

Anatomy of a Software Code Audit Process

Software has become a major component of products that are produced by most technology companies and is rarely written from scratch.

What you will learn...

- A software code audit establishes code ownership.
- The earlier an audit is performed the easier, and cheaper it is to fix any problems discovered during the audit.
- Automated code scanning is the most efficient way to conduct an audit.
- Audits produce detailed reports on what is in a software product and what obligations must be met.

What you should know...

- Resourceful developers use third party and open source code to speed up development time and reduce development costs.
- Most open source licenses have certain legal obligations that have to be met.
- Uncertainty around code ownership can stall product launches and negatively impact M&As.
- A common mistake is to start a code audit process in the last step of a transaction.

Resourceful software development organizations and developers use a combination of previously created code, commercial software and open source software, and their own creative content to produce the desired software product or functionality. Anytime a product containing software changes hands there is a need to understand its composition, its pedigree, its ownership, and any third-party (including open source software) licenses or obligations that govern its use by its new owners.

Avoiding Uncertainties in a Technology Transaction

Technology transactions that involve software may include the launch of a product into the market, *mergers & acquisitions* of companies with software development operations, or technology transfer between organizations whether they are commercial, academic or otherwise public. Any uncertainty around either ownership of software or compliance with the licenses associated with software can:

- deter downstream users,
- reduce ability to create partnerships,
- create litigation risk to the company and the downstream users,
- increase risk and threaten closures in funding deals,
- negatively impact M&A activities,
- increase product time to market, and
- affect company valuation.

So how can all of this be avoided?

A software code audit is a good way to determine what is in your *software product*. A software code audit should not be confused with the more common place software audit process; the latter generally has to do with making sure you have paid for the software applications (e.g. Microsoft Office) you are using in your organization. Software code audits identify building blocks (files or software modules or packages, or even five lines of external code) that are used in a product or exist in the code inventory of an organization.

The audit process establishes code ownership, licensing or copyright obligations around any third party content in the code portfolio, authorship, package versions and export restrictions. Software code audits can also highlight alignment with the policies around either use or delivery of software in a particular organization. Software code audits can also pinpoint code-reuse between different portfolios within or across organizations.

The common mistake is to only start a code audit process in the last step of a transaction. Starting the audit *in anticipation* of a transaction allows for timely correction of any shortcomings detected during the audit. You certainly do not want to delay a transaction because of uncertainties uncovered during the audit.

What Is Software Code Audit Process?

Except in simplest, smallest code portfolios, a manual audit of a company's code portfolio takes time, is

inaccurate, and expensive. Automated code scanning solutions can sift through large portfolios quickly and efficiently, detecting outside code and retrieving licensing and other attributes of external components. While automated code scanning solutions operate very fast, there is still a human element to a thorough audit project. Our experience has shown that most of the time an audit is taken by the front-end and back-end processes.

The software code audit process usually involves

- Establishing a legal framework (NDA) between the parties involved and the auditor.
- Question and answer between the parties involved to establish:
 - the objectives of the audit, to understand the company or product that is audited,
 - the specific business of the target companies.
 - their third party software practices,
 - the software environment that is used in the target company, and
 - their open source adoption policy (if any)

In some cases, all code that must be audited is not in one place, or must be *assembled* before an audit is carried out. Depending on the size of the project, the front-end process can take 1-5 days.

Software Code Scanning and Detection

Once the legal framework is in place, the code is available, and the environment discovery process is complete, an automated scanning application is set up. The complete job is broken into logically-meaningful segments (for example, identifiable subprojects and modules), and then the actual automated code scanning is carried out. Ownership warnings generated by the automated application (such as proprietary code without appropriate headers or copyright information, or conflicting license information) are brought to the attention of the staff, and either resolved or marked for further action.

The reports created by the automated solution are reviewed by an expert audit staff, and a final executive report is assembled. Depending on the size of the audit project, this step can take as little as a couple of days (small project containing thousands of files) and up to two weeks (for a very large portfolio of hundreds of thousands of software files).

End-Results of a Software Code Audit

The end result of a software code audit is a combination of two reports.

The first is a high-level executive overview report that is custom created by the audit staff. This report defines the software code audit environment, the process used,

and the major findings, in simple graphical and tabular format. Attention is drawn to specific packages, files or licenses. Information on commercial or open source software components, a description what each piece of software does, who created it, and related references on public-domain project websites should be provided.

Important information such as copyright owners, licenses associated with the discovered software packages, and optionally encryption or export obligations, are tabulated. The text of all licenses that are discovered is included with this report. The report lists all external content, including complete third party software files, modules or projects, or snippets of code that have a code structure similar to known open source projects. The findings of a software code audit must be verifiable; therefore references or hyperlinks to all information that is discovered would be provided. The second report is a detailed machine-generated report, listing:

- all packages, files, licenses, copyrights, etc. associated with all software files in the target portfolio, and
- optionally, a license obligation report, summarizing the obligations associated with all licenses found in the portfolio.

The detailed report can be very large, and is normally provided as a back up to the high level executive report. This report is normally consulted if a specific project or file requires further scrutiny.

How Much Does a Software Code Audit Cost?

Generally the cost of an audit is proportional to the complexity of the project, which in turn can be roughly defined as the number of files in the target portfolio, the nature of the packages (commercial or public domain) used in the portfolio, and the information that is available about those packages. Most audits (thousands and up to hundreds of thousands of software files) fall within a \$5-\$40K range.

If you're planning a specific transaction involving software assets, whether it's an M&A, equity investment, product introduction, demand for IP indemnity, commercialization of research or other event, conduct a software code audit as early as possible in the transaction. Knowing what's in the code can speed up transaction times and reduce costs associated with fixing problems at the last minute.

KAMAL HASSIN

Kamal Hassin, Director, R&D and Product Management at Protecode, is a thought-leader in the area of open source licensing and is the author or co-author of a number of papers on Software Intellectual Property management. Kamal has a Bachelor of Engineering degree and a Masters degree in Technology Innovation Management from Carleton University. He can be reached at khassin@protecode.com.