에게나 프로그램의 소스코드에 대한 동등한 접근을 보장하고 책임과 권한을 공유하며 지속적인 개발자와 사용자의 기여에 의해서 프로젝트가 발전하는 커뮤니티
  - 설립주체(자발적 vs. 기업), 참가자 유형별(개발자 vs. 사용자)<sup>14)</sup> 등으로 구분
- (커뮤니티와 오픈소스 개발) 오픈소스 커뮤니티 참여자들의 지식 창출은 사회적, 집단적 검토·수정을 통해 체계적 형식지로 통합되고, 이러한 지식은 다시 커뮤니티에 내재화되는 누적적, 순환적 혁신모델
- 우선 개인이나 기업이 개발한 최초 SW를 공개하면 공개된 실행파일과 소스코드의 유용성으로 인해 개발자 사회에서 관심을 받음
    - 개발자(소스코드 및 실행파일의 사용자)는 프로젝트 운영자가 제공하는 관련 정보를 이용하거나 Q&A, 토론 등에 참여
    - 개발자는 공개된 소스코드에 자신만의 새롭고 개선된 코드를 추가하거나 버그를 보고하고 수정하는 기여자, 커미터, 메인테이너 등 역할로 커뮤니티에 참여
  - 프로젝트 리더는 사용자들에게 적극적인 오픈소스의 유용성을 알리고 적극적인 동참을 독려하기 위해 뉴스그룹 등 다양한 활동을 전개

<그림4> 오픈소스 SW 개발모델 및 개발 과정



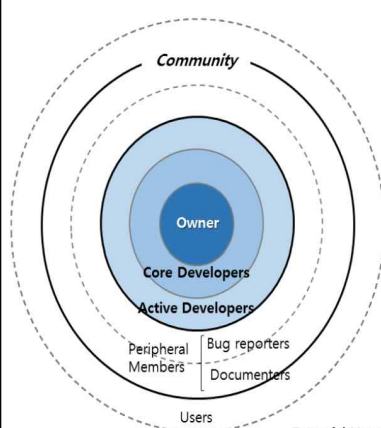
- (프로젝트 운영 성공요소) 초기 버전을 지속적으로 개선하고 유지할 수 있는 개발자들의 확보와 그들 간의 활성화된 사회적 상호작용

\* 투명성(오픈소스 비전, 로드맵, 릴리스 계획, 형상관리 계획, 커미터 자격 조건, 새 기능 추가 또는 패치의 제출 과정 등)과 오픈소스 프로젝트에 대한 문서화



- 통상 오픈소스 프로젝트는 핵심(core)개발자, 프로젝트 리더, 코드 작성자, 사용자(active users)로 구성되고, **양과 모형**이며 **프로젝트별로 크기(size)**에서 차이를 보임<sup>15)</sup>
- 커뮤니티는 일반적으로 기술적 단계, 오픈소스 단계, 생태계의 3가지 단계로 성장

<그림5> 오픈소스 커뮤니티 구조 및 구성원별 역할

커뮤니티 구성원 <sup>16)</sup>	커뮤니티 구성원별 역할	
	<b>소유자</b>	초기 오픈소스를 개발한 창시자로 커뮤니티의 비전과 방향에 대한 의사결정, 라이선스 체계, 비즈니스 모델 수립 등 프로젝트에 대한 대부분의 중요한 책임과 역할
	<b>핵심개발자</b>	소수의 핵심 개발자 그룹, 프로젝트의 결과물에 대해 코드 승인권한과 투표의 권리를 갖고 배포 관리와 같은 책임
	<b>액티브 멤버</b>	액티브 개발자는 새로운 기능을 위한 코드개발과 문서작업, 버그를 수정하는 개발자로 코어 멤버들과 함께 커뮤니티의 중추적인 역할을 수행 버그 수정자는 해당 프로젝트의 소스코드에 이해를 바탕으로 커뮤니티에 보고된 버그들을 수정하며 특별한 권한이 없음
	<b>주변 멤버</b>	커뮤니티의 대다수를 차지, 커뮤니티 내의 지식과 관심사, 버그 등을 공유하며 산발적으로 버그수정이나 마이너한 기능 수정, 테스트에 참여하면서 프로젝트에 기여
	<b>일반 사용자</b>	커뮤니티 멤버가 아닌 해당 오픈소스 사용자로 때때로 의견이나 질문 등을 통해 프로젝트에 기여

□ (커뮤니티 참여 동기) 개인(미시적) 개발자 수준과 조직(거시적) 수준의 참여 동기는 기술·경제·사회적 동기에서 **뚜렷한 차이**를 보임

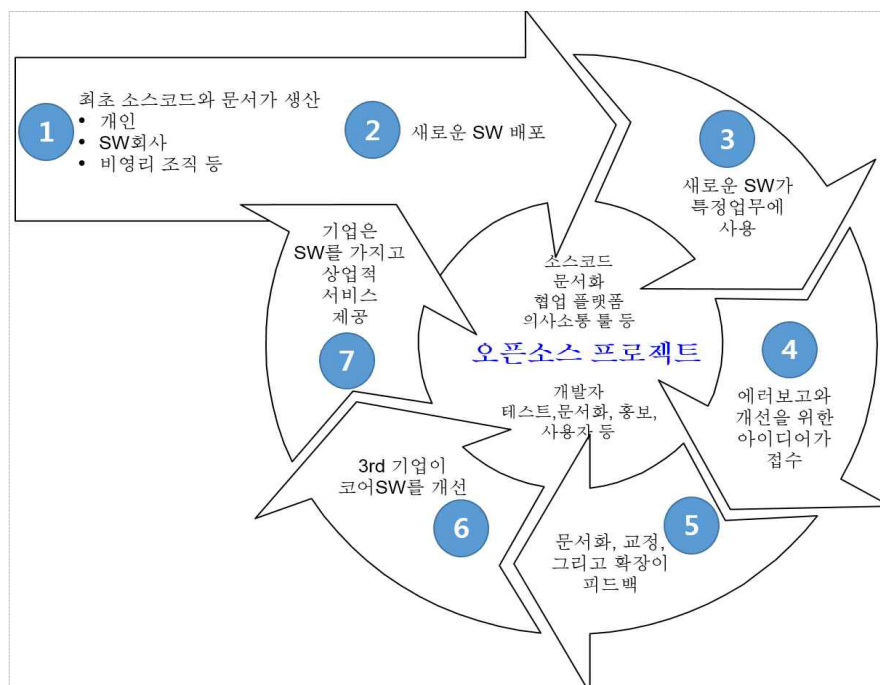
- 기업은 개발비용 절감, 개발기간 단축, 고품질 SW개발, 종속성 탈피 등이고, 개인 개발자는 이타주의와 개인주의 사이의 동기가 골고루 분포

<표3> 오픈소스 개발 참여·공유 동기<sup>17)</sup>

	개인 개발자 수준(미시적) 동기	조직 수준(거시적) 동기
<b>기술적 동기</b>	<ul style="list-style-type: none"> <li>-개인적 기술적 필요성을 충족시키기 위해</li> <li>-동료평가 시스템 활용으로 SW개발 효율성을 높이기 위해</li> <li>-최첨단 기술을 익혀 사용하기 위해</li> <li>-사용자 및 고급 개발자와 <b>상호작용</b>을 통한 기술학습 기회확보</li> <li>-사용자 니즈파악 및 멘토십을 활용한 학습</li> </ul>	<ul style="list-style-type: none"> <li>-개발비용이 높아지고 질은 떨어지는 SW 위기에 대응하기 위해(오픈소스 개발방식 도입을 통해 <b>적은 비용으로 신속히 고품질의 SW개발</b>)</li> <li>-단순하고 지루한 개발작업(테스트나 문서화)를 사용자와 나누어서 협업으로 수행하기 위해</li> <li>-오픈소스 커뮤니티를 통한 <b>기업 R&amp;D활동의 보완 및 강화</b></li> <li>-기업의 혁신을 촉진하기 위해</li> <li>-소스코드 공개로 SW개발 및 활용의 투명성을 확인하기 위해</li> </ul>
<b>경제적 동기</b>	<ul style="list-style-type: none"> <li>-경력관리(오픈소스 경력)에서 이익을 얻기 위해</li> <li>-코딩 스킬 향상을 위해</li> <li>-스톡옵션으로 부를 얻기 위해</li> <li>-낮은 기회비용(잃을 것이 없음)</li> </ul>	<ul style="list-style-type: none"> <li>-오픈소스를 가지고 투자자의 열정을 이끌어 내기위해</li> <li>-상용SW 중심의 SW산업을 서비스 산업 패러다임으로 전환하기 위해</li> <li>-기업의 브랜드 가치를 높이기 위해</li> <li>-관련 제품과 서비스 등을 판매하여 간접적인 수익을 얻기 위해</li> <li>-개발도상국에 SW를 수출하기 위해</li> <li>-시장독점 SW에 비해 저렴한 가격으로 <b>비용절감</b>을 위해</li> </ul>
<b>사회, 정치적 동기</b>	<ul style="list-style-type: none"> <li>-자기 성취욕구(SW기획, 개발에서 경영진이 아니라 개발자 자신이 직접통제로 성취욕구 달성)</li> <li>-자신의 능력을 알리기 위해</li> <li>-코딩 자체에 대한 동기</li> <li>-커뮤니티 소속감, 이타주의</li> </ul>	<ul style="list-style-type: none"> <li>-<b>MS에 대항</b>하는 사회운동</li> <li>-디지털 격차의 극복(SW는 누구에게나 자유로와야 함)</li> <li>-미래의 직업모델 개발(SW영역을 넘어서는 새로운 작업방식으로 오픈소스 SW개발방식)</li> </ul>

- (기업과 오픈소스 커뮤니티) 최근 기업은 자체적으로 개발한 SW의 소스코드를 공개하고 자체적으로 커뮤니티를 구축하여 운영하거나 후원하는 추세가 증가
  - (기업)은 자체적으로 구축 혹은 후원하는 커뮤니티를 SW생산기지(developer community), 네트워크 혹은 판매 채널(사용자)로 활용<sup>18)</sup>
    - 기업이 구축·운영하는 커뮤니티는 사용자에서 시작하여 SW생산 커뮤니티로 발전
    - 기업은 커뮤니티에 접근(access)하여 회사와 커뮤니티의 전략을 일치(align)시키고, 커뮤니티에서 개발된 결과물을 통합(integrate)·동화(assimilate)시키는 능력이 필요
  - (기업의 오픈소스 활용단계) 특정 SW가 생산되어 소스코드가 공개되면, 기업은 이것을 활용하면서 자별적인 개발자 혹은 사용자의 개선사항 등을 피드백 받아 반영하거나, 혹은 제3의 기업이 핵심기능을 개발하는 등 당초 SW가 선 순환적으로 지속적으로 발전

<그림6> 기업의 오픈소스 활용절차



- (벤처기업과 커뮤니티) 오픈소스 경험정도, 인적자본 보유수준 등에 따라 벤처기업의 혁신 성과(매출액, 이익 등)가 증가하는 경향
  - 커뮤니티와 협력한(vs. 협력 없음) 벤처기업이 상대적으로 혁신성과가 더 탁월
  - 커뮤니티와의 공동 작업하는 벤처기업은 높은 품질 등으로 VC의 주목을 받아 재원부족을 보완할 수 있음
  - 커뮤니티와 협력하는 벤처기업은 SW생산에 사용된 프로그램에 대해 라이선스 비용을 지불하지 않아 원가절감이 가능



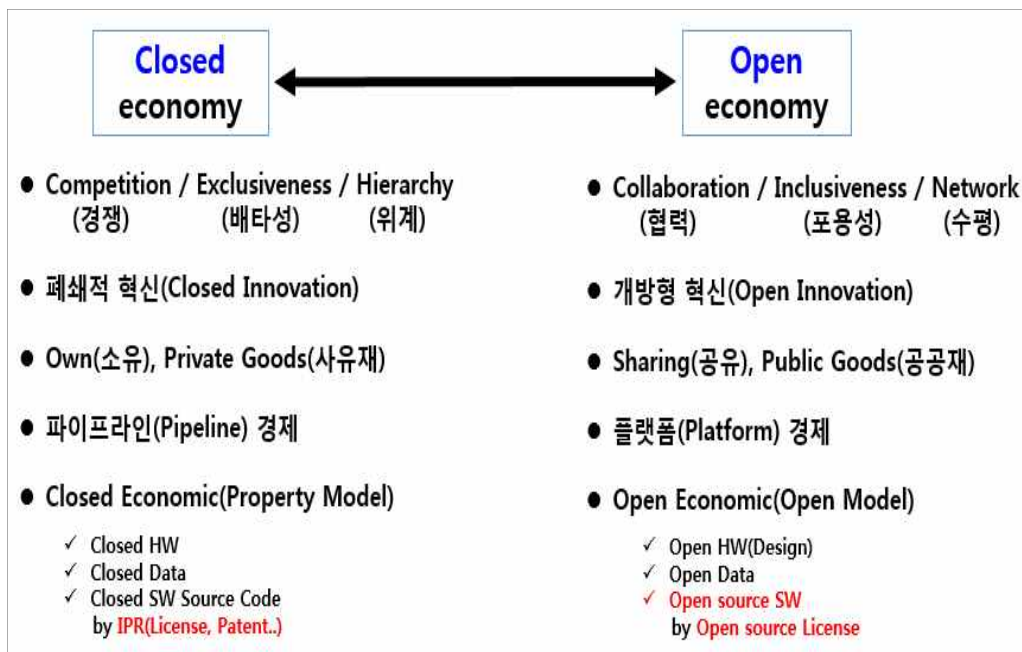
## II. 오픈소스 중요성

### 1

### 경제 · 사회 패러다임 변화와 오픈소스

- (경제·혁신 패러다임) ICT 발전으로 경제 및 사회 패러다임이 경쟁 → 협업 수직 → 수평, 폐쇄 → 개방, 소유 → 공유, 파이프라인 → 플랫폼 기반, 제조 → 서비스 등으로 급속히 전환 중
- (경쟁·폐쇄적에서 협업·개방형 혁신) 기업 활동의 초점이 기업 내부중심 의 폐쇄적 혁신에서 각 단계마다 외부와 협력하는 개방형 혁신으로 이동

<그림7> 경제·사회 패러다임 변화와 오픈소스



- (소유에서 공유·구독 경제) 제품, 생산설비, 서비스 등을 소유할 필요 없이 필요한 만큼 빌려 쓰고, 타인에게 빌려 주는 공유·구독 경제로 변화
  - (파이프라인에서 플랫폼 경제) 하나의 가치사슬(R&D, 생산, 유통, 소비)로 연결된 파이프라인 경제(규모의 공급경제)에서 외부 생산자와 소비자의 상호 작용으로 새로운 가치 창출이 가능한 플랫폼 기반 경제(규모의 수요경제)로 급속히 전환 중<sup>19)</sup>
  - 기존 각 산업 플레이어(SW사업자 포함)는 과거 정보시스템을 기반으로 타 산업과 협업하고 SW를 활용하여 산업별 플랫폼 사업자로 진화
    - 기존 산업은 제조업(스마트 팩토리), 금융(Fintech), 농업(Smart Farm) 등 스마트, 지능화, 연결된 제품 및 서비스를 생산하고 소비하는 환경으로 변모
  - 플랫폼 기반 생태계는 생산자, 판매자, 소비자의 구분이 없어지고 고정된 네트워크(이해관계자)가 아닌 임의적인 형태로 진화
    - 파이프라인 시대에는 자동차, 전자, 정보통신산업 등과 같은 업종 구분이 있으나, 플랫폼 시대는 업종 간 경계가 사라져 모호해짐
- \* 포터와 헤펠만(2015, HBR<sup>20)</sup>)은 기존 제조업들은 IT/SW를 활용하여 제품→스마트 제품→스마트, 연결된 제품→제품 시스템→시스템의 시스템 단계 기업으로 변모되며 경쟁 상대가 확대되고 모호해진다고 주장

<그림8> 산업별 플랫폼 기반 사업자(예시)

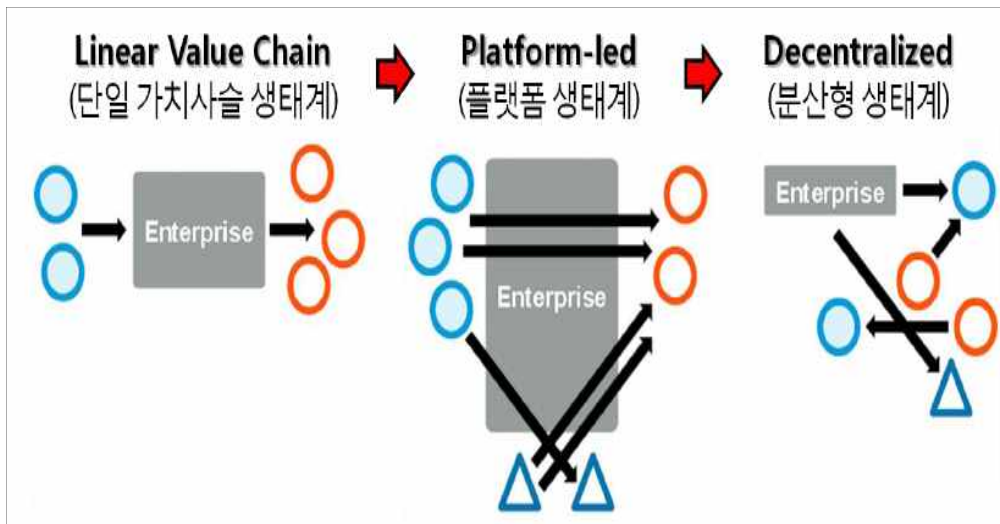
플랫폼간 경쟁방식 개념도	산업별 대표 플랫폼 기반기업 사례																										
	<table border="1"> <tr><td>농업</td><td>John Deere, Intuit Fasal</td></tr> <tr><td>교육</td><td>Udemi, Skillshare, Coursera, edX, Duolingo</td></tr> <tr><td>에너지</td><td>Nest, Tesla, Powerwall, GE, EnerNOC</td></tr> <tr><td>금융</td><td>Bitcoin, Lending club, Kickstarter</td></tr> <tr><td>의료</td><td>Cohealo, Simphyinsured, Kaiser Permanente</td></tr> <tr><td>게임</td><td>Xbox, Nintendo, Playstation</td></tr> <tr><td>인력·직업</td><td>Upwork, Fiverr, 99designs, Sittercity, LegalZoom</td></tr> <tr><td>지역서비스</td><td>Yelp, Foursquare, Groupon, Angel's List</td></tr> <tr><td>물류와 배달</td><td>Munchery, Foodpanda, Haier Group</td></tr> <tr><td>미디어</td><td>Medium, Viki, YouTube, Wikipedia, Huffington Post, Kindle</td></tr> <tr><td>소매업</td><td>Aamzon, Alibaba, Walgreens, Buberly, Shopkick</td></tr> <tr><td>교통</td><td>Uber, Waze, BiablaCar, GrabTaxi, Ola Cabs</td></tr> <tr><td>여행</td><td>Airbnb, TripAdvisor</td></tr> </table>	농업	John Deere, Intuit Fasal	교육	Udemi, Skillshare, Coursera, edX, Duolingo	에너지	Nest, Tesla, Powerwall, GE, EnerNOC	금융	Bitcoin, Lending club, Kickstarter	의료	Cohealo, Simphyinsured, Kaiser Permanente	게임	Xbox, Nintendo, Playstation	인력·직업	Upwork, Fiverr, 99designs, Sittercity, LegalZoom	지역서비스	Yelp, Foursquare, Groupon, Angel's List	물류와 배달	Munchery, Foodpanda, Haier Group	미디어	Medium, Viki, YouTube, Wikipedia, Huffington Post, Kindle	소매업	Aamzon, Alibaba, Walgreens, Buberly, Shopkick	교통	Uber, Waze, BiablaCar, GrabTaxi, Ola Cabs	여행	Airbnb, TripAdvisor
농업	John Deere, Intuit Fasal																										
교육	Udemi, Skillshare, Coursera, edX, Duolingo																										
에너지	Nest, Tesla, Powerwall, GE, EnerNOC																										
금융	Bitcoin, Lending club, Kickstarter																										
의료	Cohealo, Simphyinsured, Kaiser Permanente																										
게임	Xbox, Nintendo, Playstation																										
인력·직업	Upwork, Fiverr, 99designs, Sittercity, LegalZoom																										
지역서비스	Yelp, Foursquare, Groupon, Angel's List																										
물류와 배달	Munchery, Foodpanda, Haier Group																										
미디어	Medium, Viki, YouTube, Wikipedia, Huffington Post, Kindle																										
소매업	Aamzon, Alibaba, Walgreens, Buberly, Shopkick																										
교통	Uber, Waze, BiablaCar, GrabTaxi, Ola Cabs																										
여행	Airbnb, TripAdvisor																										

\* 플랫폼 사업자는 각 산업에서 시작하여 산업별 대표 플랫폼으로 성장(상기 그림에서 의료 B)된 이후에 타 산업분야로 확장되어가는 플랫폼 간 경쟁시대가 도래

□ (플랫폼에서 분산형 생태계로 진화) 과거 일 방향 단일 가치사슬(파이프라인) 생태계에서, 다수 참여자(NN) 간 거래하는 플랫폼 생태계를 거쳐, 향후 분산형 생태계로 진화할 전망이다

- 분산형 생태계는 어떤 기업과도 협업 가능한 완전히 연결된(fully networked) 생태계
- 인터넷, 모바일, DB 등 舊 정보기술이 현재의 플랫폼 생태계를 이끌었다면, AI, 블록체인, 클라우드 등은 제품·서비스 간, 산업 간, 플랫폼 간 경계를 허물고 데이터, SW 기반 연결을 극대화시켜 분산형 생태계 구축을 주도

<그림9> 과거·현재·미래 비즈니스 생태계 진화과정



□ (패러다임 전환과 오픈소스) 패러다임 전환은 SW생산에서 자발인 개발자들의 협업 생산과 소스코드 공개·공유로 새로운 SW가 개발되는 혁명적 SW 개발방식의 출현에 기인

- 오픈소스는 세상의 모든 것이 플랫폼을 중심으로 서로 연결되어 공존·공생·공진화로 변화하는 패러다임에서 생산되고 공유되는 公共財

\* 오픈소스는 사용에 대한 배타성(Exclusivity)과 경쟁(Rivalry)의 정도에 따라 4가지로 구분할 때, 배타성과 경쟁 정도가 낮은 전형적인 公共財

## 2.1 국가혁신체제와 오픈소스

- (국가혁신체제)는 국가 구성원이 경제적으로 가치 있는 지식을 생성·확산하고 혁신활동에 활용도를 높이는 국가차원의 지식의 순환 구조를 구현하는 것<sup>22)</sup>
  - 주요 성공요인으로 구성원의 새로운 지식의 창출, 상호작용을 통한 지식 창출과 학습 전달을 위한 사회적 연결 관계(사회 인프라), 학습 참여, 新 기술 개발 의지를 제시
    - \* 사회 인프라란 지식을 전달시킬 통로로 개인간 교류, 기업간 파트너십, 산업과학계의 연구개발협력 등
  - 3가지 핵심요소를 의도적(예: 정책적)으로 구현하는 것은 매우 어려운 과제
    - \* 국가혁신체제 구축 어려움은 기술지식이 특정 구성원에 고정되는 경직성(lock-in), 정책수행과 연구개발을 위한 자원의 부족, 제대로 정비되지 않은 사회 인프라, 적극적으로 상호작용을 할 동기 부족을 거론
    - \* 대학에서 연구 개발하여 기업으로 기술이전을 시키고 지역 등에 클러스터를 구축하여 운영하는 정부정책은 정책효과를 거두기가 매우 어려움
- (국가혁신체제와 오픈소스) 오픈소스를 국가혁신체제의 구축과 운영에 활용
  - (오픈소스는 지식의 공급원) 오픈소스로 생성된 풍부한 기술지식과 프로젝트의 결과물들은 혁신성장에서 가치 있는 지식의 공급원
    - 개인이나 기업의 노력으로 생성된 지식을 공유하도록 강요하는 것이 어려워 대부분 혁신체제는 학계의 연구개발로 창출·공유
    - 반면, 오픈소스는 SW개발에 필요한 지식과 기술이 공개된 것으로 결과물은 상업적 활용은 금지되지만, 개발과정에서 축적된 기술을 누구나 제한 없이 활용이 가능
      - \* 오픈소스 학습·활용을 위해서는 수준별 SW 기술능력이 필요
  - (오픈소스는 사회 인프라) 오픈소스 커뮤니티가 제공하는 참여자간의 경제적 또는 사회적 연결 관계는 혁신체제에서 상호작용(쌍방향)을 위한 사회 인프라
    - 오픈소스 커뮤니티에서의 의사소통은 혁신체제가 요구하는 쌍방향적 상호작용 기반의 학습 기반을 제공하고 오프라인에서는 암묵지도 용이하게 전파 가능
  - (오픈소스와 학습·기술개발 참여의지) 새로운 기술을 활용하는 것에 비해 성공적인 적용사례가 많은 오픈소스는 학습 및 새로운 기술 개발 의지를 높여줌
    - 성공한 오픈소스의 기술지식은 새로운 기술을 받아들이는 이의 입장에서는 상대적으로 학습과정이 용이하고 성과에 대한 확신을 가질 수 있음
    - 기업이 새로운 기술을 활용할 때 호환성과 성과 불확실성으로 수용에 소극적이 되고 선발기업이 네트워크 효과를 누리면 기술개발 의지가 낮아짐

## 2.2 산업혁신과 오픈소스

□ **(개방형 혁신과 오픈소스)** 기업이 오픈소스 역량을 확보하는 것은 개방형 혁신을 목표를 달성하는데 필요·충분조건

- **(개방형 혁신)**은 기업이 내부 아이디어를 외부와 공유하거나, 외부 것을 내부에 동화(활용)시킬 경우, **기술(비즈니스) 혁신이 가능해지고 가속화됨**<sup>23)</sup>

\* 내부 아이디어의 외부와 공유는 자사에는 이익이 없더라도 타사의 수익 창출에 도움, 반대로 타사 아이디어를 내부에 활용하는 것은 외부에서 혁신을 가져 와서 자체 혁신에 적용하는 능력(프로세스 및 지식)을 갖추게 됨

- **(오픈소스)**는 SW기업이 내부 소스 코드를 외부와 공유하고, 외부 소스 코드를 내부로 가져 오는 것까지 가능하므로 **아이디어와 지식을 공유하는 차원 이상임**  
- 기업은 직접 모든 일(SW개발 등)을 직접 하지 않고, 외부에서 개발된 소스코드를 가져와 효과적인 혁신 추진이 가능하여 **개방형 혁신을 촉진하는 핵심 매개체**

\* SW산업에서 개방형 혁신은 요구공학 프로세스에서 새로 생성된 아이디어와 소비자 혜택 발견 등과 같이 내·외부에서 점점 더 이용되며, 어려움은 올바른 파트너 찾기<sup>24)</sup>

□ **(기술 혁신과 오픈소스)** 오픈소스는 **기존 혁신 결과물의 재사용 또는 수정으로 혁신이 가속화 되고, 외부와 더 많이 공유하고 협력하면 새로운 기술개발이 더 쉽다는 것을 증명**

- 소스코드 공개와 공유로 경제 주체간의 개발 노력이 조정되어 중복된 노력의 결과에 대한 위험이 감소하고, 품질 보장이 가능해져 후속 기술개발 혁신을 촉진  
- 반면, 독점 기술혁신에 종사하는 경쟁 업체는 기술의 모든 측면을 비밀로 간주(IPR)하여 지식을 공유하거나 도움을 줄 가능성은 거의 없음
- 기술혁신과 오픈소스와의 관계에 대한 연구에서 각각의 성공에 필요한 조직역량이 매우 유사하다고 보고<sup>25)</sup>

<표4> 기술혁신과 오픈소스 성공을 위한 공통 조직역량(예)

조직역량	기술혁신과 오픈소스 세부 역량
외부 공동체와의 협업	-기업이 자체적으로 모든 것을 하는(기술개발, SW개발, 제조 등) 것보다 외부와 협력하는 것이 혁신을 더 가속화시킴 -소스코드는 기업 혁신을 위한 외부 지식 및 리소스 중 하나
아이디어를 외부로 공유	-내부 아이디어 발굴은 충분하지 않고 아이디어를 외부에 공유하고 외부에서 자사에 기꺼이 기여할 수 있는 지 확인하면 혁신을 가속화함
조직 학습, 외부 아이디어의 내부 동화	-외부 기술 혁신이나 오픈소스를 관찰하는 것은 불충분, 외부 지식과 외부 코드를 자사에 동화시킬 수 있는 역량을 갖추어야함
재사용/수정의 효율성	-오픈소스와 기술 혁신 모두에서 기존 혁신 결과물의 재사용 또는 수정으로 혁신을 가속화
고객 가치에 대한 전략적 접근	-Morgan & Finnegan은 이것을 "전략적 오픈소스"라고 지칭 -전략적으로 어떤 것을 아웃소싱하고 어느 것을 사내 제작해야하는지, 또 어떤 부분은 고객이 가치 창조에 참여토록 할 것인지 등
새로운 기술개발에서 낮은 진입 장벽	-외부와 더 많이 공유하고 협력하면 새로운 기술개발이 더 쉬움

## 3.1 4차 산업혁명

□ (4차 산업혁명) 경제 패러다임 변화에서 4차 산업혁명은 가치사슬 기반의 파이프라인 경제에서 플랫폼 기반 경제로 전환되는 것<sup>26)</sup>을 의미

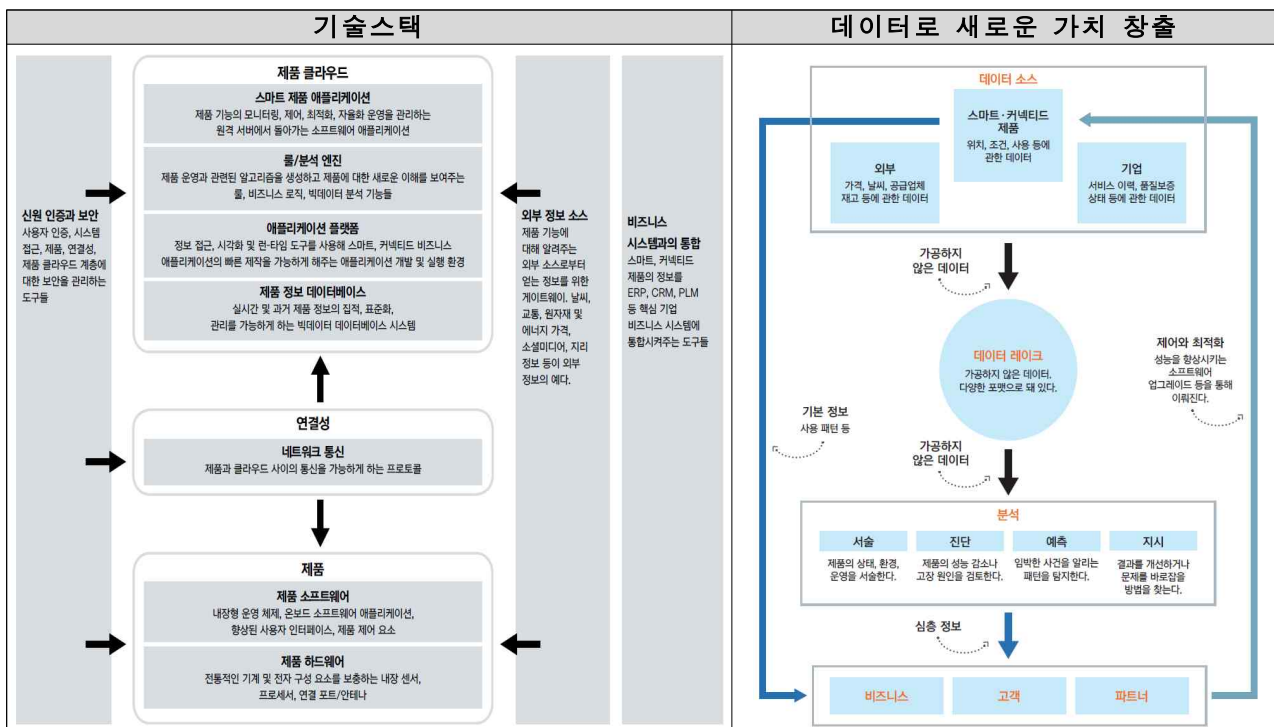
○ 舊 정보시스템 환경과 완전히 다른 새로운 (기술) 인프라를 이용하여 제품이 플랫폼과 연결(Connectivity)되고 지능화(Intelligence)된 새로운 시스템을 구축·운영하는 것

- 스마트 & 연결 제품(SCP : Smart & Connected Product)을 만들고자 하는 기업은 다수 층으로 구성된 “기술 스택”이라는 완전히 새로운 기술 인프라가 필요<sup>27)</sup>

\* 제품에 내장된 HW, SW 및 운영체제, 연결성을 위한 네트워크 통신, 제조사 혹은 제3자 서버에서 작동하는 SW로서 제품 정보 DB를 저장하는 제품 클라우드, SW애플리케이션 개발을 위한 플랫폼, 룰 엔진 및 분석 플랫폼, 제품 외부에서 작동하는 스마트 제품 애플리케이션 등

\* 또한 모든 층을 가로지르는 요소로는 신원 인증과 보안구조, 외부 데이터 접근을 위한 게이트웨이, SCP에서 얻는 정보를 ERP나 CRM 같은 기타 비즈니스 시스템에 연결하는 도구

<그림10> 4차 산업혁명 구현 새로운 기술스택



○ (데이터로 새로운 가치 창출) SCP에서 얻은 데이터와 외부 소스와 기업 내부 소스에서 얻은 데이터로 구성된 데이터 저수지(Big data)에 모인 데이터를 분석하여 새로운 가치 창출에 활용

○ 4차 산업혁명은 현실(사물, 제품 등)을 IoT·블록체인으로 연결(센싱)하여 다양한 데이터를 수집·저장·가공(Big data)하고 분석·판단(AI 추론, 학습)하여, SCP가 즉각 반응하거나 혹은 사용자가 n-스크린 단말로 통제할 수 있는 자동화된 시스템을 구축·운영하는 것

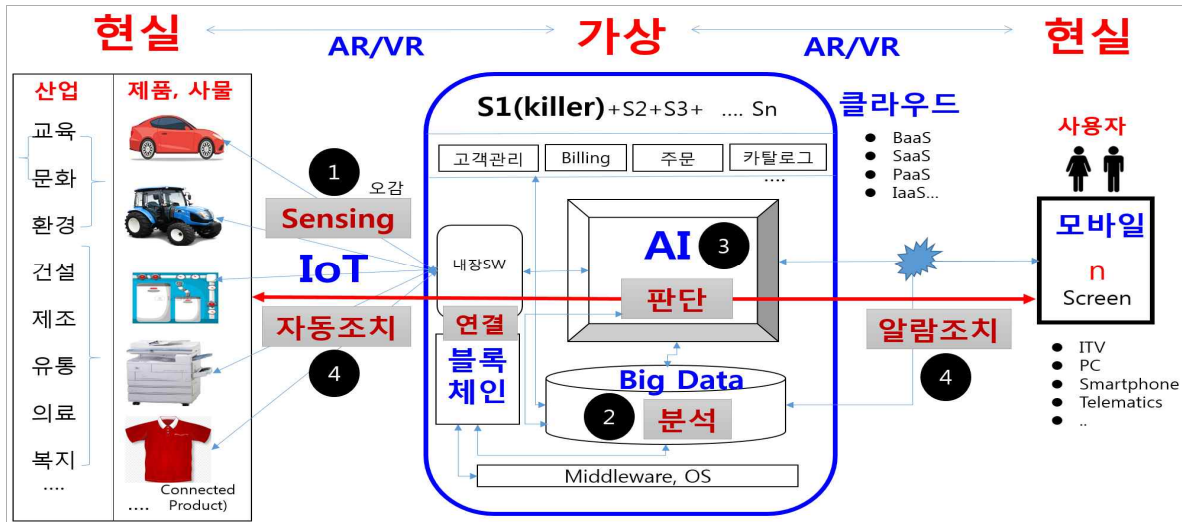
\* 舊 정보시스템은 HW/OS, 시스템SW 기반위에 조직별 특화된 주요 업무와 인사 회계 등 ERP(보조업무)를 구축·운영



□ (4차 산업혁명 구현기술) AI(뇌·신경), 블록체인(근육), IoT, Big data(혈액), AR/VR, 이들 기술의 공유·구독이 가능하도록 해주는 클라우드(뼈대) 컴퓨팅 처리 방식

○ 포터와 헤펠만이 제시한 기술스택은 아래와 같이 도식화가 가능

<그림11> 4차 산업혁명 구현 기술(예시)

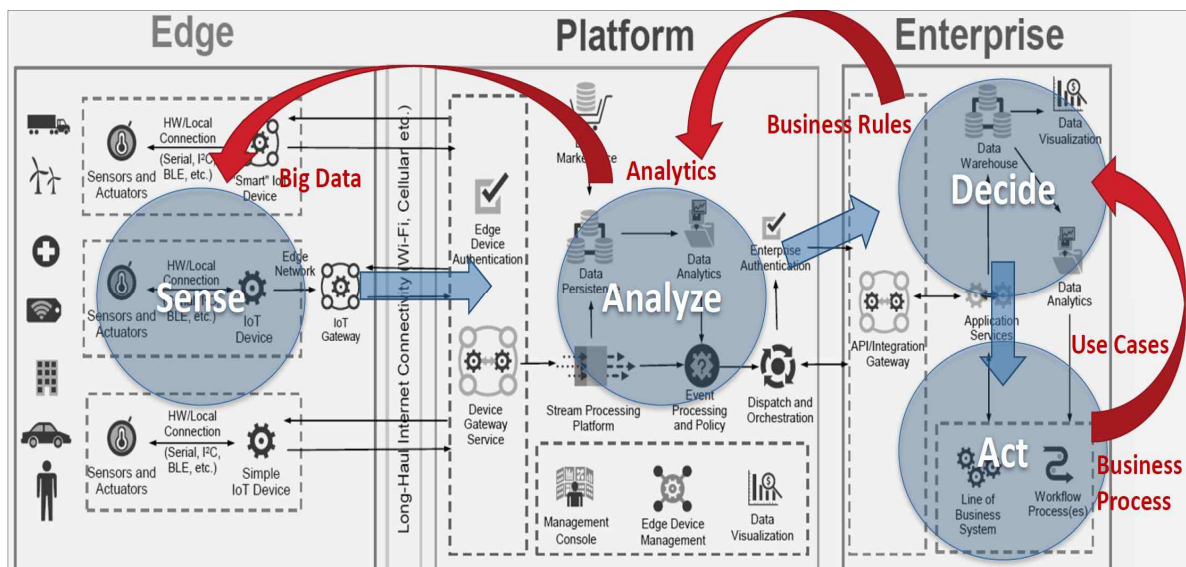


○ 기술 스택을 구축하고 지원하기 위해서는 상당한 투자가 수반되며, SW 개발, 시스템 엔지니어링, 데이터 분석, 온라인 보안 전문 지식 등 지금까지 기업에서 찾아보기 힘든 다양한 분야의 새로운 기술 역량이 요구

- (AI) 다양한 형태의 데이터를 수집하여 학습·추론으로 분석·판단
- (5G, IoT, 블록체인) 4G보다 20배 이상인 5G와 사물(IoT)·조직(블록체인)을 연결
- (클라우드) 컴퓨팅 자원(HW, 응용서비스 등) 구독서비스(IaaS, PaaS, SaaS 등) 기술

○ 궁극적으로 플랫폼은 연결된 제품과 기존 내부, 외부 시스템과 연결되어 데이터를 수집·분석·활용하는 선 순환적 자동화된 시스템을 구축하는 역량이 요구

<그림12> 4차 산업혁명 구현 세부 적용기술(예시)<sup>28)</sup>





### 3.2 4차 산업혁명과 오픈소스

- (4차 산업혁명과 오픈소스) 4차 산업혁명 구현을 위한 기술 인프라는 과거와는 완전히 새로운 인프라가 요구되어 대부분 기업이 오픈소스를 전략적으로 활용
  - AI, 블록체인 등 핵심 기술들은 제품 개발 후, 대부분 오픈소스로 공개·공유되며, 기타 시스템 구축에 필요한 대부분 SW도 오픈소스 활용이 가능

<그림13> 4차 산업혁명 구현 오픈소스 기술(예시)

IoT	EVO OPENIoT BRILLO oneiota mycroft iotsys Contiki	Network	보안
AI	TensorFlow learn Caffe TORCH K Keras		
Block-Chain	MultiChain eris openchain		
Cloud		Big-Data	
openstack. Delta Cloud cloudstack OpenNebula EUCALYPTUS APACHE KVM		Lumify APACHE STORM APACHE SAMOA HPC SYSTEMS Hadoop	
DB/DBMS	mongoDB MariaDB CUBRID PostgreSQL MySQL		
WAS/WEB	NGINX GULP.JS Bower GlassFish Bootstrap ANGULARJS		
Management Software		Application	
ProjeQtOr GANTT project freedcamp Clubhouse Bitrix24 ProjectLibre OpenProject OrangeScrum		SoapUI JUnit	
Server/OS	ubuntu CentOS fedora Linux ORACLE LINUX SUSE redhat. VULinux		

연결과·센싱		데이터 (수집·저장·처리)	분석	분석·시각화
센싱 (IoT)	블록체인			
<u>Horizontal platforms</u> ·Canopy ·Chimera IoT ·DeviceHive ·DSA ·Pico Labs ·M2MLabs ·Nimbits ·OSIoT ·RabbitMQ ·SiteWhere ·webinos ·Yaler  <u>Middleware</u> ·IoT Sys ·OpenIoT ·OpenRemote ·Kaa 등	·Bitcoin ·Ethereum, EOS ·Fabric(IBM) ·Sawtooth(intel) ·Iroha, Openchain ·HydraChain ·MultiChain ·Loopchain ·Elements ·Stella, NEO, Lisk ·R3 Corda ·Wanchain ·NXT, ARK ·STRATIS ·ICON, CARDANO ·theloop, ripple ·Credits	<u>수집</u> ·내부(Sqoop) ·외부(Crawling) ·로그(Flume)  <u>분산저장</u> ·Hadoop HDFS ·Solr, Beam ·Pulsar, CockroachDB ·Neo4j, InfluxDB  <u>클라우드</u> ·Openstack ·Eucalyptus ·Cloud Stack ·OpenNebula  <u>처리</u> ·Mapreduce ·PIG(스크립트언어) ·HIVE(데이터 요약, 질의, 분석)	<u>SPARK</u> ·Java 등 API ·SQL ·Streaming ·MLLIB ·Graph  <u>딥러닝 플랫폼</u> ·TensorFlow ·Theano, Torch ·Keras, CNTK ·PyTorch ·Apache MXNet ·Apache SINGA ·BigDL, Caffe ·Chainer ·DeepLearning4j	·BIRT ·Zeppelin ·Knowage ·SpagoBI ·MapD ·Jaspersoft ·Klume ·Metabase ·Matomo ·Seal Report

- (선진국과 오픈소스) 선진국은 SW산업을 포함하여 모든 산업(금융, 공공, 의료, 교육 등)에서 다양한 新 사업, 新 서비스를 개발을 위해 오픈소스를 활용
  - (글로벌 SW기업)은 AI, 블록체인 등에서 새로운 SW를 개발할 때부터 소스코드를 공개하여 사용자 저변 확대 등 빠른 기술혁신 및 표준기술 선점전략 추진
    - 글로벌 기업들이 많은 투자를 통해 개발한 SW를 공개하는 주된 이유는 자사 주도의 생태계 조성을 위해

\* 예를 들면 인공지능 구현 모듈들을 직접 개발하는 것은 많은 투자가 필요하나, 이미 개발된 인공지능 개발 툴이 공개되면 규모가 작은 스타트업도 손쉽게 참여가 가능

<표5> 글로벌 SW기업의 오픈소스 활용현황

구분	MS	IBM	Google	Facebook	아마존	관련 재단 등
AI	<u>Project Oxford</u> <u>MS CNTK</u> <u>Azure ML</u>	<u>SystemML</u> <u>Watson</u> <u>CODAIT</u>	<u>DeepMind</u> <u>Tensorflow</u>	<u>Torch</u>	AWS SageMaker	<u>OpenAI</u> <u>Partnership on AI</u>
블록체인	Azure(EBaaS)	Bluemix BaaS	-	암호화폐	AWS	<u>Linux재단</u> <u>(Hyper Ledger)</u>
IOT	Azure	<u>Bluemix</u>	Brillo	Parse	AWS	<u>AllSeen(AllJoyn), oneM2M(&amp;CUB</u> <u>E/MOBIUS), 리눅스재단(Linux</u> <u>Y)</u> <u>OIC, COMPOSE, Eclipse,</u>
빅데이터	Azure	<u>Bluemix</u>	<u>Mapreduce</u>	<u>OCP(Open</u> <u>Compute</u> <u>project</u>	AWS	<u>아파치재단(Hadoop 등)</u>
클라우드	<u>Azure</u> <u>Hyper-V ONE</u> <u>.Net Core</u>	<u>Bluemix</u>	IaaS & PaaS	<u>Cassandra</u>	AWS	<u>Openstack</u> <u>Cloudstack 등</u>

\* 이탈릭체는 각 글로벌 기업이 소스코드를 공개한 오픈소스 프로젝트를 의미하며 시차에 따라 차이가 있을 수 있음

\* IBM은 오픈소스 AI프로젝트로 깃허브내에 CODAIT(Center for Open Source Data and AI Technologies)를 운영

- (미국 기업)은 78%의 회사가 오픈소스 기반으로 운영, SW가 필요할 때 66% 이상의 회사가 오픈소스가 있는지 먼저 확인하고, 39%의 회사가 오픈소스 프로젝트를 계획 중이며, 47%의 회사가 자사의 툴과 프로젝트를 오픈소스로 공개<sup>29)</sup>

- 미국 기업에서 일반적으로 사용되는 1,000개의 응용 프로그램에 대한 최근 감사에서 오픈소스 구성 요소가 96%로 발견, 특히 상용SW에서도 57%가 오픈소스<sup>30)</sup>

- 또한 미국 기업이 활용하는 SW 중에서 오픈소스 사용을 조사<sup>31)</sup>에서 거의 대부분 기업이 핵심기술 부문에서 오픈소스를 활용

\* AI·빅데이터(97%), IoT(91%), VR·게임(100%), 인터넷&인프라SW(100%), Fintech(100%), 통신(98%), 마케팅(93%), 기업용 SaaS(95%) 등

- (오픈소스 활용 유형) SW산업을 포함하여 모든 산업에서 오픈소스를 신 제품 (서비스) 개발, 다양한 비즈니스 모델 등에 활용

- (SW산업과 오픈소스 활용) SW산업을 전통적인 분류인 시스템통합(IT서비스), 상용(패키지), 인터넷 서비스 등으로 구분할 때 오픈소스를 다양하게 활용

- 시스템통합 기업은 예전에는 주로 상용SW 개발 툴을 활용하여 개발을 하였으나, 최근에는 오픈소스 개발 툴을 사용하고 특정 SW를 개발할 때 오픈소스를 활용

- 상용SW 기업)은 SI 기업에서 사용하는 개발 툴과 개발 시의 소스 사용 이외에 개발환경 인프라 구축을 대부분 오픈소스를 활용

- 인터넷 서비스 기업은 다수 사용자를 대상으로 하는 인터넷 서비스의 특성상 상용SW를 활용한 서비스 구축 및 운영에 한계가 있기 때문에, 아이디어를 구현하는 단계에서 오픈소스를 찾아 새로운 서비스 개발에 활용

<표6> SW산업에서 오픈소스 활용유형(예시)

구분		사업모델	세부 비즈니스 방법	대표 기업
SW 기업	상용 SW	오픈소스 활용으로 새로운 상용SW 생산·판매	-인프라SW 부문에서 이미 개발되어 공개된 소스 코드를 활용하여 새로운 상용SW를 생산·공급 -애초 라이선스가 상업용 판매를 허락하는 경우에만 가능, 불허용 시 라이선스 위반문제	-전문가들은 대부분 국내 상용 SW사가 해당, 증거 불충분
		오픈소스 완제품 공급 + 부가 서비스 제공	-오픈소스 커뮤니티가 개발한 완제품을 공급(유통)하고 기술지원 등 부가서비스로 수익 창출	-레드햇 등 국내 진출 글로벌 기업
		소스코드 공개하고 비즈니스 모델 전환	-패키지 판매가 아닌 기술지원, 컨설팅, 교육 등 부가서비스를 제공 수익모델로 변화	-알티베이스 -큐브리드 등
	IT 서비스	오픈소스 응용SW플랫폼 + 추가모듈개발 + 응용SW개발	-블록체인 플랫폼(Hyper Leger)에 가속기술(Nexledger Accelerator)개발하고 통관시스템을 구축	-삼성SDS 등 대부분 IT서비스 기업
	인터넷 서비스	오픈소스를 신규서비스 생산에 활용	-새로운 서비스 개발을 위한 아이디어를 채택하면 관련 SW를 오픈소스를 활용	-네이버, 카카오 등
	클라우드	오픈소스 활용 서비스 개발	-오픈스택 기반의 클라우드 서비스를 개발	-네이버 등
	임베디드	오픈소스 활용 HW제조	-네트워크 장비 등 대부분 기기 제조사가 해당	-HW제조사
	게임	게임 서비스 개발	-게임사가 오픈소스 기반 빅데이터 솔루션을 기반으로 고객관리 시스템에 적용 등	-게임사
Non-SW 기업	금융, 공공 의료, 제조 교육 등	○ 완제품 오픈소스를 自社の 정보시스템 구축에 도입하거나 自社 개발업무에 SI기업에게 오픈소스 활용으로 개발할 것을 위탁, 오픈소스 활용 자체 개발 ○ 오픈소스를 활용하여 문제해결 지식 흡수가 가능하여 기업 혁신 메커니즘에 활용 등		

- (Non-SW 산업과 오픈소스 활용) 교육, 금융, 제조, 공공 등 Non-SW 산업에서도 정보시스템 인프라 구축, 4차 산업혁명 구현 기술스택에 필요한 기술들을 오픈소스로 활용하는 추세 증가
- (전체 산업과 오픈소스 활용) 모든 기업은 SW를 활용하여 비즈니스 모델 전환, 혁신 제품 혹은 서비스 개발을 추진할 때 아래와 같은 유형으로 오픈소스를 활용

<표7> 전체 산업에서 오픈소스 활용유형(예시)

활용유형	세부 내용
(1)IT인프라	- 기업 내부의 IT인프라(OS, NMS, 등 시스템SW) 구축에 오픈소스를 활용
(2)리소스	- AI, Blockchain, Bigdata 등 신 기술을 개발 및 적용하는 데 오픈소스를 활용
(3)H/W	- 전자제품 등 HW생산에 내장되는 SW(임베디드)를 오픈소스를 활용
(4)상용S/W	- ERP, 게임, 워드프로세스, 이러닝 등 응용프로그램 기반 상용SW 개발에 오픈소스 활용
(5)구독과 부가서비스	- 자체 개발한 후 소스코드를 오픈하거나 혹은 오픈소스 커뮤니티가 공개한 오픈소스를 구독 방식(월 사용료 등)으로 제공하며 기술지원, 교육, 컨설팅 등 부가서비스를 제공
(6)IT서비스	- 공공이나 금융 등 민간 기업이 오픈소스를 활용하여 SI(IT서비스) 프로젝트를 수행
(7)클라우드	- 오픈소스를 활용하여 다양한 서비스를 개발하여 제공

□ (SW 개발방법론과 오픈소스) 전통적인 SW개발 단계와 DevOps 개발방법에서도 27가지의 오픈소스 도구가 활용

<표8> SW개발 단계별 활용가능 오픈소스 툴

요구공학	분석 설계	역공학	구현/통합 빌드 단계	테스트	유지보수
Client-Server 모델에서 OSRMT	White Star UML, Argo UML, MSQL, Workbench 등	Eclipse UML, Eclipse IDE	Java(Eclipse와 #Project), C++(Eclipse C, MSVS), Continuous Integration (Maven, Ant), Code Generator 등	GNU Debugger, Xnee, PMD, CPCheck, Junit, CPPUnit 등	Bugzilla, Mantis, Mindquarry, Trac, Redmine 등

## 4

## AI 산업과 오픈소스

### 4.1 AI(인공지능)과 Tensorflow 딥러닝

□ (인공지능) 인간처럼 작동하고 반응하는 지능형 기계, 혹은 감각(sense), 사고(reason)하고 행동(act)하고 적응(adapt)할 수 있는 프로그램

\* 기계학습(ML)은 알고리즘이 시간에 따라 더 많은 데이터에 노출됨에 따라 성능이 개선, 딥러닝은 기계학습의 하위 구성요소로 광대한 데이터로부터 **다층 신경망**이 학습하는 것

○ (AI 분류) 인간대비 AI 능력 기준(Weak, Strong, Super, 혹은 보조, 증강된, 자율지능), 활용범위(Narrow vs. General) 또는 설명 가능한 인공지능(XAI) 등으로 분류

○ (AI 기술 분류) 문제 영역(X축) + 해결방법(Y축) = AI Technology Solution

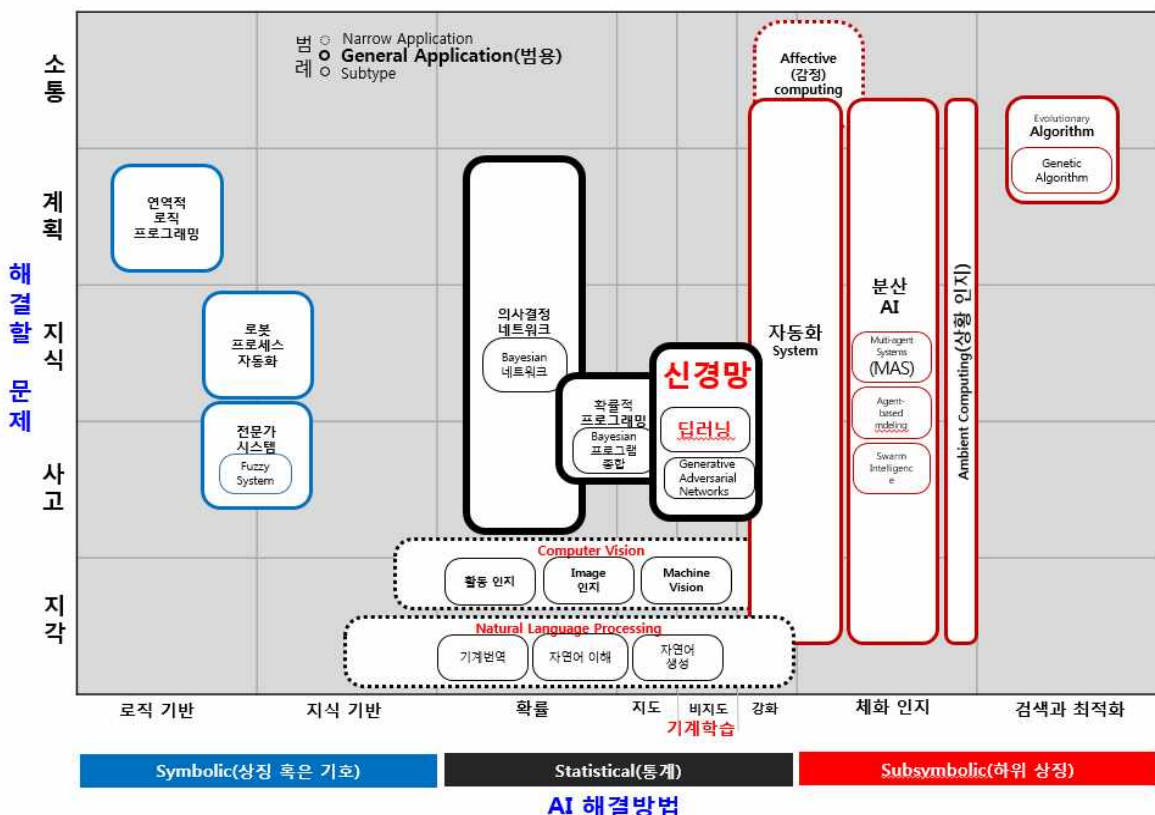
- 문제(지각·사고·지식·계획·소통)와 기술(로직·지식기반·확률·기계학습·체화지능·검색과 최적화)의 조합으로 다양한 심리·사회적 이론과 기술이 융합

- 딥러닝은 인간이 어떻게 생각하고 지식을 축적·의사결정 하는가? 라는 문제(신경망)를 인간의 학습 이론을 적용(기계 학습)한 대표적 범용(General Purpose) 인공지능

- 인공지능은 단일 기술이 아니라 목표(문제 해결)를 위해 상호 보완적인 하나 이상의 다양한 AI기술을 활용하여 해결

\* 지각, 신호/음성/시각인식, 의미이해, 자연어처리, 추론, 의사결정론, 행동계획, 불확실성 처리, 지식표현, 신경망, 학습 등으로 3대 성공 원동력은 알고리즘, 빅데이터, 컴퓨팅 파워

<그림14> 인공지능 세부 기술32)



- (AI 플랫폼과 데이터, 도메인 지식) AI 플랫폼은 다양한 영역의 문제를 해결하기 위한 도구이며, 실제 구현을 위해선 **도메인(Domain) 지식**과 센싱 혹은 운영 시스템으로부터 확보되는 **데이터와 결합**이 필요

\* 언어, 교통, 의료, 금융 등 AI가 적용되는 해당 영역에 대한지식(Domain Knowledge)이 선행되지 않으면 실제로 AI 기술을 구현하는 것은 매우 어려움

\* AI는 알고리즘만으로는 작동하지 않으며, 대량의 데이터를 통한 학습 및 테스트를 거쳐야 상용화가 가능

- (AI 기술 발전 전망) 최근에는 이미지(시각), 언어 등으로 입력(센싱)되는 데이터를 저장·가공하여 학습(지도, 비지도, 강화)시켜 새로운 알고리즘이 개발되며 발전

- 향후에는 문제해결을 위한 **체화된 인지(Embodied cognition)** 등 기술을 적용한 **상황(awareness) 인지, 자동화 시스템, 분산된 AI** 등으로 발전될 전망

<표9> 인공지능 세부 적용기술

문제/해결		세부 요소 기술	기술 세부 내용
사고 · 지식 · 계획	로직	Inductive Logic Programming(ILP)	-형식적(formal) 논리를 사용하여 <b>진리를 DB화</b> 하고, 데이터에서 파생된 가설을 공식화
		Robotic Process Automation(RPA)	-특정 작업을 수행하는 사용자를 <b>관찰하여 수행할 규칙 및 동작 목록을 추출</b> 하는 기술
		Expert Systems	-인간 의사 결정 프로세스를 모방(emulate)하기 위해 <b>hard-coding된 규칙</b> 이 있는 컴퓨터 프로그램, 대표 기술은 <b>Fuzz 시스템</b>
지각		컴퓨터 시각(CV)	-디지털 이미지를 획득하고 이해하는 방법(일반적으로 활동 인식, 이미지 인식 및 기계 비전으로 구분)
		자연어 처리(NLP)	-자연 언어 데이터(언어이해, 언어생성 및 기계번역)
사고 · 지식	확률 · 기계 학습	의사결정 네트워크	-Map(directed acyclic graph)을 통해 변수 집합과 확률론적 관계를 나타내는 베이지안 네트워크/추론 등 일반화(generalization) 이론을 적용
		확률적 프로그래밍	-특정 변수를 하드코딩하지 않고 확률모델로 작업하는 프레임 워크 -Bayesian Program Synthesis(BPS)가 대표 이론
		신경망 (Neural Network)	-인간의 신경망을 모방한 느슨하게 모델링된 알고리즘의 클래스 -딥러닝 (Neural net with multiple layers) -Generative Adversarial Networks(GANs) : 상호 학습시키는 2개의 신경망 네트워크
커뮤니케이션	검색 · 최적화	Evolutionary Algorithms(EA)	-생물학 기반 메커니즘(예 : 돌연변이, 복제 등)을 사용하는 진화론적 계산이라는 보다 광범위한 컴퓨터 과학 영역 - <b>유전자 알고리즘</b> 은 자연 선택 과정에 따라 "가장 적합한" 후보 솔루션을 선택하는 검색 추론인 EA를 장 많이 사용
문제 전체	기계 학습	Autonomous Systems	-지능형 지각, 기민한 객체 조작, 계획 기반 로봇 제어 등
		Affective Computing	-감정 인식, 해석 및 시뮬레이션
	체화 인지	Distributed AI(DAI)	-서로 상호작용하는 자율적인 "에이전트"에 배포하여 문제를 해결하는 기술 클래스 -Multi-agent systems(MAS), Agent-based modeling (ABM), Swarm Intelligence -집단적 행동은 분산된 자기 조직화된 에이전트의 상호작용에서 나온다는 원리 적용
		Ambient Intelligence (Aml)	-외부 자극을 <b>상황인지</b> 하기 위한 센스, 지각, 반응하는 디지털내로 물리적 기기를 요구

□ (인공지능과 오픈소스) 인공지능 세부 기술 중에서 딥러닝, 시각 혹은 이미지(Computer vision), 자연어 처리 분야(NLP) 등에서 다양한 오픈소스 프로젝트가 진행 중

○ 인공지능 관련 오픈소스 프로젝트 수는 2017년(14,000개)을 기점으로 급격히 증가<sup>33)</sup>

- 학습, 추론, 인식 등 알고리즘을 개발할 수 있는 인공지능 플랫폼이 오픈소스로 공개되면서 이를 활용한 기술과 서비스 개발이 비약적으로 증가

<표10> 인공지능 주요 분야별 오픈소스 프로젝트 현황

기계 학습 (딥러닝)	이미지(image) 처리 <sup>34)</sup>	음성(speech) 처리 <sup>35)</sup>	자연어(NLP) 처리	감성(affection) 처리
-TensorFlow -Theano -Torch -Keras -CNTK -PyTorch -Apache MXNet -Apache SINGA -BigDL -Caffe -Chainer -Deeplearning4j -Dlib -Intel Data Analytics -OpenNN -MATLAB + -H2O.ai	-OpenCV -MaltLab -ViFeat -CUDA(NVIDIA) -pcl -AForge -SimpleCV -GPUImage -SciPy -BoofCV -DLib -OpenFace(CMU) -Colorado State University(CSU) -YOLO -openBR -openGL -SPM -MIA,SciPy	-Kaldi -Julius -Wav2Letter++ -DeepSpeech2 -CMUSphinx -Mozilla open source speech to text -Htk -Dragon Naturally speaking -SimonVoxForge -SPRAAK -SPRACH -Project DeepSpeech -SRILM -GMTK	-NLTK -Apache Lucene and Solr -Apache OpenNLP -GATE and Apache UIMA -Apache cTakes -Apache Mahout -SpaCy -TextBlob -Textacy -PyTorch-NLP -Retext -Compromise -MALLET -Natural Nlp.js -OpenNLP -CogCompNLP	-JSiento -Synesketch -emotion-recognition- neural-networks -Emotion -EmotionRecognition -face_classification -Emotion-Recognition- RNN -EmotionNet -emotion-conv-net -DeepSentiment -Voice-Emotion-Detect -Emotion_Recognition -Emotion_Voice_Recognition -Chainer

○ 깃허브(Github) 상에서 2000년부터 2018년까지 생성된 인공지능과 관련된 주요 프로젝트들의 목록을 수집·분석한 결과, 인공지능과 관련된 오픈소스 주요 기술 동향이 알고리즘, 프로그래밍 언어, 응용분야, 개발 도구의 범주로 구분

<표11> 인공지능 관련 토픽10 빈도 현황<sup>36)</sup>

2009~2012		2013~2015		2016~2018	
토픽	빈도	토픽	빈도	토픽	빈도
Machine Learning	57	Machine Learning	652	Machine Learning	15,842
Artificial Intelligence	26	Python	211	Deep Learning	8,299
Python	16	Artificial Intelligence	193	Python	5,174
Neural Network	15	Neural Network	149	Neural Network	4,523
Natural Language Processing	12	Deep Learning	144	Tensorflow	3,644
Java	11	Java	95	Artificial Intelligence	2,555
Genetic Algorithm	7	Natural Language Processing	78	Keras	1,766
C++	7	Data Science	57	Reinforcement Learning	1,493
Java Script	5	C++	43	Data Science	1,339
Deep Learning	5	Computer Vision	38	Convolutional Neural Network	1,278



□ (딥러닝과 오픈소스) 기업·대학·재단(커뮤니티)이 개발·공개된 딥러닝 플랫폼은 약 20여개로 SW산업을 포함하여 **쏘 산업에서 다양한 응용서비스 개발에 활용**37)

- 딥 러닝 SW의 주된 차이점은 **프로그래밍 스타일(명령 vs. 선언)**, 분산 대응 멀티 GPU 여부, 프로그래밍 언어, 기타 제공 기능 여부 등

<표12> 딥러닝 분야 오픈소스 프로젝트38)

No	딥러닝SW	개발자	발표	오픈소스	license	개발언어	스타일	Interface	OpenMP	OpenCL	CUDA
1	TensorFlow	Google Brain	2015	Yes	Apache 2.0	C++, Python, CUDA	선언	Python(Keras), C/C++, Java, Go, JavaScript, R, Julia, Swift	No	SYCL 지원으로 개발예정	Yes
2	Microsoft (CNTK)	Microsoft Research	2016	Yes	MIT	C++	선언	Python (Keras), C++, Command line, BrainScript(.NET 개발예정)	Yes	No	Yes
3	Keras	Francois Chollet	2015	Yes	MIT	Python		Python, R	Theano를 backend로 사용할 때	Theano, Tensorflow backends 사용가능	Yes
4	Caffe	Berkeley Vision	2013	Yes	BSD	C++	명령	Python, MATLAB, C++	Yes	개발 중	Yes
5	PyTorch	Gregory Chanan (Facebook)	2016	Yes	BSD	Python, C, C++, CUDA	명령	Python, C++	Yes	분리된 package를 통해	Yes
6	Theano	Universite de Montreal	2007	Yes	BSD	Python	선언	Python(Keras)	Yes	개발 중	Yes
7	Torch	Ronan Collobert	2002	Yes	BSD	C, Lua	명령	Lua, LuaJIT, C, utility library for C++/OpenCL	Yes	3 party implementation	Yes
8	Apache MXNet	Apache Software Foundation	2015	Yes	Apache 2.0	Small C++ core library	선언+명령	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl	Yes	roadmap	Yes
9	Apache SINGA	Apache Incubator	2015	Yes	Apache 2.0	C++		Python, C++, Java	No	Supported in V1.0	Yes
10	BigDL	Jason Dai (Intel)	2016	Yes	Apache 2.0	Scala		Scala, Python			No
11	Chainer	Preferred Networks	2015	Yes	BSD	Python		Python	No	No	Yes
12	Deeplearning4j	Deeplearning4j community	2014	Yes	Apache 2.0	C++, Java		Java, Scala, Clojure, Python (Keras), Kotlin	Yes	No	Yes
13	Dlib	Davis King	2002	Yes	BSL	C++		C++	Yes	No	Yes
14	Intel Data Analytics	Intel	2015	Yes	Apache 2.0	C, Python, Java		C++, Python, Java	Yes	No	No
15	OpenNN	Artelnics	2003	Yes	GNU LGPL	C++		C++	Yes	No	Yes
16	Wolfram Mathematica	Wolfram Research	1988	No	Proprietary	C++, Wolfram Language, CUDA		Wolfram Language	Yes	No	Yes
17	Intel Math	Intel		No	Proprietary			C	Yes	No	No
18	MATLAB +	MathWorks		No	Proprietary	C, C++, Java, MATLAB		MATLAB	No	No	x

□ (**Tensorflow와 오픈소스**) 구글이 내부적으로 공개(DistBelief)하여 검색, 음성 인식, 번역, 지도, 유튜브 서비스 등 개발에 사용되던 기계 학습용 엔진을 2015년에 오픈

- (정의) Tensorflow=딥러닝을 위한 오픈소스 SW 라이브러리, AI 애플리케이션을 위한 심층적인 학습 모델을 만드는 사람들을 위한 공통 도구

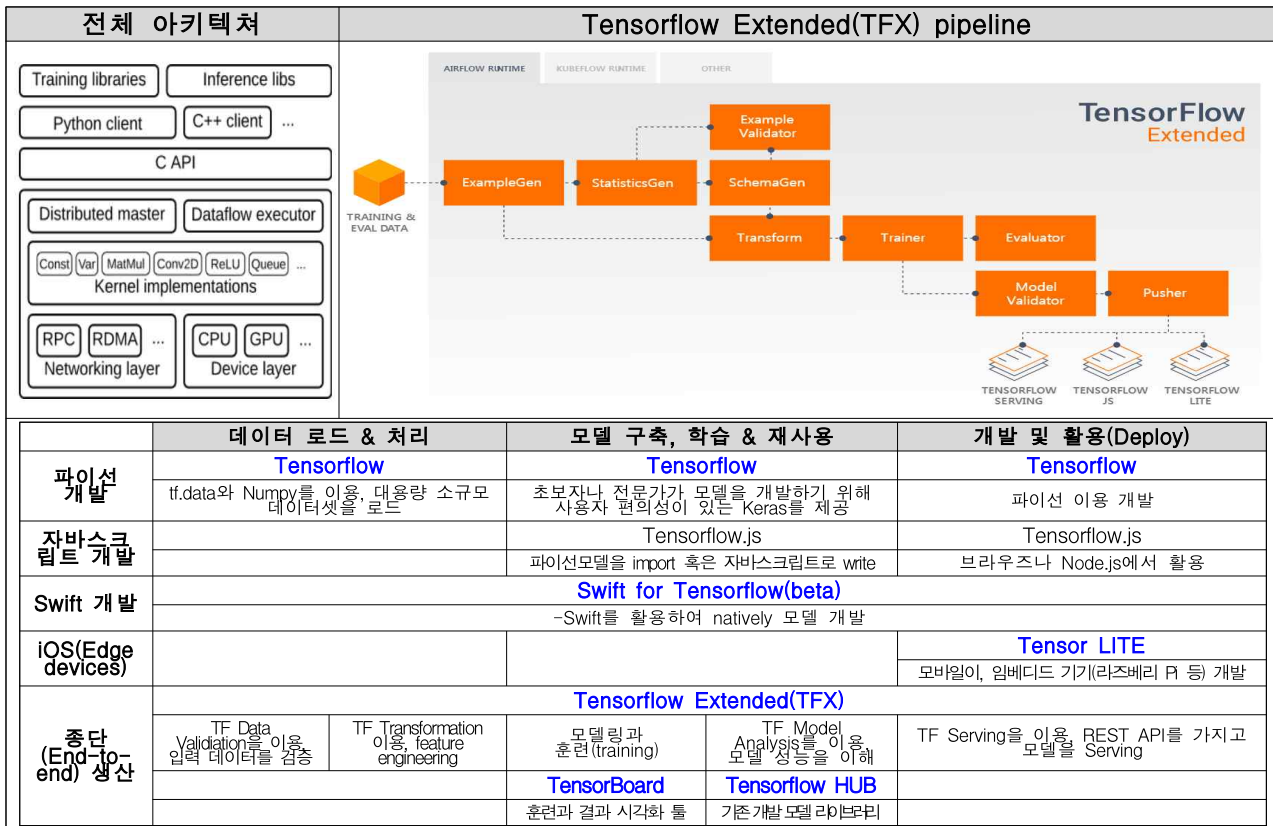
- (아키텍처) 데이터 입력 및 처리, 모델 작성, 모델 훈련과 예측의 3부분으로 구성된 데이터 파이프 라인 설계와 데이터 흐름 그래프 아키텍처

- 데이터 로딩(데이터 생성, 데이터 증강), 학습 데이터/평가 데이터로 분리, 학습(DNN, RNN, CNN, VAE 등 알고리즘 커스텀마이징), 평가(GAN), 모델 저장(정확도 확인 및 데이터 정제 수행), 서비스 활용 順

\* 다차원 배열의 데이터를 입력(tensors)하면, 다중 작업 시스템(선언형 프로그래밍)을 통해 흐르고(flow), 다른 쪽 끝에서 출력되는 구조로 수행 작업은 플로우 차트(그래프)로 제시

\* 전체 아키텍처는 C API를 기준으로 Runtime과 user 모듈이 구분

<표13> Tensorflow 아키텍처<sup>39)</sup>



- (개발 및 운영 환경) 언어(파이선, 자바스크립트), 소형기기, 분산 환경 등 다양한 환경에서 작동하고 다른 딥러닝 플랫폼(예, Keras 등)과 연결되어 사용 가능

\* 초보자용(Lite), 전문가용(TFX), 소형 기기용(Mobile & IoT) 등 3가지 버전

- (주요 알고리즘) 다양한 딥러닝 알고리즘을 제공하고 다른 통계 알고리즘도 지원

<표14> Tensorflow의 알고리즘(예시)<sup>40)</sup>

구분	대표 알고리즘	지원되는 통계 알고리즘
Element-wise mathematical operations	-Add, Sub, Mul, Div, Exp, Log, Greater, Less...	-Linear regression -Classification -Deep learning classification -Deep learning wipe and deep -Booster tree regression -Boosted tree classification
Array operations	-Concat, Slice, Split, Constant, Rank, Shape ...	
Matrix operations	-MatMul, MatrixInverse, MatrixDeterminant ...	
Stateful operations	-Variable, Assign, AssignAdd, ...	
Neural-net building blocks	-SoftMax, Sigmoid, ReLU, Convolution2D..	
Checkpointing operations	-Save, Restore	
Queue and synchronization operations	-Enqueue, Dequeue, MutexAcquire, MutexRelease, ...	
Control flow operations	-Merge, Switch, Enter, Leave, NextIteration	

- (Tensorflow 딥러닝 생태계 구축) 딥 러닝 라이브러리 중에서는 가장 인기 있으며, 공개된 기술을 개선하고, 오류를 수정하는 활동들 모두가 협업으로 추진되며, 분산 시스템 및 기계 학습에 대한 추가 개발이 지속적으로 진행 중

- 소스코드 공개 이후에 8,000명 정도의 개발자가 소스코드 저장소를 분기(fork)

\* 74,000여개 Star, 36,000여 Fork, 480명 이상의 개발자가 참여하여 SW품질을 높이며, TFLearn과 TensorLayer, PrettyTensor 같은 래퍼 라이브러리 등장, TF.Learn은 TensorFlow 내부에서 분산된 기계학습 고급 Python 모듈

- 바이너리 배포판은 50만 번 다운로드, 사용자들은 Tensorflow를 사용하는 수십 가지의 새로운 딥러닝 모델을 개발하고 공유

- (Tensorflow와 국내 연구현황) 국내 학계에서도 쏘 산업 분야에서 Tensorflow를 활용하여 새로운 알고리즘을 개발하는 연구가 활발히 진행

<표15> Tensorflow 활용 국내 연구사례(예시)

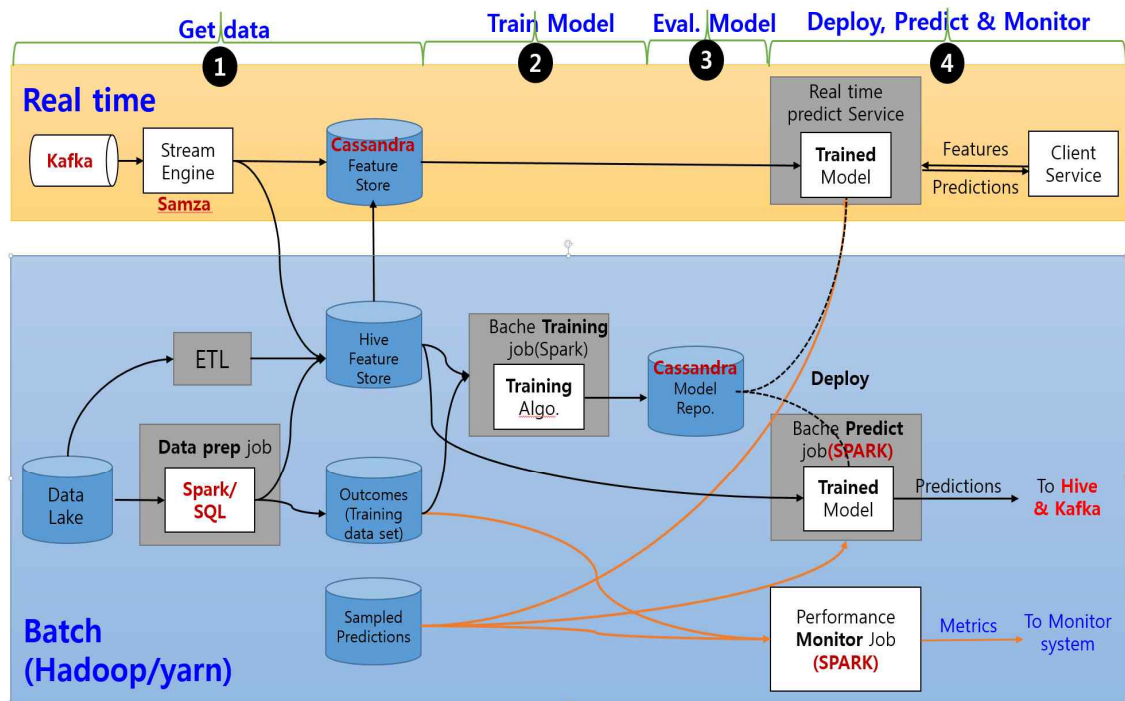
분야	저자	발표	Tensorflow 활용 세부내용
통신	이성호 외	2016	-TensorFlow 기계학습 도구를 이용한 응용 트래픽 분류
IT	황범순외	2017	-Tensorflow를 이용한 실시간 로그 분석 시스템
IT	김도현외	2016	-Tensorflow로 구현한 효과적인 구조의 멀티모달 딥러닝 동영상 분류 시스템
금융	송유정, 이종우	2017	-Tensorflow를 이용한 주가 변동 예측 딥러닝 모델 설계 및 개발
금융	송유정외	2017	-Tensorflow를 이용한 주가 예측에서 가격-기반 입력 피쳐의 예측 성능 평가
연구	김동훈외	2018	-TensorFlow 를 기반으로 한 ICT 응용연구 분석
농식품	정수호,여현	2018	-Tensorflow 기반 과일 분류 시스템 플랫폼 구축
IT	한명훈,이석호	2018	-텐서플로 기반의 모바일 증강현실
IT	김영준,강수용	2018	-안드로이드 시스템 서비스를 이용한 TensorFlow 라이브러리 활용
IT	김정진외	2017	-CNN을 이용한 이미지 분류
IT	백영태외	2017	-OpenCV 영상처리와 TensorFlow 딥 러닝 영상처리 기반의 지능형 실종자 색인 시스템
교통	신동,김병만	2016	-Convolutional Neural Network와 Tensorflow를 활용한 차량 모델 판별
IT	김남성	2017	-머신 러닝을 위한 윈도우 환경에서의 텐서 플로우 활용
농식품	권혁민,김화중	2018	-Tensorflow LSTM 모델을 활용한 음식점 방문지수 예측
농식품	김범준외	2018	-꽃 이미지 분류를 위한 Tensorflow 기반의 CNN 설계
의료보건	원진영외	2018	-독거노인 위급 상황 탐지를 위한 TensorFlow Object Detection API 기반 자체 인식 절차
공공	장성봉외	2019	-텐서 플로우 신경망 라이브러리를 이용한 세수 시계열 데이터 예측
IT	박혜진외	2018	-텐서 플로우 FCNN 모델을 활용한 재활용 물품의 구성요소의 패턴인식 시스템 구축
국토	박진아	2018	-Tensorflow 기반의 콘볼루션 신경망을 이용한 암석 이미지 분류
농식품	양미혜	2017	-농장 데이터 및 Tensorflow를 이용한 머신러닝 기반 방울토마토의 최적 수확시기 예측
농식품	김성진외	2018	-라즈베리파이를 활용한 해충 인식 시스템 설계

## 4.2 오픈소스 기반 AI 적용사례

□ (우버 미켈란젤로 플랫폼과 오픈소스) 빅데이터, 분석 등 솔루션에 오픈소스를 활용하여 기계학습 서비스 플랫폼(Michelangelo)을 개발(2015)·운영

- 초기 승차 공유 서비스(승객/기사)에서 시작하여, 9가지 차량호출 서비스, 음식 주문 배달 서비스, 우버 헬스(환자 이송) 서비스 등 B2C, B2B 서비스로 확장하여 궁극적으로 MaaS(Mobility as a Service) 기업을 지향<sup>41)</sup>
- (ML as-a-service) 우버 사내 엔지니어 및 데이터 엔지니어가 대규모의 기계 학습 솔루션을 쉽게 구축(build)하고 배포(deploy)할 수 있는 플랫폼
  - \* end-to-end 머신러닝 워크플로우로 데이터 관리, 모델 학습(Train), 모델 평가(Evaluate), 모델 활용(Deploys), 예측 (predictions), 예측 모니터링(Monitor predictions) 등 6단계로 구성
- (음식 배달 시스템, Uber EATS ETD)은 음식 주문을 위한 음식점 검색에서 주문, 음식 준비, 배달, 사후관리까지 전체 고객 경험 제고를 위해 AI를 개발 활용
  - 배달 시간예측, 음식점 검색랭킹, 검색 자동완성, 음식점 랭킹 등 AI모델을 구현
    - \* 음식 주문 전, 음식 준비 시간, 배달되는 시간을 예측하여 디스플레이하여 종업원이 업무에 사용

<그림15> 우버 음식배달(EATS) 플랫폼<sup>42)</sup>



- (데이터 준비) 훈련을 위한 데이터는 우버 내부 시스템(data lake)에서 Spark sql을 사용하여 지속적으로 수집
- (특성 저장) 배치와 실시간 기계학습을 위해 공유된 feature computation 엔진과 배치용은 Apache Hadoop용 DW인 Hive feature store에 저장하고, 실시간용은 비정형 DBMS인 Cassandar 혹은 분산 저장(HDFS) 시스템에 저장
- (모델 훈련) 비 딥러닝 알고리즘으로 Spark Mllib와 Xboost, 딥러닝으로 Caffe, Tensorflow를 지원
- (모델 응용 및 관리) 어느 하나의 도시가 아닌 미국 전체 도시의 데이터를 실시간 혹은 배치기반의 예측을 하도록 설계
- (모델 서빙과 실시간 모니터링) 예측 모델에서 배치는 서비스 코어를 스파크 작업에 연결, 실시간은 우버 RPC 서비스에 의해 호스팅되며 2가지 모드의 모든 예측은 진단 및 모니터링을 위해 Kafka(메세지 브로커)에 기록
- (우버 플랫폼과 오픈소스 활용) 우버는 전체 아키텍처와 EATS 플랫폼 구축에 오픈소스와 자체개발 컴포넌트를 조합하여 구축, 활용된 주요 오픈소스는 HDFS, Spark, Samza, Cassandra, MLLib, XGBoost, TensorFlow 등
- 우버는 가능하면 오픈소스에서 분기(fork)하고, 필요 시 기여할 수 있는 성숙한 오픈소스를 사용하고, 自社 환경에 맞는 오픈소스가 없으면 자체적으로 개발

<표16> 우버 플랫폼과 오픈소스 활용

우버 전체 아키텍처		우버 EATS 플랫폼에 활용된 오픈소스	
구분	오픈소스	오픈소스	주요기능
Machine learning	-SPARK	Apache Kafka	-대용량의 실시간 로그 처리에 특화되어 설계된 메시징 시스템 -모든 Uber의 서비스들에서 로깅된 메시지들을 처리
Deep learning	-Tensorflow -Keras -Caffe	HDFS	-Hadoop Distributed FileSystem, 하둡 네트워크에 연결된 기기에 데이터를 저장하는 분산형 파일시스템
Business Intelligence	-Warehouse	Apache Hive	-데이터 웨어하우스 시스템, 배치 특성 저장소
Interactive Queries	-Presto	Apache SPARK	-배치처리를 위한 엔진 -Spark SQL(SQL로 데이터 분석), Spark Streaming(실시간 분석), MLlib(머신러닝 라이브러리), GraphX(그래프 분석)
Realtime Queries	-Elasticsearch -Pinot, MemSQL	Samza	-Apache Kafka를 비롯한 여러 소스에서 스트리밍(실시간) 처리를 해 주는 엔진
Batch analytics	-Hive	Cassandra	-온라인 데이터 저장 데이터베이스
클라우드	-uDocker	XGBoost	-학습 방법 중에서 Bagging 방법(샘플링)을 변형하여 이전 모델이 예측하지 못한 데이터에 가중치를 주어서 다음 모델이 더 잘 학습하도록 하여 bias를 감소시키기 위해 쓰는 방법
사용자 단말기기	-Rstudio -Shiny -Jupyter	Tensorflow	-개발자가 ML 기반 애플리케이션을 쉽게 구축하고 배포
		data lake	-Uber 내부정보시스템에서 생성된 데이터 저장소

- (국내 DoAI사 AI 오픈소스 활용<sup>43)</sup>) 의료분야에서 AI 솔루션을 개발하는 스타트업으로 Tensorflow 등 오픈소스 AI 플랫폼을 활용하여 새로운 모델개발 혹은 서비스를 제공
- AI기반 병리 영상 판독 AI, 보건의료 빅데이터, 유전체 데이터, 약물 스크리닝 데이터에서 의미 있는 패턴을 찾아 개인 특성에 맞는 질병 예방, 진단 및 치료 방법 개발  
\* 2018년 7월 창업, GC녹십자에서 20억 규모의 Seed 투자, 2019년 실리콘밸리 본사 설립, 베트남지사 설립 추진
  - 질병 진단 및 예측을 위한 AI 서비스 개발을 위해서는 의료 도메인지식과 SW 지식을 보유한 개발자가 필요

<표17> 의료분야 AI와 SW개발자 필요 역량

구분		SW개발자					
		AI(인공지능)		SW제품 및 서비스 개발 운영(의료 개발자)			
		AI엔지니어	데이터사이언티스트	기획	데이터 엔지니어	어플리케이션 개발자	
						프론트엔드	백엔드
의료도메인지식	Medical Image	-Medical image AI 모델 연구 및 개발 -AI 프로젝트 제안 및 관리	-AI 모델링을 위한 유전체, 약물 관련 데이터 피싱, 시각화	-의료 플랫폼 서비스 기획 -Front 서비스 기획 및 운영의 개선 업무	-데이터 분석/모델링을 위한 데이터 엔지니어링 -데이터 파이프라인 설계, 개발 운영	-웹 서비스 기반 의료 SW의 사용자 인터페이스 설계 -알고리즘을 효율적으로 제품화할 수 있는 웹 프론트엔드 구현	-웹 서비스 기반 의료 AI SW의 서버 기능 구현 및 운영 -클라우드 환경에서 DB 및 웹서비스 운영, 데이터 모델의 상용화 서비스 구현
	Bioinformatics	-신약개발 및 재조합 AI 모델 연구 및 개발 -문헌논문, 임상정보 등에서 지식추출 및 추론	-AI를 이용하여 유전체, 약물 관련 데이터에서 인사이트 도출 -아래 종류의 데이터 통합 및 관리	-웹 / 모바일 UI/UX 전략 수립 및 기획	-데이터웨어하우스 설계, 개발 운영	-의료 AI 분야 최신 연구결과 수확적 모델 요구사항을 이해하고 서비스기능 구현	-빠르고 안정적인 대용량 데이터 서비스를 구현하고 운영

- (국내 AI분야 중소기업) 국내 중소기업은 자연어 처리(텍스트, 음성)기반의 AI 응용서비스를 개발하여 타 기업 시스템을 구축하거나 서비스를 공급

<표18> 국내 AI분야 주요 중소기업과 오픈소스 활용

기업명	AI 기술분야	솔루션(서비스)	오픈소스 활용 혹은 인력채용
와이즈넷	검색, 자연어, 음성인식	AI 자동응대	-대용량 분산처리기술(Spark, ElasticSearch)
NHN다이렉스트	검색, 자연어 처리, 음성인식	대화형 채팅 및 자동상담	-하둡활용 빅데이터 협업필터용 분산처리 모듈
마인즈랩	자연어처리, 음성인식	마음에이아이(maum.ai)	-다수 화자 음성분리 필터 개발
코난테크놀로지	자연어 처리, 음성인식	QA 시스템	-하둡과 스파크 기반 빅데이터
솔리드웨어	머신러닝 기반 예측 모델링 툴	인공지능기반 데이터 분석 시스템 다빈치랩스	-미확인
뷰노	딥러닝	의료영상 및 사진판독의 진단 뷰노넷 딥러닝 엔진	-딥러닝 플랫폼 자체개발
스캐터랩	감성 자연어처리 딥러닝 기술	사람과 일상적 대화 나누는 AI 핑퐁	-오픈소스 커뮤니티의 활발한 유저 혹은 오픈소스 개발 경력

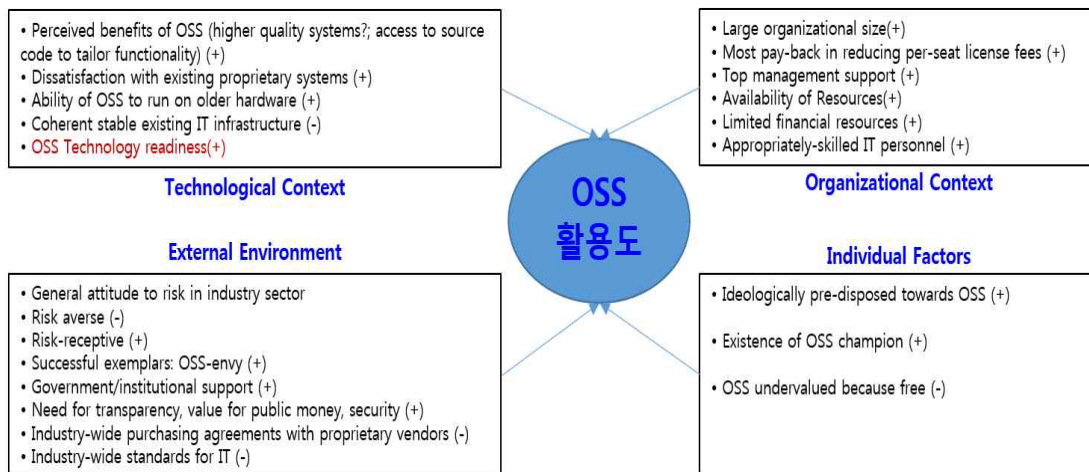
\* 상기 자료는 언론 등에 발표된 자료를 기반으로 해당 기업의 인력 채용 공고를 활용하여 작성

## 5

## 오픈소스 활용역량

- (오픈소스 활용도의 선행요인) 오픈소스를 활용하는 능력에 영향을 주는 주요 요인은 TOEI(기술·조직·환경·개인) 차원으로 구분<sup>44)</sup>

<그림16> 오픈소스 활용능력에 미치는 선행요인



- (기업 수준 오픈소스 역량) 기업이 오픈소스를 전사적 관점에서 활용하는 역량 수준은 오픈소스 적용능력 수준, 수용 단계, 거버넌스 역량 단계 등으로 진단이 가능
- (오픈소스 활용능력 수준) 기업이 내·외부적으로 오픈소스 기반으로 혁신을 달성하기 위한 오픈소스 역량수준(Level of open source capability)<sup>45)</sup>은 6단계로 구분



<표19> 조직의 오픈소스 역량 수준

단계	오픈소스 관련 세부 역량 수준
1단계	-가장 낮은 수준으로 내부 소스라고 함 -오픈소스에서 아이디어를 얻었지만 외부 코드를 사용하거나 내부코드를 외부로 공유 않음
2단계	-소비자(사용자)로서 외부 코드를 가져와 사용하는 수준
3단계	-오픈소스를 다른 오픈소스와 결합하거나 독점SW와 결합하여 고객에게 제공
4단계	-자사가 주도적으로 오픈소스 프로젝트를 수행 -오픈소스를 공개하지만 실제로는 외부 커뮤니티가 활발하지 않음
5단계	-코드 기여, 패치 혹은 후원 등의 방법으로 외부 오픈소스 프로젝트에 참여하지만 프로젝트와의 관계는 일방향이거나 밀접하지 않음 -외부 프로젝트 참여는 적극적이지 않지만 패치 혹은 다른 지원 등의 기여 활동
6단계	-외부 오픈소스 프로젝트에 공동 리더로서 참가 -단지 오픈소스 프로젝트에 패치 등을 지원하는 기여자 역할이 아니고 오픈소스 프로젝트와 자사가 필요한 것을 적극 공유 -오픈소스 프로젝트 방향을 잡고, 개발자에게 오픈소스 프로젝트에 대해 기여할 시간을 지원

○ (오픈소스 수용단계)는 인지, 관심, 평가시도(evaluation·trial), 몰입(commitment), 제한적 전개(limited deployment), 전반적 전개(general) 단계로 발전<sup>46)</sup>

- 현재 대부분 우리기업은 일부 대기업을 제외하고 대부분 1단계 수준

<표20> 오픈소스 수용단계

단계	수용 수준
1단계	-핵심 의사결정권자가 오픈소스를 인지하고 있으나 활용하지 않고 있음
2단계	-조직 내부적으로 오픈소스를 학습하는 등 적극적으로 노력하는 단계
3단계	-조직이 특정 오픈소스 제품을 구입하여 평가 또는 TRIAL을 시작
4단계	-조직이 특정 오픈소스 제품을 SW생산 프로젝트에 사용하기로 결정
5단계	-조직의 정보시스템에 오픈소스의 일부 또는 부분적으로 활용하는 단계
6단계	-조직 정보시스템 및 사업 전체에 오픈소스 사용을 확대해 나가는 단계

○ (오픈소스 거버넌스 역량 수준) 전사차원에서 기업의 오픈소스 도입-> 활용 -> 관리 -> 폐기 등 전반적인 오픈소스 활용 라이프사이클에 대한 통합적인 관리능력 수준

- 순차적(정책수립 -> 획득 -> 적용 -> 운영 및 유지)과 비 순차적(모니터링, 교육, 컴플라이언스) 세부 단계별 능력 확충이 필요

\* IT서비스 및 상용SW사, 인터넷 사업자 등 SW사업자 모두는 SW개발 기획 단계부터 제품화까지 모든 단계에서 체계적인 오픈소스 컴플라이언스(준수) 체계가 필요

<표21> 기업의 오픈소스 거버넌스 역량 수준

활동단계 및 활동요소	설명
정책 수립	정책수립 -목표와 전략에 따라 반드시 지켜야 할 규정과 지침을 수립함
	컨설팅 -오픈소스 적용과 전략수립을 위한 자문서비스를 제공함
	조직구성 -효율적인 인력구성과 역할과 책임에 따른 운영방안을 제시함
획득	요구분석 -고객 또는 사용자의 고민, 요구사항 등을 분석함
	조사 -새로운 오픈소스 또는 특정분야에 적합한 오픈소스를 찾음
	분석 -오픈소스의 속성을 구분하고 상태나 수준을 정리함

	평가	-각 속성에 가중치를 부여하고 평가모델을 적용하여 채점함
적용	계약	-오픈소스의 도입 및 활용, 배포에 대한 일련의 책임과 의무에 대해 조건과 규정을 체결
	설계	-요구 분석결과에 따라 기능과 사양을 미리 구성함
	개발	-오픈소스 프로그램을 변경 및 결합함
	패키징	-오픈소스의 설치가 편리하도록 단일프로그램으로 묶음
	시험	-요구수준에 맞는지 품질과 성능을 확인함
	배포	-오픈소스를 저장매체, 웹사이트, 장비 등을 통해 전달함
운영 및 유지	설치	-오픈소스를 운영할 수 있는 장비에 탑재함
	운영	-오픈소스를 실행시켜 정상적인 상태로 지속적으로 가동시킴
	유지보수	-최상의 운영 상태를 유지하도록 제반작업을 수행함
	기술지원	-추가적인 요구사항을 반영이나 문제해결등 공학적인 오픈소스 서비스를 제공함
	커뮤니티	-소스코드 기여, 재정적 지원, 활동교류, 참여방법을 제시함
관리 및 개선	컴플라이언스	-라이선스 의무사항 준수 및 법적문제를 해결함
	교육	-오픈소스의 도입, 활용, 배포에 대한 이해력을 높이기 위해 지식전달 및 스킬향상
	모니터링	-오픈소스 적용 이후의 상황을 파악하고 피드백을 수렴함

□ (개발자 수준 오픈소스 활용역량) 개발자가 오픈소스를 활용하여 새로운 SW를 생산할 수 있는 역량은 SW생산 기술역량, 오픈소스 커뮤니티 참여역량<sup>47)</sup>으로 구분

○ (SW 기술능력 수준) 단순 기술능력, 적용능력, 간단한 SW생산 능력, SW재설계 능력, 숙련된 SW 생산능력 수준으로 발전

- SW재설계 능력(4단계)은 오픈소스로 생산된 SW제품과 생산과정 자체를 수정할 수 있는 수준, 숙련된 SW생산 능력(5단계)는 오픈소스로 생산된 제품보다 기능적으로 더 차별화된 신제품을 개발할 수 있는 수준

\* 현재 우리는 간단한 SW생산(3단계) 혹은 일부 분야에서 SW 재설계 능력(2단계) 수준

- SW산업에서 원천 기술이 부족한 우리는 단기적으로 SW재설계 능력자를 육성하고 ,장기적으로 숙련된 고급 SW 개발자 양성이 시급

<표22> SW 기술능력 단계와 오픈소스<sup>48)</sup>

SW 기술능력 수준	세부 SW 기술능력 수준	단계	SW산업 발전 수준별 국가분류 <sup>49)</sup>		오픈소스를 통한 기술 능력 발전	
			SW수출 국가분류	대상 국가	순차적 (단기적) 발전 가능성	도약 (장기적) 발전 가능성
숙련된 SW 생산능력 (Skilled SW Production)	-글로벌 생산혁신(지역/글로벌 사용자 요구에 맞게 새로운 프로그램 개발) -주요 프로세스 변혁 : SW생산 프로세스의 재설계 -프로세스 혁신 : 완전히 새로운 SW 생산 프로세스 설계	①	선발 SW 수출국	-OECD국가(미국, 캐나다, 영국, 독일, 프랑스, 벨기에, 네덜란드, 핀란드, 일본, 스위스, 호주), 아일랜드, 이스라엘, 인도	-	1단계 (숙련된 SW 생산 능력)
SW 재설계 능력(SW Redesign)	-지역/글로벌 사용자 요구에 맞게 프로그램 재설계 -소규모 프로세스 변혁 : SW생산 프로세스 변형	②	체제전환 SW수출국	-러시아, 중국	1단계 (숙련된 SW생산 능력)	
간단한 SW생산 능력 (Simple SW Production)	-사용자 인터페이스 작성 -응용SW사이에 데이터를 교환할 수 있는 프로그램 작성 -소규모 유틸리티 프로그램 작성	③	신흥SW 수출국	-한국, 브라질, 멕시코, 칠레, 아르헨티나, 필리핀, 태국, 스리랑카, 파키스탄, 루마니아, 폴	2단계 (SW 재설계)	

	-사용자의 요구에 맞게 SW 변형			란드, 체코, 헝가리, 남아공	능력)	
<b>적용 능력</b> (Adaptation without Production)	-패키지 기반 응용SW 개발	④	후발SW 수출국	-쿠바, 요르단, 이집트, 방글라데시, 베트남, 인도, 이란 등	3단계 (간단한 SW생산 능력)	
<b>단순 기술 능력</b> (non-producti on technical capabilities)	-DB 데이터 입력, SW 설치 및 문제해결 -메뉴 시스템 사용 -SW 사용법 교육 가능	⑤	비경쟁 국가	-대부분 저개발국으로 아프리카(잠비아, 나이지리아, 모잠비크), 남미(볼리비아, 파라과이), 아시아(라오스, 아프간, 시리아)	4단계 (적용능력)	

- **(오픈소스 프로젝트 개발자 역량)** 글로벌 오픈소스 커뮤니티에 활동하는 개발자는 사용자, 공헌자, 커미터급으로 구분하고 각자의 능력과 권한에서 차이를 보임
  - 오픈소스 프로젝트가 성공하기 위해서는 프로젝트 관리, 소스코드 개발, 문서화(매뉴얼/번역), 커뮤니티 관리, 커미터와 컨트리뷰터 모집, 꾸준한 기술공유 및 홍보가 필요
  - 현재 우리나라에서 구축된 커뮤니티는 총 248개(사용자 : 214개, 개발자 : 34개)가 활동 중, 국내 거주 오픈소스 기여자는 약 10,950명 수준<sup>50)</sup>

<표23> 오픈소스 커뮤니티 참여역량 수준

사용자	공헌자(contributor)	커미터(commiter)	Maintainer
오픈소스 프로젝트 결과물 사용자	-컨트리뷰션을 하는 모든 사람, 컨트리뷰션은 오픈소스 프로젝트에 참여하고 기여하는 모든 활동 -버그 제보와 수정(30%), 문서화(28%), 새로운 기능 추가(18%), Refactoring(9%), update version(7%), improve error(3%), improve resource usage(2%), 테스트 사례 추가(1%)	-컨트리뷰션 내용을 리뷰하고 프로젝트에 반영할지 결정하는 사람 -Git과 같은 중앙 저장소에 변경 사항을 커밋	-프로젝트 방향 설정, 관리하는 사람. 보통 커미터 중 일부가 맡음

- 또는 **오픈소스 프로젝트 능력**은 소스코드 결함을 발견(detect)하고 제거(removal)하는 능력과 기능을 향상(functionality-enhancement)시키는 능력으로 측정<sup>51)</sup>

\* 오픈소스 프로젝트 성과의 선행요인은 지식공유의 질과 량, 제품특성, 팀 구조, 개발자의 수, 의사소통의 질, 팀의 노력, 개발자 경험과 배포 속도 등

- **(딥러닝 분야 AI 개발자 역량)** 통계에 대한 기본적 이해와 AI 알고리즘의 생성, 개선, 그리고 궁극적인 데이터 산출 과정에 대한 이해는 필수적
  - **초급(딥러닝 AI 플랫폼 활용 능력)** 오픈소스인 Pytorch, Tensorflow 등 하나 이상의 딥러닝 플랫폼 사용과 알고리즘 구현 능력을 보유
  - **중급(AI 플랫폼 재설계 능력)** 오픈소스로 배포된 딥러닝 플랫폼에 새로운 기능을 추가하거나 다른 오픈소스 모듈을 결합하여 서비스 개발 가능
  - **고급(AI 플랫폼 차별화 능력)** 오픈소스로 공개된 딥러닝 플랫폼과 차별화된 다른 플랫폼을 설계하고 개발할 수 있는 능력

<표24> AI개발자 오픈소스 역량 수준

	딥러닝 분야 요구 능력	Computer vision 분야 요구 능력
초급	-Pytorch, Tensorflow, Keras 등 하나 이상의 딥러닝 플랫폼을 다운하여 설치하고 데이터 관리, 모델도출, 예측결과 도출 능력 -Pytorch, Tensorflow, Keras 등 하나 이상의 딥러닝 프레임워크 사용 능력 -Python/C++/Java 등 이용 알고리즘 구현 능력 -Machine Learning 알고리즘을 활용 능력	-Computer vision 오픈소스를 깃허브 등에서 찾아 설치하고 딥러닝 플랫폼과 연계,활용할 수 있는 능력 -Python/C++/Java 등을 이용한 알고리즘 구현 능력 -주요 Computer vision 알고리즘의 원리, 필요성, 한계에 대한 지식 -Computer vision 알고리즘을 활용할 수 있는 능력
중급	-오픈소스를 활용하여 재설계 할 수 있는 능력 -오픈소스로 배포된 딥러닝 플랫폼에 새로운 기능을 추가 혹은 결합할 수 있는 능력 -Machine Learning 관련 연구, 업무 및 프로젝트 경험 -딥러닝 오픈소스 프로젝트에서 컨트리뷰션 할 수 있는 능력 -AI 논문을 읽고 내용을 파악 가능	-오픈소스로 배포된 Computer vision 플랫폼에 새로운 기능을 추가 혹은 결합할 수 있는 능력 -Computer vision 관련 연구, 업무 및 프로젝트 경험 -실제 세계의 데이터를 활용하여 주도적으로 문제를 풀어본 경험, 개발한 AI 모델을 이용하여 실제 서비스를 해본 경험, Computer vision 관련 오픈소스 프로젝트에서 컨트리뷰션할 수 있는 능력
고급	-오픈소스로 공개된 딥러닝 플랫폼과 차별화된 플랫폼을 설계하고 개발할 수 있는 능력 -딥러닝 관련 오픈소스 프로젝트에 커밋할 수 있는 능력 -주요 학회 논문 게재 가능	-오픈소스로 공개된 Computer vision 관련 플랫폼과 차별화된 다른 플랫폼을 설계하고 개발할 수 있는 능력 -주요 학회 논문 게재

※ 상기 등급별 AI 역량은 AI인력을 채용하는 공고를 참조하여 수정한 것임

□ (오픈소스 역량과 미국경제) 국가가 성장하기 위해 지식 자산에 점점 더 의존함에 따라 **오픈소스 활용은 엄청난 경제적 결과를 초래**<sup>52)</sup>

○ 미국의 경우 오픈소스 개발 작업은 경제 전반에 널리 퍼져 있으며, 숙련된 노동력을 구축하고 **전체 산업 평균보다 훨씬 높은 임금을 제공**

- 오픈소스 인력은 미국의 **고 부가가치 고용 성장**에 크게 기여, 오픈소스 관련 업무가 널리 보급되고, 산업이 오픈소스의 혜택에 노출됨에 따라 SW 개발 개방형 혁신 프로세스가 학습 및 모방 과정을 통해 전파되고 발전

- **오픈소스 인력은 미국 경제의 모든 산업 부문을 관통하고 지속 가능한 성장을 보장**

\* 오픈소스 관련 SW 개발 일자리 수는 2008년에 801,590 ~ 1,201,391건 수준으로 예측

- 미국 정부는 새로운 성장 동력 및 일자리 창출을 위해 **다양한 수준과 전문 분야의 숙련된 오픈소스 인력을 충분히 공급해야** 한다고 주장

○ 또한 리눅스 재단이 조사한 **2018년 오픈소스 일자리 보고서**<sup>53)</sup>에 따르면

- 미국기업이 SW 인력을 채용할 때 83%가 오픈 소스 인재를 채용

- 고용자의 87%가 오픈소스 인력확보가 어렵고, 오픈소스 전문가 중 55%는 새로운 직장을 찾기가 쉬우며, 87%는 오픈소스가 자신의 경력을 높임

\* 기술 전문가들의 전반적인 실업률이 2018년(1.9%), 2017년(3%)

- 고용자는 최근 외부 컨설턴트 고용을 중단하고 **기존 직원을 새로운 오픈소스 기술에 대해 교육시키고 인증 취득을 지원하는 것**에 집중

### III. 우리의 현실

◇ 일부 대기업을 제외하고 중소기업은 오픈소스 중요성 인식과 활용 능력 등에서 선진국과 비교해서 현저히 낮은 수준

#### □ 국내기업의 오픈소스 활용 동향

- (시장 규모) 공급자 시장규모는 2,337억원('18년), 전년 대비 23.7% 성장, 전 세계 시장 대비 1.5% 수준, 新 정보시스템 환경에서 잠재적 시장 규모(투자 규모를 기반 추정)는 7.4조원('18년)으로 추정
  - \* 공급자 시장규모에서 솔루션별로 OS/Server(18.1%), Web/WAS(16.9%), DB/DBMS(13.6%) 순, 전통적으로 시장을 주도했던 3대 솔루션 비율이 48.6%를 차지
  - \* 잠재적 시장규모에서 Bigdata, IoT, Cloud, AI 등 부문은 20.8%('17년)에서 26.9%('18년)으로 증가
- (활용 수준) 국내 전체기업들의 오픈소스 활용률은 48.5%('18년)로 전년대비 4.2%p 증가, 전체 산업을 ICT/Non-ICT로 구분할 때, ICT기업(50.6%), Non-ICT기업(47.5%)
  - 4차 산업혁명 관련 기술을 구현할 때, 오픈소스 활용 기업(42.4%), 향후 활용 예정(51.2%), 기술별로 빅데이터(53.6%), 클라우드(42.4%), IoT(31.2%), AI(24%) 순
    - \* ICT 업종별로 IT서비스 기업(55.8%), 패키지 SW(25.5%), 임베디드 SW(9.7%), 클라우드(8.9%) 순, NON-ICT 산업별로 금융(62.1%), 공공(48.3%) 서비스업(49.3%) 순
  - 활용분야는 정보시스템 구축(47.2%), 유지보수 및 기술지원(27.8%), 오픈소스 기반 제품·서비스 개발(21%), HW제조(13.3%) 순
    - \* ICT기업은 제품·서비스 개발(47.4%), 정보시스템 구축(35.5%), NON-ICT기업은 제품·서비스 개발(9.3%), 정보시스템 구축(52.3%) 순으로 차이를 보임
- (인력·커뮤니티·라이선스 등) 오픈소스 인력은 약 11천명으로 전세계 대비 0.04% 수준, 커미터급 최고급 개발자는 780명 수준
  - 약 370개 커뮤니티 중에서 사용자(322개, 86%), 개발자(52개, 14%)로 조사(NIPA, '17)
    - \* 기술 분야별로 개발환경 분야(29%), 빅데이터, 운영체제, 웹분야(11%) 등으로 분포

<표25> 오픈소스 커뮤니티 및 인력현황 비교

구분	국외	국내	비고
커뮤니티 수	약 167,000개	약 370여개	Github/NIPA 정리(0.2% 수준)
개발자 수 (커미터 수)	약 28,000,000명 (약 40,000명)	약 15,323명 (약 780여명)	Github기준(0.05% 수준)
글로벌 프로젝트 수	약 800건	약 10건	1.2% 수준

- 전체 기업 중에서 커뮤니티 구축·운영 수준(5.4%), 운영 목적은 판매 채널(46.4%), 네트워크 활용(25%), SW개선 활동(17.9%) 순
  - \* 커뮤니티를 후원하는 기업(2.8%), 커뮤니티에 참여하는 기업(6.2%) 수준, 후원 목적은 고객과의 파트너십 확대(57.1%), 품질 개선(28.6%), 참여목적은 오픈소스 관련 지식 학습, 업무에 적용할 오픈소스 확보 등
  - \* 해외기업(약 65%)가 오픈소스 프로젝트에 참여(North Bridge & Blackduck, 2017), 국내 기업(7%) 수준

- 전체 기업 중에서 거버넌스 체계 구축(9.7%), 미 구축 (90.3%), ICT 업종별로 IT서비스(11.8%), 클라우드(27.8%), 패키지(2%), 임베디드(0%) 순

\* 거버넌스 구축관련 애로사항은 인력 확보(74.0%), 예산 부족(36%), 운영 노하우 부족(36.0%) 등

- **(라이선스 인식 및 적용현황)** 오픈소스 인식에 대해 모름(66.7%), 라이선스 문제 발생 경험은 없음(88.7%), 있음(11.3%) 수준

- 문제 경험은 복제 및 수정권한 허용(5.4%), 변경사항 고지(3%), 라이선스 종료 의무(2.4%), GPL활용 시 소스코드 공개요구(1.6%) 등

\* 국내 H사는 '13년 PDF 문서변환 오픈소스인 아티팩스社 GhostScript를 활용, 라이선스 위반으로 美법원 피소('16.2), 205만불 지급('17.12 종료)  
\* 대응 체계는 자체 관리(42.9%), 체계 없음(23.2%), 기관 신고 및 도움 요청(34%) 순

- **(오픈소스 교육현황)** 국내 대학과 국내 기업은 아직까지 체계적인 오픈소스 교육 과정 개발과 교육이 진행되고 있지 않는 상황

- **(국내 주요 대학)** 오픈소스의 중요성을 깨닫고, 관련 기술과 실무 수업을 도입하여 교육 환경을 변화시키는 중이다, 여전히 중요성에 비해 관련 교육이 충분하고 체계적이지 않음

<표26> 국내외의 OSS 교육 현황

구분	과목명/프로그램	강의 내용(구성)
CMU (2016)	Open Source Software Development Master Program	1.The Business of Open Source and Linux/ 2.Linux - Heart of the Operating System/ 3.Communities and Organizations/ 4.Distributions - Completing Linux/ 5.Human Resources - Getting Top Talent/ 6.The Corporate Bazaar/ 7.Business Models - Making Money/ 8.Integrating Open Source Into your Business/ 9.Open Source - Navigating the Legal Path to Freedom/ 10.The Cost of Linux and Open Source/ 11.Using Linux and Open Source/ 12.Integrating Open Source Into your Business
MIT (2015)	Open Source Software Project Lab/Department Program	1.Intro-Studio/ 2.Git-Studio/ 3.Studio-Testing/ 4.Studio- Debugging/ 5.Presentations-Presentations/ 6.Studio-Development process/ 7.Studio-TBD/ 8.Studio-Presentations/ 9.Presentations- Performance Engineering/ 10.Studio-Studio/ 11.Project management and collaboration/ 12.Open source business models/ 13.Final Presentations(1,2,3)
성균관대학교 (2016)	오픈소스SW 개론/학부과정	1.오픈소스SW 정의와 개념/ 2.OSS역사와 초기 프로젝트/ 3.주요 단체 및 OSS 프로젝트/ 4.분야별 OSS 종류 및 활용사례/ 5.OSS 라이선스의 특징 및 종류/ 6.OSS 활용 및 프로젝트 수행방법/ 7.OSS 비즈니스 모델 및 전략/ 8.Git/GitHub 다가가기(1)/ 9.Git/GitHub 다가가기(2)/ 10.GitHub 활용방법/ 11.주요 OSS 커뮤니티 현황/ 12.OSS 활동방법/ 13.팀프로젝트1/ 14.팀프로젝트2
한양대학교 (2015)	오픈소스SW 개론/학부과정	1.오픈소스SW 개요, 오픈소스SW 전반 대한 소개/ 2.산업에 적용된 오픈소스 프로젝트 소개1/ 3.산업에 적용된 오픈소스 프로젝트 소개2/ 4.산업에 적용된 오픈소스 프로젝트 소개3/ 5.오픈소스SW 바르게 사용하기, 저작권/라이선스의 의미/ 6.Open source startup with github(git & github)/ 7.오픈소스 활동가의 경험 공유/ 8.오픈소스 프로그래밍 실습/ 9.오픈소스 프로그래밍 실습/ 10.오픈소스 프로그래밍 실습/ 11.오픈소스 프로그래밍 실습 (*모든 회차별 외부 강사 특강 형식으로 진행됨)
제주대학교 (2016)	오픈소스 개발 방법론/ 학부과정(카카오 트랙)	1.강의소개/ 2.오픈소스 SW 소개/ 3.오픈소스 라이선스 소개/ 4.기업에서 OSS 활용 사례/ 5.소셜코딩 실습1, Github/ 6.소셜코딩 실습2, Markdown, PR/ 7.오픈소스 개발 방법론/ 8.기업에서 OSS 활용 사례/ 9.조별프로젝트/ 10.파이썬 소개 및 실습/ 11.파이썬 실습/ 12.조별 프로젝트/ 13.오픈소스 커뮤니티 참여하기/ 14.오픈소스 커뮤니티 및 참여 활동에 대한 보고
경희대학교 (2016)	오픈소스 SW개발/학부과정	1.Orientation/ 2.Open Source Software(OSS)/ 3.Open Source Software(OSS)/ 4.OSS 개발 환경 및 형상 관리 (Eclipse, Build Tools)/ 5.OSS 개발 환경 및 형상 관리(Subversion, Git)/ 6.프로젝트 발표/ 7.OSS Contribution/ 8.OSS Committer 초청/ 9.OSS Committer 초청/ 10.프로젝트 발표/ 11.Framework/ 12.COMPUTING Platform/ 13.Distributed Computing(Docker, OpenStack, CloudStack)/ 14.프로젝트 발표

- **(국내 기업의 오픈소스 교육 현황)** 국내 기업들은 오픈소스 교육관련 애로사항으로 강사 부족(63.4%), 낮은 참여도(34.1%), 경영자의 낮은 인식(21.1%), 교육과정 구성 어려움(20.3%) 등



## IV. 시사점

- (오픈소스와 라이선스) 오픈소스는 SW의 소스코드가 특정 라이선스로 공개되어 수정·복제·사용·재배포 등이 자유로운 SW를 지칭
  - 기업은 모듈화, 재 사용성이 높은 오픈소스에 새로운 기능을 추가하여 차별화된 SW를 생산할 수 있으므로 개발기간 단축, 비용 절감, 그리고 최신 SW기술 학습에 활용할 수 있는 등 **다양한 이점**을 제공
  - 하지만 사용자는 해당 SW를 공개한 커뮤니티(저작권자)가 규정한 **라이선스**를 반드시 준수해야 할 의무가 있음
    - 이것을 위반하면 해당 오픈소스에 대한 이용권리가 박탈되고 제품화한 경우에는 판매 중단 등으로 기업 이미지가 훼손되어 치명적 손실을 초래
- (오픈소스 중요성) 4차 산업혁명 구현을 위해 오픈소스 **중요성이 재조명**
  - 최근 경제·사회 패러다임이 경쟁(협업), 폐쇄(개방), 소유(공유), 파이프라인(플랫폼) 경제로 변화하는 중심에는 오픈소스를 활용한 **혁명적 SW 개발방식** 출현에 기인
    - 국가혁신체제 관점에서도 오픈소스는 지식전달 체계인 사회 인프라 확충과 구성원의 학습과 신기술 개발의지를 고취시켜주는 개방형 혁신의 핵심 수단(촉매제)로 작동
  - 4차산업혁명은 舊 정보시스템 환경과 완전히 다른 **새로운 (기술) 인프라**를 이용하여 모든 것이 플랫폼과 연결(Connectivity)되고 지능화(Intelligence)된 새로운 시스템을 구축·운영하는 것으로
    - 新시스템 구축은 상당한 투자와 AI, 블록체인, 클라우드 등 다양한 분야의 새로운 기술 역량을 요구
      - \* (AI) 다양한 형태의 데이터를 수집하여 학습·추론으로 분석·판단, (5G, IoT, 블록체인) 4G보다 20배 이상인 5G와 사물(IoT)·조직(블록체인)을 연결, (클라우드) 컴퓨팅 자원(HW, 응용서비스 등) 구독 서비스(IaaS, PaaS, SaaS 등) 기술
    - 이전과 다른 특이한 현상은 기업이 SW를 개발하면 소스코드를 공개하여 사용자가 시스템 구축에 활용하고 새로운 SW개발에 참여토록 유도하여 자사 중심의 생태계를 구축
      - \* 예를 들면, 인공지능(AI)은 딥러닝, 자연어 처리 분야 등 세부 신기술 개발이 오픈소스 프로젝트로 추진, 오픈소스 딥러닝 플랫폼은 Tensorflow 등 20여종
      - \* 우버는 승차 공유 플랫폼에 오픈소스를 활용하여 ML as-a-service인 Michelangelo를 구현, 특히 음식 배달 플랫폼에는 HDFS, Spark, Samza, Cassandra, MLlib, XGBoost, TensorFlow 등 오픈소스를 활용

- 또한 4차산업혁명 구현을 위한 핵심 기술에서 **오픈소스의 중요성**은 이전의 舊 정보시스템 구축과 비교할 때 **완전히 다른 차원**을 보임
  - 과거에는 일부 완제품 형태의 오픈소스 제품(LAMP)을 사용하는 데 머물러 오픈소스 자체가 산업으로 성장하지 못하는 한계를 보였으나
  - 최근에는 인식(자연어·음성·이미지 등)과 딥러닝 등 **AI분야 SW가 오픈소스로 공개·공유**되어 AI 자체 기술도 비약적으로 발전될 뿐만 아니라, 제조·농업·금융 등 다른 연관 산업의 경쟁력을 강화시키는 핵심 성장 동력으로 자리매김

□ (시사점) 성공적인 4차 산업혁명 구현의 주역으로 활동할“**오픈소스를 활용할 수 있는 등급별 인력 10만 명 양성**”이 절실한 시점

- 등급별 인력은 아이디어를 구현할 오픈소스를 찾아서 自社 환경에 적용할 수 있는 수준과 오픈소스를 기반으로 기능을 추가하여 새로운 SW를 생산(차별화)할 수 있는 수준 등
- 현재 우리원은 오픈소스 활용능력 배양을 위해 **軍 장병(년간 1,000명), 대학생(1,000명), 커뮤니티 지원(년간 40개), 오픈 프론티어 지원(년간 50명)** 등과 오픈소스를 활용한 기술개발, 창업지원, 국제협력 등 맞춤형 지원 사업을 추진 중으로
  - 각 경제 주체(대학, 기업 등)는 대학생, 재직자 등을 대상으로 체계적인 오픈소스 교육 과정을 수립하여 실행할 필요
- “**4차 산업혁명을 누가 어떻게 구현할 것인가 ?**” 라는 질문에 대해
  - 현재 우리 조직에는 오픈소스를 활용할 수 인력이 어느 수준인지를 인지하고, 어떻게 단기간에 양성하여 활용할 것인가? 에 대해 이해하고 고민하는 사람이 증가하길 기대

## <참고 문헌>

- 1) Cusumano, M. A. (2008). The changing software business: Moving from products to services. *Computer*, 41(1), 20-27.
- 2) 송위진. (2002). 한국형 오픈소스 소프트웨어 기술개발 전략. 정책자료, 1-49.
- 3) Leveraging open source licenses and open source communities in hybrid commercial open source business models Author. Karl Michael Popp.
- 4) Hippel, E. V., & Krogh, G. V. (2003). Open source software and the “private-collective” innovation model: Issues for organization science. *Organization science*, 14(2), 209-223.
- 5) 김형채. (2018). 오픈소스 커뮤니티의 구축 및 운영. *한국통신학회지 (정보와 통신)*, 35(5), 83-89.에서 인용
- 6) <https://www.olis.or.kr/>, 오픈소스SW라이선스 종합정보시스템
- 7) Karsten Reincke, Greg Sharpe.(2013). "오픈소스 라이선스 해설-오픈소스 라이선스준수 방법, Deutsche Telekom AG, 2013. 8. 1 (번역:정보통신산업진흥원 공개 SW 팀)
- 8) Lerner, J., Tirole, J., 2005. The scope of OpenSource licenses. *Journal of Law Economics and Organization* 21 (1), 20-56.
- 9) Koski, H., 2005. OSS production and licensing strategies of the software firms. *Review of Economic Research on Copyright Issues* (2), 111-125.
- 10) Philippe Heinze/Jens Burke, Open-Source-Compliance, CCZ 2017, 56.
- 11) <https://tom.preston-werner.com/2011/11/22/open-source-everything.html>
- 12) [https://en.wikipedia.org/wiki/Comparison\\_of\\_source-code-hosting\\_facilities](https://en.wikipedia.org/wiki/Comparison_of_source-code-hosting_facilities)
- 13) Piva, E., Rentocchini, F., & R Lamastra, C. (2010). Is Open Source about innovation? How interactions with the Open Source community impact on the innovative performances of entrepreneurial ventures.
- 14) West, J., & O'mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry and innovation*, 15(2), 145-168.
- 15) Crowston, K., & Howison, J. (2006). Assessing the health of open source communities. *Computer*, 39(5), 89-91.
- 16) Stephanos Androutsellis-Theotokis 외 3 인,(2014). OpenSource Software : A Survey from 10,000 Feet(10,000 피트에서 바라본 오픈소스 소프트웨어, 한티미디어, 김종배 옮김), 2014, pp. 114.
- 17) 송위진. (2002). 오픈소스 소프트웨어의 기술혁신 특성: 리뷰. *기술혁신학회지*, 5(2), 212-227.에서 인용
- 18) Leveraging open source licenses and open source communities in hybrid commercial open source business models Author. Karl Michael Popp.
- 19) 마샤 밴 엘스타인외, 이현경 번역 (2017), 플랫폼 레볼루션(Platform Revolution)
- 20) Porter, M. E., & Heppelmann, J. E. (2015). How smart, connected products are transforming companies. *Harvard Business Review*, 93(10), 96-114.
- 21) 김상윤. (2019), 기업은 어떻게 AI를 도입하는가?, POSRI 이슈 리포트
- 22) 이상목 (2019), 지역혁신체제의 성공을 위한 오픈소스 활용 방안에 대한 사례 연구
- 23) Harison, E., & Koski, H. (2010). Applying open innovation in business strategies: Evidence from Finnish software firms. *Research Policy*, 39(3), 351-359.
- 24) Fernandez, S., & Svensson, R. B. (2017, August). A survey of practitioners use of open innovation. In

- 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 305-312). IEEE.
- 25) 송위진 (2002) 오픈소스 소프트웨어의 기술혁신 특성 : 리뷰
- 26) 마샬 벤 엘스타인외, 이현경 번역 (2017), 플랫폼 레볼루션(Platform Revolution)
- 27) Porter, M. E., & Heppelmann, J. E. (2015). How smart, connected products are transforming companies. *Harvard Business Review*, 93(10), 96-114. Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard business review*, 92(11), 64-88.
- 28) Gartner, Delivering Digital Business Value with Integrated IoT Solutions, Aug. 2016. IOT Architecture & Method, Copyright © 2016. Dr. June Sung Park. All rights reserved. 47|에서 인용
- 29) Minsuk-Lee, OpenSource Software and GitHub, Jun. 26. 2015.
- 30) The 2018 open source jobs report, the linux foundation & dice
- 31) 2019 Synopsys Open Source Security and Risk Analysis, Percentage of audited codebases containing open source components by industry (2018 & 2017)
- 32) [https://medium.com/@Francesco\\_AI/ai-knowledge-map-how-to-classify-ai-technologies-6c073b969020](https://medium.com/@Francesco_AI/ai-knowledge-map-how-to-classify-ai-technologies-6c073b969020)
- 33) <https://audiotech.com/trends-magazine/open-source-artificial-intelligence/>
- 34) <https://medium.com/computer-vision-technology-drives-the-future/top-11-tools-to-build-a-computer-vision-application-358b922849ee>
- 35) Kłosowski, P., Dustor, A., Izydorczyk, J., Kotas, J., & Ślimok, J. (2014, June). Speech recognition based on open source speech processing software. In *International Conference on Computer Networks* (pp. 308-317). Springer, Cham.
- 36) 정지선, 김동성, 이홍주, & 김종우. (2019). 텍스트 마이닝 기법을 활용한 인공지능 기술개발 동향 분석 연구: 깃허브 상의 오픈소스 소프트웨어 프로젝트를 대상으로. *지능정보연구*, 25(1), 1-19.
- 37) <https://opensource.com/article/18/5/top-8-open-source-ai-technologies-machine-learning>
- 38) <https://www.analyticsvidhya.com/blog/2019/03/deep-learning-frameworks-comparison/>
- 39) <https://www.tensorflow.org/guide/extend/architecture>
- 40) Girija, S. S. (2016). Tensorflow: Large-scale machine learni<http://www.bikorea.net/news/articleView.html?idxno=18000ng> on heterogeneous distributed systems. Software available from tensorflow. org.
- 41) KB 지식 비타민, 2018. 3.4(18-20호), 종합 운송플랫폼을 꿈꾸는 우버(Uber)의 구상과 과제, KB금융지주 경영연구소
- 42) Li, L. E., Chen, E., Hermann, J., Zhang, P., & Wang, L. (2017, July). Scaling machine learning as a service. In *International Conference on Predictive Applications and APIs* (pp. 14-29).
- 43) <http://www.doai.ai/>
- 44) Glynn, E., Fitzgerald, B., & Exton, C. (2005, November). Commercial adoption of open source software: an empirical study. In *Empirical Software Engineering, 2005. 2005 International Symposium on* (pp. 10-pp). IEEE.
- 45) <https://allisonrandal.com/2017/11/25/capabilities-for-open-source-innovation-background/>
- 46) Gallivan, M. J. (2001). Organizational adoption and assimilation of complex technological innovations: development and application of a new framework. *ACM Sigmis Database*, 32(3), 51-85.
- 47) Ghapanchi, A. H., & Aurum, A. (2012). The impact of project capabilities on project performance: Case of open source software projects. *International Journal of Project Management*, 30(4), 407-417.
- 48) 장승권, 고경민, & 이희진. (2005). 오픈소스 소프트웨어와 개발도상국의 소프트웨어산업 발전전략. *기술혁신학회지*, 8, 297-322.
- 49) Carmel, Erran(2003a), Taxonomy of new software exporting nations
- 50) 한국IT비즈니스진흥협회 (2016). 오픈소스 커뮤니티 및 커미터 현황 조사
- 51) Ghapanchi, A. H., & Aurum, A. (2012). The impact of project capabilities on project performance: Case of open source software projects. *International Journal of Project Management*, 30(4), 407-417.
- 52) Ghafele, R., & Gibert, B. (2018). Open Growth: The Economic Impact of Open Source Software in the USA. In *Optimizing Contemporary Application and Processes in Open Source Software* (pp. 164-197). IGI Global.
- 53) The 2018 open source jobs report, the linux foundation & dice

※ 본 이슈리포트의 내용은 무단 전재할 수 없으며, 인용할 경우, 반드시 원문출처를 명시하여야 합니다.