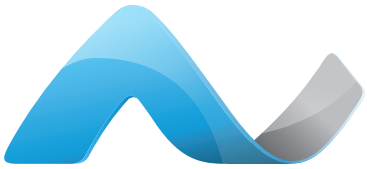


Antelink

Getting full benefits of open source software

Antelink S.A.S
http://www.antelink.com
contact@antelink.com
twitter:@antelink
+33 1 42 39 30 78
18 Rue Yves Toudic,
75010 Paris,
France

Summary

Organizations get many benefits from open source at a very low cost. Open source allows them to ship software faster and cheaper while maintaining high quality. Yet, if not well managed, open source can entail hidden costs due to license compliance issues and software vulnerabilities (OSS risks).

Existing solution meant to mitigate these risks arrive too late during the software lifecycle and increase the costs associated with correcting license or vulnerability issues. Furthermore, current solutions base their analysis on unreliable techniques that produce too many false positives and thus increase the cost of mitigating risks.

Antelink proposes an end-to-end solution to manage license compatibility, upgrade and vulnerability risks. The Antepedia Suite helps you to find quickly and cost-effectively the OSS risks introduced during your software development lifecycle.

Pervasive open source

Enterprises get many benefits from open source at small cost. Open source enables companies to deliver software faster,cheaper, and with higher quality.

This is the main motivation that moves companies towards the adoption of open source software. According to Accenture[1], about 80% of the large companies have adopted open source. Furthermore, for almost every day-to-day requirement there is an open source component that can be re-used.

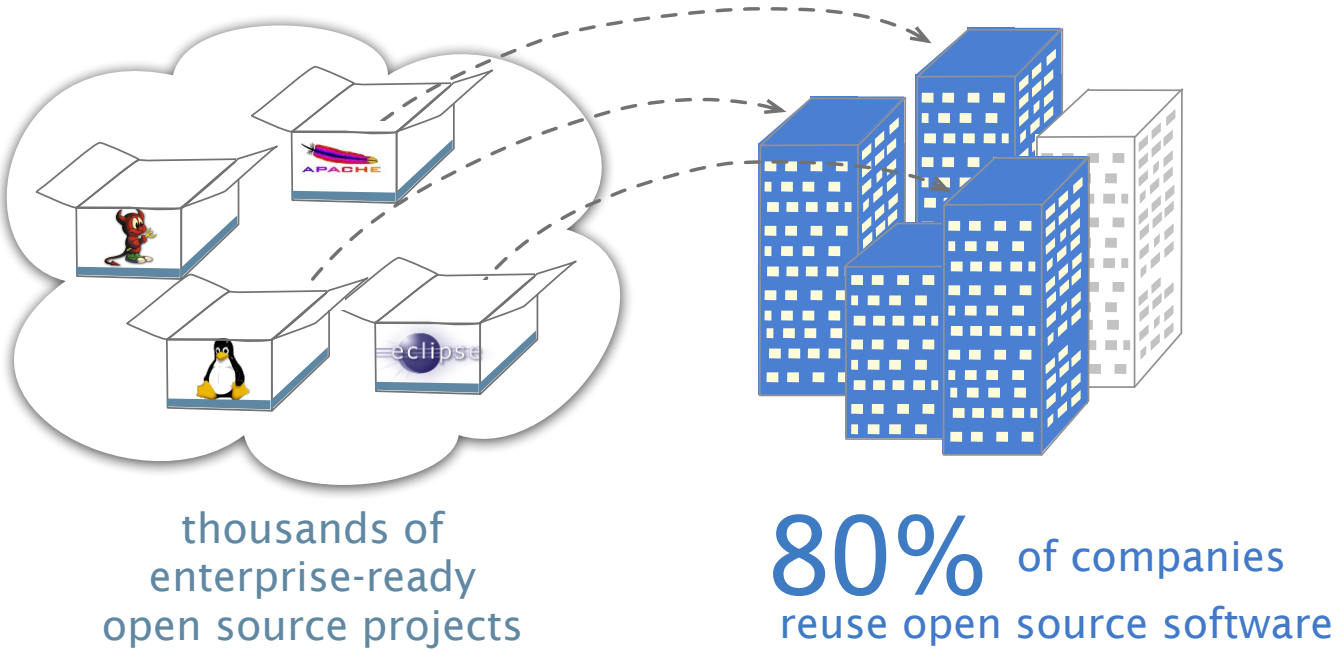


Figure 1: 80% of the large companies have adopted open source.

"A recent survey by Gartner, Inc. found that more than half of organizations have adopted open source software (OSS) solutions as part of their IT strategy. Nearly one-third of respondents cited benefits of flexibility, increased innovation, shorter development times and faster procurement processes as reasons for adopting OSS solutions." [2]

Open source Reuse Challenges

Despite the benefits brought by open source, if it is not well managed it can entail hidden risks that can compromise a company software assets.

These risks correspond to **license compliance**, **obsolescence**, and **vulnerability issues**. They derive from the oblivious usage of open source components and happens all along the development process, from coding to distribution.

> "Oblivious use open source components can compromise a company's software assets"

Many companies have taken actions to mitigate some of these risks, particularly license compliance related risks. They have introduced license policies and review procedures meant to guide those involved in the development process to comply with open source license (and the company licensing schema). Yet, these policies and procedures are seldom enforced, and when enforced it is too late in the development process. Thus, when a compliance issues is detected it implies elevated correction costs or event the re-development of the software itself.

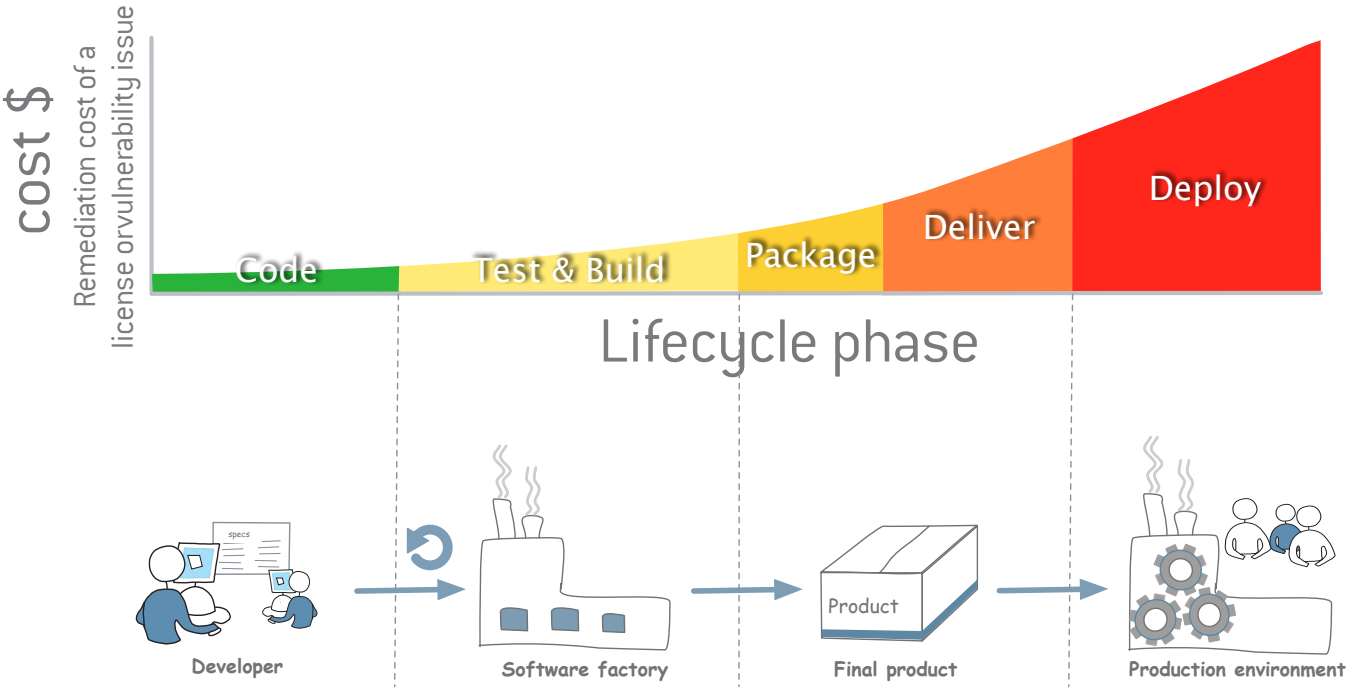


Figure 2: The later an issue is detected, the higher its correction cost will be.

License Compliance Risks

Reusing preexisting code or components leads to a series of legal issues that must be carefully considered. Dealing with the legal aspects of collaboratively developed software is difficult and time-consuming.

First, the wishes of the original author must be respected. They are usually expressed through a license agreement such as GNU GPL, Apache Software License, etc. Implies complying with the legal obligation imposed by the software author.

Second, available license information may be not trustable or change in different version of the product releases. This makes it difficult to decide which obligations to comply with. Furthermore, a single software can include tens of components that can induce secondary dependencies.

Third, the originating project and authors of a component may be unknown or difficult to track. Therefore, discussing with the software authors and knowing their real wishes may not be possible.

The most important risk introduced by license compliance is “not complying” with the original author wishes. The risk here consists in endangering not only software assets, but also hardware equipment built on top of it.

For example (see figure 3), on December 14th 2009, BusyBox, an open source linux tools vendor filled a lawsuit against Westinghouse for license violation. Indeed, Westinghouse was manufacturing HDTVs that included a set of modules from BusyBox. Later, on August 3rd 2010, the court ruled in favor of BusyBox for willful copyright infringement and granted it damage compensations and an injunction against Westinghouse. As a result of this, Westinghouse lost compensation money (about $ 150.000), lost revenue due to the injunction, and lost inventory corresponding to millions of dollars (all HDTV were donated to charity)[3].

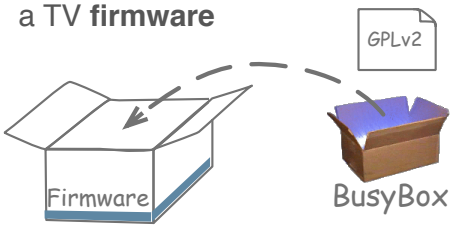
The case of Westinghouse and BusyBox exemplify the potential damages that violating license obligations can produce. Risks like this could be mitigated early enough, without the need for a lawsuit.

Can prevent

If the developers would have been aware of BusyBox license, it could have been replaced

BusyBox included in the firmware

Developers **included BusyBox** sources into a TV **firmware**

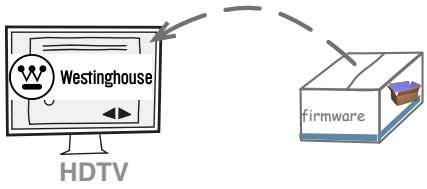


Can prevent

If Westinghouse engineers would have been aware of the presence of BusyBox in the firmware, they could have removed it

Westinghouse includes BusyBox into its HDTV

Westinghouse includes BusyBox modules **into** the firmware of its **HDTV**



Too late

At this point it was too late. Westinghouse have violated the GPLv2 license

BusyBox fills a lawsuit against Westinghouse

December 14th 2009 - BusyBox fills a **lawsuit** against **Westinghouse**
BusyBox engineer found about the BusyBox components in the Westinghouse HDTV.

Settlement: Westinghouse assessed damages, lost revenue,and lost inventory

August 3rd 2010 - Settlement: Westinghouse **assessed** $150.000 in **damages** , **lost revenue**, and millions of dollars of **inventory** lost (all HDTV were donated to charity)

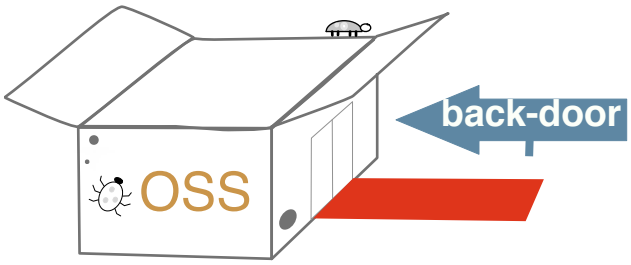
Figure 3: BusyBox, an open source linux tools vendor filled a lawsuit against Westinghouse for license violation.

Vulnerability Risks

There is no such thing as perfect software and open source software is not the exception. Fortunately, open source communities release patches for critical vulnerabilities very quickly. However, users are not always aware either of patches, or the vulnerabilities.

> "There is no invulnerable software only those whose vulnerabilities have not been discovered yet."

The major risk introduced by vulnerabilities is not the vulnerability itself, but you not being aware of them, thus they compromise your software assets without you noticing it. And, although several vulnerability databases are publicly available[4], searching each reused components is very time consuming, and therefore vulnerability reports are often ignored.



> "Upgrading is the best way to mitigate vulnerability risks."

For example, consider the case of the Spring Framework, a widely used inversion of control framework for Java™. On October 28th 2010, a vulnerability report was disclosed [5] for this framework's security module +ranging from versions 2.0.0 to 3.0.3. This vulnerability allows intruders to bypass the framework security mechanisms and gain access to sensitive information. Over a year after the vulnerability was disclosed, thousands of open source and close source commercial product were using the affected framework versions.

Obsolescence Risks

Unmaintained open source software introduces risks related to evolution and vulnerability fix. Chances are that bugs and vulnerabilities in an unmaintained component will never be solved and new functionalities never added.

> “Unmaintained software Introduces heavy risks.
> If vulnerabilities are found,
> they are not likely to be corrected.”

Typically, you will notice that a component is unmaintained when you find that there is no community support and you need to report/ solve a bug.

When you include unmaintained components, you have the choice to maintain it yourself, or to switch to an equivalent maintained components. Either invest money maintaining and driving a software component or search for an alternative equivalent component that may not exist.

Obsolescence can happen because of several reasons, loss of traction, loss of incentive, loss of a vital contributor and community member stimulator.

> “Knowing when a project is unmaintained mitigates obsolescence risks.”

Knowing too late that you're using a obsolete component may imply:

- No bug fixing and no new feature
- No support for latest releases of platforms, frameworks, languages, etc.
- Lack of support when a bug is found, or difficulty to correct the problem by yourself.
- Increased risks, if you have to upgrade in a hurry.

Risk assessment and responsible OSS reuse

The risks that come from reusing open source software (and any software product in general) should not be a source of distrust. Companies can mitigate these risks early enough so they and the open source community could get full benefits. Yet, companies have troubles doing this.

Existing solutions meant to mitigate risks arrive too late during the software development process. They usually adopt a corrective rather than a preventive approach, thus increasing the costs associated with the correction of license or vulnerability issues. Furthermore, some solutions base their analysis on :

(1) Unreliable techniques that produce too many false positives and thus increase the time consumption of these tasks.

(2) Incomplete data sources that leave important components out of the analysis.

In order to perform efficient risk assessment you need to use the **right tools** at the **right time** in your **software lifecycle**, and according to the **maturity stage** of your software.

Reducing reuse risks is not only a way to protect your company's software assets but also to make a responsible use of open source and respecting the open source communities.

"Support and get full benefits of open source software:
Care about licenses,
Track updates, and
Watch for Vulnerabilities."

Antelink's Approach

At Antelink, we strongly promote the adoption of open source software and emphasizes that risk associated with it should be mitigated early in the development process. Likewise, we think that everyone along the development process should be involved in this task, and we provide the tools to do so.

The world's largest knowledge base

In order to address License, Vulnerability, and Obsolescence risks, we have build the world's largest knowledge base of open source projects. We have collected the content of publicly available source hosts (see figure 4) such as Source Forge, Google Code, The Eclipse Foundation, etc.. and distribution places such as Linux distributions, The Maven Central repository, Eclipse update Site, etc .

Combined, these sources contains several hundred thousand of open source Project. Antepedia — Antelink's Knowledge Base of open source Projects — contains more than a million open source projects, which aggregate about 500 million files and growths at a rate of 1000 new projects each day.

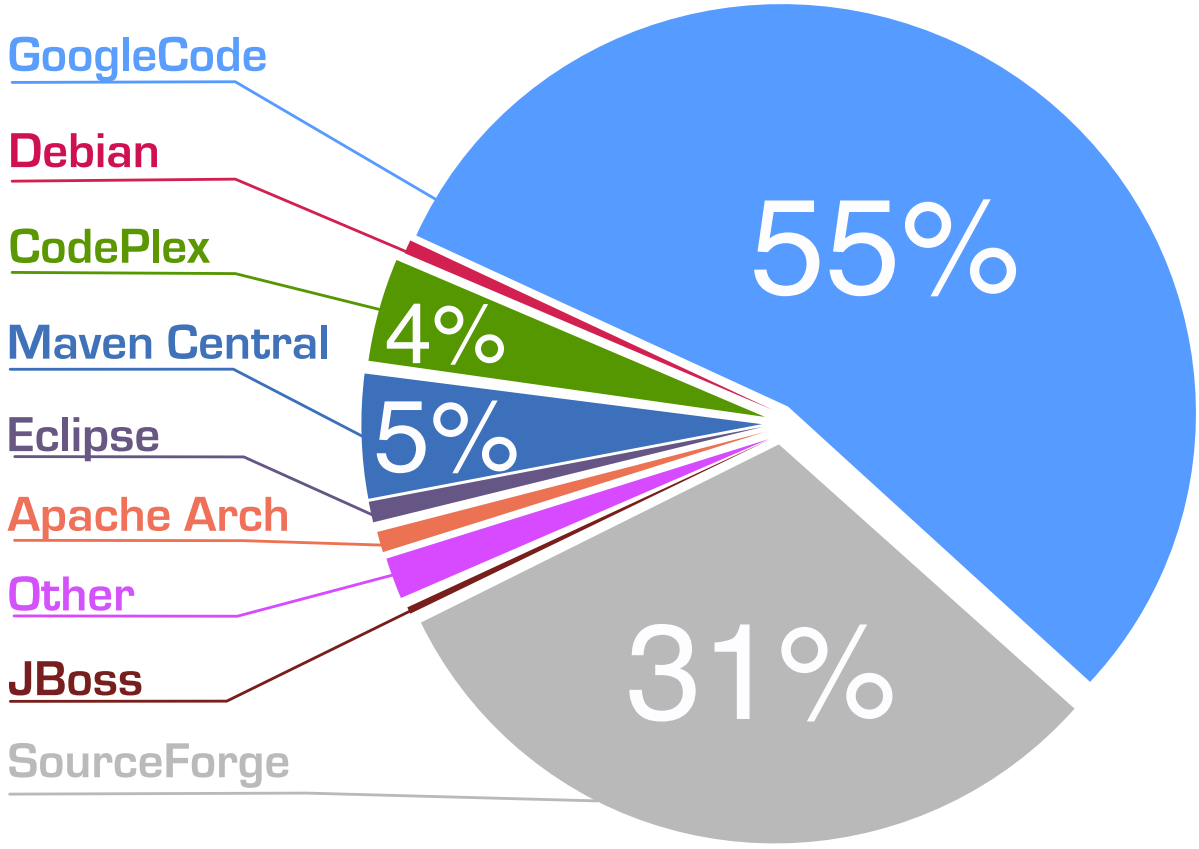


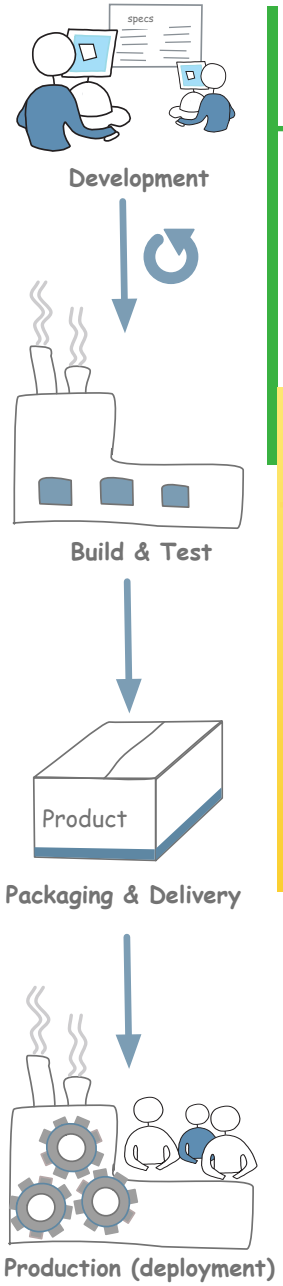
Figure 4: Distribution Antepedia project sources.

Antepedia — An integrated solution

Antelink provides a integrated solution that addresses the needs that arise during the whole software lifecycle, from developers to managers and from development throughout production.

Antelink tools exploit Antepedia— Antelink's knowledge base — to provide license, vulnerability, obsolescence, and update information. It is composed of four main products, each of them fits the information needed at the different stages of the software development process (see figure below)

"Efficient risk assessment means that you use the right tools at the right time in your software lifecycle according to the maturity stage of your software."



Antepedia Developer: A solution dedicated to developers. It is an extension to the popular IDEs (**Eclipse**, **IntelliJ IDEA**, **etc**.) and empowers developers to be aware about file licenses, component vulnerabilities, and updates. It helps you enforce your licensing policies and avoid vulnerabilities.

Antepedia Notifier: A solution dedicated to build engineers. It introduces the concept of *continuous open source detection*. You plug it into your SCM (Git, SVN, or Mercurial) and it analyses your files commit after commit. It generates summary reports and notifications (transmitted either through e-mail or JIRA™) about licenses, vulnerabilities, and updates.

Antepedia Auditor: A solution dedicated to code and package auditors, and managers. It analyses a bulk of files and generates an audit report with license, vulnerabilities, and update information.

Antepedia Search: A publicly available service for every one. It analyzes one file at the time and gives license and origin information.

The Antepedia Suite delivers early risk assessment for safer open source software reuse. It enables you to produce software product fast, harness all the power of open source without headaches, and reduces your risk assessment and legal bill.

With the Antepedia Suite you will:

- Execute accurate license compliance audit and Intellectual property (IP) right management.
- Enforce your license compliance policy from day one.
- Generate periodic Bill-of-Materials (BOM) reports.
- Improve collaboration between developers, project managers, and the legal department of your organization.
- Integrate open source detection early in your development process
- Reduce the costs associated with the correction of license compliance issues

"Be pro-active, empower everyone involved in the software lifecycle to mitigate risks that can doom your software assets."

To learn more about Antelink and the Antepedia Suite, visit http://www.antelink.com

---

[1] Accenture Open Source Survey 2010.
http://www.accenture.com/us-en/Pages/insight-open-source-survey-2010.aspx

[2] Gartner Survey Reveals More than Half of Respondents Have Adopted Open-Source Software Solutions as Part of IT Strategy http://www.gartner.com/it/page.jsp?id=1541414

[3] Best Buy, Samsung, Westinghouse, And Eleven Other Brands Named In SFLC Lawsuit
http://www.softwarefreedom.org/news/2009/dec/14/busybox-gpl-lawsuit/

[4] The National Vulnerability Database http://nvd.nist.gov/
The Open Source Vulnerability Database http://osvdb.org/ and others.

[5] Spring Security URL Path Parameter Constraints Bypass
http://osvdb.org/show/osvdb/68931