

OSW 연구개발 Guideline

EunAh Lee, PhD



제4차 산업혁명

- 4차 산업혁명의 핵심: 초 연결 지능화
- DNA
 - Data
 - Network
 - Artificial Intelligence
- DNA를 고도화 시키는 기반은 SW - OSW 중요성 높음

공개 SW

- Open source SW
 - SW의 저작권자가 해당 소스코드를 공개하여 이를 사용, 복제, 수정, 배포할 수 있는 권한을 부여한 SW
- Open source SW License
 - 공개 SW의 사용, 복제, 수정, 배포와 관련하여 허용되는 권한 범위를 명시한 이용 허락 조건

공개SW의 장점

집단지능을 활용 개발 (개발 과정에 누구나 참여)
개방형 기술혁신 (개발기간 단축, 생산성 향상, 표준기술 선도)
다양한 환경에서의 상호 운용성 확보 용이

• 공개SW 활용 연구의 장점

개발효율성

- 공개SW는 SW개발을 위한 핵심 자산
- 상용SW의 96%가 공개SW 활용을 통해 개발기간 단축 및 비용절감 실현

기업 저변확보

- 소스코드 개방을 통해 SW저변 확산 용이



인재양성

- 공개SW 통한 SW교육의 효과적 활용 및 실무 SW기술 학습

최신기술 도입

- 소스코드 원천기술에 대한 접근으로 선진SW기술습득 용이
- 개방적인 SW개발문화에 따른 기술혁신과 아이디어 수용 가능

벤더 종속성 탈피

- 국내 시장에서의 독과점 SW 대체재 역할

단계	주요 활동	공개SW 개발 관련 핵심 검토사항
제안 단계	사업계획서 작성 및 제출	<ul style="list-style-type: none"> 사업 제안서(RFP)를 참조하여 공개SW 개발 방식을 적용한 연구개발 계획 작성 공개SW 라이선스 선정 평가기준 확인 및 관련내용 반영



수행 단계	연구개발 계획 수립		<ul style="list-style-type: none">소스코드(산출물) 공개 범위 설정소스코드(산출물) 공개 시점 설정소스코드(산출물) 공개 방법 설정
	▼		
	연구개발 수행 (반복)	요구사항 분석	<ul style="list-style-type: none">기존 공개SW 활용 시 조사, 분석, 평가, 계약
		설계	<ul style="list-style-type: none">모듈라 설계 방식 적용
		구현	<ul style="list-style-type: none">소스코드 통합 및 형상관리이슈 추적 및 관리소스코드 저장소 및 호스팅 환경 구축패키징
		테스트	<ul style="list-style-type: none">테스트 자동화 환경 구축공개SW 라이선스 검증 및 보안 취약점 점검소스코드 릴리즈 및 프로젝트 공개
	▼		
결과(연차) 보고서 작성		<ul style="list-style-type: none">최종(연차) 결과보고서 작성사업화 및 커뮤니티 운영 방안 포함	

활용 단계	



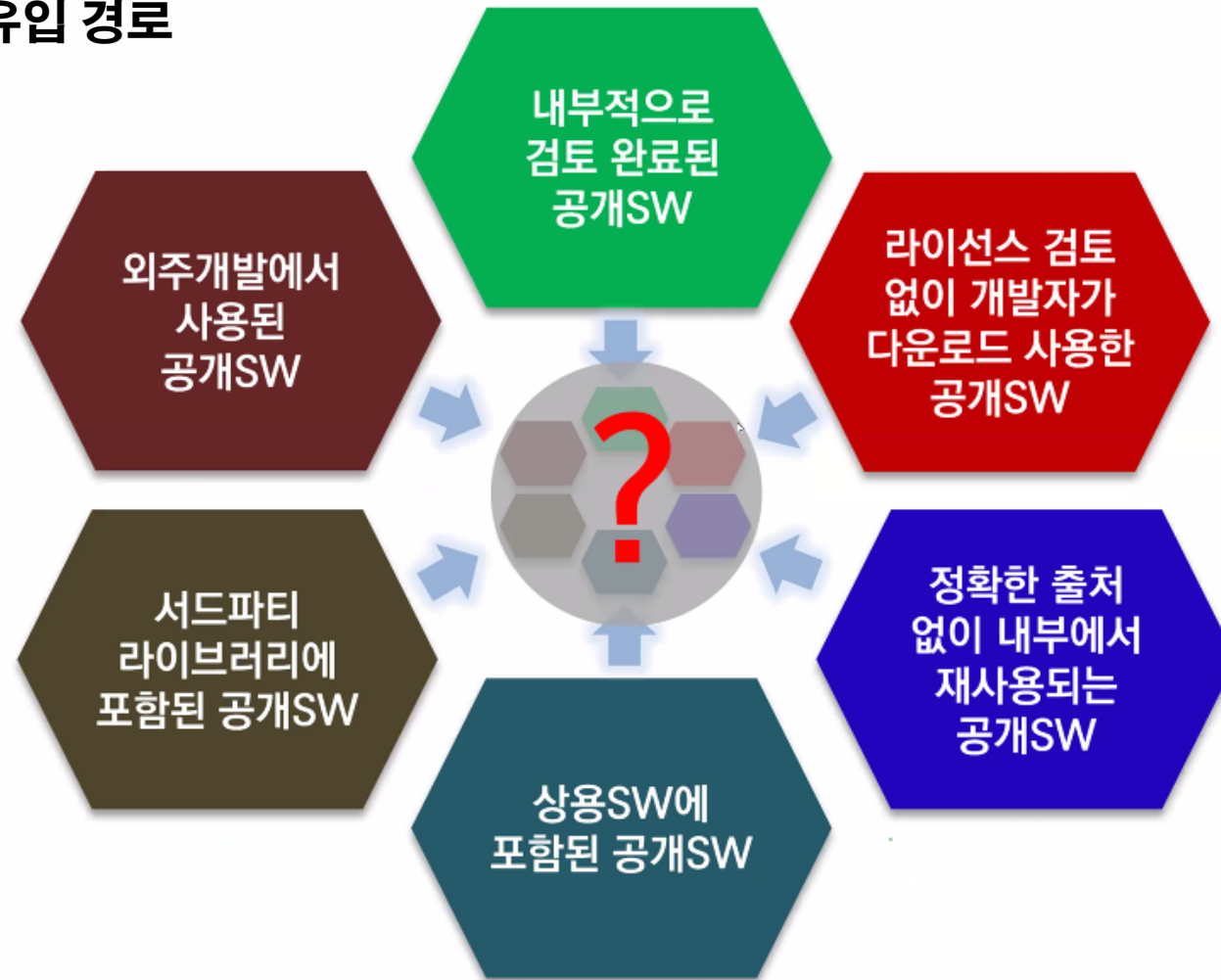
활용 단계	커뮤니티 운영		<ul style="list-style-type: none"> 연구개발 산출물을 일반에 공개하고, 외부 개발자 참여 유도를 위한 커뮤니티 개설 커뮤니티 행동규칙 및 의사결정 방안 수립 커뮤니티 멤버십 관리 및 작업 할당 커뮤니티 활성화를 위한 홍보활동 수행
	▼		
	사업화		<ul style="list-style-type: none"> 사업화를 위한 공개SW 비즈니스 모델의 구체화 및 실행

공개SW 라이선스

- 전체시스템을 자체 개발하여 외부에 공개하는 경우
 - 연구개발 종료 후의 사업화 및 비즈니스 모델에 따라 전략적으로 라이선스를 선정
- 기존 공개SW를 일부 또는 전체적으로 활용하여 추가 개작 후 외부에 공개하는 경우
 - 활용하고자 하는 기존 공개SW 라이선스를 확인하여야 함
- 연구개발과제에서 활용할 기존 공개SW는 목록으로 만들어 관리
- 필요 시 조사, 분석, 평가, 계약(듀얼 라이선스의 경우)등의 과정을 거쳐 자체적 검증과정을 거칠 수 있음

공개 SW 라이선스 검증의 필요성

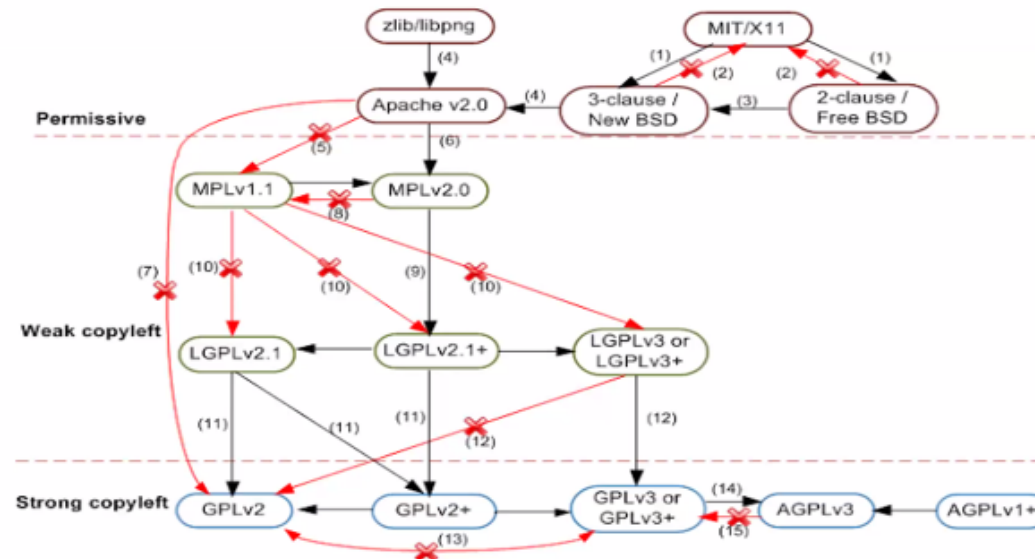
다양한 공개SW 유입 경로



공개SW 라이선스 분류

- 모든 공개SW에는 저작권이 존재함에 따라 공개SW 기반 개발 및 배포시에는 **저작권자가 부여한 라이선스 의무사항을 준수**해야 함

Permissive	Copyleft
<ul style="list-style-type: none"> 사용함에 있어서 별다른 요구사항을 부여하지 않고 광범위한 권한을 부여 라이선스 요구사항 최소화 (라이선스 copy, notice보전) 	<ul style="list-style-type: none"> 소스코드 공개 원 저작물 뿐 아니라 2차 저작물에도 동일한 라이선스가 부여 됨



고지의무

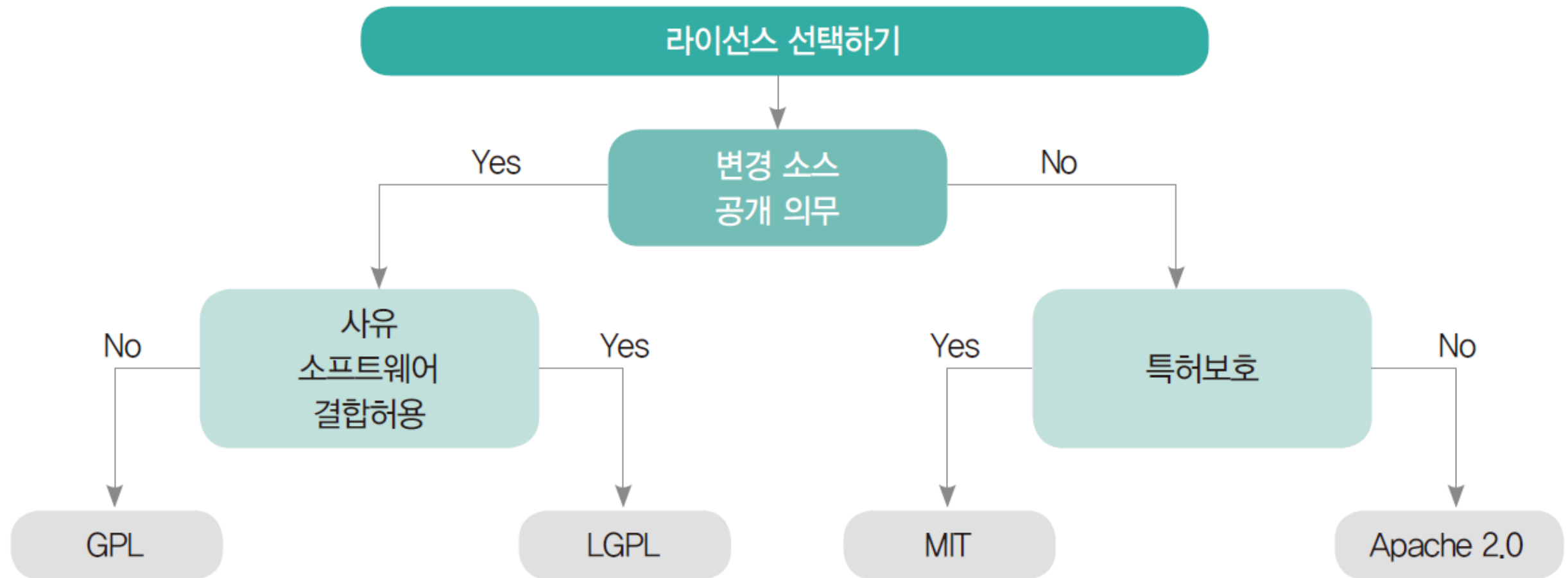
▷ BSD, MIT

수정코드에 대한 공개

▷ MPL, EPL, LGPL

결합된 모든 코드에 대한 공개

▷ GPL



주요 공개SW 라이선스

- 주요 공개SW 라이선스인 GPL, LGPL, CPL, MPL 등은 공개SW 코드 뿐 아니라 사용자 코드 공개의무 발생

라이선스	주요 의무사항	공개 범위
GPL (General Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, GPL 명시소프트웨어 수정 및 링크 시 모든 코드를 GPL에 의해 공개	<ul style="list-style-type: none">GPL 코드와 연결된 모든 코드
LGPL (Lesser General Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, LGPL 명시소프트웨어 라이브러리 수정 및 링크 시 라이브러리 소스 코드를 LGPL에 의해 공개, 파일단위 복제 및 수정시 GPL에 의해 공개	<ul style="list-style-type: none">LGPL 라이브러리를 구성하는 모든 코드
MPL (Mozilla Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, MPL 명시특허 권리 포기 조항(공개SW 사용자에게 특허권리 주장 할 수 없음)소프트웨어 수정 사용시 해당 파일을 공개	<ul style="list-style-type: none">MPL 코드를 수정한 해당 파일
EPL (Eclipse Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, EPL 명시특허 권리 포기 조항(공개SW 사용자에게 특허권리 주장 할 수 없음)소프트웨어 수정 사용시 해당 모듈을 공개	<ul style="list-style-type: none">EPL 코드를 수정한 해당 모듈



공개SW 라이선스 주요 의무사항

라이선스 주요규정

1

사용권 고지의 의무

- 누가 해당 SW를 개발하였는지 공지함
- 고객에게 어떤 OSS를 사용하였는지 알림

2

소스 코드 공개의 의무

- 일부 공개SW 라이선스는 공개SW로 개발한 결과물의 소스코드 공개를 의무화

3

저작권 고지의 의무

- 소스코드 상에 표시되어 있는 저작권 관련 문구는 절대로 삭제하여서는 안 됨

라이선스 관련 리스크

1

최종 판매 회사의 관리 책임

- 라이선스는 배포시점에 준수의무 발생
 - 버라이즌, 시스코 사례 : 금전적 배상

2

오픈소스 관련단체

- FSF (Free Software Foundation)
 - 리처드 스톨만 설립 / GNU프로젝트 운영
- OSI (Open Source Initiative)
 - 오픈소스 소프트웨어 정의 / 사용 장려
- SFLC (Software Freedom Law Center)
 - 오픈소스 개발자들을 위한 법률 서비스

공개SW 사용에 대한 의무는 최종배포자에게

공개SW 라이선스의 의무 사항

저작권 고지	라이선스 사본 포함(영문)
보증의 부인	책임의 제한
소스코드 공개의무	설치정보 제공



사용 및 배포에 대한 모든 책임은 배포자!

공개SW 라이선스 별 의무사항 비교

라이선스 의무사항	Commercial License	GNU General Public License v2.0 or later	GNU General Public License v3.0 or later	GNU Lesser General Public License v2.1 or later	GNU Lesser General Public License v3.0 or later	GNU Affero General Public License v3.0	Eclipse Public License 1.0	Mozilla Public License 1.1	Apache License 2.0	BSD 3-clause "New" or "Revised" License
배포권리(오브젝트 /바이너리 코드 배포)	X	X	X	X	X	X	X	X	X	X
배포(코드배포에 의해서만 부여되는 의무사항)	X	O	O	O	O	O	X	O	X	O
소스코드 배포 / 강제적 공유 의무사항	X	O	O	O	O	O	O	O	△	△
복제 권한 허용	X	O	O	O	O	O	△	△	△	△
수정(개작)권한 허용	X	O	O	△	O	O	△	△	△	△
역설계 권한 허용	X	O	O	O	O	O	△	△	△	△
차별적 제한 금지	X	O	O	O	O	O	△	O	△	X
추가 복제에 대한 로열티 및 수수료 금지	X	O	O	O	O	O	O	O	△	△
특허보복(특허소송 제기 시 라이선스 종료)	X	X	O	X	O	O	O	O	O	X
명시적 특허라이선스 (특허 소송을 제기하지 않음)	X	X	O	X	O	O	O	O	O	X
DRM 금지	X	X	O	X	O	O	O	O	O	X
고지(특정법률 혹은 속성)	X	O	X	O	X	O	X	X	X	X
변경사항 고지	X	O	O	O	O	O	O	O	O	X
변경사항에 대해 원저작자에게 사용허가	X	X	O	X	O	O	O	O	X	X
다른 사람을 대신한 보증의 부인	O	O	X	O	X	X	O	O	O	X
다른 사람의 책임의 제한	O	O	X	O	X	X	O	O	O	X
배포 /사용으로 인해 발생한 원저작자의 클레임에 대한 배상	X	X	X	X	X	X	O	O	O	X
배포 시 라이선스 사본 포함	X	O	O	O	O	O	O	O	O	O
광고/홍보 시 배포자, 저작자, 특정상표사용 금지	X	X	X	X	X	X	X	O	O	O
원 코드와 동일조건 허가	X	O	O	O	O	O	O	O	X	X
라이선스 확장 범위 (공개범위)	X	코드 기반의 산출물 (per GPL)	코드 기반의 산출물 (per GPL)	동적 라이브 러리 (per LGPL)	동적 라이브 러리 (per LGPL)	코드 기반의 산출물 (per GPL)	모듈 (per EPL/ CPL)	파일 (per MPL)	X	X

O : 필수 의무사항, X : 필수 의무사항 없음, △ : 명문화된 조항 없음

공개SW 라이선스의 양립성

라이선스 선정 시 서로 다른 의무사항을 가진 라이선스 간 양립(compatibility)이 가능하도록 검토 필요

GNU General Public License 2.0

+

Apache License 2.0

≠

GNU General Public License 2.0과 Apache License 2.0은 라이선스
의무사항이 충돌하여 함께 사용하여 재배포 할 수 없음

MPL 1.1(Mozilla public License)
다른 소스 코드와 결합하여 사용할 경우
수정된 MPL 파일을 반드시 MPL과 동일
한 조건으로 공개



+

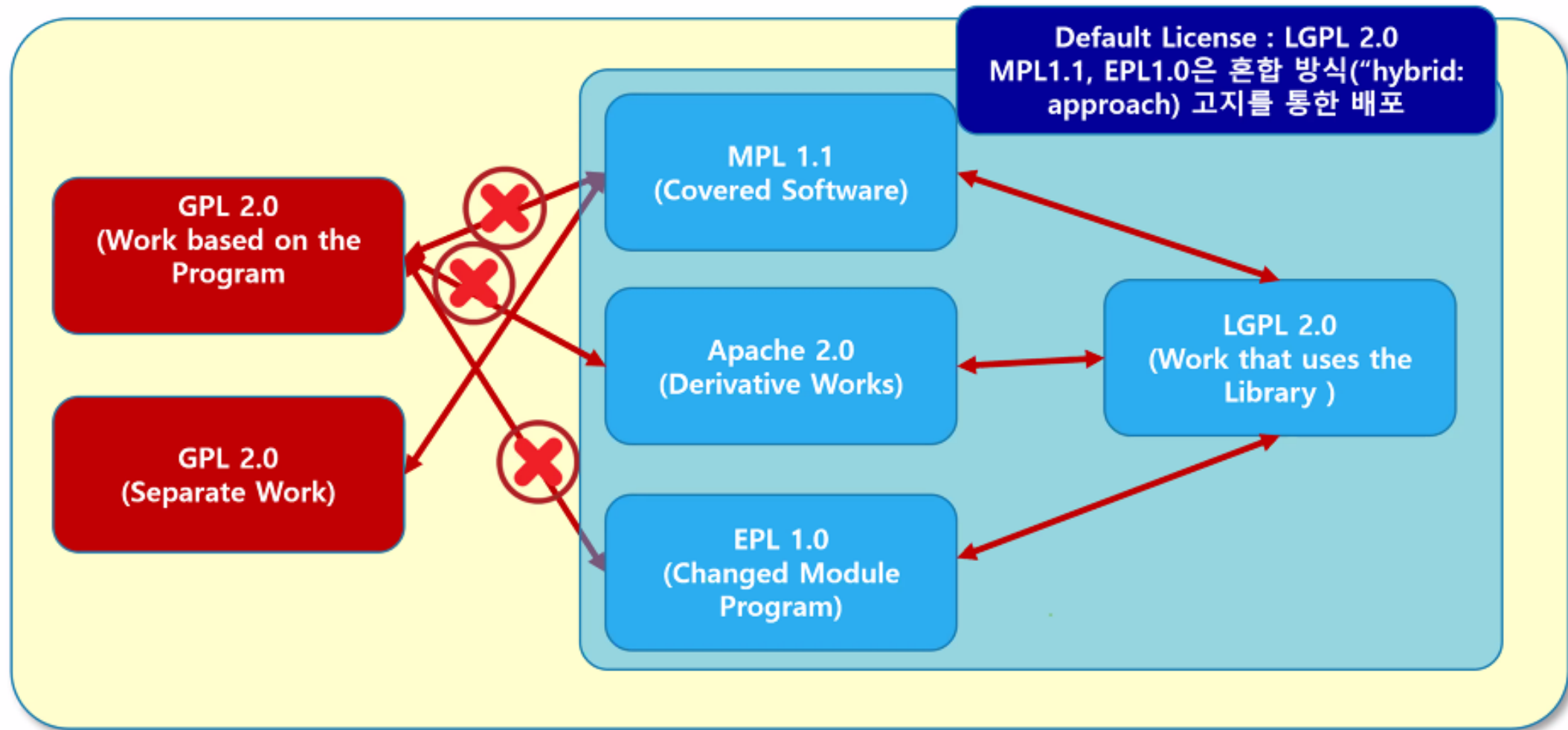
GPL 2.0(General public License)
다른 소스 코드와 결합하여 사용할 경우
결합하는 모든 소스 코드를 GPL과 동일한
조건으로 공개



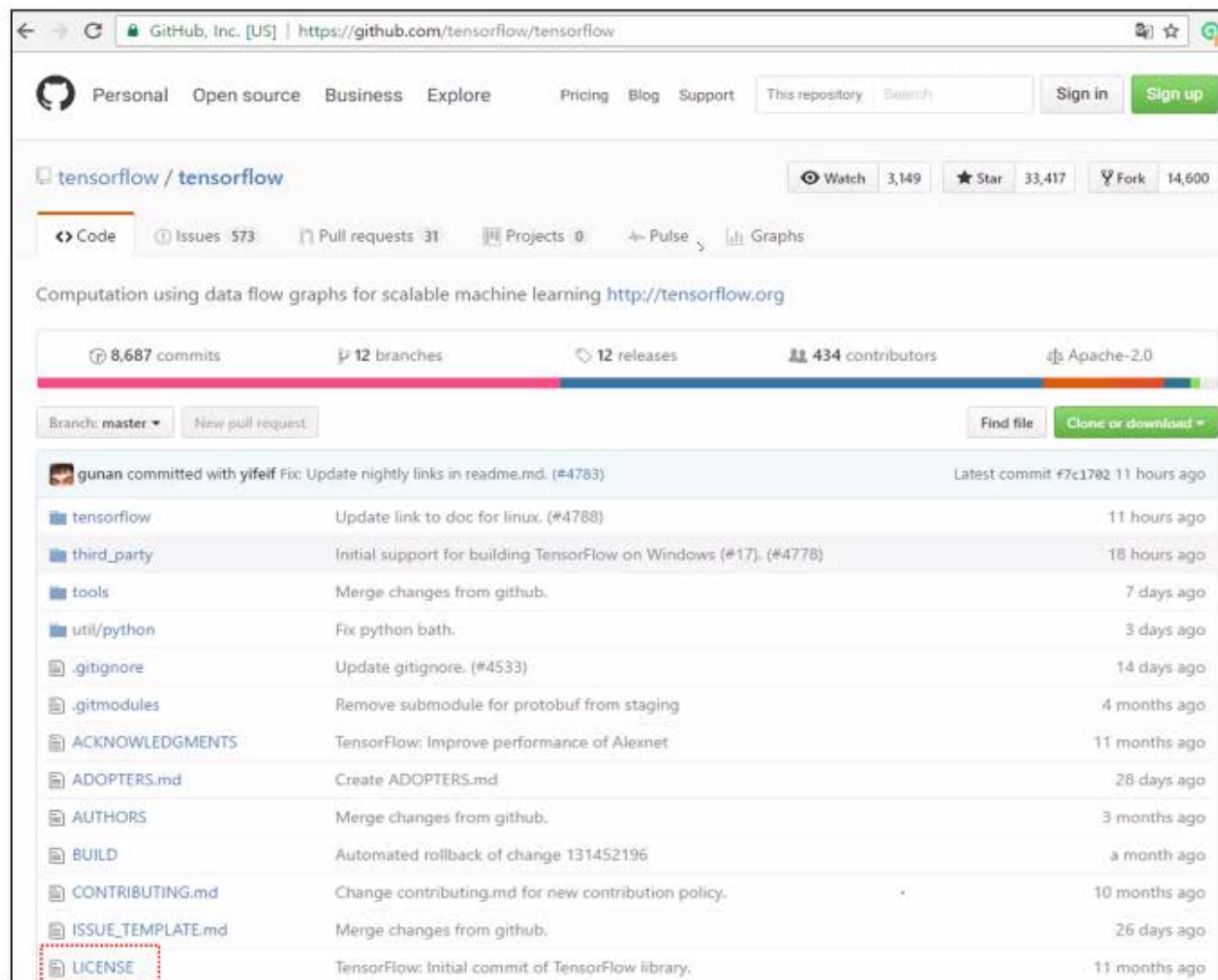
=



MPL 2.0의 경우 secondary License를 GPL, AGPL 등으로 정의하여 다른 라이선스로 확대가능



공개SW 라이선스 고지 방법



The screenshot shows the GitHub repository page for TensorFlow. The repository is named 'tensorflow / tensorflow' and has 3,149 watches, 33,417 stars, and 14,600 forks. It is licensed under Apache-2.0. The page displays a list of recent commits, including updates to the README, documentation, and the LICENSE file. The 'LICENSE' file is highlighted with a red dashed box.

Commit	Message	Time
gunan committed with yifei	Fix: Update nightly links in readme.md. (#4783)	Latest commit #7c1702 11 hours ago
tensorflow	Update link to doc for linux. (#4788)	11 hours ago
third_party	Initial support for building TensorFlow on Windows (#17). (#4778)	18 hours ago
tools	Merge changes from github.	7 days ago
util/python	Fix python bath.	3 days ago
.gitignore	Update gitignore. (#4533)	14 days ago
.gitmodules	Remove submodule for protobuf from staging	4 months ago
ACKNOWLEDGMENTS	TensorFlow: Improve performance of Alexnet	11 months ago
ADOPTERS.md	Create ADOPTERS.md	28 days ago
AUTHORS	Merge changes from github.	3 months ago
BUILD	Automated rollback of change 131452196	a month ago
CONTRIBUTING.md	Change contributing.md for new contribution policy.	10 months ago
ISSUE_TEMPLATE.md	Merge changes from github.	26 days ago
LICENSE	TensorFlow: Initial commit of TensorFlow library.	11 months ago

공개SW 라이선스 고지 방법

- 삼성전자에서는 Apache License Version 2.0과 같이 코드 공개 의무사항이 없지만 사용자 고지 의무사항을 준수하기 위해 Open Source Announcement를 고지하고 있음

Open Source Announcement

Some software components of this product incorporate source code covered under the BSD and Australian Public License B (OZPLB).

And this product includes software developed by the NetBSD Foundation, Inc. and its contributors.

Component	License
Windows Template Library	Microsoft Permissive License (Ms-PL)
Khronos Group - OpenGL ES 1.X	SGI Free Software License B 1.0
Android Open Source Project	Apache License Version 2.0
Code Project	Code Project Open 1.02 License
Common L4	Australian Public Licence B (OZPLB) Version 1-0
Copyright (c) Regents of the University of California	Regents of University of California License
Copyright (c) Carnegie Mellon University	CarnMellon Mach OS License
NetBSD	BSD 1.0

Microsoft Permissive License (Ms-PL)

Published: October 18, 2005

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

* The terms "reproduce," "reproduction" and "distribution" have the same meaning here as under U.S. copyright law.

* "You" means the licensee of the software.

* "Licensed patents" means any Microsoft patent claims which read directly on the software as distributed by Microsoft under this license.

2. Grant of Rights

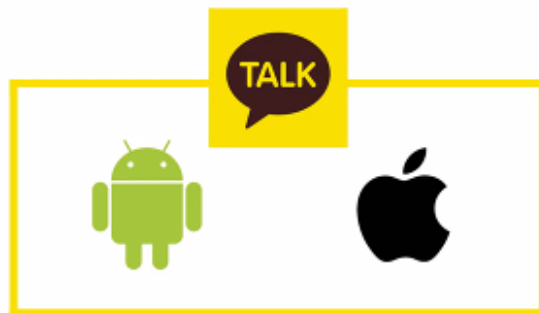
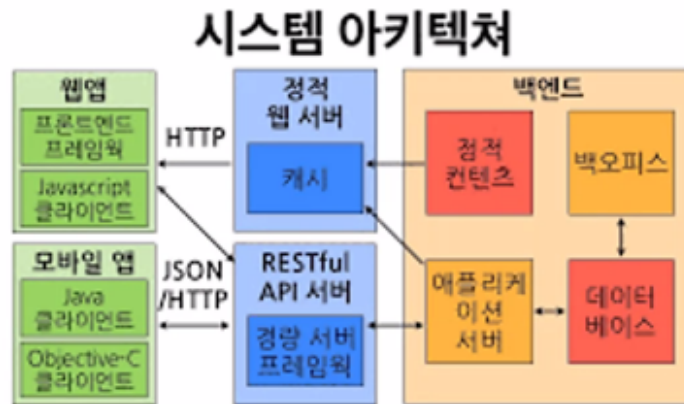
1. Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, Microsoft grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce the software, prepare derivative works of the software

and to distribute the software, or any derivative works thereof, under the terms of this license.

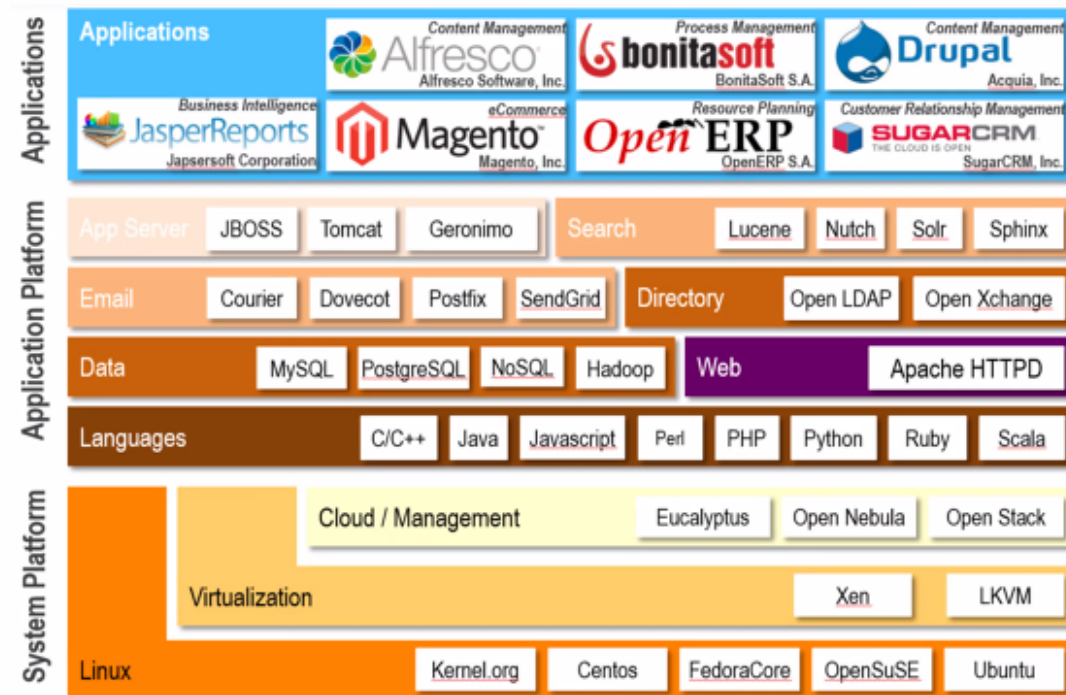
[출처] <http://opensource.samsung.com/reception.do>

SW별 라이선스 검증 수행

■ SW별로 라이선스 검증 수행이 이루어져야 함



분류 체계(예)



SW Engineering

- SW 개발 방법론과 프로젝트 관리

SW 개발 방법론

마케팅과 같이 성공을 위해서 해야 할 여러 일들에 대한 전략적인 측면의 일 (Strategic Discipline)

이슈해결 근간 정의, 요구 최종산출물과 결과 속성 규정, 주요산출물 서술&구성, 기술적 역할과 책임 규정, 비즈니스 요건에 대한 진도 파악

Waterfall, OOP(객체지향방법론), CBD(Component기반 방법론), UP, Agile, 스크럼

프로젝트 관리

영업과 같이 성공을 위한 운영적인 측면의 일 (Operation Discipline)

일의 계획과 관리를 위한 근간 정의, 최종 산출물과 결과의 일정&예산목표와 균형 방법 규정, 산출물을 위한 일 서술&구성, 관리역할 및 책임 규정, 프로젝트 계획에 대한 진도파악

PMBOK, PRINCE2

공개 SW 연구개발 과제의 개발 유형

- **연구개발 완료 후 최종 산출물만을 공개하는 과제**
 - 일반적 비공개 연구개발 프로세스 방법론과 유사 (Waterfall model)
 - 관련경험이 부족할 경우 시행착오를 겪을 수 있으므로 한시적으로 프로젝트를 비공개로 하되 산출물 저장소에서는 이력이 지속적으로 쌓이도록 유지한 후 일부라도 공개가능한 시점이 되었을 때 프로젝트를 공개로 전환하는방식도 고려 가능
- **연구개발 초기부터 산출물을 공개하고 외부 개발자의 참여를 유도하여 협업하는 과제**
 - 실제 대부분의 공개 SW 커뮤니티에서 적용하는 개발방식
 - 반복 순환 방법론(Agile process)

Waterfall Process/구조적 방법론

절차 중심의 SW 개발 방법론

- 코드를 제한된 구조에서 생성하여 순차적으로 실행시키는 특징
- 알고리즘 단위인 순차(sequencing), 선택(selection), 반복(iteration)의 구조로 코드를 표현
- 프로세스 단위로 문제 해결, 코드 보안 유지, 유지보수 용이

단점

소프트웨어
개념

요구사항
분석

Requirements

아키텍처
설계

Design

상세설계

Implementation

구현 및
디버깅

Verification

시스템
테스트

Maintenance

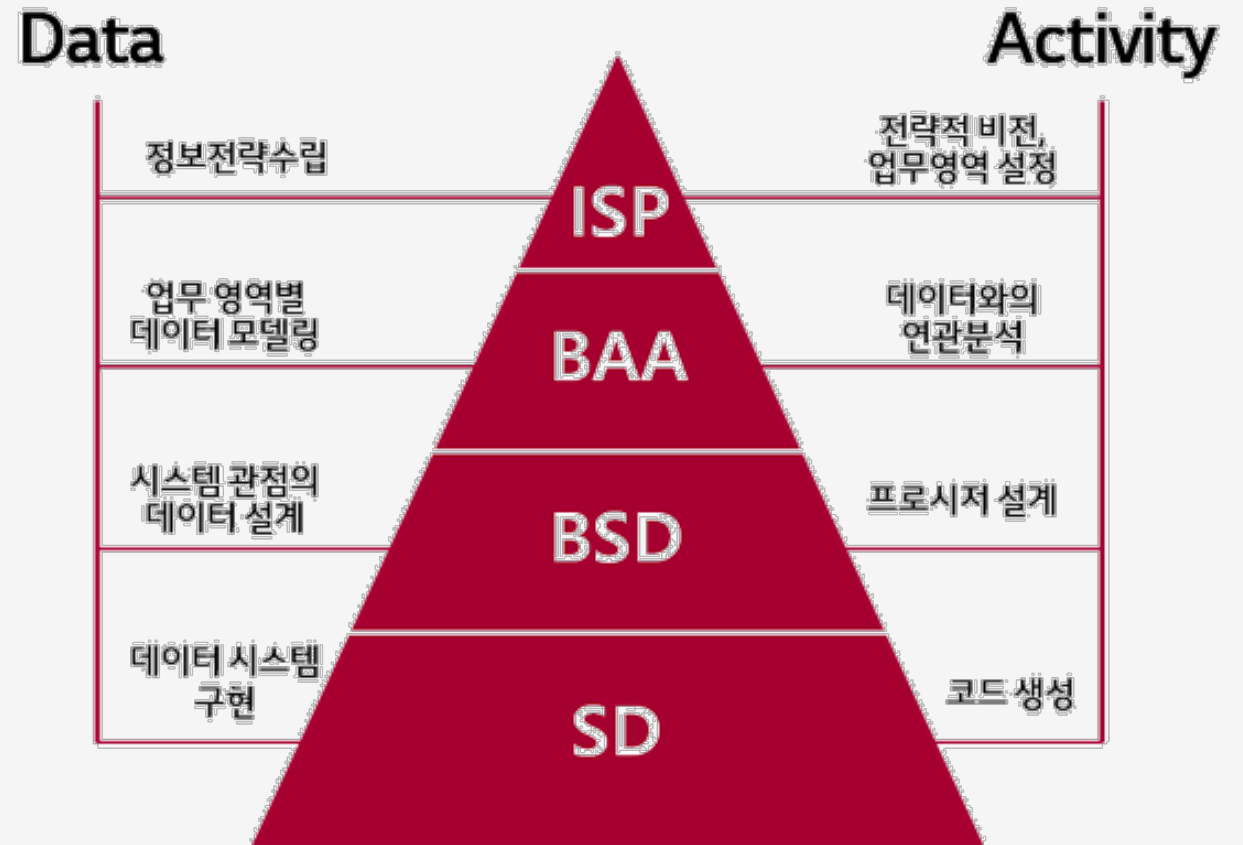
- 실제 시스템 개발 패턴에 맞지 않는다.
- 이전 단계에 대한 오류 수정 및 추가 개선이 용이하지 않다.
- 종종 엄청난 양의 문서작업을 야기한다.

장점

- 시스템 개발에 필요한 task들을 한눈에 파악할 수 있다
- 각 단계의 완료 정의가 매우 명확하다
- 대규모 시스템 개발 프로젝트의 프레임워크를 제공한다

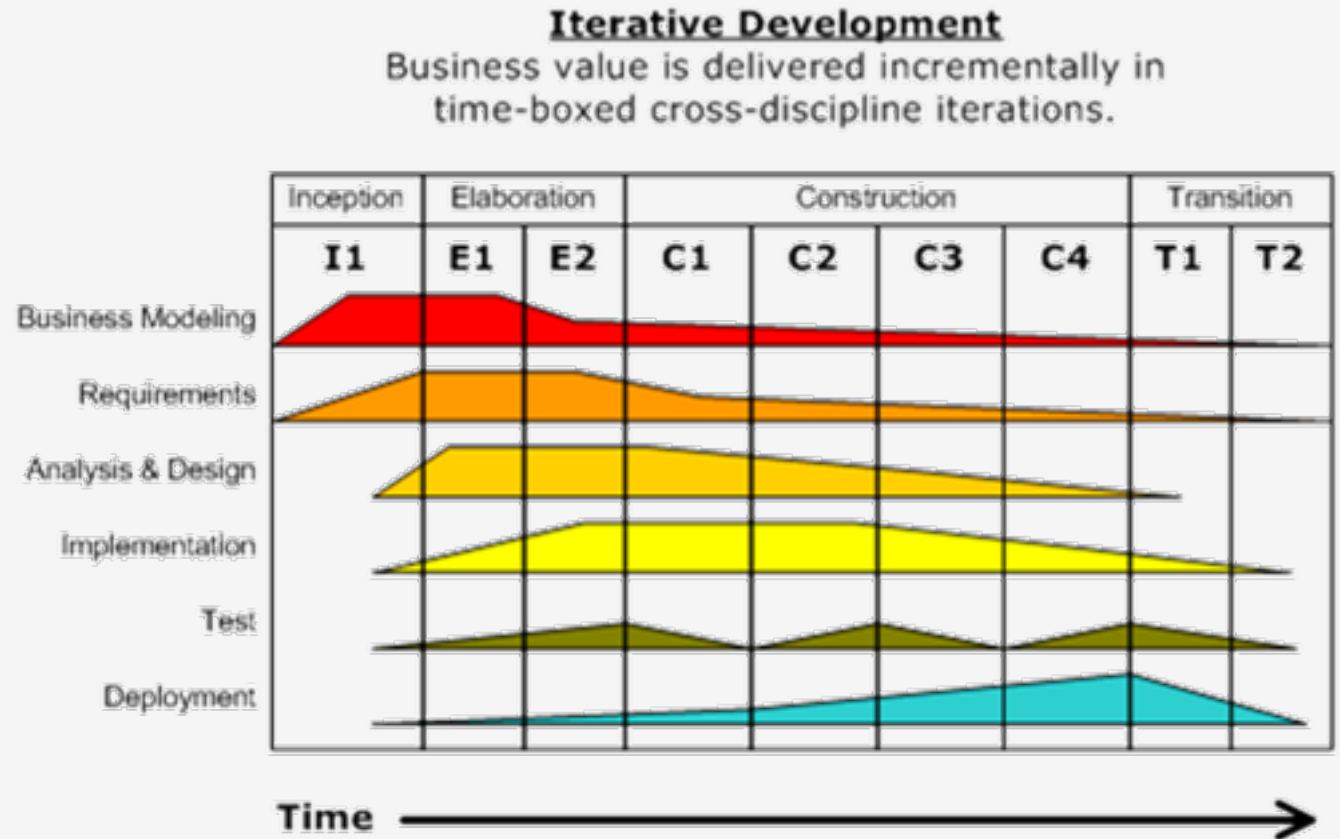
정보 공학적 방법론

- 설계와 구현 단계에서 데이터를 우선적으로 개발
- 문제 영역을 세분화하고 Top-down 방식으로 전개
- 기존 보다 빠르게 결과물을 낸다는 특징



객체 지향 방법론

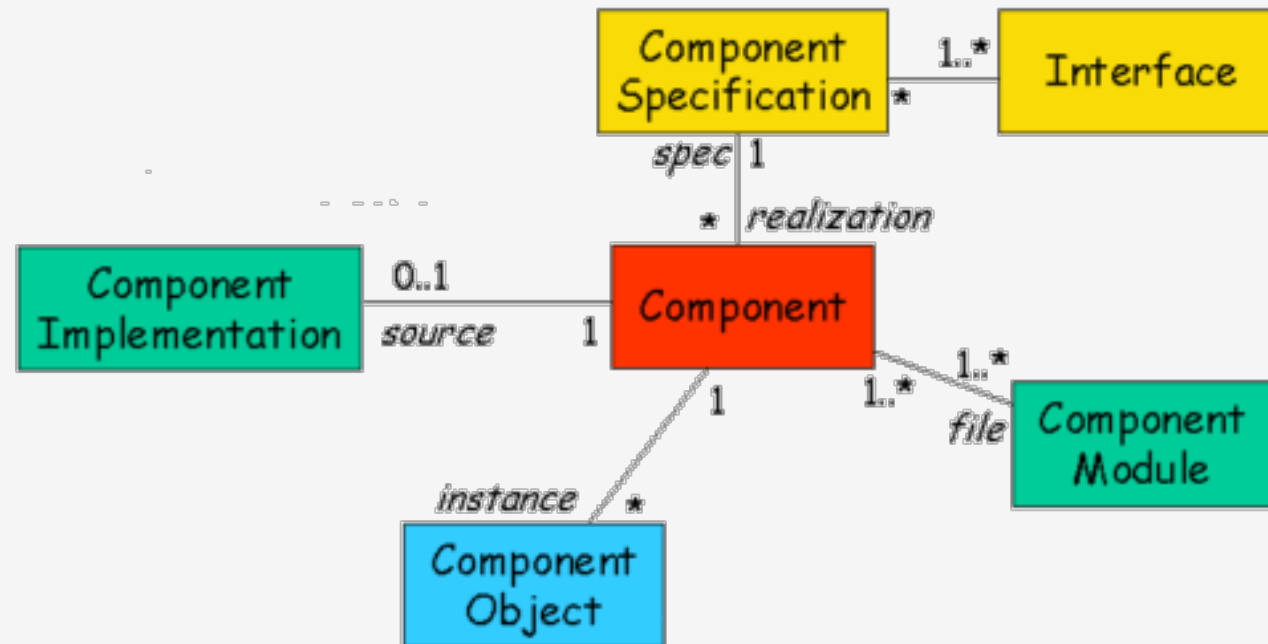
- 개발단계에서 반복과 점증적 (Iterative & Incremental) 모델을 사용하여 사용자의 요구사항을 반영하고, 모든 단계를 유기적으로 협력 시켜 전체 프로세스의 방향성을 유지하고 재사용성을 높이는 특징



RUP(Rational Unified Process) 모델. Rational에서 제안한 객체지향 프로젝트 수행에 맞도록 개발된 UML 기반의 객체 지향 방법론

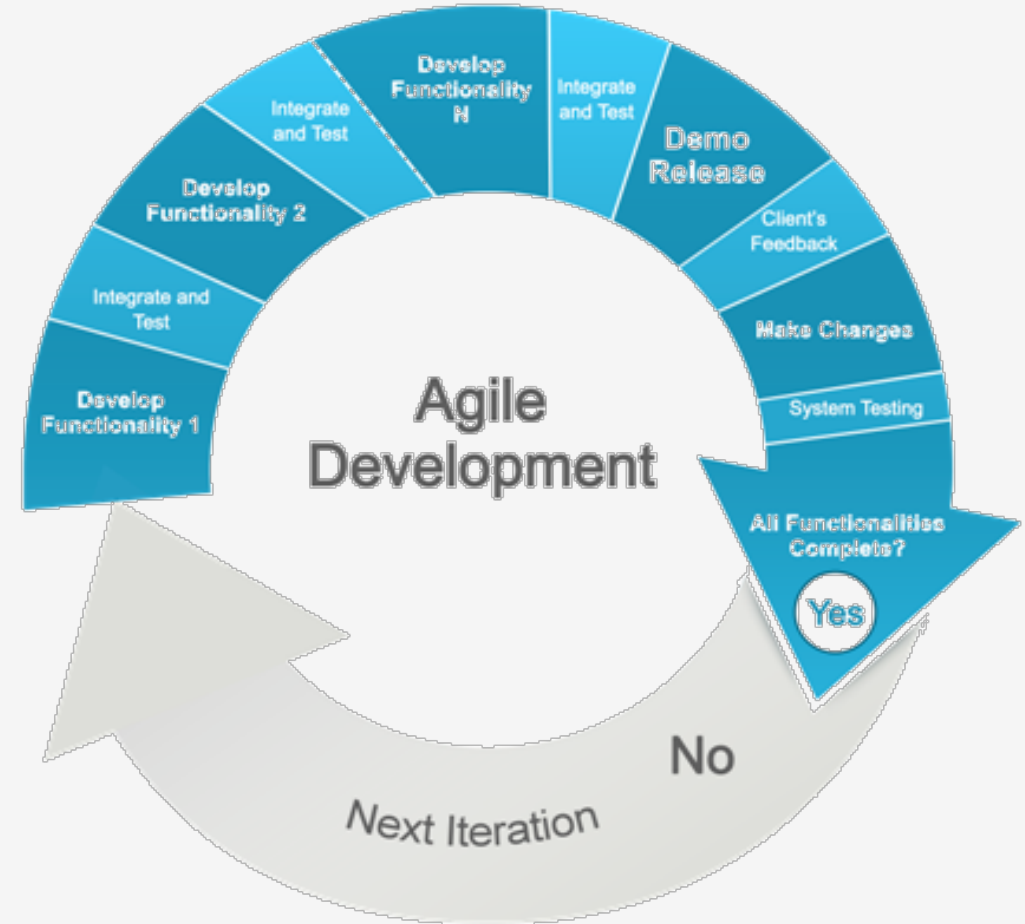
CBD (Component-Based Development) 분석 방법론

- 문제를 조각으로 나누어 각각 component를 생성한 후 다시 조합하는 재사용성에 초점을 둔 방식
- 아무리 복잡한 시스템도 단계별로 나누어 생각하기 때문에 전체 시스템이나 프로그램에 영향을 주지 않고 빠르게 문제를 해결한다는 특징



Agile Development (애자일 방법론)

- 과도한 문서화나 형식적 절차에 따르는 비용 소모 감소 장점
- 고객과의 협력 중시, 프로세스나 도구에 국한되지 않는 자기 적응적(Self-adaptive) 방식
- 일정 주기를 가지고 프로토타입을 생성하므로 요구사항 반영이 쉽고 변화에 빠르게 대응 가능



오픈 프로젝트의 운영

공개SW 설계과정 - 개발 방식의 특성 상 Architecture 중요

- 소스코드 공개 이후 외부개발자 참여
- 소스코드의 응집도가 낮도록 가능한 한 많이 분리된 모듈들로 시스템 설계 필요
 - Cf. 윈도우에 대한 Linus B. Torvalds의 비판:
“MS 윈도우가 소스코드를 오픈한다 하여도 리눅스가 이룩한 만큼의 발전을 이룰 수 없다 ”
(윈도우는 소스코드의 응집도가 높아 모듈 단위의 개별적 개발이 불가능)
- 대형 공개SW 프로젝트의 핵심 아키텍처는 시스템에 필수적인 모듈들을 지원하는 플랫폼과 그 상위에 있는 서로 다른 모듈들의 집합으로 구성
- 타인이 이해할 수 있도록 명확하게 설계&기술 필요

모듈라 설계방식의 특징

- 설계 디자인이 뚜렷하고 명확하며 이해하기 쉽다
- 모듈 간 느슨한 결합은 설계 시에 다른 모듈들에 영향을 주지 않고 작업을 수행할 수 있게 함
- 높은 자율성 제공 (개발자/기여자 사이의 보다 적은 상호작용 요구)
- 시너지 효과와 더불어 협업의 기회를 촉진
- 실험적 탐구적인 구현 시도들이 보다 안전하게 수용될 수 있음
- 코드의 변경 및 개선이 전체 시스템을 위태롭게 하지 않고도 수행 가능

Core 멤버
소유자
코어 개발자

Active 멤버
액티브 개발자
버그 수정자

주변 멤버
주변 개발자
버그 보고자
액티브 유저
제휴 단체/기업

단계	핵심 검토사항		비고
사업계획서 작성	공개SW 관련 항목 작성		
	과제선정 평가지표 확인 및 반영		
	공개SW 라이선스 선정		
연구개발 계획 수립	소스코드	공개 범위 설정	
	소스코드	공개 시점 설정	
	소스코드	공개 방법 설정	
연구개발	요구사항 분석	조사	기존 공개SW 활용 시
		분석	
		평가	
		계약	
	설계	모듈라 설계	
	구현	소스코드 통합 및 관리	
		소스코드 형상 관리	
		이슈 추적 및 관리	
		저장소 및 호스팅 설비	
		패키징	
	테스트	테스트 자동화	
		공개SW 라이선스 검증	
		보안 취약점 점검	
		릴리즈	
		론칭	
커뮤니티 운영	커뮤니티 운영 목적		
	커뮤니티 구조		
	커뮤니티 운영		
	커뮤니티 활성화(홍보 등)		

오픈 프로젝트의 시작

1. 해당되는 OSS 커뮤니티를 찾는다
2. 해당되는 OSS 오픈 프로젝트가 없는 경우 → 단계별로 신규 커뮤니티 구축 필요

일반적인 사람들의 생각:
(하지만 이런 방법으로 성공한 그룹이 별로 없음)

논문발표 및 기술의
개념 정립

오픈 소스 프로젝트
생성

오픈소스 커뮤니티
구축

실제 일어나는 일들의 성장 순서 (Community Building Stages)



Technical Stage



Open Source
Stage



Ecosystem Stage

Technical Stage

- Web site
- Source code repository system
- Mailing list
- IRC/slack
- Bug and feature tracking

OpenSource Stage

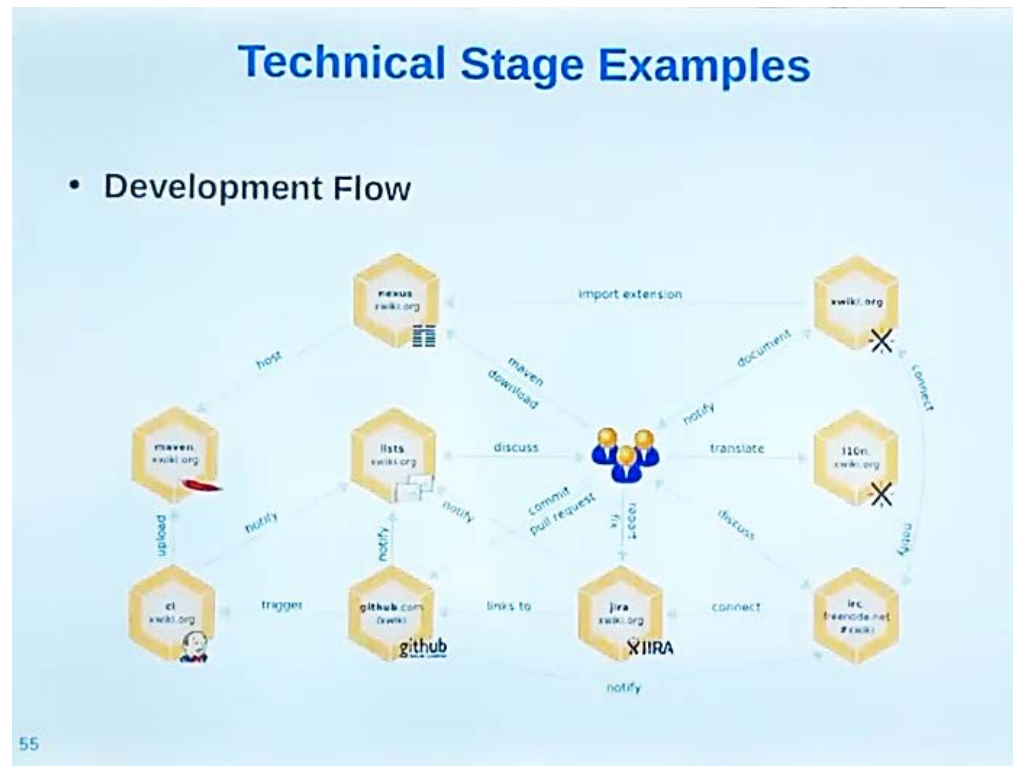
- Documentation collaboration tool (Wiki)
- Milestone and release tracking
- Forums
- Automated build and test system

Ecosystem Stage

- Branding (logo, style guide, official colors or fonts)
- Reporting system (codebase activity reports)
- Promotion tools

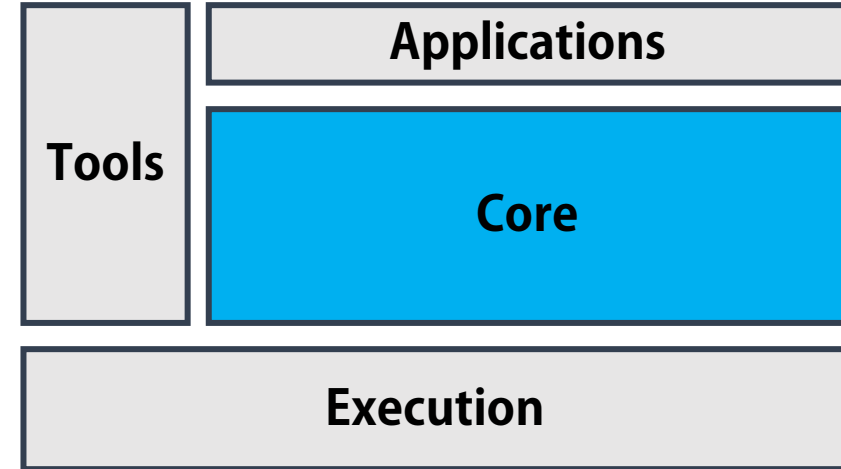
Technical Stage

- Web site
- Source code repository system
- Mailing list
- IRC/slack
- Bug and feature tracking



55

Component license management



- 라이선스 정책
- 잘 동작하는 SW
- 프로그램 데모
- 프로그램 적용사례
- Third Party 도구 지원

Governance 구축 시 필수적으로 준비할 다섯가지

- Overview
- Roles and responsibilities
- Decision making process
- Contribution process
- Support

Template for a benevolent dictator governance document

- <http://oss-watch.ac.uk/resources/benevolentdictatorgovernancemodel>

Template for a meritocratic governance document

- <http://oss-watch.ac.ur/resources/meriocraticgovernancemodel>

References

1. SW 개발 방법론
<https://multicore-it.com/46>
2. SW 개발 방법론
2020 OSS 개발자대회 온라인 강의 '공개SW 이해하기' 제2강
3. 공개 소프트웨어 연구개발 수행 가이드라인
(2018.06. 과학기술정보통신부 & 정보통신산업진흥원)
4. 2020 OSS 개발자 대회 라이선스 검증 멘토링 (OpenUP/백영석 책임)

EOD