Measuring Your Open Source Program’s Success

オープンソース　プログラムが成功しているのかを測定する

Open source program managers must demonstrate the ROI of their efforts. This guide provides an overview of some of the standard ways that organizations evaluate their open source programs, projects, and contributions.

オープンソース　プログラムのマネージャーは、成果のROIを明らかにしなければなりません。 このガイドでは、組織がオープンソースのプログラム、プロジェクト、およびコントリビューションを評価するための標準的な方法について、その概要を説明します。

Learn what to measure, how to define success, and how to best use this information to advance your open source program objectives, demonstrate effectiveness, and gain support.

何を測定するか、どのように成功を定義するか、そしてその情報をどのように使ってオープンソース　プログラムの目標達成を推進し、成果を明らかにし、サポートを得ていくのかについて、学びます。

Contents

How to define success

Why set goals?

How to set goals

Common goals

What to track

More metrics

Final words

内容

成功をどのように定義するのか

なぜ目標を設定するのか

どのように目標を設定するか

共通の目標

何をトラッキングするのか

その他のメトリクス

結論

View All Guides »

Contribute on GitHub »

Contributors to this Guide

Chris Aniszczyk

Chris Aniszczyk

COO of CNCF

Christine Abernathy

Open Source Developer Advocate at Facebook

Joe Beda

Co-founder and CTO at Heptio

Gil Yehuda

Senior Director of Open Source

at Oath (Yahoo + AOL)

Sarah Novotny

Kubernetes Community Manager at Google

Section 1

セクション　1

How to define success

成功をどのように定義するのか

Smart organizations understand the value of investing in open source development and set goals related to open source use and participation. But every open source program defines success a little differently. The goals you set, and metrics you track, will vary according to the reasons you’re investing in open source — whether it’s to recruit developers, bring in new ideas and technologies through open innovation, achieve faster time to market, lower development costs, or myriad other reasons.

賢明な組織は、オープンソース開発への投資の価値を理解し、オープンソースの活用、参加に関連する目標を設定します。 しかし、各々のオープンソース　プログラムは、成功に対して、少しずつ異なった定義をしています。 設定した目標、トラッキングするメトリクスは、開発者を募集するためであったり、オープンなイノベーションを通じて新しいアイデアや技術を導入することであったり、市場投入までの時間を短縮したり、開発コストを削減したりするなど、あなたがオープンソースに投資する理由により、異なったものになるでしょう。

It’s important to set goals according to your unique strategy — and seek buy-in from the executive team to ensure that the open source strategy aligns with the overall business strategy. That said, there are some standard ways that open source program managers measure success, regardless of their industry, product, or business strategy. These include:

・Their developers’ participation and level of influence in external open source projects

・Their organization’s reputation in open source communities

・Their ability to recruit and retain talented developers

・The general health of the organization’s own open source projects and the business-critical projects its developers contribute to

・How well they manage open source license compliance

自身の戦略に従って目標を設定することが重要です。また、経営幹部チームの支持を得るためにも、オープンソース戦略が貴社のビジネス戦略に沿っていることが重要です。 とは言え、業界、製品、ビジネス戦略にかかわらず、オープンソースのプログラムマネージャーが成功を測る標準的な方法がいくつかあります。 以下がその方法です。

・あなたの会社の開発者が参加している外部のオープンソース　プロジェクトでの影響力

・ オープンソース　コミュニティにおけるあなたの組織の評判

・才能のある開発者を雇用し、維持する能力

・あなたの会社のオープンソース　プロジェクトおよび、開発者がコントリビューションしていて、ビジネス上重要なプロジェクトの健全性

・オープンソースライセンス　コンプライアンスをどのようにして正しく遵守しているか

Section 2

セクション　2

Why set goals?

なぜ目標を設定するのか

Before we dive into what an open source program tracks and how to do it, let’s talk for a minute about what you can achieve by setting goals and measuring against them.

どのオープンソース　プログラムをどのようにトラッキングするかを議論する前に、目標の設定と、目標に対する達成度を測定することによって得られるものについて、少し話しましょう。

First and foremost, tracking progress against your goals helps you make sure the open source projects you’re investing in (whether external or internal) remain healthy — that they are answering to the community, representing the company well, and helping to meet the broader goals of the open source program. Regular tracking can help set benchmarks for open source projects and serve as an early warning system that allows for course correction if a project veers off track, falls out of legal compliance, or simply needs to be retired.

まず重要なことは、あなたの目標と、それに対する進捗状況をトラッキングすることにより、投資しているオープンソース　プロジェクト（外部か内部かにかかわらず）が健全な状態にあるかどうか、すなわち、彼らはコミュニティに応え、会社をよく代表し、オープンソース　プログラムのより広い目標を達成するのをサポートしているか、を確認できるようになるということです。定期的なトラッキングは、オープンソース　プロジェクトのベンチマークに役立ち、早期警戒システムとして機能します。プロジェクトが軌道から外れたり、法的コンプライアンスが守られていなかったり、プロジェクトを単に終息させる必要がある場合などで、正しい方向へ修正させることができるようになります。

Careful (and strategic) measurement also makes great fodder for reports to upper management. Regular reporting helps ensure the program stays on track with its goals and the overall business strategy and helps program managers gain internal support for the program among executive leadership (especially if you’re meeting or exceeding those goals!)

注意深く（そして戦略的に）実施された測定評価は、上層部への報告において大切なインプットになります。 定期的に報告を行うことにより、プログラムがその目標と全体的なビジネス戦略にそって順調に推進されること確かにし、プログラムマネージャーが経営幹部から、プログラムにたいするサポートを得るのを助けます。（特にプログラムが目標を達成していたり、あるいは目標を超えていたりする場合は）

Facebook’s open source program office, for example, periodically posts the month-over-month results from its open source projects internally and sends an executive report to management.

たとえば、Facebook社のオープンソース　プログラムオフィスでは、オープンソース　プロジェクトの前月比の成果を定期的に内部で公開し、経営幹部に対しても報告書を送付しています。

“Reports are just a good way to raise awareness. Even though Facebook places a high value on open source (as an organization), it’s still always a good thing to market yourself internally all the time and show your value.”

「報告書は意識を高める良い方法です。 Facebook社ではオープンソースを組織としても高く評価いていますが、絶え間なく、自分自身を内部的にマーケティングして、あなたの価値を示していくことは、いつでも常に良いことです」

[Christine Abernathy](https://twitter.com/abernathyca) – Open Source Developer Advocate at Facebook

Publishing the results publicly on a regular basis also helps raise awareness of your organization’s open source activities among potential partners, users, and developers.

結果を定期的に公表することで、将来のパートナー候補、ユーザー、および開発者に対して、あなたの組織のオープンソース活動に対する認識を高めるのに役立ちます。

Getting the word out about results — the good, the bad, and the ugly — increases your program’s transparency, accountability, and thus credibility, in open source communities. See example open source report cards from Facebook and Google.

良かったもの、悪かったもの、たとえ醜いものでも、結果について言葉に出して表明することで、オープンソース　コミュニティで、あなたのプログラムの透明性が増し、説明責任に対する姿勢が理解され、信頼性が向上します。 [Facebook](https://code.fb.com/android/facebook-open-source-2016-year-in-review/)社や[Google](https://opensource.googleblog.com/2016/10/google-open-source-report-card.html)社のオープンソース通信簿の例を参照して見てください。

Section 3

セクション　3

How to set goals

どのように目標を設定するか

It’s OK to have lofty goals for your open source program, but set reasonable expectations for how you will get there and on what kind of timeline. First, it’s helpful just to start measuring to establish a baseline for performance. Set up the right tools for collecting data and make sure the data sources are clean and in a format you (and your manager) can understand. Many organizations create a dashboard of metrics for their open source programs, to track all of the data in one place and provide project snapshots that can help assess progress at a glance. (See our guide on Tools for Managing Open Source Programs.)

CNCF project analytics dashboard

あなたがオープンソース　プログラムに対する高い目標を設定することはかまいませんが、そこにどのように到達するのか、どのようなタイムラインで到達するのか、についてのリーゾナブルな道筋を設定してください。 まず、パフォーマンスのベースラインを確立するために、測定を開始しましょう。 データを収集するための適切なツールをセットアップし、正しいデータソースから、あなた（そしてあなたのマネージャー）が理解できるフォーマットで出力されることを確認してください。 多くの組織では、オープンソース　プログラムのメトリクスのダッシュボードを作成し、すべてのデータを一か所でトラッキングでき、進行状況を一目で把握できるプロジェクト　スナップショットを提供します。 （オープンソース　プログラムを管理するためのツールに関する私たちのガイド:[Tools for Managing Open Source Programs](https://www.linuxfoundation.org/resources/open-source-guides/tools-managing-open-source-programs/)を参照してください）。

Next, get all of the open source program managers and stakeholders together — at Facebook, this includes the engineering leads and project maintainers — and decide as a group where you want to go over the next 3-6 months (small, achievable increments). At the end of that time period, look back at how you did and adjust goals and tactics for the next pass, based on the previous pass.

次に、すべてのオープンソース　プログラムマネージャーとその関係者（Facebook社ではエンジニアリングのリーダーとプロジェクト　メインテナーも含まれます。）を対象にした、グループとして、次の3〜6ヶ月間で、小規模で、達成可能な目標を決めます。 その期間の終わりに、あなたがどのように目標を達成したかを振り返り、それに基づいて、次のステップに対する目標と戦術を調整します。

“I tend to find metrics based on what the community is feeling as pain, and try to change those metrics for the better to improve the community health.”

「私はコミュニティが苦痛を感じているメトリクス（指標）を見つけ出し、そのようなメトリクス（指標）が、コミュニティの痛みをなくし、健全になるように良い方向に変わるようにする傾向があります。」

[Sarah Novotny](https://twitter.com/sarahnovotny) – Kubernetes Community Manager at Google

In addition to the baseline performance metrics, here are some more things to consider as you’re setting goals for your program:

ベースライン　パフォーマンスのメトリクスに加えて、プログラムの目標を設定する際に考慮すべき点がいくつかあり、以下に示します。

・**Strategy alignment:** Do your goals align with core business strategy, product goals, and other internal business goals?

**戦略との調和：**あなたの目標は、あなたの会社のコアビジネス戦略、製品目標、およびその他の内部ビジネス目標と調和していますか？

**・Level of control:** Do your program managers have direct control over results, or does that live with engineering, legal, or some other function? Set goals that are achievable, within your control.

**・コントロールできる範囲**：プログラムマネージャーは結果に対して、直接的に対応できる立場ですか？それとも、エンジニアリング、法律、その他の組織と共同で対応しなければなりませんか？ あなたのコントロールできる範囲内で達成可能な目標を設定してください。

**・Project variation:** Goals can and should vary by project, depending on its purpose, community composition, technology stack, and other variables. For example, Facebook has noticed that its Javascript projects tend to be forked more often. They’ve learned (after many cycles of tracking!) these metrics aren’t necessarily the best indicators of project health for this type of project.

**・プロジェクトのバリエーション：**目標は、プロジェクトの目的、コミュニティの構成、技術のスタック、およびその他の変数に応じて、プロジェクトごとに異なっているはずです。 たとえば、Facebook社は、同社のJavascriptプロジェクトは他のプロジェクトより、頻繁にフォークされる傾向にあることに気づきました。多くのトラッキング　サイクルの後に、これらのメトリクスは必ずしもこのタイプのプロジェクトの健全性を測定するための最良の指標ではないと、ようやく、彼らは学びました。

**・Quantity vs. quality:**　Not all goals should be tied to a number. Process improvements that boost the quality of your projects are just as important, if not more so. Just because you’re hitting all the numbers, doesn’t necessarily mean your projects are healthy. Conversely, a very small project that’s not growing could also be extraordinarily healthy. “You could have a half-dozen core contributors and a dozen people who are active but not maintainers, but there are healthy discussions, and pull requests are handled in a short manner, and ponderous feature discussions have a clear beginning, middle, and end. That might be an incredibly healthy community, but it’s not going to have the gazillion stars or forks on GitHub, because it may be a niche project,” Novotny says. “So I tend to look at how the community is interacting with itself, how new leadership is being grown and mentored, and how any pain points are evolving.”

**・質と量:**すべての目標を数値化する必要はありません。 プロジェクトの品質向上のためのプロセス改善は、数値化以上に重要な場合もあります。 あなたがすべての数値化目標を達成しているからといって、必ずしもあなたのプロジェクトが健全であるとは限りません。 逆に、成長が止まっているように見える、小さなプロジェクトが非常に健全であることもあります。「あなたは6名程度のコア　コントリビューターと活発ではあるがメインテナーではないその倍程度のコミュニティで、健全な議論があり、 プルリクエストを簡単な方法で処理し、厄介な機能追加のディスカッションも、明確な、工程で処理されているプロジェクトがあるかもしれません。 それは信じられないほど健全なコミュニティかもしれませんが、それはニッチなプロジェクトなので、GitHubに人気度を表すGitHub starやフォークを多くは持っていないでしょう。」とNovotnyは言います。 「したがって、私は、コミュニティはどのように交流しているのか、新しいリーダーシップがどのように成長し、育てられているのか、悩みはどのように解決、改善されているのかを見るほうです。」

Section 4

セクション　4

Common goals

共通のゴール

When it comes to measuring your open source program’s success, it’s tempting to focus on the quantitative metrics for your projects: total number of contributors, lines of code, number of projects, etc. We’ll discuss what to measure to assess your project health in the next section. But first, there are many other important ways to measure your program’s success than strictly by these numbers.

オープンソース　プログラムの成功を測る場合は、コントリビューターの総数、コードの行数、プロジェクトの数など、プロジェクトの定量的なメトリクスに焦点を当てがちです。プロジェクトの健全性を評価するために何を測定すべきかについて次のセクションで議論しますが、これらの数値を厳密に測定することよりも、プログラムの成功を測定する他の多くの重要な方法があり、まずは、それについて紹介します。

“I think using metrics as a way to inform the trend is good. Using them as the only method of success leads you to trouble.”

「傾向を知らせる手段としてメトリクスを使用することは良いと思います。 成功のための唯一の方法としてそれのみで評価していると、トラブルにつながります。」

[Joe Beda](https://twitter.com/jbeda) – a founding engineer of Kubernetes at Google and Co-founder and CTO at Heptio

Kubernetes is one of the highest velocity open source projects on GitHub, attracting more than 80,000 commits from 2,760 developers at 1,181 companies over the last three years. But from the start, the project has managed its success in terms of whether its users were excited about the technology and using it, not by “some list of open source metrics,” Beda said.

Kubernetesは、GitHub上で最も速い速度で開発されているオープンソース　プロジェクトの一つで、過去3年間で1,181社の2,760人の開発者から8万件以上のコミットが提供されくらい人々を引き付けています。 しかし、このプロジェクトはスタート当初から、ユーザーはその技術とそれを使用することにエキサイティングであるかどうかという点から成功を勝ち取ってきました。「いくつかのオープンソースのメトリクスリストによる評価ではありません。」とBedaは言います。

Below are some of the common goals behind an open source program office, and the top ways that program managers measure against these goals to track the overall progress of the program. Some of these goals can’t be measured, per se, but are about improving processes, efficiencies, and quality. Others can be measured by conducting surveys or other methods of assessment such as regularly, and actively soliciting verbal or written feedback. (Talk to your team!)

以下に、オープンソース　プログラムオフィスで採用されている一般的な目標のいくつかと、プログラム管理者が、これらの目標のために、プログラムの全体的な進捗状況をトラッキングするために測定している代表的な方法を示します。 これらの目標の一部は、それ自体を直接測定できないもので、プロセス、効率性、品質の向上に関するものが含まれています。 いくつかのものは、調査を実施や、口頭で、または書面での積極的、定期的な評価行うような方法で測定可能です。（あなたのチームと話してみましょう。）

Goal #1 Ensure the efficient and legally compliant use of open source code.

目標＃1効率的かつ合法的なオープンソース　コードの使用を保証する。

This is where organizations typically start when they get involved in open source. They realize engineering is consuming a lot of open source software either in their infrastructure, or in their products and services, or both. A program office helps centralize policies and decision-making around open source consumption, track its use, and ensure the organization doesn’t run afoul of its legal obligations under the various open source licenses. Programs can also keep track of how well they help developers resolve any legal issues they may encounter.

これは組織がオープンソースに関わるようになったとき、最初に向き合う典型的な課題です。 まず、エンジニアリング部門がインフラストラクチャや製品やサービス、あるいはその両方で多くのオープンソースソフトウェアを使用していることに気づくでしょう。 プログラムオフィスは、オープンソースの活用に関するポリシーと意思決定を一元化し管理し、そのオープンソースの使用状況をトラッキングし、組織がさまざまなオープンソースライセンスの下で法的義務を遵守するようにします。 プログラムオフィスが、開発者が遭遇した法的問題を解決するために、どれだけ役立っているかをトラッキングすることもできます。

Some of the most common ways to measure against this first goal are aimed at ensuring that policies and processes are working the way they were intended and that the organization remains in legal compliance:

この第1の目標に対する、最も一般的な評価方法は、ポリシーとプロセスが意図したとおりに働いていること、組織が法令遵守を維持していることを確認することを測定するものです。

・How much open source code do you consume?

・どの程度オープンソース　コードを使用しているか。

・How well is that consumption tracked?

・使用についてどの程度うまくトラッキングされていますか。

・The policy for using open source code is clear and developers are aware of it.

・オープンソース　コード使用に対するポリシーは明確で、開発者はそれを認識しています。

・The processes and tools for bringing in code are clear and developers are following it.

・コードを取り込むためのプロセスとツールは明確で、開発者はそれに従っています。

・Which products and services are third-party code being used in?

・サードパーティのコードが使用されている製品とサービスはどれですか。

・How many compliance issues are you having and how quickly are they resolved? (You do you have a compliance program, right? See our legal resources from the Open Compliance Program for more on this topic.)

・コンプライアンスの問題は今、いくつありますか、どれほど迅速に解決されていますか？ （コンプライアンスプログラムを持っていますか。このトピックの詳細については、私たちが提供している法務関連情報、「[Open Compliance Program](https://compliance.linuxfoundation.org/)」を参照してください）。

Goal #2 Increase developer productivity.

目標＃2開発者の生産性の向上

Once you’re tracking and managing your open source use, you’ll want to make it easier for developers to contribute to open source projects. If your engineers have to go through layers of red tape to submit a bug fix or new feature to a project that your business depends on, you’re wasting precious time and resources. Developers also save considerable time over the long run by contributing upstream, rather than maintaining a separate fork of the project which accrues technical debt over time.

オープンソースの使用をトラッキングし、管理し始めると、開発者が行うオープンソース　プロジェクトに対するコントリビューションを、より簡単にできるようにしてあげたいと思うようになるでしょう。 あなたのエンジニアが、あなたのビジネスが依存しているプロジェクトにバグフィックスや新機能を提出するために、過度の規制で、融通のきかない手続きを経なければならない場合、貴重な時間、リソースを浪費しています。 開発者は、時間の経過とともに技術的な負債が蓄積されてくるプロジェクトからフォークしたものを保守するのではなく、アップストリームにコントリビューションすることによって長期的に多くの時間を節約することもできるようになります。

“We try to set ourselves up to be like those people on the marathon route who give water to the runners. We encourage the developers to take a couple of extra steps in our direction knowing that by doing so they actually will get to their goal in the long run. We really try to set up a program to be a support service, not a speed bump.”

「私たちは、マラソンランナーに水を与えるためにいるマラソンルート上の人のように自分自身を設定しようとします。 私たちは、私たちのところに寄るための追加ステップを取ることによって、開発者は長期的には、目標に到達することを実際に知ることができるという点で、私たちのところに寄ることを奨励します。 速度を減速させるためのスピードバンプではなく、サポートサービスになるようにプログラムを設定しようとしているのです。」

[Gil Yehuda](https://twitter.com/gyehuda) – Senior Director of Open Source at Oath (Yahoo + AOL)

>Metrics related to this goal are aimed at greasing the wheels for developers to contribute back to open source projects, as well as increasing the overall amount of code your organization contributes back upstream. Once you remove barriers to contribution and make the approval process clear and quick, you can expect more contributions and efficiencies. Things to track include:

この目標に関連するメトリクスは、開発者がオープンソース　プロジェクトにコントリビューションすることを円滑に進め、組織がアップストリームにコントリビューションする全体的なコード量を増やすことを狙ったものです。 コントリビューションのために障壁となっているものを取り除き、承認プロセスを明確かつ迅速にすることにより、より多くのコントリビューションが効果的に行われることが期待できます。 トラッキングするものは次のとおりです。

・Number of commits made to external projects identified as strategic to the organization

・組織にとって戦略的であると位置づけられる外部プロジェクトに対するコミットの数

・Number of developers contributing. Also, who are they and which projects do they contribute to?

・コントリビューションしている開発者の数 また、彼らは誰で、どのプロジェクトに貢献しているか

・Number of project maintainers employed by the organization (hired and grown)

組織に働いているプロジェクト　メインテナーの数（雇用された人、組織内で育てた人の内訳）

・Project health for the projects you contribute to

・コントリビューションしているプロジェクトの健全性

・Sentiment analysis: your organization’s reputation in open source communities

・センチメント分析（感情分析）：オープンソース　コミュニティにおけるあなたの組織の評判

・Are developers aware of the policy for contributing? (You have one, right?)

・開発者はコントリビューション　ポリシーを認識しているか（あなたはポリシーを持っているはずですが、それで良いですね）

・Do they follow the process for contributing? (ie must they sign a CLA, etc.)

・彼らはコントリビューションのためのプロセスに従っているか（CLAに署名しなければならないなど）

・Do they ask you for help and are you prompt in providing it?

・彼らがあなたに助けを求めているときに、速やかにサポートしていますか

・Amount of time between software releases – is it increasing or decreasing?

・ソフトウェア　リリース間隔 - 増加傾向ですか、減少傾向ですか

・What are the engineering costs associated with contributing upstream vs. maintaining forked code?

・フォークされたコードの保守のためのコストとアップストリームへのコントリビューションに関連するエンジニアリングコストはどのようになっていますか

Goal #3 Create and grow open source projects.

目標＃3オープンソース　プロジェクトを創設、そして成長

This is the primary goal of many open source programs at large, engineering-focused organizations such as Facebook, Google, Microsoft, Twitter, and many others. They’re creating hundreds (or even thousands) of open source projects that aim to solve hard technology problems. The goal is to attract outside users and contributors who bring in new ideas and help advance the technology at a faster pace — a concept University of California, Berkeley, professor Henry Chesbrough calls open innovation.

これはFacebook社、Google社、Microsoft社、Twitter社などの技術系組織の多くのオープンソース　プログラムが最重要目標にしているものです。 彼らは困難な技術的な問題を解決することを目指す数百（または数千）のオープンソース　プロジェクトを創出しています。 目標は、新しいアイデアをもたらし、より速いペースで技術を進歩させるくれる外部からのユーザー、コントリビューターを引き付けることです。カリフォルニア大学バークレー校のHenry Chesbrough教授はこのコンセプトをオープンイノベーションと呼んでいます。

“How do you actually get the smartest people in the world working at your company? Well, you open-source stuff and then you convince them to contribute to your projects.”

「あなたの会社で働いてくれる、世界で最も優秀な人材を実際にどうやって得るのですか。 そなんです、あなたはオープンソースで、あなたのプロジェクトにコントリビューションしてくれるように彼らを説得するのです。」

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

The many data points available to measure project health are key to tracking against this goal (see the top 5 in the next section). But there are other considerations as well:

プロジェクトの健全性を測定するために利用可能な、多くのデータポイントは、この目標に対する状況をトラッキングするためのキーです（次のセクションの上位5個を参照してください）。 しかし、他にも同様に重要なデータポイントがあります。

・Is there a clear policy for creating new open source projects and are developers aware of it?

新しいオープンソース　プロジェクトを設立するための明確なポリシーが存在し、開発者はそれを認識していますか

・Is there a clear and easy process for creating new projects and are developers following it?

・新しいプロジェクトを立ち上げるための明確かつ容易なプロセスがあり、開発者はそれに従いますか

・How easy is it for outsiders to contribute to your organization’s projects

・外部の人があなたの組織のプロジェクトにコントリビューションするのは簡単ですか

・Project maintainers are welcoming and helpful

・Projects are well-maintained and supported

・Code is well documented

・How to contribute is well-defined

・プロジェクトのメインテナーは歓迎され、みんなのためになっていますか

・プロジェクトは十分に維持管理され、サポートされています

・コードは十分に文書化されている

・コントリビューションの方法は明確に定義されています

・Other quantitative metrics such as number of new contributors, number of issues created, amount of time it takes to close issues, etc. (see the next section

・新しいコントリビューターの数、作成されたイシューの数、それをクローズするまでに要した時間など他の定量的なメトリクス　（次のセクションを参照してください）

・Number and diversity of external contributions your projects receive

・あなたのプロジェクトが外部から受けとるコントリビューションの数や多様性

・Popularity of your organization’s projects: GitHub stars, social media followers, etc.

・あなたの組織のプロジェクトの人気度：GitHubのstar、ソーシャルメディアのフォロワーなど

・Number of users in deployment and/or production

・導入や本番運用しているユーザー数

・Number, breadth, and quality of projects your organization launches. For example, mobile or data, infrastructure-related projects, etc.

・組織が立ち上げるプロジェクトの数、利用層、品質など。（例えば、モバイル分野、データ処理分野、インフラ関連プロジェクトなど）

・Performance increases in your project and related product

プロジェクトおよび関連製品のパフォーマンスの向上度

・Time between releases

・リリース間隔

Goal #4 Recruit and retain developers.

目標＃4開発者の採用、維持

Participating in and creating open source projects as an organization is a great way to attract developers — and onboard them quickly, with fewer resources devoted to training. Developers who use or contribute to your projects will already be familiar with your processes, tools, and technologies when they join the organization. (See our guide on Recruiting Open Source Developers.)

組織としてオープンソース　プロジェクトを立ち上げたり、参加したりすることは、開発者を引き付ける良い方法です。プロジェクトに魅了された彼らは、トレーンイングもほとんど必要なしに、彼らはすぐにプロジェクトに参加できるでしょう。プロジェクトを活用したり、プロジェクトにコントリビューションしたりしている開発者は、組織に参加するときには、すでにプロセス、ツール、およびテクノロジに精通しているでしょう。 （「オープンソースデベロッパーの採用：[Recruiting Open Source Developers](https://www.linuxfoundation.org/resources/open-source-guides/recruiting-open-source-developers/)」に関する当社のガイドをご覧ください）

But chances are that you as a program manager will not have a direct role in recruiting developers, and it may not be clear what immediate effect your organization’s open source participation has on hiring. To help make a more direct connection between program efforts and recruiting, Facebook conducts a biannual survey which asks new hires three basic questions:

しかし、プログラムマネージャーが開発者を採用するさいに直接的役割を果たすわけでもなく、また、あなたの組織のオープンソース参加が直接的に採用に影響を与えているのかについても、明らかではないことがあります。 プログラムの取り組みと採用の関係をより直接的に結びつけるため、Facebook社は、新人に次の3つの基本的な質問を年2回のアンケートで行っています。

1. Are they aware of the company’s open source program?

1．同社のオープンソース　プログラムを知っていますか

1. How did that awareness influence their decision to join the company?

2．その意識は、会社に入社する意思決定にどのような影響を与えましたか

1. Does their experience with open source apply to the work that they are doing now?

3．オープンソースでの経験は、今の仕事に生かされていますか

“We use the survey to measure the health of our open source culture and it speaks to the overall effectiveness of how people view our open source projects. It’s good to know the numbers trend upwards.”

Christine Abernathy – Open Source Developer Advocate at Facebook

「この調査を使って私たちのオープンソース文化の健全性を評価し、人々がオープンソース　プロジェクトをどのように見ているか、総合的に、有効性を示してくれます。 数字が上向きであれば、うれしくなります。」

[Christine Abernathy](https://twitter.com/abernathyca) – Open Source Developer Advocate at Facebook

Other common metrics for developer recruitment include:

開発者の採用に関するその他の一般的なメトリクスは以下の通りです。

・Which open source projects employees use and contribute to

・How new recruits heard about the organization

・The number of developers you’re bringing on through open source projects

・How many project maintainers you recruit (and grow)

・How long it takes to onboard new hires

・How open source developers are advancing in their careers

・Developer contributions are assessed as part of job performance

・Developers are recognized and rewarded for contributions

・Developers receive help and support in contributing

・従業員はどのオープンソース　プロジェクトを利用しているのか、どのプロジェクトにコントリビューションしているのか

・新入社員は組織をどのようにして知ったのか

・オープンソース　プロジェクトを通して、何名の開発者が採用できているのか

・何名のプロジェクトのメインテナーを採用したり、育てたりできているのか

・新入社員が、仕事ができるようになるまでの時間

・オープンソースの開発者がどのようにしてキャリアを伸ばしているのか

・開発者によるコントリビューションは、仕事の成果の一部として評価されているか

・開発者はかれらのコントリビューションが認められて、報われているか

・コントリビューションにおいて、開発者は支援やサポートが得られているか

Goal # 5 Promote open source culture.

目標＃5オープンソース文化のプロモーション

A lot of how open source programs contribute to engineering talent also comes down to cultivating the culture and practice of open source within your organization. That’s because organizations that embrace open source are known as good places for developers to work and innovate. Open source program managers are often ambassadors for an open source ethos within their organizations, as well as overseeing the policies and practices for collaboration.

技術系の人材に役立っているオープンソース　プログラムの多くは、あなたの組織内で、オープンソースの文化と実践を育成することにも繋がっています。 オープンソースを採用している組織は、開発者が仕事をし、大きく成長するための良い場所として知られているからです。 オープンソース　プログラムマネージャーは、コラボレーションのためのポリシーやその実践を監督するとともに、多くの場合、組織内のオープンソースの価値観をもたらす、「大使」となっています。

It’s important to track how open source culture advances within your organization to measure your program’s effectiveness. Some common ways to measure adoption of open source culture include:

オープンソース　プログラムの効果を評価するために、あなたの組織でオープンソース文化がどのように進化しているかをトラッキングすることは重要です。 オープンソース文化の浸透度を測定する一般的な方法としては以下があります。

・Awareness of and support for the open source strategy and program among management and individual contributors across all departments, from engineering to marketing and public relations

・エンジニアリングからマーケティング、広報まで、すべての部門の経営陣からコントリビューターの間でオープンソースの戦略とプログラムが認識され、サポートを受けている

・Branding and awareness in open source communities – how your organization is perceived

・オープンソース　コミュニティにおけるブランド力、認識度 - あなたの組織がどのように認識されているか

・Participation – you are participating actively in open source communities in a positive way

・参加度 - あなたの企業が積極的にオープンソース　コミュニティに参加しているか

・Training and mentorship – you work with developers to improve their open source contributions and projects, find opportunities to contribute, and learn the tools and processes of open source communities, ensure that contributors receive support from their peers and managers

・トレーニングと指導 - 開発者と協力して、オープンソースへのコントリビューション、オープンソース　プロジェクトを改善し、コントリビューションできる機会を見出し、オープンソース　コミュニティのツールとプロセスを学び、コントリビューターが同僚やマネージャーからサポートを受けられるようにする

・Adoption of a common toolset

・一般的なツールセットの採用

・Code quality is acceptable for open source/ external consumption

・コードの品質がオープンソース、外部利用可能のレベルにある

・Advocating on behalf of the organization – attending and speaking at conferences, writing articles or tutorials, etc.

・会議に出席したり、話したり、記事やチュートリアルを書くなど、組織を代表して、オープンソースのアドボケーターになっている

・Sponsoring foundations, groups, or hackathons

・ファウンデーション、グループ、ハカソンのスポンサーになる

Goal # 6 Align open source community interests with product interests.

目標＃6オープンソース　コミュニティの向かう方向と製品戦略を合わせる

Community advocacy is a fairly new, but increasingly popular, role in open source programs. You will often act as a liaison between your projects’ developer and adopter communities — representing the voice of external users building on your open source code and funneling information back to the product management team.

コミュニティ　アドボカシー（Community advocacy）は、オープンソース　プログラムの役割としては、新しいものでしたが、急速に広がりつつあります。あなたのオープンソース　コードを利用している外部のユーザーの声を、製品管理チームにフィードバックすることで、プロジェクトの開発者コミュニティとそれを利用しているユーザコミュニティの架け橋としての役割をはたすでしょう。

This is an important role that ensures your products and services are benefiting from your open source communities, and thus that your open source program stays in line with the organization’s broader business strategy and objectives. Some metrics to track success in your advocacy include:

これは、オープンソース　コミュニティから恩恵を受けている、あなたの企業が提供している製品やサービスをより確かなものしていくために重要な役割を果たします。その結果、オープンソース　プログラムは、組織の広範なビジネス戦略や目標に沿ったものであり続けるでしょう。 アドボカシー活動が成功しているかをトラッキングするメトリクスのいくつかを以下に示します。

・How many contributions are coming from outside the organization?

・組織外からのコントリビューションの件数はどれくらいか

・How many full-time contributors are outside your organization?

・あなたの組織の外にはフルタイムのコントリビューターは何人いるか

・How much externally contributed code is making it back into products?

・外部からのコントリビューションがどの程度が製品に含まれているか

・How many hires are coming from open source contributions?

・オープンソース　コントリビューターを何名雇用しているか

Section 5

セクション　5

What to track

何をトラッキングするのか

There are many ways to measure success and track progress for open source programs. Project health isn’t the only thing to track, but it’s still very important. The problem is, there is so much data available around open source projects. Anything you can get data on, you can collect and track. Again, the metrics each organization tracks — and what they do with the data — depends heavily on its own goals for the program, and its unique challenges in the market and in the open source community.

成功を評価したり、オープンソース　プログラムの進捗をトラッキングしたりする方法はたくさんあります。 トラッキングするのはプロジェクトの健全性だけではありませんが、健全性をトラッキングすることはやはり非常に重要です。 問題は、オープンソース　プロジェクトには利用可能なデータがあまりにも多く存在することです。 データとして存在していれば、それを収集してトラッキングすることができます。 また、各組織がどのようなメトリクスをトラッキングするか、それらをどのように扱うのかは、あなたのオープンソース　プログラムの目標、市場やオープンソース　コミュニティにおけるあなた独自の課題に大きく依存していることを再度確認してください。

“We gather the data that we can because that data is available but we don’t live in the numbers. We live in ensuring we have the right outcomes.” しかし

「データは存在していれば、可能な限り収集します。しかし、私たちは数字を得るためにデータを収集しているのではありません。私たちは、確実に、正しい成果を得るためにそれを収集しているのです。」

[Gil Yehuda](https://twitter.com/gyehuda) – Senior Director of Open Source at Oath (Yahoo + AOL)

For some (crazy or fully automated) program managers the answer is just to track all the things. But for large organizations in particular, there are so many projects it would be impossible to track everything and be able to make any sense out of it. So what are the real indicators of an open source project’s health?

（完全に手順が自動化されていたり、病的な収集癖があったりする）プログラムマネージャーにとっては、その答えは、すべてのものをトラッキングすることです。 しかし、特に大規模な組織では、すべてをトラッキングして、それらの意味を理解することが不可能なくらいプロジェクトが非常に多く存在します。 オープンソース　プロジェクトの健全性を評価するための本当の指標は何でしょうか。

Here are the top metrics for assessing overall project health in your open source program. These are only a starting point for more rigorous and thoughtful analysis. Keep in mind that these are the tips for helping program managers responsible for ensuring the health of multiple projects. The projects themselves should also track their own metrics for health. GitHub’s guide on open source metrics gives a great overview of what project maintainers should pay attention to.

ここでは、オープンソース　プログラムのプロジェクトの全般的な健全性を評価するための最重要メトリクスを示します。 これらは、より厳密で、示唆に富む分析を行う出発点にすぎません。 これらは、複数のプロジェクトにたいして、健全性を確保する責任を持っているプログラムマネージャーを支援するための秘訣（tips）になります。 もちろん、プロジェクト自身も、健全性評価のために専用のメトリクスをトラッキングする必要があります。 オープンソースのメトリクスに関するGitHubのガイド（[GitHub’s guide on open source metrics](https://opensource.guide/metrics/)）は、プロジェクトのメインテナーが注目すべき点について、優れた概要を説明してくれています。

These numbers are easily collected from GitHub using free and open source tools, as well as commercial offerings. Measure them at regular intervals (monthly, quarterly, and annually) to help benchmark progress for individual projects, as well as rolled up into aggregate counts for the program as a whole. Use them in reports to management, and to help your developers improve your projects.

これらのデータは、無償のオープンソースツール、商用サービスを使用してGitHubから簡単に収集できます。 定期的（毎月、四半期、毎年）にそれらを測定して、個々のプロジェクトの進捗をベンチマークするとともに、オープンソース　プログラム、全体としてもまとめます。 管理者へのレポートでそれらを使用したり、それらを開発者がプロジェクトを改善するために使用したりもします。

“We periodically just try to check and see, are the projects healthy or not, and just give them advice on what they should do better. But we don’t directly manage. We just give them the data and then sort of nudge them when we can, or when we have to.”

「プロジェクトが健全かどうか、私たちは定期的にチェックするようにしています。そして、彼らに改善すべき点についてアドバイスを与えるだけです。 しかし、私たちは、直接的な管理はしません。できるときに、あるいはしなければいけない時に、かれらにデータを与え、軽く改善を促すだけです。」

[Christine Abernathy](https://twitter.com/abernathyca) – Open Source Developer Advocate at Facebook

Number of contributors (and the ratio of external to internal contributions)

コントリビューターの数（内部コントリビューションと外部コントリビューションの比率）

Projects start with the majority of contributions coming from internal developers and evolve to include more outside contributions as the source code is used or forked. The healthiest projects that are sustainable over time have extremely diverse communities with the bulk of contributions coming from other companies in the project ecosystem that have taken commercial dependencies on that code. (Remember the 1,000+ companies contributing to Kubernetes?)

プロジェクトの初期段階は、大部分のコントリビューションは内部開発者でなされます。その後、ソースコードが使用またはフォークされるにつれて、より多くの外部からのコントリビューションを含むように進化していきます。 時間の経過とともに、安定的に継続している最も健全なプロジェクトは、プロジェクトのエコシステム内のそのコードがビジネスに直結している他の企業からの多くのコントリビューションを受けている、非常に多様なコミュニティを持っています。 （Kubernetesは1000以上の企業からコントリビューションを受けていますよね。）

Projects that are consistently attracting new external contributors are likely doing a good job maintaining the project, welcoming contributors, and incorporating feedback from the community. (Note: This may still be true for projects that aren’t growing their contributor base!)

新しい外部のコントリビューターを常に引き付けているプロジェクトは、プロジェクトを維持し、コントリビューターを歓迎し、コミュニティからのフィードバックを取り入れる良い仕事をしている可能性が高いようです。 （注：これは、コントリビューターの総数が増加していないプロジェクトでも当てはまるようです。）

Number of pull requests submitted, open, and accepted (and length of time they remain open)

サブミットされ、オープンにされ、受け入れられたプルリクエストの数（また、それらがオープンであった時間）

When a contributor finds a bug or has a feature request that they can (and have clearance to) patch or write themselves, they do so and submit it as a pull request (PR). Tracking the number of pull requests, and what happens with them, demonstrates how much code is being submitted by contributors outside of your employ and is thus an indicator of the level of outside interest in your projects.

コントリビューターがバグを見つけたり、パッチやコード提供するためのクリアランスを持っている人がそのための機能要求（feature request）を作成したりした時に、プルリクエストとして、それをサブミットします。 プルリクエストの数とそれらの状況をトラッキングすることは、あなたの会社の従業員以外のコントリビューターによって、どれだけのコードが提供されたかがわかるので、プロジェクトにおける外部の関心の高さをはかるための指標になります。

The length of time PRs remain open also indicates how responsive and welcoming your project maintainers are to outside contributors. If a PR sits for too long without response, potential contributors may take their good ideas elsewhere.

プルリクエストがオープンの状態の時間の長さは、あなたのプロジェクトのメインテナーが外部のコントリビューターに、いかに早く対応し、かれらに応えているかを示します。 プルリクエストは応答がなく、長時間放置されると、潜在的なコントリビューターはかれらの良いアイデアを他の場所で実現することになるかもしれません。

“When we have a good project, we probably don’t have a pull request that’s open for more than, I would say, two to three months at most. And that’s actually a lot.”

「良いプロジェクトは、おそらく2〜3か月以上プルリクエストをオープンの状態で放置しないでしょう。 残念だけど、それは実際にはたくさんありますが。」

[Christine Abernathy](https://twitter.com/abernathyca) – Open Source Developer Advocate at Facebook

Keep in mind that these metrics are highly dependent on the size of the project. Facebook’s smaller projects will try to keep the number of open pull requests at 10 or less. But keeping PRs at this limit would be challenging for bigger projects that have a lot of community input compared to the number of maintainers. Reviewing those pull requests takes time so bigger projects tend to have longer open PRs.

これらのメトリクスはプロジェクトの規模に大きく依存していることを心に留めて置いてください。 Facebook社の小規模なプロジェクトは、オープンな状態のプルリクエスト数を10以下に抑えるように努力しています。 しかし、プルリクエストを10以下に保つことは、メインテナーの数に比べ、コミュニティのインプットが非常に多い大規模なプロジェクトにとっては難しいでしょう。 これらのプルリクエストをレビューするには時間がかかるので、大規模プロジェクトではオープンな状態のプルリクエストがより長い時間になる傾向があります。

Facebook’s open source office frequently runs queries in the database and picks the top five projects with the most open PRs. They pinpoint a few issues and then take the opportunity to open up a dialogue with the project’s maintainers. They ask them a couple of questions to get to the root of the issue and see what might help solve the problem. More often than not, it’s simply a matter of refocusing their attention and reminding them that it’s important to keep the community happy. But occasionally, digging into the numbers points to deeper problems with a project. A lot of open PRs, or old PRs, could mean that only one or two people are maintaining the project — a potential red flag.

Facebook社のオープンソース　オフィスでは、頻繁にデータベースを検索し、オープンな状態のプルリクエストが最も多い上位5つのプロジェクトを選定しています。 彼らはいくつかの問題点を特定して、プロジェクトのメインテナーと議論を始めます。 彼らは問題の根本原因を特定し、問題解決を促進するために、彼らに2、3質問をします。 多くの場合、コミュニティに注意をはらい、コミュニティが幸せである状態を保つことが重要であることを再確認させるという簡単なものです。 しかし時には、数字を深く分析し、プロジェクトの深い問題を指摘することもあります。 多量のオープンな状態のプルリクエスト、古いプルリクエストの存在は、わずか1〜2人でプロジェクトを維持していることを示している場合があります。これは潜在的な赤信号です。

Number of issues submitted (and length of time they remain open)

サブミットされたイシューの数（また、それらがオープンであった時間））

Users who do not have the time, permission, or ability to submit a pull request, but encounter problems with your code can submit their bugs and feature requests as an issue. The number of issues, and how they are addressed, can indicate your projects’ levels of user adoption as well as how responsive maintainers are to user needs.

プルリクエストをサブミットするための時間、権限、またはその技術を持っていなくても、コードに問題が発生したユーザーは、バグや機能要求をイシューとしてサブミットすることができます。 イシューの数とそれにどのように対応しているかは、どの程度プロジェクトがユーザーに採用されているかと、メインテナーがユーザーのニーズに対して、いかに早く対応しているかを示してくれます。

This number depends, of course on how issues are tracked. For a project that’s only using GitHub to track bugs, issues are probably going to stay open for a much shorter time than a project that’s using GitHub for issues that includes feature requests. These considerations drag down, or drag up, the issue age.

この数は、もちろん、イシューのトラッキング方法によって異なります。 GitHubをバグのトラッキングのみで使用しているプロジェクトは、機能要求を含むイシューに対してもGitHubを使用しているプロジェクトと比較するとイシューがオープン状態である時間ははるかに短いでしょう。 これらの違いで、イシューの滞在時間は短くなったり、長くなったりします。

Number of commits per contributor (external vs. internal)

コントリビューターあたりのコミット数（外部と内部の比較）

The number of external commits a project has relative to the total is another indication of how effective the project is at innovating in the open — bringing in new ideas from the outside. Healthy projects will see the ratio of external contributors increase over time. Measuring the number of commits per contributor also helps to assess whether your projects are attracting new contributors and if those new contributors stick around.

プロジェクトの総コミット数の中で、外部からのコミットの数の比率は、外部から新しいアイデアを得るというオープン性に対する革新性で、プロジェクトがいかに成果を上げているかを示す指標になります。健全なプロジェクトでは、時間の経過と共に外部コントリビューターの割合が増えてきます。 コントリビューターごとのコミット数を測定することは、あなたのプロジェクトが新しいコントリビューターを引き付けているかどうか、そして新しいコントリビューターがコントリビューション継続してくれているかを評価するのにも役立ちます。

Number of external adopters

外部採用者の数

Each open source project should have a way to track organizations that opt to adopt the software in a production environment. Whether it’s through an ADOPTERS.md file or a simple list in the README, the key thing is to track this list and ensure it grows over time. If the number of external adopters stops growing or shrinks, it can signal everything from project maturity to project obsolescence.

各オープンソース　プロジェクトに、本番環境でそのソフトウェアを採用している組織をトラッキングする方法が必要です。 ADOPTERS.mdファイルやREADMEのシンプルなリストを通して調べる場合でも、重要なのはこのリストをトラッキングし、時間の経過と共に採用者が増加していることを確認することです。 外部採用者の数の増加が止まり、減少していると、それはプロジェクトが成熟した状態から衰退への道を歩み始めたというシグナルです。

Number of projects created or contributed to (program-wide)

作成したプロジェクトやコントリビューションしているプログラムの数

Track these metrics for each project your organization releases, but also the projects your developers are actively contributing to. In the process of creating your open source strategy, you should have identified the business-critical projects your organization is using and earmarked some investment for contributing to those projects. It’s important to measure your organization’s open source success not only by the health of your own open source projects but by its open source activity as a whole. This includes the health of the projects you rely on for product development and business operations, as well as ensuring your organization is legally compliant with the open source licenses of any projects you do use or release. (See our Open Compliance Program publications.)

あなたの組織がリリースするプロジェクトごとにこれらのメトリクスをトラッキングするだけでなく、あなたの組織の開発者が積極的にコントリビューションしているプロジェクトもトラッキングします。 オープンソース戦略を作成するプロセスでは、組織が使用しているビジネス上重要なプロジェクトを特定し、それらのプロジェクトにコントリビューションするための投資を考慮しておく必要があります。 組織のオープンソースの成功を、自身のオープンソース　プロジェクトの健全性だけでなく、オープンソース活動全体で測ることも重要です。 これには、あなたの会社の製品開発やビジネスオペレーションが依存しているプロジェクトの健全性だけでなく、使用したり、リリースしたりしているすべてのプロジェクトがオープンソースライセンスに法的に準拠していることを確認することも含まれます。 （[Open Compliance Program publications](https://compliance.linuxfoundation.org/references/compliance-related-publications)を参照してください）。

Section 6

セクション　6

Other metrics to track

トラッキングすべきその他のメトリクス

The basic project metrics are a good starting point to get your thumb on the pulse of your open source contributions. But successful program managers require a deeper dive into other important metrics.

基本的なプロジェクト　メトリクスは、あなたのオープンソースへのコントリビューションの実情を正確に把握するための第一歩になります。 しかし、プログラムマネージャーとして成功するためには、他の重要なメトリクスに対しても深い洞察が求められます。

Here are a multitude of other things you can and maybe should be tracking, depending on your goals. Remember that the number itself isn’t the goal — it’s the process of tracking them over time and finding patterns in the data that can inform project and process improvements. Measure for each project, and across projects for a comprehensive view of your program’s output and results.

トラッキング可能な多くのものが他にもあり、以下に紹介します。 あなたの目標に合わせて採用してください。数値そのものは目標ではないことを再度心に留めて置いてください。それはトラッキングを初めて時間と共に、プロジェクトやプロセスが改善されていることを知るためのパターンをデータの中に見つけるプロセスです。 プロジェクトごとにも測定し、プロジェクト全体に渡っても、あなたのオープンソース　プログラムの成果と結果を全体的に測定、評価します。

・Popularity/awareness

・人気度、認知度

・Visitors to the project website

・Total number of followers on GitHub/ GitLab

・Number of followers on social media accounts such as Twitter, Facebook, or LinkedIn

・News clips and media mentions

・Number of meetups organized and hosted (e.g., via meetup.com)

・プロジェクトのウェブサイトへの訪問者

・GitHub / GitLabでのフォロワー総数

・Twitter、Facebook、LinkedInなどのソーシャルメディアアカウントのフォロワー数

・ニュースクリップやメディアでの言及

　・設定したり、ホスティングされたりしているミートアップの数（たとえば、meetup.com経由）

・Influence

・影響度

・Number of employees in a maintainer/leadership role in your strategic projects

　・Diversity of contributors to your projects

・Patches rejected, and why

・Adoption

・Number of downloads

・Number of forks created

・Number of contributing external companies

・Stages of adoption (# of deployments in PoC and production)

・Number and quality of commercial dependencies (products) – This can be tracked by looking at the companies contributing to your projects, as well as following the news and trade press.

・あなたの戦略的プロジェクトでメインテナー、リーダーシップの役割をしている従業員の数

・あなたのプロジェクトへのコントリビューターの多様性

・拒否されたパッチとその理由

　　・採用されている状況

・ダウンロード数

・作成されたフォークの数

・コントリビューションしてくれている外部企業の数

・導入段階（POCおよび本番環境での導入数）

・企業のビジネスを支えているもの（製品）の数と質 - これは、あなたのプロジェクトにコントリビューションしている企業を調べたり、ニュースや業界誌の報道をフォローしたりしてトラッキングできます。

・Program costs

・プログラムの費用

・Staff: engineering, PR & marketing, legal

・Infrastructure and support

・Tools

・Conference attendance and travel

・Training

・Memberships and donations

・スタッフ：エンジニアリング、広報、マーケティング、法務

・インフラストラクチャとサポート

・ツール

・会議出席と出張

　　・トレーニング

・メンバーシップの会費、寄付

Section 7

セクション　7．

Final words

結論

Organizations evaluate their open source programs, projects, and contributions in whatever way makes the most sense for their needs. The most important thing to remember is to set a strategy and incremental goals to achieve it. What you track, and how, will naturally follow.

組織は、オープンソース　プログラム、プロジェクト、コントリビューションを評価し、かれらのニーズに最も合った方法で評価します。 覚えておくべき最も重要なことは、それを達成するための戦略と段階的な目標を設定することです。 そうすれば、あなたがトラッキングするもの、それをどのようにトラッキングするかは自然と分かるようになります。

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。