ソフトウェア業界におけるオープンソースコンプライアンス関連の4つの課題

事業運営については短期的目標と長期的目標を計画しますが、ソフトウェア業界が全体で対処する必要のあるオープンソースコンプライアンス関連の問題として、**スケール**、**精度**、**コスト**、**スピード**という4つの課題が挙げられます。

会社の規模、オープンソースポリシー、製品の種類などによってこれらの課題の解決策は異なりますが、どのような解決策を取るかに関わりなく、これらの問題はしばしば絡み合って同じように現れます。

スケール

この問題は、小規模なスタートアップ企業では既存の大企業ほど大きな問題として捉えられない場合があります。私たちはフォーチュン100社と連携していますが、これらの企業では何万人という開発者たちが毎日毎日次の2つの仕事をしています。それは、

- 新しいソフトウェアの開発、そして
- 自社製品・サービスにデプロイするための、オープンソースソフトウェアの再利用および用途変更です。この再利用には、オープンソースコンポーネントをそのまま再利用する場合と、コードスニペット(オープンソースコンポーネントから複数のコード行を適宜コピーして、別のコンポーネントに再利用)という形で部分的に再利用する場合の二通りがあります。

何千人という開発者がこれら2つの作業にいそしんでいることを考えたとき、スケールの問題は現実味を帯びてきます。

- このようなオープンソースソフトウェアの大量流入を管理するために、組織内のプロセスをスケールすることはできるでしょうか?
- オープンソースソフトウェアへの貢献といった側面を管理するために、組織内のプロセスをスケールすることはできるでしょうか?
- デプロイされているツールは、オープンソースアクティビティのレベルを取り扱えるでしょうか?
- 組織のオープンソースソフトウェア取り扱い手順の流れを良くするために、最適化すべき、場合によっては排除すべき不要なチェックポイントはないでしょうか?
- コンプライアンスツールは、ビルドシステムに統合するための利便性を提供するとともに、派生元の識別プロセスがよりシームレス・透過的に行えるようになっているでしょうか?
- 使用しているコンプライアンスツールは、プログラミング言語非依存でしょうか?

上記はすべて、オープンソースソフトウェアの使用状況(およびオープンソースソフトウェアへの貢献)に応じたスケールの拡大・縮小が可能な方法でオープンソースコンプライアンス対策を最適化できるよう、組織が自ら尋ね、解決に挑む必要のある問題点です。

精度

オープンソースコンプライアンスを専門とする人にとって、ソースコードの派生元とライセンスの識別精度は最も気になることの一つでしょう。そもそも、オープンソースコンプライアンス対策の主な目的は、コードとライセンスの派生元を識別したうえで、その結果に応じてライセンス義務を満たすための計画を立てることにあります。しかし、以下のような理由により、精度に関する問題は残ります。

- 既存ツールの中には、いわゆる「ナレッジ ベース」— 既知のオープンソースコードがすべて格納されたデータベース — を保持していないものがあります。このようなツールは、コードをスキャンして検出したライセンスと著作権情報を提示するにとどまります。
- 一方、急速に進むオープンソース開発に遅れを取るまいと、ナレッジベースを最新の状態に保つという難題に挑んでいるツールプロバイダー(ナレッジ ベースを保持するプロバイダー)もいます。しかし、多くの場合ナレッジベースの更新は数か月おきにしか行われないため、オープンソースソフトウェアの早期適用には間に合わないことが多々あります。
- 多くのツールプロバイダーは、スニペット検索機能をサポートしていません。このようなツールは、そのまま使用されているオープンソースコンポーネントしか検出できません。一般的に、このようなツールは、あるオープンソースコンポーネントから別のオープンソースコンポーネントに、あるいは自社ライセンスでライセンス保護されているコンポーネントにコピーされたコードスニペットを検出することはできません。
- どのコード一致が真の一致なのでしょうか?これは、コードをスキャンした結果疑わしいコードについて数十件から数百件の一致候補を受け取る、コンプライアンス担当者にはおなじみの質問です。先にも挙げた、その他の何千というオープンソースコンポーネントにコードが再利用されている、非常に人気のあるデータ圧縮ライブラリ、zlib を例にとって見てみましょう。コードの派生元が zlib であるソフトウェアコンポーネントをスキャンした場合、現在市販されているほとんどのツールは、コードの派生元とライセンスの識別で大きな問題が生じます。このようなツールは、通常、異なるライセンスの一致を何百件も提示します(zlib には自由ライセンスがあるため、zlib からコピーされたコードは、ターゲットコンポーネントのライセンスによって再ライセンスを受けることが多々あります)。開発者ではないコンプライアンス担当者は、検出されたこれらのソフトウェアコンポーネントをどのように識別し、この問題を解決すればよいのでしょうか?これは、ツールプロバイダーが対処する必要のある問題です。

コスト

オープンソースコンプライアンスはサポート機能と見なされることが多いため、すべてのエンジニアリング責任者は、適用されるオープンソースライセンスには確実に準拠しながらも、そのコストを抑えたいと考えています。しかし、どうすればコンプライアンスの確保にかかるコストを妥当な低レベルに抑えられるのでしょうか?これには、以下の点を考慮する必要があります。

- ツールのライセンス料にかかるコスト
- 組織のインフラストラクチャ内で行う特化カスタマイズと統合にかかるコスト
- ツールの実行に必要なサーバーハードウェアの初期コストと、これらのサーバーの保守にかかる継続的なコスト
- 偽陽性をすべてチェックしてクリアする役目のスタッフを含め、コンプライアンス確保のみを目的としたリソースにかかるコスト
- コンプライアンスの確保は高くつく可能性があります。我々が業界としていかにコストを妥当に抑えられるかは、今後取り組む必要のある課題の一つです。

スピード

不動産業者にとって「立地条件、立地条件、立地条件」が共通のキーワードであるように、オープンソースコンプライアンスにおいては「スピード、スピード、スピード」が共通のテーマです。コンプライアンス対策と開発との歩調を合わせて、コンプライアンスが遅れを取らないようにするには、どうすれば良いのでしょうか?何千人もの、場合によっては何万人もの開発者がオリジナルコードを書き、オープンソースコードを再利用しているとき、どうやってその膨大なコード行の本体に然るべきスキャン・識別・追跡を行って、コンプライアンスを確保すればよいのでしょうか?

FossIDは、開発プロセスにシームレスに統合し、コードベース内からフリー&オープンソースソフトウェア(FOSS)の断片(コンポーネント全体にはじまりコードスニペットにいたるまで)を検出する最先端のオープンソーススキャナーを提供します。FossIDのソフトウェアは、ライセンス義務とコンプライアンスの問題を明らかにし、御社が優れた製品の開発に集中できるように貢献します。

www.fossid.com
@fossid_ab
linkedin.com/company/fossid-ab

GET IN TOUCH!

Discover all FossID
products and services at
www.fossid.com

FossID AB
Gåsgränd 3
111 27 Stockholm
Sweden

FossID K.K.
1-10-3-200 Roppongi, Minato-ku
Tokyo 106-0032
Japan