

Creating and Implementing An Open Source Policy:

Managing Use and Reducing Risk

Bernard Golden

Chief Executive Officer
Navica

NAVICA

Executive Summary

Many organizations are now using open source software because of its clear benefits:

- > Low cost
- > Flexibility of use
- > Lack of vendor lock-in
- > Ability to modify the product to better suit business needs

However, open source software licenses impose obligations that must be observed in order to obtain those benefits. For example, the Apache Software License requires that any documentation accompanying a product containing Apache-licensed software include an attribution reflecting the presence of Apache-licensed code in the product. Particularly worrisome to many organizations, particularly software vendors, are the obligations imposed by the so-called copyleft licenses like the GNU General Public License, which can impose source code distribution obligations upon companies

Because open source software is available for free download, it may be present in a code base with little awareness by most employees of a company. In order to ensure that the company is complying with open source obligations and also to protect the company from potential risk, organizations should create an Open Source Policy outlining the ways open source software may be selected, approved, managed, and distributed.

This whitepaper presents a five phase process, based upon the Navica Open Source Policy Process, which provides companies a way to implement their own Open Source Policy. The Open Source Policy Process contains these five phases:

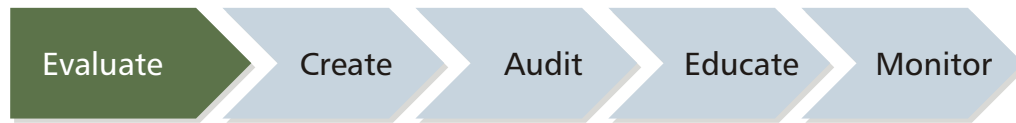
- » **Evaluate:** This phase uses interviews to understand the company's open source use profiles; in other words, how it currently uses open source software in its infrastructure and products. This allows the range of the company's obligations to be defined so that the resulting process may ensure compliance.
- » **Create:** This phase defines and creates the company's Open Source Policy: the way open source use is approved and the process for managing its use from approval to release.
- » **Audit:** This phase examines the company's existing source base to identify what open source products are present and the license compliance obligations for those products. If there are compliance shortcomings, the Audit phase also includes creating and implementing a mitigation plan to bring the company into compliance.
- » **Educate:** This phase ensures that all company employees involved with open source are trained on the company's Open Source Policy so that its requirements and processes are followed going forward.
- » **Monitor:** This phase reflects a steady-state of open source use that observes the company's Open Source Policy. By ensuring that the Policy process is followed, ongoing monitoring ensures that no compliance issues occur.

This is a graphical representation of the Open Source Policy Process:



By implementing an Open Source Policy Process, companies may be certain that they achieve the greatest benefit from open source software while reducing their risk exposure significantly.

Phase 1: Evaluate Open Source Use Profiles



Phase Goals	Phase Tasks	Phase Outcomes
Establish Open Source Use Profile	Interview Employees Involved with Open Source Identify Open Source Use Profile and Risk Exposure	Open Source Use Profiles Open source Risk Factors Identified Open Source Compliance Requirements Defined Company-Wide Understanding of Open Source Use and Obligations

Phase Goals

The goal of the evaluation phase is to establish the company’s open source use profiles -- a baseline of how the organization uses open source software and of the license implications for its business operations and objectives. It is critical to understand how open source is used by the organization because the business and engineering goals affect the impact that open source will have upon the organization.

For example, if an organization incorporates open source components into a product that it then distributes, it is going to be very concerned with the distribution implications of the license or licenses of the open source components.

By contrast, if an organization incorporates open source components into its internal IT infrastructure, with no plans to distribute any software beyond the four walls of its data center, it is going to be much less concerned with the distribution implications of the license or licenses of the components. However, it may still be very concerned with understanding how open source is used within the organization

because it may wish to prevent open source product proliferation – that is, it may have an approved list of open source components to be used within the organization’s infrastructure and wish to prevent non-approved open source components from being implemented.

With the increasing interest by companies to leverage open source development models (e.g., the desire to allow distributed developers employed by different company divisions throughout the world) to collaborate on joint development, a third set of open source concerns comes into play: the desire to enable internal development groups to leverage collaborative development practices without triggering license conditions that would require general release of source code outside the company.

As these three examples indicate, it’s important to understand how the company makes use of open source so that these use case profiles can serve as an input to the creation of the company’s open source policy.

Phase Tasks

Interview employees to establish open source use profiles

The best way to understand the way a company uses open source is to interview knowledgeable persons. By getting each person to describe how his or her group makes use of open source, an overall use context can be established. Therefore, the first task in this phase is to perform interviews with a representative sample of employees of the company.

Typical questions that should be asked during the open source use profile interviews include:

- » [Does the organization modify open source code or use it as-is?](#) If the organization has no intention of distributing the source code received with open source products, its exposure to the redistribution and attribution requirements of open source licenses is nil. However, if an organization creates and distributes derivative works of open source products it can be subject to a number of license conditions. Specifically, software companies that incorporate GPL-licensed code into their products (which may thereby become derivative works) may find themselves inadvertently converting a proprietary product to an open source product and thereby affecting business model assumptions of the product.
- » [Does the organization distribute open source products?](#) Even an organization that makes no changes to open source products may, nevertheless, distribute them. For example, a software company may distribute a CD of its products; that CD may also include GPL-licensed open source utilities. Even organizations that do not think of themselves as software companies may, in fact, be software distributors. An insurance company may send CDs of software to independent agents and thereby become a distributor.

The question of distribution is important because, depending upon the license of the open source product, the source code developed by the organization might also need to be distributed. While it is clear that most software companies would prefer not to have their products become subject to being open source, even non-software companies may prefer to keep their source code private as it may represent proprietary business processes or trade secrets.

- » [Does the organization license software from the outside, either via commercial arrangements or outsourced contract development?](#) Both of these circumstances represent another avenue that software comes into the organization, and can expose the obtaining organization to open source compliance issues if the software contains open source components.

- » [Does the organization contribute to open source products?](#) Some companies contribute to open source products as part of their business strategy. An example of this would be hardware companies that contribute drivers to the Linux operating system; by participating in Linux development, their hardware business is aided. Other companies, particularly large technology companies, employ people who contribute to open source products in order to foster the general growth of open source. Google is known for employing individuals whose open source efforts do not directly help Google's business; however, their employment presumably provides brand enhancement or indirect competitive advantage to the company.
- » [Do employees contribute to open source projects in their spare time?](#) In contrast to the last bullet point, this question refers not to company-assigned open source activity, but, rather, to activity undertaken for personal interest, fulfillment, or whim. In the abstract, there is nothing wrong with this type of activity, but it can present issues in regards to ownership of the employee-created IP (most companies assert ownership of employee inventions even if worked on in off-hours) as well as potential for infringement of company IP or trade secrets if the employee works on an open source project similar to the work he or she does during working hours.

The goal of these interviews is to form a perspective on how open source is being used throughout the company today as well as potentially used in the future.

It is not unusual for members of different company divisions to have a different – even a very different – perspective on how open source is used within the company. After all, different divisions may address markets by selling software products, offering software-as-a-service (SaaS) over the Internet, selling hardware products that contain embedded software or software stored on a device storage medium like a hard drive, or even giving away software products with an open source license as a strategy to drive trial usage in order to build demand for a commercially-licensed product. Naturally, each division would have its own perspective on how open source is being used in “the company.”

More surprising, however, is the fact that members of the same division can have very different perspectives on how open source is used within the organization. They may have different opinions on how open source use is currently approved, or even about which open source products are actually being used at the time of the interview.

This variance of opinion illustrates why it's important to explicitly interview a number of individuals so that a broad understanding of the range of open source use may be formed. Only by capturing the complete spectrum of open source use can a policy be created that addresses all use cases rather than those provided by a too-small sample size.

With regard to the types of individuals that should be interviewed, diversity of function is the key. Relying on an attorney's perspective of how open source use is approved is likely to provide an incomplete picture. More important from a de facto use perspective is to interview members of the engineering team who actually download and implement open source components. Be aware, however, that individuals may inadvertently or deliberately misstate the work practices regarding open source software in an attempt to avoid judgment or the prospect of further work to mitigate improper work practices.

This last aspect – that of individuals inadvertently or deliberate misstating current open source use and processes – illustrates the importance of interviewing a number of individuals so as to gain a broad understanding of how open source is actually being used within the organization.

To capture all open source use requirements, interviews should be held with individuals with the following roles within the organization:

- > Attorneys responsible for licensing and IP;
- > Engineers responsible for product design and implementation;
- > Marketing Managers responsible for collateral, product packaging, pricing, and company protection from IP tainting
- > Operations personnel responsible for building software products and deploying software packages onto company-located servers

Questions or uncertainty about the proper way to manage open source may also arise during interviews; individuals may not understand what they should do to comply with license requirements. Any questions should be noted so that they may be addressed during the Education phase of the Policy Process.

Identify Open Source Use Profile and Risk Exposure

The second task is to develop a profile of open source use that reflects the totality of individual use cases; in other words to develop a profile that is a superset of all the specific scenarios described during the interviews. This superset represents the overall use of open source throughout the organization.

Based on this open source use profile, the organization can define its open source license compliance requirements. For example, if an engineer described the use of a GPL-licensed component within an appliance shipped to end users, that scenario would carry implications regarding the need to ship the component's source code with the appliance as well as the need to segregate the GPL-licensed code from any company proprietary code shipped with the appliance.

Non-technology companies may also need to evaluate their open source use profile and risk exposure. Governmental regulations that require compliance with financial or health standards like SarbOx or HIPAA may mean that a company needs to track its use of open source. The existence of these type of regulations means that companies that do not think of themselves as software companies may nonetheless need to carefully track their approval and use of open source software.

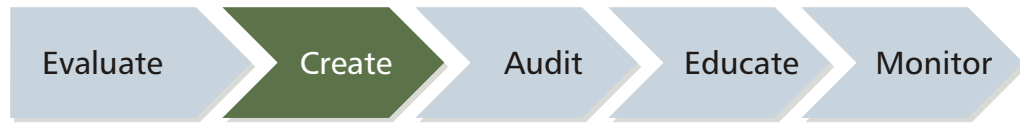
To the extent that any practices were identified during the interviews that are not consistent with open source license requirements, these practices should be identified as presenting risk exposure to the organization and documented as needing to be addressed as part of the Audit phase of the Policy Process. The use scenario should also be documented so that it may be presented as part of the training that goes on during the Education phase of the Policy Process.

Evaluation Phase Outcomes

The outcome of the Evaluation phase of the Policy Process is the following:

- > An organization-wide understanding of the company's open source use profiles; that is, how open source is currently being used throughout the organization.
- > An understanding of the likely compliance requirements of the licenses carried by the open source components whose use has been identified through the interview process.
- > A description of the current risk exposure presented by any identified compliance shortcomings exposed by mapping compliance requirements against current practices.
- > A complete understanding of the open source use case profiles that may be used as an input to the next phase of the Policy Process: Create, where the organization's Open Source Use Policy is defined.

Phase 2: Create Open Source Policy



Phase Goals	Phase Tasks	Phase Outcomes
Create Open Source Policy	Write Open Source Policy Promulgate Open Source Policy Document Implement Open Source Policy Process	Company-wide Open Source Risk Management Common Process for All Open Source Activities

Phase Goals

The goal of the Create Phase is to develop an open source policy that serves as the basis for how open source software is used throughout the organization. This policy will underpin organizational processes relating to open source selection, approval, use, and deployment and is designed to ensure that the organization observes all open source license requirements, thereby reducing any risk factors associated with open source to a minimum.

With an open source policy in place, the organization has a framework within which individual instances of open source use can be judged. In addition, an Open Source Review Board, a typical recommendation of the policy creation process, provides an institutional resource for employees to call upon with questions regarding licenses.

Phase Activity

Coming out of the interviews that take place in the Evaluation phase, one should have a good general comprehension of the ways the organization uses open source software. Based upon this, this phase begins by defining the policy and process for company use of open source going forward.

Since a number of different organizations were interviewed during the Evaluation phase, one should expect that the use requirements will be broader than any one group’s specific requirements. However, since the Open Source Policy must be used across the enterprise, this should be expected.

Writing the Open Source Policy

To take open source use from an ad hoc activity to a formalized, methodical organizational process requires that a policy be created and documented. Failing to concretely document the policy implies that the organization will manage its use of open source inconsistently and raise its risk exposure. Consequently, writing an Open Source Policy is vital to ensure the best possible outcome from open source.

The Open Source Policy embodies three things:

1. Compliance requirements that the organization must observe: Open source licenses, in addition to the rights they offer to the software user, also impose obligations. These obligations vary according to the license and, crucially, according to the type of use made of the open source software. Rather than depending upon individual employees to read, comprehend, and interpret individual licenses, the Open Source Policy should identify the obligations of the organization for each license. For example, the Policy should identify the requirement for copyleft-licensed software to be segregated from company proprietary software. With these obligations documented, it is much easier to ensure that the organization is complying with the obligations.

2. The process to obtain approval for open source use or activity: Because it is important to understand and comply with open source license requirements, the Open Source Policy ordinarily implements an approval process. This process outlines how an employee may request permission to: (1) use an open source component or (2) participate in an open source project outside of work. A request form is usually part of the newly-created policy; many organizations host such request forms online so that the approval process may be automated. The request form usually includes a section for approval/denial as well as for comments. The process typically originates with an individual engineer and is reviewed and approved by his/her direct manager, perhaps by a senior engineering manager, and is then reviewed and given final approval by an Open Source Review Board.

Having a defined approval process makes using open source much more efficient and ensures that no important compliance requirements are overlooked or ignored. It also reduces the knowledge burden upon individual engineers and engineering management, thereby simplifying their work.

3. Organization resources for open source: An Open Source Review Board (OSRB) is almost always part of an Open Source Policy. The Review Board serves as a center of expertise for the organization as well as the ultimate decision-maker regarding open source use. The Review Board typically is made up of representatives from these organizations:

Legal: The Legal Department has the ultimate responsibility that the company's products comply with legal requirements, including license and contract compliance. It is vital for the OSRB to have Legal presentation to ensure that all legal implications of open source decisions are addressed.

Engineering/Development: Engineering/Development is the organization that actually uses open source or creates products that incorporate open source. Decisions about what open source components should be used typically commence with Engineering, so their representation on the OSRB is important.

Operations: This group is responsible for the company's production computing environment. The makeup of the approved software stack is important to this group: if the production environment is used internally, ensuring that its software components are approved and at the proper version is critical; if the production environment is used to provide computer services external to the company, ensuring that all software components are tracked and license conditions complied with is critical.

Marketing: Marketing is responsible for ensuring that all necessary attributions are complied with as well as business planning for products that incorporate open source components. Participation in the OSRB ensures that this group can comply with all requirements as well as help assess the business implications of open source decisions.

Finance and/or Compliance: Some industries are subject to regulatory conditions or requirements relating to certifying their software stacks or data tracking compliance. Participation in the OSRB by groups responsible for certifying compliance may be appropriate, depending upon the specific conditions of the company.

The OSRB receives the open source use requests discussed previously and provides feedback, along with a decision regarding the request. The decision is not usually a denial; more commonly, if the request cannot be immediately approved, the OSRB offers recommendations that would enable the request to be approved after some changes are made to the mode of use. For example, it is often important to segregate GPL-licensed products from products that carry a proprietary license in order to ensure that no possible derivation can be imputed; the originator of the request may not have thought about this issue and the implications for how the ultimate product will be shipped – the OSRB’s feedback means that product code segregation must be implemented in order for approval to be obtained.

The OSRB also acts as a resource for general open source questions. If someone in the company is seeking to understand a particular open source issue better, he or she may contact the OSRB for information on the issue or guidance about where more information is available. To this end, an OSRB often has knowledgeable individual contributors attached to it to act as resources that may be called upon by other parts of the company. By using these resources, requesters may raise the probability that their ultimate open source use request will be approved quickly.

Reviewing and Promulgating the Open Source Policy

Every group that will be affected by the Open Source Policy should review the policy document in draft form so that they may provide comments and understand what activities it will require in terms of day-to-day work.

If internal legal counsel is not very familiar with open source licensing issues, using an outside counsel more familiar with open source may be appropriate during the Policy review task. This helps ensure that the Policy – particularly the obligation activities outlined to confirm license compliance – are appropriate and complete.

Providing all affected groups an opportunity to review and comment on the draft Policy also increases the likelihood that they will observe the policy once it is in effect.

Implementing the Open Source Policy

Two activities that are part of the Open Source Policy have already been outlined: a process to request permission to use an open source component and a process to request permission to participate in an outside open source project. Another important activity is to integrate the Open Source Policy into the existing project management methodology used by the organization.

Most organizations have an established methodology for managing projects – initial proposal, project review at key milestones, task accomplishment by phase, and so on. Use of open source and verification of adherence to the Policy should become one more aspect of the overall project management process. Since projects must adhere to the established policy, linking open source use to it raises the probability of compliance with the policy and also ensures that compliance occurs on an ongoing basis rather than as a last-minute exercise. This is particularly important since late compliance with open source requirements is likely to result in project delay; ongoing verification of compliance makes it less likely that late validation will impose extra costs on the project.

At this point, the Open Source Policy “goes live.” That is, it now becomes the responsibility of every group to observe the Policy and follow its requirements. There will probably be a transitional period while projects that are already underway determine the status of their open source use and come into compliance with license requirements through mitigation efforts. This is understandable and allowance made for compliance activities. For further information regarding initial compliance and mitigation, please see the Audit Phase part of this whitepaper.

Implementation Considerations

Many organizations implement their Open Source Policy in phases. Rather than attempting to simultaneously begin using the Policy throughout the entire company, they will begin with one division, typically one that has a significant base of open source use. Once the process is established and is consistently being used, it is rolled out to other groups. A methodical rollout plan is created to ensure that all departments are eventually using a consistent process.

Decentralized companies face a unique challenge: how to allow local autonomy while ensuring consistent practices. In environments like these, a common practice is to define the policy centrally, but allow each autonomous organization to manage its own open source use, with a local OSRB. This enables local control and quick decision-making. A central OSRB exists as a referral resource for unusual open source use cases. Typically, the decision to refer a particular open source use case to the central OSRB is initiated by the attorney attached to the local OSRB.

The goal of both gradual rollouts and decentralized policy decisions remains the same: a single Open Source Policy applied consistently throughout the organization.

Create Phase Outcomes

The Create phase is crucial for a company’s future success with open source. During this phase the company’s business strategy and its open source rights and obligations come together to form its Open Source Policy.

Based on the Policy, the organization can move forward to ensure it observes open source license requirements and that all of its employees understand and follow the correct procedures for using open source. In addition, it may create an Open Source Review Board to serve as a company expertise resource as well as an open source approval mechanism.

The Open Source Policy should be documented so that there is a written resource that defines how open source will be managed throughout the company. Once the Policy is available, it should govern the company’s open source use.

Phase 3: Audit the Company Code Base



Phase Goals	Phase Tasks	Phase Outcomes
Understand Actual Open Source Use Within Company	Identify Open Source Presence via Interviews or Automated System	Complete List of All Open Source Components within Company Software Assets
	Mitigate Open Source Compliance Issues	All Open Source Compliance Issues Addressed

The Audit Phase establishes what open source products are currently present in the company's source base. Obviously, the presence of open source is a vital concern for software vendors, since open source licensing conditions – particularly the so-called copyleft licenses – can affect the business and revenue potential of the vendor.

Organizations that do not consider themselves software vendors may still need to establish what open source products are actually present in their source base or infrastructure. As previously noted, many organizations that do not consider themselves vendors may, nevertheless, still be distributing software and thereby be affected by open source licenses. Furthermore, even organizations that use software solely within the confines of their data center may still need to establish what open source products they are running. While these organizations may not be subject to redistribution conditions, it still is important to understand their actual open source use for other reasons. For example, many organizations may be running different open source products that offer similar functionality. Supporting multiple products inevitably is more expensive than standardizing on a single product due to skill and staffing requirements.

There are two main tasks in the Audit Phase:

- Identify what open source products are present in the source base.
- Mitigate any non-compliance with open source licenses.

Identifying Open Source Presence

The interviews in the Evaluation Phase were primarily concerned with establishing open source use profiles; in other words, understanding how the company uses open source and the license implications of that use.

As part of those interviews, many open source products that are currently being used by the company probably will be identified. The question then is, why perform an audit? It's bound to take time, employee and management attention, and perhaps impose financial costs as well. So, why perform an audit?

The reason is because human memory is imperfect. People may feel that they know the complete list of open source components present in a given code base, but they may, in fact, have forgotten about a component hurriedly integrated to temporarily solve a particular problem and then accidentally left in place.

In addition, open source components may have been integrated by employees no longer working for the company or contractors employed for a short period and long since let go. Consequently, there perhaps will be open source components incorporated into the company's software base which no current employee is aware of.

A third reason for auditing a company's source base is that it may have obtained code from outside sources, either third party providers or external subcontractors. Even though those outside sources may have provided assurance that the code they delivered contains no open source, it is common for externally sourced code to contain open source code, either inadvertently or deliberately.

Consequently, it is necessary to perform a thorough audit of an organization's own code base to identify what open source code is actually contained within it.

Methods to identify Open Source Code

There are essentially three methods organizations may use to identify the presence of open source code in their source base.

1. Interview employees: The discussion directly above has noted the shortcomings of this approach. While interviews are quick and relatively easy to perform, they are unlikely to identify some or even most of the open source products being used. However, despite the lack of thoroughness, interviews are vital because they can establish the context in which an open source product was selected.

2. Audit the source base via the use of a home-grown tool: The company may have its employees create a tool to audit its source base. These tools will go through a source base and identify open source components. Techniques often used for home-grown tools include searching for character strings with open source license language (e.g., a string that contains the term “GPL”). This approach has the advantage of efficiency through using an automated approach as well as being relatively inexpensive because employee time is used to create the tool.

On the other hand, the range of techniques available to the average organization does not address all the potential techniques that actually exist; perhaps worse is the fact that obfuscation techniques may have been used to disguise the insertion of open source code into a source base (i.e., techniques like removing license language may have been used to prevent easy identification of the presence of open source code); home-grown systems are notoriously unable to successfully identify such code.

3. Audit the source base with one of the commercially available software solutions: These solutions, from companies like Black Duck Software, typically use sophisticated techniques and algorithms to sift through a source base to identify the presence of open source code. In comparison to a typical home-grown tool, commercial offerings usually contain a larger proportion of the available techniques to identify open source code. Furthermore, via the use of a technique called code printing (analogous to fingerprinting, in that pattern recognition is used to establish matches), these automated products can identify as open source even deliberately obfuscated code; they also can examine a binary code object and establish whether open source code is present.

Using one of the commercially available solutions provides the highest level of assurance regarding the thoroughness of the audit procedure. Even with their use, however, employee interviews are important, because the same segment of open source code may have been released under several different licenses. Interviews can establish under which license the code was obtained.

Beyond auditing the source base to identify any open source code present in the source base, these solutions also offer information regarding the obligations presented by individual open source components found within a source base, thereby providing a starting point for mitigation efforts.

Whatever technique is used, a thorough audit of the source base in question is vital. Without knowing the actual state of affairs, it is impossible to know whether the company is in compliance with the obligations it is under due to its use of open source.

The outcome of this task is a complete list of what open source products are present in the subject code base along with the obligations of those products. Because the obligations are based, in part, by the mode of use of the open source product, the audit must identify the obligations according to use

profiles. Only by this method may the company be sure it is complying with the conditions of the open source licenses carried by the source present in its code base.

Mitigating Open Source compliance issues

The output of an open source audit often – even usually – identifies elements of non-compliance with the obligations of the open source components present in a company’s source base. While open source license holders do not impose license fees, they are quite insistent that the product’s use comply with its license obligations. Failing to comply can force the company to quickly re-engineer products to come into compliance. Obviously, even in the absence of licensing fees, this still presents an actual cost for compliance efforts.

Beyond the actual costs are other, non-monetary costs. For example, a company that is heavily involved with open source markets (e.g., a hardware company that sells products for Linux-based computers) is dependent upon good will from open source advocates -- being discovered as non-compliant with open source license conditions invariably will harm the company’s reputation with an important constituency.

Companies that use open source as part of their production environment can also find compliance issues that require mitigation efforts. If the license conditions of a given open source component cannot be complied with, modifying the production environment can be extremely disruptive to business operations, imposing both monetary and non-monetary costs.

It is crucial, therefore, for a company to quickly respond to its open source code audit and come into compliance with the open source license terms and conditions. Internal or external experts can review the audit results and identify mitigation actions that should be taken to ensure compliance. This review process can be time-consuming and expensive, however, and is prone to the kind of mistakes typical of repetitive manual work. An alternative for identifying mitigation actions is to use the automated commercial solutions discussed earlier. As part of their license identification, they identify obligation actions in a list format. This list may be used as a working “punch list,” enabling methodical working through of mitigation actions, with the outcome being a high level of assurance that the company is now in compliance. Whatever review method is used, it is critical that all mitigation requirements be identified and executed. Failing to do so presents the company with an ongoing risk exposure.

There are a number of actions a company may take to come into compliance with an open source license. Merely listing the open source products and licenses in a product’s end user documentation might be sufficient. On the other hand, if an open source component carries a GPL license and is incorporated in a company product that is sold with a proprietary license, a significant amount of work may be necessary to restructure the product and remove the component. The important thing is that the company identify all compliance issues and achieve license compliance.

Audit Phase Outcomes

The Audit Phase is what most people think of when they decide to “get control” of their use of open source software. However, without an initial policy evaluation and creation phase, performing an audit inevitably raises the question “Now that we know about what we’re doing with open source, what shall we do?”

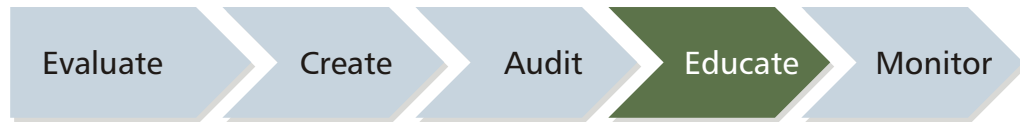
With a foundation of understanding the business goals and use profiles, however, the company can judge if its actual open source use is consistent with those goals and the licenses carried by the open source components discovered during the Audit phase.

There are two critical outcomes from the Audit phase of an Open Source Policy:

- > A list of the open source components identified during the examination of the company’s source base along with the licenses associated with those products.
- > A mitigation action plan to address any conflicts between the company’s business goals and use profiles and the actual use of open source products in the company’s source base.

With a thorough plan to audit the company’s source base and a committed attitude to address mitigation issues, a company can be assured that, at the conclusion of the Audit phase, it will be in compliance in its use of open source.

Phase 4: Educate Employees



Phase Goals	Phase Tasks	Phase Outcomes
Ensure All Employees Understand and Comply with Company Open Source Policy	Create Open Source Policy	Employee Awareness of Open Source Policy
	Train Current Employees regarding Open Source Policy	Compliance with Open Source Policy
	Integrate Open Source Policy Training into New Employee Orientation Process	Efficient and Low-Risk Use of Open Source Software

It is critical that all individuals involved with procuring, creating, modifying, or deploying software be aware of their organization’s open source policy and follow its directions regarding open source. A vital step in ensuring that all employees follow the company’s Open Source Policy is educating them about it.

Any company that has developed an Open Source Policy needs to consider two groups when creating its open source education plan:

1. The first group is current employees. While some of them will probably have been involved in creating the company’s Open Source Policy and many of them will have undoubtedly heard about the policy, most will be unaware of the policy’s existence or its specific requirements.
2. The second group is employees that will join the company in the future. Obviously, they will not be aware of the policy prior to being hired. While they may informally learn portions of the policy from other employees, this minimal awareness is not sufficient to ensure full compliance with the Policy.

Consequently, the company must plan a two-pronged Education phase: a one-time first phase to communicate the new Open Source Policy to relevant employees currently employed; and an ongoing second phase to educate new hires on a periodic basis as necessary.

Creation of Educational Materials

The purpose of the Education phase is to make everyone aware of the Policy and to follow its directives. Therefore the education materials that are created should include the following information:

- » [An overview of open source](#): It's important that individuals being educated in the company's Open Source Policy understand what open source is. Introducing employees to the Open Source Definition promulgated by the Open Source Initiative provides a convenient way to introduce open source software and distinguish it from proprietary software.
- » [An overview of open source licenses](#): A discussion regarding the different types of open source licenses and their compliance conditions is important so that employees can understand the rationale for the company's policy.

[A description of the ways the company uses open source](#): By describing how the company uses open source and mapping it against the license overview just concluded, the compliance requirements for the company can be outlined.

- » [An outline of the company's Open Source Policy](#): This ensures that attendees understand how they are expected to manage their interactions with open source software.
- » [Presentation of the processes employees must follow to interact with open source](#): This discusses the request forms that employees must submit to gain approval to use open source software in their products as well as to gain approval to participate in open source projects outside their normal work duties. If open source auditing is integrated with the normal project management processes, that should be presented in this section as well.

Delivering Education

When the company's Open Source Policy is first created, it's important that it be communicated personally. There will probably be some engineers who will feel that the Policy and its processes are an intrusion into their work. Communicating the Policy in person gives everyone an opportunity to ask questions and express their opinions.

The Policy may be presented in a formal classroom setting, less formally in "brown bag" discussions, or via web-based training mechanisms. No matter what delivery vehicle is chosen, every employee who works with open source software must participate and sign a document that they acknowledge and understand the company's Open Source Policy.

For new employees, education about the company's Open Source Policy should be integrated into existing orientation information. Most companies have new employees spend a portion of their first day or two learning about the company and signing up for a health plan, 401k, and so on. Part of that orientation is typically a discussion about the role of intellectual property, along with having the new employee sign a document indicating he or she understands the company's intellectual property rules. This is a perfect opportunity to incorporate the company's new Open Source Policy and present it as part of the company's entire intellectual property policy.

In addition to one-time education events, the policy should also be presented periodically to ensure employees are reminded about the policy and its requirements. Furthermore, the policy should be posted online so that employees can refresh their memories regarding the policy between official presentations in training or organizational meetings.

Education Phase Outcomes

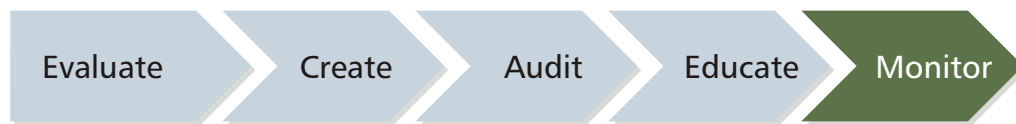
Expecting someone to follow company policy without communicating it is folly. All employees who work with open source in some fashion must be educated about the company’s Open Source Policy so that they can understand it and follow its requirements.

When the policy is first created, an intensive communication effort must be made to cover all current employees. Open Source Policy education should also be made part of the company’s orientation materials presented to new employees.

By educating all current and future employees in the company’s Open Source Policy, the company can ensure that compliance with the policy will be high; in addition, this education will ensure that no one has any reason to assert that they were unaware of how they should use open source software, thereby preventing late discovery of open source presence in products that might delay product release.

Education can help the company’s open source efforts be managed efficiently, and, as noted, it’s unrealistic to expect an employee to comply with a policy unless it is fully communicated. By creating a formal education effort, the company can gain the most benefit from its Open Source Policy.

Phase 5: Monitor Ongoing Policy Compliance



Phase Goals	Phase Tasks	Phase Outcomes
Ongoing Monitoring of Open Source Use to Raise Compliance and Reduce Risk	Create Monitoring Mechanisms	Efficient Use of Open Source Software
	Integrate Monitoring into Existing Project Management Process	Complete Monitoring Through Use of Automated Audit Software
	Regular, Periodic Auditing of Company Source Base	All Open Source Obligations Observed to Reduce Risk Exposure

The Monitor phase of the Open Source Policy is the ongoing steady-state management of open source within the company. Once the Open Source Policy has been introduced into the organization and it is being followed, it should be regularly monitored to verify compliance.

Creating Monitoring Mechanisms

The most effective method of ensuring compliance with a company’s Open Source Policy is to integrate it into the organization’s project management process. Most organizations have a formal project management process that tracks projects by key milestones. Attaching the Open Source Policy to that process enables the key tasks of the policy to be tracked as part of the milestone tracking that already

exists. Since projects are closely managed by milestone, this will raise the compliance level of the policy.

It should be relatively straightforward to map the Open Source Policy tasks to the appropriate milestones within the organization's project management process. For example, at the design phase milestone, compliance with the Open Source Policy process would be established by:

- > Identifying all open source components that have been selected for the project.
- > Verifying that open source use requests have been submitted and approved for these components.
- > Verifying that the OSRB has approved the use of these components.

Later in the project, at the Release to Market milestone, compliance would be established by:

- > Verifying that product files are organized to ensure GPL code is segregated from proprietary code.
- > Verifying that all necessary attribution is present in product documentation.
- > Verifying that source code is available in some format for all GPL-licensed code.

Of course, as many have observed, actions may differ from words. While verbal verification may raise compliance levels, verifying compliance with automated tools can validate verbal assurances and guarantee that non-compliance does not occur inadvertently or deliberately. The same automated options described in the Audit phase of this whitepaper may be applied during the Monitor phase as well. While home-grown tools may be a better choice from the perspective of cost, the commercial offerings provide more functionality and do not require employees to be diverted from regular duties to work on the home-grown system.

Whether a home-grown or commercial system is used, the output from the product should become part of the milestone reviews that are part of the organization's project management process.

[Integrating monitoring into engineering process](#)

Another option to take advantage of automated monitoring systems is to integrate them into the ongoing engineering process. For example, running the auditing software once a week against a development code base ensures that new insertions of open source code are identified quickly, offering the opportunity to address compliance issues early in a project's lifecycle.

Incorporating auditing software into the engineering process places responsibility for compliance with individual engineers, which is ideal from an organizational perspective – the party responsible for incorporating the open source product is responsible for addressing any compliance issues associated with the product.

[Monitoring Phase Outcomes](#)

Making the Open Source Policy part of the organization's project management process helps it be efficient in using open source. By integrating policy tasks into the existing process, compliance rates are raised with the added benefit that open source is perceived as just one of the software options available to the company.

Integrating automated audit systems into the project management and engineering processes verifies compliance with the policy and with the requirements of the licenses carried by the open source components used by the company. While automated audit systems are not a prerequisite for successful monitoring, they make the process easier.

Ongoing monitoring ensures that the company is observing all its obligations regarding open source software, thereby reducing its risk exposure to license compliance issues. An ongoing program of monitoring is important so that the company will never find itself late in a project needing to embark on a crash compliance program.

Conclusion

Establishing and following an Open Source Policy is appropriate for any organization making open source software an important part of its software stack or commercial offering.

This whitepaper has presented a five-phase Open Source Policy process. Based upon the Navica Open Source Policy Process, this process enables companies to create an open source policy and manage their use of open source software.



Beginning with an [Evaluation](#) phase, the process establishes the business goals and product configurations that make up the rationale for open source usage. Once Evaluation is complete, the company may [Create](#) an Open Source Policy for the organization, outlining the processes the organization will follow in using open source as well as how it will participate in open source activities.

The [Audit](#) phase assesses how well the organization's actual open source use matches the requirements laid out during the Evaluation phase. Auditing provides concrete evidence about what open source products are currently being used and also provides the basis for the company to implement mitigation activities that will bring it into compliance with its open source obligations.

[Education](#) is a necessary component of any Open Source Policy since employees cannot be expected to adhere to the Policy without understanding it fully. Both current and future employees must be educated; this whitepaper offered some recommendations about how to deliver training at the time of the Policy creation as well as making it part of employee initial orientation.

Once the Policy is in place and all products are in compliance with open source licensing conditions, an ongoing [Monitoring](#) phase is important to continue obligation observation. Linking license compliance activities to existing project management milestone reviews offers the best opportunity to ensure compliance. Integrating the Policy into existing engineering practices ensure the most efficient possible monitoring of open source use.

Risk reduction is an important element of any IT endeavor, and implementing an Open Source Policy helps reduce risk for this critical segment of software.

NAVICA Overview

Navica is a leading open source management consulting firm helping clients take advantage of the disruptive technology wave that open source software represents. Located in Silicon Valley, Navica consults to enterprises, ISVs, and venture capital firms throughout the world. Representative clients include CMEA Ventures, Emulex, SugarCRM, Compiere, OpenLogic, and Red Hat. More may be learned about Navica at its website www.navicasoft.com

To contact Navica to request more information, email info@navicasoft.com.

About Bernard Golden

Navica's founder and CEO, Bernard Golden, is a recognized authority on open source software. Called "a renowned open source expert" (IT Business Edge) and "an open source guru" (SearchCRM.com), he is regularly featured in magazines like Computerworld, InformationWeek, and Inc. His blog "The Open Source" is one of the most popular features of CIO Magazine's website. He is a frequent speaker at industry conferences like LinuxWorld, the Open Source Business Conference, the Red Hat Summit, and he served as program chair at the inaugural Open Source in Mobile conference in Amsterdam. He is the author of "Succeeding with Open Source," (Addison-Wesley, 2005, published in four languages), used in over a dozen university open source programs throughout the world.