OPENCHAIN

Reference Training Slides

Open Source Training per ISO 5230:2020

Queste diapositive seguono la legge statunitense. Diverse giurisdizioni legali possono avere requisiti legali diversi

Queste diapositive non contengono consigli legali

Cosa sono le diapositive di riferimento di OpenChain?

OPENCHAIN

- Il progetto OpenChain definisce i requisiti chiave di un programma di Open Source Compliance di qualità
- Il requisiti sono descritti nello standard internazionale per la Open Source Compliance ISO 5230:2020
- Queste diapositive di training aiutano le aziende a soddisfare i requisiti dello standard internazionale.
- Queste diapositive aiutano le aziende a soddisfare i requisiti inerenti la Specification Section 2.0. Possono essere utilizzati anche per la formazione generale sulla compliance.

Per saperne di più: https://www.openchainproject.org

Contenuti

OPENCHAIN

1. Cos'è la Proprietà Intellettuale?
2. Introduzione alle Licenze Open Source
3. Introduzione alla Compliance Open Source
4. Concetti chiave del software per la revisione Open Source
5. Esecuzione di una revisione Open Source
6. Gestione della Compliance End-to-End (Processo di esempio)
7. Evitare le insidie della Compliance
8. Linee guida per gli sviluppatori

CAPITOLO 1

Cos’è la Proprietà Intellettuale?

Cos'è la "Proprietà Intellettuale"?

- Copyright: protegge i lavori d'autore originali
  - Protegge l'espressione (non l'idea sottostante)
  - Sono soggetti al copyright software, libri e lavori simili
- Brevetti: invenzioni utili che sono innovative e non ovvie
  - Monopolio limitato per incentivare l'innovazione
- Trade secrets: protegge preziose informazioni riservate, confidenziali
- Trademarks: protegge i marchi (parola, logo, slogan, colore, etc.) che identificano la sorgente del prodotto
  - Protezione dei consumatori e del marchio; evita la confusione dei consumatori e la diluizione del marchio

Questo capitolo si concentrerà su copyright e brevetti,
le aree più rilevanti per l'Open Source Compliance.

Concetti di Copyright nel software

OPENCHAIN

- Regola base: il copyright protegge i lavori creativi
- Il copyright generalmente si applica a lavori letterari, come libri, film, immagini, musica, mappe
- Il Software è protetto dal copyright
  - Non la funzionalità del software (che è protetta dai brevetti) ma l'espressione (creatività nell'implementazione)
  - Nella protezione sono inclusi il codice sorgente ed il codice binario
- Il possessore del copyright ha il controllo solo sull'opera che ha creato, non su qualsiasi altra creazione indipendente
- La violazione può verificarsi se si copia senza il permesso dell'autore dell'opera protetta da copyright.

Diritti di Copyright più rilevanti per il Software

- Il diritto di riprodurre il software – facendone delle copie
- Il diritto di creare ‘opere derivate’ – facendo delle modifiche
  - Il termine ‘opera derivata’, o derivative work, deriva da US Copyright Act
  - Si tratta di un "termine d'arte" nel senso che ha un significato basato sullo status e non è strettamente legato alla definizione del dizionario.
  - In generale fa riferimento ad una nuova opera basata su un’opera originale al quale è stato aggiunto del lavoro così creativo da renderlo una nuova opera originale e non una semplice copia.
- Il diritto di distribuirlo
  - La distribuzione è generalmente vista come la fornitura di una copia del software, in formato binario o in codice sorgente, ad un'altra entità (un individuo o un'organizzazione al di fuori della tua azienda o organizzazione).

Nota: l'interpretazione di ciò che costituisce un’ «opera derivata" o una "distribuzione" è oggetto di dibattito nella comunità Open Source e all'interno dei circoli legali Open Source.

Concetti di Brevetto nel Software

OPENCHAIN

- I brevetti proteggono la funzionalità – questo può includere una metodologia di funzionamento, come un programma per computer
  - Non protegge idee astratte o leggi di natura.
- Una domanda di brevetto deve essere presentata in una specifica giurisdizione al fine di ottenere un brevetto in quel paese. Se un brevetto viene concesso, il possessore ha il diritto di impedire a chiunque di esercitare quella funzionalità.
- Altre parti che desiderano utilizzare la tecnologia possono richiedere una licenza di brevetto (che può concedere diritti di utilizzo, produzione, vendita, offerta per vendita e importazione della tecnologia)
- La violazione può verificarsi anche se altre parti creano autonomamente la stessa invenzione.

Licenze

OPENCHAIN

- Una ‘licenza’ è il modo in cui il possessore di copyright o di un brevetto concede dei permessi o diritti ad un’altra persona.
- La licenza può essere limitata a:
  - Tipi di utilizzo consentiti (commerciale / non commerciale, distribuzione, lavori derivati / da realizzare o già fatti, fabbricazione)
  - Termini esclusivi o non esclusivi
  - Ambito geografico
  - Durata perpetua o limitata nel tempo
- La licenza può avere delle condizioni sui diritti, ciò significa che è possibile ottenere la licenza solo se si rispettano determinati obblighi.
  Esempio: fornire l'attribuzione o dare una licenza di tipo reciprocal
- Può anche includere termini contrattuali riguardanti garanzie, indennizzo, supporto, aggiornamento, manutenzione

Verifica le tue conoscenze

OPENCHAIN

- Che tipo di materiali sono protetti dalla legge del copyright?
- Quali sono diritti di copyright più importanti per il software?
- Può essere richiesto il brevetto per il software?
- Quali diritti vengono concessi al possessore del brevetto?
- Se si sviluppa autonomamente il software, è possibile che si abbia bisogno di una licenza di copyright da parte di terzi per quel software? Una licenza di brevetto?

CAPITOLO 2

Introduzione alle Licenze Open Source

Licenze Open Source

- Per definizione, le licenze Open Source rendono il codice sorgente disponibile in base a termini che regolano la modifica e la redistribuzione dello stesso
- Le licenze Open Source possono introdurre condizionamenti relativi alla fornitura di attribuzioni, alla conservazione di dichiarazioni di copyright o a un'offerta scritta per rendere disponibile il codice sorgente.
- Un insieme di licenze note è quello approvato da Open Source Initiative (OSI). Queste licenze sono basate sulla loro definizione di Open Source, Open Source Definition (OSD). Una lista completa di licenze approvate da OSI è disponibile al seguente link: http://www.opensource.org/licenses/

Licenze Open Source Permissive

- Le licenze Open Source Permissive: il termine ‘permissive’ viene utilizzato spesso per descrivere le licenze Open Source che sono solo minimamente restrittive
- Esempio: BSD-3-Clause
  - La BSD license è un esempio di licenza permissiva che consente l’illimitata ridistribuzione, per qualsiasi scopo, del codice sorgente o dell’object code, purchè le note sul copyright e l’esclusione di garanzia della licenza vengano mantenute.
  - La licenza contiene una clausola che limita l'uso dei nomi dei contributori, per l'approvazione di un’opera derivata, senza autorizzazione specifica.
- Altri esempi: MIT, Apache-2.0

Reciprocità delle Licenze & Licenze Copyleft

- Alcune licenze richiedono che se vengono distribuiti i lavori derivati (o software nello stesso file, stesso programma o in altri contesti), la loro distribuzione avvenga con gli stessi termini dell'opera originale.
- Ciò fa riferimento all'effetto "copyleft" o "reciprocal"
- Esempio di reciprocità della licenza GPL 2.0:

È necessario che qualsiasi opera distribuita o pubblicata, che contenga in tutto o in parte un programma o sia derivata dallo stesso o da qualsiasi sua parte, sia concessa in licenza [...] in base ai termini di questa Licenza.

- Le licenze che includono clausole di reciprocità o Copyleft includono tutte le versioni di GPL, LGPL, AGPL, MPL e CDDL

Licenze Proprietarie o Closed Source

OPENCHAIN

- Una licenza su software proprietario (o licenza commerciale o EULA) ha delle limitazioni sull'utilizzo del software, la modifica e/o la distribuzione dello stesso
- Le licenze proprietarie sono uniche per ciascun fornitore: esistono tante varianti di licenze proprietarie quanti sono i fornitori e ciascuna deve essere valutata individualmente
- Gli sviluppatori Open Source spesso usano il termine "proprietario" per descrivere una licenza commerciale non-Open Source, anche se entrambe le licenze Open Source e proprietarie sono basate sulla proprietà intellettuale e forniscono una concessione di licenza a tale proprietà.

Altre situazioni di Licenza Non-Open Source

- Freeware – software distribuito con licenza proprietaria a costo nullo o molto basso
  - Il codice sorgente può essere o non essere disponibile e la creazione di opere derivate è solitamente limitata
  - Il software freeware di solito è completamente funzionale (nessuna funzionalità è bloccata) ed è disponibile per un uso illimitato (non si verificherà nessun blocco nei giorni di utilizzo)
  - Le licenze software freeware di solito impongono restrizioni alla copia, distribuzione e creazione di opere derivate del software, nonché restrizioni sul tipo di utilizzo (personale, commerciale, accademico, ecc.)
- Shareware - software proprietario fornito agli utenti su base di prova, per un tempo limitato, gratuitamente e con funzionalità o caratteristiche limitate
  - L'obiettivo dello shareware è dare ai potenziali acquirenti l'opportunità di utilizzare il programma e giudicarne l'utilità prima di acquistare una licenza per la versione completa del software.
  - La maggior parte delle aziende è molto diffidente nei confronti di Shareware, perché i fornitori di Shareware spesso si rivolgono alle aziende per pagamenti di licenze di grandi dimensioni dopo che il software si è diffuso liberamente all'interno delle loro organizzazioni.

Altre situazioni di Licenza Non-Open Source

- “Non-commercial” – alcune licenze hanno la maggior parte delle caratteristiche di una licenza Open Source, ma sono limitate all'uso non commerciale (ad esempio CC-BY-NC).
  - L'open source per definizione non può limitare il campo di utilizzo del software
  - L'uso commerciale è un campo di utilizzo, quindi qualsiasi restrizione impedisce che la licenza sia Open Source

Pubblico Dominio

OPENCHAIN

- Il termine **pubblico dominio** fa riferimento a software non protetto dalla legge e quindi utilizzabile dal pubblico senza richiedere una licenza.
- Gli sviluppatori possono includere con il loro software una *dichiarazione di pubblico dominio*
  - Esempio: “Tutto il codice e la documentazione in questo software è stato dedicato al pubblico dominio dagli autori.”
  - La dichiarazione di pubblico dominio non è la stessa di una licenza Open Source
- Una dichiarazione di pubblico dominio tende a rinunciare o ad eliminare ogni diritto di proprietà intellettuale che gli sviluppatori potrebbero avere sul software. Questo con il fine di rendere chiaro che il software possa essere usato senza alcuna restrizione. Ma l’applicabilità di queste dichiarazioni è ogetto di controversie nella comunità Open e la sua efficacia varia da giurisdizione in giurisdizione.
- Spesso la dichiarazione di pubblico dominio è accompagnata da altri termini, come le esclusioni di garanzia; in tali casi, il software può essere considerato sotto licenza piuttosto che di pubblico dominio

Compatibilità delle Licenze

- La compatibilità della licenza è il processo per garantire che i termini della licenza non siano in conflitto.
- Se una licenza richiede di fare qualcosa e un'altra vieta di farlo, le licenze sono in conflitto e non sono compatibili se la combinazione dei due moduli software fa scattare gli obblighi previsti da una licenza.
- GPL-2.0 ed EPL-1.0 estendono i loro obblighi ai lavori derivati che verranno distribuiti.
  - Se un modulo della GPL-2.0 è combinato con un modulo della EPL-1.0 ed il modulo risultante viene distribuito, allora il modulo dev
    - (secondo la GPL-2.0) essere distribuito solo con licenza GPL-2.0, e
    - (secondo la EPL-1.0) solo con licenza EPL-1.0.
  - Il distributor non può soddisfare entrambe le condizioni e per questo il modulo non potrà essere distribuito.
  - Questo è un esempio di *incompatibilità delle licenze.*

La definizione di «opera derivata" è soggetta a diversi punti di vista nella comunità Open Source e la sua interpretazione giuridica può variare da giurisdizione a giurisdizione.

Notices

OPENCHAIN

Gli avvisi, come il testo nei commenti nelle intestazioni dei file, spesso forniscono informazioni sull'autore e sulla licenza. Le licenze Open Source possono anche richiedere il posizionamento di avvisi all'interno o accanto al codice sorgente o alla documentazione per dare credito all'autore (un'attribuzione) o per chiarire che il software include modifiche.

- **Copyright notice** – un identificatore posto sulle copie dell'opera per informare della proprietà del copyright. Esempio: Copyright © A. Person (2016)
- **License notice** – un avviso che specifica e riconosce i termini e le condizioni Open Source della licenza inclusi nel prodotto.
- **Attribution notice** – un avviso incluso nella versione del prodotto che riconosce l'identità degli autori originali e / o degli sponsor dell'Open Source incluso nel prodotto.
- **Modification notice** – un avviso che hai apportato modifiche al codice sorgente di un file, come l'aggiunta della tua nota di copyright all'inizio del file.

Multi-Licensing

- La concessione di licenze multiple si riferisce alla pratica di distribuire il software in base a due o più diversi gruppi di termini e condizioni contemporaneamente
    - Ad esempio, quando il software è "con doppia licenza", il proprietario del copyright offre a ciascun destinatario la scelta di due licenze
- Nota: questo non deve essere confuso per le situazioni in cui un licensor impone più di una licenza e devi rispettarle *tutte.*

Verifica le tue conoscenze

OPENCHAIN

- Cos'è una licenza Open Source?
- Quali sono le limitazioni tipiche di una licenza Open Source permissiva?
- Elenca i nomi di alcune licenze Open Source permissive.
- Cosa significa 'reciprocità' di una licenza?
- Elenca i nomi di alcune licenze copyleft-style.
- Cosa è neessario fare per distribuire il codice con licenza copyleft?
- Freeware e Shareware software sono considerati Open Source?
- Cos'è una multi-license?
- Quali informazioni si possono trovare nei Notice Open Source e come possono essere utilizzati?

CAPITOLO 3

Introduzione all'Open Source Compliance

Obiettivi dell'Open Source Compliance

OPENCHAIN

- **Conoscere gli obblighi.** Dovrebbe esserci un processo per identificare e tenere traccia dei componenti Open Source presenti nel software.

- **Soddisfare gli obblighi di licenza.** Il processo dovrebbe essere in grado di gestire gli obblighi di licenza Open Source che derivano dalle business practices dell'organizzazione.

Quali obblighi di conformità devono essere soddisfatti?

OPENCHAIN

A seconda delle licenze Open Source coinvolte, gli obblighi di conformità possono consistere in :

- **Attribuzione e Notice.** Potrebbe essere necessario fornire o conservare il copyright e il testo della licenza nel codice sorgente e / o nella documentazione del prodotto o nell'interfaccia utente, in modo che gli utenti a valle conoscano l'origine del software e i loro diritti in base alle licenze. Potrebbe anche essere necessario fornire notice relativi a modifiche o copie complete della licenza.
- **Disponibilità del codice sorgente.** Potrebbe essere necessario fornire il codice sorgente per il software Open Source, per le modifiche apportate, per il software combinato o collegato e gli script che controllano il processo di compilazione.
- **Reciprocità.** Potrebbe essere necessario mantenere versioni modificate o lavori derivati con la stessa licenza che regola il componente Open Source.
- **Altri termini.** La licenza Open Source può limitare l'uso del nome o del marchio del titolare del copyright, può richiedere che versioni modificate utilizzino un nome diverso per evitare confusione o può terminare in caso di violazione.

Problemi di conformità Open Source: Distribuzione

OPENCHAIN

- Diffusione del materiale ad un ente esterno
  - Applicazioni scaricate sul computer o dispositivo mobile di un utente
  - JavaScript, web client, or other code that is downloaded to the user's machine
- Per alcune licenze Open Source, l'accesso tramite computer network può essere un evento "trigger"
  - Alcune licenze definiscono l'evento trigger per includere l'autorizzazione all'accesso al software in esecuzione su un server (ad esempio, tutte le versioni di Affero GPL se il software viene modificato) o nel caso di "utenti che interagiscono con esso in remoto tramite computer network"

Problemi di conformità Open Source: Modifica

OPENCHAIN

- Modifiche al programma esistente (ad esempio, aggiunte, eliminazioni di codice in un file, combinazione di componenti insieme)
- In alcune licenze Open Source, le modifiche possono comportare obblighi aggiuntivi alla distribuzione, come ad esempio:
  - Fornire un avviso (notice) delle modifiche
  - Fornire il codice sorgente
  - Le modifiche avranno la stessa licenza del componente Open Source

Programma di conformità Open Source

OPENCHAIN

Le organizzazioni risultate conformi alle regole Open Source hanno creato i propri programmi di conformità Open Source (costituiti da politiche, processi, formazione e strumenti) per:

1. Facilitare l'uso efficace dell'Open Source nei loro prodotti (commerciali o altro)
2. Rispettare i diritti dello sviluppatore / proprietario Open Source e rispettare gli obblighi di licenza
3. Contribuire e partecipare alle comunità Open Source

Implementazione delle pratiche di compliance

OPENCHAIN

Preparare processi aziendali e personale sufficiente da gestire:

- Identificazione dell'origine e della licenza di tutto il software interno ed esterno
- Monitoraggio del software Open Source all'interno del processo di sviluppo
- Esecuzione della Open Source Review e identificazione degli obblighi di licenza
- Adempimento degli obblighi di licenza quando il prodotto viene distribuito
- Supervisione per il programma di Open Source Compliance, creazione di policy e decisioni di conformità
- Training

Vantaggi derivanti dalla compliance

OPENCHAIN

I vantaggi di un solido programma di Open Source Compliance includono:

- Maggiore comprensione dei vantaggi dell'open source e del modo in cui influisce sulla tua organizzazione
- Maggiore comprensione dei costi e dei rischi associati all'utilizzo dell'Open Source
- Maggiore conoscenza delle soluzioni Open Source disponibili
- Riduzione e gestione del rischio di violazione, maggiore rispetto delle licenze Open Source scelte da sviluppatori / proprietari
- Favorire le relazioni con la comunità Open Source e le organizzazioni Open Source

Verifica le tue conoscenze

OPENCHAIN

- Cos'è l' Open Source Compliance?
- Quali sono i due obiettivi del programma di Open Source Compliance?
- Elenca e descrivi importanti pratiche commerciali di un programma di Open Source Compliance.
- Quali sono alcuni vantaggi di un programma di Open Source Compliance?

CAPITOLO 4

Concetti chiave del software per la revisione Open Source

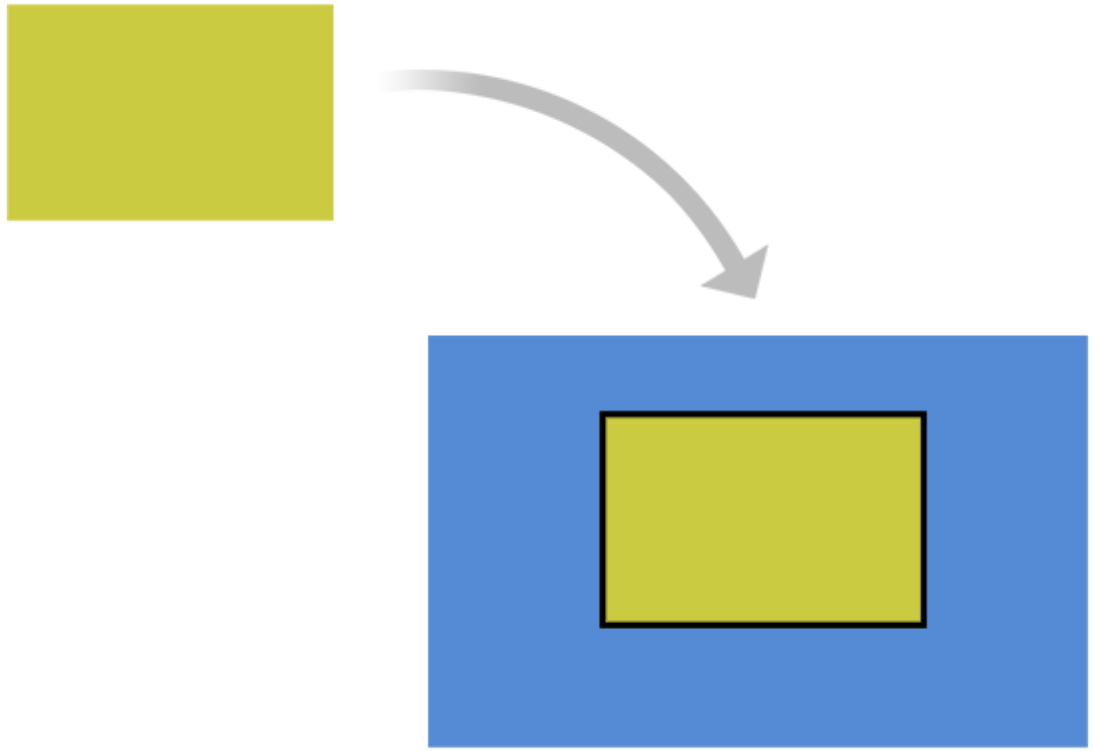
Come utilizzare un componente Open Source?

Gi scenari comuni includono:

- Incorporazione
- Collegamento
- Modifica
- Traslazione/traduzione

Incorporazione

Uno sviluppatore può copiare porzioni di un componente Open Source nel tuo prodotto.



I termini più pertinenti includono:

- Integrare
- Fondere
- Incollare
- Adattare
- Inserire

Collegamento

Uno sviluppatore può collegare o unire un componente Open Source con il tuo prodotto software.

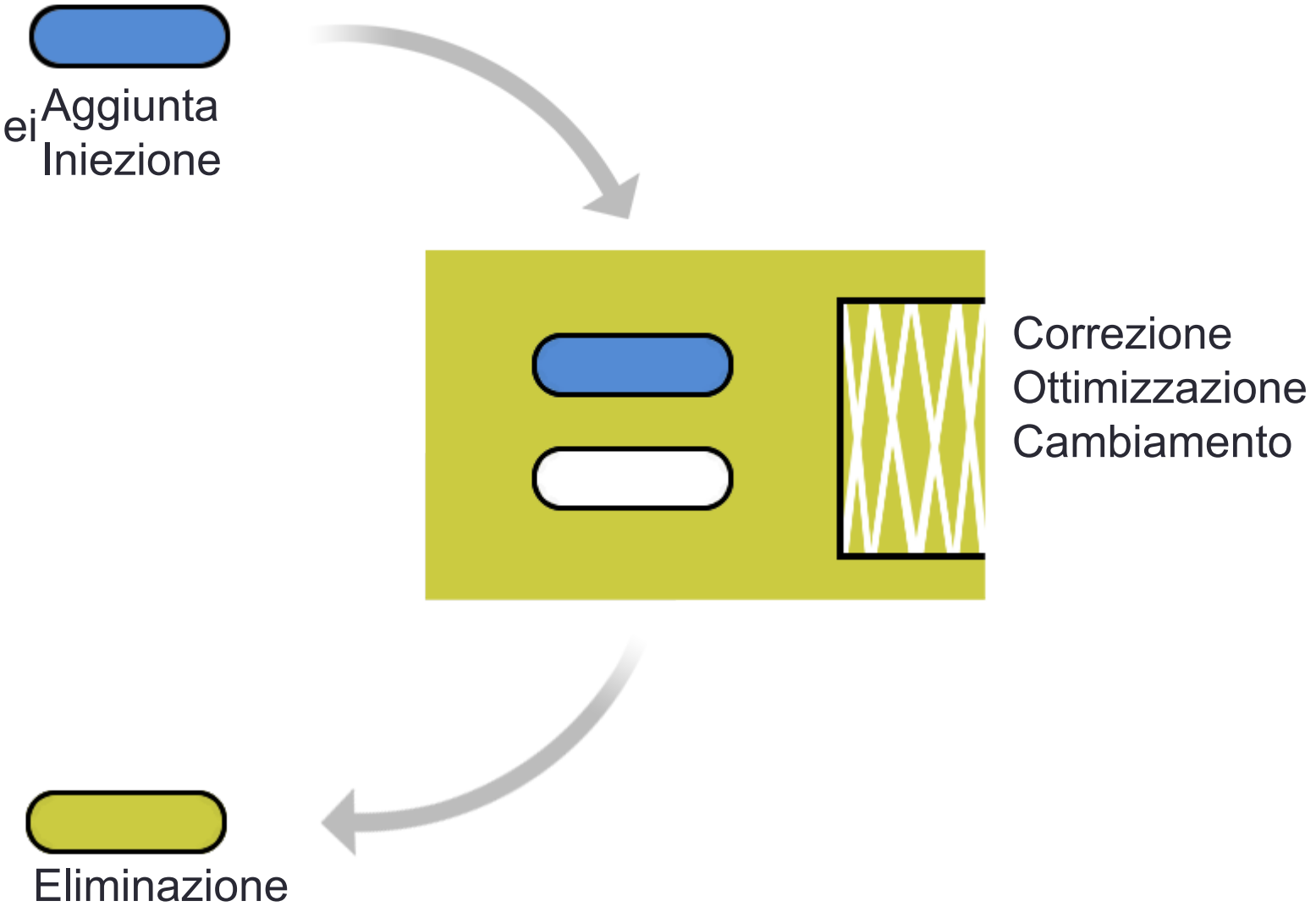
I termini più pertinenti includono :

- Collegamento static/dinamico
- Accoppiamento
- Combinazione
- Utilizzo
- Packaging
- Creazione di interdipendenza

Modifica

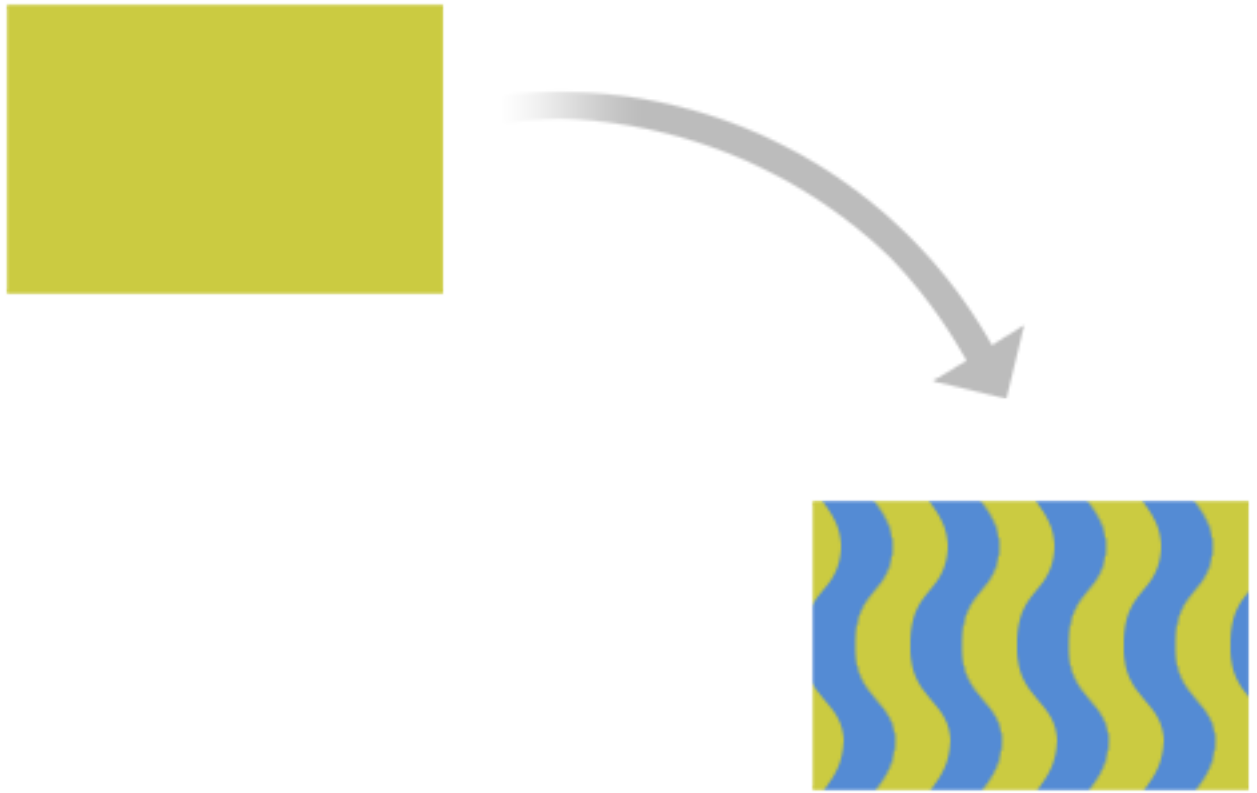
Uno sviluppatore può apportare dei cambiamenti alla componente Open Source. Questo include:

- Aggiunta / iniezione di nuovo codice nel componente Open Source
- Correggere, ottimizzare o apportare modifiche al componente Open Source
- Eliminare o rimuovere codice



Traduzione

Uno sviluppatore può modificare il codice facendolo passare da uno stato ad un altro.



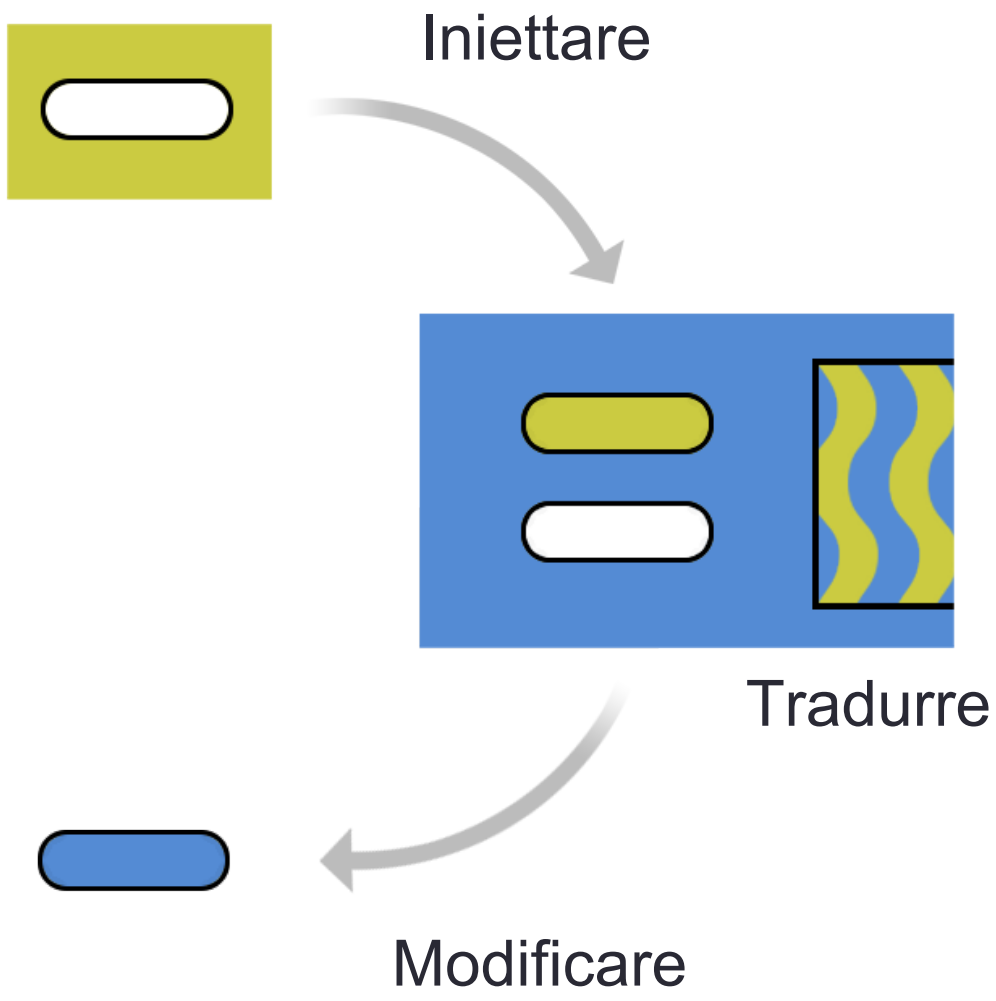
Esempi:

- Traduzione dal cinese all'inglese
- Conversione da C++ a Java
- Compilazione in binario

Strumenti di sviluppo

Gli strumenti di sviluppo possono eseguire alcune di queste operazioni dietro le quinte.

Ad esempio, uno strumento può iniettare parti del proprio codice nell'output del tool.



Come viene distribuito il componente Open Source?

OPENCHAIN

- Chi riceve il software?
  - Cliente / Partner
  - Community
  - Un'altra persona giuridica all'interno del gruppo aziendale (può essere considerata come distribuzione)

- Quale formato per la consegna?
  - Codice Sorgente
  - Binario
  - Pre-caricamento su un dispositivo

Verifica le tue conoscenze

OPENCHAIN

- Cos’è l’incorporazione?
- Cos’è il collegamento?
- Cos’è la modifica?
- Cos’è la traslazione/traduzione?
- Quali fattori sono importanti nella valutazione di una distribuzione?

CAPITOLO 5

Esecuzione di una review Open Source

Open Source Review

OPENCHAIN

- Dopo l’avvenuta gestione del programma e del prodotto e dopo che gli ingegneri hanno esaminato l'utilità e la qualità delle componenti Open Source proposte, dovrebbe essere avviata una revisione dei diritti e degli obblighi associati all'uso delle componenti selezionate
- L’elemento chiave di un programma di Open Source Compliance è un processo di revisione Open Source. Con questo tipo di processo l'azienda può analizzare il software Open Source utilizzato e comprenderne i diritti e gli obblighi
- Il processo di revisione Open Source include i seguenti step:
  - Raccolta di informazioni pertinenti
  - Analisi e comprensione degli obblighi derivanti dale licenze
  - Fornire una guida compatibile con la politica aziendale e gli obiettivi di business

Avviare una Open Source Review

Chiunque lavori con l'Open Source in azienda dovrebbe essere in grado di avviare una Open Source Review, inclusi i Program Managers o i Product Managers, gli Ingegneri ed i Legali.

Nota: Il processo spesso inizia quando un nuovo software basato su Open Source viene selezionato da tecnici o fornitori esterni.

Quali informazioni c'è bisogno di raccogliere?

OPENCHAIN

Quando si analizza l'utilizzo dell'Open Source, è utile immagazzinare informazioni relative all'identità del componente Open Source, le sue origini è su come il componente Open Source sarà usato. Possono essere inclusi:

- Nome del Package
- Status della community attorno al package (attività, appartenenza diversificata, reattività)
- Versione
- URL da dove eseguire il Download o URL del codice sorgente
- Possessore del Copyright
- Licenza e relative URL
- Attribution, notice e relative URL
- Descrizione delle modifiche che si intendono apportare

- Lista delle dipendenze
- L'uso che si intende fare della componente nel prodotto che la includerà
- Il primo rilascio del prodotto che include il package
- Specificare dove sarà mantenuto il codice sorgente
- Possibili approvazioni precedenti in altri contesti

Se il fornitore è esterno:

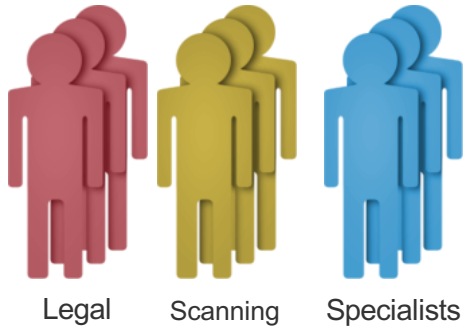
- Punto di contatto con il team di sviluppo
- Avvisi di Copyright, attribution, codice sorgente per eventuali modifiche da parte del vendor, se necessarie per soddisfare gli obblighi di licenza.

Team di Open Source Review

Il team per effettuare una Open Source Review include i rappresentanti dell'azienda che supportano, guidano, coordinano e revisionano l'uso dell'Open Source. Questi rappresentanti possono includere:

- Legali per identificare e valutare gli obblighi delle licenze
- Strumento di supporto per la scansione del codice sorgente e per identificare e monitorare l'utilizzo dell'Open Source
- Ingegneri specialisti che hanno a che fare con interessi di business, licenze commerciali, conformità all'esportazione, ecc., i quali possono essere influenzati dall'uso dell'Open Source

Analisi dell'utilizzo Open Source proposto



Il team di Open Source review dovrebbe valutare le informazioni raccolte prima di fornire indicazioni sui problemi. Ciò può includere la scansione del codice per confermare l'accuratezza delle informazioni.

Il team di Open Source review dovrebbe considerare quanto segue:

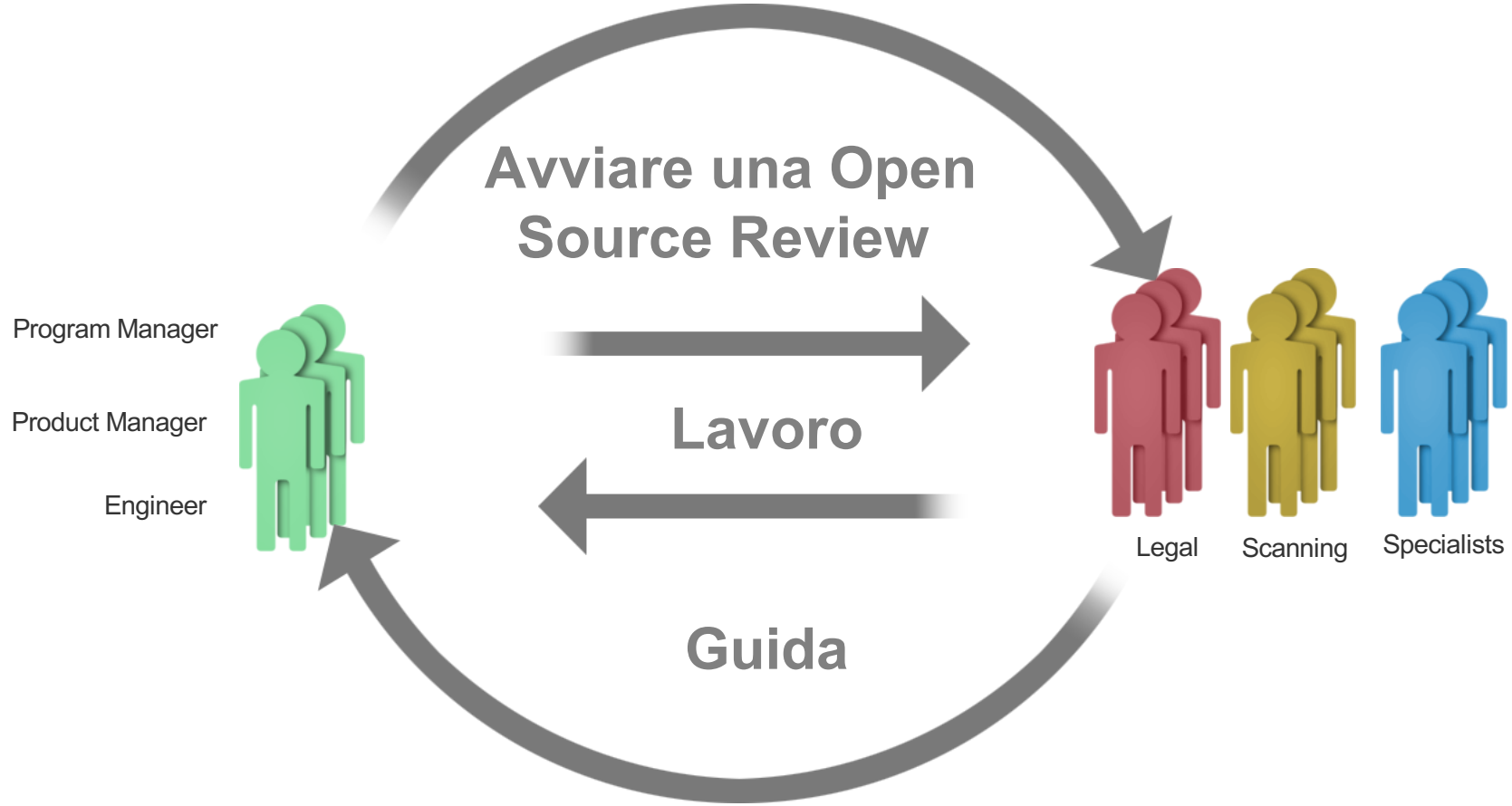
- Il codice e le informazioni associate sono completi, coerenti e accurati?
- La licenza dichiarata corrisponde a ciò che è contenuto nel codice?
- La licenza consente l'utilizzo con altri componenti del software?

Tool di scansione del codice sorgente

OPENCHAIN

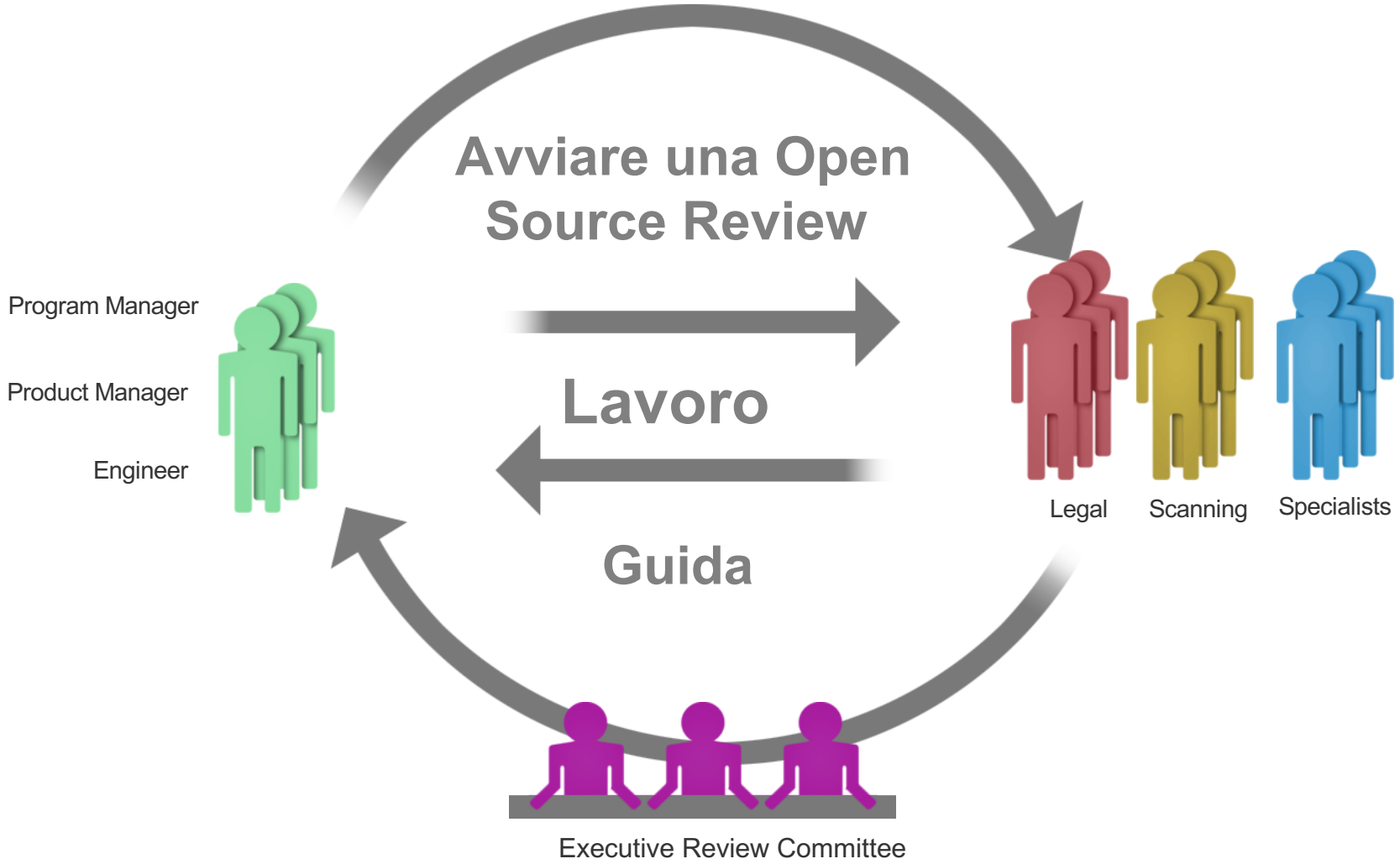
- Ci sono differenti tool automatizzati ed adatti alla scansione del codice sorgente.
- Tutte le soluzioni soddisfano esigenze specifiche e, per questo motivo, nessuna risolverà tutte le possibili sfide
- Le aziende scelgono la soluzione più adatta alla loro specifica area di mercato e prodotto
- Molte aziende utilizzano tool automatizzati e revisioni manuali
- Un buon esempio di strumento di scansione del codice sorgente disponibile gratuitamente è FOSSology, un progetto approvato dalla Linux Foundation: https://www.FOSSology.org

Lavorare con l'Open Source Review



Il processo di revisione Open Source è trasversale a molte discipline ed include i team di ingegneria, business e legali. Dovrebbe essere interattivo per garantire che tutti questi gruppi comprendano correttamente i problemi e possano creare una guida chiara e condivisa.

Supervisione dell'Open Source Review



Il processo di revisione Open Source dovrebbe avere una supervisione che, in fase di esecuzione, risolva i conflitti e approvi le decisioni più importanti.

Verifica le tue conoscenze

OPENCHAIN

- Qual è lo scopo dell'Open Source Review?
- Qual è la prima azione da intraprendere se si desidera utilizzare componenti Open Source?
- Cosa dovresti fare se hai una domanda sull'uso dell'Open Source?
- Che tipo di informazioni potresti raccogliere per una revisione Open Source?
- Quali informazioni aiutano ad identificare chi concede in licenza il software?
- Quali informazioni aggiuntive sono importanti quando si esamina un componente Open Source che proviene da un fornitore esterno?
- Quali misure possono essere intraprese per valutare la qualità delle informazioni raccolte in una Open Source review?

CAPITOLO 6

Gestione della Compliance End to End (Processi di Esempio)

Introduzione

- Con il termine di Compliance Management si intende far riferimento ad un insieme di azioni finalizzate alla gestione delle componenti Open Source usate nei prodotti. In alcuni casi le aziende hanno processi simili per la gestione delle componenti proprietarie. Nella specifica OpenChain le componenti Open Source sono chiamate "Supplied Software".
- Queste azioni spesso includono:
  - L'Identificazione di tutte le componenti Open Source utilizzate nel Supplied Software
  - L'Identificazione e il tracciamento di tutte le obligations che determina l'utilizzo di queste componenti.
  - La Conferma che tutte le obligations sono state o saranno soddisfatte.
- Le aziende di piccole dimensioni possono usare una semplice checklist e le aziende più grandi un processo dettagliato.

Esempio di una Checklist per piccole e medie imprese

OPENCHAIN

Attività di Compliance da effettuare continuamente:

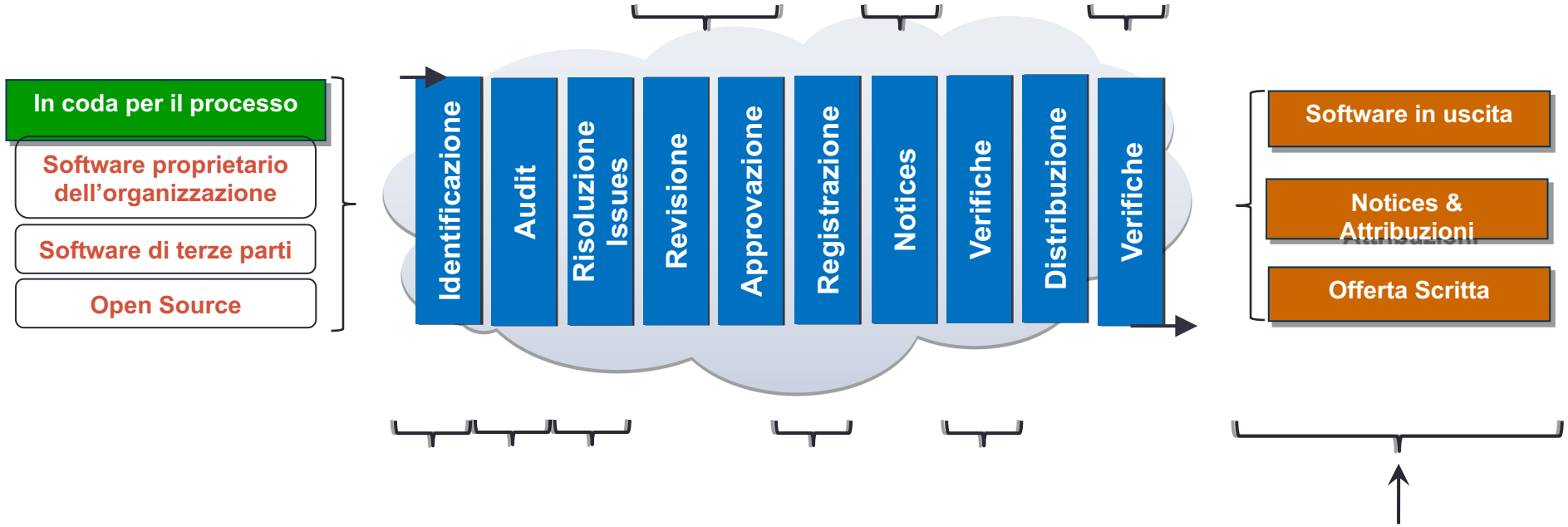
1. Individuazione precoce dell'Open Source nel ciclo di acquisizione/sviluppo procurement/development del software
2. Revisione ed Approvazione di tutte le componenti Open Source utilizzate
3. Verifica delle informazioni necessarie a soddisfare le obligations previste dall'Open Source
4. Revisione ed Approvazione di ogni contributo in uscita a progetti Open Source

Requisiti per il supporto all'attività di Compliance:

1. Costituzione di uno staff adeguato alle attività di Compliance ed individuazione chiara delle aree di responsabilità
2. Adeguamento dei Processi di Business esistenti per supportare il programma di Open Source compliance
3. Predisposizione di attività di training sulle policy Open Source dell'organizzazione accessibili da tutti
4. Tracciamento dell'avanzamento di tutte le attività di Open Source compliance

Le checklists dettagliate sono disponibili al seguente link: https://www.linuxfoundation.org/projects/opencompliance/self-assessment-compliance-checklist

Esempio di Processo in una grande impresa



Revisione ed approvazione dei record di compliance relativi alle componenti OpenSource

Redazione delle notices per la pubblicazione

Verifiche post pubblicazione

Identificazione delle componenti Open Source per revisione

Scansione o auditing del codice sorgente – e – conferma dell'origine e della licenza del codice sorgente

Risoluzione delle issues individuate dall'audit secondo quanto previsto dalla policy open source dell'organizzazione

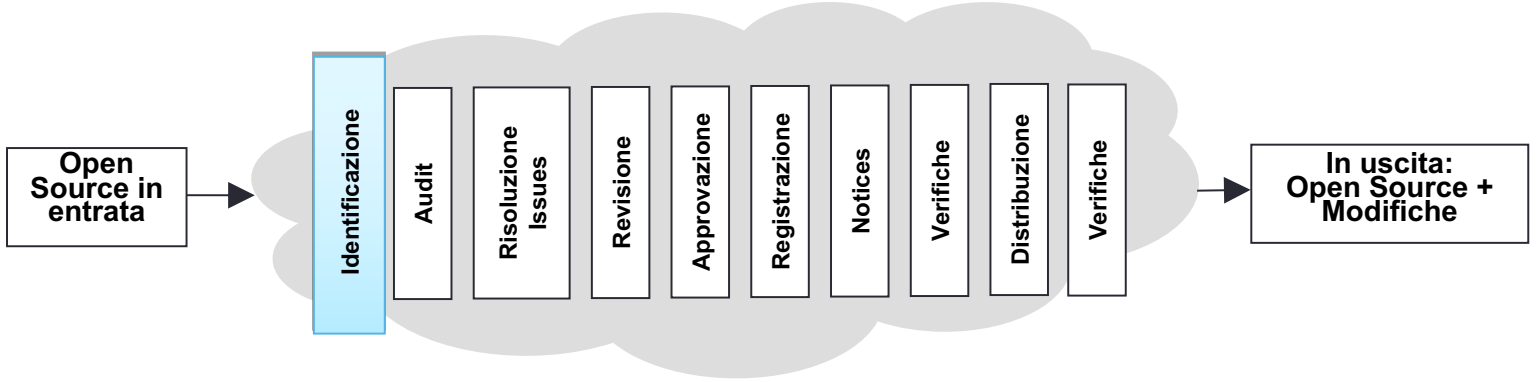
Tracciamento del software/versione approvato/a in un inventario per prodotto e per release

Verifica dei pacchetti di codice sorgente per la distribuzione – e – verifica che siano prodotte le notices appropriate

Pubblicazione del codice sorgente, delle notices e produzione di un'offerta scritta

Esempio di Processo di Compliance Management End-to-End

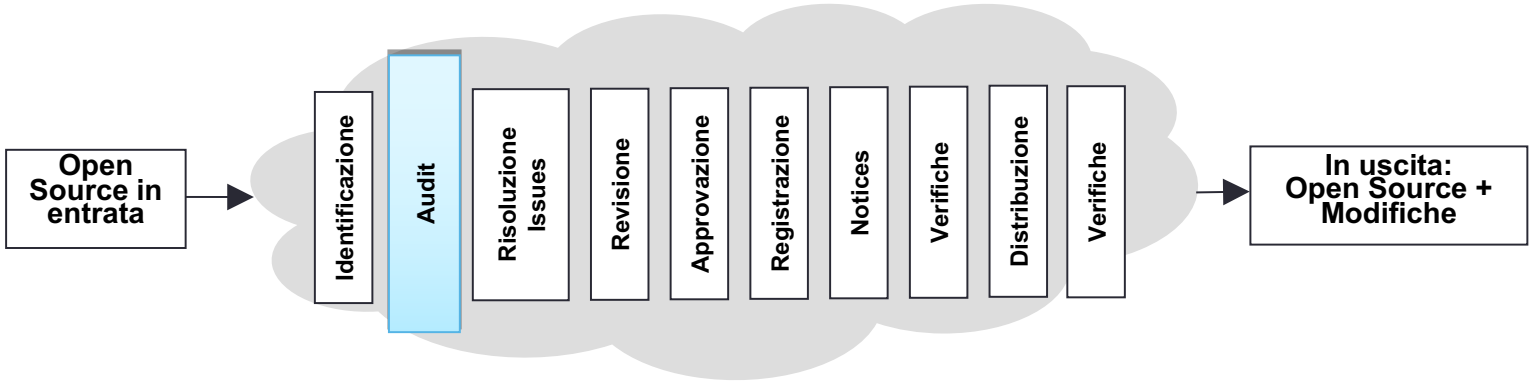
Identificazione e tracciamento dell'Open Source utilizzato



Identificazione delle componenti Open Source

- Passi:
  - Richieste in entrata dall'ingegneria
  - Scansione del software
  - Due diligence del software di terze parti
  - Riconoscimento manuale di nuovi componenti aggiunti al repository
- Risultato:
  - Produzione (o Modifica) di un record di compliance per l'Open Source
  - Richiesta di un audit per revisionare il codice sorgente con l'obiettivo di stabilire se è esaustivo o meno rispetto a quanto previsto nella policy Open Source

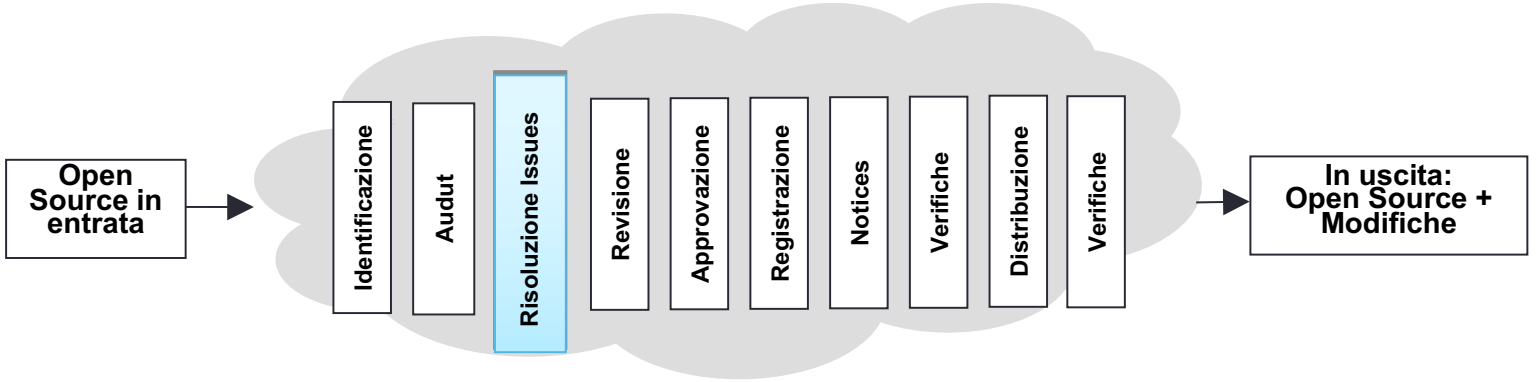
Auditing del Codice Sorgente



Identificazione ed auditing delle licenze Open Source

- Passi:
  - Identificazione del codice sorgente per l'audit
  - Eventuale scansione del codice sorgente mediante apposito software
  - Revisione e verifica degli alert provenienti dall'audit o dalla scansione e dell'origine del codice
  - Gli audit e le scansioni sono ripetuti iterativamente sulla base dei cicli di sviluppo e di rilascio del software.
- Risultato:
  - Un report che identifica:
    1. L'origine e le licenze del codice sorgente
    2. Le Issues che devono essere risolte

Risoluzione Issues

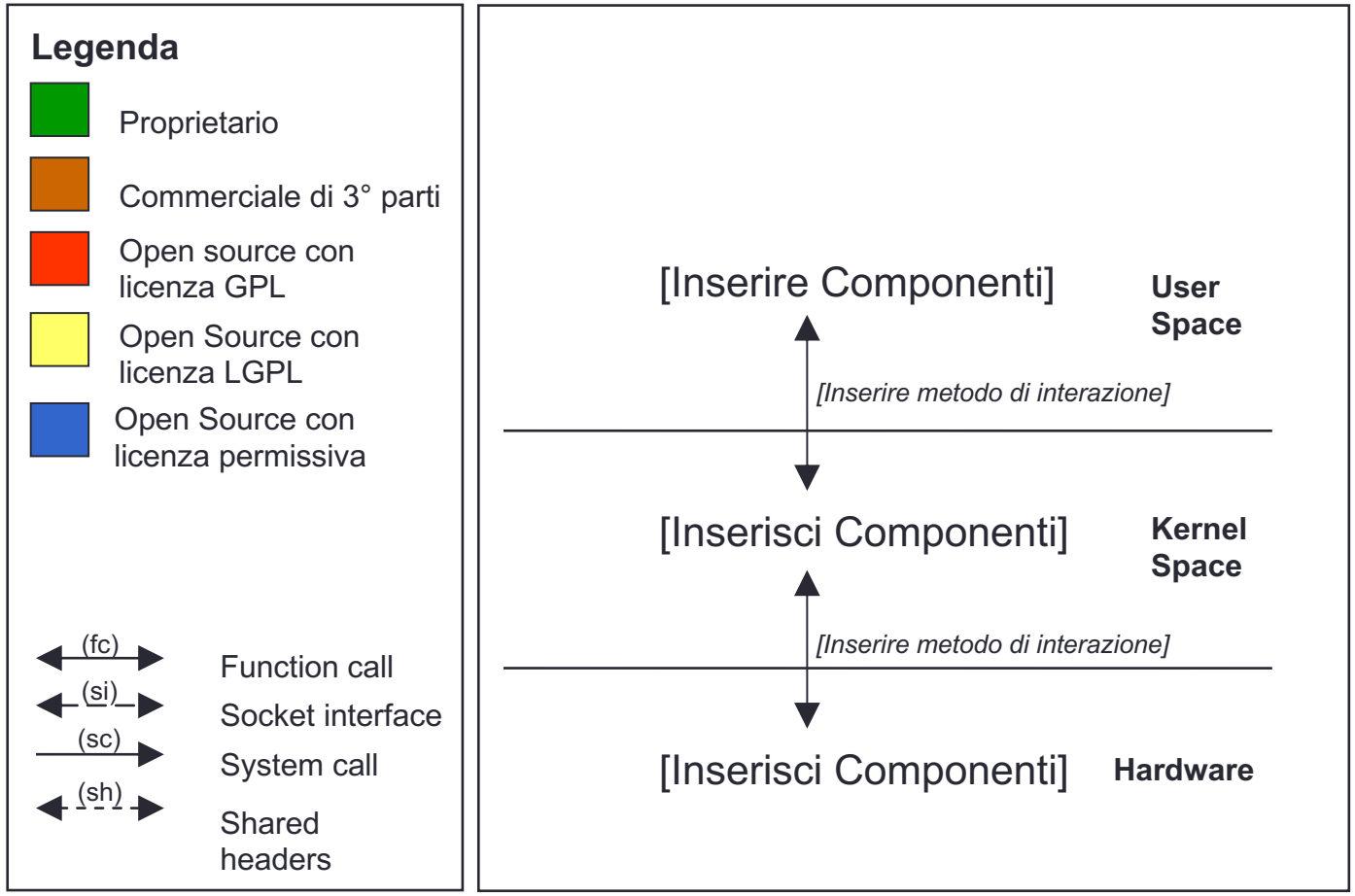


Risoluzione di tutte le issues identificate nell'audit

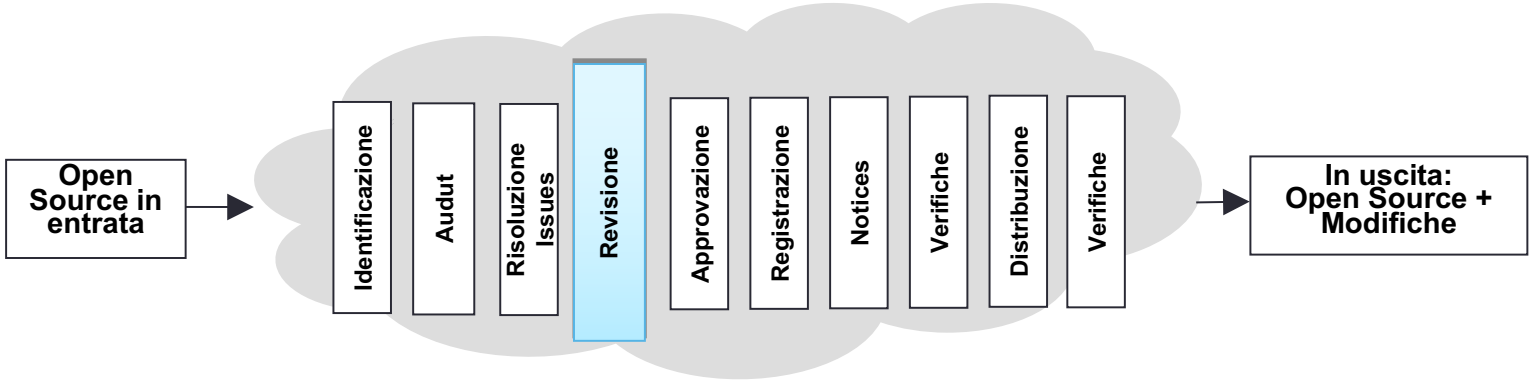
- Passi:
  - Comunicazione di feedback all'ingegneria per risolvere le issue che nel report di audit sono state indicate in conflitto con la policy Open Source
  - L'ingegneria effettua la revisione dell'Open Source presente nel codice sorgente (si veda la slide successiva per il template)
- Risultato:

  Una fix per ciascun file e per ciascuna violazione di licenza secondo quanto evidenziato nel report di audit

Revisione dell'Architettura (Template di esempio)



Revisione



Revisione delle issues risolte per confermare che la risoluzione effettuata corrisponde a quanto previsto nella policy Open Source

Passi:

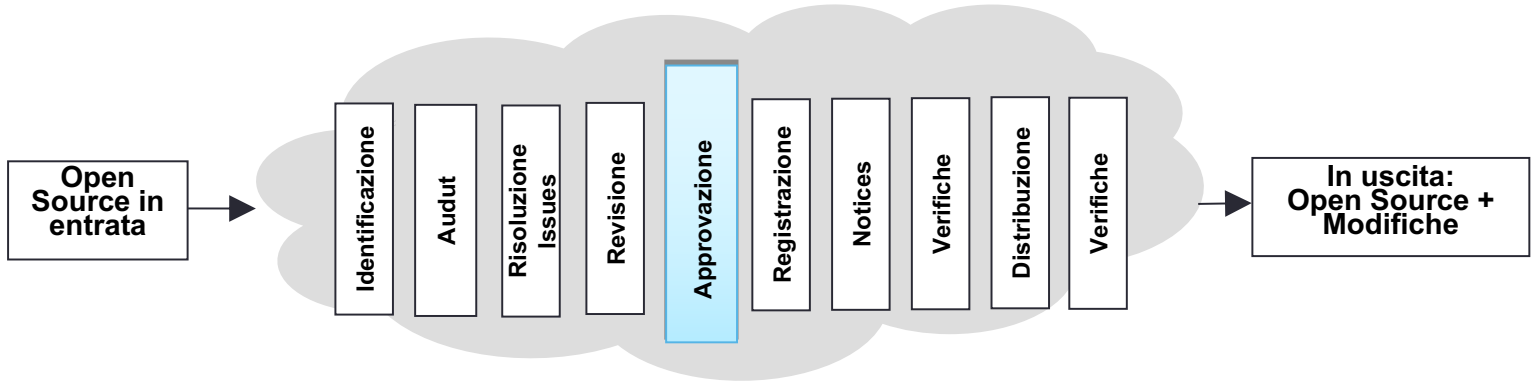
- Definizione di adeguati livelli di autorità nello staff che si occupa della revisione
- Revisione delle issues sulla base di quanto è previsto nella policy Open Source

Risultato:

- Assicurare che il software indicato nel report di audit sia conforme alle policy Open Source
- Preservare gli alert presenti nel report di audit e etichettare come “Risolte” le issues che sono pronte per lo step successivo (es: Approvazione)

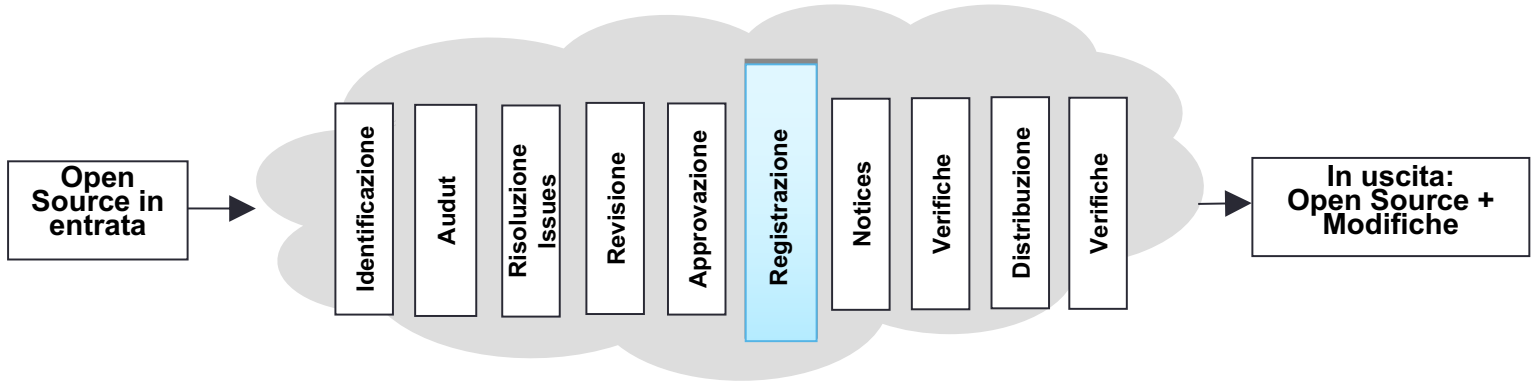
Approvazione

- Sulla base dei risultati dell'audit e della revisione effettuati precedentemente, l'utilizzo del software può essere o meno approvato.
- L'approvazione dovrebbe specificare la versione delle componenti Open Source approvate, il modello di utilizzo approvato per ciascuna componente, e qualsiasi altro obbligo applicabile in virtù della licenza Open Source utilizzata
- L'approvazione dovrebbe essere effettuata sulla base di livelli di autorità appropriati

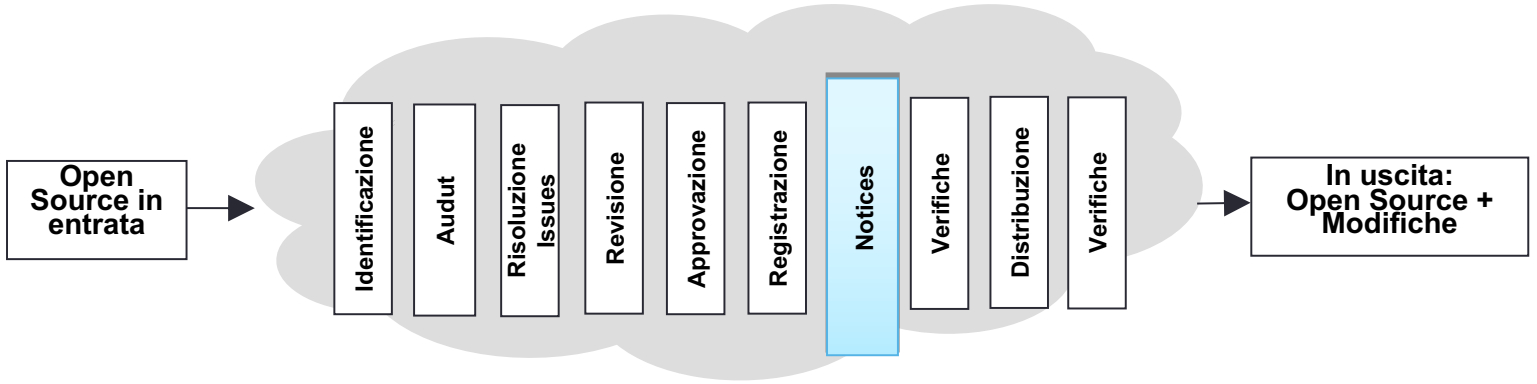


Registrazione/Tracciamento dell'Approvazione

- Una volta che una componente Open Source è stata approvata per essere utilizzata in un prodotto, questa dovrebbe essere aggiunta al software inventory per quel prodotto
- L'approvazione e le sue condizioni dovrebbero essere registrate in un sistema di tracking
- Il Sistema di tracking dovrebbe rendere evidente che è necessaria una nuova approvazione qualora si intenda utilizzare una nuova versione di una componente Open Source o qualora si proponga un nuovo modello di utilizzo

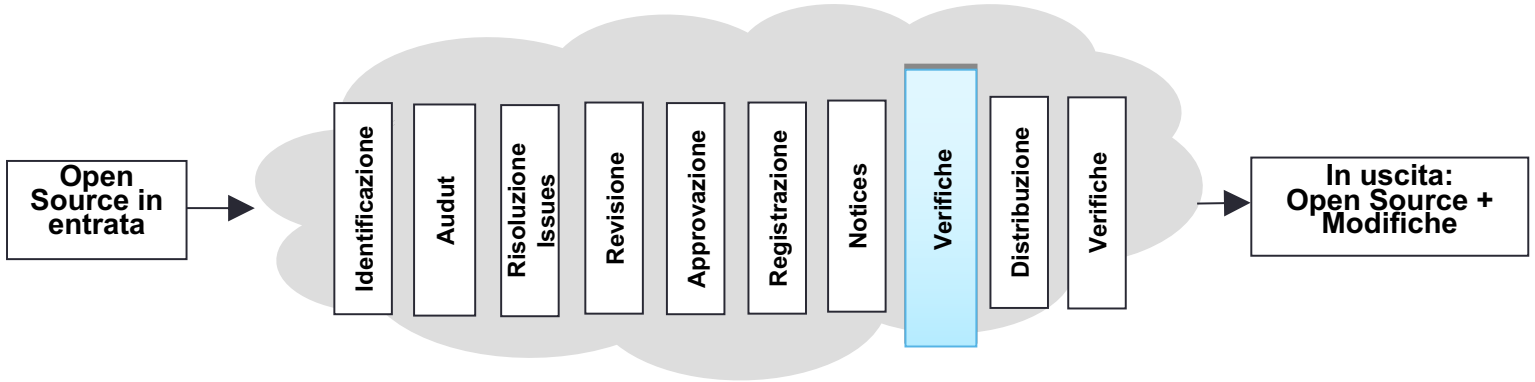


Notices



- Predisporre delle notices appropriate per ogni componente Open Source utilizzata in una release di prodotto:
  - Rendere noto l'utilizzo di software Open Source fornendo il copyright e le notices di attribuzione
  - Informare l'utente finale del prodotto su come ottenere una copia del codice sorgente Open Source (laddove applicabile, ad esempio in caso di GPL e LGPL)
  - Riprodurre per intero il testo dell'accordo di licenza del codice Open Source incluso nel prodotto se necessario

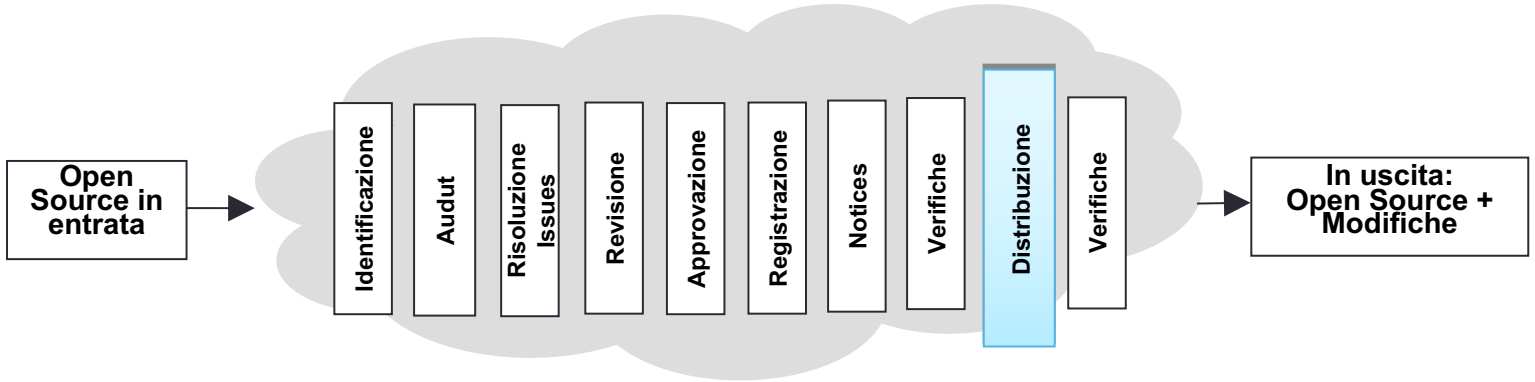
Verifiche Pre-Distribuzione



Verificare che il software da distribuire è stato revisionato ed approvato

- Passi:
  - Verificare che i pacchetti Open Source che devono essere distribuiti sono stati identificati ed approvati
  - Verificare che il codice sorgente revisionato coincide con la versione binaria presente nel prodotto
  - Verificare che tutte le notices appropriate per informare l'utente finale dei propri diritti e delle modalità con cui può richiedere il codice sorgente delle component Open Source identificate sono state incluse
  - Verificare di soddisfare gli altri obblighi identificati
- Risultato:
  - Il pacchetto software pronto per essere distribuito contiene solo sotware revisionato ed approvato
  - "Distributed Compliance Artifacts" (come definito nelle specifiche OpenChain), contiene gli appropriati file notice che sono inclusi nel pacchetto di distribuzione o negli altri metodi di delivery utilizzati

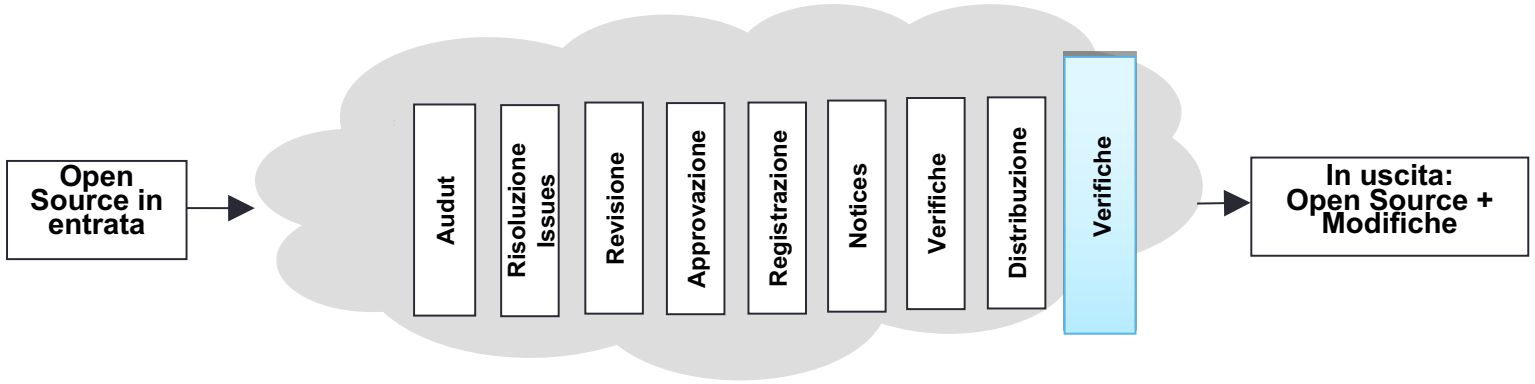
Distribuzione del codice sorgente



Se necessario fornire il codice sorgente di accompagnamento

- Passi:
  - Fornire il codice sorgente di accompagnamento insieme ai tool di build associati e alla documentazione (es: mediante l'upload su un sito web di distribuzione o incluso in un pacchetto di distribuzione)
  - Il codice sorgente di accompagnamento è identificato con etichette che indicano a quale prodotto e versione corrisponde
- Risultato:
  - Soddisfacimento dell'obbligo di fornire il codice sorgente di accompagnamento

Verifiche Finali



Verificare la conformità con gli obblighi previsti dalle licenza

- Passi:
  - Verificare che il codice sorgente di accompagnamento (se previsto) è stato caricato o distribuito correttamente
  - Verificare che il codice sorgente caricato o distribuito corrisponde alla stessa versione che è stata approvata
  - Verificare che le notices sono state adeguatamente pubblicate e rese disponibili
  - Verificare che le altre obbligazioni sono rispettate

- Risultato:
  - Verifica che tutti gli artefatti di Compliance (Verified Distributed Compliance Artifacts) sono stati adeguatamente forniti

Verifica le tue conoscenze

- Che cosa viene effettuato durante una due diligence di compliance (per il nostro processo di esempio, descrivi ad alto livello gli steps che vengono eseguiti)?
  - Identificazione
  - Audit del codice sorgente
  - Risoluzione delle issues
  - Revisione
  - Approvazione
  - Registrazione/tracciamento dell'approvazione
  - Notices
  - Verifiche Pre-distribuzione
  - Distribuzione del codice sorgente di accompagnamento
  - Verifica
- Che cosa richiede una revisione dell'architettura?

CAPITOLO 7

Evitare le insidie della compliance

Insidie della Compliance

Questo capitolo descrive alcune potenziali insidie nel processo di compliance, relative a:

1. Proprietà Intellettuale
2. Conformità alle Licenze
3. Processo di Compliance

Le insidie sulla Proprietà Intellettuale

OPENCHAIN

Tipologia & Descrizione	Come scoprirle	Come evitarle
Inserimento non pianificato di software Open Source copyleft in software proprietario o di 3° parti: Questo tipo di problema si verifica durante il processo di sviluppo quando l'ingegneria aggiunge codice Open Source nel codice sorgente, che si è stabilito essere proprietario, in conflitto con la policy Open Source.	Questo tipo di problema può essere scoperto mediante scansione o audit del codice sorgente per individuare: • Codice sorgente Open Source • Avvisi di Copyright Tool automatici di scansione del codice possono essere utilizzati per questo scopo.	Questo tipo di problema può essere evitato nei seguenti modi: • Offrendo sessioni di training all'ingegneria rispetto alle issues di compliance, ai diversi tipi di licenze Open Source e alle conseguenze che determinano l'inserimento di software Open Source nel codice sorgente proprietario. • Effettuando regolarmente scansioni del codice sorgente o effettuando audit per tutto il codice sorgente che è nell'ambiente di build.

Le insidie sulla Proprietà Intellettuale

OPENCHAIN

Tipologia e Descrizione	Come scoprile	Come evitarle
Link non pianificati tra software Open Source con licenza copyleft e codice sorgente proprietario: Questo tipo di problema si verifica quando si crea un link tra software che hanno licenze tra loro incompatibili. L'effetto da un punto di vista legale del linking è oggetto di dibattito nella community Open Source.	Questo tipo di problema può essere scoperto utilizzando dei tool di dependency tracking che individuano i link tra diversi componenti software.	Questo tipo di problema può essere evitato nei seguenti modi: 1. Offrendo sessioni di training all'ingegneria per aumentare la consapevolezza sui rischi legali ed evitare di linkare componenti software con licenze che sono in conflitto con le proprie policy Open Source 2. Utilizzando continuamente tool di dependency tracking sul proprio ambiente di build
Inserimento di codice proprietario in codice Open Source con licenza copyleft mediante modifiche al codice sorgente	Questo tipo di problema può essere scoperto utilizzando gli audit o le scansioni per identificare ed analizzare il codice sorgente che è stato inserito nel componente Open Source.	Questo tipo di problema può essere evitato nei seguenti modi: 1. Offrendo sessioni di training all'ingegneria 2. Effettuando audit regolari al codice

Insidie sulla conformità alle Licenze

Tipo & Descrizione	Come evitarle
Il codice sorgente di accompagnamento, i file di licenza o altre notices informative non sono forniti	Questo tipo di problema può essere evitato mediante l'acquisizione del codice sorgente e l'inserimento di una voce nella checklist del ciclo di rilascio del prodotto che deve essere verificata prima che il prodotto venga reso disponibile all'esterno.
La versione di codice sorgente di accompagnamento fornita non è corretta	Questo tipo di problema può essere evitato aggiungendo uno step di verifica nel processo di compliance per garantire che il codice sorgente di accompagnamento associato alla versione binaria venga reso disponibile.
È fornito un codice sorgente di accompagnamento relativo alle modifiche effettuate su un componente Open Source non corretto	Questo tipo di problema può essere evitato aggiungendo uno step di verifica nel processo di compliance per assicurare che venga pubblicato il codice sorgente delle modifiche effettuate, e non solo il codice sorgente originale del componente Open Source

Insidie sulle conformità alle Licenze

OPENCHAIN

Tipo & Descrizione

Non sono state evidenziate le modifiche fatte al codice sorgente delle componenti open source:

Non sono state evidenziate le modifiche fatte al codice sorgente delle componenti Open Source, in contrasto con quanto richiesto dalla licenza (o , sono state fornite informazioni rispetto alla modifica effettuata che hanno un insufficiente livello di dettaglio o risultano poco chiare e non soddisfano quanto previsto dalla licenza)

Come evitarle

Questo tipo di problema può essere evitato nei seguenti modi:

1. Inserire uno step di verifica da effettuare prima del rilascio del codice sorgente che prevede di evidenziare le modifiche effettuate
2. Offrire una sessione di training all'ingegneria per garantire che effettui l'update delle attestazioni di copyright o delle informazioni sulle licenze di tutte le componenti Open Source o del sofware proprietario che si ha intenzione di rilasciare come pubblico

Fallimenti del Processo di Compliance

Descrizione	Come evitarli	Come prevenire
Mancata richiesta di approvazione da parte degli sviluppatori sull'utilizzo dell'Open Source	Questo tipo di problema può essere evitato offrendo sessioni di training all'ingegneria sulle policy Open Source dell'azienda e sui processi.	Questo tipo di problema può essere evitato nei seguenti modi: 1. Effettuando periodicamente delle scansioni complete del software per individuare eventuali utilizzo «non dichiarati» dell'Open Source 2. Offrendo training al gruppo di ingegneria sulle policy Open Source e sui processi 3. Includendo la compliance nella performance review dei dipendenti
Mancata formazione sull'Open SOurce	Questo tipo di problema può essere evitato facendo in modo che il completamento della formazione sull'Open Source sia parte del piano di sviluppo professionale del dipendente, e il completamento sia monitorato ai fini della performance review	Questo tipo di problema può essere prevenuto esigendo che la formazione sull'Open Source da parte del gruppo di ingegneria venga completato entro una specifica data

Fallimenti del processo di Compliance

Descrizione	Come evitarli	Come prevenirli
Mancato controllo del codice sorgente	Questo tipo di problema può essere evitato nei seguenti modi: 1. Effettuando scansioni/audit periodici sul codice sorgente 2. Facendo in modo che l'auditing sia una milestone in un processo di sviluppo iterativo	Questo tipo di problema può essere evitato nei seguenti modi: 1. Fornendo personale adeguato per non rimanere indietro rispetto alla pianificazione 2. Effettuando audit periodici
Mancata risoluzione dei risultati dell'audit (analizzando i warning segnalati da uno strumento di scansione o audit)	Questo tipo di problema può essere evitato non consentendo che un ticket di compliance possa essere risolto (cioè chiuso) se il rapporto di audit non è finalizzato.	Questo tipo di problema può essere prevenuto implementando dei blocchi di approvazione nel Processo di Open Source compliance
Mancata richiesta di revisione dell'Open Source in maniera tempestiva	Questo tipo di problema può essere evitato avviando le richieste di revisione dell'Open Source precocemente anche se l'ingegneria non ha ancora deciso sull'utilizzo di codice Open Source	Questo tipo di problema può essere prevenuto attraverso l'educazione all'utilizzo dell'Open Source

Garantire la Compliance prima della consegna del prodotto

- Le organizzazioni devono fare il modo che la Compliance diventi una priorità prima che il prodotto (in qualsiasi forma) venga reso disponibile
- Dare priorità alla compliance promuove:
  - Un utilizzo più efficace dell'Open Source all'interno dell'organizzazione
  - Relazioni migliori con la community e le organizzazioni Open Source

Stabilire relazioni con la community

OPENCHAIN

Se un'organizzazione utilizza l'Open Source in un prodotto commerciale, la cosa migliore da fare è quella di creare e mantenere buoni rapporti con la community – in particolare, con le communities che si occupano dei progetti Open Source che l'organizzazione usa ed inserisce nei propri prodotti commerciali.

Inoltre, mantenere buoni rapporti con la community può essere utile per ricevere suggerimenti sul modo migliore per essere compliant e per ricevere aiuto se si verifica una issue di compliance.

Mantenere buoni rapporti con le software communities può essere utile per due motivi: ricevere improvements (upstreaming improvements) ed avere supporto dagli sviluppatori

Verifica le tue conoscenze

OPENCHAIN

- Quali tipi di insidie possono capitare relativamente alla Open Source Compliance?
- Fai un esempio di violazione della proprietà intellettuale
- Fai un esempio di violazione della licenze.
- Fai un esempio di violazione del processo di compliance.
- Quali sono i benefici che derivano dal dare priorità alla compliance?
- Quali sono i benefici che derivano dal mantenere buoni rapporti con la community?

CAPITOLO 8

Linee Guida per lo sviluppatore

Linee Guida per lo sviluppatore

OPENCHAIN

- Selezionare codice di alta qualità, ben supportato dalla community Open Source
- Cercare una guida
  - Richiedere approvazione formale per ogni componente Open Source che si sta utilizzando
  - Non controllare il codice non revisionato
  - Richiedere approvazione formale per contribuire a progetti Open Source esterni
- Preservare le informazioni di licenza esistenti
  - Non rimuovere o alterare in alcun modo i copyright esistenti delle licenze Open Source o altre informazioni sulla licenza da qualsiasi componente Open Source che utilizzi. Tutte le informazioni sul copyright e sulla licenza devono rimanere intatte in tutti i componenti Open Source
  - Non rinominare le componenti Open Source a meno che non sia richiesto nella licenza (es: richiedere renaming delle versioni modificate)
- Acquisire e conservare le informazioni sul progetto Open Source che sono necessarie per il processo di revisione

Anticipare i requisiti del processo di Compliance

OPENCHAIN

- Includere il tempo necessario per attuare nei piani di lavoro quanto stabilito nella policy Open Source
  - Seguire le linee guida per gli sviluppatori per l'utilizzo di software Open Source, in particolare incorporando o collegando il codice Open Source in codice sorgente proprietario o di terze parti o viceversa
  - Rivedere le architetture ed evitare di mettere insieme component con licenze Open Source tra loro incompatibili.
- Effettuare sempre verifiche di compliance – per ogni prodotto
  - Verificare la compliance prodotto per prodotto: se è approvato l'utilizzo di una componente Open Source per un prodotto, non necessariamente significa che la stessa componente sarà approvata per essere utilizzara in un secondo prodotto
- E per ogni upgrade ad una versione più nuova della componente Open Source
  - Assicurare che ogni nuova versione della stessa componente Open Source sia revisionata ed approvata
  - Quando si effettua l'upgrade ad una nuova versione di una componente Open Source è necessario assicurarsi che la nuova versione abbia la stessa licenza della vecchia (è possibile che durante un upgrade di versione vengano apportate modifiche al licensing)
  - Se si verifica un cambio di licenza in un progetto Open Source, assicurarsi che le attività di compliance siano aggiornate e che la nuova licenza non crei conflitti

Applicare il processo di Compliance a tutte le component Open Source

OPENCHAIN

- In-bound software
  - Adottare misure per capire che tipo di Open Source è incluso nel software fornito dai fornitori
  - Valutare gli obblighi da soddisfare per tutto il software che sarà incluso nei prodotti dell'organizzazione
  - Controllare sempre il codice sorgente ricevuto dai fornitori o, in alternativa, stabilire come politica aziendale che i fornitori siano tenuti a fornire sempre un rapporto di verifica del codice sorgente

Verifica le tue conoscenze

OPENCHAIN

- Indica alcune linee guida generali che gli sviluppatori possono seguire quando lavorano con l'Open Source.
- Dovresti rimuovere o alterare le informazioni di intestazione della licenza Open Source?
- Indica alcuni passi importanti nel processo di compliance.
- In che modo una nuova versione di una componente Open Source può creare nuove issues di compliance?
- Quali rischi si corrono con il software in-bound?

Approfondisci gratuitamente le Basi della Compliance per gli sviluppatori sulla pagina della Linux Foundation:
https://training.linuxfoundation.org/linux-courses/open-source-compliance-courses/ compliance-basics-for-developers