Tools for Managing Open Source Programs

オープンソース プログラムを管理するためのツール

The road to strategic use of open source starts with a carefully planned, organized, and empowered open source program office to guide and manage its creation, distribution, and use. But that’s just a first step. To get such an office underway and running smoothly, you need the right tools. These mission-critical tools will be used to track goals and metrics in departments from engineering and legal to executive leadership, PR, and marketing, and give employees all the resources they need to gather data, provide snapshots of performance, and manage the daily use of open source within your company.

This guide provides details and scenarios for how to get your open source tool collection started, including information about the most important tools to use to track and manage your open source projects. Many of the tools have been created and open-sourced by The Linux Foundation and other leaders in the field, providing free and easy access for your projects. You’ll also find an example dashboard setup which combines and displays information from multiple tools.

オープンソースの戦略的な活用への道は、オープンソースの作成、配布、利用に対する指導と管理のために慎重に計画され、組織化され、権限移譲されたオープンソース プログラム オフィスから始まります。しかし、それは単に、最初の一歩に過ぎません。オープンソース プログラム オフィスを軌道に乗せて、円滑な運用を進めるためには、適切なツールが必要です。同オフィスの活動に必要不可欠なツールは、エンジニアリング部門と法務部門から始まり、経営幹部、広報部門、マーケティング部門に至る広い部門で、目標や評価基準の達成度をトラッキングするために使用され、また、データの収集、プロジェクト達成状況のスナップショットの提供、さらには、会社内の日々のオープンソース使用状況の管理のために必要なあらゆるリソースを従業員に与えるために使用されます。

このガイドは、オープンソース プロジェクトのトラッキングと管理に使用する最も重要なツールに関する情報も含め、オープンソース ツールの収集を始める方法の詳細と手順を提供しています。これらのツールの多くは、Linux Foundationやこの分野のリーダーが作成し、オープンソース化したものなので、あなたのプロジェクトで自由に、簡単に使用することができるでしょう。また、複数のツールの情報を組み合わせて表示するダッシュボードの設定例も提示しています。

Contents

Why you need special tools

How to select and plan your tools

Elements of a basic toolset

Tools for managing source code

Tools for tracking project health

Tools for communications and collaboration

Tools for corporate scale GitHub management

Final words

内容

オープンソース プログラムの管理のためになぜ特別なツールが必要になるのか

ツールをどのように選択し、どのように使用するかの検討

基本的なツールセットの構成要素

ソースコード管理のためのツール

プロジェクトの健全性をトラッキングするツール

コミュニケーションとコラボレーションのためのツール

企業規模のGitHub管理のためのツール

結論

View All Guides »

ガイド一覧

Contribute on GitHub »

GitHub上で貢献する

Contributors to this Guide

このガイドの貢献者

Chris Aniszczyk

Chris Aniszczyk

COO

Cloud Native Computing Foundation

Jeff McAffer

Jeff McAffer

Director of the Open Source Programs Office at Microsoft

Section 1

セクション1

Why you need special tools for open source program management

オープンソース プログラムの管理のためになぜ特別なツールが必要になるのか

Once your open source program office is up and running, it’s time to collect the right software tools that will allow your development teams to manage, track, guide, and advance their open source projects, contributions, and releases.

オープンソース プログラム オフィスが設立されたら、まず、同オフィスの開発チームがオープンソースのプロジェクト、コントリビューション、リリースを推進、管理、トラッキング、指導することができるようにするために、適切なソフトウェア ツールの収集を始めます。

・Provide a workplace for collaboration and code building

・コラボレーションとコード開発の場の提供

・Manage project health

・プロジェクトの健全性の管理

・Automate critical and repeatable tasks such as code review and tracking and license compliance

・コードのレビューとトラッキング、ライセンス コンプライアンスのチェックなど重要で何度も繰り返されるタスクの自動化

・Generate data to prove ROI for your program office and open source strategy, in general

・プログラム オフィスとオープンソース戦略のROIが分かるようにするデータの作成

・Oversee project quality and to make sure that guard rails are in place if issues arise.

・プロジェクトの品質を監視、および問題が発生した場合にガードレールが設置されていることの確認

Having the right targeted tools as you begin your open source path will also make jobs easier for developers and other employees, will provide better results, and will become the basis for successful collaboration and communications for a company’s open source projects.

あなたがオープンソースを始めるとき、適切に的を絞ったツールを用意することで、開発者や他の従業員の仕事はより簡単になり、より良い成果が得られるようになります。それらのツールは企業のオープンソース プロジェクトにおいて、コラボレーションとコミュニケーションを成功させるための基礎になるでしょう。

“If you have more than 100 code repositories or 100 people that you’re trying to manage, you really can’t have someone doing it manually with spreadsheets anymore. Obviously, people still do it that way. But it starts to become ad hoc and laborious. That’s where tools come into play. They allow you to scale.”

「もし、あなたが100個以上のコード リポジトリ、または100人以上をマネージメントしなければならない立場にあれば、スプレッドシートを使用した手作業でそれらを管理することはできません。まだ、そのようにしている人はいます。 しかし、そのやり方は、いずれその場しのぎの面倒なものになっていきます。 そここそ、ツールが活躍する場です。それらのツールはあなたのマネージメントがより大きな規模になることを許してくれます。」

[Jeff McAffer](https://twitter.com/jeffmcaffer) – Director of the Open Source Programs Office at Microsoft

“At the end of the day, you need tools to automate your life, otherwise you’re going to waste a lot of time doing things manually.”

「結局、あなたの仕事を自動化するツールが必要になります。そうでなければ、あなたは手作業のために、多くの時間を費やさなければならなくなります。」

[Chris Aniszczyk](https://twitter.com/cra) – COO of the Cloud Native Computing Foundation and former head of open source programs at Twitter

Section 2

How to select and plan your tools

ツールをどのように選択し、どのように使用するかの検討

Most of the early discussions about which open source tools are needed by a company will depend on its business, products, and services and how it serves its customers and employees. As the planning process and strategy map are developed by its open source program office, tools can be chosen to integrate the company’s goals, processes and infrastructure.

企業がどのオープンソース ツールを必要としているかについての初期段階の議論の大部分は、企業のビジネス、製品、サービス、そして企業が顧客や従業員にどのように役立つのか、などによって異なったものになります。計画工程と戦略マップが、オープンソース プログラム オフィスによって作成されるとともに、会社の目標、プロセス、インフラストラクチャと調和のとれたツールが選択されます。

Ultimately, the only way to know which tools you will need is to understand what you want to do with open source.

Below are the basic steps for choosing the tools you’ll need for managing your open source program office:

最終的に、必要なツールを知る唯一の方法は、オープンソースであなたの会社が何をしたいのかを理解することです。オープンソース プログラム オフィスの管理に必要なツールを選択するための基本的な手順を以下に示します。

1． Get buy-in and selection preferences from developers and community members. To accomplish this, you’ll want to conduct detailed discussions with developers and community members. They can describe what tools have been or would work best for them. Take those recommendations and requests very seriously. Listen to the people who are going to get you to your goal. They have most likely been using many of these tools already, so benefit from their experiences.

**1．開発者とコミュニティ メンバーから同意と選択の好みを得ることです。** これを得るためには、開発者やコミュニティ メンバーと詳細な議論が必要です。彼らは、どのツールが彼らのために最善であったのか、最善でありそうかについて説明できます。 彼らの推薦や要求を真剣に受け止めてください。 あなたに目標を達成させようとしてくれている人に耳を傾けましょう。 彼らはすでにこれらのツールの多くをすでに使用している可能性が高く、彼らの経験は有益です。

2． Understand necessary software dependencies and integrations for business-critical applications. This means understanding and knowing which open source software your business depends on so you can stay up to date with security issues and ensure software continuity.

**2．ビジネス上重要なアプリケーションが依存している、また、それに統合されている必須ソフトウェアを理解しなければなりません。** これは、あなたのビジネスが依存しているオープンソース ソフトウェアを理解、把握することを意味します。それにより、セキュリティ問題の最新状況を常に把握でき、ソフトウェアの継続的利用を確実にすることができます。

3． Research existing tools and decide what you can use as-is, or build out to suit your needs. Don’t start from scratch for every tool. See what is out there and being used in the open source communities you are in and get advice and feedback about those tools. Linger in online development communities to see what works and ask for recommendations and advice. Ask questions at open source conferences, talk to fellow developers in Birds-of-a-Feather sessions, and learn from others who are already doing what you want to do.

**3．既存のツールを調査し、そのまま使用するのか、ニーズに合わせて改造するのかを決めます。** どのようなツールであってもゼロから開発してはいけません。 まずどんなものがあるか、どのようなものがオープンソース コミュニティで使用されているのかを調べて、彼らからツールに関するアドバイスやフィードバックを得てください。開発コミュニティにアクセスして、何がうまくいっているかを調べて、推奨、助言を求めてください。オープンソース カンファレンスで質問したり、BOFセッションで開発者仲間と話したりして、あなたがやりたかったことを既にやっている人から学びましょう。

Once selected, the tools must then be implemented, which requires several additional steps:

ツールを選択したら、それを実装しなければなりませんが、そこにはいくつかのステップが必要になります。

1． Create an internal infrastructure to support, manage, and use the tools. Through your newly-formed open source program office, designate someone to maintain and build the internal infrastructure that will distribute the tools through an online internal portal where they are kept and organized into tasks and features. In this tool portal you can make the tools available to all developers or restrict them to specific users through authentications and permissions based on their jobs and requirements.

**1．ツールをサポート、管理、使用するためのインフラストラクチャを社内に構築します。**新たに設立したオープンソース プログラム オフィスは、それらのインフラストラクチャを構築、保守する人を指名します。ツールは、その役割、機能などで整理した社内オンライン ポータルを通してそこから提供されます。このツール ポータルでは、ツールをすべての開発者が使用できるようにすることもできますが、認証とアクセス許可を使用して、彼らの業務内容、要件に基づいてアクセスを特定のユーザーに制限することもできます。

2． Provide training plans for employees who will use the tools. Just getting the tools isn’t enough. Now you have to be sure that your developers know how to use them and are mastering their capabilities. This is where training programs, whether online, in classrooms or in small lunchtime group settings, will be important to reap the benefits of their use. Ask your developers which learning methods work best for them and let them choose how they want to learn.

**2．ツールを使用する従業員のためにトレーニング計画を作成します。** ツールを入手するだけでは不十分です。開発者がツールの使い方を理解し、ツールの特性を習熟できるようにすることが必要です。トレーニング プログラムは、オンライン、教室、ランチタイム ミーティングなど、どこで実施するにしても、ツールの利用で成果を出すために重要です。どのような学習方法が彼らのために最も効果的か開発者に聞いて、学習方法を彼らに選択させてください。

3． Ensure the tools are centrally visible in your organization. Make it easy for developers to find and use them, preferably integrated into any existing developer dashboards that track development progress. Again, this is where the internal tool portal is going to help your company organize and distribute the critical tools for your operations.

**3．ツールが組織内のどこからでも目立つようにします。**できれば、開発の進捗状況を把握するためにいつも使っている開発者ダッシュボードに統合し、開発者が簡単にツールを見つけて使用できるようにします。ここでも、社内のツール ポータルが業務に必要なツール群を体系化し、配布するのに役立ちます。

Implementation is helpful to keep in mind as you are choosing your tools, as this may also affect your decision. A tool with a steep learning curve, for example, may require more training.

あなたが自ら選択したツールの実装を記憶に留めることが重要です。ツールの選択が次の決定に影響するかもしれないからです。 たとえば、習得が容易なツールが、より多くのトレーニングを必要とすることもありえます。

Leverage existing tools

既存のツールを生かそう

After you have a good idea of what your team needs to meet your organization’s open source goals and the possible limitations of your own dependencies and infrastructure, the first step is to explore and learn about existing tools that are ready-built and available for you today. Since most are open source tools themselves, if they don’t meet your exact needs at the start, your development teams can contact the builders of the tools to see if they can collaborate and contribute to take the tools in new directions by adding features.

組織のオープンソースの目標を達成するためにチームが必要としていることと、あなたの組織の権限範囲、インフラストラクチャでの制約を理解したら、次のステップは、今すぐに利用可能な既存のツールを調査し、学ぶことです。ほとんどのツールはオープンソース ツールなので、もしもそれらが、正確にあなたのニーズを満たしていない場合でも、開発チームがツール作成者に連絡して、機能を追加し、新しい用途向けにツールを拡張するためにコントリビューションが可能かどうかを確認できます。

Ironically, many open source program offices don’t always reuse the tools developed by others, or collaborate with other companies to work on the tools they require to manage their open source programs. Often, they want to do that, but many businesses, including Facebook and Microsoft, already have existing tool suites which were in place before collaboration really became a discussion topic. Because they already have their tool sets and have made those investments, they seem to have less desire to adopt those of other companies.

皮肉なことに、多くのオープンソース プログラム オフィスは、他の人が開発したツールを再利用したり、オープンソース プログラムを管理するために必要なツールを他の企業と協業したりするとは限りません。彼らはそうしたいと思うこともありますが、Facebook社やMicrosoft社など多くの企業では、コラボレーションが実際に問題提起されるよりも前にツールセットが既に揃っており、それに投資を行ってきたため、他の企業のものを採用する意欲が低いようです。

That’s where companies that are just starting to build out their own open source programs have a significant advantage. Since they are now establishing their own open source program offices and diving into open source, they don’t have to be bothered with such limitations.

そこにオープンソース プログラムを構築し始めたばかりの企業の大きな利点があります。彼らは、オープンソース プログラム オフィスを開設しオープンソースに飛び込んだばかりなので、そのような制約に悩まされる必要はありません。

Instead, they can wisely take advantage of the experiences and successes of others and build their open source toolboxes using the proven tools created by companies which led the way in recent years. The Linux Foundation’s open source industry organization, the TODO Group, has been working on assembling a tool-filled “Open Source Program Office in A Box” starter package, which would give companies the ability to launch their open source efforts with a cohesive, pre-assembled kit of tools. The starter package isn’t yet ready, but the hope is that eventually it could make it easier for a company to deploy and configure the tools they need with less initial effort. Some of the members of the TODO Group working on this project include Adobe, Capital One, Comcast, Facebook, Google, eBay, IBM, Microsoft, Samsung, and Twitter.

代わりに、彼らは他の人の経験や成功を要領よく利用して、すなわち、近年オープンソースを主導している企業によって作成された実績のあるツールを使ってオープンソースのツールボックスを構築することができるからです。 Linux Foundationのオープンソース業界組織[TODO Group](http://todogroup.org/blog/sample-job-req/)は、ツールを満載した「オープンソース プログラム オフィス インボックス」スターター パッケージを作成することに取り組んでいます。このパッケージにより、多くの企業は、ツールがまとまって組み込まれているツールキットを入手したうえでオープンソース活動を開始することができるようになります。 スターター パッケージはまだ提供ができていませんが、最終的には、このパッケージにより、初期作業を軽減しながら必要なツールを導入して設定することを容易にすることが期待されます。 このプロジェクトに取り組んでいるTODOグループのメンバーは、Adobe社、Capital One社、Comcast社、Facebook社、Google社、eBay社、IBM社、Microsoft社、Samsung社、Twitter社などです。

Create a dashboard

ダッシュボードの作成

Along with the proper tools, companies should also incorporate central dashboards which allow them to monitor and track their open source projects and development in real time. Many companies likely have such dashboards for existing development work and applications and may be able to integrate the existing dashboards with their open source work. If not, they should create or adopt new dashboards to improve the management of their open source deployments.

適切なツールと共に、企業は、オープンソース プロジェクトと開発をリアルタイムで監督、トラッキングできるセントラル ダッシュボードも組み込む必要があります。多くの企業は、既存の開発作業やアプリケーション用のダッシュボードを持っており、既存のダッシュボードをオープンソースの作業と統合することができるかもしれません。そうでない場合は、オープンソース配置・展開の管理を強化するために新しいダッシュボードを作成または採用する必要があります。

Section 3

セクション 3

Elements of a basic toolset

基本的なツールセットの構成要素

The abundance of tools available for managing and reporting on open source projects can quickly become overwhelming. If your open source program is just getting started, it helps to focus your research on just a few of the basic tools that you’ll need to get up and running.

オープンソース プロジェクトの管理とレポートに利用できるツールは、極めて豊富ですが、過剰とも言えます。オープンソース プログラムを始めたばかりであれば、プログラムの立ち上げと実行に必要ないくつかの基本的なツールに焦点を当てて調査を始めるのが良いでしょう。

Then as your program grows and you’ve gained more experience using these tools, you can start to adopt new tools to help you automate and streamline your processes as the need arises. Remember that you want the tools you choose to complement and support your internal culture and processes — not lead them.

あなたの会社のオープンソース プログラムが成長し、これらのツールを使用してより多くの経験を積んだ後に、必要に応じてプロセスを自動化、合理化するための新しいツールを採用するのが良いでしょう。企業の文化、プロセスを補完、サポートするツールを選択することを忘れないでください。文化、プロセスを牽引するツールを選ぶのではありません。

The sections below give the basic categories of tools that pretty much all open source programs use on a daily basis. This is a good way to organize your research.

あなたの体系的な調査に便利なように、以下の各セクションでは、ほとんどすべての企業のオープンソース プログラムが日常的に使用するツールを基本的なカテゴリに分類しています。

Automate processes

プロセスの自動化

Tools which automate processes are among the most important you will select and use for your company’s open source program. The tasks for such tools are broad, including automating procedures for contributor license agreements (CLAs), which are legal documents stating that a developer created the code and didn’t copy it from anywhere else illegally. Traditionally these kinds of agreements were done manually by printing out the agreements and then signing and faxing them in to comply. But in a world of email and instant communications, that’s crazy today. Instead, the process can be automated using bots that request electronic signatures and then track and handle the submissions.

プロセスを自動化するツールは、あなたの会社のオープンソース プログラムで使用する最も重要な部類に属します。これらのツールは幅広い機能を持っています。たとえば、開発者自身がコードを作成し、違法な場所からコピーしていないことを示す法的文書であるコントリビューター ライセンス契約 （CLAs） の手順を自動化するツールがあります。従来、この種の契約は、契約書を印刷し、それに署名してファックスで送信するといった手動ベースで行われてきました。しかし、今日の電子メールとインスタント コミュニケーションの世界では、その方法はありえません。代わりに、ボットが電子署名を要求し、それ以後、常に提出物をトラッキングし、処理することで、そのプロセスを自動化することができます。

Other automation tools can tell you who exactly is contributing to your projects and can help remove procedural friction which slows down progress in projects as they get larger and scale to meet the needs of companies.

他の自動化ツールは、プロジェクトに誰が貢献をしているのかを正確に知らせてくれ、企業のニーズを満たすためにプロジェクトが大きく拡大していく中で、プロジェクトの効率を下げている手続き上のオーバーヘッドを取り除くのを助けます。

In Microsoft’s open source program office, where some 8,000 repositories are managed on GitHub involving some 11,000 contributors, about 40,000 internal requests came in to use open source in projects in 2016, according to the company. To manage those requests as well as the code that’s created and the code versions which are being updated, the company turns to tools which can automate the chaos. And because the code is likely being used in potentially hundreds of other projects, it must be tracked carefully so that if a security bug arises all software impacts can quickly be mapped out and fixed. At such a large scale, automation is critical and manual updates would be almost impossible.

Microsoft社によると、約11,000人のコントリビューターが参加し、約8,000個のリポジトリをGitHubで管理している同社のオープンソース プログラム オフィスは、2016年には、プロジェクトでオープンソースを使用したいというリクエストを約40,000件受け取りました。これらのリクエストだけでなく、作成されたコード、更新されたコードのバージョンを管理するために、同社はこの混乱を自動化で解決するツールを使い始めました。また、コードは潜在的に何百もの他のプロジェクトで使用されている可能性があるため、当該コードは細心の注意をもってトラックされなければなりません。セキュリティ上のバグが発生した場合、関連するすべてのソフトウェアへの影響を迅速に特定し、それを修正する必要があるからです。このような大規模な作業環境では、自動化が不可欠であり、手作業で行うことはまず不可能です。

Manage critical tasks

重要なタスクの管理

Other important tools to be considered and acquired are those which help manage critical tasks, such as project management, tracking project health and ensuring clear and quick communications between developers, open source communities, and others inside a company.

考慮、採用すべきその他の重要なツールとしては、重要なタスクの管理を支援するものがあります。たとえば、プロジェクトの管理、プロジェクトの健全性のトラッキング、および、開発者、オープンソース コミュニティ、企業内の他の人々との明確で迅速なコミュニケーションの実現などです。

Source code management

ソースコード管理

Most corporate software projects being developed through open source program offices use GitHub as their centralized hosting and development platform.

オープンソース プログラム オフィスを通して開発されている企業のソフトウェア プロジェクトのほとんどは、[GitHub](https://github.com/about)を集中ホスティングおよび開発プラットフォームに使用しています。

GitHub is an online source code management site that allows open source developers to manage and house their code in a central “repository” or storage space where participants can collaborate and build their code together. Some 64 million open source coding projects are hosted within GitHub today, involving some 23 million developers.

GitHubは、オンライン ソースコード管理サイトであり、オープンソースの開発者がコードを中央の「リポジトリ」またはストレージ スペースに管理して保管することができ、参加者は協調してコードを作成することができます。今日、GitHubは、約2,300万人の開発者を含む約6400万のオープンソース コーディング プロジェクトをホスティングしています。

GitHub users can add code, review submitted code, propose changes, get and offer feedback and provide project management using the service. GitHub uses the Git Version Control System, the open source project developed by Linux creator Linus Torvalds which provides organization for the code and people who are collaborating on open source. Each “contributor” has their own copy of the project repository they are working on, where they can make changes on their own computer and then submit it back to the project for future inclusion. That “pull request,” (example here) or code contribution, is then reviewed, discussed, modified and approved or rejected by the project organizers.

GitHubユーザーは、コードの追加、提出されたコードのレビュー、変更の提案、フィードバックの送信と受信、さらには、GitHubのサービスを通してプロジェクト管理を行うこともできます。GitHubは、Linuxの作成者Linus Torvaldsによって開発されたオープンソース プロジェクト[Git Version Control System](https://git-scm.com/)を使用しており、ソースコードとオープンソース上で協調している人々を結びつける機構を提供しています。それぞれの「コントリビューター」は、自分が作業しているプロジェクト リポジトリのコピーを持ち、自分のコンピュータで変更してから、将来リリースへの組み込みを見込んでプロジェクトに再提出します。その「プルリクエスト（pull request）」（[例](https://github.com/GitHub/opensource.guide/pull/402/files)）、あるいは、コード コントリビューションは、その後、プロジェクトのまとめ役によってレビューされ、議論され、変更され、承認または却下されます。

License compliance

ライセンス コンプライアンス

Also important are code scanning and compliance tools, which help track code provenance and license requirements. It’s important for companies to watch over the open source code being brought into its own infrastructure, products, and services to ensure license requirements are met.

コードの出所とライセンス要件をトラッキングするのに役立つコード スキャンとコンプライアンス ツールも重要です。 企業は、自社のインフラストラクチャ、製品、サービスに組み込まれているオープンソース コードのライセンス要件を満すよう管理していくことが重要です。

Your applications, for example, could include several thousand open source components. To protect your company from legal issues it’s critical to know these details. In scenarios that are high risk, users must dive into the code to deeply validate and verify that the licenses are what they say they are, depending on where your business is on a risk spectrum. (See our guide on using and distributing open source code.)

たとえば、あなたのアプリケーションには数千のオープンソース コンポーネントが含まれているかもしれません。企業を法的問題から守るためには、これらの詳細を知ることが不可欠です。リスクが高いと判断される状況では、ビジネスにどの程度リスクが影響するかの観点で、どのようなライセンスなのかについて、コードを詳しく点検、検証しなければなりません （本ガイド集の「オープンソース コードの使用と配布」を参照してください）。

“You must understand your risk profile, because in the end scanning is all about risk management. You can stick your head in the sand at one end then just trust and hope that you are OK. Or you could say ‘If I get sued, it’s going to devastate my business.’ You need to really be sure. So, you crack open the package and you look through all the lines of code and you find everything that could possibly be in there.”

「あなたはリスク特性（risk profile）を理解しなければなりません。結局、スキャンすることがリスク管理のすべてなのです。現実を直視しないで、あなたが大丈夫であることをただ信じて、願っているだけなのか。それとも、『もし私が訴えられれば、それは私のビジネスを破滅させることになるだろう。』と言えるのか。本当に安全であることが必要です。 だから、あなたはソフトウェアのパッケージを調べ、コードのすべての行をチェックして、そこに存在しているかもしれないリスクのすべてを見つけ出さなければなりません。」

[Jeff McAffer](https://twitter.com/jeffmcaffer) – Director of the Open Source Programs Office at Microsoft

Section 4

セクション4

Tools for managing source code

ソースコード管理のためのツール

As we discussed earlier GitHub is the go-to source code management system for most open source program offices these days. But GitHub alone won’t meet all your program’s code management needs — especially as you scale up your efforts.

先ほど説明したように、GitHubは、最近のほとんどのオープンソース プログラム オフィスで人気のあるソースコード管理システムです。しかし、GitHubだけでは、特に活動の規模が大きくなったときに、あなたのオープンソース プログラムのコード管理のニーズをすべて満たすことはできません。

Some of the tools used in the world of open source are aimed at improving GitHub itself by adding features it lacks, such as support for checking Developer Certificate of Origin (DCO) statements to be sure that code can be legally licensed and used in an open source project.

オープンソースの世界で使用されているツールのいくつかは、GitHubに欠けている機能を追加することを目的としているものがあります。例えば、コードが合法的にライセンスされ、オープンソース プロジェクトで使用できることを保証する開発者証明書 (Developer Certificate of Origin: DCO) のチェックをサポートするものもあります。

GitHub also has some deficiencies when it comes to code reviews, so there are available tools that can automatically send questionable code back to the contributors who created it and ask them to review and make needed changes. GitHub doesn’t have a way to force someone to review their code, so these clever tools make that happen to improve workflows.

GitHubにはコード レビューに関して、いくつかの不足もあります。そのため、疑問の余地のあるコードを作成したコントリビューターに自動的に返送し、レビューしたうえで必要な変更を加えるように要求するツールがあります。GitHubにはだれかにコード レビューを要求する機能はないので、これらのツールを使ってワークフローを改善することができます。

Other GitHub-specific tools expand on GitHub’s performance metrics capabilities, which tend to be very project specific rather than providing detailed information across whole organizations. For companies that maintain many open source code repositories across multiple GitHub projects, better tools are needed to organize and aggregate them to make sense of it all. A wide range of such tools are available from Amazon, Netflix, and Microsoft to help with those tasks.

他のGitHub向けのツールとしては、組織全体に渡って詳細な情報を提供するのではなく、個別プロジェクト向けにGitHub実績指標の提供機能を拡張したものがあります。 複数のGitHubプロジェクトにまたがり、多くのオープンソース コード リポジトリを保有している企業が意味のある指標を得るには、それらを整理し、集約するための良いツールが必要です。Amazon社、Netflix社、Microsoft社などから幅広いツールがそれらのために提供されています。

Here are some of the most popular and useful source code management tools which can streamline and help your GitHub presence:

GitHub活用を効率化し、支援するために、人気があり、かつ有用と思われるソースコード管理ツールをいくつか紹介します：

Source code scanning and license compliance

ソースコード スキャンとライセンス コンプライアンス

Antepedia Reporter – A commercial, fee-based application from Antepedia, Reporter is a report-generation product which lets developers, project managers, legal advisors and others create license compliance audits and IP rights management reports about the open source, public and private components in your code base.

Antepedia Reporter（リンク先更新中） – Antepedia社の商用有償アプリケーションReporterは、オープンソース、パブリック、またはプライベートのコンポーネントのコード ベースを調べ、開発者、プロジェクト管理者、法律顧問などがライセンス コンプライアンス監査とIP権利管理を確認するためのレポートを作成する製品です。

Black Duck Hub – The commercial Hub service scans code to identify all embedded open source components, and then automatically searches for known vulnerabilities for remediation. It can send alerts when new vulnerabilities are found in your code.

[Black Duck Hub](https://www.blackducksoftware.com/products/hub) – 商用のサービス ハブで、コードをスキャンして、コードが含むすべてのオープンソース コンポーネントを識別し、救済策の適用が必要な既知の脆弱性を自動的に検出します。新しい脆弱性がコード内に見つかった場合にも警告を送信してくれます。

[Black Duck Protex](https://www.blackducksoftware.com/products/protex) – Protex is a commercial, fee-based license compliance management tool from Black Duck which integrates with existing tools to automatically scan, identify and inventory open source software, while also enforcing license compliance and corporate policy requirements.

Black Duck Protex - Protexは、既存のツールと統合され、オープンソース ソフトウェアを自動的にスキャン、識別し、目録を作成して、ライセンス コンプライアンスと企業ポリシー要件の遵守を支援するBlack Duck社が提供する商用有償のライセンス コンプライアンス管理ツールです。

Copyright review tools – This collection of command line tools help make initial copyright file construction and subsequent review and update easier.

[Copyright review tools](https://wiki.debian.org/CopyrightReviewTools) - この一連のコマンドライン ツール群は、最初の著作権ファイルの作成と、それに続くレビューおよび更新を容易にしてくれます。

dep-checker – A dependency checker tool from The Linux Foundation, dep-checker performs a complete analysis of linkages between code packages.

[dep-checker](http://git.linuxfoundation.org/dep-checker.git/) - Linux Foundationの依存性チェックツールdep-checkerはコード パッケージ間のリンク状況をすべてチェックしてくれます。

FlexNet Code Insight – Flexera, which acquired licensing compliance vendor Palamida in 2016, offers FlexNet Code Insight to help automate corporate open source use among developers, legal teams and security staffers.

[FlexNet Code Insight](https://www.flexera.com/products/software-composition-analysis/flexnet-code-insight.html) - 2016年にライセンス コンプライアンス ベンダーPalamidaを買収したFlexera社は、FlexNet Code Insightを提供し、オープンソースを活用する企業の開発者、法務チーム、セキュリティ スタッフの業務自動化を支援します。

FOSSA – This is a commercial tool that automatically performs code dependency tracking, license compliance scanning in the background.

[FOSSA](https://fossa.io/) - コードの依存性トラッキング、ライセンス コンプライアンス スキャンをバックグラウンドで自動的に実行する商用ツールです。

FOSSology – A Linux Foundation project, FOSSology is an open source license compliance software toolkit which can run license, copyright and export control scans from the command line. A database and web UI are also available to create compliance workflows.

[FOSSology](https://www.fossology.org/) - Linux FoundationのプロジェクトFOSSologyは、コマンドラインで、ライセンス、著作権、輸出規制管理の必要事項をスキャンするためのオープンソース ライセンス準拠のソフトウェア ツールキットです。 コンプライアンス ワークフローを作成するためのデータベースとWebユーザー インターフェースも用意されています。

janitor.git – Code Janitor is an open source tool that helps evaluate source code for compliance with open source licenses. From The Linux Foundation, Code Janitor can be used with other products to check code.

[janitor.git](http://git.linuxfoundation.org/janitor.git/) - Code Janitorは、オープンソース ライセンスに対するコンプライアンスについてソースコードを評価するオープンソース ツールです。Linux Foundationから提供される[Code Janitor](http://www.static.linuxfound.org/sites/default/files/lf_foss_compliance_cjt.pdf)は他の製品と共に使用し、コードをチェックすることができます。

LicenseFinder – Detects the licenses of the code being used in your projects, compares those licenses against a user-defined whitelist and then provides an actionable report.

[LicenseFinder](https://github.com/pivotal-legacy/LicenseFinder) - プロジェクトで使用されているコードのライセンスを検出し、そのライセンスをユーザーが定義したホワイトリストと比較し、結果を実用的なレポートとして提供します。

Protecode Enterprise Analyzer – This commercial application is used to analyze and identify all code in any directory to determine code ownership and ensure open source license compliance based on predetermined internal policies.

[Protecode Enterprise Analyzer](https://www.synopsys.com/software-integrity/security-testing/software-composition-analysis.html) - この商用アプリケーションは、ディレクトリ内のすべてのコードを識別、分析し、コードの所有権を特定し、前もって定めた社内ポリシーに基づいて、オープンソース ライセンスのコンプライアンス状況を確認します。

scancode-toolkit – From nexB, the ScanCode suite of utilities scans code for licenses, copyright and dependencies to find, discover and inventory open source and third-party components used in your code.

[scancode-toolkit](https://github.com/nexB/scancode-toolkit) – nexB社から提供されるScanCodeユーティリティ スイートは、コードで使用されているオープンソースおよびサードパーティのコンポーネントを検索、検出し、目録を作成するためにコードのライセンス、著作権および依存関係をスキャンします。

SPDX – The Software Package Data Exchange (SPDX) specification is a standard format used to describe the components, licenses and copyrights associated with software packages. The SPDX standard aids compliance with free and open source software licenses by standardizing the way license information is shared between developers and companies. The SPDX specification is developed by the SPDX workgroup, which is hosted by The Linux Foundation. The group offers open source tools to help users of SPDX documents.

[SPDX](https://spdx.org/tools) - Software Package Data Exchange (SPDX） の仕様は、ソフトウェア パッケージに含まれるコンポーネント、ライセンス、および著作権を記述するために使用される標準フォーマットを規定するものです。SPDX標準は、開発者と企業の間でライセンス情報を共有する方法を標準化することにより、フリーおよびオープンソース ソフトウェア ライセンスのコンプライアンスを支援します。SPDXの仕様はLinux Foundationが主催するSPDXワークグループによって開発されました。同グループは、SPDXドキュメントのユーザーを支援するオープンソースのツールを提供しています。

WhiteSource – Provides licensing, security, code quality and reporting analysis for managing open source components in real-time by automatically and continuously scanning dozens of open source repositories.

[WhiteSource](https://www.whitesourcesoftware.com/) - オープンソース コンポーネントをリアルタイムで管理できるように、数十のオープンソース リポジトリを自動的、継続的にスキャンし、ライセンス情報、セキュリティ情報、コード品質、および分析レポートを提供します。

Bug tracking

バグ トラッキング

Bugzilla – Server-based software featuring an advanced query tool that can remember searches, integrated email capabilities and a comprehensive permissions system. Bugzilla is used by Mozilla as its bug tracking system.

[Bugzilla](https://www.bugzilla.org/) – 検索履歴を記憶する高度なクエリ ツール、統合された電子メール機能、包括的なアクセス許可システムを備えたサーバーベースのソフトウェア。Bugzillaは、バグ トラッキング システムとして[Mozilla](https://bugzilla.mozilla.org/)で使用されています。

GitHub Issues – GitHub’s own integrated feedback and bug tracker, GitHub Issues is available as part of GitHub’s project hosting.

[GitHub Issues](https://help.github.com/articles/about-issues/) – GitHubで使用されているフィードバックとバグ トラッカーを統合したシステムです。GitHub Issuesは、GitHubのプロジェクト ホスティングの一環で利用可能です。

GitLab – This bug tracking tool unifies issue tracking, code review, Git repository management, activity streams, wikis and more in a single UI to assist your open source projects.

[GitLab](https://about.gitlab.com/) - このバグ トラキッング ツールは、イシュー トラッキング、コード レビュー、Gitリポジトリ管理、配信技術であるアクティビティ ストリーム、Wikiなどを統合したツールで、単一のUIで操作でき、オープンソース プロジェクトを支援します。

JIRA – From Atlassian, JIRA contains custom filters, developer tool integrations, customizable workflows and rich APIs to integrate JIRA with other applications.

[JIRA](https://www.atlassian.com/software/jira) – Atlassian社のJIRAは、カスタム フィルター、開発者ツールの統合、カスタマイズ可能なワークフロー、およびJIRAを他のアプリケーションと統合することができる豊富なAPIを提供しています。

Archiving and release management

アーカイブとリリース管理

Artifactory – Also from JFrog, Artifactory is a repository manager which supports software packages created in any code language. It integrates with all major DevOps and continuous integration and continuous deployment tools.

[Artifactory](https://jfrog.com/artifactory/) - JFrog社のArtifactoryは、いかなるコード言語で作成されたソフトウェア パッケージでもサポート可能なリポジトリ マネージャーです。すべての主要なDevOps、および、継続的インテグレーション・継続的デリバリー (CI/CD) ツールと統合できます。

Bintray – An archiving tool from JFrog that allows companies to publish their code release archives to maintain storage for older and larger files.

[Bintray](https://bintray.com/) - JFrog社のアーカイブ ツールで、コード リリースのアーカイブを公開することにより、古くて大きなファイルのためのストレージを維持・管理できるようにします。

Docker Hub – A cloud-based registry service which allows users to link to code repositories and build and test their images. It also stores manually-pushed images and links to Docker Cloud so users can deploy images to project hosts. Docker Hub is a centralized resource for container image discovery, distribution and change management, collaboration and workflow automation throughout the development pipeline.

[Docker Hub](https://hub.docker.com/) - ユーザーがコード リポジトリにリンクし、Dockerのイメージを構築し、テストができるクラウド ベースのレジストリ サービスです。 また、手動で、プッシュしたイメージや[Docker Cloud](https://docs.docker.com/docker-cloud/)へのリンクも格納できるので、ユーザーはイメージをプロジェクト ホストに展開できます。Docker Hubは、コンテナ イメージの検出、配布および変更管理、コラボレーション、ワークフローの自動化のすべてを一元化した開発パイプラインの中心となるツールです。

github-release – The built in functionality part of GitHub which lets users package and edit releases of projects on GitHub so they are available for use by other community members.

[github-release](https://github.com/aktau/github-release) – GitHubの機能の一部で、ユーザーはGitHub上のプロジェクトの[リリースをパッケージ化したり、編集したり](https://help.github.com/articles/about-releases/)できるので、他のコミュニティ メンバーがそれらを利用できるようになります。

Section 5

セクション 5

Tools for tracking project health

プロジェクトの健全性をトラッキングするツール

Monitoring and tracking the overall health of open source projects as they grow and mature is a core task for an enterprise open source program. To accomplish it, you must gather tools which report on how individual open source projects are performing and being received by their communities – often across dozens, hundreds or even thousands of projects at once. The tools also must be able to roll the data into meaningful, useful, and actionable information about overall project performance across your entire open source portfolio.

成長、成熟しているオープンソース プロジェクトの全体的な健全性を監視し、トラッキングすることは、企業のオープンソース プログラムの中心となる重要作業です。 これを遂行するには、個々のオープンソース プロジェクトがどのように実施されているのか、コミュニティからどのように受け止められているのかについて、レポートしてくれるツールが必要です。数十、数百、時には数千のプロジェクトを同時に扱えなければなりません。このツールはまた、オープンソースのポートフォリオ全体に渡って、プロジェクト パフォーマンス全般について、有用で、行動に繋がる実用的な情報をレポート出力してくれなければなりません。

The bottom line here is it’s all about the critical and useful insights you can glean from the data – not about vanity metrics such as detailing how many “watcher” stars a project has logged, how many contributors have been part of the project since its start, or other metrics that lack important context.

閲覧した人の数、プロジェクト開始当初からの参加コントリビューター数などのような重要でない、価値のない指標に着目してはいけません。ここで重要なのは、データから重要かつ有用な洞察を得ることです。

The best project health tools must also help the project teams be responsive to the communities which support their efforts and encourage engagement and diversity with contributing developers. That means the tools help maintainers quickly respond to questions or feedback posted by community members so they remain enthusiastically engaged and don’t get bored and move on to other projects.

最も優れたプロジェクトの健全性トラッキング ツールは、コントリビューターとともにプロジェクト チームの活動を支え、参画と多様性を促進してくれているコミュニティに対し、プロジェク トチームが上手く対応するのに役立つものでなければなりません。つまり、ツールは、コミュニティ メンバーが投稿した質問やフィードバックにメンテナーが迅速に対応するのを支援します。そうすることでコミュニティ メンバーは熱心に関与し続け、退屈になったり、他のプロジェクトに移ったりすることはありません。

Some open source communities will have large groups of contributors, while others will have small niche groups of community members. The project health tools need to be able to work with projects of all sizes.

オープンソース コミュニティには、大きなコントリビューター グループを持つものもあれば、小さく、ニッチなコミュニティ メンバーのグループから構成されているものもあります。プロジェクトの健全性管理ツールは、あらゆる規模のプロジェクトに対応できる必要があります。

“Regarding existing tools and systems, my hope is that we’re quickly getting to a point where a company’s open source program office should not need to create any tools or technologies on their own. They should be able to find and use existing open source tools which can be used to manage their open source programs.”

「ツールやシステムは既に存在しているので、企業のオープンソース プログラム オフィスは自分でツールやテクノロジを作成する必要はないとすぐに気づくでしょう。彼らはオープンソース プログラムを管理するために使用できる既存のオープンソース ツールを見つけて使用できるはずです。」

[Jeff McAffer](https://twitter.com/jeffmcaffer) – Director of the Open Source Programs Office at Microsoft

Here are some of the most popular and useful project statistics and project health tracking tools:

最も一般的で、有用なプロジェクトの統計情報ツールとプロジェクトの健全性を管理するツールは次のとおりです。

CatWatch– CatWatch is an open source metrics dashboard from Zalando that fetches GitHub statistics for your GitHub accounts, processes and saves your GitHub data in a database. The data reveals the popularity of your open source projects, your most active contributors and other interesting statistics.

[CatWatch](https://github.com/zalando-incubator/catwatch) - CatWatchはZalando社が提供するオープンソースのメトリクス ダッシュボードで、あなたのチームのGitHubアカウントに対応したGitHub統計情報を取得し、GitHubデータを処理し、データベースに保存します。このデータは、オープンソース プロジェクトのポピュラリティ、最も活動的なコントリビューターおよびその他の興味深い統計情報をレポートしてくれます。

Gander – Gander is a dashboard which generates usable metrics for a range of open source projects in one quick look. Created by PayPal, Gander is designed for individuals who are responsible for running Open Source Program Offices or keeping track of multiple open source projects.

[Gander](https://github.com/paypal/Gander) - Ganderは、さまざまなオープンソース プロジェクトについて、有用なメトリクスを一目で分かるように表示するダッシュボードです。 Ganderは、オープンソース プログラム オフィスの運営責任者や複数のオープンソース プロジェクトの管理責任者向けに、PayPal社によって作成されたものです。

GHCrawler – Created by Microsoft, GHCrawler is a GitHub API crawler that crawls a GitHub-hosted project and automatically tracks, retrieves, and stores its contents. GHCrawler is primarily intended for people trying to track sets of organizations and data repositories.

[GHCrawler](https://github.com/Microsoft/ghcrawler) – Microsoft社によって作成されたGHCrawlerは、GitHubがホストするプロジェクトをインターネット上で調査（クロール）し、そのコンテンツを自動的にトラッキング、取得、保存するGitHub APIで実装された調査ツールです。GHCrawlerは、多数の組織、データ リポジトリをトラッキングしようとする人向けのツールです。

Gittagstats – Gittagstats is a tool which generates statistics reports from a set of tags for a Git repository. The tool was created by Qualcomm.

[Gittagstats](https://github.com/mcharleb/gittagstats) - Gittagstatsは、Gitリポジトリのタグから統計レポートを生成するツールです。このツールはQualcomm社によって作成されました。

Grimoire Lab – Bitergia has a variety of open source tools to measure open source project statistics, from mailing lists to meetup groups.

[Grimoire Lab](https://chaoss.github.io/grimoirelab/) – Bitergia社は、メーリングリストからMeetupグループまで、オープンソース プロジェクトの統計情報を測定するためのさまざまなオープンソース ツールを提供しています。

OSS-dashboard – The Open Source Program Dashboard, which comes from Amazon, is a multi-function dashboard which can be used to view and monitor many GitHub organizations and or users at one time.

[OSS-dashboard](https://github.com/amzn/oss-dashboard) – Amazon社のOSS-dashboardは、多くの機能を提供するダッシュボードで、多くのGitHubの組織、ユーザーを一度に表示して監視することができます。

OSS Tracker – OSS Tracker, from Netflix, collects data about a GitHub organization and aggregates it across all projects within that organization in a single user interface. All repositories are listed and metrics are combined for an organization, but community managers can also organize projects into functional areas and appoint administrators to assign management and engineering leads.

[OSS Tracker](https://github.com/Netflix/osstracker) – Netflix社のOSS Trackerは、GitHub組織内のすべてのプロジェクトに対して、単一のユーザー インターフェースでデータを収集し、集約してくれます。組織のすべてのリポジトリが一覧表示され、メトリクスは統合されます。また、コミュニティ マネージャーが、プロジェクトを機能別に編成して情報収集したり、管理とエンジニアリングの役割を管理責任者に割り当てたりすることもできます。

“The goal is to have the tools, along with transparent data and metrics-related information, which can be used to guide the organization.”

「目標は、トランスペアレントなデータとメトリクス情報が提供され、組織を導くために使用できるツールを持つことです」

[Chris Aniszczyk](https://twitter.com/cra) – COO of the Cloud Native Computing Foundation

The TODO Group also offers a helpful list that adds other tools as well:

TODOグループはまた、他のツールも加えた有益なリスト（[helpful list that adds other tools](https://github.com/todogroup/awesome-oss-mgmt)）も提供しています：

For better code reviews

より良いコード レビューのために

mention-bot – Developed by Facebook, this tool automatically mentions potential reviewers for code contributions by community members to speed up the review process.

[mention-bot](https://github.com/facebookarchive/mention-bot) – Facebook社によって開発されたこのツールは、レビュー プロセスをスピード アップするために、コミュニティ メンバーによるコントリビューションコードに対して適切なレビュアーを自動的に指名します。

PullApprove – Brings more formalization to code contributions – or pull requests – by improving code quality through peer-review, enforcing style guidelines, catching errors and providing security checks on code.

[PullApprove](https://about.pullapprove.com/) - ピア レビューによるコードの品質の向上、スタイル ガイドラインの適用、エラーの検出、および、コードのセキュリティ チェックの提供によりコードのコントリビューション、すなわちプル リクエストに定型化されたプロセスを適用します。

sentinel – A repository management bot which reviews and tests code contributions, builds a list of maintainers for the repository and communicates the status of a pull request with users.

[sentinel](https://github.com/habitat-sh/sentinel) コードのコントリビューションのレビューとテストを行い、リポジトリのメンテナーのリストを作成し、プル リクエストのステータスをユーザーに伝えるなど、リポジトリ管理を支援するボットです。

For Contributor License Agreements

CLA (コントリビューター ライセンス アグリーメント) 関連

CLA Assistant – Contributed by SAP, the CLA Assistant streamlines workflows by handling the legal side of contributions for users. The Assistant asks code contributors to sign CLAs as they make their code contributions and authenticates each contributor with his or her GitHub account. It also updates the status of a pull request when the contributor agrees to the CLA and automatically asks users to re-sign the CLA for each new pull request if changes are made to the CLA.

[CLA Assistant](https://github.com/cla-assistant/cla-assistant) – SAP社から提供されるCLA Assistantは、コントリビューションの法務関連処理をサポートし、ユーザーの作業を簡略化します。CLA Assistantは、コード コントリビューションを行うために、GitHubアカウントで認証する際に、コード コントリビューターがCLAに署名するように求めます。 また、コントリビューターがCLAに同意すると、プル リクエストのステータスが更新されます。CLAに変更が加えられると、新しいプル リクエストごとにCLAに再署名することを自動的にユーザーに求めます。

CLA Portal – From VMware, CLA Portal adds a workflow to enable contributors to digitally sign a CLA for pull requests to your GitHub repositories. When a developer opens a pull request, they are prompted to sign the agreement if needed. Also included is an administrator interface for CLA authoring, CLA-to-project mapping, and agreement reviews.

[CLA Portal](https://github.com/vmware/claportal) – VMware社から提供されるCLA Portalは、コントリビューターによるGitHubリポジトリへのプル リクエストに対応してCLAにデジタル署名するワークフローが追加されています。 開発者がプル リクエストをオープンすると、必要に応じてCLAに署名するよう促されます。 また、CLAの起草、CLAからプロジェクトへのマッピング、および契約書レビューのための管理者用インターフェースも提供されています。

DCOB – A Developer Certificate of Origin Bot which helps to enforce developer certificate of origin sign-offs for each code change in a pull request. The DCOB sets the status for each accepted code change, as required by the Developer Certificate of Origin.

[DCOB](https://github.com/chef/dcob) - 開発者オリジン証明書支援ボット（Developer Certificate of Origin Bot）は、プル リクエストのコード変更ごとに開発者証明書の署名を促すのをサポートします。 DCOBは、開発者オリジン証明書の要求に従って、アクセプトされたコード変更のステータスを設定します。

GitHub Management at Corporate Scale

企業規模でのGitHub管理

hubcommander – A Slack bot for GitHub organization management, HubCommander uses chat-ops – or conversation-driven development – to help manage GitHub projects. It creates a simple way to perform privileged GitHub organization management tasks without granting administrative or owner privileges to your GitHub organization members.

[hubcommander](https://github.com/Netflix/hubcommander) - GitHubの組織管理のためのSlackボットHubCommanderは、chat-ops (Conversation Driven Development:会話主導型の開発) を使用し、GitHubのプロジェクトの管理を支援します。GitHub組織のメンバーに管理権限または所有者権限を与えることなく、権限の必要なGitHub組織管理作業を行うための簡単な方法を提供します。

opensource-portal – From Microsoft, this tool is designed to help large organizations with their large-scale GitHub management operations, onboarding and more. This is one of a suite of tools provided by the Open Source Programs Office at Microsoft.

[opensource-portal](https://github.com/Microsoft/opensource-portal) – Microsoft社から提供されているopensource-portalは、大きな組織で、大規模なGitHubの管理操作、オンボーディング（新しい人の参加手続き）などを支援するように設計されています。 これは、Microsoft社のオープンソース プログラム オフィスによって提供される一連のツールの1つです。

settings – This app syncs repository settings defined in .github/settings.yml to GitHub, enabling pull requests for repositories.

[settings](https://github.com/probot/settings) - このアプリケーションは、.github / settings.ymlで定義されたリポジトリ設定をGitHubに同期させ、リポジトリに対するプル リクエストを可能にします。

zappr – Zappr is a GitHub integration built to enhance project workflows. From Zalando, zappr helps developers to increase productivity and improve open-source project quality by removing bottlenecks around pull request approval and helping project owners halt “rogue” pull requests before they’re merged into the project master branches.

[zappr](https://github.com/zalando/zappr) - Zapprは、プロジェクト ワークフローを強化するために構築されたGitHubインテグレーションです。 Zalando社から提供されているZapprは、プル リクエスト承認に関連するボトルネックを解消し、プロジェクト オーナーがプロジェクトのマスター ブランチに統合する前に不良なプル リクエストを停止させることで、開発者の生産性の向上とオープンソース プロジェクトの品質向上を支援します。

Project Quality

プロジェクト品質

CII Best Practices Badging – From The Linux Foundation, the Core Infrastructure Initiative (CII) Best Practices badge is a way for Free/Libre and Open Source Software (FLOSS) projects to show that they follow best practices. Projects can voluntarily self-certify for free by using this web application to explain how they follow each best practice.

[CII Best Practices Badging](https://bestpractices.coreinfrastructure.org/ja) - Linux FoundationのCore Infrastructure Initiative (CII) Best Practices Badgeは、Free / Libre and Open Source Software (FLOSS) のプロジェクトがベストプラクティスに沿っていることを証明する手段です。 プロジェクトは、このWebアプリケーションを使用して、彼らがどのようにベストプラクティスに沿って活動しているかを説明することにより、自発的に無償で自己証明を行うことができます。

CodeClimate – Code Climate empowers organizations to take control of their code quality by incorporating fully configurable test coverage and maintainability data throughout the development workflow. It’s free for open source projects!

[CodeClimate](https://codeclimate.com/) - Code Climateは、開発ワークフローを通して、完全に再構成可能なテスト カバレッジと保守用データを利用可能として、組織がコード品質を管理できるようにします。 オープンソース プロジェクトには無料で提供されています。

Section 6

セクション 6

Tools for communications and collaboration

コミュニケーションとコラボレーションのためのツール

Of course, open source development isn’t just about the code. It also requires healthy communications and collaborations between a diverse group of people who are working on the projects inside and outside of enterprises, as well as by staff members in a company’s Open Source Program Office.

もちろん、オープンソース開発はコードがすべてではありません。オープンソース プログラム オフィスのスタッフとだけではなく、企業内外のプロジェクトで仕事をしているさまざまなグループの人々との健全なコミュニケーションとコラボレーションが必要です。

For that developers can lean on tools they may already be using for other projects, including Internet Relay Chat (IRC), where developers can post inquiries and get quick responses to development-related topics. Another example is TWiki, which is an open source enterprise Wiki and web collaboration platform where developers can discuss code and projects and related topics.

そのために、[IRC (Internet Relay Chat)](http://www.irc.org/links.html) などの他のプロジェクトで既に使用しているツールを利用して、開発者が問い合わせを投稿したり、開発関連のトピックに素早く応答 ([quick responses to development-related topics](https://blog.andrewray.me/irc-the-secret-weapon-of-developers/)) したりすることができます。 もう一つの例は、オープンソースの企業用WikiおよびWebコラボレーションのプラットフォームである[TWiki](http://twiki.org/)です。開発者は、コードやプロジェクト、および関連するトピックについて議論することができます。

Communications can also be fostered through social media platforms, web portals, open source project repositories and other places where input, questions and discussions can be found and fostered.

ソーシャルメディア プラットフォーム、Webポータル、オープンソースのプロジェクト リポジトリ、さらには、インプット、質問、ディスカッションが行われ、それを育てるような場を通してもコミュニケーションを促進することができます。

Other useful tools include <mention-bot from Facebook, which can help get fast input turnaround on pull requests by automatically mentioning potential reviewers for reviews. This is especially appreciated when GitHub projects get too big for community members to subscribe to all of a project’s notifications.

他の有用なツールには、Facebookが提供する[<mention-bot](https://github.com/facebookarchive/mention-bot)があります。これは、適切なレビュアーにレビューを自動的に依頼することによって、プル リクエストに対するインプットを得るターンアラウンドを速めるのに役立ちます。 このツールは、コミュニティのメンバーがプロジェクトの通知にすべて目を通すことができないくらいに、GitHubプロジェクトが大きくなりすぎた場合に特に有効です。

Then there’s Slack, which is an online team project management and communications platform where users can access and share messages and files, organize workflows, perform searches for information and more. Slack can be configured to receive notifications for support requests, code check-ins, error logs and other tasks as well.

そして、オンラインでチーム プロジェクト管理とコミュニケーション プラットフォームを提供する[Slack](https://slack.com/intl/ja-jp)では、ユーザーがメッセージやファイルにアクセスしたり、これらを共有したり、ワークフローを編成したり、情報検索を行ったりすることができます。Slackは、サポート リクエスト、コードのチェックイン、エラー ログ、その他のタスクの通知を受け取るように設定することもできます。

And don’t forget your company’s public relations and marketing staff when it comes to shouting out your company’s participation and support of open source. Social media accounts with sites including Twitter, Reddit, Facebook, LinkedIn, Google+ and others are important, as well as the use of internal and external blogs and websites. Customer Relationship Management (CRM) software, as well as email blasts and newsletters, can help companies keep customers and clients informed about their open source progress.

そして最後に、あなたの会社がオープンソースに参加し、サポートしていることを発信してくれる広報・マーケティング要員の登場です。Twitter、Reddit、Facebook、LinkedIn、Google+などのサイトのソーシャル メディア アカウントは、内部および外部のブログやWebサイトと同様に重要です。Customer Relationship Management (CRM) ソフトウェアは、電子メールの配信やニュースレターと同様に、企業が顧客にオープンソースの進展状況を知らせるのに役立ちます。

Section 7

セクション 7

Tools for corporate-scale GitHub management

企業規模のGitHub管理のためのツール

When it comes to the tools your company provides and uses for its corporate open source projects, the most important ones are arguably those which help companies manage their corporate-scale GitHub operations. GitHub is a perfect platform for many operations, but for large, complex companies such as Google, Microsoft, Facebook, Twitter, LinkedIn and others, there can be many limitations to using the standard GitHub offerings.

企業のオープンソース プロジェクトに使用するツールについて言えば、企業規模のGitHub運用を管理するのに役立つツールが最も重要なものであることは間違いありません。GitHubは多くの作業にとって完全なプラットフォームですが、Google社、Microsoft社、Facebook社、Twitter社、LinkedIn社などの大規模で複雑な企業では、GitHubの提供する標準サービス範囲では多くの制限を受けます。

Large enterprises need many more capabilities, including things like identity management, settings and permissions management, security and two-factor authentication enforcement, as well as deeper means to understand and track code repositories.

大企業には、コード リポジトリをより深くトラッキングし、理解するための手段に加えて、アイデンティティ管理、設定とアクセス許可管理、セキュリティと2段階認証の実施などの機能も必要です。

That’s where specialized, automated tools often need to be built to handle tasks such as onboarding, offboarding, enforcing security policies and giving developers request access to repositories.

そのような環境では、オンボーディング（新しい人の参加手続き）、オフボード（脱退手続き）、セキュリティ ポリシーの実施、開発者にリポジトリへのアクセスを要求するなどの業務を処理するために、特別な自動化ツールを構築する必要があります。

Microsoft responded to its own unique requirements by building its own tools to handle many such tasks to streamline and improve its open source program. Microsoft has a healthy presence on GitHub, with some 1,345 repositories and involving about 3,580 developers to date.

Microsoft社は、オープンソース プログラムの効率化、改善に向けて、そのような多くの業務を処理する独自のツールを構築することで、自身の要件に対応してきました。Microsoft社はGitHubに約1,345のリポジトリを持ち、約3,580の開発者を抱え、健全な状態のGitHub [(healthy presence on GitHub](https://github.com/Microsoft)) を維持しています。

“That management of your GitHub presence is something that as you scale, it becomes important. You get a GitHub organization, which is a collection of repositories, and then you get members and you have teams. Managing all of that stuff becomes a little bit complicated, especially if it starts to scale out to hundreds of repositories, hundreds of people and multiple organizations on GitHub.”

「GitHubの管理は、その規模が拡大するにつれ、重要になってきます。あなたはリポジトリの集合となるGitHubの組織（GitHub organization）を作成し、メンバーを集め、チームを構成します。特にGitHubが何百ものリポジトリを持ち、何百人もの人々が参加し、複数の組織を持つように拡大すると、このようなことをすべて管理することは少しばかり複雑になります。」

[Jeff McAffer](https://twitter.com/jeffmcaffer) – Director of the Open Source Programs Office at Microsoft

One of the things Microsoft created was a custom-built self-service GitHub management and onboarding portal for organizing its projects, repositories, and teams. On its simplest level, the web-based portal allows developers to map their Microsoft company ID to their GitHub ID, which bolsters system security and helps simplify the organization of large numbers of developers who are involved in large numbers of important projects.

Microsoft社が作成したものの一つは、プロジェクト、リポジトリ、チームを組織化するカスタムビルドでセルフサービスのGitHub管理とオンボーディングのポータル（[GitHub management and onboarding portal](https://www.jeff.wilcox.name/2015/11/azure-on-github/)）でした。 最も簡単なレベルでは、Webベースのポータルにより、開発者はMicrosoft社の企業IDをGitHub IDにマップすることができ、システムのセキュリティを強化し、多数の重要なプロジェクトに携わる多数の開発者の組織（GitHub organization）を簡素化できるようにしました。

The portal also lets employees authenticate with GitHub and Microsoft, which creates a “virtual link” of their identities so they can do their work while giving them needed permissions for tasks depending on their work roles. If employees leave the company, the system can be adjusted to remove or reclassify their access rights as needed.

また、ポータルでは、GitHubとMicrosoft社の認証を元に、アイデンティティの「仮想リンク」を作成し、業務に応じて、業務に必要なアクセス許可を与えるので、従業員はそれに従って作業を行うことができます。 従業員が退社する場合、システムは必要に応じてアクセス権を削除、あるいは、再分類します。

The portal runs on one or more cloud servers and relies on a cache to help with sessions and reduce pressure on the GitHub API. The Microsoft portal, which averages about 1,000 unique users daily as a tool for its engineers, is part of the company’s growing open source efforts, which now includes more than 10,000 engineers who are using, contributing to and releasing open source code.

ポータルは1つ以上のクラウド サーバー上で動作し、キャッシュ機能を活用してセッションを支援し、GitHub APIの負荷を軽減しています。 技術者のためのツールとして平均して、毎日約1,000人のユニーク ユーザー（UU）をサポートするMicrosoft社のポータルは、オープンソース コードを使用、コントリビューション、リリースする1万人以上の技術者と共に、ますます拡大している同社のオープンソース活動の一部になっています。

Section 8

セクション 8

Final Words

結論

Hey, nobody said it was going to be simple to move your company into the world of open source. But plenty of other companies, including giants like Microsoft and Google have done this before you and have provided detailed road maps, code, suggestions, and more to make your own journey easier.

あなたの会社がオープンソースの世界に移行するのは簡単だと誰も言わなかったのですか。Microsoft社やGoogle社のような超大手企業を含め、多くの企業が、あなたの会社より先に、すでにそれを実現しています。彼らは、あなたが彼らより簡単にそれを実現できるように、詳細なロードマップ、ソースコード、提言などを与えてくれます。

The creation of an open source program office and the selection of a package of critical tools to get your efforts started are within your grasp. And they are likely already inspiring great anticipation among your developers, many of whom are probably already contributing to open source projects on their own (or at work, under cover of darkness).

オープンソース プログラム オフィスの設立と、あなたが活動を開始するために必要な必須ツールパッケージは、あなたの手の届くところにあります。 また、すでにあなたの会社の多くの開発者が、オープンソースに触発されて、オープンソース プロジェクトにコントリビューションしている可能性があります（目立たないように行っているのかもしれませんが）。

By collaborating on open source projects and inviting others to collaborate with you, your company can gain immeasurable benefits and drive its progress forward with energy and innovation.

オープンソース プロジェクトで協業し、他の人たちにあなたと協調してもらうことで、あなたの会社は計り知れないメリットを得て、さらに、エネルギーと革新をともなってあなたの会社の進歩を推進することができます。

Having the right tools is critical to empowering your company’s open innovation.

適切なツールを選択することは、あなたの会社のオープン イノベーションに力を与える上で非常に重要です。

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](http://todogroup.org/)をご覧ください。

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。