» The Open Compliance Program

A Five-Step Compliance Process for FOSS Identification and Review

By Ibrahim Haddad (Ph.D.), The Linux Foundation

A White Paper By The Linux Foundation
http://www.linuxfoundation.org

Introduction

This white paper is one in a series that focuses on the various practical aspects of ensuring free and open source software (FOSS) compliance in the enterprise. This paper provides an example of a compliance process for FOSS identification and review that consists of five steps. The focus of the paper is around using and integrating FOSS with proprietary and third party source code in a commercial product.

The goal with the FOSS compliance process is to ensure that any software (proprietary, third party, FOSS) that gets into the product base has been audited, reviewed and approved and that the company has a plan to fulfill the license obligations resulting from using the various software components integrated in the product. This type of compliance due diligence is often tracked and executed via such a process.

Simple FOSS Compliance Process

The 5-step identification and review process includes the following phases (Figure 1):

- Scanning the source code
- Identifying and resolving any discovered issues
- Performing license review
- Performing architectural review
- Deciding the approval for this software component

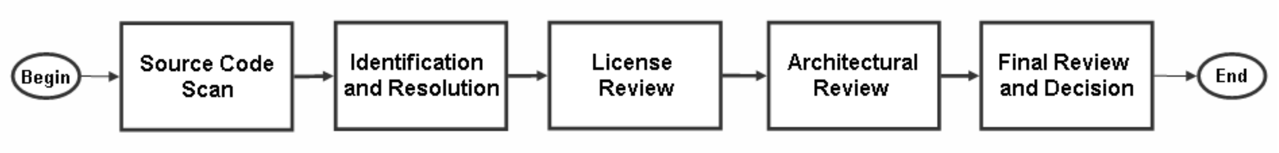

Figure 1: Example FOSS compliance process

The process is not completely linear going from source code scan to final review. You can create various workflows from each process phase to any other process phase. Some of these are illustrated in a later section.

Source Code Scan

In this phase of the process, all the source code is scanned using a source code scanning tool (see “Tools” section). There are several factors (Figure 2) that can trigger a source code scan, which include:

- An incoming request from Engineering to use a FOSS component. This incoming request often comes to the compliance officer via an online form that Engineering staff use when they request approval to use external source code.
- A periodically scheduled full platform scan. Periodic full platform scans are very useful to uncover FOSS that sneaked its way into the software platform without a formal request form that helps track the usage of the component and its obligation fulfillment.

- A usage change in a previously approved software component: In many cases, a software component is approved for usage in a certain product in a given technical context. If the usage type changes or the component is to be used in a different product, the compliance need to be re-evaluated for the new usage or new product.
- Source code received from a third party software provider which may or may not have incorporated FOSS in it. FOSS compliance due diligence applies to all software, proprietary and 3rd party.
- Source code downloaded from the web with unknown author and/or license which may or may not have incorporated FOSS in it.
- A new proprietary software component entering the build system where engineering may or may not have borrowed FOSS code and used it in this proprietary software component

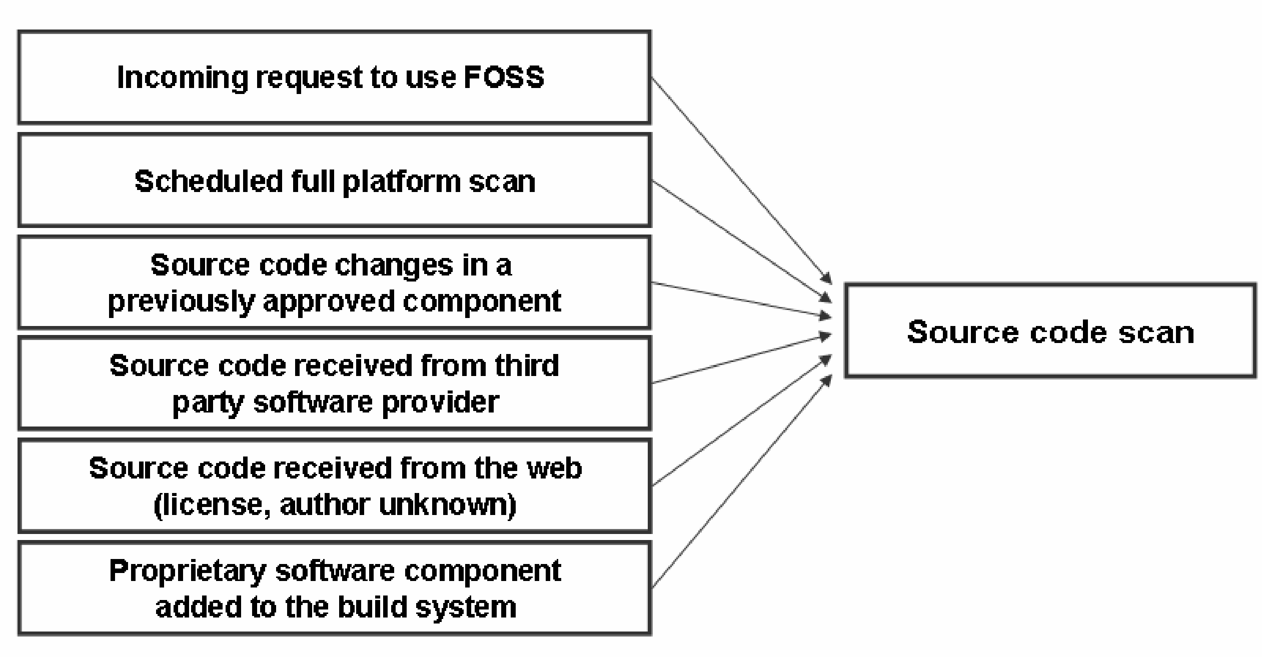
Incoming request to use FOSS
Scheduled full platform scan
Source code changes in a previously approved component
Source code received from third party software provider
Source code received from the web (license, author unknown)
Proprietary software component added to the build system
Source code scan

Figure 2. Elements that can trigger a source code scan

Later in this paper, we present the inputs to this step and the output from it going into the next step ("Identification and Resolution").

Identification and Resolution

In the identification and resolution step, the auditing team inspects and resolves each issue flagged by the source code scanning tool. Scanning tools typically identify potential matches of source code to cataloged open source projects. If code matches more than one open source project, pedigree analysis must be performed during this step to identify the true origin of the code and its corresponding license(s).

License Review

In this third step of the process, the compliance team reviews the report generated by the source code scanning tool, the license information of the software component, and comments in the compliance ticket. If needed, legal counsel will be requested to advise and decide on the incoming and outgoing licenses of the software component in question. Once done, the compliance ticket is forwarded for architectural and linkages review (next step in the process). If a possible licensing issue is discovered, for instance one related to incompatible licenses in the software component's source code, the issue will be flagged and the compliance ticket will be reassigned to Engineering to rework code and eliminate the licensing conflict.

In some cases, if the licensing information is not clear or if it is not available, the compliance team, as instructed by legal counsel, should contact the project maintainer or the FOSS developer to clarify the ambiguities and to receive a confirmation of the license under which that specific software component is licensed.

Architecture Review

In the architecture review step, the compliance team performs an analysis of the interaction between the FOSS, proprietary and third party code. The architecture review consists of examining an architectural diagram that identifies:

- Components that are Open Source (used "as is" or modified)
- Components that are proprietary
- Components that are originating from third party software providers
- Component dependencies and interactions
- Communication protocols, interfaces, and linkage mechanisms
- Components that live in kernel space versus user space,
- Shared header files,
- etc.

If the compliance team discovers any architectural issues, they would forward the compliance ticket to Engineering with direction to resolve the issues. If there are no issues, then the ticket moves into the final step in the process.

Final Review and Decision

The final review is the last step in the process, during which the compliance team approves or rejects usage. In most cases, if a software component reaches the final review, it will be approved unless a new condition has presented itself (such as a change in the way the software component is being used). Once approved, the compliance team prepares the list of license obligations for the approved software component and passes it to appropriate departments for fulfillment.

Inputs and Outputs

In this section, we examine the inputs and outputs of every step in the process. Figure 3 provides a summary of these inputs and outputs.

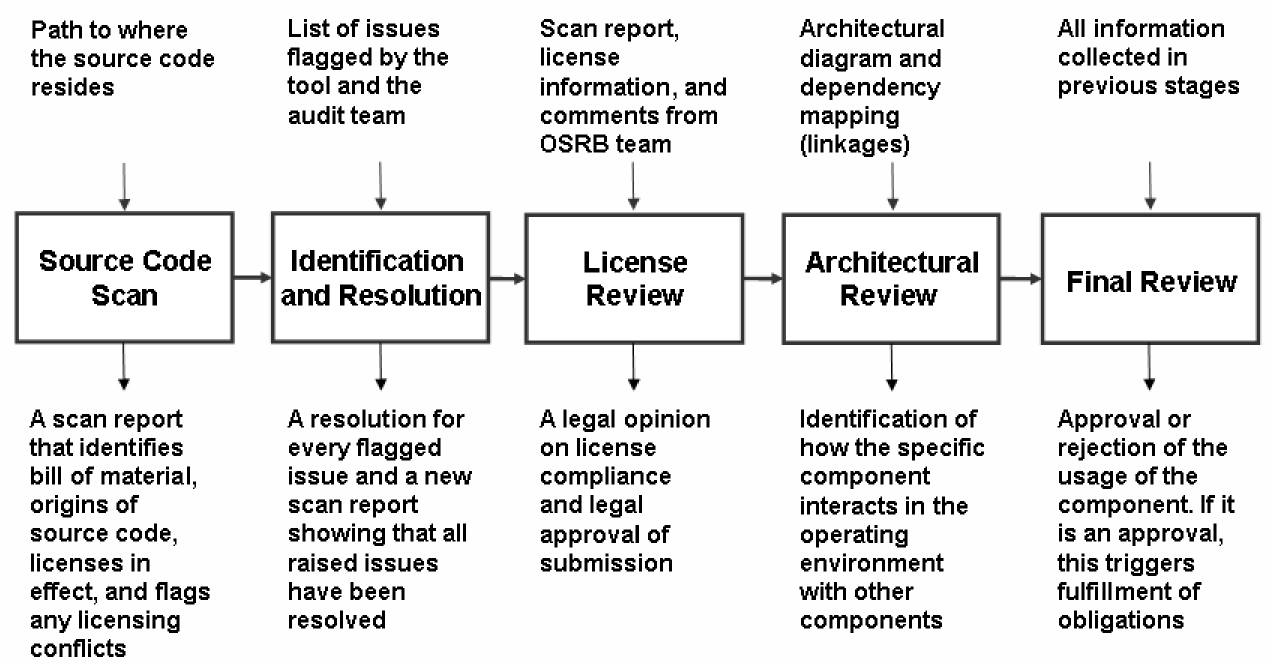
Path to where the source code resides
List of issues flagged by the tool and the audit team
Scan report, license information, and comments from OSRB team
Architectural diagram and dependency mapping (linkages)
All information collected in previous stages
Source Code Scan
Identification and Resolution
License Review
Architectural Review
Final Review
A scan report that identifies bill of material, origins of source code, licenses in effect, and flags any licensing conflicts
A resolution for every flagged issue and a new scan report showing that all raised issues have been resolved
A legal opinion on license compliance and legal approval of submission
Identification of how the specific component interacts in the operating environment with other components
Approval or rejection of the usage of the component. If it is an approval, this triggers fulfillment of obligations

Figure 3. The inputs and outputs of every phase in the example FOSS compliance process

Source Code Scan

Input

- A usage request form that an engineer has filled out and submitted. The form includes all information about the FOSS and other code to be scanned in addition to the location of the source code in the source code repository system.
- A periodic full platform scan that takes place every x weeks to ensure that no software component has been included into the platform without a corresponding compliance ticket.

Output

The output of this phase is a source code scan report that provides information on:

- Bill of materials
- Licenses in effect, license texts and summary of obligations
- License conflicts
- File inventory
- Identified files
- Dependencies
- Code matches
- Files pending identification
- Source code matches pending identification
- etc.

The report is then attached to the compliance ticket and becomes automatically visible to all parties involved with compliance for this component.

Identification and Resolution

Input

Scan report generated by the scanning tool. The report flags licensing issues such as incompatible licenses or licenses whose use is in conflict with declared corporate policy.

- If there are no issues, then the compliance officer will move the compliance ticket one step in the process to reach the licensing review step.
- If there are issues to be resolved, then the compliance officer creates sub-tasks within the compliance ticket and assigns them to Engineering to be resolved. The sub-tasks will include a description of the issue and a potential solution that might be implemented by Engineering.

Output

Leaving this step, all sub-tasks of any given compliance ticket must be closed and all issues discovered in the earlier step are resolved. In some cases, the compliance team may perform a new source code scan and generate a new scan report confirming that earlier issues do not exist anymore.

Licensing Review

Input

- Source code scan report
- A confirmation that all the issues that were identified in the scanning step have been resolved
- Copies of the license information of the software component attached to the compliance ticket (README, COPYING, AUTHORS files available in the source code packages)
- Comments from the compliance team in relation to this compliance ticket (concerns, additional questions, etc.)

Output

- An expert opinion on licensing and compliance of this component
- A decision on the incoming and outgoing license(s) of the software component in question.
  - » The incoming license is the license under which you received the software package. The outgoing license is the license under which you are licensing the software package.
  - » The incoming and outgoing licenses are in the plural form because in some cases a software component can include source code incoming from various sources and licensed under different but also compatible licenses.

Architecture Review

Input

- The architectural diagram which illustrates the interaction between the FOSS and your proprietary and third party code.

- The output of the linkage analysis tool which allows you to find potentially problematic code combinations at the static and dynamic link level.

Output

The result of the architecture review is an analysis of the licensing obligations that may extend from FOSS to the proprietary and third party software components.

Final Review and Decision

Input

A complete compliance record which includes the following:

- Source code scan report
- List of discovered issues and information on how they were resolved and who closed them
- Expert opinion on compliance and decision on incoming and outgoing licensing
- Architectural diagram and information on how this software component interacts with other software components

Output

A decision of either approving or denying the usage of the software component in question.

Note – Post Approval

After the approval, for each of the approved software components, the compliance team will:

- Update the software inventory to reflect that the specific software component version x is approved for usage in a product y, version z, etc.
- Issue a task to the documentation team to update end user notices in the product documentation to reflect that FOSS packages are being used in the product
- Trigger the actions to satisfy the source code distribution obligations when the product ships

FOSS Compliance Tools

A number of open source and commercial tools may play an important role in implementing the FOSS compliance process.

FOSSology

FOSSology is a source code scanning tool which provides a framework for software analysis that allows discovery of licenses, parsing of RPM spec files, determination of file types, and unpacking of input files (such as .tar, .gz and .iso) into their component files.

» Download the tool: http://www.fossology.org/
» For more information: http://www.linuxfoundation.org/sites/main/files/publications/lf_foss_compliance_fossology.pdf

Dependency Checker Tool

The Dependency Checker tool identifies source code combinations at the dynamic and static link levels and provides a license policy framework that enables FOSS compliance officers to define combinations of licenses and linkage methods that are to be flagged if found as a result of running the tool.

- Access source code via git: http://git.linuxfoundation.org/?p=dep-checker.git
- Subscribe to the mailing list: https://lists.linux-foundation.org/mailman/listinfo/dep-checker-dev
- For more information: http://www.linuxfoundation.org/sites/main/files/publications/lf_foss_compliance_dct.pdf

Code Janitor Tool

The Code Janitor provides linguistic review capabilities to make sure developers did not leave comments in the source code about future products, product code names, mention of competitors, etc. The tool maintains a database of keywords that are scanned for in the source code files to ensure source code comments are sanitized and ready for public consumption.

- Access source code via git: http://git.linuxfoundation.org/?p=janitor.git
- Subscribe to the mailing list: https://lists.linux-foundation.org/mailman/listinfo/code-janitor-dev
- For more information: http://www.linuxfoundation.org/sites/main/files/publications/lf_foss_compliance_cjt.pdf

Binary Analysis Tool

The Binary Analysis tool offers a modular framework that assists with auditing the contents of compiled software. It helps to discover what components were used to create compiled code.

- Download the tool: http://www.binaryanalysis.org/en/content/show/download
- For more Information: http://www.binaryanalysis.org/en/content/show/documentation

OSS Discovery Tool

This is a free, open source scanning tool that helps enterprises find the open source software included in their internal applications and installed on corporate workstations and servers.

- Download the tool: http://www.openlogic.com/products/scanners.php#oss-discovery

Commercial Tools

In addition to the above mentioned FOSS compliance tools, there are several vendors of commercial compliance tools. These vendors include: Black Duck Software, OpenLogic, Palamida and Protecode.

Example Study Cases

Figure 1 illustrated the 5-Step FOSS identification and review process without taking into consideration the back-and-forth that may occur between the various steps in the process. Figure

4 presents four different cases of compliance tickets as they go through the process.

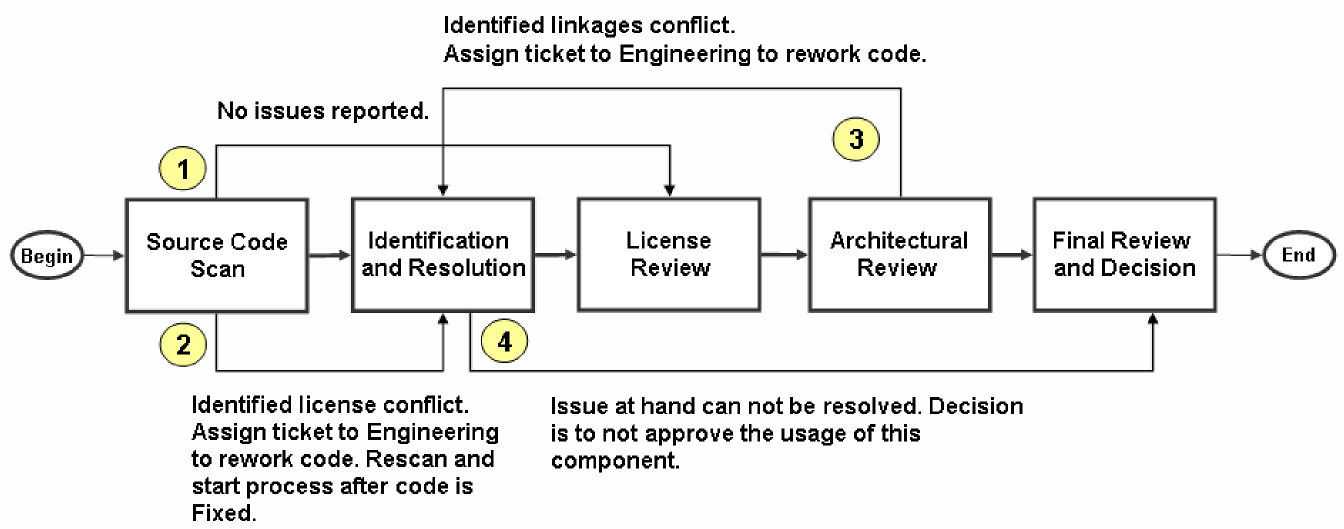
Identified linkages conflict.
Assign ticket to Engineering to rework code.
No issues reported.
1
3
Begin
Source Code Scan
Identification and Resolution
License Review
Architectural Review
Final Review and Decision
End
2
4
Identified license conflict. Assign ticket to Engineering to rework code. Rescan and start process after code is Fixed.
Issue at hand can not be resolved. Decision is to not approve the usage of this component.

Figure 4. Example of specific compliance cases being processed withing the FOSS compliance process

Case 1: No Issues – Clean Bill of Materials

In this scenario, the source code scanning tool reports that there are no licensing issues with respect to the code scanned. The software component may be 100% proprietary and includes only proprietary source code that the tool was not able to match to any FOSS code. In this case, we assume the fast track and the compliance ticket for that specific component will be forwarded for architectural and linkages analysis.

Case 2: Combining Source Code with Incompatible Licenses

The software component scanned consists of source code that originated from multiple sources and the source code has incompatible licenses and as such can not be combined. In this case, the source scan report will be generated and the compliance ticket will be assigned to the engineer who internally owns that software with a request to remove the source code with the conflicting license.

Once the component owner reworks the code, the software component will be scanned again to verify that issue has been resolved and the ticket proceeds for licensing review.

Case 3: Problem with Software Linkages

The compliance ticket has passed licensing review and it is in the architectural review. The compliance officer discovers an issue with the linkages that needs to be fixed prior to approving usage of this software component. The compliance officer moves the compliance ticket back into the resolution phase assigning it to engineering to resolve the linkage issue. Once done, the compliance officer re-verifies the linkages before sending the compliance ticket to the final review step prior to approval.

Case 4: Unable to Resolve

In this scenario, whatever issue that was discovered after scanning the source code is not easy to resolve and engineering may decide to stop using this software component and design a better solution that is free from any compliance issue. For example, license incompatibility issues that can not be resolved may lead to the decision of designing and implementing a new solution.

Conclusion

We provided review of an example FOSS identification and review process that consists of five steps. Throughout the paper, we looked at what happens in these various steps, identified the inputs and outputs of each step and examined four different case scenarios.

How can you help your company be ahead of the curve on FOSS compliance? Check out the resources that the Linux Foundation is making available, ranging from professional training to tools and everything in between. Details are in the resources section.

Thanks for reading and stay tuned for more publications on the topic of FOSS compliance!

Linux Foundation Resources

- **Open Compliance Program:** http://www.linuxfoundation.org/programs/legal/compliance
- **Professional and Comprehensive Compliance Training:** The Linux Foundation offers hands-on training from compliance experts for individuals and companies responsible for achieving compliance with open source licenses and establishing an open source compliance program, as well as for those who simply want to learn more about compliance. Options available include live onsite training in addition to instructor-led live remote training. http://www.linuxfoundation.org/programs/legal/compliance/training-and-education
- **Open Compliance Program Self-Assessment Checklist:** An extensive checklist of practices found in industry-leading compliance programs. Companies can use this checklist as a confidential internal tool to assess their progress in implementing a rigorous compliance process and help them prioritize their process improvement efforts. http://www.linuxfoundation.org/programs/legal/compliance/self-assessment-checklist
- **Compliance Publications:** http://www.linuxfoundation.org/publications
- **Open Compliance Directory and Rapid Alert System:** http://www.linuxfoundation.org/programs/legal/compliance/directory
- **Compliance Tools:** http://www.linuxfoundation.org/programs/legal/compliance/tools
- **The Software Package Data Exchange™:** The SPDX™ specification is a standard format for communicating the components, licenses and copyrights associated with a software package. http://www.spdx.org/
- **FOSSBazaar:** An open community of technology and industry leaders who are collaborating to accelerate adoption of free and open source software in the enterprise. http://fossbazaar.org/

Acknowledgments

The author would like to express his gratitude to Philip Koltun, Ph.D., (Director of the Open Compliance Program at The Linux Foundation) for his review and valuable input.

About the Author

Dr. Ibrahim Haddad manages The Linux Foundation's Mobile Linux initiative and works with the community to facilitate a vendor-neutral environment for advancing the Linux platform for next generation mobile computing devices.

About the Open Compliance Program

The Linux Foundation's Open Compliance Program is the industry's only neutral, comprehensive software compliance initiative. By marshaling the resources of its members and leaders in the compliance community, the Linux Foundation brings together the individuals, companies and legal entities needed to expand the use of open source software while decreasing legal costs and FUD. The Open Compliance Program offers comprehensive training and informational materials, open source tools, an online community (FOSSBazaar), a best practices checklist, a rapid alert directory of company's compliance officers and a standard to help companies uniformly tag and report software used in their products. The Open Compliance Program is led by experts in the compliance industry and backed by such organizations as the Adobe, AMD, ARM Limited, Cisco Systems, Google, HP, IBM, Intel, Motorola, NEC, Novell, Samsung, Software Freedom Law Center, Sony Electronics and many more. More information can be found at http://www.linuxfoundation.org/programs/legal/compliance.

The Linux Foundation promotes, protects, and advances Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.

To learn more about The Linux Foundation, the Open Compliance Program or our other initiatives, please visit us at http://www.linuxfoundation.org/.

THE
LINUX
FOUNDATION