

# 企業でのOSS活用とOSSライセンス

オープンソースソフトウェア（以下、OSS）の普及が進み、さまざまな案件でOSSと関わる機会が増えてきています。しかしながら、OSSのライセンスをきちんと理解しないまま利用すると重大なライセンス違反を招き、後に問題になる可能性があります。OSSを利用する前に改めてライセンスについて理解を深めるとともに、それを踏まえて企業でOSSを活用する際のポイントを解説します。加えて、OSS同様に近年注目が高まっているオープンデータに関する動きについても紹介します。

## ▶▶ 1. OSSライセンスを理解することがなぜ重要か

今や、業界・業種を問わず、さまざまなITシステムでOSSが活用されています。Linux OSなどの基本ソフトウェアからデータベース、アプリケーションといった幅広い分野でOSSが存在しています。レッドハットの最新の調査「State of Enterprise Open Source」によると、95%の企業が「オープンソースはソフトウェア戦略上で重要なものになっている」と回答しています[1]。OSSの利用が広がり、OSSを全く使わないシステムは考えられないと言っても過言ではないでしょう。そしてオープンなソースコードが充実してきたことにより、ビジネスでのOSS活用は今後いっそう増えていくと予想されます。

しかしながら、これらのOSSを間違った使い方で活用してしまうと、予期せぬライセンス違反による起訴などの重大なトラブルに発展してしまうケースもあります。本来であれば外部に公開したくない自社で開発したソフトウェアのソースコードを公開しなければならなくなった、ライセンス違反でFree Software Foundation（以下、FSF）に起訴されたなど、多くの望ましくない事例が実際に起きています。特に多くのメンバー、社員や外部パートナー、協力企業などで製品やサービスを開発し提供している場合、どこでどのようなオープンソースが使われているかをきちんと理解し、製品に関わるメンバー、

および外部パートナー企業も含めて徹底することが重要になります。

また、企業が提供するサービスにおけるOSS利用について以前は特に制限がありませんでしたが、企業がOSSを活用するだけでオープンソースの発展にあまり寄与していないことに対してオープンソース・コミュニティから不満が上がり、OSSの開発元が商用利用を制限するライセンスに変更するといったことも出てきています[2]。ライセンス条件に則って使用していれば大丈夫ではなく、企業の開発者がオープンソース・コミュニティにどのように貢献できるかも考慮しながらOSSを発展させていくことが重要になります。

## ▶▶ 2. OSSライセンスの許諾条件と代表的なOSSライセンス

OSSのライセンスを紹介する前に、いくつかの重要な点について述べておきます。

### 2-1. OSSの定義

OSSとは、ソースコードが無償で公開されており、誰でも複製・配布・改良できるソフトウェアのことであるとFSFが定義しています[3]。有名なOSSの例としては、例えばWebサーバーとして使われているApache、データベースのMySQL、OSとしてはRed Hat Enterprise Linux(RHEL)、CentOS、Ubuntuなどがあります。

OSSの対義語は、プロプライエタリー・ソフトウェアです。MicrosoftのWindowsやOfficeなどの市販ソフトウェアがそれに当たります。ソースコードが公開されておらず、自由に複製、改良などはできません。

OSSとフリーソフトウェアを混同しているケースを見かけますが、フリーソフトウェアは主にソースコードが公開されていないプロプライエタリー・ソフトウェアで、かつ無償で提供されるソフトウェアのことを指します。

なお、OSSはすべて「無料(フリー)」というわけではなく有料の場合もあります。多くの場合は技術サポートをつけて有料で提供しています。レッドハットのLinux OS「Red Hat Enterprise Linux」がその一例です。なおレッドハットは技術的サポートだけではなく、オープンソース導入、開発での法的な保証も行っています。

## 2-2. OSSライセンスの許諾条件

ここではOSSライセンスの許諾条件について説明します。

### ●著作権の表示

OSSを配布する際にはOSSに含まれていた著作権表示などを残しておく必要があります。

### ●免責事項(無保証)

OSSは利用や再配布、改変などがOSSライセンスに従い自由に行われるため、ソフトウェアの動作やバグの除去、そのソフトウェアを利用したことにより生じた損害等についてはなんら責任を負わないと定められています。

### ●コピーレフト

コピーレフト(copyleft)とは著作権(コピーライト:copyright)に対する考え方です。著作権を保持したままで二次的著作物(派生品)、組み合わせて使うソフトウェアなども含め、すべてのものがそれらの利用、再配布、改変を可能にするために作られています。コピーレフトでは、そのソフトウェアを再配布する人は、ソフトウェアを変更してもしなくても、それをコピーし変更を加える自由と一緒に渡さなければならないということを主張しています。コピーレフトによって、すべての利用者が自由を持つことが保証されるといった考えから作ら

れました。

### ●開示の義務

各ライセンスによってコピーレフトの考えをどこまで適用するかが異なります。改変した場合にはどのような改変を行ったか、ソースコードを提供しなければならないか、派生品と一緒に組み合わせたソフトウェアについての取り扱いはどうなるかなどです。次章で紹介する代表的な各ライセンスにおけるこれらの対応については表1を参照してください。

## 2-3. 代表的なOSSライセンス

ここでは代表的なOSSライセンスを紹介します。

### ●GPL(General Public License)

GPLはフリーソフトウェアの普及を目的とするFSFの理念に基づいたソフトウェア・ライセンスです。GNUプロジェクトで開発されたソフトウェア、およびその派生品に使われ、派生物にも自由を保証させようとしています。従ってGPLのコードを含むソフトウェアはGPLで配布しなければなりません。コピーレフト型ライセンスの代表的存在です[4]。

GPL v3では、1991年にリリースされたv2であいまいだった特許に関する部分などを明確化しています。例えば特許について、そのコードを利用した第三者を訴える権利の放棄を要求しています。特許そのものを放棄する必要はありませんが、公開したコードは自由に使うことが求められているため、その利用者に対して特許の使用料などを請求できないことになっています。主なOSSとして、Linuxカーネルがあります。

### ●LGPL(GNU Lesser General Public License)

GPLライセンスの制約を緩めたものです。GPLではGPLライセンスのソフトウェアと一緒に使われるソフトウェアにもGPLライセンスであることが要求されます。しかしLGPLでは、LGPLのライセンスであるプログラムを動的リンクなどで呼び出す場合、その呼び出す元のソフトウェアに関してはLGPLであることは要求されません。従って、動的ライブラリー(Microsoft Windowsオペレーティング・システムでは.dllファイルなど)がこのライセンスを使用する例が多いです[5]。

## ●Apache ライセンス 2.0

Apacheソフトウェア財団 (ASF) によるソフトウェア向けライセンス規定です[6]。この団体は自分たちの開発したソフトウェアを不特定多数の人たちに無償で提供することを目的にライセンスが決められています。ソフトウェア以外にもフォントやWordPressのテーマやプラグインなどにも使われています。

他のOSSライセンス同様、著作権表示と免責事項表示の保持を求めています。Apacheライセンスで提供されるOSSを改変して新たな派生品を作成した場合は、その派生品に関しては異なるライセンス条件を適用することができます。しかし、修正箇所は特許の適用や商標の利用についての説明や制限がかけられています。企業にとってより使いやすいライセンスです。

## ●MITライセンス

マサチューセッツ工科大学 (MIT) が開発したGUIシステムであるX Window Systemに適用するためにこのMITライセンスを作成しました[7]。Apacheライセンス同様、元のソフトウェアに修正を加えた派生品に関してはコピーレフトの対象ではなく、ソース提供の義務はありません。変更箇所の明示も必要ありません。

表1は、代表的なOSSライセンスにおける許諾条件を整理したものです。これらのOSSライセンスの中では、Apache、MITライセンスがビジネス利用において制約が比較的少ないOSSライセンスだと言えます。

なお、GitHubが2015年に発表した情報によると、OSSライセンスの利用率のトップ5は、1位「MIT」

(44.69%)、2位「その他」(15.68%)、3位「GPL v2」(12.96%)、4位「Apache」(11.19%)、5位「GPL v3」(8.88%)となっています[8]。米国Blackduck社の調査では、2018年時点、1位「MIT」(38%)、2位「GPL v2」(14%)、3位「Apache 2.0」(13%)、4位「ISC」(10%)、5位「GPL v3」(6%)となっています[9]。

## ▶▶ 3. 企業でOSS活用に取り組むためには

### 3-1. 企業でのOSS活用のポイント

ここからは企業でOSSを活用する際に考慮すべき点について紹介します。

#### ①OSS利用ポリシーの策定

前章までに述べたライセンスOSSをどのように活用していくか、また著作権についてどのように扱うか(例えば就業中の従業員が作成したOSSをどのように扱うか)など、OSS調達や利用のポリシーを決める必要があります。

#### ②OSS利用ガイドラインの作成

ライセンス概要とポリシーだけでは開発現場での正しいOSS活用は徹底されません。それぞれの開発現場の状況に即した具体的な詳細ガイドラインを作成する必要があります。

#### ③教育の実施

ポリシーやガイドラインが策定されても、従業員が理解していなければ事故が起こります。OSSに関わる可能性がある人全員を対象にトレーニングを実施するべきでしょう。例えば、全従業員向けに実施されるインテグリティなどの法的研修と同じように、OSSを利用す

表1. 代表的なOSSライセンスにおける許諾条件

	GPL	LGPL	Apache	MIT
著作権の表示	必須	必須	必須	必須
変更箇所の表示	必須	必須	必須	必須ではない
ライブラリーでの使用	必須でない	必須	必須でない	必須でない
ソースコードの開示	必須	必須	必須でない	必須でない
特許の利用	可	可	可	記述なし
派生品の扱い(他のライセンス)	禁止	可	可	可



る際は年1回のOSSガイドラインに関するトレーニング受講を必須条件にすることなどが考えられます。

#### ④開発委託時の注意

大規模な開発では開発の一部を委託することがあります。その際、委託先の企業がポリシー・ガイドラインに沿っているかを確認する必要があります。

#### ⑤ソフトウェアの検査

最近では、製品などにOSSを意図せず利用していないか検証してくれるソフトウェアが出てきています。それらを使って検査しておくことも必要です。

上記の項目を達成するための体制の構築も必要となってきます。ポリシーやガイドラインの適用時に疑問があった場合の問い合わせ先も必要となってきます。関連会社がそれらをきちんと理解するために、策定したポリシーやガイドラインを公開しておくことも有効です。

IBMでは、従業員向けに「Open Source@IBM」という内部向けサイトを作成し、OSS活用に関するポリシーやガイドライン、問い合わせ先が確認できるようにしています。また、外部向けには同じくOpen Source@IBMという名称のサイト[10]を公開し、IBMのOSS活用に関するポリシーのほか、IBMのOSSに関する専門家による知見、オープンソース・コミュニティでの活動やOSSを活用した社会貢献活動など、IBMのOSSに対する取り組みを紹介しています。

### 3-2. 製品に組み込む場合などの注意事項

ポリシーやガイドラインなどを策定した後、成果物が出来上がった際には、最終確認として以下のようなチェック項目を設けておくといでしょう。なおプロジェクトによっては、OSSはメインで使っていないくてもなんらかの形でOSSが使われている可能性もあります。成果物としてプログラムのソースコードが入っている場合には確認が必要です。

- その製品、成果物に含まれているプログラムはすべて自社製か。
- OSSや他社、他人のプログラムが含まれているかどうかツールで確認したか。
- 上記で「いいえ (OSSを使用している)」の場合、OSS利用のためのライセンスは守っているか。
- コピーレフト型のOSSライセンスはプログラムのソー

スコードを開示しているか。

- コードの開示が必要ないライセンスでも変更箇所を明示する必要がある場合がある。変更箇所を明示しているか。

### 3-3. 企業でのOSSへの貢献

近年、企業がオープンソース・コミュニティになんらかの貢献をすることは、そのソフトウェアを改良するためだけではなく、その企業が持つ技術力を世の中に周知したり、OSSのようなエコシステムをサポートしていることの証明を目的として、従業員のオープンソース・コミュニティへの参加を推奨する企業が増えています。ライセンス、特許、著作権などさまざまな考慮事項はありますが、それらを理解して貢献していくことが企業にとっても重要になるでしょう。

## ▶▶ 4. アップストリーム・ファースト

OSSの改良や機能追加に対する考え方として、“アップストリーム・ファースト”があります。企業で修正、機能追加したソフトウェアを自社内だけで改良し続けていくのではなく、コミュニティと一緒に改良、そして成長させていこうという考え方です。図1でその違いを説明します。図1の上の図が元々コミュニティ版であったOSSのある時点で自社製品版として開発を進める場合、下の図が自社の改良も一度コミュニティ版に追加してからコミュニティの最新リリース版をベースに製品開発を進めていくやり方です。

使えそうなOSSがあるが自社にとって足りない機能などがあった場合、その部分だけを自社開発して進めるのがスピーディーで競争力も発揮できそうですが、それと同等、場合によってはそれ以上の改良がアップストリーム版で起こる可能性は大いにあります。特にコミュニティが活発なOSSではバージョンが1つ上がるごとに数千から数万行のコードが変更されます。自社製品にそれを取り入れて追従していこうとするとかなりの作業が必要となってきてしまいます。もし機能の追加は諦めたとしても脆弱性などがあった場合にそれらを自社だけで改良していくのは限界があります。

一方で、アップストリーム・ファーストの考えでは、

まずは自社の改良もコミュニティ版として開発還元するということが基本になっています。その際、他のコミュニティあるいは他者との調整などが必要になってきますが、コミュニティ版と自社製品の最終版の差異は小さくなり、後々の変更を少なくすることができるため、トータルでの作業コストを大幅に少なくすることができます。またコミュニティ版に改良を加えることでより多くの人々が恩恵を受けることになり、そのソフトウェア自体の価値が上がり世の中での採用が進んでいくでしょう。またそれらを積極的に改良していった会社ということで、ユーザーおよびコミュニティからの支持も増えるでしょう。まさにコミュニティとともにOSSを成長させていくという考え方です。OSSで成功している会社はこのアップストリーム・ファーストを重要視している傾向が強いです。

こうした理由により、今後は企業がアップストリーム・ファーストを実施しコミュニティと連動した開発を進めていくことが、ユーザーに最大のメリットをもたらすOSSの健全な発展につながります。今後OSSに関わる方はぜひコミュニティへの貢献を意識して、OSSの利用だけではなく発展を目指しましょう。

## 5. オープンデータの活用と動向

OSS活用と同様に、近年オープンデータ活用についての注目も高まっています。行政を中心に、より透明性を高め国民、市民の行政との協働を推進するオープン・ガバメントの流れから、公共データの活用促進、AI開発や機械学習に使えるオープンデータへの関心が高まりつつあります。この流れを受けて各団体、企業によりオープンなデータがより活用できるような取り組みがここ数年盛んになりつつあります。最後にこの動きについて紹介します。

### 5-1. Community Data License Agreement (CDLA)

OSSと同じように、データをオープンに共有・使用するためのデータ・ライセンス・アグリーメントである「Community Data License Agreement (CDLA)」が、2017年にThe Linux Foundationより発表されました[11]。CDLAはThe Linux Foundationがホストするプロジェクトにおいて共同開発され、OSSがソフトウェアのプログラミングに改革をもたらしたように、データに関しても同じような発展をもたらせないかとい

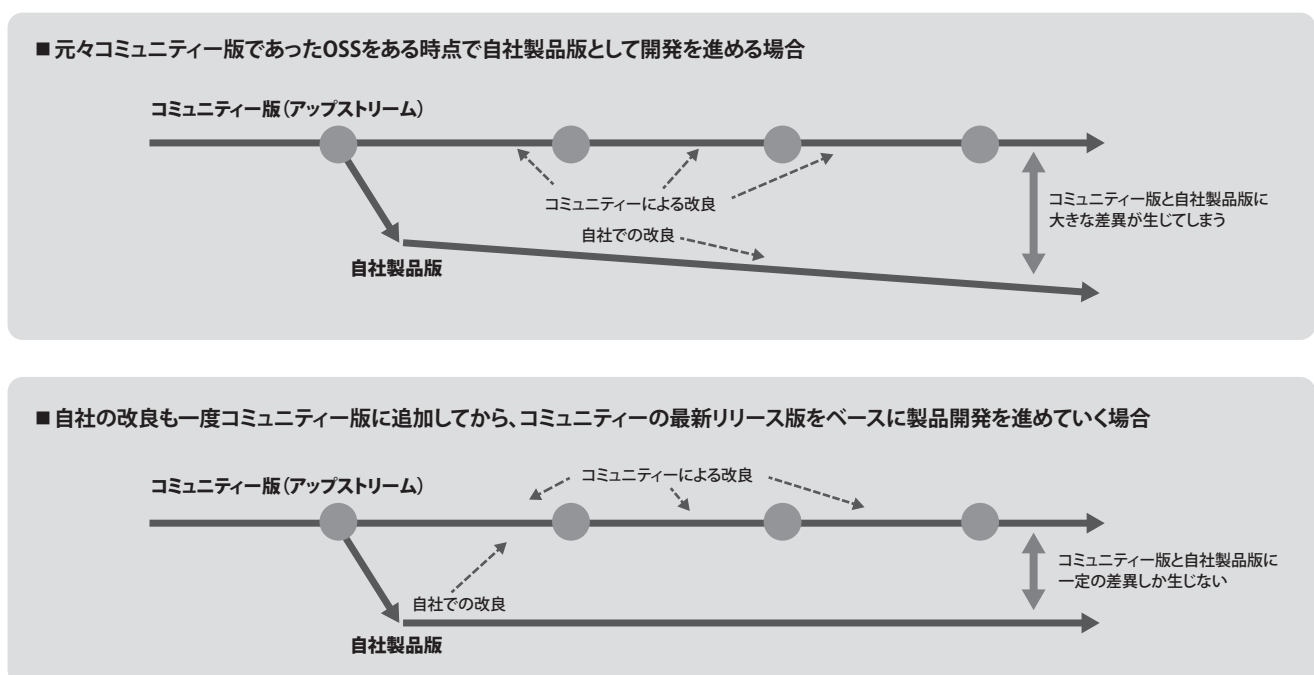


図1. アップストリーム・ファースト

う考えのもと策定されました。既存のオープンデータに適用されているライセンスはビジネス利用での記述があまりなかったため、そこを明確にするという目的もありました。2018年4月には、日本の貢献者グループによる日本語参考訳が公開されています[12]。CDLAには、CDLA-Sharing、CDLA-Permissiveの2つの種類があります。

### ●CDLA-Sharing

コピーレフトの概念をデータに応用するライセンスで、ライセンスの下でデータの使用と改変ができます[13]。データに加えた変更を共有する必要があります。OSSライセンスのコピーレフトの考えを意識して作られたGPLライセンスに近い考え方です。

### ●CDLA-Permissive

ライセンスの下で誰もがデータを使用、改変できます。データに変更を加えた場合であっても共有する必要はありません。変更を加えたデータを公開する場合は、このライセンスならびにファイルを改変したことに対して明確な表示を含める必要があります[14]。OSSライセンスのApache、MITライセンスに似た考えに基づいています。

## 5-2. IBM Data Asset eXchange (DAX)

機械学習用のオープンなデータセットを企業でも活用できることを目標として、IBMは「IBM Data Asset eXchange (DAX)」を展開しています[15]。2020年2月12日時点では27のオープンデータが公開されており、ビジネスで活用できる基盤ができています。DAXで公開されているデータにおいてもThe Linux FoundationのCDLAが多く採用されています。

## ▶▶ 6. おわりに

オープンソース活用の動きが活発になり、そのコンセプトがオープンデータへと波及し、新たなビジネスでの活用へと広がっています。このようなオープンな活動にはライセンスへの理解と同様、これらの活動に関わっているコミュニティとの連携が重要になってきます。コミュニティとともに、また自身もコミュニティの一員として貢献し、ITによるイノベーションを加速させていきましょう。

### [参考文献]

- [1] Red Hat : State of Enterprise Open Source, <https://www.redhat.com/en/enterprise-open-source-report/2020>
- [2] Redis、MongoDB、Kafkaらが相次いで商用サービスを制限するライセンス変更。AWSなどクラウドベンダによる「オープンソースのいいとこ取り」に反発, <https://www.publickey1.jp/blog/19/redismongodbkafkaaws.html>
- [3] The Free Software Definition, <http://www.gnu.org/philosophy/free-sw.en.html>
- [4] GPL Licence, <https://www.gnu.org/licenses/gpl-3.0.en.html>
- [5] LGPL Licence, <https://www.gnu.org/licenses/lgpl-3.0.en.html>
- [6] Apache Licence, <http://www.apache.org/licenses/>
- [7] OSI MIT licence, <https://opensource.org/licenses/MIT>
- [8] Open source license usage on GitHub.com, <https://github.blog/2015-03-09-open-source-license-usage-on-github-com/>
- [9] Black Duck Software Composition Analysis, <https://www.synopsys.com/software-integrity/security-testing/software-composition-analysis.html>
- [10] IBM : OpenSource@IBM, <https://developer.ibm.com/open/>
- [11] Community Data License Agreement, <https://cdla.io/>
- [12] The Linux Foundation : データの使用や公開に関する法的文書を貢献者グループが日本語化, <https://www.linuxfoundation.jp/press-release/2018/04/community-data-license-agreement-reference-translations/>
- [13] Community Data License Agreement – Sharing, Version 1.0, <https://cdla.io/sharing-1-0/>
- [14] Community Data License Agreement – Permissive, Version 1.0, <https://cdla.io/permissive-1-0/>
- [15] IBM : Data Asset eXchange, <https://developer.ibm.com/exchanges/data/>



日本アイ・ビー・エム株式会社  
デベロッパー・アドボカシー事業部  
クライアント・デベロッパー・アドボケイト

小園井 康志  
Osonoi Yasushi

日本IBM入社後、利用体験と併せてクラウド技術をお客様に広めるクライアント・デベロッパー・アドボケイトとして活動中。また個人的には幅広くLinux Foundationなどのオープンソースソフトの業界活動に関わっている。日本初のLinuxカーネル開発者会議であるカーネルサミットやLinuxCon、OSS summit、DrupalCamp DENなどの主催に関わる。