On the Verge of an M&A? Don't Ignore Open Source Due Diligence

WhiteSource

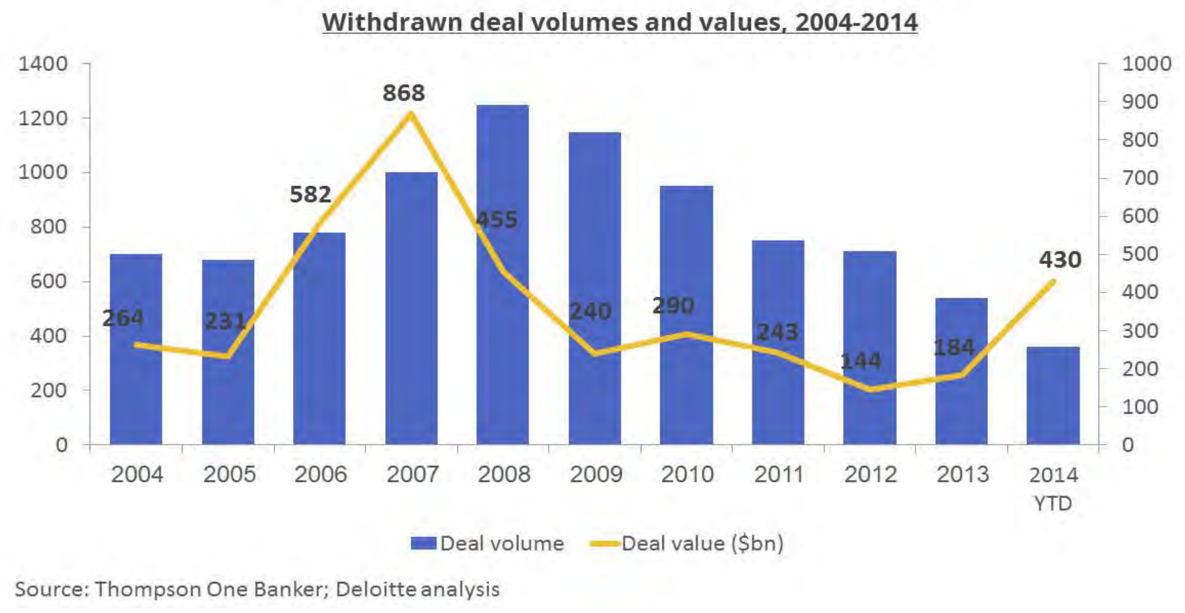
When a merger and acquisition process is happening, the acquiring team's first goal is to assess and understand the value of the acquired company. This is achieved by a full and thorough investigation of the company's financials, technology, litigations, sales, and all other relevant pieces of information. This investigation procedure is the due diligence process. Due diligence is done to ensure that the acquiring team knows everything there is to know before signing the contract

Remember that coffee is for closers

Your purpose as a seller is to close – and close quickly!

Typically 20% of global M&A deals are withdrawn or terminated and percentages increase in smaller acquisitions. The consequences of coming unprepared might impact the confidence of the acquiring team in the company's worth; furthermore, failed transactions often have lasting consequences on the 'almost acquired' company image. Withdrawn deals in 2014 reached $430B, which is the highest number since 2008.

It is no secret that time can be a major deal-breaker, so your #1 goal should be to prepare in advance in order to reduce due diligence duration.



Successful M&A starts with great due diligence reports

The due diligence process is a significant step in building the trust between the two parties involved in the M&A.

Software due diligence is a crucial part of each software company's due diligence process and it is usually the longest process. Implications can occur in every piece of code, so companies are asked to open their code inventory reports, elaborate on the development and deployment environments, third-party libraries, patch tools, and licenses.

The main goals of the software due diligence are to assess the value of the product and know if there are any red flags. There are 2 types of red flags – correctable ones and those that cannot be corrected. A correctable red flag is something like lack of documentation that you can fix. An uncorrectable red flag (which typically means walk away) can be like that of violating some third-party intellectual property.

Software due diligence is a bit like having a home inspection done when purchasing a house. Some problems are more serious than the others. For example, if you find out that there is mold or asbestos in the basement — you will be forced to reconsider.

So how should you prepare your software for a quick and successful audit with no red flags?

Being proactive is the key

Being proactive is critical as it can significantly reduce the due diligence duration. First, you have to create accurate and up-to-date inventory list of your software. Almost every commercial software product today consists of proprietary code, third-party libraries and open source components.

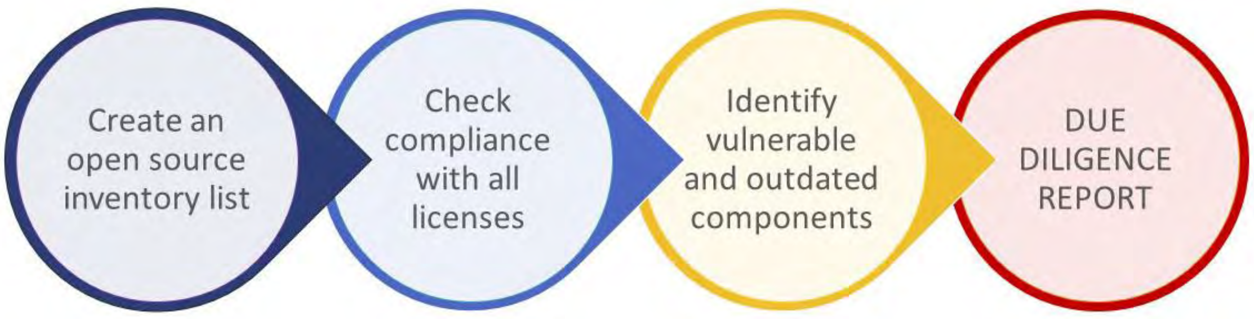
Start by organizing your code in a systematic top-down structure to identify all the proprietary, commercial and open source components as each group will have similar compliance characteristics. And then prepare accurate and updated software Build of Material (BOM) for each group separately.

Brace yourself for handling open source - you might be using more of it than you estimate

It is no surprise that everyone wants to leverage open source. But open source comes with responsibility — like licenses to track, security vulnerabilities to manage, and updates to make. This puts open source reporting on the center stage during M&As. Acquiring companies are worried to inherit known open source issues, like losing their proprietary rights, breaking licensing terms, security risks, and other patent implications. The 2015 future of open source survey shows that only 27% of companies have a formal open source policy to approve new components that are added to their software.

Remember that unmanaged use of open source can not only delay processes, it can also reduce the value or lose the entire deal.

3-step Open Source Due Diligence Checklist - Here's How to Prepare


Create an open source inventory list
Check compliance with all licenses
Identify vulnerable and outdated components
DUE DILIGENCE REPORT

Step #1 - Create an open source inventory list

This is the most crucial step since without knowing what components you use, you cannot know what open source licenses you are required to obey, if any security vulnerability affects your product and if you should consider updating your libraries with newly released versions.

You need to identify and list out all the open source components that you use, including all dependencies. With this exercise, you will be able to identify the relevant licenses, security vulnerability and quality issues relevant to your products.

There are 2 methods you could use for this step:

The manual Do-It-Yourself method

The manual Do-It-Yourself method does not work great for a due diligence process, since accuracy is of great importance and manual work is error prone by default. Also, it is almost impossible to manually track all dependencies of all your open source libraries. Dependencies are critical as a library may have a different license that one of its dependencies, but you are still obligated to comply with it.

In the manual method, you ask your developers to keep a spreadsheet of all open source components and to update it every time they add or remove an open source component, but there are always significant gaps between what was documented and the reality. You can read this resource to see why manually tracking open source components is so error prone.

The automated method

There are two main technologies that can help you automatically identify all your open source components. There are the code scanners, which scan your code for snippets. The problem with this technology is that it is not continuous and you may find critical error minutes before release when it is complicated and expensive to fix.

There are also build process monitoring tools, like WhiteSource, which automatically discovers all open source components in your build within minutes. This technology offers real-time alerts when a problematic component is added to the build, or when a security vulnerability is discovered. Check our comparison table to learn more about the difference between code scanner and WhiteSource.

Step #2 - List all the identified open source licenses and check compliance

Once you've identified all the open source components that you're using, you are required to identify all the licenses that your components have, and their dependencies. At this point, you'll have to carefully check what is it that each license requires you to do and whether or not your company meets these requirements.

In a due diligence process, the acquiring team would like to close a deal with a clean slate, and, of course, without purchasing any legal implications. Therefore, the acquiring team will carefully check the software licenses and if your company is in full compliance with all open source licenses.

Why care for open source licenses so much - isn't open source free?

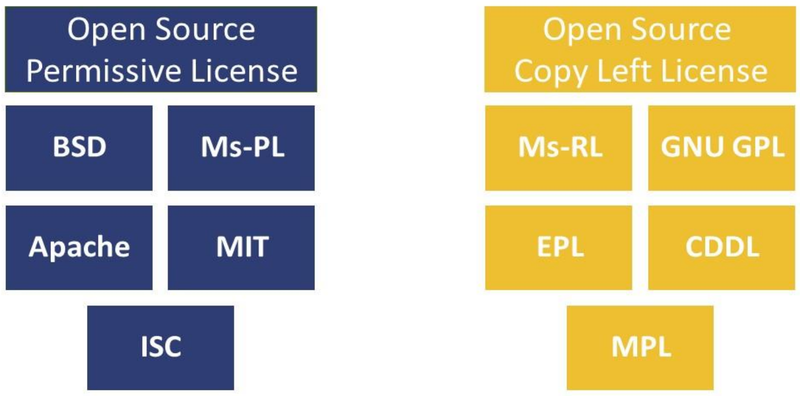
There is a big misconception about open source. It is believed to be free in the sense that it has no strings attached, but it is actually only free of charge (free beer vs. free speech). In fact, open source components are, by law, commercial items. They are indeed free to use, but they come with a license which you have to comply with to avoid legal and business risks.

There are two main types of open source licenses: permissive and copyleft. Permissive open source licenses have minimal requirements regarding how it could be redistributed. Copyleft licenses, like the GNU GPL, are a different story.

Just like copyright is a law that restricts the right to use creative works without the permission of the author. When an author releases a program under a copyleft license, he makes a claim on the copyright of the work and issues a statement that other people have the right to use as long as the reciprocity obligation is maintained.

This means that any software that is written based on any component, which are copyleft licensed, must be released as open source. The result is that such software is required to release its full source code and all of the rights to modify and distribute the entire code. This might substantially reduce your company's value in the eyes of the acquiring company, as it as has severe implications on your ability to protect your intellectual property.

Here are the most common examples for permissive and copyleft licenses:



Step #3 - Identify vulnerable and outdated components

Security vulnerabilities might exist in your software without you even being aware. Uncovering these issues during an M&A process can damage your process, as security vulnerabilities can be exploited by malware or hackers. Monitoring and revealing security vulnerabilities require substantial time and efforts, and its inheritance increases the risk to the acquiring team.

Provide accurate data with minimal efforts.

Within the due diligence process, you must not rest on laurels; usually, when an offer is made, there is a short window of time to complete the process successfully, and your goal is to provide accurate data. Fast. With minimal effort.

When both the company and the acquiring team are fully aware of the licenses used and existence of security and quality issues, they can proceed with the deal quickly and efficiently; therefore, prior preparation for due-diligence is highly crucial, as it will save a lot of valuable time (and headaches).

Once you have prepared your open source inventory list and verified compliance with your open source licenses — your next step is to see if you are exposed to any known security vulnerabilities due to your open source components.

How to identify open source security vulnerabilities?

The NIST database maintains a record of all the reported open source security vulnerabilities. You need to compare your open source inventory list with the NIST repository and check if any of the reported vulnerabilities apply to your product.

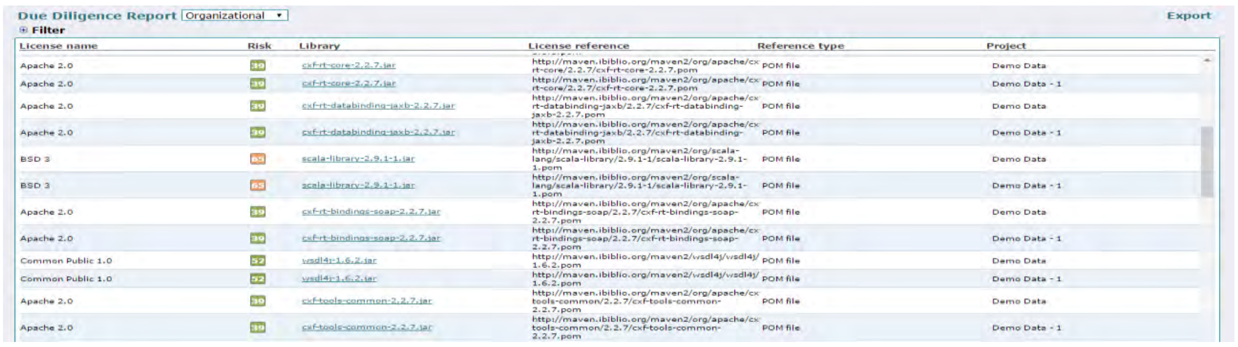
Manually tracking all CVEs is almost impossible as thousands of security vulnerabilities get published every year in the NIST database. Finding out those that affect the specific versions of the specific open source components you use can be extremely laborious and the output will surely be error-prone.

As with creating inventory list. Most companies tend to choose an automated tool for generating due diligence reports, since the report's accuracy has legal implications. Both technologies of code scanning and build process integration, which generates the inventory reports, applies here as well. Once these tools identified your inventory they cross reference the results with the NIST database and immediately alert you when a match is found.

The good news is that open source communities are often quick to fix vulnerabilities and bugs, so most chances are a new patch or version exists that can fix a reported issue. This is why WhiteSource alerts you if there's a new version or patch that can remediate a security issue.

No Time to Prepare for the Upcoming M&A? Try WhiteSource Today

WhiteSource is the only solution in the market that offers 100% accurate and comprehensive due diligence reports within minutes. Our solution can integrate with your build tool/server (we support all popular programming languages and build tools) and auto-discover all your open source components including dependencies. It later pulls all the relevant information about your libraries and generate a due diligence report with a full license reference for each library, as required. Our reports cover all open source related topics right from licenses to security and quality.



License name	Risk	Library	License reference	Reference type	Project
Apache 2.0	39	cxf-rt-core-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx rt-core/2.2.7/cxf-rt-core-2.2.7.pom	POM file	Demo Data
Apache 2.0	39	cxf-rt-core-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx rt-core/2.2.7/cxf-rt-core-2.2.7.pom	POM file	Demo Data - 1
Apache 2.0	39	cxf-rt-databinding-jaxb-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx rt-databinding-jaxb/2.2.7/cxf-rt-databinding-jaxb-2.2.7.pom	POM file	Demo Data
Apache 2.0	39	cxf-rt-databinding-jaxb-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx rt-databinding-jaxb/2.2.7/cxf-rt-databinding-jaxb-2.2.7.pom	POM file	Demo Data - 1
BSD 3	65	scala-library-2.9.1-1.jar	http://maven.ibiblio.org/maven2/org/scala-lang/scala-library/2.9.1-1/scala-library-2.9.1-1.pom	POM file	Demo Data
BSD 3	65	scala-library-2.9.1-1.jar	http://maven.ibiblio.org/maven2/org/scala-lang/scala-library/2.9.1-1/scala-library-2.9.1-1.pom	POM file	Demo Data - 1
Apache 2.0	39	cxf-rt-bindings-soap-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx rt-bindings-soap/2.2.7/cxf-rt-bindings-soap-2.2.7.pom	POM file	Demo Data
Apache 2.0	39	cxf-rt-bindings-soap-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx rt-bindings-soap/2.2.7/cxf-rt-bindings-soap-2.2.7.pom	POM file	Demo Data - 1
Common Public 1.0	52	wsdl4j-1.6.2.jar	http://maven.ibiblio.org/maven2/wsdl4j/wsdl4j/1.6.2.pom	POM file	Demo Data
Common Public 1.0	52	wsdl4j-1.6.2.jar	http://maven.ibiblio.org/maven2/wsdl4j/wsdl4j/1.6.2.pom	POM file	Demo Data - 1
Apache 2.0	39	cxf-tools-common-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx tools-common/2.2.7/cxf-tools-common-2.2.7.pom	POM file	Demo Data
Apache 2.0	39	cxf-tools-common-2.2.7.jar	http://maven.ibiblio.org/maven2/org/apache/cx tools-common/2.2.7/cxf-tools-common-2.2.7.pom	POM file	Demo Data - 1

Start your free trial with WhiteSource today to see if you have any red flags in your open source components and ensure you are prepared for your upcoming M&A. You can also submit your details for a free 30 minutes consultation with our open source due diligence experts.

For more details regarding open source due diligence, please watch our 'Due Diligence Made Easy' webinar where one of our founders, Dr. Ron Rymon, a serial entrepreneur, shares his experience as an entrepreneur and investor going through two different open source due diligence processes as part of two M&A transactions.

About WhiteSource

WhiteSource enables engineering, security and compliance officers to effortlessly manage the use of open source components in their software, allowing developers to focus on building great products. WhiteSource fully automates all open source management needs: component detection, security vulnerability alerts, license risk and compliance analysis along with policy enforcement and post-release tracking. Designed for engineering executives and security officers, a complete suite of control, reporting and management tools make managing open source truly effortless. Follow us on twitter: @whitesourcesoft or join our LinkedIn page