

オープンソースソフトウェア管理プロセス

初めに

現在多くの企業はソフトウェアの品質向上プロセスの一環として、既存のソフトウェア品質チェック項目に、オープンソースやサードパーティソフトウェアライセンスの管理を加えようとしています。

オープンソースソフトウェア管理プロセス

(Open Source Software Adoption Process、OSSAP)

オープンソースソフトウェア管理プロセスの 8 項目は Protecode 社が数多くのお客様に対して行った企業合併買収 (M&A) 時のソフトウェア IP 監査や出荷前のオープンソース管理へのコンサルティングの経験に基づいてまとめられています。



OSSAP の 8 つのステップ

1. ソフトウェアのライセンスポリシーの確立、設定

これは組織で利用を許可する IP 及びライセンスコンプライアンスを定義する必須のステップです。

対象となる組織は製品グループ、プロジェクトグループ等となります。利用を許可されるライセンス条項、ベンダー、ソフトウェアパッケージ等をポリシーとして決定します。またここではパッケージの事前承認の方法を定義し、ソフトウェア開発の各段階でポリシーの違反が検出された場合どう対処するかを決めます。ツールにライセンスポリシーを記録しておけば、オープンソースソフトウェア適用ポリシーを各種ステップで使用される自動ライセンス管理ツールが使用できます。

一般的にビジネス管理者、ライセンシングもしくは法務部門、開発エージェンシーの代表者などがソフトウェアライセンスポリシーを決定し、ソースコードのポリシー違反の検出のワークフローを管理します。

2. ソフトウェアパッケージの事前認証

組織内でソフトウェアパッケージの事前認証を決定する手順と実施方法を決めます。このステップ必須ではありません。

パッケージの事前認証にはきちんと定まったサードパーティソフトウェア管理ポリシーが必要となります。

ソフトウェアパッケージの事前認証プロセスは以下のようになります。

- a) 要求。開発者は特定のパッケージの利用許可を要求します。要求には例えばパッケージに関する情報、名前、著者、ライセンス、パッケージを得るための付加的な情報等なるべくたくさん含めます。またパッケージが製品でどのように使用されるかを特定することも必要です。
- b) 記録。要求と了承ステータスを追跡するデータベースに記録します。
- c) 審査。審査員が要求を審査します。一般的に審査は(手動もしくは自動スキャン)による要求されたパッケージの監査が必要とされます。
- d) 要求の承認

一旦パッケージが承認されたら、パッケージ承認を記録し、そのステータスを組織で利用できるようにします。

3. 既存のコードに対するアセスメント

既存のコードのチェックや組織内で既に存在するベースラインの確立を含みます。このステップは必須となります。

ステップ 1 でのソフトウェアライセンスポリシーが登録されたツールやステップ 2 で事前認証されたパッケージのデータベースを持つツールによりベースラインが確立されることが理想的です。

4. サードパーティソフトウェアのアセスメント

この必須のステップは、企業が入手する全てのパッケージに関してライセンスの監査が行われます。

この場面ではアウトソースした物や契約企業から入手する物も含まれ、またソフトウェアの評価時に入手した物や、サードベンダーからの購入、オープンソフト、ステップ 2 で事前認証したパッケージも含まれます。

ステップ 3 で行った組織で既にベースラインが確立された監査と同様に、ステップ 1 で決められたライセンスポリシーが必要になります。ここでは以前のステップ同様、ライセンスポリシーが登録された自動ツールを使用することが理想的です。この時に事前認証されたパッケージデータベースが更新される可能性があります。

5. 定期的なソフトウェアのスキャン

このステージは一般的ですが、もしステップ 6 やステップ 7 による自動ライブラリチェックインやリアルタイムによる予防的なアセスメントを行えば実施しなくとも良いかもしれません。

定期的に行うソフトウェア監査は一週間おき、もしくは一カ月おき等の事前に決定された間隔で実行されるのがベストです。長いインターバルは監査時間を増加させます(新たなソフトのアセスメントはさらなる時間が必要なため)。また違反した内容を修正するコストは、開発が進めば進むほど、大きくなってしまいます。

また、ライセンスポリシーとパッケージ認証データベースにリンクした自動的なツールはここでも大変有効です。インテリジェントなツールは以前に解析された全てのソフトウェアを比較して差分のみの検出が可能で、解析時間の短縮が可能です。

自動的なオープンソース管理ツールを使用している組織では、定期的なソフトウェアオーディットが可能になります。

6. リアルタイムライブラリチェックイン

この任意のステップはライセンス義務に照らし合わせて、ソースコード管理システムにコミット(チェックイン)された全ての内容をチェックします。

ライブラリチェックインアセスメントは会社の製品に含まれるオープンソースをほぼリアルタイムで見つけることが可能になります。決められたオープンソースポリシーからの違反が発見され、この段階で修正されることは、下流工程での必要なアクションに対して、時間とコストの節約になります。このステップを行うにはソフトウェアライブラリシステム、ステップ 1 でのライセンスポリシーとステップ 2 での事前認証されたデータベースにリンクした自動監査ツールが必要となります。

ライブラリにチェックインした物に対してのポリシー違反はチェックインした人に対して自動的に通知されます。スキャンされたレポートは検査され、全ての違反が除去され事前認証されたパッケージもしくはさらなる検査もしくは適切にタグが付けられます。

7. リアルタイム自動スキャン

この任意のステップは開発者の PC でライセンスコンプライアンスを確実にするために有効です。

この手続きはそれが開発と開発者を邪魔すること無しに、バックグラウンドで自動的に実行されることが必要です。開発者は開発中にオープンソースフォージウェブサイトからのパッケージ、USB などのストレージメディア等の内容にアクセスし、ソースコードを取得します。Protecode 社のデベロッパーアシスタントと呼ばれる自動ツールを使用すれば、開発者のワークステーションに統合され、オープンソースのコピー、編集を検出します。ステップ 1 での組織でのライセンスポリシーやステップ 2 での事前認証パッケージデータベースとの連携により、可能性のある違反は全て開発者に直ぐに警告されます。問題を修正し、もし組織のポリシーが許すならばコメントを入れて開発を続けることができます。またログが自動生成されますので、プロジェクトマネージャによる将来的な参照もしくは支援の為のレビューが可能となります。

8. リリース前のソフトウェアアセスメント

この必要なステップはマーケットにリリースされる直前に製品に関連した内容と義務の完全な理解を確実にするステップです。

このステップはリリース時の製品品質チェックリストもしくはゲートに含まれつつあります。最終ビルドプロセスと連携した自動ツールはソフトウェアパッケージの正確なリストとリリース可能な製品コンテンツを供給する為のタスクを単純化します。どのようにパッケージが使用されるか、と言ったライセンス義務に関するアクションリストを含んだ最終的な製品リストはリリース時のチェックリストを完全な物にします。ライセンスの非互換や暗号化の内容等、特定の地域での配布制限はこのステージで明確にされるべきです。

全ての上記 8 ステップが完全なオープンソースソフトウェア管理プロセスを実現します。ここで重要な事は正確で効率的にコンプライアンスを守るためにポリシーを確立し、プロセスを最大限自動化することです。

Protecode の IP 管理ソリューション

Protecode のソリューションにより、オープンソース管理プロセスは、経済的に管理できるプロセスになり、サードパーティソフトウェアの適用を効率的に安全にします。Protecode の先進的なライセンス検出とレポート機能により、最も良いオープンソースのソリューションをライセンス義務違反すること無しに、開発者が自由に選択可能になります。このプロセスは、全ての企業ソフトの IP の問題に関して多くのトレーニングを開発者にしていることなく、開発プロセスに影響すること無しに、信頼性が高くセキュアな知的財産コンプライアンスを、開発者の開発プロセスを通して容易に適用可能にします。

| OSSAP ステップ | Protecode System 4™ コンポーネント |
|-------------------------|-------------------------------------|
| 1.ソフトウェアライセンスポリシーの確立、設定 | Enterprise Server |
| 2.ソフトウェアパッケージの事前認証 | Code Administrator |
| 3.既存のコードに対するアセスメント | Enterprise Analyzer |
| 4.サードパーティソフトウェアのアセスメント | Enterprise Analyzer |
| 5.定期的なソフトウェアのスキャン | Enterprise Analyzer |
| 6.リアルタイムライブラリチェックイン | Library Auditor |
| 7.リアルタイム自動スキャン | Developer Assistant |
| 8.リリース前のソフトウェアアセスメント | Build Analyzer |