Most companies using open source software know they need an open source policy. However, when it comes to creating a policy companies often don’t know where to start and spend months debating policy details and researching options. This guide is intended to help you write an open source software policy. But perhaps more important, it will also help you figure out who to include in the policy creation process so that your company is likely to agree upon and use the open source software policy once it’s been written.

Why Do You Need an Open Source Software Policy?

At first many companies question the need for an open source software policy—primarily because they think it will be too difficult to create. Policy creation does require a lot of work and cooperation, but by following this guide you should be able to create and roll out an open source software policy in less than a month.

Some of the main benefits to having an open source software policy include:

- **Ensuring the company is in agreement about how to use open source software.** Companies often start drafting an open source policy when somebody in management realizes they don’t know how much their IT department or software products depend on open source software. A clearly-stated policy can help ensure that everyone in the company is on board with your open source strategy and that employees feel like they’re empowered to use open source software where appropriate.

- **Maximizing the benefit of open source software.** By creating a policy, you will put processes in place that will enable employees to use open source software effectively as well as share knowledge and workload between teams. For example, three different teams won’t independently figure out how to get support for the same project, and different employees won’t independently evaluate the same upgrade. Having an open source software policy and sharing information across the enterprise will enable your company to maximize the benefits of the open source software it uses.

- **Minimizing the legal, technical, and business risks of using open source software.** Executive management and attorneys are often very concerned about being sued for using open source software, getting caught without sufficient technical support, or receiving bad publicity related to how open source software is used. An open source software policy can not only minimize those risks but also show concerned employees how the company is addressing those needs.

The Process of Writing an Open Source Policy

The key to writing an open source software policy is just to get started! Companies typically either write, approve, and adopt an open source software policy within a month, or else they spend months working on a policy and yet fail to gain company-wide approval on the finished product. By following these steps you can ensure your company has an approved and adopted policy within a few weeks:

- **Identify key stakeholders**

- **Get stakeholders to buy into the concept of a policy**

- **Figure out your company’s strategy**

- **Create the first draft of the policy**

- **Get widespread review and acceptance, starting with your stakeholders**

- **Repeat last two steps as necessary**

Stakeholders

Your first step—the one important to making sure that your policy is adopted—is to identify the stakeholders in your policy. There are several types, ranging from those who care desperately because you might change their ability to do their job (like developers) to those who care desperately because they think a bad policy might get them fired (like CIOs and those who report to them.) Be sure to include people who you think will disagree with your policy—better to address their disagreements upfront than to have them fight the policy because they don’t like one part of it.

While you’ll have to use whatever strategies work best in your organization, getting all of the stakeholders involved as soon as possible will help your policy get widely adopted more quickly. The more your stakeholders feel like it’s their policy, the better. The best case scenario is if they can say they reviewed it and feel comfortable with you (or whoever is writing the policy) taking care of the details.

As you put together your list of stakeholders you should consider:
Developers—the people who will have to follow the policy

IT staff, as they probably download and use open source software

Managers of teams that use open source software

Attorneys

CIO and staff

Technical architects; many companies have architectural committees, and they should be involved

You’ll probably have two groups of stakeholders: the key stakeholders who will need to approve the policy, and the people who will be affected by the policy.

**Strategy**

An open source software policy is meant to help your company’s strategy—both its general strategy and its open source strategy.

For example, you might see open source software as a way to reduce your IT costs. In this case your strategy should not be to “use as much open source software as possible,” but rather to “reduce IT costs by using open source software.” This will enable all of your employees to make the right decision when it comes to choosing between open source and proprietary solutions. The policy will then help your company reduce IT costs—not just by encouraging the use of open source software that has no licensing fees associated with it, but by also leveraging the preexisting open source knowledge of your IT staff.

On the other hand, you might want to build an online community around your company’s product. In this case your strategy might be to use open source software in order to encourage outside developers to help build out features that enable your community to grow. Your interest is in the community, not in making money through software sales, so you might even want to open source your own software and grow a community around it.

Identifying your open source strategy upfront will not only help you figure out your policy, but it will also help you explain to others why it’s the right policy for your company.

Here’s an example of what an open source strategy statement might look like:

**Strategy**

Our company will use open source software in order to:

- Lower total cost of ownership
- Add security to existing solutions
- Enable faster time to market by using existing solutions
- Leverage the knowledge that the company’s technical resources have in existing open source technologies

Open source software should be evaluated in all software acquisition processes as both a viable alternative as well a way to negotiate prices on non-open source software.

**Background Statement**

Many open source software policies start by talking about the problem they’re trying to solve. Whether or not you need this section probably depends on your company. You may need this section if:

- Management doesn’t know how much open source software the company uses or depends on.

- There are widely varying opinions on how much open source software is used.

- There’s an open source software rule or policy that conflicts with reality (e.g., “No open source allowed,” but your IT infrastructure is built upon open source software).

- There are big disagreements on how the company should use open source software.

If you decide you need a background statement, make it brief. Some key things to include in the background statement (if you know them) are:

- How much open source software the company currently uses.

- How much open source software has or will save you—not just monetary savings, but also consider reductions in development time, consulting time,
learning curves, and so on.

- Any risks to be aware of—for example, your 24x7 website depends on an open source software package for which there is currently no defined support process.

A background statement in an open source policy might look like this:

**Background Statement**

Our company is using 10 open source software packages. Our IT staff estimates that the company has saved...

However, most of these open source packages have been downloaded from the Internet by IT engineers, and there are not adequate support contracts in place. If something goes wrong our website could go down, taking the IT environment to work nights and weekends to repair the damage.

While open source software provides a lot of advantages, both technical and financial, it also comes with licenses and requirements that we need to be sure to follow in order to manage the legal and technical risks.

Due to the unsettled business and legal atmosphere surrounding open source software, we are implementing this Open Source Policy to help assess the company’s current use of open-source software and to manage the risks associated with the use, modification, and distribution of open source software by our company in the future.

**Scope**

Your company may have many policies, and it may be obvious who they cover. However, with open source software policies it is often necessary to define not only who is covered but what is covered.

**Who's Covered**

Most companies with an open source software policy adopt the policy company-wide. However, for many organizations it makes more sense to have a very brief company policy that simply states the strategy and provides some general guidelines for how to use open source software. Then, each division is allowed to elaborate and expand on the policy to meet its needs. In other cases, a particular division creates a policy and later tries to get the company as a whole to adopt it.

Don’t forget to specify subsidiaries and agents in this definition. You might want to consult with an attorney about whether giving open source software to the company’s agents or subsidiaries is considered distribution, as distribution triggers clauses in some open source software licenses.

Also, you might want to specify whether or not the open source software policy applies to everyone or just to employees in certain jobs or roles. For example, you may not care how your IT staff uses open source software in the internal IT environment, but you want to ensure that all software developers working on applications that are distributed to others are aware of the open source software policy.

Whatever you decide, make sure it’s clearly stated and that you have the right people review and approve the policy. The example below illustrates one approach to defining who’s covered by a policy:

**Who's Covered**

This policy is mandatory and applies to all company personnel, all of the company’s wholly owned subsidiaries, and any persons or independent contractors developing or managing technology for the company. It is the responsibility of the manager in charge of an independent contractor to ensure that the independent contractor is aware of, and follows, this policy.

This open source policy is owned and maintained by the Open Source Review Board. All questions about this policy should be directed to opensource@ourcompany.com.

The Open Source Review Board consists of representatives from the following groups:

- OSS/Program Manager – John
- Engineering – Mike
- IT – Kate
- Legal – Ben
- Sponsor (CFO) – Joe
- Procurement – David
- Architecture Steering Committee – Jennifer

**What’s Covered**

There’s a lot of misunderstanding and disagreement about what exactly qualifies as "open source software." One common misconception is that freeware, free to use,
and free for personal use only licenses are open source software licenses.

Companies typically decide that any software released under a license approved by the Open Source Initiative (OSI) is considered open source software. You need to define what is covered and what is not covered by your company's open source software policy. Many companies have different standards for open source software used in IT, development, and production environments.

You'll also need to think about what qualifies as open source software and what usage cases need to be covered.

- Does the policy apply to using software inside the company?
- Does it apply to shipping software?
- To freeware? People often assume anything that doesn't cost money is open source software—that's not true, and you need to explicitly include or exclude that type of software.

An open source policy might address the issue of what qualifies as open source with language like this:

**What's Covered**

All open source software packages—whether the open source is used internally, as part of a product, or as part of a web service—needs to be reviewed through the Open Source Approval Process.

For the purpose of this policy, open source software is defined as any software released under an OSI-approved license (see http://opensource.org). Software released under licenses that look similar to OSI-approved licenses should also be reviewed through the Open Source Approval Process.

Freeware is not considered to be open source software and should be reviewed through the regular software procurement process. Software that is free to use or obtained from the web is not considered to be open source software unless it is also released under an open source software license.

**Ownership**

Your open source software policy will be a living document. As business conditions change, your company becomes more comfortable using open source software, and new open source software packages and licenses become available, you'll want to adapt the policy to the new situations. In addition, an individual or team needs to be available to answer questions, provide training, and approve any exceptions.

Although an individual usually writes the open source software policy, a committee or open source review board should own the policy. A review board is most effective when it represents the main divisions in the company. If every group feels like they're adequately represented, you'll get better buy-in and compliance later on.

**When to Make Exceptions and Who's Allowed to Make Them**

Only a couple of people should be allowed to make exceptions to the open source software policy: the owners of the policy, and the business group or manager who's allowed to decide that the business benefit outweighs the risk of breaking from the policy.

**Approval Process**

While it's easier create an open source software policy before establishing an approval process, in reality the opposite usually occurs. Someone is responsible for making open source decisions, and he or she ends up pulling together a group of people to create an informal approval process. Once established, even informal approval processes can become quite elaborate and efficient.

If you don't have an approval process, you'll want to send one up. If you have one, you'll need to review it with your open source policy in mind.

**Who's Part of the Open Source Review Process?**

You may want to establish a cross-divisional open source review board. This is usually the same team that owns the open source software policy. It should definitely include an attorney or work closely with one.

You'll also want the review board to include any teams that will use approved open source software, as it's important that they agree with the policy as well as the process. They can give you feedback to make sure the process integrates smoothly with their workflow so that reviews happen quickly and issues are resolved smoothly.

**What Information Should Be Reviewed?**

Decide what information should be included in the review process. At a minimum you'll want each open source usage request to include:

- Name of the employee submitting the request.
- Name and version number of the requested open source software package.
• Source of the software (website from which it was downloaded).

• List of the licenses associated with the software and any bundled components.

• Business reason for the request.

• Platform on which the software will be installed.

• Plan for integrating the open source software into other software currently being used or produced.

• Whether or not there are plans to modify the source code (Y/N). If yes, explain the reason for modification.

• Whether or not there are plans to distribute the open source software, either modified or not (Y/N). If yes, explain the reason for distribution.

• Plan for supporting the software, monitoring for security updates, and performing upgrades.

Next, decide on the process for reviewing requests. You’ll probably save a lot of unnecessary reviews by asking local management and attorneys to review requests before they’re escalated to the corporate open source review board.

Best practices for open source approval processes include:

• Ensure quick turn-around on all requests. If the approval process takes weeks, developers will skip it because they need to download open source software and get their job done. Your approval process should take one or two weeks at most.

• Ensure all review board members are committed to the process. You should be prepared to replace people as the original members move onto other interests or discover they don’t have enough time anymore.

• Have a really good “pre-approved open source list”. You can pre-approved by license, package name, and/or usage models.

• Get lots of attorneys involved. While there may be an attorney on the review board, it will help if you have attorneys close to the team involved. They will best be able to understand how the open source software will be used, and they can help field questions. Educate these attorneys over time.

• Create an exception process, but do your best to set it up in a way that minimizes the use of exceptions.

• Make public within your organization all open source requests as well as the review board results. If people can see what has been requested and why certain packages were approved or denied they can tailor their requests appropriately, saving you a lot of time.

• Define some standard business reasons for your organization. Depending on your open source strategy these could vary from “saving time by using existing technology” to “saving money by eliminating licensing fees” to “getting buy-in from technical adopters by enabling them to see how product is developed.”

Audits

Although your policy can require all open source software be reviewed and approved through a defined process, it’s impossible to enforce by brute force—open source software can easily and freely be downloaded from the Internet. In order to make sure your approval process is effective and that employees are following it, you’ll need to define an audit process.

The flowchart below illustrates a typical open source audit process:
Sourcing

Your policy should explicitly state the websites and methods through which employees are allowed to obtain open source software (sourcing) and how to decide whether a particular open source package is the right piece of software for the job (selection). Who can download software? Where can they go to download it? Do they need permission before downloading? Before using? Before distributing?

Many companies allow developers to download software and try it before going through the approval process. Attorneys are often uncomfortable with this because employees can use the software before the license is reviewed. However, businesses typically balance this concern with the fact that the risk is relatively small while the developer is just evaluating the software.

Most companies end up adopting a free-for-all open source download and maintenance program—every team that needs an open source software package is permitted to download it and figure out their own support options. This is highly inefficient and often leads to the situation where a company is using many different versions of the same software package as well as duplicating evaluation and upgrade processes every time a new version is released.

Other companies have adopted the idea of an open source repository in which approved open source packages are stored. Employees who need to use open source software are required to download it from the repository instead of the Internet. While an excellent idea, this approach often leads to failure because it’s difficult for a single company that’s not in the open source software business to create and maintain a comprehensive open source repository. The repository often ends up outdated and missing many useful open source software packages. If you go with this approach, you’ll need a very good process for handling requests to add new open source packages or versions to the repository. Alternatively, you can use an open source repository provided by a third-party, like OpenLogic Exchange (OLEX) from OpenLogic.

A policy that defines how and where employees can obtain open source software might look like this:

Sourcing

You may try out open source software downloaded from:

- The OpenLogic Exchange (OLEX) repository. You can create an OLEX login by visiting https://olex.openlogic.com. Once you have created a login, you must request to be added to the command's OLEX account by contacting support@openlogic.com.

- Red Hat, as the company has a support contract with Novell.

Software downloaded from any other sources should be separated from production systems until approved through the Open Source Approval Process.

Alternatively, your policy can allow employees to make a judgment call when it comes to sourcing:
Open source software may be downloaded from any reliable source as long as it is approved through the Open Source Approval Process before it is integrated into any of the company’s internal systems or products. Employees should remember that anyone anywhere can upload software to repositories like SourceForge, and just because software is downloaded from a known website does not mean it is safe.

Note that while many attorneys recommend that no software be downloaded until the license is reviewed and approved, this is rarely followed. Your policy needs to take into account the legal risk of downloading software for testing or trial purposes before attorneys have a chance to review it. Adding a clause like this can help mitigate the risk:

Software downloaded from any other sources should be separated from production systems until approved through the Open Source Approval Process.

Selection

Selection is the process of deciding whether or not a particular software package meets your needs and quality standards. In addition to any testing and standards processes already in place, you might want to consider additional aspects for reviewing open source software such as:

- Support: There are options for [support for open source software](#), but which options are available and which you should choose may vary from package to package.
- Community: Is there an active open source community for the package? Are bugs fixed quickly?
- Quality: Is the software package reliable? How does it compare to similar open source packages?

For a list of other factors to consider around open source package selection, see the [OpenLogic Certification Process white paper](#).

A policy might approach the issue of open source selection with language like this:

**Selection**

The Open Source Review Board will specify the certification criteria for open source software. Under all circumstances, the software must meet the company's security and reliability standards and pass the IT department's tests. Each team using open source software in a production solution is responsible for making sure all of the open source software they use meets any additional criteria specific for the production environment.

Your policy may also outline what to do in case an open source project "ends" or forks. For example, the policy might define a timeframe in which the team has to find a different open source package to replace the original package:

For projects that are abandoned (meaning there is little activity on the mailing lists and little new code is being written), an alternate project (open source or proprietary) must be used.

Mergers and Acquisitions

Don't forget to include a section in your policy about mergers and acquisitions. Enterprises often fail to investigate open source usage or policy until after a target company has been acquired. It makes much more sense to check beforehand. You'll need to make sure your company's mergers and acquisitions team is aware of the internal open source policy as well as the need to investigate open source policy and usage at target companies prior to completing any acquisitions. Open source tools such as [CheckSlogy, Deep Discovery](#), and [Deep Discovery](#) can be run to identify the open source software deployed within a target organization.

A policy might approach the issue of mergers and acquisitions with language like this:

**Mergers and Acquisitions**

In the case of an acquisition, any and all IT solutions and products within the target organization should be thoroughly reviewed for open source software. Any discovered use of open source software should be reviewed by the Open Source Review Board. Adequate time should be allocated so that any issues that are uncovered can be discussed and addressed.

Support

Technical support in the open source world has a bad reputation. While many people think there's no support available for open source software, the real problem is that there are too many choices for open source software support—and few of them resemble what people are used to in the proprietary world.

Options vary from do-it-yourself (by following mailing lists and even fixing critical bugs yourself) to paying a third party for a complete support and maintenance contract. The table below shows some of the common options:
An open source policy might address the issue of technical support with language like this:

Support
During research and development, it is acceptable to rely upon internal support and open source community mailing lists.
Commercial support is required for all open source components prior to their deployment in a production system.
There must be a support plan in place before the team uses open source software in production solutions.
Acceptable methods of support are:
- Provided by internal team or IT department (support plans and staffing must be in place)
- Combination of internal resources supplemented by an external support provider
- SLA signed with an external support partner that has been reviewed and approved by purchasing and legal

Maintenance
Once the decision to use open source has been made, the process of downloading it and getting it to work is often challenging and satisfying. However, it can be frustrating to track the project for security updates, solve problems, and figure out when to upgrade to a new version.

Your open source policy should provide guidance on maintenance concerns by addressing the following issues:

- Security updates: Who’s going to be responsible for staying abreast of security updates? This often requires subscribing to a mailing list.
- New versions: Who’s going to figure out when you should upgrade to a new version? Deciding when to upgrade can be hard because open source software projects release new versions often. While some projects are very good about specifying what’s new, what’s changed, and what’s gone, others are not. Someone will have to evaluate each release and decide whether or not to upgrade. In addition, you’ll need a process to ensure that only one or two versions (as few as possible) are deployed within the company. In particular, you’ll want to ensure the company doesn’t fall too far behind the latest version available from the open source project, as communities rarely support previous versions for long.
- Bugs: When a bug is discovered, what’s the process for reporting it to the open source project? Your policy should also specify when and how employees can fix bugs and submit bug fixes to open source projects. (You may want to provide some training about how best to create and provide patches, as it’s in your best interest to make sure patches are accepted upstream.)

The maintenance section of your open source policy should specify whether or not company employees can modify open source software. While you’ll probably want to encourage people to not modify open source software (unless they are planning on fixing a bug or adding a new feature to contribute upstream), it’s likely that some packages will require modifications to work in your environment or process. Your policy should specify when that’s allowed and how changes are documented for maintenance.

An open source policy might address the issue of ongoing maintenance with language like this:

Maintenance
In general, company employees should avoid modifying the source code of the open source software they use. However, employees are permitted to make minor changes to configurations and other similar changes to allow a software component to function properly within the intended operational environment. Substantial functional changes should not be made.
The company discourages this practice because of the difficulty in maintaining modified (forked) versions of an open source package. An exception to this rule may be granted if the open source project/community is willing to accept such changes as a contribution.
Open source software can be modified in order to add new features or to install and integrate the software into the company’s environment. All changes should be documented and should be submitted back to the open source project/community. Care should be taken to make sure any changes are written in a way that will be acceptable to the project/community. The goal should be to avoid losing forked or customized versions of any open source software packages.

Contributions
Your open source policy should include a clear statement about whether or not employees can contribute code changes to open source software projects. There are several situations in which you'll probably will want employees to contribute. For example, if an employee fixes a bug, you'll want him or her to contribute the patch to the project so that you don't have to maintain a unique patch through every version upgrade. In addition, by contributing code your company might gain credibility within the open source community. If your company actively contributes to a project, the community will be more likely to recognize employees and quickly respond to their emails when issues arise.

**Situations to consider when thinking about whether employees can contribute code changes to open source software projects include:**

- Can employees post questions and answers on mailing lists? While many companies are afraid of what employees may accidentally reveal on mailing lists, it's best to give them the proper training on confidentiality and then let them work on mailing lists. Not only will they build credibility for themselves and the company, but it's often one of the best ways to get help with a problem. Disallowing employee contributions on mailing lists will impede employees' ability to use and support open source software solutions.

- Can employees post bug reports to the bug tracking system? Unless there are very good business reasons not to, you should allow employees to post bug reports in order to ensure they can be fixed in a timely manner that will work for the company.

- Can employees fix bugs and submit their patches? Employees often need to fix simple bugs in order to get things up and running. When they do, it's in the company's best interest to ensure the bug is fixed in a general way that's acceptable to the project so it can be submitted upstream. This will simplify support over the long term and help build credibility for the company.

- Can employees contribute features to an open source project? Can they become project contributors?

- Can employees work on open source software projects in their free time? On the same project(s) they work on in the office? On unrelated projects? How do you decide if a project is unrelated to the company's business?

An open source policy might address the issue of open source contributions with language like this:

**Contributions:**
The company recognizes that open source software continues to develop through the participation of talented programmers in open source projects, and that these contributions are donated to the public. The company is not opposed to employees contributing source code to open source projects. However, in order to protect the company's interests, explicit permission to do so is required.

Most changes, such as those required to make open source software packages work within the company's environment, are allowed. No other changes should be made. All changes should be documented and should be submitted back to the open source project/community. Care should be taken to ensure changes are written in a way that will be acceptable to the project/community.

You must receive permission from the Open Source Review Board prior to making a contribution of any source code when that source code:

- was created during business hours
- was created with the use of company resources
- interfaces with company systems

Some open source policies state that employees can only interact with open source communities using personal email addresses, not company email addresses. Note that if you include this rule in your open source policy, your company will never develop any credibility with the community.

**Communication**

Although many companies try to hide that fact that they use open source software, it rarely remains a secret—especially for organizations that decide to provide their own support using internal resources. It's best to establish a policy for how employees should communicate with open source communities and other interested parties, even if you intend to limit communication about open source usage as much as possible.

Your open source policy should provide guidance on communication by addressing the following issues:

- How can employees communicate with open source software projects? (For more on this topic, see the Maintenance section above.)

- How can employees communicate about open source usage at conferences? Employees will likely want to attend open source software conferences, which provide an excellent opportunity to meet open source project maintainers. If employees can talk about which open source packages they're using and even present case studies, it will help them meet the right people and gain credibility for your company. The open source software community likes hearing about how their products are being used.

- How can employees communicate about open source usage in documentation? If you ship products that contain open source software, you may have to display the open source software licenses in the product documentation (depending on the licenses). You might also want to say something to end users about what you're using and why, depending on the type of end users. For example, if you sell TVs that use open source software you may need to display one or more licenses in order to comply with them, but you probably don't need to spend a lot of time talking about the software installed on the TV. If, on the other hand, you distribute a toolkit for software developers, you'll probably want to allow more space in your documentation for discussion about the open source software you're using and why it's part of the toolkit.
Summary

Be sure to remain positive in the summary of your open source policy—after all, you want your employees to want to follow the policy—while also clearly stating when and how open source software usage triggers review. For example, the summary section of a corporate open source policy might look like this:

The company encourages the use of open source software in order to lower total cost of ownership, enable faster time to market, and leverage the knowledge of employees. All open source software should be downloaded only from approved sources. Any exceptions must be explicitly approved by the Open Source Review Board before the open source in question can be downloaded and deployed onto company systems. All teams planning to use open source software must make sure the software meets their needs, verify that adequate support options are in place, and go through the Open Source Approval Process. The Open Source Review Board will ensure that all requests for usage of open source software are approved in a timely manner.