

オープンソースソフトウェアに潜む法的リスクの低減に向けた取り組み

Legal Risk Reduction in Open Source Software

前 田 博 史*, 貞 莉 昌 史*, 仲 西 秀 基*

(* NTT コムウェア株式会社)

要 約 近年、オープンソースソフトウェアは幅広い分野でその有効性が認められ、適用が進められている。しかしその一方で、ライセンスに起因するリスクなどの懸念点も指摘されている。主要なライセンスの中には頒布時にソースコードの公開義務が課せられるものもあり、場合によっては知的財産の流出や損害賠償などの企業損失が発生する可能性がある。

本稿ではこれらのリスクの程度や発生ケースを整理した上で対処方針を示す。特に、企業におけるリスク低減の取り組みについてソースコード検査を含む運用モデルを提案する。

Abstract The last several years have shown a worldwide rise in attention and actual use of open source software, such as GNU/Linux operating system, Apache, and PostgreSQL. Meanwhile, legal risks associated with open source software are also discussed due to enforceability of licenses. Software distributed under specific open-source licenses must include source code and allow free distribution. Open-source license violation sometimes results in costly litigation and loss of intellectual property.

In this paper, we categorize potential legal risks surrounding open-source licenses, analyze GPL violation cases, and suggest risk management solutions. In particular, this paper proposes practical steps, including source code inspection, to mitigate the legal risks arising from software development.

1. は じ め に

近年、企業や官公庁においてオープンソースソフトウェア（OSS^{*1}）の適用が進展している。オペレーティングシステムのLinux、ウェブサーバ系のApacheやTomcat、データベースのPostgreSQLやMySQLに代表されるOSSはシステム導入コストの削減や特定ベンダ製品戦略に対する独立というメリットから、システムの要件を考慮しながら適材適所での利用が進められている。

一方で、OSSの適用拡大に伴い、ライセンスに起因したOSS特有の問題も国内外で発生している。国外では、2003年にSCO Groupが所有するUNIXの権利をIBMが侵害したとして、「企業秘密の流用」、「不法な干渉」、「不公正競争および契約違反」に関して10億ドルの訴訟が行われた^{*2}。また、国内では、2004年にブロードバンドルータへのLinuxソースコード混入が明らかになり、該当製品はソースコードの公開^[1]、後に生産終了を余儀なくされている。さらに2007年には、British Telecomが提供するホームハブ製品についてGPL違反が指摘され、ソースコードの公開を受け入れている^[2]。このように、OSSライセンスに起因する法的な問題は現在も発生しており、法的リスク低減のための取り組みが求められている。

1.1 現状と課題

OSSは機能拡充や品質向上により、企業や個人間での普及が進みつつある。しかし、ライセンスに関しては文言の解釈、制約条件や対象範囲、各国法との関係において一定の解釈が存在するわけではなく、開発者や利用者のライセンスに対する理解が十分であるとは考え難い状況である。一方で、開発期間の短縮やコスト削減の圧力により、意識的にOSSを利用する機会は増大しており、OSSライセンス違反のリスクはソフトウェア開発に関わる企業のあらゆる開発現場に存在しているといえる。

こうした現状を受け、現在、OSSライセンス違反のようなOSSを取り巻く法的課題についてユーザの視点から状況を整理し、対策を検討する取り組みが各所で行われている^[3]。国内では、独立行政法人の情報処理推進機構（IPA）が中心となり、OSSに関する法的リスクの調査・分析及び対策の検討が実施されている。海外においてはFree Software Foundation（FSF）を中心に、ライセンスに関連した事例の紹介や利用時の注意点などが提示されている^[4]。

しかし、いずれの場合もOSSライセンスについて、Richard Stallman^{*3}氏の見解やライセンス条項の解釈を示すことに重点が置かれ、ソフトウェア開発の現場レベルで発生している実問題とは乖離を生じさせている。例えば、1) ソフトウェア開発ツールにより自動的に生成されたソースコードが偶発的に他のOSSライセンスのソフトウェアと一致していた場合、2) ソースコードを参照していたプロジェクトと同一のソースコードを含む、より制約の強い派生プロジェクトが存在する場合、ソースコードの一致をどのように判断し、ライセンス違反の境界をどこに定めるべきかについては整理されていない。

また、現実のソフトウェア開発では、単価の安い外部企業にソースコード開発を委託するケースが多数見受けられる。IPAによると、外部委託工数比率は71.1%であるとの調査結果が示されている^[5]。このため、発注元の開発管理者がOSSライセンスを熟知していたとしても、委託先、再委託先となるにしたがい、ライセンスに対する問題意識が薄れ、発注元によるリスクコントロールが困難になる状況が生じている。

つまり、現実のソフトウェア開発の現場においては、OSSライセンスには未解釈、未整理の領域が数多く存在し、委託先企業を考慮した包括的なリスク対策は、各企業に委ねられているといえる。

1.2 本稿の目的

本稿はOSSライセンスに起因するリスクやライセンス違反の発生ケースを整理し、具体的な法的リスクの低減策を示すものである。特に、実プロジェクトで開発された各種ソースコードを分析することで、OSSライセンスの文言の解釈にとどまらない実務の視点から、リスク低減施策の在り方やリスクの管理手法について論じることを目的としている。

1.3 本稿の構成および記述範囲

本稿は1章で現状と課題を整理し、2章ではOSSライセンスの分類に従い各ライセンスの影響と関係について論じている。また、3章ではOSSライセンス違反の発生ケースを整理し、想定されるリスクと対処方針を示している。4章ではリスク低減策の全体像、5章では具体的なOSSライセンス違反の対策を示している。6章では本稿の結論を示す。

また、本稿ではOSSライセンス違反に対して技術的観点に立脚した論を展開する。したが

って、OSS ライセンスの思想や文言的解釈を中心とした法的観点、内部統制のような経営的観点、企業倫理的観点での検討や考察は本稿の対象外とする。

2. OSS ライセンスとは

2.1 OSS ライセンスの分類

Open Source Initiative (OSI) によると同組織が認定している OSS ライセンスは、2007 年の時点で約 60 種類存在している^[6]。これらは「コピーレフト^{*4}」を中心としたソースコードの公開条件やライセンスの影響範囲などにより、GNU General Public License (GPL) 型、Mozilla Public License (MPL) 型、Berkeley Software Distribution (BSD) 型の大きく 3 種類に分類することができる (表 1)。また、Freshmeat.net^[7] によると 2007 年 9 月現在、46,639 の OSS のうち 64.96% が GPL、6.54% が LGPL である。しかし、このデータには約 10% の商用ライセンスが含まれているため、実質的には OSS の約 8 割が GPL 型であると考えることができる。

表 1 OSS ライセンスの比較

種 別 ラ イ セ ン ス	改 変 可 否	改変の基準 ^{*5}				公開義務		代表的OSS
		修 正	追 加	静 的 リ ン ク	動 的 リ ン ク	改 変 部 分	以 外 改 変 部 分	
GPL型	可能	○	○	○	○	必要	必要	Linuxカーネル, MySQL
LGPL	可能	○	○	○	×	必要	必要	glibc
MPL型	可能	○	○	×	×	必要	不要	Firefox, Thunderbird
BSD型	可能	—	—	—	—	不要	不要	FreeBSD, PostgreSQL

○：改変と判断する ×：改変とは判断しない —：ライセンスに制約なし

※Firefox, Thunderbirdは共にMPL, GPL, LGPLのトリプルライセンス

1) GPL 型

GPL は Richard Stallman 氏によって作成された OSS ライセンスの中で最もコピーレフトの性格が強いライセンスで、ソースコードの改変や追加に対しては、該当部分を含むソースコードの公開が義務付けられている。これは GPL の影響下にあるソフトウェアの二次著作物が独占的に利用されることを回避するためである。さらに他のソースコードへリンクした際には、リンクしたソースコードを含むソースコード全体にライセンス条件が波及し、公開義務が発生する。また、GPL の派生ライセンスとして、コピーレフト性の弱い GNU Lesser General Public License (LGPL) が存在する。これは GPL と異なり、他のソースコードへ動的リンクを行った際には、ライセンス条件が波及しないという特徴を有している。

2) MPL 型

MPL はネットスケープ社の Netscape Public License (NPL) を基に Mozilla Foundation が策定したライセンスである。MPL では改変部分のソースコードについては公開義務が発

生するが、GPLとは異なり、他のソースコードへリンクしてもライセンス条件は波及しない。したがって、独自に追加したソースコードに関しては公開義務が発生せず、また、該当部分を異なるライセンスで公開することができるという特徴を有している。

3) BSD 型

BSD ライセンスはカリフォルニア大学バークレー校によって、UNIX 互換オペレーションシステムの配布のために作られたライセンスであり、制約事項が少ないのが特徴である。著作権表示と免責条項を含めて頒布することが利用条件になっており、それ以外は改変部分を含めてソースコードの公開条件は規定されておらず、独自に作成したソースコード部分を含め公開義務が発生することはない。したがって商用ソフトウェアへの組み込みも可能である。

このように OSS ライセンスにはコピーレフト性の程度の違いがあり、ライセンスによっては直接的に組み合わせて商用利用することができないため、企業が OSS を利用する際の大きな障壁となりうる。ただし、GPL のようにコピーレフト性が高いライセンスであっても、ソフトウェアの独立性を維持して使用する場合は、コピーレフト性が商用利用の妨げになるわけではない。例えば、GPL である Linux カーネルが提供する機能は、LGPL である glibc を介して利用することが可能であるため、上位の商用製品やアプリケーションに GPL の影響が及ぶことはない。また、2006 年に GPL 化された Java でも、GNU Classpath というライセンスの追記事項が適用されているため^[8]、Java 環境における商用製品の利用やアプリケーション構築に関しては、GPL の影響範囲とはならない。

2.2 OSS ライセンスの変更

OSS ライセンスの中には、再頒布する際にライセンスの種別を変更することが可能な場合がある。これはコピーレフト性の強さに依存しているが、例えば BSD ライセンスでは、二次著作物のライセンスに制限が設けられていないため、BSD ライセンスのソフトウェアを GPL や MPL に変更、あるいは商用ライセンスに変更することが可能である。また、MPL は再頒布時には MPL を適用する必要があるが、ライセンスの但し書きにおいて、他のライセンスへの変更を認める旨の記述がある場合には、変更が可能である^[9]。特に近年では、GPL や LGPL への変更を認めるトリプルライセンスが採用されるケースが増えてきており、再頒布に際しては MPL、LGPL、GPL への任意の変更、あるいはトリプルライセンスの継続を選択することができる。なお、LGPL2.1 にはコピーレフト性の強い GPL への変更が可能である旨が第三項に明記されているが、GPL は他のライセンスへの変更はできない。したがって LGPL から GPL へのライセンス変更は不可逆となる。これらの対応を表 2 に示す。

表2 OSS ライセンスの変更可否

変更先 変更元	GPL	LGPL	MPL	BSD	商用
GPL	○	×	×	×	×
LGPL	○	○	×	×	×
MPL	○	○	○	×	×
BSD	○	○	○	○	○

○：変更可能 ×：変更不可能

※MPLはMPL, GPL, LGPLのトリプルライセンスである場合を示す

2.3 OSSの自由と開発者にとっての自由

FSFの主張するソフトウェアの自由はコピーレフトに基づくものであり、「ソフトウェアを利用する者に与えられる複製、改変、再頒布の自由の保証」である。この枠組みと照らし合わせると、商用ライセンスへの切り替えが可能なBSDライセンスはソフトウェアの自由を保証するものではない。その一方で開発者にとって制約条件の少ないBSDライセンスこそが最も自由なソフトウェアであるとの見解もある^[10]。このように立場によって「自由」の意味は異なるが、企業やその開発者が自由な経済活動を行うにあたっては、OSS利用時の制約条件が少ないことが望ましく、特にコピーレフト性と公開義務に注目することが必要である。次章ではこれらが開発にどのように影響するかについて考察を行う。

3. ソフトウェア開発とOSSの利用リスク

企業がソフトウェア開発にOSSを利用するにあたっては、OSSの機能要件・性能要件・サポートなどに加え、OSSライセンスや特許などの知的財産への考慮が求められる。特にOSSライセンスについては、著作権法との関連について法律専門家や政府機関などで議論が行われており、「OSSライセンスは法的効力を持つ」とする見解^[3]が強い。例えばGPLの条項を満たさない場合には、著作権法に抵触する可能性がある指摘されている。本章では、このようなGPL違反について現状の分析を行い、そのリスクや影響および対処について示す。

3.1 GPL違反リスクの増大

2.1で詳述した通り、GPL型のOSSはOSS全体の約80%を占めるとされており、GPL型のソフトウェアの現状やコピーレフト性を考慮すると、大勢を占める傾向は今後も継続すると考えられる。

更にインターネットでは、開発リポジトリのSourceForge.net^[11]やOSSソースコード検索サービスのGoogle Code Search^[12]などが普及してきており、OSSのソースコードを容易に入手できる環境が整いつつある。近年のソフトウェア開発の現場では、開発期間の短縮・コスト削減などの圧力やOSSライセンスへの理解不足により、GPLのソースコードを製品やシステムに組み込んだにも関わらず、自社製品として販売するなど、OSSソースコードの不正流用によるGPL違反が散見されるようになってきており、今後も増大する可能性が危惧されている。

特に外部委託によるシステム開発においては、ソースコードレベルの把握・管理は難しい。

GPL 違反が発覚した場合、企業は該当製品の販売中止やソースコードの修正あるいは公開、訴訟や企業イメージの失墜などの影響が予想されるため、予めリスク評価を行い、対策を講じておく必要がある。

3.2 GPL 違反による影響

企業において GPL 違反が発生した場合、以下のような影響が発生すると考えられる。

- ・ ソースコードの公開
- ・ 損害賠償金請求
- ・ 製品やサービス提供の打ち切り
- ・ 企業イメージの低下 など

ソースコードの公開義務は、改変・追加部分やリンク先などが対象となり、企業にとっては、独自に開発したプログラムにもその影響が波及するため、知的財産の流出などの影響が懸念される。仮にソースコードの公開義務が履行されない場合は訴訟に発展する可能性もある。海外ではドイツにおいて GPL 違反による著作権違反が認定され、損害賠償責任が発生したケースも存在する^[13]。

その一方でユーザ側においては、FSF や GPL Violations^[14] などがユーザの代表となり、企業による GPL 違反を指摘するという過程が確立しており、ユーザによる GPL 違反の指摘が発端で製品やサービスが打ち切りとなった事例も存在している。この中には企業の不誠実な対応により訴訟に発展したケースもあり、企業には早期の対応や明確な回答などの真摯な対応が求められる。また、このような違法行為により企業イメージが損なわれることについても十分注意を払う必要がある。

3.3 GPL 違反パターン

企業において GPL で公開されている OSS を利用する場合は、FSF が定義している条項に従ってどのようなケースが GPL 違反に該当するかを理解し、リスク管理を行う必要がある。GPL の違反ケースとしては、商用ソフトウェアへの不正流用や OSS ソースコード改変時の公開義務の不履行、著作権表示削除などが想定される。ただし、GPL のライセンス条項はそのソフトウェアが第三者に頒布された時に初めて有効となるため、第三者への頒布が行われない社内システムなどの内部利用では、ソースコードの公開義務の不履行などによる GPL 違反は発生しない点を理解しておく必要がある。

以下に GPL 違反ケースについて典型的な六つの分類を示し、詳細を述べる（図 1）。ケース 1～3 は商用ソフトウェア開発における GPL 違反、ケース 4・5 は OSS そのものを対象とする際に懸念される GPL 違反、また、ケース 6 は派生 OSS により起こりうる GPL 違反のケースである。特にケース 1～3 は商用ソフトウェアへの OSS 利用時に発生する可能性があり、格段の注意が必要である。また、ケース 6 のような GPL 違反に該当するか不明確なケースはこの他にも存在している可能性があり、未知のケースに対する初期対応体制や対処方法の整備など、的確な対応を行う体制作りも企業に求められる。

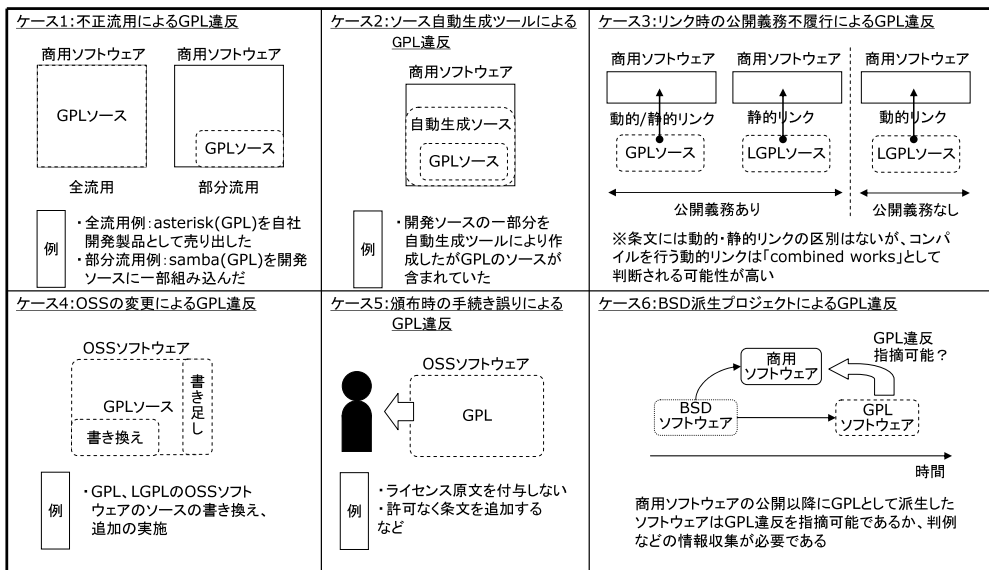


図1 GPL 違反パターン

1) ケース1：不正流用による GPL 違反

GPL で公開されている OSS を自社製品や自社の開発するシステムに組み込んでいるが、これを公表せずに頒布した場合、GPL 違反に該当する。この場合、独自に修正・追記したソースコードも含め公開し、また、GPL に従うソフトウェアであることを示す義務が発生する。不正流用のパターンとしては、OSS のソースコード全体を流用する全流用と一部のみ流用する部分流用とに分けられる。

2) ケース2：ソース自動生成ツールによる GPL 違反

近年、ソースコードの自動生成ツールが普及し、開発時に利用されるケースが見受けられるようになったが、ツールにより自動生成されたソースコードに GPL のソースコードが混入していた場合には、GPL 違反に該当する可能性がある。

また、GPL で公開されている OSS が自動生成ツールを使用して開発した場合、企業が同じ自動生成ツールを使用して開発すると、OSS と同じソースコードやテストツールなどが自社開発のソフトウェアに組み込まれてしまう可能性が発生する。このようなケースが GPL 違反に該当するかは明確になっておらず、注意が必要である^[15]。

3) ケース3：リンク時の公開義務不履行による GPL 違反

GPL の OSS とリンクしたソフトウェアを開発した場合、静的・動的に関わらず GPL の影響範囲となるため、開発したソースコードの公開義務が発生するが、LGPL の OSS とリンクした場合は静的・動的により対応が異なっており、静的リンクは改変と判断されるが、動的リンクの場合は LGPL の影響が波及せず、GPL 違反に発展することはない。ただし、リンクに関しては GPL の条文に明確な記述はなく、動的リンクであってもソフトウェア的に密結合であれば、改変に該当するなどの諸説もあるため、各方面での議論や事例を注視する必要がある^{[16][17]}。

4) ケース 4: OSS の変更による GPL 違反

GPL で公開されている OSS のソースコードに対し、書き換え・書き足しなどの変更を加え、頒布した場合はソースコードを公開する必要がある。ソースコードの公開義務は、OSS の頒布先に対してのみ発生するため、例えば、特定の顧客にのみ GPL の OSS を含むソフトウェアを納めるような場合では、その顧客に対してソースコードの開示が必要となる。なお、GPL の OSS を含むソフトウェアが一般消費者などの不特定多数の手に渡るような場合は、ソースコードを Web サイトで公開するなどの手法がとられている。

5) ケース 5: OSS 頒布時の手続き誤りによる GPL 違反

OSS を GPL のもとで頒布・再頒布するには、満たすべき条件があり、特に派生元の OSS に付与されている「GNU General Public License」の原文（英語）を引き続き付与すること、ライセンスに記される条項に許可なく修正を加えないなどの条件を満たす必要がある。したがって、著作権表示を無断で削除するなどの行為は GPL 違反となる。

6) ケース 6: BSD 派生プロジェクトによる GPL 違反

BSD ライセンスで公開されている OSS は、GPL への変更が可能であるため、これにより GPL 違反が発生する可能性が考えられる。

仮に BSD ライセンスの OSS である「ソフト A」を改変して商用製品「ソフト B」を開発した場合、「ソフト A」の派生プロジェクト「ソフト C」が GPL として公開されたとすると、「ソフト C」の GPL のライセンスが「ソフト B」に波及するかは明確になっていない。「ソフト B」は「ソフト C」の派生ソフトウェアか否かが論点になると思われるが、本ケースが GPL 違反に該当するかどうかは本稿において断定することは難しい。ただし、企業活動を行う上では、このような事例も予想されるため、各企業において対処方針を整理しておく必要がある。

3.4 GPL 違反への対処

自社製品への GPL ソースコードの混入が発覚した場合、GPL 違反にならないよう対処する必要があるが、発覚時点によっては、とり得る対処方法は多少異なる。発覚がソフトウェアの頒布前であれば、以下のような対処が考えられる。

- ・ GPL のソースコード部分を取り除く
- ・ GPL 部分を独立した OSS となるよう切り離す
- ・ GPL を適用し公開する

対処にあたっては、状況やコストを考慮して適切に実施する必要があるが、納期間近などの切迫した状態においては、迅速な対処が困難になる恐れがあるため、予めソフトウェアの開発工程に GPL 違反に対するリスクマネジメントを組み込んでおくことが必要である。

一方、ソフトウェアの頒布後に GPL 違反の指摘を外部から受け、GPL 違反が確認された場合には、GPL の影響範囲となるソースコード全てを公開する以外の対処は考え難い。これは違反部分の削除・修正などを実施してもそれを証明することができず、ソースコードの公開以外に GPL 違反の疑惑を払拭する手段が見当たらないためである。

4. リスク低減の取り組み

ソフトウェア開発において、法的なリスクを完全に回避することは非常に困難であるが、企業はリスク低減に向けた取り組みを行うことにより、ユーザや取引先の信頼を確保する必要がある。

本稿では、リスク対応の分類（低減・回避・保有・移転など）のうち、低減と回避の観点に着目し、特にリスクの低減に向けた取り組みのモデルを示す。本モデルでは三つのフェーズを定義し、各フェーズにおいて目的や対象に応じた実施内容を挙げている。具体的には OSS 利用のポリシー策定、研修、外部発注時の契約、ソースコード検査および開発後の問い合わせ対応について提言している（図 2）。各実施内容については以下に詳細を述べる。

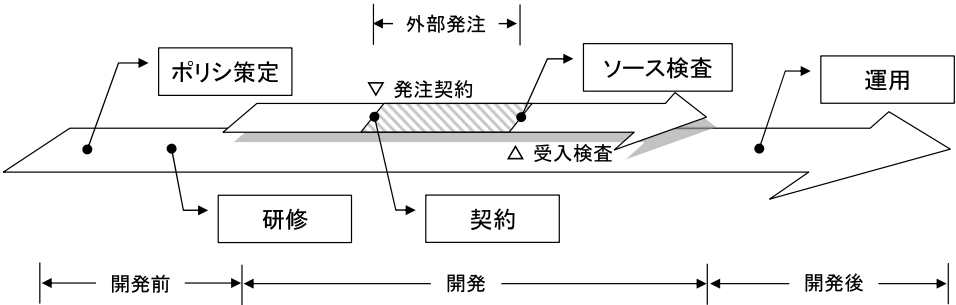


図 2 ソフトウェア開発における GPL 違反リスク低減の取り組み全体像

4.1 ポリシー策定

企業が開発するソフトウェアに OSS を利用するにあたっては、OSS の持つメリットを活かしつつ、OSS の利用に伴うリスクを低減するため、企業としての利用方針や規約・推奨などを明確化したポリシーを策定しておくことが重要である。ポリシーには OSS の定義や特徴に加えて、GPL 違反などのリスクに対する意識付けも含める必要があり、特に OSS ライセンスの影響範囲が不明確なケースや想定外のケースへの対処など、各企業で独自に検討すべき領域に關しても十分に議論した上でルール化しておく必要がある。

また、策定したポリシーを社内へ展開する際には、GPL 違反などの問題発生時に対応する体制を構築しておく必要があり、その活動には例外的なケースの対応、OSS ライセンスの動向収集、ポリシーの更新なども含まれる（図 3）。

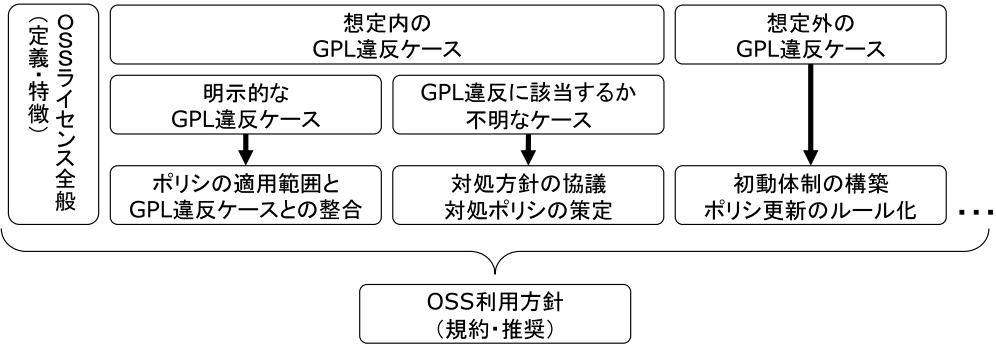


図 3 OSS 利用ポリシーの構成例

4.2 研修

ソフトウェア開発を事業としている企業であっても全ての人員が OSS を理解しているわけではなく、また、理解の程度も異なると考えられる。また、今日では顧客が OSS の利用を指定するケースも多く見受けられることから、開発の前フェーズである提案の段階に携わる営業担当者にも OSS の知識が求められるようになっている。

したがって、開発者だけでなく営業担当者やプロジェクト管理者、運用担当者などソフトウェア開発に関わる全ての人員に対して、OSS に関する基礎的な知識や自社ポリシーへの理解を促進させるとともに、営業担当者に対しては、OSS を利用する開発ソフトウェアを納入する際の注意事項、開発者に対してはどのように OSS を開発ソフトウェアで利用すべきかなど立場に応じた研修を実施することが、ライセンス違反を回避する上で有効であると考えられる。

4.3 契約

近年、ソフトウェア開発に対する価格低減や納期短縮などの圧力の高まりにより、開発のアウトソーシングが活発に行われているが、アウトソーシング先での開発において不適切な形で OSS が利用されることがないように対策をとることが必要である。

OSS の不正利用を回避するという観点では、発注先に対しても OSS に関する研修を促し、また、利用する OSS を指定するということが考えられるが、契約形態によっては、発注先の開発者個人に対する指示が不可能な場合もあり、実施および管理は発注先に頼らざるを得ない。したがって、このような場合は契約書において OSS の不正利用を制限あるいは利用する OSS を明確化することで、問題発生時に損害賠償請求権を行使できるようリスク移転の手法を併用することが考えられる。この際、基本契約書に新たに条項を追加することも考えられるが、受発注者の双方における法務手続きを考慮すると、誓約書や覚書などの形態をとることが現実的である。

4.4 ソース検査

ポリシー策定や研修、契約などによって、開発するソフトウェアに OSS が不適切な形で利用されることのないように配慮しても、OSS が混入する可能性は残存しており、GPL 違反やそれに伴うソースコード公開義務の発生が懸念される。

このような残存リスクを管理する手段として、開発したソースコードを走査し、OSS が含まれているか確認することのできるソフトウェアが発達してきており、実用化が近づいている。このような仕組みにより、これまでは人手に頼らざるを得ず、特に数百キロライン以上の大規模なソフトウェアの開発においては現実的ではなかった全ソースコードを対象とした検査の実施が可能になると期待される。

4.5 運用

ソフトウェアの開発が終了した後は運用段階に移行するが、開発したソフトウェアのバージョンアップやバグフィックスなどへの対応と同様に、OSS ライセンス違反が発生した際の対応も考慮しておく必要がある。これまで各国で報告されている事例では、エンドユーザからの指摘によりライセンス違反が発覚するケースが多い。この中には企業側の不適切な対応により

問題が拡大した事例も見受けられることから、まずは指摘を適切に受け止めることが重要である。そのためには問い合わせを受け付ける一次窓口の設置やその後の対処方法などについては予め整理し、体制を整備しておく必要があると考えられる。

5. ソースコード検査の実用化

本章ではリスク低減の取り組みのうち、近年、特に実用化が進んでおり、GPL 違反を抑止する最終的な対策として期待される「ソースコード検査」に着目し、検査の仕組みおよび運用モデルについて述べる。

5.1 ソースコード検査の自動化

OSS のプロジェクトはリポジトリサイトに登録されているものだけでも数万あるといわれており^[1]、ソースコード内にこれらの OSS が含まれているかを人的に検査することは極めて困難である。まして、大規模なシステムであればソースコードは数百キロライン以上になることもあり、人手に依存するソースコードの検査はリスク回避の現実解ではない。

ソースコードの検査を現実のものにするためには、ソースコードに内在する OSS（例えば GPL ソースコード）を自動的に検出するような仕組みが必要であるが、単純に文字列の一致を検出するだけでは十分ではなく、ソースコードへの複雑な変更や追加が行われているような場合であっても類似性を検出するような仕組みも必要である^{*6}。

近年の OSS ライセンスによるリスク管理の必要性の高まりにともない、これらの要求に応えるライセンスチェックソフトウェアの分野が成長しつつある。これにより数十万にもおよぶ OSS を対象とした自動検査が可能となり、ソースコード検査の実施が現実的なものとなっている^{*7}。次節ではソースコード検査の運用方法について提案する。

5.2 運用モデル

ライセンスチェックソフトウェアを用いてソースコード検査を実施するにあたっては、検出機能などの技術面の検討だけでなく、誰がどのような分担で実施するかなどの運用面に関しても明らかにしておく必要がある。特に、OSS ライセンス全般に対する理解、検査結果をもとにした GPL 違反の判定、違反時の対処方針の決定など専門的な判断が必要とされる業務に関しては対応可能な体制の構築が必要で、そこではシステム開発者や知的財産担当者の素養を併せ持つ専門の部署あるいは担当者を設けることが望ましい。

これらをもとにした運用モデルを以下に示す（図 4）。

・ システム開発者

システム開発者はソースコード検査を行う際に必要となるソースコードおよび諸情報の提供を行う役割を担う。ソースコード検査担当者に提示する情報としては、意図的に使用している OSS の有無、ある場合にはその詳細、過去のリリース実績などである。

・ ソースコード検査担当者

ソースコード検査担当者はシステム開発者から受領したソースコードをライセンスチェックソフトウェアで走査し、不適切な OSS が混入していないか（例：GPL 違反となるソース

コードは存在するか) 検査を実施する。また、OSS が意図的に利用されている場合には正しく利用されているか、頒布時にライセンス違反にならないかなどを確認する必要がある。ライセンス違反に該当するかどうか専門家の判断が必要なケースが発生した場合には、知的財産担当者と協議し対処を決定する必要がある。

- ・ 知的財産担当者

知的財産担当者はライセンス違反に該当するかどうか判断が難しいケースの場合、ソースコード検査担当者の依頼のもと、OSS ライセンスの最新動向や過去の判例、リスク管理の手法などの観点および独自のポリシーに基づいた対処を検討する。

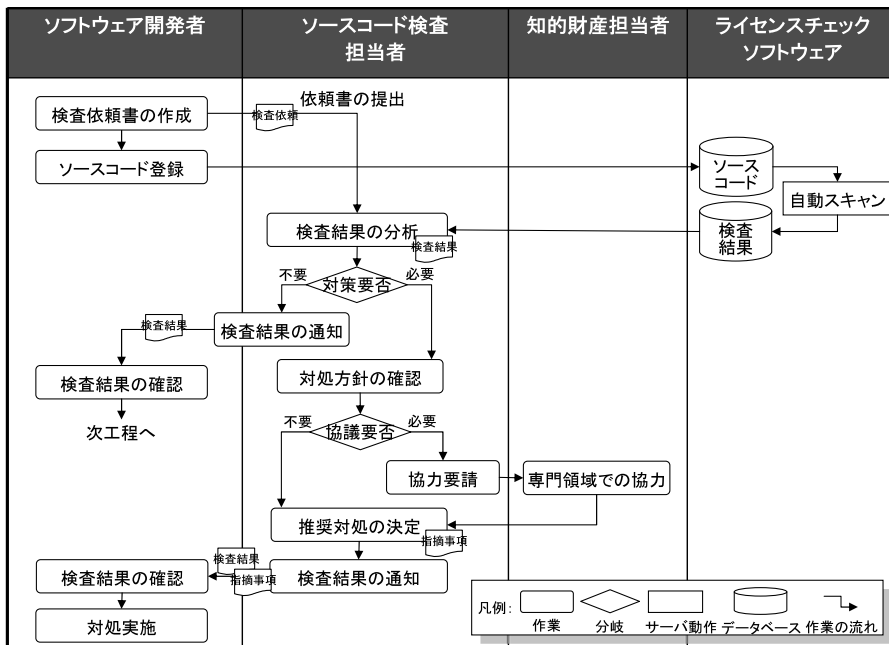


図4 ライセンスチェックソフトウェアの運用モデル

仮に GPL 違反などのライセンス違反が検出された場合には対処を施す期間が必要となるが、検査にあたってはソースコードの完全性は問われず、随時実施しても問題ないため、モジュール単位など開発者の判断に基づき可能な限り早い段階で検査を行うことが望ましいと考えられる。また、開発が完了したソフトウェアが OSS ライセンスに抵触しないことを担保するため、納品物の完成段階または受け入れ検査時にもソースコード検査を実施することが望ましい。ただし、ソフトウェアを開発する企業では、独自に開発規則が定められていることも多いため、そのプロセスに沿って実施することが望ましい。例えば試験工程などにソースコード検査を実施するよう開発規則に組み込むことも一案である。

実際にソースコード検査を実施するにあたっては、検査結果をどのように分析・判断するかが問題となる。例えば開発したソースコードが多数のOSSと一致し、複数のOSSライセンスに該当するというケースが想定される。このような複雑なケースの分析は難易度が高いため、ソースコード検査の導入初期においては、分析・判断はソフトウェア開発者以外のソースコー

ド検査者あるいは別の第三者が集中的に実施することが望ましい。そして、ソフトウェア開発者の理解が深まれば、開発者が各自でライセンスチェックソフトウェアを利用することで、開発の初期段階から OSS の不適切な混入を防止することが可能となる。また、ソースコードの走査までをソフトウェア開発者が行い、分析・判断のみソースコード検査者が実施するという折衷型の検査方法も考えられ、集中的実施から個別実施への移行期に適した形態であるといえる。

5.3 課題

実際に運用するにあたっては、検査結果の分析手法やソフトウェア開発者とソースコード検査担当者との役割分担など検討すべき課題は多い。ライセンスチェックソフトウェアによるソースコード検査で得られる情報は、あくまで OSS の混入有無であり^{*}、該当ファイルを不特定多数に頒布するかどうかなどのプロジェクトからの情報と合わせて、対処の要否や影響度などを分析する必要がある。したがって、両者の間で共有されるべき情報が円滑に伝達されるようなフレームワークが重要となる。また、検査の結果、ライセンス形態の異なる派生プロジェクトが多数検出されることも考えられるが、その際は各ライセンスの特徴や依存関係などを確認しつつ分析することが求められる。

このようにライセンスチェックソフトウェアは万能ではない。検査するソフトウェアの件数や規模が増大すると、ソフトウェア検査担当者が全ての検査を実施することが困難になると予想されるため、ソフトウェア開発者に対し検査ノウハウを展開することも考慮しつつ、各企業に即した運用モデルを構築する必要がある。

海外では電気通信事業者のホームゲートウェイやネットワーク機器会社の IP 電話・ルータなど GPL 違反事例が報告されている。しかし、国内での事例はまだ少なく、訴訟や判例も見当たらないため、GPL などの OSS ライセンスが国内法でどのように位置づけられるかは明確になっておらず、動向の把握も大きな課題であるといえる。

6. おわりに

本稿では OSS ライセンスに起因するリスクやライセンス違反の発生ケースを整理し、具体的な法的リスクの低減策、特にプロジェクトで開発された各種ソースコードを走査・分析する手法について論じた。ソースコードの検査は究極的な対処方法ではあるが、ライセンスチェックソフトウェアには機能面において解決すべき課題は残存しており、人手による高度な分析や判断が要求されるため、4 章で紹介したモデルと併用することが望ましいと考える。特に、企業はコンプライアンスの一環として、ポリシー策定や研修など、そもそもライセンスに抵触しない適切な利用の推進に率先して取り組むべきである。

近年、企業による OSS の利用は活発化する傾向にあるが、オープンソースという特性に起因するリスクについても認識されつつあり、企業によってはリスクを低減する対策が検討され始めている。このように OSS を利用するのは危険であるとして排斥するのではなく、リスクの評価や対策を適切に実施し OSS を適材適所に適切に利用する環境を整備することで、企業は OSS のメリットを最大限に享受すると共に、OSS は更なる発展の基盤を得ることができるようになると思われる。

- * 1 オープンソースの「自由」という面を強調するために, “Free and Open Source Software (F/OSS)” や “Free/Libre and Open Source Software (FLOSS)” と呼ぶことも広く受け入れられている^[18].
- * 2 複数の裁判所で数年にわたって争われてきたが^[19], 2007年8月にユタ州連邦裁判所よりUNIXの所有者はNovellであると判決が下されている.
- * 3 フリーソフトウェア運動の中心的存在で, GNUプロジェクトやFSFを設立し, 自由なソフトウェアのためのライセンスを考案した人物.
- * 4 「GNUプロジェクト」の著作権に関する根本思想で, 一度公開されたソフトウェアは, 二次著作物として再頒布された場合でも利用, 改変と再頒布の自由を与えるという考え方である. この仕組みには, コピーレフトが適用されるソフトウェアの二次著作物が企業などにより独占されることを防止し, ソフトウェアの自由を守る意図が込められている^[20].
- * 5 表中では便宜的に「静的リンク」, 「動的リンク」という分類を行っているが, FSFにおいて「静的/動的」の区別は行われておらず, 「modification」もしくは「combined works」として扱われている. また, 本文でGPL型として分類しているLGPLは, 動的リンクにより商用利用が可能であるという観点では, MPL型に分類されるとの見解もあるが, 本稿では, GNUにより定められるライセンスという観点から, LGPLをGPL型に分類している.
- * 6 例えばオリジナルファイルをコピーし, 「変数名」のみを変更した場合や「関数順序」を作為的に入れ替えた場合に対しても, 類似性を検出する機能が必要である.
- * 7 社内の性能試験においても, 近年のライセンスチェックソフトウェアは20Mbyteのサイズのソースコードを40分程度で検査完了することが確認されている.
- * 8 ライセンスチェックソフトウェアは設定された閾値に従って行われるため, 誤検出が発生することもある.

参考文献

- [1] ELECOM, <http://www.elecom.co.jp/support/news/20040426/index.html>.
- [2] Graeme Wearden, “BT accused of GPL violation,” *CNET Networks*, Jan 23, 2007, <http://news.zdnet.co.uk/software/0,1000000121,39285585,00.htm?r=4>.
- [3] 独立行政法人情報処理推進機構, ビジネスユースにおけるオープンソースソフトウェアの法的リスクに関する調査, 2005年, <http://www.ipa.go.jp/about/jigyoseika/04fy-pro/open/2004-741d.pdf>.
- [4] Free Software Foundation, <http://www.fsf.org/licensing/licenses/gpl-faq.html>.
- [5] 独立行政法人情報処理推進機構, ソフトウェア・エンジニアリング・センター, ソフトウェア開発データ白書2007, 日経BP, 2007年.
- [6] Open Source Initiative, License Index, <http://opensource.apachenews.org/licenses/index.php>.
- [7] SourceForge, License breakdown, <http://freshmeat.net/stats/#license>.
- [8] Free Software Foundation, <http://www.gnu.org/software/classpath/license.html>.
- [9] GNU Lesser General Public Licence, version 2.1, <http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>.
- [10] Michaelson, Jay, “There’s No Such Thing as a Free (Software) Lunch,” *Queue*, ACM Press 2, no. 3 (2004): 40-47.
- [11] SourceForge, <http://sourceforge.net>.
- [12] Google, <http://www.google.com/codesearch>.
- [13] Peter Judge, “German court convicts Skype of GPL breach,” *CNET Networks*, July 25, 2007, <http://news.zdnet.co.uk/software/0,1000000121,39288211,00.htm>.
- [14] gpl-violations.org, <http://gpl-violations.org>.
- [15] 田村善之, 著作権法概説, 有斐閣, 2001年.
- [16] McGowan, D. “Legal Aspects of Free and Open Source Software,” in *Perspectives on Free and Open Source Software*, J. Feller, B. Fitzgerald, S.A. Hissam and K.R. Lakhani (eds.), The MIT Press, Cambridge, MA (2005): 361-392.
- [17] Michael J. Karels, “Commercializing Open Source Software,” *ACM Press* 1, no. 5 (2003): 46-55.
- [18] Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K. (eds.), *Perspectives on Free and Open Source Software*, The MIT Press (2005).
- [19] SCO, SCO FILES LAWSUIT AGAINST IBM, <http://www.sco.com/scosource/SCOFilesComplaintAgainstIBM.html>.
- [20] O’Reilly, Tim, “Lessons from Open-Source Software Development,” *Communications of the ACM* 42, no. 4 (1999): 32-37.

執筆者紹介 前田 博史 (Hiroshi Maeda)

1999年神戸商科大学管理科学科卒業。同年、NTTコミュニケーションウェア株式会社（現NTTコムウェア株式会社）に入社し、技術開発部にてオブジェクトリポジトリ情報共有基盤の研究に従事。その後、エージェント技術・位置情報管理技術に関する商品企画・ビジネス開発を担当。現在、基盤技術本部オープンソースソフトウェア推進部に所属し、ライセンス問題対策を含む事業戦略策定に従事。

貞 莉 昌 史 (Masafumi Sadakari)

2004年同志社大学大学院工学研究科知識工学専攻修士課程修了。同年、NTTコムウェア株式会社に入社し通信事業者向け高可用基盤システムの開発業務を担当。現在、基盤技術本部オープンソースソフトウェア推進部に所属し、ライセンス問題の調査・対策立案を行う一方、オープンソースソフトウェアのビジネス利用に向けた事業戦略策定に従事。

仲 西 秀 基 (Hideki Nakanishi)

1979年慶應義塾大学工学部電気工学科卒業。1981年大阪大学基礎工学部前期博士課程修了。同年、日本電信電話公社（現NTT）横須賀電気通信研究所入社。以来、計算機自動設計研究、オンラインランザクションパッケージ（APLICOT）開発、社内システム標準化等を担当。1998年NTTコミュニケーションウェア株式会社（現NTTコムウェア株式会社）技術開発部を経て、現在、基盤技術本部オープンソースソフトウェア推進部長：OSS戦略、適用推進、基盤機能開発担当。