"Free" Software Isn't Free:
With Open Source Software, Knowledge (and Oversight) is Power

Almost every company building software applications uses open source software ("OSS"). However, leveraging OSS to develop your company's core software, without paying attention to the obligations required by the OSS licenses, comes with significant risks. Imagine your product takes off and you are poised to be the next big ticket acquisition. Your exit strategy is about to pay huge dividends. During due diligence, your acquirer realizes you used OSS and did not comply with the obligations. It looks into the OSS license requirements and finds that some or all source code that you developed must be made available to the public on the same license terms as the OSS you used, simply because of the way you used the OSS. Now, the product you invested so heavily in must be made available to the public – for free. All of the time, effort, and money you (and your investors) put into developing a proprietary product is lost and your exit strategy is at risk.

> Think about open source early, or pay the price later.
> Undoing a mistake is much more difficult doing it right
> in the first place.

Software-based startups have a huge advantage today. They can write their code faster using existing OSS packages or frameworks. As this story demonstrates, careful consideration should be given to what license terms and requirements apply to the OSS you use (as well as any third party code you use, though this resource will focus on OSS), whether you can live with the obligations that accompany your use of the OSS, and how you can protect your company long term from potentially devastating consequences of using OSS without paying attention to the applicable terms and requirements.

Here we provide some simple steps to help you navigate that risk.

What is OSS?

Source code defines a software company's products – it is closely guarded and may be protected by copyrights, non-disclosure agreements, and/or patents. Software is typically not distributed to others in source code form. Even in patent litigation, if companies have to provide the entirety of their source code for review, deliberate and

additional steps are taken to protect any source code that must be reviewed from public disclosure, because it is so valuable and loses its value when shared.

Open source software is software in source code form that its author has decided to allow others to use without paying royalties (and without violating copyright laws), as long as users comply with certain requirements specified by the associated license. These requirements differ widely – some authors get quite creative in what they require in exchange for using their OSS – but most licenses fall into a few common categories.

Types of OSS License Requirements

OSS licenses allow the OSS to be used, modified, and distributed, but each type of use may have different requirements associated with it, with varying levels of requirements for companies who incorporate OSS into their software. Common OSS license requirements include:

- Providing a copy of the OSS license with any software that contains it, as well as a disclaimer of warranties;
- Providing attribution to the authors of the OSS;
- Providing a license equivalent to the OSS license with any distributed software containing the OSS;
- Providing any modifications made to the OSS;
- Making the source code for any software incorporating the OSS available under the same terms as the OSS.

The most common OSS licenses are Apache, BSD, MIT, GNU General Public License (GPL), GNU Library or Lesser Public License (LGPL). However there are hundreds of variations on these licenses in the wild.

Steps to Ensure Compliance with License Requirements

The best time to set up a process for ensuring compliance with OSS license requirements is when your company is small and starting out. Later, it can be challenging to untangle various versions and types of OSS used throughout the years.[1] Proactively managing OSS use from the start increases the likelihood of compliance with license requirements and maintaining control over your software, and lessens engineering time spent fixing problems (such as identifying OSS in your source code, unidentified security vulnerabilities in out of date OSS, reworking code into a new package to get it in compliance).
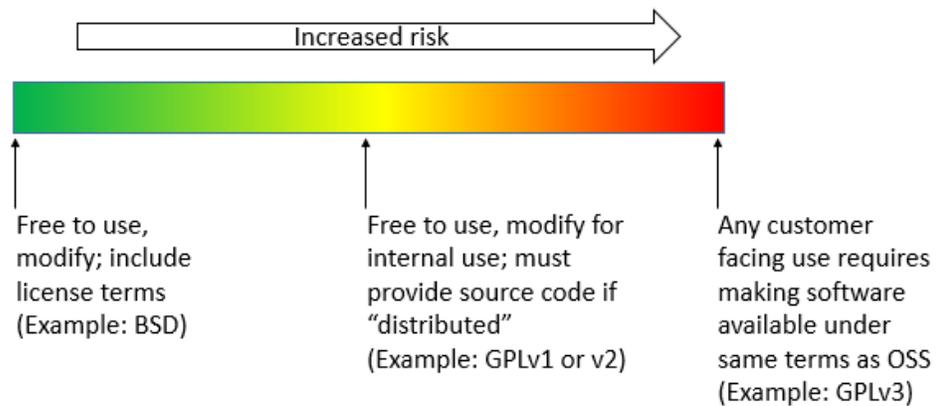
Step 1: Identify all OSS you have used or plan to use. This is true for all third party code.

Step 2: Identify where and how you are using the OSS, whether you are integrating it, modifying it, etc.

---

[1] It is not impossible to untangle the web of OSS even late in your software development. For particularly complicated situations, vendors such as Black Duck, OpenLogic, Palamida, and other companies have sophisticated tools that can identify the OSS used in your software. HIPLegal is not affiliated with any of these companies.

Step 3: Review the associated OSS licenses so you know the terms and requirements.

Step 4: Evaluate the level of concern, risk, and effort it will take to comply with the varying terms and requirements:

Increased risk

Free to use, modify; include license terms (Example: BSD)

Free to use, modify for internal use; must provide source code if "distributed" (Example: GPLv1 or v2)

Any customer facing use requires making software available under same terms as OSS (Example: GPLv3)

Step 5: Decide whether or not to use any given OSS in view of the business case associated with compliance with those terms and requirements.

Step 6: Set up, maintain, update, and pressure test internal processes for authorizing and tracking use of OSS, modification of OSS, versioning of OSS, versions of the license associated with the OSS, license terms and requirements, following compliance checklists, identity of the owner / responsible party, and enforcement of the terms and requirements.

Step 7: Educate, Train, Enforce. Educate your employees about the OSS requirements and your process for evaluating whether to use OSS before they incorporate it into your software. Train your employees on your OSS tracking and compliance process. Enforce your policy by including it in the software development lifecycle. Check in with each step of your process regularly to ensure your OSS process is being followed and to answer any questions about it.

Ignorance is not bliss when it comes to OSS – but with knowledge and careful management, OSS can be a valuable tool in the development of your products.

HIPLegal can help – give us a call if you have OSS questions or need help setting up processes or training employees on OSS processes in your company.

Note: This is not intended to provide legal advice, as every situation has different facts, and it does not create an attorney-client relationship.

HIPLegal LLP
20195 Stevens Creek Blvd., Suite 250
Cupertino, CA 95014 | 650-397-6447
contact@hiplegal.com | **www.hiplegal.com**

HIPLegal