Winding Down an Open Source Project

オープンソースプロジェクトを終了させる

This Open Source Guide is designed to offer advice about how your enterprise and your development team can plan for the day when you are ready to end or move away from an unneeded open source project. By shutting down the project gracefully or by transitioning it to others who can continue the work, your enterprise can responsibly oversee the life cycle of the effort. In this way, you can also set proper expectations for users, ensure that long-term project code dependencies are supported, and preserve your company’s reputation within the open source community as a responsible participant.

This guide will help you decide when a project is no longer useful, understand how to disengage from a project, and determine what to do about its code, repositories, websites, wikis, and other project assets as you head in a new direction.

本オープンソースガイドでは、必要のなくなったオープンソースプロジェクトを終了させる、あるいは、プロジェクトから撤退する準備が完了したときに、あなたの会社やあなたの開発チームがどのように「その日」を計画立てることができるかについてアドバイスを提供しています。プロジェクトをきれいに閉鎖させる、あるいは、活動を継続できる他の組織に移管することにより、あなたの会社は責任を持って活動のライフサイクルを見届けることができます。そうすることによって、ユーザーに正しい見通しを提示することができ、長期に渡ってプロジェクトコードに依存しているモノ（ソフトウェア）が確実にサポートでき、さらに、オープンソースコミュニティの中で責任感のある会社としての評価を保つことができます。

本ガイドは、あなたが新たな方向へと進んで行こうとするに当たり、プロジェクトが不要になる時期を判断すること、プロジェクトへの関与をやめる方法を学ぶこと、さらには、コード、リポジトリー、Webサイト、Wiki、その他の資産の処分方法を決定すること手助けします。

Contents

目次

Life cycle planning for your open source project

What does a dead open source project look like?

Why plan for the end of a project, before you even launch it?

Deciding when to end or pull out of a project

How to end an open source project

Final words

オープンソースプロジェクトのライフサイクル計画

「死んだ」オープンソースプロジェクトはどんな風に見えるか？

プロジェクトの終了計画はなぜ必要か（プロジェクトの始動前においてさえも）？

　　プロジェクトをいつ終了、移管、あるいは、それから撤退するかを決定する

　　どのようにオープンソースプロジェクトを終了させるか

　　結論

View All Guides »

Contribute on GitHub »

Contributors to this Guide

David A. Wheeler

Core Infrastructure Initiative (CII)

Dr. David A. Wheeler

Core Infrastructure Initiative (CII)

Jared Smith, Capital One

Jared Smith

Open Source Community Manager, Capital One

Jared Smith

オープンソースコミュニティマネージャー、Capital One

Christine Abernathy

Open Source Developer Advocate at Facebook

Christine Abernathy

オープンソース開発者アドボケート、Facebook

Guy Martin

Director, Open at Autodesk

Autodesk

Guy Martin

Open at Autodeskディレクター、Autodesk

Chris Aniszczyk

Chris Aniszczyk

COO of CNCF

Chris Aniszczyk

COO、Cloud Native Computing Foundation(CNCF)

Section 1

Life cycle planning for your open source project

セクション　1

オープンソースプロジェクトのライフサイクル計画

As software developers start to envision and develop new and needed business-critical open source projects, they should also incorporate clear and specific plans for each project’s complete life cycle, from its birth to its eventual demise. Such planning should also be conducted as part of a company’s overall open source strategy, incorporating all of the projects it oversees.

ソフトウェア開発者がビジネス上重要な新しいオープンソースプロジェクトを構想し、開発を始めるときは、各プロジェクトの全ライフサイクル、すなわちその誕生から最終的な活動停止までに対する明確で、具体的な計画も組み込んでおかなければなりません。そのような計画作りは、企業の一貫したオープンソース戦略の一部として、その企業が司るすべてのプロジェクトにおいて行われるべきです。

These efforts could mean planning for a future shuttering of a project, a transfer of its operations to another interested group of users, or a corporate pull-out from the initiative. Such end-of-life planning is part of the responsibility that enterprises have for projects, their users, and for the open source community that sustains it all. These plans are necessary to allow such transitions to be conducted gracefully for other users if a project is ended or transferred.

そのようなライフサイクル計画とは、将来的にプロジェクトを閉鎖すること、関心のある他のユーザーグループに運営を移管すること、あるいは、企業として開発活動から撤退することなどに対する計画作りを意味します。この終末計画作りは、プロジェクト、プロジェクトのユーザー、また、プロジェクトを支えるオープンソースコミュニティに対する企業の責任の一部を成します。このような計画は、プロジェクトが終了したり、移管されたりするときに、他のユーザーが適切に移行を行なうために必要なものです。

Why life cycle planning is important

なぜライフサイクル計画が重要なのか

Planning for the potential end of an open source project isn’t unique. Similar life cycle planning is done for proprietary software initiatives, IT systems hardware deployments, and a wide range of other business decisions. For an open source initiative, project life cycle planning could include the traditional life cycle evaluations, such as an outline of the project’s scope and vision and estimates for its growth, as well as open source-specific categories like planning for community building and early user adoption and incorporating user feedback and contributions.

オープンソースプロジェクトが終了するかもしれないということに対する計画作りは決して独特なものではありません。同様のライフサイクル計画は、プロプラエタリーソフトウェアの開発活動でも、ITシステムのハードウェア展開でも、さらには、他の幅広いビジネスの意志決定でも行われます。オープンソースの開発活動に関して言えば、プロジェクトのライフサイクル計画には、プロジェクトのスコープとビジョン、および、成長予測のような従来から行われてきたライフサイクル評価と、コミュニティ構築や早期ユーザーの採用、および、ユーザーのフィードバックや貢献の組み込みに関する計画作りのようなオープンソース固有のカテゴリーが含まれます。

It’s natural for any project to undergo development slowdowns or for user adoption to peak and then decline. Even highly successful open source projects may eventually stop being useful to their creators or users as business plans change or new technologies and innovations replace them. Thus, an exit plan is critical to ensure a smooth transition.

どんなプロジェクトであろうとも、開発がスローダウンにいたることや、ユーザーの採用が盛り上がった後に落ち込むことは自然なことです。大成功を収めたオープンソースプロジェクトでも、最終的には、ビジネス計画が変化したり、あるいは、新技術とイノベーションがそれらにとって代わり、プロジェクトを作った人々にも、そのユーザーにも有用でなくなってしまします。したがって、出口計画は円滑な移行を保証するために不可欠なのです。

Section 2

What does a dead open source project look like?

セクション　2

「死んだ」オープンソースプロジェクトはどんな風に見えるか？

If the steady flow of contributions or commits to your open source project has fallen off, that doesn’t necessarily mean a project is dead. It may simply mean that the project is mature, has achieved its development goals, and is serving its users without requiring much upkeep or updating, which could be a good thing.

順調だったオープンソースプロジェクトに対する貢献数やコミット数が落ち込んだとしても、それは必ずしもプロジェクトが「死んだ」ということを意味するものではありません。それは単に、プロジェクトが成熟に達し、開発の目標を成し遂げ、大きな維持作業や更新作業を要求されることなくユーザーをサポートし、うまく行っている可能性もあります。

On the other hand, if the number of people adopting the project and using the code has declined significantly, that could be a sign that interest is waning and the project may be dying. Other relevant metrics can include the level of project activity in general and whether users are posting inquiries and joining online discussions about the code.

一方で、そのプロジェクトを採用する人、および、コードを利用する人の数が急激に減少しているようだと、関心が弱まり、プロジェクトが死にかけている兆候なのかもしれません。他の有効な指標としては、プロジェクト活動のレベルや、ユーザーが問い合わせを行っているか、コードに関するオンラインの議論に参加しているかといったものも含めることができます。

Trouble signs to watch for

注視すべきトラブルの兆候

A dead or dying project can also be indicated by problems, such as unresolved differences about the direction of development or through lost energy from formerly involved contributors, who may have moved on to other projects that better capture their interest. An obvious sign is a decline in the project that coincides with members of your team or the community asking questions about whether it should continue, be ended, or left outright. Another indicator is that the code is no longer being patched or updated by the community to resolve vulnerabilities.

死んだ、あるいは、死にかけているプロジェクトは、問題の存在によって示されることもあります。たとえば、開発の方向に関する考え方の差を解消できないとか、あるいは、それまで積極的に参加していた貢献者からもたらされるエネルギーが失われたといったケースです。その開発者は、より強い興味を感じて他のプロジェクトへ移って行ったのかもしれません。プロジェクトを継続すべきか、終わらせるべきか、あるいは、そのまま放置しておくべきかといった質問が、あなたのチームやコミュニティのメンバーの間で同時にでてくるプロジェクトは、衰退への明らかな兆候を示しています。他の指標は、コードに対するセキュリティ脆弱性のパッチや更新がコミュニティによって提供されなくなることです。

“If you don’t have anybody asking questions, if nobody’s making contributions, if there doesn’t seem to be any new adoptions, nobody seems to be adding dependencies, and if you’re not seeing any other signs the people are using it, that’s a potential big warning sign. It may all be fine, but it’s worth checking to see if the project is dying.”

「だれも質問を上げなくなったり、だれも貢献しなくなったり、新規にソフトウェアを採用するユーザーがなく、だれも依存性のあるソフトウェアを作らなくなったなら、また、だれかがソフトウェアを利用しているという証拠を見出せなくなったなら、大きな警告灯が点灯している可能性があります。まったく問題ないのかもしれませんが、それでもプロジェクトが死にかけていないかチェックしてみる必要があるでしょう。」

Dr. David A. Wheeler – open source expert and lead for two projects with The Linux Foundation’s Core Infrastructure Initiative (CII)

[Dr. David A. Wheeler](https://www.twitter.com/drdavidawheeler)

オープンソースのエキスパート、Linux Foundationの[Core Infrastructure Initiative (CII)](https://www.coreinfrastructure.org/)で２つのプロジェクトを指導している。

These signs can be tricky to gauge even with hard data, since your code often may be accessed indirectly through package managers for other applications, rather than via direct downloads. This can make it hard to know exactly how many users you have and can accurately track as a project goes through its life cycle.

これらのサインは、しっかりとしたデータがあっても正確な判断を誤る可能性があります。コードは、直接ダウンロードされるのではなく、他のアプリのパッケージマネージャーを経由して間接的にアクセスされているかもしれないからです。これにより、プロジェクトのライフサイクルが進むにつれ、どれくらいの数のユーザーがいるのかを正確に知り、追跡調査するのが難しくなります。

Section 3

Why plan for the end of a project before you even launch it?

セクション　3

プロジェクトの終了計画はなぜ必要か（プロジェクトの始動前においてさえも）？

As you plan your open source project, related decisions about how you might someday end or transition the project are a critical part of the project’s life cycle.

オープンソースプロジェクトの計画作りにおいて、将来的にそのプロジェクトをどう終了、あるいは、移行させるかに関連した意志決定は、プロジェクトのライフサイクル計画の重要な部分となります。

Why is that? Because pulling out of a project without a plan to protect its community and users can substantially harm an organization’s reputation within the open source community and the project itself. If you manage an open source software project, remember that others depend on it and that their continued use of it depends in part on your management and administrative efforts. Certainly your company can choose to change a project’s direction or status, but part of your responsibility is to clearly, quickly, and openly communicate such changes to a project’s users so they can plan for their own needs. It would be irresponsible to close a project and take down its code without communicating those intentions to users.

なぜでしょうか？コミュニティやユーザーを保護する計画なしにプロジェクトから撤退することは、オープンソースコミュニティ、および、プロジェクトにおける組織の評判を大きく損なう可能性があるからです。あなたがオープンソースソフトウェアプロジェクトを管理しているとすると、他の人々がそのプロジェクトに依存し、彼らの安定した利用があなたの管理・監視作業に依存しているということを覚えておいてください。確かに、あなたの会社はプロジェクトの方向やステータスを変更するような選択を行うことができますが、あなたの責任の一端として、ユーザーがニーズに対応した将来計画を立てることができるように、そのような変更をプロジェクトのユーザーに明確、迅速、かつ、オープンにコミュニケーションすることが含まれています。その意図をユーザーとコミュニケーションすることなしに、プロジェクトをクローズし、そのコードを削除するのは無責任な行動となりえます。

Protect your company’s reputation

会社の評判を守る

Leaving other users in a lurch is not something you want to do. In the world of open source, this issue should not be taken lightly. That doesn’t mean your potential exit plans need to be made in detail even before you get the project underway, but frequent, open communication will help assure users that you will not drop a project without adequate notice and that you will work to prevent users from being left hanging. In addition, by building, developing, and promoting code that is easily forkable, your project can offer flexible transfer capabilities to others if you should decide to end your own involvement in the project. Although you need not announce a full disengagement procedure in your life cycle plans, you can still let users know that you intend to close or transition the project smoothly and responsibly.

ユーザーを見捨てることは、あなたの意図するところではないはずです。オープンソースの世界で、これは軽く考えるべきことではありません。だからと言って、プロジェクトの始動前から出口計画を詳細化する必要があるというわけではありません。しかし、頻繁でオープンなコミュニケーションは、あなたが十分な予告なしにプロジェクトをやめることはなく、また、ユーザーが放置されるのを防ぐために行動することをユーザーに保証する助けになります。それに加えて、容易にフォークできるコードを開発、構築し、普及させることによって、あなたがプロジェクトへの関与を止めるときに、他のプロジェクトに柔軟に移行できる道を与えることができます。あなたのライフサイクル計画において、撤退手順のすべてを公表する必要はありませんが、それでも、プロジェクトを円滑に、かつ、責任を持って閉鎖、あるいは、移行する意図があることをユーザーに知らせることができます。

Seek a diverse contributor base

多様性な貢献者を求める

Having a diverse group of code contributors is important to help your project grow by bringing in new ideas, deeper problem-solving capabilities, and more developer eyes on the code. It’s also helpful later if you decide to end or leave your project. With a diverse community involved in the effort, you could have have interested community members who potentially want to maintain it, broadening your options for future transition. If you do decide to leave a project, you would likely prefer to hand it off to other people who care about and will continue it, rather than announce its demise.

コード貢献者が多様性のあるグループで構成されることは、新たなアイデアを集め、より奥深い問題解決能力をもたらし、さらに、より多くの目がコードに注がれるために、プロジェクトの成長を助ける重要な要素となります。また、これは、後であなたがプロジェクトを終了させ、あるいは、プロジェクトから離脱することを意志決定した時にも役立ちます。プロジェクトの活動に多様性のあるコミュニティが関わっていれば、プロジェクトを保守することに関心のあるコミュニティのメンバーが存在しているかもしれないし、将来の移管においてあなたの選択肢を広げます。もしもプロジェクトから本当に離脱する決意をしたなら、あなたはおそらく、プロジェクトの解散を公表するよりは、プロジェクトに気を配り、継続する人の手に渡すことの方が好ましいと思うでしょう。

These early steps lay the groundwork for project success by establishing a level of trust that will allow a healthy ecosystem and commercial dependencies to be built around your project.

このような初期段階の準備は、プロジェクトの周りに健全なエコシステムとビジネス上の依存関係を築くことが可能な信頼レベルを確立することでプロジェクト成功の基礎となります。

“When you’re starting your project, you’re trying to get people to trust you and allay their fears about joining the project and using your code. Later, if you say, ‘Hey, this project’s going to go away soon,’ that is not going to help with trust. Instead, you should say you’re going to do your best to make it work out if it will ever be ended, and that you promise not to just drop users. Tell them you’ll let them know what is happening at each step. Give them time to transition, and work on ways to help with the transition. That can be very helpful.”

「あなたがプロジェクトを開始するとき、人々があなたを信頼し、プロジェクトに参加することやコードを使用することに対する心配を軽くしようとするでしょう。その後で、プロジェクトがもうすぐ無くなってしまうなどと言うと、信頼を損ねることになります。そうではなく、プロジェクトがいつか終了しても、プロジェクトがうまく行くように最大限の努力を尽くし、また、ユーザーを見捨てないことを約束すると言うべきです。ユーザーに対して、何が起きているかを各段階で知らせるつもりであることを伝えてください。また、ユーザーに移行の時間を与え、移行を支援する手段に取り組んでください。それがとても助けになるでしょう。」

Dr. David A. Wheeler – open source expert and lead for two projects with The Linux Foundation’s Core Infrastructure Initiative (CII)

[Dr. David A. Wheeler](https://www.twitter.com/drdavidawheeler)

オープンソースのエキスパート、Linux Foundationの[Core Infrastructure Initiative (CII)](https://www.coreinfrastructure.org/)で２つのプロジェクトを指導している。

Section 4

Deciding when to end, transfer, or pull out of a project

セクション　4

プロジェクトをいつ終了、移管、あるいは、それから撤退するかを決定する

Whatever the reason, there may come to decide whether an open source project should be ended or transferred to another entity or whether your enterprise should end its participation in the project. It could be because your business goals have changed and that the project no longer corresponds with your new direction. Or, maybe it’s because a key person or team that headed the effort leaves the company, or that project performance metrics such as participation, updates, and usage are declining drastically based on your latest user data. It’s also possible that disagreements have arisen among community members about the future of the project and it looks like changes may be needed.

理由が何であれ、オープンソースプロジェクトが終了、ないしは、他の法人へ移管されるべきかどうか、あるいは、あなたの会社がプロジェクトに参加するのを止めるべきかどうかを判断する状況になる可能性があります。その理由は、あなたの会社のビジネス目標が変わり、プロジェクトがもはや会社の新しい方向に合わなくなったからなのかもしれません。あるいは、活動を主導していた主要な人物、ないしは、チームが会社を離れたとか、参加者、ソフトウェアの更新、使用者といったプロジェクトの成果指標が最新データで判断すると急激に落ち込んでいるといったことが理由かもしれません。さらには、コミュニティのメンバーの間でプロジェクトの将来に関して意見の相違が発生し、変化が必要になったといったことが理由となることもあります。

Disagreements can influence new project directions

意見の相違がプロジェクトの方向に影響を与える

If disagreements have arisen about the project inside the community — whether about its scope, technical matters, licensing or other issues — then you might consider splitting the project to allow interested parties to pursue what’s important to them. A similar quandary helped inspire developer Linus Torvalds to create Linux back in 1991. Torvalds had been experimenting with Minix, a small UNIX-like operating system kernel developed by Andrew S. Tanenbaum to teach college students about coding. Torvalds created Linux after he decided Minix had a radically different scope and vision, leaving him to create his own separate operating system.

プロジェクトのスコープとか、技術的な問題とか、あるいは、ライセンスの方式などに関連して、コミュニティ内で意見の相違が発生すると、あなたはプロジェクトを分裂させて、関心のある人々が重要だと考えるものを追求していくことを容認しようと考えるかもしれません。そのような問題が、1991年、Linus TorvaldsのLinux開発を触発しました。Torvaldsは、Andrew S. Tanenbaumが大学生にコーディングを学習させるために開発したUNIX風のOSカーネルMinixを実験的に使っていましたが、Minixが根本的に異なったスコープとビジョンに基づいて開発されていると判断したので、Minixから離れて、彼独自のOSを作りました。

The scope of projects can certainly change over time and be readjusted as needed, but if it the project is simply no longer needed, then it can be a candidate for dissolution or transfer. Projects can even be repurposed to either find a way to make your existing users happy or to incorporate a functionality that needs new users. Ultimately, however, if you have software and nobody needs it, then the project is probably finished. Even if it’s the best software in the world, if nobody needs that capability, then nobody cares.

プロジェクトのスコープは時間が経つと間違いなく変化し、また、必要に応じて調整されることがあります。しかし、プロジェクトがもはや必要でなくなったなら、解散または移管の候補となりえます。プロジェクトは、既存ユーザーを満足させる方策を見出すか、あるいは、新規ユーザーを求めて機能を組み込むかのどちらかを行うために目的を変更することもできます。しかし、究極的に、あなたのソフトウェアをだれも必要としないようなら、プロジェクトは恐らく終わってしまっています。たとえ世界で一番優れたソフトウェアであっても、それを必要とする人がいないと、だれもそのソフトウェアを気にしません。

“That definitely does happen in different ways. We could stop using the project and its code because we just moved on to other things, or maybe the people who are maintaining it are now working on different things or have even left Facebook. Maybe nobody’s there supporting it and we’re also not using it in our own apps. Some projects are contained enough that they’re basically done being used.”

「そのようなことはいろいろな状況で本当に発生します。私たちは、プロジェクトとそのコードを使うのを止めることがありえます。その理由は、私たちが単に他のソフトウェアに移行したからだとか、そのプロジェクトを保守していた人々が他のことに取り組んでいるとか、Facebookを退職しただとかかもしれません。おそらく、そのプロジェクトをサポートする人はいなくなっており、私たちのアプリでも使わなくなっているかもしれません。いくつかのプロジェクトは沈静化してしまっており、基本的に使用されるのを終えています。」

Christine Abernathy – Open Source Team Developer Advocate at Facebook

[Christine Abernathy](https://twitter.com/abernathyca)

オープンソース開発アドボケート、Facebook

Examples from the user trenches

ユーザーの現場における事例

For business users, a wide range of factors affect decisions on whether open source projects should be ended. At software maker Autodesk, where some 190 open source projects are homegrown and in use, declining participation in a project’s development is seen as a clear indicator of code that might be past its prime, which can lead to placing a formerly thriving project into “maintenance mode.” Under that designation, the code is no longer actively maintained and if users have questions they may or may not be answered. In such a case, community members of the project can still use it, but no community support is provided.

ビジネスユーザーにおいて、オープンソースプロジェクトを終了するかどうかの決断にはたくさんの要因があります。ソフトウェアメーカーAutodeskでは、190前後のオープンソースプロジェクトが作られ、使用されていますが、プロジェクト開発への参加者の減少は、コードが全盛の時期を越えてしまったことを明確に示すものみなされ、かつて成功していたプロジェクトでも「保守モード」に置かれることに繋がる可能性があります。そのようになってしまうと、コードはもはや活発に保守されなくなり、ユーザーが質問を上げても回答されたり、されなかったりします。そのようなケースでも、プロジェクトのコミュニティメンバーは、使用を続けることはできますが、コミュニティのサポートは提供されません。

Similar usage indicators, including patch and maintenance requests, are monitored at Facebook, where some 400 open source projects have been homegrown so far, to periodically review the fates of ongoing projects. And at Capital One, where about 20 open source projects have been started in-house and are in use, projects are regularly transferred to other organizations or shut down depending on the needs of the company.

Facebookも同様に使用状況を指標として、パッチ、および、保守要求などをモニターしています。同社はこれまでに400前後のオープンソースプロジェクトを作りだし、進行中のプロジェクトの「運命」を定期的にレビューしています。Capital Oneでは、20前後のオープンソースプロジェクトが開始され、利用されていますが、同社のニーズに従ってプロジェクトは、定期的に他の組織に移管、あるいは、閉鎖されています。

Section 5

How to end an open source project

セクション　5

どのようにオープンソースプロジェクトを終了させるか

Whatever you do to end or transfer an open source project, perhaps the most important move you can make is to be candid with users at every step. Being open about your intentions helps establish trust for future work with all potential developer communities.

オープンソースプロジェクトの終了、あるいは、移管のために何をするにしても、あなたにできる一番重要なことは、すべてのステップにおいてユーザーに率直であることです。あなたの意図に関してオープンであることは、将来の作業に関して、共に活動する可能性のあるすべての開発コミュニティの信頼を確立するのに役立ちます。

Once you’ve decided to make the move, you must determine whether to end it, transfer it to another organization, or just pull out and end your direct involvement.

行動することを決めたなら、プロジェクトを終了させるのか、他の組織に移管するのか、あるいは、単に関与をやめて撤退するのかを決めなければなりません。

How transfers can be helpful

移管することがいかに有効か

By transferring the project to another organization that wants to continue to maintain it, you can then directly assist in the transition of the project’s data and other resources. This approach can help reduce the worry and risks of the existing community and its users. If the project has used common or standard interfaces, it will likely be easier to extract and move embedded data or replace it with another similar component if needed. That won’t always work, of course, because clearly some open source projects are unique. However, providing and using standardized interfaces early in the project life cycle can help make these transition efforts easier.

継続して保守することを希望する他組織にプロジェクトを移管すれば、プロジェクトのデータやその他の資源の移行を直接的に促進することができます。このアプローチにより、現行コミュニティとそのユーザーの懸念、および、リスクを軽減できます。プロジェクトが共通、ないしは、標準のインターフェースを採用していれば、ユーザーの手持ちデータを抽出して移動させたり、必要に応じて他の類似コンポーネントで置換することが容易になるでしょう。もちろん、多くのオープンソースプロジェクトはユニークな機能を提供しているので、そのようなやり方が常に上手くいくとは限りません。それでも、プロジェクトのライフサイクルの早期から共通、ないしは、標準のインターフェースを提供し、採用することは、このような移行処理を容易にするのに役立ちます。

Make needed updates when transferring projects

プロジェクトの移管において、必要な更新を実施する

Transferring a project has the added benefit of continuing to make the code available to other users, long after it was first created and nurtured by its original development community. Of course, more than just code will be affected. There’s also the documentation, Contributor License Agreements (CLAs), the project repository, wikis, websites to transfer, as well as some potential support infrastructure which will need to be reviewed and updated. Transfers can be done through an actual transfer procedure or by forwarding the project domains and other resources to a new user group.

プロジェクトを移管することには、当初の開発コミュニティによってコードが最初に開発され、育て上げられてから長い時間が経ったあとでも、他のユーザーに継続してコードを使用してもらえるというメリットがあります。もちろん、コードだけではなく、もっと多くのことが影響を受けます。移管されるものとして、ドキュメント、貢献者契約（CLAｓ）、プロジェクトリポジトリー、Wiki、Webサイトもあり、また、後にサポートのインフラとして使用されるものもありますが、そのシステムは見直され、更新される必要があるかもしれません。移管は、通常の移管手続きを通して行うこともできますが、プロジェクトドメインやその他の資源を新しいユーザーグループにフォワード（転送）することによって行うこともできます。

In lieu of transferral, the project can be moved to a new host home and accessed by sending community members to a referral site where it is continued, rather than completely disengaging from it.

移管する代わりに、プロジェクトを新たなホストシステムに移動させ、コミュニティのメンバーを照会サイトに送ることによってアクセスさせて、プロジェクトへのアクセスを完全に遮断することなく、そこで継続することができます。

“It doesn’t happen all the time, but in the past with one of our projects we moved it over to a different company. We don’t have any hard and fast rules about doing this. Typically, we’ll just move it to a different organization. When it comes to moving within groups, we sort of shop around internally and find out whether it is still being used by someone. With our open source projects, we strive toward internal adoption. So, it might be used by an entirely different team. If they are willing to maintain it, then we move it to a different team, and that’s very easy. That just means changing a label somewhere where it says who’s the owner.”

「いつもそうなる訳ではありませんが、過去に、私たちのプロジェクトのひとつを他の会社に移したことがあります。それを行うための厳格なルールはありません。通常は、単に他の組織に移すだけです。私たちのグループの間で移すとなると、私たちは社内を調べて、まだ誰かに使われていないか探します。私たちのオープンソースプロジェクトについては、社内の採用を目指しています。ですから、プロジェクトはまったく異なったチームで利用されているかもしれません。そのような人々が喜んで保守してくれるなら、そのチームに移管します。これなら簡単です。単に、所有者を示すラベルを変えるだけですから。」

Christine Abernathy – Open Source Team Developer Advocate at Facebook

[Christine Abernathy](https://twitter.com/abernathyca)

オープンソース開発者アドボケート、Facebook

Killing a project outright

プロジェクトをすぐさま消滅させる

Some organizations may simply want to kill an outdated project and retire its efforts for a myriad of reasons, from legal concerns to competitive pressures. The direction your enterprise ultimately takes is up to you. In this case, the code doesn’t need to be removed but can instead be archived, or made “read-only,” so it may still be used by others to fork it into new projects without your involvement. It can also be moved to an archiving site where it can be stored for users.

組織によっては、いろいろな理由、たとえば法務的な懸念や競合上の圧力を考慮して、単純に時代遅れになったプロジェクトを消滅させ、活動から撤退したいと考えるかもしれません。あなたの会社が打ち出す方向性は、ひとえにあなた次第です。この場合、コードは消去される必要はなく、その代わりに、アーカイブにするとか、「リードオンリー」とし、なおも他の人々によって利用されて、あなたの関与なしに、新プロジェクトにフォークされるといった可能性もあります。また、アーカイブサービスを提供するサイトに移されて、そこでユーザーのために保存されることもあります。

It’s important to always remember that just because your company doesn’t care about the code anymore it doesn’t mean that the existing community doesn’t care. It behooves you, then, to remind them as you proceed that they can fork the code and create their own uses for it without you. This also wouldn’t be a bad time to remind users who depend upon the open source code in your projects that they would be wise to mirror and save the code on their own for projects that are very important to them. Code can disappear over time and a good backup or mirror can be critical for users.

あなたの会社がコードのことを気にしなくなったからといって、既存コミュニティが気にしないということを意味するものではないということを常に覚えておくことが大切です。したがって、計画の進展とともに、コミュニティの人々がコードをフォークし、あなたの関与なしにそれを利用できるということを知らせることはあなたの義務なのです。その際、あなたのプロジェクトのオープンソースコードに依存しているユーザーに向けて、重要なプロジェクトについてユーザー自身でコードのミラーサイトを作ったり、セーブしたりすることが賢明な措置であることを再認識させるのがよいでしょう。コードはやがてなくなってしまう可能性があり、ユーザーにとって、バックアップやミラーは必須の処置です。

To make whichever option you choose work efficiently, a step-by-step disengagement plan is a good idea to keep it all organized. That means clear and early communications with users who are likely to care as well as a sufficient timeline that gives affected users time to finish their work and move to an alternate platform if needed. How much time will be enough for users? It depends. Start with at least six months, or it could be longer. The key is to set reasonable expectations and communicate your plans very clearly and quite often.

あなたの選択がどんなものであろうと、それを効率よく進めるためには、着実な撤退計画が混乱を防ぐ良い考えです。それは、影響を受けそうなユーザーへの明確で早期のコミュニケーション、および、余裕のあるスケジュールを意味し、それによってユーザーは仕掛中の作業を完了させ、また、必要なら代替プラットフォームへ移行するための時間を得ます。ユーザーは、どれくらいの時間があれば十分なのでしょうか？それは状況によって異なります。少なくとも6ヶ月前、あるいは、もっと早くから始めてください。鍵は、妥当な見通しを提示し、あなたの計画について明確に、かつ、頻繁にコミュニケーションすることです。

If your enterprise transfers the project to another group, the users should be kept up to date on what is being done to protect their interests as well.

あなたの会社がプロジェクトを他のグループに移管するのなら、ユーザーの利益を保護するために何が行われるのかについて最新の状況を伝えるべきです。

“The best practice is to just be upfront and clear with the state of your projects. If you’re no longer supporting a project, or you’re in the process of winding it down, make that painfully obvious to anybody that would stumble across your code. You want them to see it’s no longer being maintained so new users don’t start using it and have this unwritten expectation that it is out there and that you are sponsoring this project and developing it.'”

「最善の方法は、あなたのプロジェクトの状況の今後について、明確にすることです。もはやあなたがプロジェクトをサポートせず、あるいは、終了させる過程にあるのなら、あなたのコードに偶然見つけた人たちに、その状況がはっきりと分かるようにしてください。そのような人たちにコードが保守されていないことが分かり、新規ユーザーが使用し始めることなく、また、コードがそこにあって、あなたがスポンサーとなっており、開発を続けているというような『書かれざる期待』を持たないでしょう。」

Jared Smith – Open Source Community Manager at Capital One

[Jared Smith](https://twitter.com/jsmith)

オープンソースコミュニティマネージャー、Capital One

Work toward a smooth transition

円滑な移行に向けた作業

When a workable partner is secured, then approach your project disengagement like you would handle any product that is being discontinued. Decide what, if any, involvement your organization will have with the project going forward in terms of developer time and then work closely with the new entity to prepare for a smooth transition. Be sure to set expectations that the actual transfer could take months to accomplish, and then turn the project over officially to the new owners when all the needed steps are completed.

作業できるパートナーが確保できるなら、あなたのプロジェクト撤退アプローチは、中止されるどの製品にも対応できます。まず、当該プロジェクトに対して、あなたの組織は、特に開発者の作業時間の観点で、どんなことができるのかを決定し、そして、円滑な移行に備えて新しい組織と緊密に作業してください。実際の移管の完了までに数ケ月を要するという見通しを必ず明らかにし、その後すべての必要な手順が完了したときに、新しい所有者にプロジェクトを公式に移してください。

A potential problem that could crop up, though, is if the group which is interested in taking over your old project is not one you favor to operate it into the future. This can happen, and you will have to come up with a plan for action in such a case. How do you deal with this? Well, if you’re so attached to your code that you are considering not transferring it to a group that is not exactly in line with your corporate or philosophical directions, then maybe it’s not the right time to end the project. If you care about it that much, perhaps there’s a reason that you shouldn’t be leaving this project.

それにもかかわらず、発生するかもしれない問題は、あなたの旧プロジェクトを受け取るのに関心のあるグループが、将来の運用を託すのに好ましくないものだったケースです。このようなことは起こりえます。あなたは、行動計画を立てなければならなくなります。このような状況にどう対応しますか？あなたのコードに対するこだわりが強くて、あなたの会社としての、あるいは、理念としての方向性と相容れないような他グループへの移管を取りやめることを検討するなら、このタイミングはプロジェクト終了の正しい時ではないのかもしれません。そんなに強く気になるとすると、おそらく、そのプロジェクトから離脱すべきでない何らかの理由があるのでしょう。

The importance of clear communications

明確なコミュニケーションの重要性

Clearly communicating the new direction of the transferred, ended, or rebranded project will continue to be your responsibility even after the transition. Be sure to openly advise developers and community members about the status of the project with its new operators and give them clear details about how the project is now being maintained if they care to continue to use it. Don’t forget to update your project website, social media channels, and other connected community assets so participants know where to find the moved project and can continue to make their contributions and use the code.

移管、終了、あるいは、名称変更されるプロジェクトの新たな方向性について、を明確にコミュニケーションを取ることは、移行の後であっても、引き続きあなたの責任です。新たな運用者と共にプロジェクトの状況について、開発者とコミュニティのメンバーに率直に知らせ、また、彼らが継続して利用するようなら、今後どのようにプロジェクトが保守されるかについての詳細を伝えてください。また、移動後に参加者がプロジェクトの場所を見つけ、また、貢献やコード利用を継続できるようにプロジェクトのWebサイト、ソーシャルメディアチャンネル、さらには、ネットワークに繋がったコミュニティ資産を更新するのを忘れないでください。

Remember to communicate all this information to downstream organizations and users as well. This includes companies, non-profits, and others that are using your code, and who may not be aware of the demise or transfer of the project because they’re not directly involved in the development cycle. You should make efforts to advise them of what is happening through your websites, social media channels, and other outlets, especially if the project is well-known and has had a substantial following.

これらすべての情報を下流の組織とユーザーにもコミュニケーションすることを覚えておいてください。その対象には、企業、非営利組織、コードを利用している人々や、開発サイクルに直接関わっていないためにプロジェクトの解散や移管に気付かない人々も含まれます。特に、プロジェクトが良く知られていて、相当数の支持者があるときには、Webサイト、ソーシャルメディアチャンネル、その他の手段を通じて、何が起きているかについて知らせるよう努力すべきです。

“The biggest thing is to communicate the changes and tell the community of the plans ASAP. Don’t leave the users wondering if the project is still being maintained. For the most part, good code lives on, in my opinion, so if you’ve got a project that has good code, I would rather not ever take something down from GitHub but set the expectations at, ‘Hey, this project is no longer being actively maintained.'”

「一番再説なことは、変更についてコミュニケーションし、コミュニティにその計画のことをできるだけ早く、説明することです。プロジェクトが保守されるのかどうかについて、ユーザーを困惑させてはいけません。私の意見ですが、大体において良いコードは生き続けます。ですから、あなたが良いコードのプロジェクトを運営していたとすると、決してGitHubから何かを取り下げたりしないで、ただし、『このプロジェクトは今後、積極的には保守されませんよ。』と言う見通しを示してください。」

Guy Martin – Director of Open at Autodesk

Provide updates through project repositories

[Guy Martin](https://twitter.com/guyma)

Open at Autodeskディレクター、Autodesk

Another important place to post information about project changes is the project’s repository itself on GitHub or related destinations. There you can place detailed notes explaining what’s happening, including readme files to help get the messages to participants.

プロジェクトの変更に関する情報を書き込むべき重要な場所は、GitHub上のプロジェクトのリポジトリーやその関連の場所です。参加者にメッセージが届くように、Readmeファイルなどで、何が起きているのかを説明する詳細な情報を提供します。

As the project moves on, it’s time to officially transfer the assets, if necessary, to its new operators. If you are bringing in new administrators while maintaining the actual assets within your existing company, you can keep the copyrights on the affected code and let them use it with the open source licenses of their choice.

移管計画の進行にともない、新たな運用者へ公式に必要な資産を移管する時がきます。実際の資産をいままでの会社に置いたまま、新たな管理者を迎え入れるのなら、移管の影響を受けるコードの著作権を保持したまま、新たな運用者の選択するオープンソースライセンスのもとで使用させることができます。

To keep the project repositories well organized, you might consider setting up a designated area where you can store and make available all transitioned open source projects that your company no longer supports (e.g., https://github.com/twitter-archive). In this way, the projects are still available for users to get code, and potential users can also still find readme files explaining the status of the projects. This designated section of the repository can provide clarity for users and help them understand the difference between active and retired projects, while also ensuring that enterprises can properly showcase their latest live open source projects in their active repositories.

プロジェクトのリポジトリーを適切に整理するために、特定のエリアを設定し、社内でサポートされていないすべての移行済みオープンソースプロジェクト（たとえば、<https://github.com/twitter-archive>）を保存して利用できるようにすることを検討してください。　このやり方によって、当該プロジェクトは、コードを入手しようとするユーザー向けに利用可能となり、また、初めてアクセスするユーザーもプロジェクトの状況を説明するReadmeファイルを見つけることができるようになります。このリポジトリーの特別な領域は、ユーザーに活動中のプロジェクトと退役プロジェクトの違いを明確に理解する助けとなり、同時に、活動中のリポジトリーに置かれた最新のオープンソースプロジェクトを適切に企業のショウケースにすることができます。

Archiving tips

アーカイブ作成のコツ

Several steps are involved in archiving a project. For example, moving it to a read-only status helps make it clear, without having to change any URLs, that the project is now archived and no longer open to regular updates in its previous form.

プロジェクトのアーカイブ作成にはいくつかのステップがあります。たとえば、プロジェクトをリードオンリーに変えることは、URLを変えることなく、プロジェクトがアーカイブ状態になったこと、および、以前と同じ形での定常的な更新を受け付ないことを明確化する助けになります。

Also important is to provide users with a clear alternative plan for their projects, including how to get and fork the code for their ongoing uses. As part of that responsibility, you should provide users with a way to contact you so they can have their fork listed and made available to other interested users. Some companies provide this assistance, while other choose not to do so.

また、たとえば、継続して利用するためにコードを入手し、フォークする方法を示すなど、ユーザーにプロジェクトに関する明確な代替計画を提示することも重要です。そのような責任の一部として、ユーザーに対してあなたへの連絡方法を提示するべきです。それによって、ユーザーは、彼らが作成したフォークを一覧にして、他の関心あるユーザーに利用させることができるようになります。このような支援策を提供する会社もあれば、しない会社もあります。

Once a repository is archived, you cannot add or remove collaborators or teams, and issues become read-only. To make changes in an archived repository, you must unarchive the repository first. See the GitHub documentation for more details: <https://help.github.com/articles/about-archiving-repositories/>

リポジトリーがアーカイブされると、GitHub上で、コラボレーターやチームを追加したり、削除したりできなくなり、イシューはリードオンリーになります。アーカイブ状態のリポジトリーに修正を行うには、まずリポジトリーを「アンアーカイブ」しなければなりません。詳細は以下のGitHubドキュメントを参照してください。

<https://help.github.com/articles/about-archiving-repositories/>

Infrastructure updates when shuttering projects

プロジェクト閉鎖時のインフラ更新

Ending a project can also affect your project’s infrastructure and support, which must be judged on a case-by-case basis, depending on how your project was set up. Some projects use and maintain only a small set of code, so there might not be much to do. If it is all code and it’s posted on your repository, then you’re probably done and need not take additional steps to transfer or secure it.

プロジェクトの終了は、プロジェクトのインフラストラクチャとサポートにも影響を与える可能性があります。プロジェクトの構成、構築方法によっては、ケースバイケースで判断する必要があります。プロジェクトによっては、コードの内の少量部分のみを使用させ、保守していることもありますが、そのような場合はあまり大変ではないでしょう。また、プロジェクトがすべてコードでできており、すべてリポジトリーに投稿されているなら、プロジェクトの移管や保全にさらなる処置を採ることなく、おそらくなすべきことを完了しています。

Other projects, however, may require significant effort to shutter them using various backend tools to do the work. These may include services that may have been shared for many purposes, such as a test infrastructure. Although that test infrastructure has been used for the project previously, you may want to remove it before transferring the code so you can use the tool to test other work later. It may be hard to disentangle, but it can be done if desired.

しかし、作業のための各種のバックエンドツールを使っているようなプロジェクトでは、閉鎖の前にかなりの骨折りが必要となります。それらの中には、テストインフラのように、たくさんの用途のために共用してきたサービスも含まれる可能性があります。そのようなテストインフラは、プロジェクトで使われていたものかもしれませんが、他のソフトウェアのテストツールとして使うことができるように、そのコードは移管から除外したいと思うでしょう。このような問題を解決するのは少し難しいかもしれませんが、望めば実行可能です。

“Companies really need to do a better job of succession planning in open source. Just throwing something out there without proper planning and getting people from outside the company to become contributors and then expecting that it’s going to be that way forever is a challenge. You need to prepare that members of the community, whether they’re from your company or from other places, are going to come in and follow somewhat of a life cycle and eventually, people become elders or retired from a community, and you need to have succession planning for that.”

「企業は、オープンソースにおけるサクセッション・プラン（後継者育成計画）の作成で、本当に良い仕事をする必要があります。妥当な計画を作ることなく何かを放り投げ、会社の外から人々をかき集めて貢献者にしてしまい、その後で、プロジェクトは永久にこんな方向になると予想されますなどと言うのは、無茶です。会社の内外に関わらずコミュニティのメンバーがプロジェクトに関わり始め、人生の一部をそれに捧げ、最終的にはコミュニティの先輩となって引退することに対して準備を行い、そのようなことに対するサクセッション・プランを作ることが必要です。」

Guy Martin – Director of Open at Autodesk

[Guy Martin](https://twitter.com/guyma)

Open at Autodeskディレクター、Autodesk

Have a tough skin

強い心を持つ

There may also be some negative public comments and reactions posted about your actions, including unhappy users or “trolls” who disparage your move as abandoning users. Remember, however, most users are aware that priorities change over time and that enterprises might eventually have to stop some open source projects as their work matures. Every company must do this sometimes, so don’t take the comments too personally.

あなたのやることに関して、ユーザーを見捨てるものだと非難する不満ユーザーや、「トロール（荒らし）」など、否定的なコメントやリアクションが公けに書き込まれることもあるかもしれません。しかし、大部分のユーザーは、時を経て起こる優先度の変化や、大企業でもプロジェクトが成熟すると、最後にはオープンソースプロジェクトを停止させる可能性のあることを承知していることを忘れないでください。どんな企業でも時にはこれを実行しなければなりません。コメントを個人攻撃だと捉えないでください。

Section 6

Final Words

セクション　6

結論

Shuttering, transferring, or leaving an open source project is inevitably a step your enterprise will choose to take at some point, but it need not be a debilitating task. With proper planning, clear and widespread communications and the step-by-step completion of legal and procedural tasks, transitioning an open source project can be accomplished satisfactorily for all involved.

オープンソースプロジェクトの閉鎖、移管、あるいは、離脱は、ある時点において企業が不可避的に選択するステップとなりますが、必ずしもつまらない仕事とは限りません。しっかりとした計画作り、明確で広く行きわたるコミュニケーション、さらには、法務的な業務や手続きを着実にこなすことによって、オープンソースプロジェクトの移行は、関連するすべての人々に満足をもたらしてやり遂げることができます。

By developing and including simple plans to shutter an open source project even as you begin planning a fledgling project’s first steps, you will ensure a healthy life cycle management plan. These plans will help your open source projects run as efficiently and successfully as possible from their lofty starts to their eventual quieter endings.

プロジェクトの最初のステップの計画作りにおいて、オープンソースプロジェクトの閉鎖のための明確でわかりやすい計画を作成し、保持することは、プロジェクトの健全なライフサイクル管理計画を保証することになります。これらの計画は、高い目標を持ったプロジェクトの開始時点から、より静穏な終了時点まで、あなたのオープンソースプロジェクトを可能なかぎり効率的に運用し、成功するための助けとなるでしょう。

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

これらのリソースは、TODO (Talk Openly, Develop Openly) Groupとの協力によって作成されました。TODOグループは、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。本ガイド集のために時間と識見を貢献してくれたオープンソース プログラムマネージャーの皆様に特に謝意を表します。本ガイドの作成に参加してくれた企業は、Autodesk、Comcast、Dropbox、Facebook、Google、Intel、Microsoft、Netflix、Oath (Yahoo + AOL)、Red Hat、Salesforce、Samsung、VMwareの各社です。詳細は、[todogroup.org](http://todogroup.org/)をご参照ください。

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

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