Participating in Open Source Communities

オープンソース　コミュニティへの参加

Open source has become the de facto way to build software — not only in tech, but across diverse industries. As companies use open source code to build their own commercial products and services, they also see the strategic value of contributing back to those projects.

オープンソースは、技術系産業だけではなく、広く業界にまたがって、ソフトウェアを作成、構築するための事実上の標準方法となっています。また、企業が製品とサービスを提供するためにオープンソースコードを活用することが広がり、企業はオープンソース　プロジェクトにコントリビューションすることに戦略的価値を見出すようになってきました。

However, diving in without an understanding of those projects, their communities, and how they operate can lead to frustrations for those companies as well as the open source communities. Approaching open source contributions without a strategy can tarnish a company’s reputation in the open source community and incur legal risks.

しかし、オープンソース　プロジェクト、コミュニティ、そしてそれらがどのように運営されているかを理解もせずに参加すると、オープンソース　コミュニティだけでなく、参加した企業も失望することになるでしょう。 戦略なしでオープンソースへのコントリビューションを試みると、オープンソース　コミュニティにおいてその企業の評判が損なわれ、法的リスクが発生することもあります。

This guide covers what it means to contribute to open source as an organization and how to become a good corporate citizen. Learn how open source projects are structured, how to contribute, why it’s important to devote internal developer resources to participation, and why it’s important to create a strategy for open source participation and management.

このガイドでは、組織としてオープンソースにコントリビューションすることはどのような意味をもつのか、どのようにすれば良き企業市民になることができるかについての指針を提示します。 すなわち、どのようにしてオープンソース　プロジェクトを構築するのか、コントリビューションはどのように行うのか、社内開発リソースを投入することがなぜ重要か、そして、オープンソースに参加し、マネージメントするための戦略の策定がなぜ重要なのかについて学びます。

Contents

Why contribute?

How open source projects are managed

How contributions work

How organizations contribute

Be a good corporate citizen

Your contribution strategy

Final words

内容

なぜコントリビューションするのか

どのようにしてオープンソース　プロジェクトを管理するのか

コントリビューションの仕組み

どのように組織がコントリビューションするのか

良き企業市民になるために

あなたのコントリビューション戦略

結論

View All Guides »

Contribute on GitHub »

Contributors to this Guide

Stormy Peters

Senior Manager, Community Leads

Red Hat

Nithya Ruff

Senior Director, Open Source Practice

Comcast

Section 1

セクション　1

Why contribute to open source?

なぜオープンソースにコントリビューションするのか

It might be impossible to find an organization today that doesn’t benefit in some way from open source software. Some companies, like Intel, IBM, and Samsung, have entire open source programs devoted to contributing to open source communities. Other companies become consumers of open source almost accidentally when the software is brought in by system administrators or developers.

今日、オープンソース　ソフトウェアからどのような利益も得ない組織を見つけるのは不可能でしょう。 Intel社、IBM社、Samsung社のように、いくつかの企業は、オープンソース　コミュニティにコントリビューションするために。一貫したオープンソース　プログラムを持っています。 他の企業も、ソフトウェアがシステム管理者や開発者によって導入され、それを契機で、オープンソースのユーザになっています。

Many companies are commercially dependent on open source software that is critical to the success of the company, so it becomes advantageous (and necessary) to contribute to open source software projects. Since 2005, over 13,500 developers from over 1,300 different companies have contributed to the Linux kernel, and it is just a single project!

多くの企業のビジネスがオープンソース　ソフトウェアに依存していて、オープンソースは企業の成功のために非常に重要なものになっています。したがって、オープンソース　ソフトウェアプロジェクトにコントリビューションすることは必要かつ有益なものなっています。 2005年以来、1,300以上の異なる企業の13,500人以上の開発者がLinuxカーネルにコントリビューション（[contributed to the Linux kerne](https://www.linuxfoundation.org/resources/open-source-guides/recruiting-open-source-developers/)l）していますが、なんとそれは単一のプロジェクトに対してです。

“For many larger projects, we know that most of our contributors are going to be people who work at companies that need to use projects like Ceph and Gluster. We have customers, and customers often contribute to software because they’re using it. We consider both the individual participation and the company participation as success stories.”

「多くの大規模プロジェクトでは、ほとんどのコントリビューターはCephやGlusterのようなプロジェクトを使用する必要がある企業で働く人々であることがわかっています。 私たちは顧客を持っており、その顧客はそれを使用しているのでソフトウェアにコントリビューションしていることがよくあります。 私たちは、個人としての参加も企業としての参加も、ともにサクセスストーリーとして考えています。」

[Stormy Peters](https://twitter.com/storming) – Senior Manager, Community Leads at Red Hat

While some of these contributions may come from organizations that just want to give back to the community, there are plenty of strong business reasons to contribute to the open source software projects used within your organization. Here are just a few of the benefits of contributing:

これらのコントリビューションの一部は、単にコミュニティにお返しをしたいと考える組織から来ているのかもしれません。　しかし、組織内で使用されているオープンソースソフトウェア　プロジェクトにコントリビューションすることに対するビジネス上の理由もたくさんあります。 コントリビューションのメリットの一部を以下に紹介します。

・Attract talent. When you rely on open source software, the best place to find people who know the project inside and out is in the community for that project. By working publically in the community, you can attract people who see that they can get paid to work on their favorite open source project. When your employees are working side by side with these people every day, they can help you find the ones that will be a good fit for your company. (See our guide on Recruiting Developers)

**・優秀な人材を引き付けることです。**あなたがオープンソース　ソフトウェアに依存しているとき、プロジェクトを知り尽くしている人材を見つけ出すための最良の場所は、そのプロジェクトのコミュニティです。 コミュニティでオープンに仕事をすることで、自分が好きなオープンソース　プロジェクトで働くことで報酬を得ることを考えている人材を引き付けることができます。 あなたの従業員が毎日、かれらと一緒に仕事をする中で、あなたの会社に合った人材を見つける手助けにもなるでしょう。 （採用担当者向けのガイド、[guide on Recruiting Developers](https://www.linuxfoundation.org/resources/open-source-guides/recruiting-open-source-developers/)をご参照ください）

・Lower maintenance costs. Companies that start fixing bugs or adding new features and functionality to an open source project without contributing them back into the upstream project quickly learn that upgrading later to add new features or apply security fixes can be a nightmare that drives maintenance costs through the roof. Contributing your changes back into the upstream project means that they will automatically be included in future updates without incurring additional maintenance costs.

・**メンテナンスコストを削減します。**自身がオープンソース　プロジェクトに対して行ったバグ修正や機能追加をアップストリーム　プロジェクトにきちっと、フィードバックしてこなかった企業は、あとで、アップグレードしてアップストリームの新機能を追加したり、セキュリティフィックスを適用したりすることが非常に高いメンテナンスコストを必要とし、それは悪夢のような作業になるとに気づくでしょう。変更をアップストリーム　プロジェクトに戻すことで、追加のメンテナンスコストを必要とせずに、それは自動的に将来のアップデートに含まれることになります。

・Influence the direction. In open source projects, new features and functionality come from contributions, and these contributions influence the direction of the project. If you want the project to have the functionality that is important to your organization, then you need to have active contributors who can any implement potential changes. Through your contributions, you can have some influence on the direction of the project as long as your changes are aligned with the goals of the project.

**・プロジェクトの方向性に影響を与える。** オープンソース　プロジェクトでは、新しい機能追加はコントリビューションの形で行われ、これらのコントリビューションがプロジェクトの方向性に影響を与えます。組織にとって重要な機能をプロジェクトに追加したい場合は、それを実現するための変更をコード実装できる活動的なコントリビューターが必要になります。あなたの変更がプロジェクトの目標と一致している限り、あなたはコントリビューションを通して、プロジェクトの方向性にある程度影響を与えることができます。

However, you need to be a bit careful about how you engage these communities to avoid perception problems or other issues with your contributions. Every open source project has slightly different norms, expectations, and processes that need to be thoroughly understood before your organization should begin contributing. This can be achieved by having someone join the community and spend some time observing, or you can hire someone who already has a proven track record of participation in the community.

しかし、コントリビューションに対する認識などで問題が生じるのを避けるために、コミュニティにどのように参加するかについては、慎重に考えてみてください。すべてのオープンソース　プロジェクトには、少しずつ異なった、行動規範、期待される振る舞い、プロセスがあり、組織がコントリビューションを始める前にそれらを完全に理解する必要があります。 これは、誰かがコミュニティに参加し、暫くの間コミュニティを観察することで達成できます。または、すでにコミュニティに参加している実績のある人を雇うこともできます。

Section 2

セクション　2

How open source projects are managed

どのようにしてオープンソース　プロジェクトを管理するのか

At first glance, open source projects may look chaotic. People who are completely new to open source software often wonder how a group of random people can throw code together with a resulting stable product used by millions of people. It doesn’t take long to realize that this isn’t how open source software works. Almost every open source project has some structure, and the best projects will have the structure and project governance clearly described on the project website or in the documentation. (GitHub’s guides for contributors have a great overview of project anatomy.)

一見すると、オープンソース　プロジェクトは雑然としているように見えるかもしれません。 オープンソース　ソフトウェアに初めて接する人は、「無秩序な人々」で構成されるグループが、ともにコードを投稿していて、それがどのようにして、結果的に、何百万人もの人々が使用している安定した製品になるのか、とても不思議に思うでしょう。 それはオープンソース　ソフトウェアの開発方法ではないことに気づくのに多くの時間は必要としないでしょう。 ほぼすべてのオープンソース　プロジェクトは「体制」が整っており、優れたプロジェクトはその「体制やプロジェクトガバナンス」をプロジェクトのウェブサイトやドキュメンテーションで明確に記述しています。 （コントリビューターのためのGitHubのガイド、[overview of project anatomy](https://opensource.guide/how-to-contribute/#anatomy-of-an-open-source-project)は優れたプロジェクトの組織概要を提供してくれています。）

While the exact governance model varies widely across projects, there are some commonalities:

ガバナンスモデルはプロジェクトによって異なりますが、いくつかの共通点があります。

・Leader: At a minimum, there should be someone responsible for making the final decision about features, releases, and other activities. In some cases, this is a single person, for example, Linus Torvalds is the original author and has the final say on anything related to the Linux kernel. In other projects, there may be one or more committees responsible for various aspects of a project, like the Core Technical Committee that governs the Node.js project.

・**リーダー：**少なくとも、機能、リリース、その他のアクティビティについて、最終決定を下す責任者がいなければなりません。 いくつかのケースでは、これは一人の人間です。例えば、Linus Torvaldsは、Linuxカーネルの原作者であり、それに関連するものについて最終決定権をもっています。 Node.jsプロジェクトを統括するコア技術委員会（Core Technical Committee）のように、プロジェクトのさまざまな側面を担当する複数の委員会を設置しているプロジェクトもあります。

・Maintainers: Most leaders delegate some of the decisions to people who are responsible for maintaining specific parts of the project, and in large projects, these maintainers may also delegate to people who are responsible for subcomponents of their portion. For example, Linus Torvalds delegates Linux kernel documentation decisions to Jonathan Corbet.

**・メインテナー：**ほとんどのリーダーは、プロジェクトの特定の部分を保守する責任を持つ人々に一部の決定事項を委譲します。大きなプロジェクトでは、これらのメインテナーが、さらに、その中のサブコンポーネントに対する保守責任を他の人に委譲する場合もあります。 例えば、Linus Torvaldsは、Linuxカーネル　ドキュメントに関連する決定をJonathan Corbetに委譲しています。

・Committers: Some projects also have groups of people who have contributed to the project and are considered reliable and responsible enough to be allowed to commit directly to all or some parts of the project, rather than having to submit to a maintainer for review. Contributions from committers are still subject to review by maintainers or project leaders and may be reverted if there are any concerns about the contribution.

・**コミッター：**一部のプロジェクトでは、プロジェクトにコントリビューションしてきた、信頼でき、責任感のある人々のグループがあり、サブミットに対するメインテナーのレビューなしに、プロジェクト全体、または一部に直接コミットできるようにしています。 コミッターからのコントリビューションは、メインテナーやプロジェクトリーダーのレビューの対象となり、コントリビューションに懸念があれば元に戻します。

・Contributors: Many people contribute to open source projects with code, documentation and other contributions. These contributions are usually subject to a review from an experienced committer and / or maintainer before the contributions are included.

**・コントリビューター：**多くの人々が、コード、ドキュメント、およびその他のコントリビューションでオープンソース　プロジェクトに貢献しています。 これらのコントリビューションは、コントリビューションが採用される前に、経験豊富なコミッターやメインテナーからレビューを受けます。

・Users: The most important group of people in an open source project is likely to be the ones who actually use the product, since users give the project a purpose and help it grow. These valuable members of the community can provide feedback about features, bug reports and more.

・ユーザー：オープンソース　プロジェクトの最も重要なグループは実際に製品を使用しているユーザーだと思います。ユーザーはプロジェクトに意義を与え、成長を助けてくれます。これらのコミュニティの貴重なメンバーは、機能、バグレポートなどについてのフィードバックを提供してくれているでしょう。

The community is one of the things that can make or break an open source project, and having a strong, vibrant, and diverse open source community is important to the project’s success. All of the people in the roles listed above are part of this community along with people filling other critical roles in the project for documentation, marketing, user support, and so much more.

コミュニティは、オープンソース　プロジェクトを作ることも、壊すことのできるものの一つであり、強力で活気に満ちて、多様な人たちで構成されたオープンソース　コミュニティを持つことは、プロジェクトの成功にとって重要です。 上に挙げた役割を果たしている人々は、コミュニティの一部で、ドキュメンテーション、マーケティング、ユーザーサポートなど、プロジェクトの他の重要な役割を果たしている人たちとともにコミュニティを構成しています。

Section 3

セクション　3

How contributions work

コントリビューションの仕組み

The contribution process varies depending on the open source project. For example:

コントリビューションのプロセスは、各オープンソース　プロジェクトによって異なります。 例えば：

・Projects have different guidelines with information about coding style, language, formatting, bug / ticket numbers, release timing, and more.

・プロジェクトには、コーディングスタイル、開発言語、書式、バグ・チケット番号、リリースタイミングなどに関するガイドラインがあります。

・Some projects require signed contributor agreements, while others have signed-off-by or other processes.

・サインが必要なコントリビューター契約が必要なプロジェクトもあれば、GitHubのsigned-off-byなど他の方法を採用しているプロジェクトもあります。

・The project may require patches to be posted to the mailing list, but others will ask for pull requests.

・あるプロジェクトでは、メーリングリストにパッチを投稿する必要がありますが、他のプロジェクトではプルリクエストにすることを求めることもあります。

These are just a few ways that the contribution style might differ, so it’s important to start by reading the documentation about how to contribute. Many projects will include this documentation as a CONTRIBUTING or README file in the home directory of the code repository, but if not, you may need to dig into the documentation or community section of the website to find the documentation. It’s also a good idea to read some of the other documentation, community guidelines, and code of conduct if they are available to make sure that you understand exactly what behavior is expected within a particular project.

コントリビューション　スタイルはわずかですが異なっているかもしれません。そこで、コントリビューション方法についてのドキュメントを読むことから始めましょう。 多くのプロジェクトでは、このドキュメントがCONTRIBUTINGまたはREADMEファイルとしてコードリポジトリのホームディレクトリに含まれますが、そうでない場合は、ドキュメントを見つけるためにWebサイトのドキュメントまたはコミュニティのセクションやドキュメントを探さなければなりません。 特定のプロジェクトでどのような行動規範が期待されているかを正確に理解するためには、その他のドキュメント、コミュニティガイドライン、行動規範をお読みください。

If you are a first-time contributor to a project, you might consider finding a mentor or an experienced project member who can review your work and provide you with some feedback as you prepare your first couple of contributions.

プロジェクトに初めてコントリビューションする時には、最初の数件のコントリビューションについては、それをレビューし、フィードバックを与えてくれるメンターや経験豊富なプロジェクトメンバーを見つけることを考えるべきでしょう。

After submitting a contribution using the process described in the documentation, you will need to be available to respond to feedback. Common feedback would include questions about how something works or why you chose a particular approach along with suggestions for improvements or requests for changes. This feedback can be tough sometimes, so it helps to assume that the feedback is in the spirit of making your contribution better and avoid getting defensive. You may need to go through several rounds of resubmission and additional feedback before your code is accepted, and in some cases it may be rejected. There are many valid reasons why something might not be accepted, so don’t take it personally if your code is rejected, and if possible, try to learn more about why your contribution was not accepted to help increase the chances of getting your next contribution included.

ドキュメントに記載されているプロセスに沿って、コントリビューションをサブミットした後、それに対するフィードバックに返信する必要があります。 一般的なフィードバックには、どのようそれが機能するのか、特定のアプローチを選択した理由などの質問や、改善の提案、変更要求などが含まれます。 このフィードバックは時として厳しいものです。フィードバックはあなたのコントリビューションをより良いものしてくれるものという考え方で対応しましょう。自己防衛的な対応は避けるべきです。あなたのコードが承認される前に、数回の再提出や追加のフィードバックを行う必要があるかもしれませんし、場合によっては却下されることもあります。 何かが受け入れられない理由はいろいろあります。あなたのコードが拒否された場合、それを個人攻撃ととらえないようにしてください。できることなら、あなたの貢献が受け入れられなかった理由についてもっと学びましょう。 それはあなたの次回のコントリビューションが受け入れられる可能性を高めるのに役立ちます。

Keep in mind that if your contribution was accepted, you may be expected to maintain it over the long-term. This is especially true for large contributions, new features, or standalone code, like a driver for a specific piece of hardware. For small contributions and bug fixes, it is unlikely that there will be any long-term maintenance expected.

あなたのコントリビューションが受け入れられれば、あなたがそれを長期にわたって保守することを期待されるだろうということに留意してください。 特に、大きなコントリビューション、新機能、またはハードウェアの特定のドライバのような独立性の高いコードの場合は、これが当てはまります。 小規模なコントリビューションやバグ修正に対しては、長期的な保守を期待されることはないでしょう。

Section 4

セクション　4

How organizations contribute

組織のコントリビューション方法

Over the years, the relationship between some open source projects and the companies or other organizations that use or contribute to those projects has been a bit rocky. Organizations are often accustomed to forming business relationships in ways that don’t usually work for open source projects, so some organizations struggle to understand how to contribute in a way that is productive. Another challenge is that an organization can seem self-serving or troublesome if the needs of the organization aren’t aligned with the needs of the open source project, which can cause an open source community to become suspicious of the motives behind an organization’s contributions. In the past, some organizations have tried to make huge contributions that weren’t aligned with the goals of the project, and in certain projects, this history may make it harder for the community to trust organizations.

長年にわたって、いくつかのオープンソース　プロジェクトと、それらのプロジェクトを使用したり、プロジェクトにコントリビューションしたりしている企業、組織との間の関係で問題を抱えているものがあります。組織が、通常、オープンソース　プロジェクトでは機能しない方法でビジネス関係を形成することによく慣れてしまっていて、生産性高く、コントリビューションする方法を理解するために苦労している場合があります。 もう一つの課題は、組織のニーズがオープンソース　プロジェクトのニーズに合致していない場合で、コミュニティから見て、組織が利己的で、煩わしい存在にみえる場合です。オープンソース　コミュニティが組織のコントリビューションの背後にある動機を疑う可能性がある場合です。過去に、プロジェクトの目標に沿っていない大規模なコントリビューションをしようとした組織がありましたが、特定のプロジェクトでは、この歴史によってコミュニティが組織を信頼することを難しくしています。

However, there are also many success stories, like the Linux kernel, where organizations contribute in meaningful ways. The most common and easiest way for an organization to contribute to an open source project is to pay employees who have a significant amount of time devoted to participation in open source projects. In order for this to be successful, those employees need to understand the contribution processes and norms within that project to increase the chances that their contributions will be accepted. If your organization is new to a project or new to open source, you should consider hiring someone who has already contributed and is known within the open source project you want to contribute to so that they can provide guidance to the organization about contributing in ways that are more likely to be successful in that project. Experienced contributors might be willing to act as mentors to help your employees as they pursue an open source career path. (See our guide on Recruiting Open Source Developers.)

しかし、組織が意味のある形で、コントリビューションをすることで、Linuxカーネルのような、成功事例が数多くあります。組織がオープンソース　プロジェクトにコントリビューションするための最も一般的で、最も簡単な方法は、多くの時間をさいてオープンソース　プロジェクトに熱心に参加している従業員に十分な処遇を与えることです。これを上手く進めるためには、従業員がそのプロジェクトのコントリビューション　プロセスと行動規範を理解し、そのコントリビューションがプロジェクトに受け入れられる機会を増やす必要があります。あなたの組織がプロジェクトに新規に参入したり、オープンソースに初めて参入したりする場合は、コントリビューションしたいオープンソース　プロジェクト内で、すでにコントリビューションしている人や、そのプロジェクト内で良く知られた人を採用することを検討してください。そうすることで、彼らはそのプロジェクトでより成功する可能性の高いコントリビューション方法について、組織を指導することができます。経験豊かなコントリビューターは、あなたの従業員がオープンソースでキャリアパスを追求するのを助けるメンターの役割もしてくれるでしょう。 （オープンソース開発者採用（[Recruiting Open Source Developers](https://www.linuxfoundation.org/resources/open-source-guides/recruiting-open-source-developers/)）ガイドをご覧ください）

In most projects, there are other ways for organizations to participate, but these are likely to vary by project. Open source projects and the foundations that support them often need resources that can be provided by organizations, including infrastructure, funding, marketing, legal services, and much more. Many projects allow companies to sponsor or join a project in some more formal way by contributing funding and / or people to the open source project in exchange for some advisory role in the project or visibility for the company.

ほとんどのプロジェクトでは、組織が参加するための方法を提供していますが、その方法はプロジェクトごとに異なっているでしょう。 多くの場合、オープンソース プロジェクトとそれらをサポートするファウンデーションは、インフラストラクチャ、運営資金、マーケティング、法務サービスなどで、組織が提供可能なリソースを必要としています。 多くのプロジェクトでは、プロジェクトに助言できる資格（advisory role）の獲得や企業のビジビリティ向上のために、オープンソース プロジェクトに資金や人的資源の形でのコントリビューションをすることで、企業がより正式な方法でプロジェクトのスポンサーになったり、参加したりすることができます。

For example, the Node.js Foundation Board of Directors is made up of representatives from corporate members, a representative of the Technical Steering Committee, and representatives elected by the individual membership class. The corporate members comprising a portion of the board pay anywhere from $5,000 for a small organization to $250,000. While each project has a slightly different approach to sponsorship or membership, funding an open source project helps pay the project’s expenses and helps the project be successful.

たとえば、Node.js Foundation理事会の理事（[Node.js Foundation Board of Directors](https://foundation.nodejs.org/about/leadership)★原文のリンクが間違っているので修正しています。）は、企業会員からの代表者、技術運営委員会の代表者、個人会員クラスで選出された代表者で構成されています。 理事会の理事を構成することができる企業のメンバーは、小規模組織の場合は5,000ドル（小規模組織の場合）から250,000ドルを支払います。 各プロジェクトはスポンサーシップやメンバーシップに対するアプローチが少し異なりますが、オープンソース　プロジェクトへの資金提供はプロジェクトの経費を支え、プロジェクトの成功に役立ちます。

A diverse group of organizations contributes to the Cloud Native Computing Foundation’s Kubernetes project.

多様な組織がCloud Native Computing FoundationのKubernetesプロジェクトにコントリビューションしています。

Section 5

セクション　5

How to be a good corporate citizen when participating in an open source project

オープンソース　プロジェクトに参加する際、どのようにすれば良き企業市民になれるのか

If there is an underlying theme for this guide and for open source in general, it’s that every project is different. Every time you join an open source project, you’ll need to spend some time orienting yourself to the project and learning how it works.

このガイドおよび一般的なオープンソースで基礎をなすテーマがあるとすれば、すべてのプロジェクトは異なっているということです。 したがって、あなたがオープンソース　プロジェクトに参加するたびに、プロジェクトがどのように機能するかを学び、それにあなた自身を適応させていくための時間が必要になるでしょう。

For organizations participating in an open source project, each employee will need to go through this learning process for each project they participate in. Here are a few things that can help you get started off on the right foot.

オープンソース　プロジェクトに参加する組織は、各従業員が参加するプロジェクトごとに学習プロセスを進める必要があります。コミュニティと良い関係でスタートするためのいくつかの施策を以下に示します。

・Join the community. Each community will have slightly different ways of participating and different channels. Read the documentation to find out about the community and join the key communication channels. These channels may include mailing lists, forums, IRC, Slack, bug trackers, source code repositories, and more.

**・まずコミュニティに参加してください。** 各コミュニティでは、参加方法やコミュニケーションチャネルが若干異なります。ドキュメントを読んで、コミュニティについて学び、主要なコミュニケーションチャネルに参加してください。 これらのコミュニケーションチャンネルは、メーリングリスト、フォーラム、インターネットリレーチャット（IRC）、Slack、バグトラッカー、ソースコードリポジトリなどが含まれます。

・Lurk first. After you’ve joined the community, spend a significant amount of time lurking and reading the archives to soak up the culture before you start contributing. You’ll want to understand the norms and expectations of this community before you participate. The more time you spend reading and listening, the more likely it is that your first contribution will be well received.

**・最初はコミュニケーションチャネルのリードオンリー　ユーザでいてください。** コミュニティに参加した後、コントリビューションを開始する前に、まずアーカイブのリードオンリー　ユーザになり、コミュニティ文化を吸収するために相当な時間を費やしてそれを読んでください。 あなたが参加する前に、このコミュニティの行動規範や期待される振る舞いを理解しましょう。 読んだり、聞いたりするのに費やす時間が多くなればなるほど、最初のコントリビューションが受け入れられる可能性がより高くなります。

・Understand the governance. Read the documentation or website sections about project governance and leadership before contributing. You’ll want to understand how decisions are made within the project and who makes the decisions for various types of contributions.

**・ガバナンスを理解してください。** コントリビューションを開始する前に、プロジェクトガバナンスとリーダーシップに関するドキュメンテーション、またはウェブサイトの関連セクションを読んでください。 プロジェクト内で、どのように意思決定が行われ、さまざまなコントリビューションに対して、誰が意思決定しているのかについて理解してください。

・Start small. Tackle a simple bug or documentation fix to start. It will be easier to learn the process and correct mistakes on a small contribution that isn’t critical to your organization’s needs. Make your mistakes on small and less significant contributions as you work up to the more complex contributions that your organization needs.

**・小さく始めてください。** 簡単なバグやドキュメントの修正から始めてください。 あなたの組織のニーズに重要ではない小規模なコントリビューションを通して、ミスを修正したり、プロセスを学んだりするほうが簡単です。組織が必要とするより複雑なコントリビューションに取り組むために、小さくて、あまり重要でないコントリビューションでミスを犯してください。

Now that your organization has figured out how to make those first small contributions, you’ll need to build on those contributions to begin making larger contributions and having a bigger impact in the project.

あなたの組織は、最初の小さなコントリビューションでコントリビューションする方法を学んだので、あなたは、より大きなコントリビューションをして、プロジェクトにより大きな影響を与えるコントリビューションができるようになります。

・Build relationships at events. Relationships on a personal and organizational level are an important aspect of participating in an open source community. One of the best ways to build lasting relationships with other project members is by attending events. There is nothing quite like meeting someone in person to help understand them as a human being on the other side of their email address or online handle. These events have a varied mix of people from project leaders and passionate users of the product to direct participation from many of the organizations through sponsorships, booths, and demos to show how the organization contributes. Most of these events would not be possible without financial support from sponsoring organizations that allow us to get together and learn from each other while helping to achieve the goals of the project.

・**イベントで良い関係を構築してください。** 個人レベル、組織レベルで構築される人間関係は、オープンソース　コミュニティに参加する上で重要な側面を持っています。 他のプロジェクトメンバーとの継続的な人間関係を構築する最善の方法のひとつは、イベントに出席することです。 メールアドレスやオンラインのハンドル名でのやり取りなどと比べて、人間としてかれらを理解するためには直接会うことがやはり一番です。 これらのイベントには、プロジェクトリーダーや製品の情熱的なユーザーから、組織がいかにコントリビューションしているかを示すためにスポンサーシップ、ブース、デモを行っている多くの組織から参加している人まで、多様な構成の人々が集まっています。これらのイベントのほとんどは、プロジェクトの目標を達成するのを助けながら、ともに集まり、たがいに学ぶことを可能にしてくれるスポンサー組織からの財政的支援なしには成り立ちません。

・Include the community early and often. Some organizations make the mistake of developing big chunks of code in house and then dumping them into the open source project, which is almost never seen as a positive way to engage with the community. The reality is that open source projects can be complex, and what seems like an obvious change might have far reaching side effects in other parts of the project. Any significant change is likely to require some community discussion before it moves to implementation to make sure that there are no side effects and that the solution is aligned with the broader goals for the project. While you discuss it with the community, it can help to focus on the problem, rather than a specific solution, before you invest too much time in the creation of a body of code. (See Jon Corbet’s guide on How to Participate in the Linux Kernel Community)

**・早めに、高頻度でコミュニティを巻き込んでください。** 一部の組織では、大量のコードを社内で開発してから、オープンソース　プロジェクトに投入するという間違いを犯しています。このやり方は、コミュニティに参加するための肯定的な方法ではありません。 実際、オープンソース　プロジェクトは複雑なもので、明快な変更のように見える修正も、プロジェクトの他の部分で、大きな副作用を引き起こすことがあります。 重要な変更は、副作用がないこと、そしてソリューションがプロジェクトの幅広い目標に沿っていることを確認するために、実装する前にコミュニティで議論する必要があるでしょう。 コードの作成に時間をかけすぎる前に、その問題をコミュニティと議論することにより、特定の解決策ではなく、問題に焦点を当てて議論することができます。 （Linuxカーネルコミュニティへの参加方法に関するJon Corbetのガイド: [How to Participate in the Linux Kernel Community](https://www.linux.com/publications/how-participate-linux-community)を参照してください。）

・Contribute upstream. This refers to the practice of sending any changes you make to an open source project back to the original maintainers for inclusion into an upcoming release of the software. If your organization is new to open source, you may need to spend some time educating your employees about the importance of upstreaming contributions. In some cases, people may think it will be easier to do a quick and dirty patch to get something working in your infrastructure and not bother with cleaning it up and going through the process of getting it accepted into the upstream project.

**・アップストリームにコントリビューションしてください。** これは、オープンソース　プロジェクトに加えた修正変更をプロジェクトのメインテナーに送って、今後のソフトウェアリリースにその修正変更を含めるようにすることを意味しています。 あなたの組織がオープンソースにまだ不慣れな場合は、従業員に、アップストリームへのコントリビューションの重要性を教えるために、しばらく時間を費やす必要があります。 場合によっては、あなたのインフラストラクチャで使用しているソフトを動作させるために、素早く、汚れたパッチ（dirty patch）を作成する方が簡単で、アップストリームのプロジェクトに受け入れられるようなプロセスを通して、パッチをきれいにする（clean）手間を省きたいと考える人もいるかもしれません。

However, over the long-term, the quick patch that needs to be tested, updated and reapplied during every upgrade cycle is almost always going to take more time and effort than upstreaming it. This behavior can also be perceived as selfish within the community, since others might also benefit from your fixes, so it could also harm your organization’s reputation in open source communities.

けれども、長期的には、アップグレードサイクルごとにテスト、更新、再適用が必要なパッチの保守は、ほとんどの場合、アップストリームで保守されるパッチより時間も労力も必要とします。他の人もあなたの修正から恩恵を受ける可能性があるため、このような行動はコミュニティ内で、利己的であると認識されることもあり、組織のオープンソース　コミュニティでの評判にも悪影響を与えることがあります。

Best Practices to Contribute Code Upstream

コードをアップストリームにコントリビューションするためのベストプラクティス

Ibrahim Haddad, PhD @IbrahimAtLinux

・Internally to your organization

Decide to upstream for the right reasons.

Design and implement code with upstreaming in mind.

Adopt an “upstream first” policy. Submit patches upstream first, and consume in your own products downstream.

Keep your developers involved in the open source project, even if it was just a soft involvement.

あなたの組織内部で推進すること

・正当な理由で、アップストリームにコントリビューションしてください。

・アップストリームを念頭に置いてコードを設計し、実装してください。

・「上流第一（upstream first）」ポリシーを採用し、まずパッチをアップストリームに提出し、自社製品ではダウンストリームのものを採用してください。

・たとえそれは大きな関与でなくても、あなたの開発者をオープンソース　プロジェクトに関わり続けるようにさせてください。

Externally toward the project

Ensure that your contributions are useful to others.

Follow proper coding style.

Work within the submission processes of the project.

Provide documentation and explanation around your contributions.

Listen to feedback and act upon it.

Be patient and continue to rework the code until acceptance.

外部との関わりで推進すること

・あなたのコントリビューションが他の人にも役立つことを確認してください。

・適切なコーディングスタイルに従ってください。

・プロジェクトの提案プロセス（submission process）に従って作業してください。

・あなたのコントリビューションについての説明やドキュメントを提供してください。

・フィードバックに耳を傾けて、それに対応してください。

・辛抱強く、受け入れられるまでコードの改良、修正を続けてください。

One of the most challenging things for organizations is understanding how influence is earned within open source projects. Just because your organization is a big deal, doesn’t mean that you should expect to be treated like one without earning the respect of the open source community. Influence comes from participation, and some of the people contributing to an open source project will eventually earn positions of greater influence and leadership over time after they prove that they are reliable and responsible.

組織にとって最も挑戦的なもののひとつは、オープンソース　プロジェクトにどの程度の影響力を与えているかを知ることです。 あなたの組織が企業で大変重要なものだからといって、オープンソース　コミュニティから尊敬されていなければ、そのように扱われることを期待することはできません。 影響力はプロジェクトへの参加から始まります。そして、オープンソース　プロジェクトにコントリビューションする人の中で、信頼され、責任感が認められた人が、最終的に大きな影響力とリーダーシップを発揮する立場になります。

You should also expect some conflict and be ready to handle it professionally. The review process can get quite heated as people disagree with decisions, approaches or styles of contributions. It’s important to remain calm and professional while making sure that the feedback stays focused on the contribution rather than becoming personal. Keep in mind that your participation in an open source project is public and could remain on the internet forever, and one heated discussion that got out of hand could come back to haunt you as an organization or an individual at a later date. Because all of this participation is very public, offering some training about handling difficult people and resolving conflict for your employees might be a good idea.

また、対立がおこるでしょうが、プロフェッショナルにそれを処理できるようになっている必要があります。 決定、アプローチ、またはコントリビューション　スタイルで同意が得られず、レビュープロセスはかなりが白熱したものになることもあります。 フィードバックが心情的なものになるのではなく、コントリビューションに焦点を当てて、静かでプロフェッショナルに対応することが重要です。 オープンソース　プロジェクトに参加する行為はパブリックになっており、それがインターネット上に永遠に残っている可能性があることを忘れないでください。おさえの効かなくなった白熱した議論は、後々にも組織、個人が行った行為としてあなたについて回ることもあります。 参加行為はすべて、パブリックになるので、「扱いづらい、難しい人に対する対応方法」や「対立の解決方法」、に関するトレーニングを従業員に提供することは良いアイデアかもしれません。

Section 6

セクション　6

How to create your open source contribution strategy

どのようにして、オープンソース　コントリビューション戦略を構築するか

Having a deliberate and thoughtful open source contribution strategy not only helps guide your employees when participating in open source projects, but it can also help justify this participation to senior management within your organization. It’s important to start by looking at the overall business goals for the organization to figure out how your open source efforts fit into the broader strategies for your organization. (See our guide on Creating an Open Source Business Strategy.) By clearly tying your open source contribution strategy to the organization’s strategic efforts, you can show senior management why the work is important for the organization and help your employees understand the impact of their contributions.

十分に、慎重に検討されたオープンソース　コントリビューション戦略を持つことは、オープンソース　プロジェクトに参加する際に従業員の指導に役立つだけでなく、組織内の上級管理職への参加の正当性を示すのにも役立ちます。 まず、組織の全体的なビジネス目標を見て、オープンソースの取り組みが組織の幅広い戦略にどのように適合しているかを把握することが重要です。 （私たちの提供している「オープンソース　ビジネス戦略の作成」、Creating an Open Source Business Strategyを参照ください。）オープンソース　コントリビューション戦略を組織の戦略的な取り組みと明確に結びつけることで、上級管理職に、なぜコントリビューションがその組織にとって重要なのかを示し、従業員がコントリビューションの与える影響力を理解することができるようになります。

“Support from leadership and acknowledgement that open source is a business critical part of your strategy is so important. You should really understand the company’s objectives and how to enable them in your open source strategy.”

「オープンソースはビジネス戦略の重要な部分であるという認識、経営幹部からのサポートは極めて重要なものです。 あなたは、企業の目標と、オープンソース戦略の中で、それらをどのように達成していくかについて十分に理解している必要があります。」

[Nithya Ruff](https://twitter.com/nithyaruff) – Senior Director, Open Source Practice at Comcast

Once you’ve developed some goals and strategies that are aligned with the business goals, you’ll need to develop an implementation plan. These questions will help you think about some of the things that might need to be addressed in your plan:

ビジネス目標に沿った目標や戦略を立てたら、それをインプリメントするための計画を立てる必要があります。 以下の質問は、あなたの計画に取り入れることを考えなければならない、いくつかの項目を示唆するでしょう。

・Why are these contributions important? It’s easy to jump in and start talking about all of the great things you plan to do, so don’t forget to include compelling arguments for why this work is important and strategic to the organization.

・**なぜ、これらのコントリビューションは重要ですか。**あなたが計画しているすべての素晴らしいことについて話を始めることは難しくはありません。なぜこの仕事が組織にとって重要で、戦略的なのかについて説得力のある議論をすることを忘れないでください。

・What open source projects do we use within the organization? Take some time to assess which open source projects are already in use across the entire organization to determine which ones are strategic to your business. A few places you might want to focus your assessment: critical business infrastructure (operations), development and deployment tools that impact your ability to release products, and software that is important for customer-facing products or services.

**・どのオープンソース　プロジェクトを組織内で使用していますか。** どのオープンソース　プロジェクトがあなたのビジネスにとって戦略的であるかを判断するために、使用されているオープンソース　プロジェクトを組織全体にわたり評価するのにしばらく、時間をかけてください。 重要なビジネスインフラストラクチャで運用、製品のリリースに強い影響を与える開発ツール、配布ツール、および顧客に直接関係する製品、サービスに重要なソフトウェアなど、焦点を当てて評価しなければならないところがいくつかあるでしょう。

・What projects should we target for contribution? Most organizations use many open source projects, so it’s important to make sure that your plan focuses on just the most important ones. Just because a project isn’t on the target list doesn’t mean that people can’t contribute to that project, it just means that it isn’t a critical focus for your organization. If an open source project is critical to your business and has the potential to cause significant downtime or disruption to your ability to serve your customers, it’s probably a good candidate for contributions.

・**どのようなプロジェクトにコントリビューションすべきでしょうか。** ほとんどの組織は多くのオープンソース　プロジェクトを活用しているので、あなたの計画は最も重要なものに焦点を当てていることを確認する必要があります。 プロジェクトがターゲットリストに載っていないという理由だけで、人々がそのプロジェクトにコントリビューターできないということではなく、そのプロジェクトはあなたの組織にとって重要で、焦点を当てたプロジェクトではないことを意味するだけです。 オープンソース　プロジェクトが企業のビジネスにとって不可欠で、問題が発生すると、深刻なダウンタイムや顧客へのサービス提供の妨げをもたらす可能性がある場合、そのプロジェクトはコントリビューションの候補として適しているでしょう。

・What contributions are we already making? In some cases, you might already have people making changes to open source projects. They may be creating patches that are used internally, or they could already be contributing those patches back to the upstream project to avoid maintaining them. Spend some time talking to your internal teams to find potential contributions that you can build on while assessing whether or not you already have people on staff who might have the skills and interest to contribute.

**・どのようなコントリビューションをすでに実施していますか。** 場合によっては、すでにオープンソース　プロジェクトの修正、変更をしている人がいるかもしれません。 内部で使用されるパッチを作成している場合や、パッチの保守を避けるためにアップストリーム　プロジェクトにそれらを戻すことによってコントリビューションしている場合があります。 社内のチームと話をするための時間をとり、コントリビューションできるスキルや興味を持っているスタッフをすでに抱えているかどうかを評価しながら、あなたが将来できそうなコントリビューションを見つけてください。

・Do we already have the relevant expertise, or would we need to hire for it? As discussed previously in this guide, it’s important to find people who have both the skills to create the contribution along with the people skills to work with the community to have the contribution accepted. If you already have people contributing to some of these projects, you might be able to use existing staff. If not, you should consider hiring someone who is already making successful contributions to the project. As with any plan, you need to make sure that you have the resources and hiring budget required for it to be a success.

**・私たちはすでに関連する専門知識を持っていますか、あるいはそれを得るために人を雇う必要がありますか。**このガイドですでに説明したように、コントリビューションを作成するスキルと、コミュニティと協調してコントリビューションが受け入れられるようにするスキルの両方を持つ人を見つけることが重要です。 すでに、プロジェクトにコントリビューションしている人があなたの企業にいる場合、いまいるスタッフを活用することができます。 そうでない場合は、すでにそのプロジェクトにコントリビューションすることで地位を得ている人を雇うことを検討すべきです。 どんな計画でも成功のためには、それに必要なリソースと採用予算を確保しておく必要があります。

・What funding do we need for project sponsorships / corporate memberships? Look at the governance models for the projects you’ve selected to determine whether there is an option for corporate membership or sponsorship for the project or the foundation responsible for it. This provides funding to help the project be successful, and in some cases, it can help your organization get more involved in an advisory role or provide some influence into the project. Most open source projects have conferences, and in addition to funding for the project directly, you should also consider sponsoring the project’s conferences, which can be a great way to get the word out about your work and are a place to meet people that you might want to recruit.

**・プロジェクト　スポンサーや企業会員になるためにはどのような資金が必要になりますか。**あなたが選んだプロジェクトのガバナンスモデルをチェックして、そのプロジェクト、またはそのプロジェクトに責任を持っているファウンデーションの企業会員やスポンサーシップになることができるかを確認します。 これはプロジェクトが成功するための資金を提供し、場合によっては、組織がプロジェクトに助言を与えたり、影響力を与えたりするために役立ちます。 ほとんどのオープンソース　プロジェクトはカンファレンスを実施します。プロジェクトに対する直接的な資金提供に加えて、プロジェクトのカンファレンスのスポンサーになること考慮してください。カンファレンスはあなたの仕事についての意見を聞くことができる素晴らしい機会であり、また、あなたが採用したい人に出会える場所でもあります。

・How should we promote our open source efforts? Depending on your organization, marketing or promoting your open source contributions can be tricky, which is why it’s important to include this in your implementation plan to make sure that everyone knows how you plan to talk about your contributions publicly. Sponsorships and giving talks at the project’s conference or other events can be a good way to promote the work that you are doing and recruit contributors. In particular, don’t overlook your participation in local user groups where you have employees. Sponsoring those local groups and sending contributors to give talks can be a great way to recruit local people who are passionate about particular open source projects.

**・オープンソースに対する取り組みをどのようにプロモーションすべきでしょうか。**組織によっては、オープンソースへのコントリビューションをマーケティング、プロモーションすることが難しい場合があります。したがって、あなたのコントリビューションがどのように公開されていくのかについて、組織のすべての人が知ることができるようにすることを、計画の中に含めることは重要なことです。 プロジェクトのコンフェランスやその他のイベントでスポンサーになったり、講演したりすることは、あなたがやっている仕事のプロモーション、コントリビューターの採用のための良い方法です。 特に、従業員がいる地域のユーザーグループへの参加を見逃さないでください。 地域のグループのスポンサーになり、コントリビューターを派遣し、講演することは、特定のオープンソース　プロジェクトに情熱を持っているその地域の人を採用募集するのに非常に良い方法です。

・What contribution guidelines or processes do we need? These guidelines and processes should be less about rules and regulations and more about helping people be successful in making contributions to open source projects. It can help if people have guidelines and checklists to make sure that they have everything they need to make a successful contribution without running into licensing or confidentiality issues. Especially for new contributors, it can also help to have an internal review process available as a safe place to get feedback before making a contribution.

**・どのようなコントリビューション　ガイドラインやプロセスが必要でしょうか。**これらのガイドラインとプロセスは、ルールや規制に関連する部分はなるべく減らして、オープンソース　プロジェクトへのコントリビューションを成功させるために役立つ部分をより多くすべきです。 ガイドラインとチェックリストが、ライセンスや秘密保持の問題に深入りせずに、コントリビューションを成功させるために必要なものがすべて含まれているようなものであれば非常に有益なものになります。 特に、新しいコントリビューターにとっては、コミュニティにコントリビューションを行う前、コントリビューションのフィードバックをパブリックな批判で受ける前に、内部レビュープロセスを利用できれば非常に助かります。

・What training do we need to provide? It can help people to have training on best practices for making contributions along with some general open source training about licenses, governance, and norms associated with participation in open source communities. Training on conflict resolution, dealing with difficult people, and other people skills can also help avoid issues later. Your training plan should also include programs that provide experienced open source contributors as mentors for new contributors as a way to scale your efforts over time.

**・どのようなトレーンイングを提供すれよいのでしょうか。** オープンソース　コミュニティへの参加に関連したライセンス、ガバナンス、行動規範に関する一般的なオープンソース　トレーニングに加えて、コントリビューションするためのベストプラクティスのトレーニングを人々に提供すべきです。 対立の解消方法、扱いの難しい人への対処方法、および対人関係に関するスキルに対するトレーンイングは、後に、問題を回避するために役立つでしょう。トレーニングプランには、経験豊富なオープンソース　コントリビューターが新しいコントリビューターのメンターとして指導するトレーニング　プログラムを含めるべきです。これは、長期的にあなたの成果を増大させます。

・How will we determine whether the plan is successful? Every plan should have success criteria and a plan for measuring whether you achieved your goals. This should come directly out of your strategies to make sure that you are tracking those activities that are the most important for your organization, rather than the ones that are the easiest to measure. The open source GrimoireLab project is a good place to start if you need measurement and metrics tools. (See our guide on Measuring Your Open Source Program Success.)

**・計画が成功したかどうかはどのように判断するのでしょうか。**すべての計画には達成基準と目標を達成したかどうかを評価するステップが含まれていなければなりません。 これは、組織にとって最も重要な活動をトラッキングするということであり、あなたの戦略から直接導かれたものでなければなりません。測定するのが最も簡単なものをトラッキングするのではありません。オープンソースの[GrimoireLab](https://chaoss.github.io/grimoirelab/)（ハイパーリンクの間違いを修正しています）プロジェクトは、測定ツールやメトリクスツールが必要な場合に最初に訪問すべきサイトです。 （提供しているガイド、「オープンソース　プログラムが成功しているのかを評価する」: [Measuring Your Open Source Program Success](https://www.linuxfoundation.org/resources/open-source-guides/measuring-your-open-source-program-success/)を参照してください）。

・Do we need an open source office to manage all of these efforts? All of this is a lot to think about, and it can help to have an open source program office or dedicated staff who are responsible for implementing the plan. At a minimum, you’ll want to have someone responsible putting processes and training in place, providing licensing guidance, answering questions from senior management or contributors, and communicating updates throughout the organization. (See our guide on Creating an Open Source Program Office.)

**・これらすべての取り組みを管理するためにオープンソース　オフィスが必要でしょうか。**すべての取り組みは、考えなければならないことがたくさんあるものです。オープンソース　プログラムオフィス、または計画の実行を担当する専任のスタッフを持つことが必要です。少なくとも、トレーニングを実施しプロセスを導入し、ライセンスガイダンスを提供し、上級管理者やコントリビューターからの質問に応え、組織全体に最新状況を報告することなどに責任も持つ人が必要でしょう。 （「オープンソース　プログラムの作成」: [Creating an Open Source Program](https://www.linuxfoundation.org/resources/open-source-guides/creating-an-open-source-program/)（明らかに原文、ハイパーリンクなどが間違っているので修正しています。）に関するガイドを参照してください）。

11 Tips for Mastering Open Source Contributions

オープンソース　コントリビューションを習得するための11項目の秘訣

How to build a healthy environment for open source contributions in your organization

オープンソース　コントリビューションを推進できるための健全な環境を持つ組織をどのようにして構築するか。

・Establish a policy and process to guide open source contributions

・オープンソース　コントリビューターのためのポリシーとプロセスの確立

・Set up a team to oversee approvals for all open source contributions

・すべてのオープンソース　コントリビューションを承認、監督するチームの設置

・Focus contributions in the areas that will enable your technologies

・あなたのテクノロジーが活かせる領域に焦点を当てる

・Provide the needed IT infrastructure and tooling for contributors

・コントリビューターに必要なITインフラストラクチャとツールを提供

・Offer training to your staff on contribution best practices

・コントリビューションのためのベストプラクティスのトレーンイングをスタッフに提供

・Track contributions, measure impact, improve, and communicate

・コントリビューションをトラッキングし、効果を評価し、改善などのためにコントリビューターとコミュニケーションをとる

・Establish a mentorship program to train less experienced developers

・経験の少ない開発者をトレーンイングするためのメンターシップ、指導プログラムの設立

・Provide contribution guidelines, how-tos, do’s and don’ts

・コントリビューション　ガイドライン（どのようにすれば良いか、しても良いこと、してはいけないことなどを示す）の提供

・Make open source legal support accessible to developers

・開発者がオープンソース関連で法務サポートを受けられるようにする

・Hire from the open source communities you value the most

・あなたにとって最も価値のあるオープンソース　コミュニティからの雇用

・Always follow the community processes/ practices specific to each project

・各プロジェクト　コミュニティの持っている固有のプロセスやプラクティスに常に従う

Ibrahim Haddad, PhD @IbrahimAtLinux

Section 7

セクション　7

Final Words

結論

Open source projects are here to stay, and they play a critical role in the ability for most organizations to deliver products and services to customers. As an organization, if you want to influence the open source projects that drive the success of your business, you need to participate. Having a solid contribution strategy and implementation plan for your organization puts you on the path towards being a good corporate open source citizen. Good luck!

オープンソース　プロジェクトは社会に浸透しており、ほとんどの組織が製品やサービスを顧客に提供するために使用していて、重要な役割をはたしています。組織として、ビジネスの成功を促進させるオープンソース　プロジェクトに影響を与えたい場合は、オープンソース　プロジェクトに参加する必要があります。 あなたの組織のための確固としたコントリビューション戦略、実施計画を持つことが、良きオープンソース企業市民になる道につながります。 がんばりましょう。

These resources were created in partnership with the TODO (Talk Openly, Develop Openly) Group – the professional open source networking group at The Linux Foundation. A special thanks goes out to the open source program managers who contributed their time and knowledge to making these comprehensive guides. Participating companies include Autodesk, Comcast, Dropbox, Facebook, Google, Intel, Microsoft, Netflix, Oath (Yahoo + AOL), Red Hat, Salesforce, Samsung and VMware. To learn more, visit: todogroup.org

これらのリソースは、TODO (Talk Openly, Develop Openly) グループとの協力により作成されました。TODOグループは、The Linux Foundation傘下のプロフェッショナル オープン ソース プログラム ネットワーキング グループです。 これらの総合ガイドの作成に時間と知識を費やして頂いたオープンソースのプログラムマネージャーに大変感謝致します。 参加企業は、Autodesk、Comcast、Dropbox、Facebook、Google、Intel、Microsoft、Netflix、Oath（Yahoo + AOL）、Red Hat、Salesforce、Samsung、VMwareなどです。 詳しくは、todogroup.orgをご覧ください。

This work is licensed under a Creative Commons Attribution ShareAlike 4.0 International License.

この資料は、Creative Commons Attribution ShareAlike 4.0 International License (CC BY-SA 4.0：クリエイティブ・コモンズ 表示 - 継承 4.0 国際ライセンス) の下でライセンスされています。

Sign up to get updates! Be the first to know when we add more open source guides and other content like this.

最新情報を受け取りましょう！オープン ソース ガイド シリーズなどのコンテンツが追加されるとお知らせします。通知ご希望のかたはこちらからお申し込みください。