Q&A

Hassib Khanafer

Q. Does a software development firm need an open source policy?

A. Software development has evolved from the guarded approach of building commercial products entirely in-house to building them by piecing together proprietary, third-party, open source, and contractor code. The wide availability of open source software (OSS) spares developers from having to reinvent the wheel, while accelerating development and reducing costs. Indeed, the wide availability, ease of access, and lack of financial cost of OSS lead many developers to believe that it is a risk-free solution to many of their pressing development problems. However, similar to any third-party commercial software, developers need to respect the licensing and copyright terms that govern how the OSS code can be used, address the security vulnerabilities that may be associated with the software, and abide by export control regulations if the used software contains implementations of encryption algorithms.

Licensing can be a particularly complex issue for organizations wishing to leverage OSS as part of their software products. Although there are many different types of OSS licenses, they generally fall under the categories of copyleft or permissive licenses. Copyleft licenses, such as the GNU General Public License (GPL), generally require that products containing GPL-licensed code be released under the same license. In contrast, permissive licenses, such as the MIT License, grant the user more flexibility in terms of how the software can be used. For example, the MIT License allows users to do whatever they want with the software as long as a copy of the license accompanies the copied software. The onus is on the user of an OSS component to make sure that they are abiding by the obligations of the license.

Similar to its proprietary counterparts, OSS is not immune to security vulnerabilities. Developers need to make sure that the specific versions of the OSS component they are using are not associated with known security impairments that could expose their clients. Developers are required to take the appropriate actions, for example, to upgrade to new versions that are free from the vulnerabilities or replace the vulnerable component with another open source component. Lastly, software vendors who plan to export their software should be aware that many jurisdictions, including the United States, the United Kingdom, and Canada, among many others, place stringent regulations on the export of software that contains encryption algorithms or cryptography. These restrictions apply regardless of whether the encryption algorithms form part of an open source module integrated into a software product or are part of the proprietary code.

In all but the smallest of code portfolios, managing the aforementioned risks can be daunting. These challenges may discourage organizations from leveraging OSS in their products. Thus, to make sure that license and copyright obligations are addressed, minimal interruption to the product development cycle is incurred, and the opportunity to use available high-quality OSS is exploited, software development firms should implement an internal open source policy.

An open source policy clearly defines the objectives of using OSS in the enterprise, and it describes how those objectives tie into the overall business strategy. As an example, using OSS components may allow the company to focus its software efforts solely on areas of technology that truly differentiate the company's offering. Or, deployment of OSS may expedite the development of a product, which may tie in to an overreaching enterprise strategy of reducing time to market. The policy also defines the rules that govern the internal and external use of open source software. As an example, while the policy could be lenient in terms of open source licenses used internally for building and testing the product, it could be very stringent in terms of limiting what components can be shipped as part of the product (e.g., the policy could state that no GPL-licensed software should be part of the distributed product).

Furthermore, the policy should clearly define the team that is responsible for its development, evolution, and implementation. Representatives from both the business team (e.g., product line managers) and the devel-

Q&A. Does a software development firm need an open source policy?

Hassib Khanafer

opment team (e.g., architects) could be responsible for the development and evolution of the policy, while the development team could be responsible for its execution.

Other aspects defined by the policy should include:

- the sources from which OSS components may be obtained (e.g., main project websites versus forked sites)
- the forms in which the components may be downloaded (i.e., source files or binaries)
- the ongoing maintenance of the OSS components used (e.g., the processes for applying regular updates of the components and emergency patches such as fixes for security vulnerabilities)
- the steps that should be taken if a policy violation is detected

In support of its open source policy, a company can employ open source compliance tools, which can be integrated into any or all stages of the development cycle. These tools are similar to static code analysis tools that developers employ as part of their quality assurance testing. Whereas the latter are used to check software for potential coding issues, the former are used to check the code for presence of open source components and report on associated licenses and copyrights, known security vulnerabilities, and encryption content. Both static code analysis tools and open source management tools have similar usage patterns, although the user community for the latter is larger and could include legal and licensing teams. Some development organizations deploy these tools at the end of their development cycle, while others prefer to integrate the tools throughout the development cycle, which decreases remedial efforts that may be needed prior to a product release.

Open source management tools can be integrated into the development environment, where they can continuously monitor the use of OSS in real time and help developers with the early detection and remediation of potential policy violations as they arise. Additionally, these tools can be integrated with the build infrastructure of products; as an example, the nightly build could trigger the tool to check the code base for any newly used OSS and their licenses, security vulnerabilities, and other attributes. This approach is similar to how companies use unit test frameworks (e.g., Junit), where the execution of test suites is triggered by the software build process.

Hence, the growing adoption of OSS components in the production of software products mandates that the users of such software establish and implement internal open source policies that govern and manage the use of such software. The introduction and implementation of such policies is best supported by the use of open source management tools that automate the analysis of software code portfolios and aid the removal of any uncertainty around adopting open source software.

About the Author

Hassib Khanafer is the Chief Technology Officer at Protecode, a provider of open source license and security management solutions that can be used throughout the software development lifecycle to ensure license compliance. Hassib is a technology enthusiast who has been in the software industry for more than 25 years. His experience spans the domains of network management, OSS license management, financial applications, human resource applications, enterprise collaboration tools, oil and gas maintenance planning applications, e-commerce systems, and software management tools. Prior to joining Protecode, he worked in different positions in Nortel Networks, Siemens, Avaya Inc., and Kuwait Gulf Oil Company. Hassib holds a Bachelor's degree in Electrical Engineering from the University of North Carolina at Charlotte, United States, and a Master's degree in Computer Engineering (Software Systems) from Kuwait University.

Citation: Khanafer, H. 2015. Q&A. Does a Software Development Firm Need an Open Source Policy? *Technology Innovation Management Review*, 5(5): 45–46. http://timreview.ca/article/897

Keywords: open source management, open source license compliance, security vulnerabilities