



제품소프트웨어 패키징 (Git과 GitHub를 활용한 소스코드 관리)

License 개념 및 관리



학습내용

- 오픈소스 License의 이해
- 오픈소스 License 종류
- 오픈소스 License 검증 도구



학습목표

- 오픈소스 License가 무엇인지 설명할 수 있다.
- 오픈소스 License의 종류를 설명할 수 있다.
- 오픈소스 License 검증 도구를 설명할 수 있다.



오픈소스 License의 이해

1. 오픈소스 소프트웨어란?

1) 오픈소스란?

- 자유롭게 누구나 열람하거나 수정이 가능하고, 다른 사람들에게 배포를 할 수 있는 소프트웨어
 - 공정한 운영을 위해 License 규칙 준수

2) License의 종류

- GNU, BSD, Apache, MIT 등 70여 가지 이상
- 오픈소스를 처음 만들고 진행하는 사람들에 의해 어떤 License를 적용할지 결정

3) 오픈소스에 대한 오해

- 오픈소스를 만드는 사람은 무조건 소스를 무료로 공개하고 주기만 한다는 것
- 오픈소스 사용자는 아무 제한 없이 공짜로 쓰기만 하면 된다는 것
- 오픈소스의 참된 의미
 - 오픈을 통해 본인의 프로젝트가 커지는 것
 - 다양한 사람들의 피드백을 통해 더 좋은 소스코드로 발전하는 것

4) 오픈소스에 의한 선순환 구조

- 오픈소스 실적을 통해 유능한 개발자로 성장
- 소프트웨어 산업을 리딩하는 리더의 역할
- 직접 개발하는 것보다 짧은 시간과 노력으로 소프트웨어 개발
- 혜택을 받은 만큼 오픈소스에 기여
- 오픈소스의 발전으로 기업에서도 더 좋은 소프트웨어 개발



오픈소스 License의 이해

1. 오픈소스 소프트웨어란?

1) 오픈소스란?

- 커뮤니티에 의해 집단으로 만들어진 소프트웨어
 - License에 대한 규칙에 얽매이지 않음
 - 동료들끼리 서로 소스코드를 검토하고 운영
 - 모든 의사결정과 문서화 등은 오픈소스에 참여하는 모든 사람들에 의해 이루어짐

2) 오픈소스 활용 현황

- 돈을 주고 소프트웨어를 구입 · 설치하던 시대
 - 일부 착한 사람들에 의해 무료로 배포하는 것
 - 기업에서도 비용절감을 위해 사용하지만 사후관리 미흡으로 도입은 꺼려하는 분위기
- 최근
 - 78% 이상의 회사가 오픈소스 기반으로 운영, 3% 이하만 오픈소스를 전혀 사용하지 않음
 - 업무 방식 : 소프트웨어 개발이 필요할 때 66% 이상의 회사가 오픈 소스가 있는지를 먼저 확인
 - 회사 내의 엔지니어의 50% 이상이 오픈소스 관련 일에 참여하고 있는 회사가 절반
 - 39%의 회사가 현재 오픈소스 프로젝트를 계획하고 있으며 47%의 회사가 자사에서 개발한 프로젝트를 오픈소스로 공개
 - 보안 부분 : 55% 회사가 오픈소스가 오히려 보안에 안전하다고 응답
- 오픈소스는 이제 소프트웨어 산업에서 단순한 트렌드 수준이 아닌 기본 개념으로 인식되고 있으며 이미 많은 산업에서 활용



오픈소스 License의 이해

2. 오픈소스 소프트웨어 장점

1) 특정 제품이나 제조사에 한정되지 않고 모두 지원하는 것이 목표

- 오픈소스 자체가 비즈니스 모델로 가치를 보유
 - 주로 큰 프로젝트에 해당
 - 예) 리눅스 : 운영체제 판도를 바꾸고 영리를 추구하는 기업 존재
 - 예) 웹 브라우저인 파이어폭스 : 오픈소스를 기반으로 모질라라는 영리 기업이 운영

2) 핵심역량을 발휘할 수 있는 부분에 집중

- 특정부분을 타겟으로 오픈소스 진행
- 세계의 전문가들 참여
- 다양한 사람들과의 만남
- 본인의 역량을 알림
- 새로운 도전과 사업의 기회 부여

3) 기업에서 인력 양성 및 리크루팅의 기회 부여

- 오픈소스는 단순히 무료로 사용하게 해주는 소프트웨어 개발 방법이 아닌 산업 전체에 이로운 방향을 제시하는 방법



오픈소스 License의 이해

3. 오픈소스 프로젝트 알아보기

1) 리눅스 운영체제

- 초기
 - 윈도우와 달리 무료로 배포
 - 많은 사람들이 사용
- 현재
 - 리눅스의 발전 가속화, 웹 발전, 모바일 시대
 - 오픈소스는 모두에게 이익이 되는 비즈니스로 발전
 - 수많은 기업들이 이익 발생

2) 안드로이드 등장

- 컴퓨터 시대의 윈도우와 같이 유료 운영체제가 독점하는 상황 붕괴
- 다양한 소프트웨어 프로젝트들이 모두 오픈소스를 기반으로 이루어는 추세
- 수많은 기업들이 이익 창출
- 다양한 오픈소스 프로젝트가 운영되고 있으며 소프트웨어 기술은 더욱 더 많이 발전하는 중



오픈소스 License 종류

1. 오픈소스 License 관리

1) 오픈소스

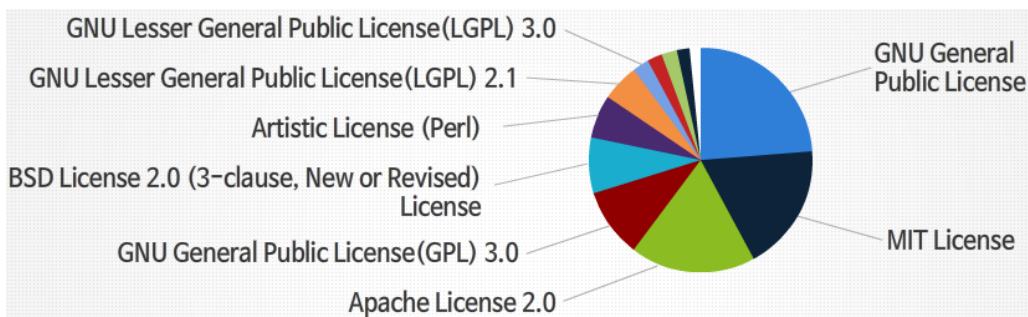
- 주인이 없고 별도의 특허 등으로 보호받지 못하는 형태
 - 잘못된 방향으로 운영되는 것을 막기 위하여 자체적으로 License 정책을 만들어 보호

2) 오픈소스 License의 특징

- 모든 사람들이 제보자가 될 수 있어 저작권이 보호되는 효과를 기대
- 성격에 따른 다양한 License 정책 보유
- 오픈소스 운영 그룹에서 License 선택 가능

3) 오픈소스 License의 사용

- 2015 통계자료



- 전혀 새로운 License를 만들어서 고시하고 운영
- 대부분 현재 만들어진 몇 개의 License를 따르는 것이 추세



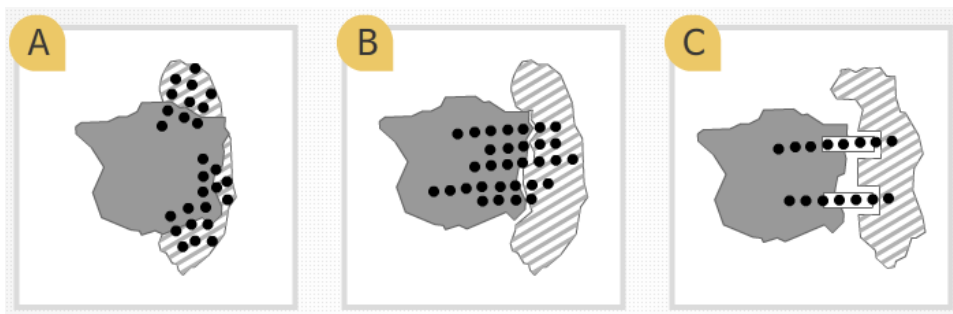
오픈소스 License 종류

1. 오픈소스 License 관리

4) 오픈소스 License에서 중요한 것

- 새롭게 생산되는 소스코드의 저작권
- 기존 소스를 활용해 새로운 결과물을 만들어낸 파생물에 대한 License 정책
 - 파생물(사전적 의미) : 하나 이상의 기존 저작물을 번역/편곡/희곡화/소설화/영화화/녹음/요약 또는 기타 방법으로 제작/개작/적용한 저작물

5) 오픈소스 소프트웨어 관점에서 파생물이 아닌 경우



- 그림의 A는 오픈소스를 사용하면서 내 소스코드에 포함시켜서 빌드를 하였기 때문에 파생물로 보고, C는 오픈소스에 내 소스코드에 내가 수정한 내용이 적용되었기 때문에 파생물로 봄
- B의 경우 오픈소스와 내 소스코드를 분리하였고 오픈소스 코드의 내용도 변화시키지 않고 사용
- 존재하는 API만을 이용하여 완전히 분리해서 빌드하거나 실행되는 것은 파생물이 아닌 것



오픈소스 License 종류

2. 주요 오픈소스 License

- 오픈소스 License는 기존에 진행된 대형 오픈소스 프로젝트를 중심으로 발전
- 이름을 따서 License 이름을 정하는 경우가 많음

	GPL	LGPL	MIT (BSD, MPL)	Apache
저작권 보호	O	O	O	O
상용 SW 혼용 가능	O	O	O	O
파생물 공개 의무	O	O	X	X
같은 License 유지	O	O	X	X
특허권행사 가능	X	X	X	O
비공개SW와 혼용	X	O	O	O

- Apache(Apache) License : 매우 느슨하게 정책을 하고 있어 다른 소스와 혼용할 수 있으나 파생물은 공개할 의무가 없어 매우 자유로운 License



오픈소스 License 종류

2. 주요 오픈소스 License

1) GPL

- GPL : 자유 소프트웨어 재단인 OSF에서 만든 License
- 주요 의무사항
 - 모든 소스 및 헤더 파일 배포
 - 컴파일 해서 나온 바이너리가 아닌 소스코드만으로 컴파일 가능하게 배포
 - 컴파일을 할 수 있는 설정인 Configure나 Makefile 등의 스크립트도 제공
- 공개 의무 범위
 - 소스와 링크되는 앱이나 모듈 등
 - 별도 프로세스로 동작해도 공유 공간에서 링크되면 공개
 - 객체지향 언어에서 GPL 소스 안의 클래스를 상속, 서브 클래스를 사용할 경우 모두 공개

2) LGPL

- LGPL : GPL의 너무 엄격한 규정으로 인해 활성화 되지 않자 보완을 위해 만들어진 License 규정
 - GPL과 동일한 자유 소프트웨어 재단에서 생성
 - GPL에 Lesser를 앞에 붙여서 좀더 느슨한 규정 생성
- 주요 의무사항
 - LGPL오픈 소스코드와 같이 컴파일 해서 사용할 경우 전체 소스 코드를 공개
 - LGPL 라이브러리로 변환 후 정적/동적 링크하여 사용해야 하며, 이 경우 LGPL소스 코드만 공개
- GPL과 LGPL의 공통점
 - 컴파일에 필요한 모든 스크립트 제공
 - GPL에서 LGPL로 발전하듯이 점점 소스코드 전체를 자유롭게 활용하는 추세로 변해가면서 오픈소스가 점점 더 발전



오픈소스 License 종류

2. 주요 오픈소스 License

3) MPL

- MPL
 - Mozilla 재단에서 브라우저의 소스코드를 공개하기 위해 만들어진 License
 - Firefox 브라우저에서 이 규정을 따르고 있으며 모질라 소프트웨어의 규모가 크기 때문에 잘 알려지게 된 오픈소스 License
- 주요 의무사항
 - MPL 오픈소스와 새롭게 개발된 코드가 컴파일 될 경우 MPL소스만 공개하고 MPL License 사용 고지 : MPL License의 오픈소스 사용 시 새롭게 개발된 부분은 소스코드 공개 제외, 사용여부만 고지하면 높은 자유도 보장
 - MPL 오픈 소스와 새롭게 수정한 MPL 오픈 소스만 공개하고, MPL 파일과 분리된 새로 추가한 파일은 공개 제외 : 상용 소프트웨어를 만드는 곳에서 MPL License의 오픈소스를 자유롭게 사용



오픈소스 License 종류

2. 주요 오픈소스 License

4) CPL/EPL

- CPL
 - IBM 사에 의해 만들어졌고, OSI와 자유소프트웨어 재단이 License로 승인
 - 오픈소스 협력 개발 환경을 지원하기 위해 타 License 환경에서도 소프트웨어와 콘텐츠를 사용할 수 있게 하는 CPL 콘텐츠 적용
- EPL
 - Eclipse 재단에서 자사 소프트웨어를 위해 만들어진 License
 - CPL을 대체하고 특정 용어 등과 관련된 특허 소송 침해 건을 제거하여 GPL보다 약한 상호주의 조항 보유
 - 기업 친화적인 오픈소스 License이며, Eclipse 재단에서 관리
- CPL과 EPL License 주요 내용
 - 배포 시에 License 사본 첨부
 - 소스코드 제공의무와 범위는 원소스와 모듈 단위로 구분
 - 저작권 고지를 해야 함
 - 타 License로의 배포를 허용하여 좀더 유연하게 운영
 - License가 특허 소송 제기 시에 License를 종료하도록 하여 기업 친화적인 License 정책



오픈소스 License 종류

2. 주요 오픈소스 License

5) Apache License

- Apache License
 - 웹 서버 오픈소스로 유명한 Apache 재단에서 자체적으로 만든 소프트웨어에 대한 License
 - 재배포 시 원본 소스코드 또는 수정한 코드는 공개하지 않아도 되므로 상용화를 위한 소프트웨어를 개발하는 기업에서 가장 환영할 만한 License
- 주요 의무사항
 - Apache License 소스 코드에 Apache License를 통해 개발된 소프트웨어라는 것을 명시
 - 새로 개발되는 소스코드를 공개하지 않고 상용화 서비스를 하려는 소프트웨어 프로젝트에서는 Apache License가 적용된 오픈소스를 활용

6) BSD License

- BSD License
 - 유닉스의 양대 뿌리 중 하나인 버클리의 캘리포니아 대학에서 배포하는 공개 소프트웨어의 License
 - GNP나 GPL보다 훨씬 개방적인 License로, 간단한 문구로 되어있어 매우 심플한 형태
- 주요 의무사항
 - 재배포 시 원본 소스코드 또는 수정 코드 공개 제외 : Apache License와 같이 매우 자유로운 형태의 License



오픈소스 License 종류

2. 주요 오픈소스 License

7) MIT License

- MIT License
 - 공대로 유명한 MIT 공대에서 자기 학교의 소프트웨어 공학도들을 돕기 위해 개발된 License
 - BSD License를 기초로 작성된 BSD 계열의 License 중 하나
- 주요 의무사항
 - BSD와 같이 자유도가 높아 많이 활용
 - 대표적으로 X윈도 시스템에서 사용

8) wxWidget License

- wxWidget License
 - 기본적으로 LGPL v2.0 혹은 이후 버전을 따르는 License
 - LGPL과 소스코드 공개 범위 동일
- 주요 의무사항
 - wxWidget을 링크 하여 사용하는 앱은 소스를 비공개 설정
 - wxWidget의 원본 코드와 수정된 부분의 소스 코드는 공개를 해야 하는 규정
 - 예외 조항에 의해서 오프젝트 코드 배포 제외

9) FIPL

- FIPL : MPL과 동일한 수준의 의무사항을 가지고 있는 License
- 주요 의무사항
 - NOTICE 파일을 통해 License 사용에 대한 고지
 - FIPL이 적용된 원본소스와 수정한 부분의 소스 공개
 - FIPL소스 파일과 분리된 새로운 추가한 파일은 공개 제외



오픈소스 License 검증 도구

1. 오픈소스 License 검증 도구

- License를 검토를 항상 수동으로 해야 하는 어려움 해결
- 현재 내 코드에서 오픈소스 License를 위반한 사항이 있는지 등을 체크하여 검출
- 사전에 License 문제 해결
- 도구가 완벽하게 신뢰하는 결과를 가져올 수 없기 때문에 항상 보조적인 수단으로 사용하는 것이 좋음
- 사용하지 않는 것 보다는 더욱 신뢰성을 높일 수 있기 때문에 오픈소스를 활용하였을 경우에는 도구를 사용하는 것을 추천



오픈소스 License 검증 도구

2. Protex

1) Protex 효과

- 현재 가장 많이 활용되고 있고 그에 따라 방대한 데이터베이스 확보
- 오픈소스에 대한 최신 정보를 바탕으로 실시간으로 검증을 수행하므로 신뢰도가 높아 많이 사용
- 개발시간 단축과 지적재산권 분쟁 위험 감소

2) Protex 검증 수행 절차



- **Protex의 기본 동작 형태**
 - 서버에 접속하여 최신의 License 정보를 실시간으로 체크
 - 인터넷이 연결되어 있는 상태에서 수행
 - 추가되는 수 많은 오픈소스를 실시간으로 지원하여 좀더 정확한 검증을 하기 위한 방법
- **Protex의 오픈소스 검증 데이터**
 - 지속적으로 5천 개 이상 수집되어 확장
 - 사용자 코드 컴포넌트를 구축하여 보안 취약점을 경고해 주는 기능
 - 매월 2~3회 이상 업데이트를 수행하여 항상 최신의 정보로 검증이 될 수 있도록 지원



오픈소스 License 검증 도구

2. Protex

3) Protex 검증 방법

- 정책수립
 - 공개되는 개발 프로젝트에 적용할 사용자 의무사항 정의
 - 의무사항에 부합되는 License를 생성하고 적용
- 컴포넌트 기획
 - License 의무 사항에 부합하여 상호 충돌 없이 활용할 수 있는 오픈소스 License 파악
 - 프로젝트 License에 부합하여 활용 가능한 오픈소스 프로젝트 및 결합방식을 검토
- 프로젝트 등록
 - 검증 서버에 접속하여 클라이언트 소프트웨어를 설치
 - 클라이언트 소프트웨어를 통해 검증 서버에 접속하여 해당 프로젝트로 등록
- 분석
 - 소스코드를 분석하는 단계를 수행
 - 분석 완료 후 프로젝트에 사용된 오픈소스 코드 확인
- 확인
 - 오픈소스 코드의 License 및 출처 확인
 - 사용된 오픈소스 코드의 결합형태 확인
- 시정조치
 - License와 충돌하는 오픈소스 코드를 확인
 - 해당 부분을 수정하거나 삭제할지를 선택하여 조치를 취한 후
 - 다시 한 번 검증을 수행하여 최종적으로 문제가 없을 때 프로젝트를 완료



오픈소스 License 검증 도구

3. FOSSology

1) FOSSology 오픈소스 License 검증 도구



2) FOSSology 활용

- 2007년 HP에서 시작
- 현재는 공개 소프트웨어 프로젝트인 sourceforge.net 에 GPL로 공개
- 누구나 무료로 사용, 기능 업데이트
- 문자열 검색을 통해 공개 소프트웨어 사용여부 확인 - 속도가 빠름
- 개발자 임의로 주석 삭제를 할 경우 판별 불가
 - 소스코드에서 오픈 소스 License가 어떠한 것이 존재하는지를 단순히 확인하는 수준에서는 활용
 - 신뢰도가 높지 않기 때문에 보조 수단으로 활용하는 것이 좋음



! 핵심정리



오픈소스 License의 이해

1. 오픈소스 소프트웨어란

- 자유롭게 누구나 열람하거나 수정이 가능하고, 다른 사람들에게 배포를 할 수 있는 소프트웨어
- GNU, BSD, Apache, MIT 등 70여가지 이상

2. 오픈소스 소프트웨어의 장점

- 특정 제품이나 제조사에 한정되지 않고 모두 지원하는 것이 목표
- 핵심역량을 발휘할 수 있는 부분에 집중
- 기업에서 인력 양성 및 리크루팅의 기회 부여



! 핵심정리



오픈소스 License 종류

1. 오픈소스 License 관리

- 주인이 없고 별도의 특허 등으로 보호받지 못하는 형태
- 운영 실수를 막기 위한 방안으로 자체적 License 정책 운영

2. 주요 오픈소스 License

- 오픈소스 License는 기존에 진행된 대형 오픈소스 프로젝트를 중심으로 발전
- GPL, LGPL, MPL, CPL/EPL, Apache License, BSD License, MIT License, wxWidget License, FIPL

! 핵심정리



오픈소스 License 검증 도구

1. Protex

- 현재 가장 많이 활용되고 있고 그에 따라 방대한 데이터베이스 확보
- 오픈소스에 대한 최신 정보를 바탕으로 실시간으로 검증을 수행하므로 신뢰도가 높아 많이 사용
- 개발시간 단축과 지적재산권 분쟁 위험 감소

2. FOSSology

- 2007년 HP에서 시작
- 현재는 공개 소프트웨어 프로젝트인 sourceforge.net 에 GPL로 공개
- 누구나 무료로 사용, 기능 업데이트
- 문자열 검색을 통해 공개 소프트웨어 사용여부 확인
- 개발자 임의로 주석 삭제를 할 경우 판별 불가