

Rendere compatibile il vostro software open source con la GPL o con un'altra licenza.

di [David A. Wheeler](#)

Publicato il 6.5.2002, revisionato il 26.4.2011

traduzione di Andrea Montagner

<http://www.dwheeler.com/essays/gpl-compatible.html>

Questo saggio sostiene che gli sviluppatori di Free-libre / open source software (FLOSS, alias OSS / FS) dovrebbero utilizzare una licenza esistente adottata in larga scala compatibile con la General Public License (GPL), in particolare le licenze GPL, LGPL, MIT / X, o nuova BSD. Sostiene inoltre che contro la proliferazione di licenze FLOSS, ove possibile, si deve selezionare solo le licenze FLOSS da quella lista (con l'eventuale aggiunta di Apache 2.0). Questo articolo è stato precedentemente stampato, con una serie di modifiche, come un articolo su OsOpinion.

1 Introduzione

Questo saggio è un appello a tutti quegli sviluppatori che stanno sviluppando [Free-libre / open source software](#) (FLOSS, alias FOSS o OSS / FS). Sviluppatori: Per favore, ove possibile, **usate per il software una licenza (standard) esistente e ampiamente utilizzata, una che sia nota per essere compatibile con la [GNU General Public License \(GPL\)](#)**. In particolare, vi prego, scegliete le licenze [GPL](#), [LGPL](#), [MIT/X](#), o la [nuova BSD](#) (o, in alternativa, la licenza Apache 2.0). Se create una vostra propria licenza – [e vi prego ardentemente di non creare una nuova licenza](#) - almeno assicuratevi che la nuova licenza sia compatibile con la GPL. Non è necessario usare la GPL - nemmeno la [Free Software Foundation \(FSF\)](#), la sviluppatrice della GPL e il suo sostenitore più accanito, afferma che assolutamente tutti i software devono essere licenziati GPL (per esempio, guardate [l'approvazione delle modifiche della licenza di Ogg Vorbis da parte di Richard Stallman della FSF](#), e notate che la FSF ha sviluppato e mantiene la [Lesser GPL](#) come licenza di compromesso). Ma assicuratevi che la vostra licenza FLOSS del software sia GPL-compatibile, cioè che [possiate combinare il vostro codice con il codice rilasciato sotto GPL in un programma più ampio](#), e scegliete una delle licenze FLOSS "standard". Il resto di questo saggio spiega perché e come farlo.

2 Le licenze incompatibili GPL rischiano la mancanza di supporto (le più popolari GPL)

Perché una licenza GPL compatibile? Perché se il progetto FLOSS non è compatibile con la GPL, c'è un rischio significativo che non riusciate a ricevere un sostegno sufficiente da altri sviluppatori per sostenere il vostro progetto.

Molti sviluppatori FLOSS preferiscono la licenza GPL, perché credono che la GPL (1) fornisca un

migliore quid-pro-quo per gli sviluppatori, (2) istituisca una collaborazione tra le persone e le società meglio dei consorzi, (3) [protegga il loro lavoro in un ambiente quotidiano meno formale](#) e/o (4) incoraggi, aumentandola, la quantità di software libero. Esempi di questo tipo di pensiero comprendono [“Freedom Fighters di Jeremy Allison”](#) (che fa notare che i miglioramenti di NetApp al codice con licenza BSD non sono un'involuzione ma derivano dai miglioramenti al codice con licenza GPL). [“Strip Mining of Open Source”](#) di ITPRO e le [affermazioni di Eben Moglen che le licenze in stile BSD sono "una licenza davvero buona per l'uso da parte del vostro concorrente"](#). La puntualizzazione di Moglen è che lui crede che un'impresa debba utilizzare una licenza che richieda ad altri di restituire le loro modifiche (come la GPL) se non vuole fornire un pranzo gratis ai suoi concorrenti. Molte persone [sostengono la GPL](#), anche se l'opinione che la GPL sia una licenza buona o necessaria, *non* è una posizione universalmente accettata dagli sviluppatori FLOSS (ad esempio, vedete [Montague](#)).

Ma anche se non vi piace la GPL, a molti potenziali co-sviluppatori piace... e il vostro progetto ha più probabilità di avere successo se vi adeguate a loro. Di solito anche i sostenitori accaniti della GPL sosterranno programmi FLOSS con altre licenze GPL-compatibili, se si sceglie di non utilizzare la GPL. Per esempio, la FSF sostiene da tempo X Window System (X11), anche se è stato rilasciato sotto licenza MIT / X. Tuttavia, se la vostra licenza non è compatibile con la GPL, gli sviluppatori possono invece creare un concorrente in modo da trarre vantaggio dal codice GPL.

E c'è una *montagna* di codice GPL da sfruttare:

1. Freshmeat.net il [10 novembre 2003](#) ha riportato che il 69,66% dei 32.592 progetti software (pacchetti) di cui tiene traccia sono con licenza GPL (le successive due licenze più popolari sono state la LGPL al 5,29%, e le licenze BSD che, sommate, danno un 4,82%) . Alcuni progetti Freshmeat sono proprietari, ma è chiaro che la GPL è la principale licenza FLOSS in questo insieme. [È anche possibile vedere le ultime statistiche di Freshmeat.net.](#)
2. SourceForge.net il [10 novembre 2003](#) ha riferito che la GPL ha totalizzato il 71% dei 45.736 progetti ospitati con licenze open source approvate dall'OSI (quelle successive più diffuse erano la LGPL, il 10%, e le licenze BSD, 7%). Non tutti i progetti di SourceForge sono attivi o hanno prodotto molto codice, ma, nondimeno, questo offre una ragionevole dimostrazione che la GPL è ampiamente usata. [Potete anche vedere le ultime statistiche SourceForge.net.](#) Un altro modo per ottenere le statistiche di SourceForge - e molte altre - è attraverso [FLOSSmole](#). FLOSSmole è uno strumento per interrogare i dati relativi ai *repository* FLOSS, ed è stato reso semplice per l'analisi dei dati da molte fonti.
3. Nel mio scritto [More than a Gigabuck: Estimating GNU/Linux's Size](#) , ho scoperto che Red Hat Linux, una delle più popolari distribuzioni GNU / Linux, ha avuto fisicamente oltre 30 milioni di linee del codice sorgente della versione 7.1, e che [il 50.36% delle righe di codice è stato autorizzato unicamente sotto licenza GPL](#) (le più comuni erano la licenza MIT, 8,28%, e la LGPL, 7,64%). Se si considerano le linee che sono a doppia licenza (sia sotto licenza GPL che sotto un'altra licenza, consentendo agli utenti e agli sviluppatori di scegliere la licenza d'uso), tutte le linee di codice GPL ammontano complessivamente al 55,3%.
4. [Savannah](#) ospita 1986 dei maggiori progetti-GPL (al 11 Novembre 2003) e incoraggia vivamente le licenze GPL-compatibili.
5. La [free software directory della FSF](#) elenca il software FLOSS. Il 24 settembre 2002, ho scaricato la loro intera directory, ed ho scoperto che aveva 1.550 voci, di cui 1363 (87,9%) hanno utilizzato la licenza GPL, 103 (6,6%) la licenza LGPL, 32 (2,0%) una licenza BSD o simil-BSD, 29 (1,9%) la Licenza Artistica, 5 (0,3%) hanno utilizzato la licenza MIT, oltre ad altre licenze ancora più rare. La FSF preferisce la licenza GPL, quindi le statistiche della directory FSF possono essere falsate della percentuale di software GPL che viene registrata,

ma la directory fornisce ancora un'ulteriore forte riprova che la GPL è una licenza ampiamente utilizzato per il FLOSS.

6. In "Homesteading the Noosphere" (section 2) di Eric S. Raymond è riportato che nel luglio 1997 circa la metà dei pacchetti software con espliciti termini di licenza nel Metalab North Carolina (ex Sunsite) hanno utilizzato la GPL. Al momento Metalab era il più grande archivio del software FLOSS.
7. Lo studio [Who is Doing It \(WIDI\)](#) ha esaminato l'elenco di applicazioni BerliOS SourceWell. Delle 1136 applicazioni rintracciate in quel momento, le prime tre [licenze] sono risultate la GPL al 85,9% (976/1136), la LGPL al 4,6% (52/1136), e quelle in stile BSD al 2,8% (32/1136).
8. Che dire di ciò che è *utilizzato*, piuttosto che di ciò che si è *sviluppato*? Un sondaggio [MITRE del 2003 sull'impiego di FLOSS da parte dell'US Department of Defense \(DoD\)](#) ha rivelato che il 52% delle applicazioni FLOSS riportate sono stati rilasciate sotto la licenza GPL; le successive licenze più diffuse sono state BSD (6%), Apache (5%), varie "Community " (4%), e LGPL (3%). Anche se avete affermato che tutte le licenze non-GPL sono state semplicemente varianti di BSD (che non è vero), la GPL ha una semplice maggioranza tra le licenze del FLOSS in uso.
9. [Un sondaggio tra sviluppatori FLOSS in Asia di Shimizu ed altri nel 2003-2004](#) ha scoperto che in tutta l'Asia, il 64,7% ha preferito le licenze "GPL-compatibili" (presumibilmente intendevano licenza GPL o LGPL) mentre il 15% quelle stile BSD, e in Giappone, il 42% ha prediletto le licenze "GPL-compatibili", mentre 30,2 % quelle stile BSD.

["Producing Open Source Software: How to Run a Successful Free Software Project" di Karl Fogel](#) dice che ["se si desidera che il codice possa essere miscelato liberamente con codice GPL - e c'è un sacco di codice GPL in giro - si dovrebbe scegliere una licenza GPL compatibile."](#)

Voi ignorate questa quantità di codice e supporto a vostro rischio e pericolo per il progetto.

3 Altri progetti dimostrano l'importanza della compatibilità GPL

Volete altre prove che la compatibilità GPL è importante? Diversi grandi progetti di alto profilo FLOSS hanno subito dei cambiamenti, spesso dolorosi per rendersi compatibili con la GPL:

1. Il largamente diffuso linguaggio [Python](#) ha subito importanti cambiamenti della licenza nel 2001, in modo da essere compatibile con la GPL dalla versione 2.1 in avanti.
2. Il popolare editor di testo [vim](#) è divenuto GPL-compatibile dopo una [lunga discussione legale](#). La rivista Linux Journal ha premiato molte volte vim come il più popolare editor, quindi questo non è un programma sconosciuto.
3. Anche [Mozilla](#) ha subito un complesso schema di re-licenziamento, ancora una volta, per risolvere le rilevate incompatibilità tra le sue licenze più anziane (in particolare la MPL) e la licenza GPL.
4. Pure [Zope](#) è passato dalla sua "Zope Public License version 1" (che era incompatibile GPL) ad una licenza GPL-compatibile.
5. L'Università della California a Berkeley ha rilasciato una grande quantità di codice (la Berkeley Software Distribution, BSD) che è diventata la base dei vari sistemi * BSD. La licenza BSD originale comprendeva un imbarazzante requisito pubblicitario che era incompatibile con la GPL. Dopo molte richieste (comprese le note che la licenza era

- incompatibile con la GPL), Berkeley ha deciso di cambiare la licenza nel 1999, rendendo questo codice GPL-compatibile (la [FSF ha una pagina che tratta della clausola pubblicitaria BSD originale](#) dal proprio punto di vista).
6. Apache ha cambiato la propria licenza con la "versione 2.0", e uno dei suoi obiettivi era quello di essere "compatibile con altre licenze open source, come la GPL". Purtroppo, la [Apache Software Foundation crede che la Apache License versione 2.0 sia compatibile con la GPL versione 2](#), mentre il [team legale della Free Software Foundation afferma che Apache License versione 2.0 non è compatibile con la GPL](#). Da quel momento, la FSF e Apache Software Foundation hanno lavorato insieme, e la GPL versione 3 è nota per essere compatibile con la licenza Apache versione 2.0 ([il co-fondatore di Apache, Brian Behlendorf, è compiaciuto della compatibilità tra licenze Apache 2.0 / GPLv3](#)). Questa lunga storia rende chiaro che entrambe le parti ritengono che la compatibilità GPL sia un vantaggio importante per una licenza FLOSS.
 7. Il complesso di strumenti [Qt](#) (alla base di KDE) aveva originariamente una licenza incompatibile con la GPL. I suoi proprietari ora usano una doppia licenza Qt con GPL in modo che Qt è compatibile con la GPL.
 8. Come osservato nella [storia di Wine](#), la "storia del [progetto Wine](#) circa le licenze ha suscitato numerosi dibattiti." Il progetto WINE originariamente aveva la BSD vecchia licenza, incompatibile con la licenza GPL; questa incompatibilità con la GPL ha spinto gli sviluppatori a passare alla licenza X11 compatibile GPL nel gennaio 2000. Molti sviluppatori hanno espresso preoccupazione per l'appropriazione del codice da parte di soggetti commerciali, così nel marzo 2002 gli sviluppatori si sono accordati per passare Wine alla licenza LGPL. Il progetto "Rewind" è stato creato per coloro che volevano un codice di base con licenza X11, ma la maggior parte gli sviluppatori hanno deciso di concentrare i loro sforzi sulla sincronizzazione con Wine sotto LGPL, e la stragrande maggioranza di sviluppi e nuove funzionalità per primi compaiono là. Il progetto Wine riporta che, poco dopo aver cambiato la licenza LGPL, lo sviluppo ha cominciato ad assumere un ritmo maggiore (sono iniziate ad apparire più patch, il leader Alexandre ha fornito più pubblicazioni CVS, e sono state segnalate più applicazioni funzionanti).
 9. [Alfresco ha modificato la licenza del proprio programma professionale di gestione dei contenuti](#) da MPL a GPL (entrambe licenze FLOSS, ma la MPL è incompatibile con la GPL). [Un articolo di Stephen Shankland su Alfresco](#) riferisce che Matt Asay (vice presidente del commerciale di Alfresco) va dicendo che il passaggio a GPL elimina alcune barriere e permette ad Alfresco di essere in grado di integrarsi facilmente con altri progetti GPL. Si cita pure Raven Zachary, analista di 451 Group, che dice che, "l'uso di Alfresco della licenza GPL per la sua Community Edition permette contributi comunitari potenzialmente maggiori a causa della familiarità di licenza e dei comprovati standard".
 10. [Tre organizzazioni di ricerca pubblica francese hanno lanciato un progetto che ha redatto nuove licenze Free Software, conformi al diritto francese](#): CeCILL, CeCILL-B, e C-CeCILL. Esse hanno ritenuto importante la compatibilità GPL, anche quando si creano nuove licenze. La licenza CeCILL è molto simile alla GPL ed è esplicitamente compatibile con la GPL. Le [CeCILL Frequently Asked Questions \(FAQ\)](#) (dai loro autori) dicono: "CeCILL è tuttavia compatibile con la GNU GPL: quando un software sotto CeCILL è integrato con un software sotto licenza GPL, l'opera risultante può essere distribuito sotto la licenza GNU GPL". Si noti che la [FSF concorda che la CeCILL sia compatibile con la GPL](#).
 11. Nel giugno del 2010 Google ha pubblicato la sua implementazione WebM del codec multimediale VP8, in parte perché la vecchia patente era GPLv3 e incompatibile GPLv2. In June 2010 [Google ha modificato la licenza della sua implementazione WebM del codec](#)

[multimediale VP8](#), in parte perché la vecchia licenza era incompatibile con GPLv3 e GPLv2.

Molteplici importanti progetti non si sottopongono a dolorosi cambiamenti nella licenza a meno che non abbiano un motivo per farlo. La GPL è così popolare che la compatibilità GPL è ormai un requisito importante in una licenza. Eirik Eng, presidente di Trolltech (proprietaria di Qt), ha spiegato le azioni di Trolltech dicendo: "ultimamente la GPL è stata utilizzata sempre più per i nuovi sorgenti [software] aperti". L'articolo ["Replacing init with Upstart"](#) nota che quando gli autori stavano iniziando il loro progetto "Upstart", hanno osservato i sistemi esistenti come punto di partenza, e Sun SMF e il programma di Apple 'launchd' "sono stati immediatamente esclusi a causa dei problemi di licenza. E' stato importante per noi che la soluzione sia indiscutibilmente libera in modo che altre distribuzioni possano adottarla: molti avevano già respinto quelle per ragioni di incompatibilità GPL".

Il sistema operativo Sun Solaris CDDL, una licenza incompatibile con la GPL FLOSS. Sun ha tutto il diritto di rilasciare il proprio software sotto qualunque licenza scelga (nel rispetto della legge), naturalmente. Tuttavia, [nell'articolo di Groklaw del 2005 "Sun Responds to Criticism of CDDL"](#) si afferma che, poiché la CDDL non è compatibile con la GPL, Sun non otterrà il sostegno della comunità che spera di raccogliere con l'uso della CDDL. L'articolo prevede che "Il risultato sarà, però, che Linux continuerà a svilupparsi più rapidamente e seppellirà la licenza di Sun e il suo codice ... Sun è libera di tagliare fuori se stessa da ciò, se lo desidera, ma raccoglierà ciò che semina. Se ha immaginato che il mondo avrebbe abbandonato la GPL e adottato al suo posto la CDDL, credo che ormai si rendano conto che non succederà ... Dubito seriamente che qualsiasi licenza che sia incompatibile GPL / LGPL verrà ampiamente adottata". Ciò è stata una previsione estremamente accurata. Nel 2008, [Theodore Ts'o ha rilevato che "Open Solaris \[non è riuscita a sviluppare\] una comunità Open Source Development ... dal punto di vista del business, mi meraviglierei se Sun fosse davvero in grado di sostenere la propria squadra di ingegneria Solaris se vorrà realmente fare da sé tutto il lavoro ... Con Linux, abbiamo un grande vantaggio in quanto i miglioramenti del kernel provengono da diverse aziende dell'ecosistema, invece di essere pagati da una sola azienda ... non è chiaro come Sun giustifichi i costi di progettazione Solaris ai propri azionisti."](#) Infatti, [Emily Ratliff ha fatto un po' di calcoli ed ha scoperto che "Linus \[Torvalds\] ottiene più patch \[per Linux\] mentre sta lavandosi i denti di quante ne arrivano ad OpenSolaris in una settimana". Il fallimento di Solaris nel conquistare una più ampia comunità di sviluppo è divenuto uno dei maggiori argomenti discussione su Slashdot nell'aprile 2008 quando si è dimesso Roy Fielding.](#) Questa incapacità di ottenere una comunità è sicuramente causato da altro più che da problemi di licenza, ma le incompatibilità di licenze sembrano essere parte della storia, come [Michael Dolan annota nel 2008](#).

Nel novembre 2006 Sun ha annunciato che avrebbe rilasciato la sua implementazione di Java. La licenza? La GPL, insieme ad alcune eccezioni simil-LGPL per la propria libreria - e *non* la CDDL, che aveva usato in precedenza con il rilascio di Solaris. Quando serviva ottenere la fiducia di una comunità, è stata scelta una licenza GPL-compatibile.

Ecco qui una prova ulteriore: alcuni progetti che hanno cercato di passare da una licenza compatibile con una licenza incompatibile con la GPL si sono scissi, con i loro capi originali sostanzialmente rimossi dal loro incarico. Una "scissione" ["fork"] è un tentativo di creare un progetto concorrente copiando da un progetto esistente e convincendo sviluppatori e utenti a passare invece al progetto concorrente. Le scissioni nel mondo FLOSS sono essenzialmente come un "voto di sfiducia" in un parlamento. La *possibilità* di provocare una scissione è un'importante garanzia nel FLOSS, perché la facoltà di scissione impedisce ai capi di diventare tirannici. Ma le scissioni raramente riescono: la situazione deve essere piuttosto grave per determinare che la stragrande

maggioranza degli sviluppatori e distributori passi a qualcos'altro (*non* sto parlando di rami di controllo delle versioni, che sono a volte chiamati *fork* soprattutto con git. Ci deve essere un intento di indurre la gente a smettere di usare il precedente progetto, e a passare invece al nuovo progetto. [Rick Moen ha un articolo sulle scissioni, se volete maggiori dettagli](#)). Eppure ci sono state scissioni con successo causate dal fatto che la guida del progetto ha cercato di imporre agli altri una licenza incompatibile con la GPL. Questa è una prova convincente che la compatibilità GPL è di fondamentale importanza per gli altri sviluppatori e utenti.

XFree86 è forse l'esempio più noto di scissione causata da un tentativo di cambiare le licenze. Il progetto XFree86 storicamente ha portato allo sviluppo di un popolare server X, e tradizionalmente la maggior parte del suo codice ha usato la semplice licenza open source "MIT / X", che è compatibile con la GPL. Il presidente XFree86, David Dawes, ha deciso di cambiare la licenza di XFree86 nel 2004 per una che non era compatibile con la GPL, soprattutto per fornire agli sviluppatori più credito. Il problema non consisteva nel dare credito: il problema era l'incompatibilità GPL. Questo tentativo di modifica ha condotto ad un esodo di massa da XFree86, non appena quasi tutti gli sviluppatori e gli utenti hanno immediatamente abbandonato XFree86. Guardate la [mia appendice su XFree86](#) per informazioni più dettagliate su questo.

Purtroppo, alcune persone non imparano. Essenzialmente la stessa cosa è accaduta nel 2006, quando l'autore di "cdrtools" ha cercato di passare ad una licenza incompatibile con la GPL: tutti i principali distributori hanno invece dato inizio e sono passati ad un nuovo progetto ([cdrkit](#)). L'originale suite cdrtools è un complesso di strumenti per la scrittura di informazioni su CD e DVD. In questo caso, [Jörg Schilling \(autore di cdrtools\) aveva cambiato la sua licenza dalla GPL alla CDDL incompatibile con la GPL](#). Jörg Schilling affermava che non c'era alcun problema, ma questo è semplicemente una sciocchezza. Jörg non è un avvocato, mentre tutti gli avvocati che hanno esaminato la questione, così come gli sviluppatori di entrambe le CDDL e le licenze GPL, hanno convenuto che le licenze sono incompatibili. Debian ha immediatamente creato una scissione del progetto denominata cdrkit. Fedora, Gentoo Linux, Mandriva Linux, OpenSUSE e Ubuntu sono poi passati tutti a cdrkit. [L'annuncio di Debian del passaggio](#) ha chiarito che l'*unico* motivo di questa (riuscita) scissione è stato l'incompatibilità con la GPL: "Allora, perché la scissione? La masterizzazione di CD / DVD è un affare complicato che ha bisogno di un sacco di conoscenze, sicché scindere una così grossa raccolta [di software] non è un passo da prendere alla leggera. Richiede un grande sforzo di sviluppo che potrebbe essere impiegato meglio altrove. In passato, noi, i manutentori Debian di cdrtools, abbiamo avuto un rapporto di collaborazione buono e reciproco con Jörg Schilling ... [Ma poi] Jörg Schilling ha rilasciato parti di versioni recenti di cdrtools sotto la licenza [CDDL]. La CDDL è incompatibile con la GPL. La stessa FSF dice che questo è il caso delle persone che hanno contribuito alla redazione della CDDL ... [Dato che l'autore] ha sostanzialmente ignorato ciò che abbiamo detto, non abbiamo altra scelta, eccetto la scissione...". A questo punto, quasi a nessuno importa ciò che fa l'autore originale con la sua versione: la versione che la gente effettivamente *utilizza* è controllata da altri.

[Volker Lendecke](#) di Samba ha dichiarato che tutti i rimedi della Commissione Europea che coinvolgono il rilascio di dati di Microsoft di dati dovrebbero richiedere l'uso di una licenza GPL-compatibile. Quindi, anche i rilasci di dati - non solo il codice del software - sono in corso di esame per vedere se sono compatibili con la GPL.

Più recentemente, nel 2011 il [Dipartimento di Giustizia americano e l'Ufficio federale tedesco dei cartelli hanno costretto CPTN \(una holding di proprietà di Microsoft Inc., Oracle Corp., Apple Inc. e EMC Corp.\) a cambiare le modalità di acquisito dei brevetti software di Novell in modo che la comunità open source sia protetta](#) e, in particolare, il Wall Street Journal ha riferito che "tutti i

brevetti acquisiti saranno soggetti alla GNU General Public License (GPL)". Questo fa apparire che un requisito sarà che i brevetti devono essere concessi per l'uso con qualsiasi programma rilasciato sotto GPL ... ma forse non per i programmi incompatibili con la GPL.

4 Alcune licenze incompatibili GPL da evitare

Esistono molte licenze incompatibili con la GPL sotto cui dovrete evitare, possibilmente, di rilasciare il software. Abbiamo già ricordato la CDDL. Qui ce ne sono alcune altre:

1. Evitate la licenza originale BSD ("4-clause"). La licenza originale comprendeva una clausola ora chiamata la "odiosa clausola pubblicitaria BSD". Questa stabiliva che:

Tutto il materiale pubblicitario riguardante le funzioni o l'utilizzo di questo software deve riportare la seguente dichiarazione:
This product includes software developed by the University of California, Berkeley and its contributors.

[Questo prodotto include software sviluppato dall'Università della California, Berkeley e dai suoi collaboratori.]

Ciò l'ha resa incompatibile con la GPL, così come ha introdotto i [problemi pratici notati dalla FSF](#). Il problema è che quando si meschia il codice (e si meschia) e ognuno aggiunge il proprio nome (l'hanno fatto), la gente non poteva fare pubblicità senza elencare potenzialmente centinaia di persone - rendendo questa clausola del tutto impraticabile come software in scala. Nel giugno 1999, dopo due anni di discussioni, l'Università della California ha rimosso questa clausola dalla licenza di BSD. Così, questa è una licenza obsoleta che non deve essere più utilizzata: piuttosto usate licenze come la "nuova BSD", detta anche "BSD modificata" o "3-clause BSD".

2. Evitate la "Academic Free License" (AFL). Si è sostenuto che l'AFL è stata un "aggiornamento compatibile" da "licenze come la BSD e MIT", ma questa è una affermazione fuorviante: come osserva il FSF, "la licenza BSD modificata e la licenza X11 (nota anche come MIT) sono GPL-compatibili, ma la AFL non lo è" (almeno la versione 1.1 e 2.1 sono incompatibili, e credo che pure la versione AFL 3 non sia compatibile). L'AFL ha alcune belle caratteristiche da un punto di vista giuridico, ma la sua incompatibilità con la GPL è una grave mancanza che sminuisce completamente i vantaggi. Progetti come [SHPTRANS](#) hanno evidenziato l'incompatibilità AFL come un problema grave, e hanno scelto di non utilizzare la AFL per questo motivo. In un certo periodo Per un certo periodo alcune persone nell'OSI hanno raccomandato l'AFL, ma a partire dal 30 Luglio 2007 l'OSI elenca la licenza AFL come "ridondante rispetto a licenze più popolari", e puntualmente non la comprende tra le "licenze che sono molto popolari e ampiamente utilizzate o con le forti comunità".
3. Per ora, evitate la "Common Public License" 1.0 (CPL), anche se la situazione potrebbe cambiare. La CPL non è così popolare come le licenze GPL, LGPL, nuove BSD, o MIT/X11, ma è più usata di altre licenze. La CPL è ancora nella lista OSI delle "licenze che sono popolari e ampiamente utilizzate o con forti comunità". Alcune altre licenze, come ad esempio la licenza Eclipse, si basano sul CPL. E' noto che la CPL è incompatibile con la GPLv2, quindi per il momento è probabilmente meglio evitarla. Tuttavia, è possibile che la CPL sia compatibile con licenza GPL versione 3, che la renderebbe molto più gradita e simile alla licenza Apache 2.0 (che è incompatibile con la GPLv2, ma compatibile con la GPLv3). Al momento, mi piacerebbe aspettare finché ci saranno state lunghe analisi

- giuridiche per dirimere la questione.
4. Evitate la "Licenza Software Open" (OSL). Questa dovrebbe essere una licenza fortemente protettivo come la GPL. Ma è incompatibile con la GPL, e quasi nessuno usa questa licenza, sicché un tale codice non può usare o essere utilizzato dalla stragrande maggioranza dei progetti FLOSS. [La FSF ritiene inoltre che i termini di distribuzione rendano illegali i normali processi di sviluppo, creando un grave problema pratico](#). Nuovamente, per un certo periodo alcune persone nell'OSI hanno raccomandato l'OSL, ma a partire dal 30 luglio 2007 la OSI elenca la licenza OSL come rientrante nella categoria "Altre / Varie licenze", e puntualmente non la include tra le "licenze che sono popolari e ampiamente utilizzate o con forti comunità".
 5. Evitate la licenza "Artistic 1.0" (di Perl). Anche se destinato ad essere una licenza OSS, la licenza Artistic 1.0 è una licenza confezionata malamente, che crea ogni sorta di problemi legali. La Free Software Foundation (FSF) dice che non è una licenza libera, perché il suo testo è vago e aperto ad interpretazioni errate. [La comunità Perl, che l'ha sviluppata, ha condiviso questa valutazione](#) ed ha riscritto la licenza Artistic per risolvere tutti i problemi individuati. [L'OSI ha "rimpiazzato" la licenza, raccomandando vivamente a tutti gli utenti di passare alla Artistic 2.0](#). Fedora (una distribuzione Linux molto popolare) non permette più progetti che utilizzino la Artistic 1.0, e [per Fedora 10 ha pianificato di rimuovere tutti i progetti che impieghino solo la licenza Artistic 1.0](#). In realtà, questa licenza è *probabilmente* compatibile con la GPL, ma è così difficile capire che cosa voglia dire che io la includo qui. Raccomando l'utilizzo delle licenze MIT e nuove BSD invece delle licenze Artistic dove è ragionevolmente possibile.
 6. Evitate l'uso delle licenze "Creative Commons" per il software. La [FAQ di Creative Commons FAQ dice: "Le licenze Creative Commons non sono destinate ad essere applicate al software. Non dovrebbero essere utilizzate per il software ... \[Non distinguono, quando necessario,\] tra oggetto e codice sorgente ... Incoraggiamo vivamente ad utilizzare una delle molto valide licenze software oggi disponibili \[invece\]"](#). ["I 5 motivi per non scegliere una licenza CC per il codice"](#) di Jay Tuley , spiega ancor di più. Il [debian-legal Summary of Creative Commons 2.0 Licenses](#) raccomanda che gli autori desiderosi di creare opere compatibili con "Debian Free Software Guidelines" di Debian non dovrebbero utilizzare nessuna delle licenze della suite delle licenze Creative Commons: licenze con elementi come il "Non commerciale" o "Non opere derivate "sono fundamentalmente incompatibili con il FLOSS. Gli autori che utilizzino o intendano utilizzare la licenza Attribution 2.0 dovrebbe prendere in considerazione una simile licenza di software libero come ad esempio una licenza BSD o stile MIT ... e gli autori che usino o prevedano di usare la Attribution-ShareAlike 2.0 dovrebbero prendere in considerazione una licenza simile per il software libero come la GNU General Public License [GPL]. Creative Commons ha "conglobato" la [GPL](#) e la [LGPL](#) se desiderate utilizzare il motore di ricerca Creative Commons.
 7. Evitate la "Mozilla Public License" (MPL). Questa è stata originariamente creata da Mozilla, ma la sua incompatibilità con GPL ha causato così tanti problemi che il prodotto è stato rilicenziato sotto una tripla licenza GPL/LGPL/MPL. Pure altri ex utenti della MPL, come [Alfresco](#), l'hanno pure abbandonata. Anche il creatore originario della MPL l'ha abbandonata come propria licenza esclusiva a causa della incompatibilità con la GPL: non ripetete i loro errori. [Google non elenca più la MPL come una licenza raccomandata, anche se una volta lo faceva](#). In effetti, c'è stato un momento in cui il codice di Google non avrebbe accettato progetti MPL: [ora il codice di Google permette qualsiasi licenza OSS, ma ancora non comprende la MPL tra le licenze raccomandate](#). Il [processo di revisione della MPL](#) sta tentando di rendere la versione 2 della MPL compatibile con la licenza Apache 2.0, ma non

- si sa se la MPL2 sarà GPL-compatibile.
8. Evitate la CDDL. Questa creazione di Sun è simile alla MPL. Neppure l'originario creatore della MPL la usa più (in esclusiva) e l'esperienza di OpenSolaris e cdrtools dimostra che la licenza è un vero ostacolo. I progetti basati sulla CDDL tendono ad avere relativamente scarsi livelli di partecipazione e ritengo che una ragione chiave sia la licenza.
 9. Evitate la "NASA Open Source Agreement" version 1.3. Questo è uno di quei rari casi in cui l'OSI ha accettato la licenza come licenza per il software open source (per la definizione di open source), ma la FSF ha stabilito che la licenza *non* è una licenza per software libero (per la definizione di Software Libero). Il problema è che richiede che le modifiche siano vostre "creazioni originali". Lo sviluppo di software FLOSS dipende dalla combinazione di codice di terze parti, cosicché se venisse interpretato letteralmente, questo accordo vieterebbe praticamente tutto il reale lavoro e sarebbe certamente incompatibile con la GPL. E' meglio evitare questa licenza. La NASA mi ha detto che stanno per modificare la licenza per rimuovere questa bizzarra limitazione: ciò la renderebbe migliore, anche se non so se il risultato sarà compatibile con la GPL.

5 Valutate attentamente l'incompatibilità con la GPL

Ci sono delle ragioni per cui un progetto FLOSS sia incompatibile con la GPL, ma soppesate le idee di questo tipo con un occhio estremamente critico. A volte un programma o una libreria era originariamente di proprietà, e può essere rilasciato solo come FLOSS sotto condizioni incompatibili con la GPL. In tal caso, è meglio che il codice sia buono e praticamente completo al momento del rilascio, in quanto molti sviluppatori non vorranno averci a che fare. Molti progetti FLOSS sono già gravati da una licenza incompatibile con la GPL, ed è troppo difficile cambiarla. Alcuni includono altri requisiti che ritengono fondamentali, come l'aggiunta di più clausole brevettuali per le licenze, la difesa, o la risoluzione reciproca (come la licenza pubblica di IBM).

Ma ricordate che questi presunti benefici potrebbero non superare i molti svantaggi. Notate tutti i progetti sopra elencati, che sono passati da una licenza incompatibile con la GPL ad una compatibile. Un esempio dei problemi - in realtà, uno di quelli più importanti - è la [licenza OpenSSL che è ampiamente considerata come incompatibile con la GPL](#) (anche dai legali della FSF, da GNOME e Debian, sulla base di dettagliate analisi indipendenti). Mentre gli [sviluppatori della OpenSSL proclamano che è compatibile con la GPL](#) e "sostengono" l'uso di OpenSSL da parte di software GPL, i campi minati legali sono tali che la OpenSSL è stata spesso aggirata o abbandonata [passando ad un altro programma \(come ha fatto CUPS\)](#). Le abituali alternative ad OpenSSL sono la [Mozilla Network Security Services \(NSS\)](#) e la [GNU Transport Layer Security \(GNU TLS\)](#).

6 La proliferazione delle licenze considerata nociva

Naturalmente [la proliferazione delle licenze FLOSS è stata in generale identificata da molti come un problema da risolvere](#). E' stata una questione osservata [nell'indirizzo del LinuxWorld 2005 di Martin Fink, Vice Presidente di HP](#), ed altri hanno supportato questo punto di vista come [Groklaw](#). I clienti trovano una lunga lista di licenze del tutto sconcertante: è difficile capire cosa può essere messo

assieme legalmente e le aziende possono facilmente ritrovarsi a spendere enormi somme di denaro per consulenze legali, se devono cominciare a esaminare di più delle poche principali licenze. Come osservato in ["The Open Source Licensing Implosion" di Serdar Yegulalp](#) (*InformationWeek*, Aug 8, 2008), "Non è solo che ci sono "troppe licenze open source", ma che le conseguenze per la creazione spensierata di altre di nuove si sono alla fine concretizzate... Era più semplice farla franca con un'ampia proliferazione di licenze quando l'open source era ancora una varietà di uccello relativamente rara ed esotica nel bestiario del software. Ora che l'open source sta divenendo (gasp) un fenomeno pubblico, usare una delle licenze meno comuni o proporre una personale lavora più spesso contro di voi piuttosto che il contrario... Non saranno i programmatori che determineranno quali siano le migliori licenze open source – saranno i consumatori del software. Essi saranno quelli che ridurranno la foresta di licenze a pochi alberi ben potati e curati. É meglio per noi tutti non perdersi tra di loro”.

Questa non è una nuova osservazione: [Bruce Perens l'aveva già notato nel 1999](#), "Non scrivete una nuova licenza se è possibile usarne una di quelle qui elencate. La propagazione di licenze molto differenti e incompatibile lavora a detrimento del software Open Source perché i frammenti di un programma non possono essere utilizzati in un altro programma con una licenza incompatibile". Il [Software Release Practice HOWTO di Eric S Raymond](#) afferma con forza (come un titolo!) "non scrivere una propria licenza se è possibile evitarlo", e suggerisce che gli sviluppatori utilizzino "una delle licenze standard riportate nel sito OSI, se di fatto possibile".

Se dubitate che la proliferazione delle licenze possa costituire un problema, basta che guardiate [Licensing Guideline di Fedora](#) e [la pagina Licensing di Fedora](#) – costituiscono una complessa documentazione di licenze che può essere registrata e analizzata elettronicamente – giusto per assicurarvi della conformità delle licenze. Inoltre considerano specificamente se una licenza sia compatibile GPLv2 e GPLv3, dimostrando chiaramente l'importanza della compatibilità GPL.

La denuncia di Fink ha provocato un certo miglioramento, ad esempio [Intel ha pubblicamente abbandonato la propria licenza FLOSS](#) in modo da poter usare invece licenze standard. L'[Open Source Initiative's License Proliferation Committee Report](#), un gruppo apparso dopo i commenti di Fink, ha raccomandato di indirizzare ciò attraverso la classificazione delle licenze. Senza sorpresa la categoria "Licenze che sono popolari e ampiamente utilizzate o con forti comunità" ha ricompreso le licenze MIT/X, nuove BSD, LGPL, e GPL. Il suo elenco completo, al 19-09-2006, era Apache License 2.0, licenze BSD nuove e semplificate, GPL, LGPL, MIT, MPL, CDDL, CPL 1.0, e Eclipse Public License (EPL). Personalmente questo elenco è sovradimensionato: le MPL, CDDL, CPL e EPL hanno un numero infinitamente piccolo di progetti che le utilizzano.

Allo stesso modo, il [John Cowan's FLOSS license wizard](#) (ad agosto 2008) raccomanda solo poche licenze FLOSS, a seconda delle vostre esigenze. Le uniche licenze FLOSS che vuol raccomandare sono le licenze GPLv3, LGPLv3, BSD revisionate e Apache 2.0.

Il 14 agosto 2008 ho eseguito un rapido controllo del codice di Google ed ho scoperto che essi consentono solo le seguenti licenze: Apache 2.0, Artistic/GPL, GPLv2, GPLv3, LGPL, MIT e nuove BSD. Notate che tutte sono compatibili GPL (Apache 2.0 è compatibile soltanto con GPLv3 e non con GPLv2). Come sopra osservato, una volta essi accettavano la MPL incompatibile GPL, ma ora non più.

["The Open Source Licensing Implosion" su InformationWeek](#) di agosto 2008 nota correttamente che "le conseguenze della creazione spensierata di nuove [licenze FLOSS] stanno finalmente diventando concrete [per molti] ... la stragrande maggioranza dei prodotti open source là fuori usa una piccola manciata di licenze ... Il resto tende ad essere cose a se stanti o derivati... Era più facile

farla franca con una abbondante proliferazione di licenze nel passato quando l'open source era ancora una varietà relativamente rara ed esotica... Ora che l'open source sta divenendo (gasp) di uso corrente, l'impiego di una delle licenze meno comuni o il saltar fuori [con una licenza personale] è per voi controproducente più spesso di quanto non lo sia... [le comunità saranno interessate] se voi state utilizzando una licenza a cui non è stata data alcuna sorta di radicale pubblico cambiamento... i consumatori di software [ridurranno] la foresta di licenze a pochi alberi ben potati e mantenuti. Il meglio per noi tutti per non perdersi tra di loro”.

7 Altre cose da considerare

Sebbene sia possibile relicenziare il software FLOSS per renderlo compatibile GPL, ciò potrebbe essere abbastanza difficile – sarebbe molto meglio evitare un errore ed partire già compatibili GPL sin dall'inizio. La modifica delle licenze solitamente richiede l'approvazione del relicenziamento da parte di ogni detentore di diritti e una riscrittura del codice nel caso in cui non venga concessa l'approvazione. Quando i progetti FLOSS crescono di dimensioni, ciò è incredibilmente complicato, perché dopo molti anni spesso è impossibile contattare tutti i contributori. Per fortuna, molti progetti non devono subire questa trasformazione, perché l'hanno evitata sin dall'inizio. Il W3C, per esempio, [afferma esplicitamente che tutte le sue versioni del software sono FLOSS e sono compatibili con la GPL](#). Questa dichiarazione mostra che crede che la compatibilità GPL sia molto importante e questa politica elimina i problemi che sarebbero derivati da incompatibilità con la GPL.

Vale la pena notare che la GPL ha una base legale molto robusta, che è particolarmente attraente per molti sviluppatori. La GPL è facile da rispettare (non ha nemmeno bisogno di soldi per farlo), ma le persone devono prendere sul serio i suoi requisiti. [Eben Moglen è un avvocato che si è occupato della GPL per anni](#) ed ha una prova eccellente che la GPL è realmente applicabile. Ha anche spiegato la base giuridica della GPL, che risulta essere molto forte. La paralegale [Pamela Jones ha scritto un'altra spiegazione delle basi legali della GPL per i non addetti](#): come notano entrambi, la GPL è una licenza, non un contratto, che la rende particolarmente facile da applicare. [“Taking the Case: Is the GPL Enforceable?” di Jason B. Wacha \(General Counsel, Monta Vista Software\)](#) è una dettagliata analisi giuridica di un avvocato che dimostra che “la GPL è un accordo applicabile”. Un [tribunale tedesco ha trovato che ha valore legale](#), per le molteplici ragioni affermate da Moglen e Jones. Questo è stato in risposta ad [alcune iniziative legali da parte di netfilter](#). [“Enforcement of the GNU GPL in Germany and Europe” di Till Jaeger](#) mostra che la GPL lì è abbastanza applicabile (al 2010). Nel 2006, un tribunale statunitense ha stabilito che la GPL era stata sufficientemente legittimata: Daniel Wallace ha tentato di invalidare la GPL in [Daniel Wallace vs. Free Software Foundation](#), e non solo l'azione di Wallace è stata rigettata, ma Wallace è stato pure condannato a pagare le spese legali. In questo caso, il giudice (John Daniel Tinder, United States District Court) ha anche dichiarato che “La GPL incoraggia, anziché scoraggiarla, la libera concorrenza e la distribuzione di sistemi operativi per computer, i cui benefici si riverberano direttamente sui consumatori. Tali benefici comprendono prezzi più bassi, miglior accesso e maggiore innovazione”. Ciò è stato appellato e la [U.S. Court of Appeals \(7th circuit\) ha formalmente stabilito nel 2006 che la GPL non viola le regole antitrust statunitensi e che "la licenza GPL e il software open-source non hanno nulla da temere dalle leggi antitrust"](#) – creando una difesa ancora più forte contro casi futuri. La dimostrata forza legale della GPL è interessante per molti soggetti che devono scegliere una licenza per il loro software open source. Ma questa forza deriva dal prendere seriamente i suoi requisiti - il che significa pure che molta gente deve valutare l'incompatibilità con serietà.

Più in generale, c'è una prova crescente che le licenze FLOSS sono realmente applicabili. [“Ruling](#)

[Bolsters Open-Source Software: A Licensing Agreement Is Declared Enforceable Under Copyright Law](#)” di Robert A. Guth (*Wall Street Journal*, 14 Agosto 2008, Pagina B6) ha riportato che la Court of Appeals for the Federal Court ha stabilito nell'agosto 2008 che un'altra licenza (la Artistic Licence) era applicabile ai sensi del diritto d'autore. Ciò ha dato sostegno ad “un principio alla base del software open source e di altre creazioni che al grande pubblico è permesso di modificare e distribuire liberamente ... Alcuni esperti legali hanno detto che la sentenza potrebbe contribuire a sostenere le basi giuridiche di altri progetti open source. 'E' una decisione chiara che se qualcuno non segue le condizioni che inserite in una licenza open-source, voi avete a disposizione un'azione a tutela del diritto d'autore' ha dichiarato Wendy Seltzer, una collega presso il Berkman Center for Internet & Society alla Harvard Law School". [Ecco la sentenza: questo è il sunto di Bruce Perens](#).

Ma l'ampio utilizzo della GPL semplifica pure la ricerca di aiuto per conformarsi ad essa. "[A Practical Guide to GPL Compliance](#)" del [Software Freedom Law Center](#) mostra i semplici passi che quasi tutte le organizzazioni possono applicare con facilità in modo da poter conservare la conformità. Le regole sono in realtà piuttosto semplici.

8 Un problema collegato: l'utilizzo della licenza standard (CC-BY-SA) per altre opere

Come problema connesso, le opere sottoposte a diritto d'autore diverse dal software (come la documentazione) dovrebbero essere rilasciate sotto una licenza standard se intendete che siano liberamente riutilizzabili. Anni fa non era chiaro di quale licenza si trattasse: [l'applicazione di una licenza a contenuti aperti era un disastro](#).

Tuttavia, a partire dal 2009, sembra che la emergente licenza di comune "consenso" per le opere soggette a diritto d'autore diverse dal software sia la [licenza Creative Commons \(CC\) Attribution-Share Alike \(CC-BY-SA\)](#). Pertanto, accertatevi che la [licenza CC-BY-SA \(unported version 3.0\)](#), oppure una licenza compatibile con essa, sia una delle licenze che utilizzate per rilasciare del materiale liberamente riutilizzabile.

Storicamente, [Wikipedia](#) e i relativi progetti sono stati rilasciati sotto la GNU Free Documentation License (GFDL), ma a far data dal maggio 2009 [Wikipedia è passata all'uso della licenza CC-BY-SA come sua licenza principale](#). La maggioranza del materiale di Wikipedia ora è soggetto a doppia licenza (è disponibile sotto entrambe le licenze GFDL e CC-BY-SA), ma la CC-BY-SA è la sola licenza che sia applica a *tutto* il suo materiale. L'unica logica per effettuare questa variazione è stata che: "Senza il cambiamento, non avremmo potuto condividere il testo con i progetti che utilizzano le licenze Creative Commons Attribution / Share-Alike. Le licenze Creative Commons vengono utilizzate da centinaia di migliaia di autori in tutto il mondo (v. statistiche), divenendo rapidamente lo strumento legale maggiormente usato per rilasciare diritti su opere diverse dal software. Questa barriera di interoperabilità con altre organizzazioni non a scopo di lucro e comunità online, che condividono liberamente la conoscenza, è quindi in contrasto con la missione di Wikimedia". E' interessante notare che Richard Stallman e la FSF, l'autrice della GFDL, hanno lavorato proprio per rendere possibile questo cambiamento.

Nel 2009, [Fedora ha cambiato in massa le licenze \(compreso wiki di Fedora e guide\) portandole a CC-BY-SA](#) spiegando: "Perché? Perché è vitale che Fedora scenda da un'isola di contenuti e si unisca al resto del mondo ... Per essere onesti, questo cambiamento è probabilmente un po' in ritardo". Alcuni dei motivi adottati sono stati:

- Stiamo cambiando le licenze dall'OPL (Open Publications License) 1.0. [L'autore ed il solo unico curatore di sempre della OPL ha scritto due anni fa perché la gente avrebbe dovuto smettere di usare la sua licenza a favore delle licenze CC](#). Si prega di leggere ciò: è ben ragionato e rende gli argomenti convincenti, tanto più perché proviene dall'autore della OPL. Costui l'ha scritto due anni fa, ma cita se stesso dal 2003 quando ha invece raccomandato di utilizzare una licenza Creative Commons. Ha detto abbastanza.
- Siamo su un'isola quasi soli con la OPL ... siamo in grado di passare da un piccolo arcipelago al più grande continente di contenuti aperti [open content] sotto la bandiera della CC BY SA.
- Altre organizzazioni che hanno un contenuto che possiamo riutilizzare in Fedora e a cui contribuire a nostra volta, come Wikipedia e GNOME, sono passate o stanno passando alla CC BY SA ...
- Se non avete mai visto la quantità di contenuti aperti presente, ad esempio, in flickr.com e Wikicommons, per favore date un'occhiata. Per gli autori di contenuti, si tratta di andare praticamente da zero utili oggetti multimediali aperti disponibili a decine di milioni di fotografie, diagrammi, e così via, che possiamo non solo riutilizzare liberamente, ma a cui possiamo contribuire a nostra volta...

Anche in questo caso, non c'è alcun obbligo che CC-BY-SA sia l'*unica* licenza, ma in quasi tutti i casi non è saggio utilizzare una licenza che *non* sia compatibile con la comune licenza standard ... per tutte le stesse ragioni valide per il software. Per esempio, ci sono un sacco di inconvenienti ad impiegare la GFDL, l'OPL, o le varie licenze Creative Commons "non commercial" (-NC) (per maggiori informazioni su quest'ultime, vedete [The Case for Free Use: Reasons Not to Use a Creative Commons -NC License](#)), se il vostro obiettivo è quello di consentire l'uso del lavoro degli altri e il riutilizzo futuro del vostro lavoro. In breve, usando una licenza standard, oppure una licenza compatibile con una, il vostro lavoro può utilizzare ed essere combinato con il lavoro di altri.

E' interessante notare che [un disegno di legge per libri di testo aperti](#) è stato introdotto dal senatore Durbin. Questi suggeriscono che vedremo un contenuto molto più aperto in futuro. Ciò suggerisce che in futuro vedremo molti più contenuti aperti.

Quindi, come potete fare questo? Potete farlo attraverso il rilascio del vostro lavoro sotto diritti CC-BY-SA, come in almeno una delle sue licenze. Potete anche fare ciò rilasciandolo sotto una licenza compatibile, come il CC-BY (Creative Commons-Attribution) e la licenza [licenza CC0](#) (CC0 è essenzialmente un rilascio al pubblico dominio, in un modo che è legalmente efficace in tutte le giurisdizioni). Ora, tornando al software ...

9 Rendere il software FLOSS compatibile con la GPL

Così, come accertarvi che il vostro software sia compatibile GPL? Fortunatamente, ci sono tre modi semplici per fare ciò:

1. Un modo, naturalmente, è quello di usare la [GPL](#).
2. Un altro è quello di utilizzare una licenza diversa che sia nota per essere compatibile GPL, in particolare le licenze [LGPL](#), [MIT/X](#), o [nuova BSD](#) (BSD modificata). Ovviamente ci sono diverse licenze che potete scegliere rimanendo compatibili con GPL. [La FSF mantiene un elenco di licenze che ha stabilito essere3 compatibili GPL](#). La [pagina Fedora Core Licensing](#) ha una tabella veramente utile che mostra le licenze e se sono o non sono compatibili. [Non](#)

[usate la vecchia/originala licenza BSD, che è generalmente ritenuta come incompatibile con la GPL e ha molti problemi pratici](#): anche [Berkeley è passata alla nuova licenza BSD per il suo software](#). Potreste prendere pure in considerazione l'uso dell'Apache 2.0, ma notate che mentre è compatibile con la GPL versione 3, gli sviluppatori della GPL sono sicuri che non sia compatibile con la GPL versione 2.

Se state considerando la nuova licenza BSD, [Bruce Perens](#), [la FSF](#), ed io raccomandiamo invece l'utilizzo della licenza MIT/X (la FSF chiama questo la licenza X11). Esistono svariate ragioni per preferire la licenza MIT/X in luogo della nuova BSD. In primo luogo, MIT/X è molto più semplice, una cosa buona per una licenza. In secondo luogo, alcuni non legali hanno sostenuto che la nuova BSD non è compatibile con la GPL: non credo che questo sia un vero problema, ma invece è un fraintendimento della legge, in quanto esperti avvocati sia di Red Hat, sia della Free Software Foundation, sull'argomento hanno confutato questa affermazione. Infine, non vi è la confusione di avere due licenze comuni diverse con lo stesso nome.

Potete utilizzare un'altra licenza preesistente nota per essere compatibile con la GPL, anche se l'uso di una licenza insolita rischia di causare il mancato supporto al vostro progetto da parte degli sviluppatori (alcuni di essi non vogliono i rischi e i problemi di lavorare sotto una licenza non familiare e inusuale). Il mio [FLOSS License Slide](#) mostra come diverse comuni licenze siano compatibili (o meno).

3. Una terza alternativa è il doppio licenziamento (o anche triplo licenziamento) del programma. Cioè, rilasciare sotto più di una sola licenza allo stesso tempo, una dei quali sia compatibile con la GPL (come la GPL o LGPL), e lasciare che l'utente scelga quale licenza userà. Progetti ampiamente utilizzati come Perl, Mozilla, e Qt fanno ciò (il [progetto Mozilla utilizza anche una tripla licenza \(MPL/GPL/LGPL\)](#)). Le [Debian Free Software Guidelines \(DFSG\) FAQ](#) notano che “La GPL è particolarmente comune nelle doppie licenze perché consente di collegare il codice con la grande massa di codice GPL disponibile, comprese alcune importanti librerie, di essere incorporato in altri lavori sottoposti a GPL... Ciò è quasi sempre dovuto ad un progetto che inizia con una licenza fatta in casa incompatibile GPL, poi ci si rende conto di aver commesso un errore e si scopre che il rilicenziamento in questa forma è la soluzione migliore”. Un problema con il doppio licenziamento è che potete *solo* incorporare codice esterno che sia compatibile con *tutte* le licenze. Così, il doppio licenziamento può rendere impossibile l'uso legale di qualche codice potenzialmente utile fino a che manterrete le licenze multiple. E' uno dei modi più ragionevoli per correggere un errore, ma in primo luogo è meglio evitare errori... specialmente quando c'è una prova evidente che l'uso di una licenza incompatibile con la GPL è un errore per i progetti FLOSS.

Se non potete usare nessuno di questi approcci più semplici, allora si può utilizzare pure una licenza compatibile GPL, ma a questo punto avrete bisogno dell'aiuto di un avvocato – ed in generale io sconsiglio la cosa. Ciò perché l'analisi della licenza può rapidamente diventare molto complicata. Per esempio, la FSF ha sviluppato la GPL e mantiene un [elenco di licenze che ritiene essere compatibili o meno con la GPL](#). Notate il numero di punti sottili messo in evidenza nella loro analisi! Il mio punto non è discutere la loro analisi legale (con cui non tutti concordano): voglio solo dimostrare che la determinazione della compatibilità GPL può essere complicata, talché esistono ottime ragioni per mantenere le cose semplici. [La FSF mette a disposizione qualche consiglio](#), ma dal momento che potrebbe non essere la proprietaria del software, ci potrebbe essere più spazio per contrasti di quanto potreste pensare. E in aggiunta, quando vengono rilasciate le revisioni della GPL, possono facilmente sorgere nuovi problemi. E' probabile che sia molto più difficile aggiornare

le licenze se disponete di una strana licenza unica. In breve, il modo migliore per rendere un programma compatibile GPL è quello di mantenere le cose semplici.

Se state presentando delle modifiche ad un progetto FLOSS esistente che utilizza una licenza incompatibile GPL, includete una nota nella vostra proposta che state sottoponendo a doppia licenza le vostre modifiche con una licenza compatibile GPL (come la licenza GPL, LGPL o MIT/X): ciò renderà molto più facile per il progetto accordare successivamente la doppia licenza all'intero programma.

Se state rilasciando del software sotto la GPL (da sola o come doppia licenza), dovrete includere la clausola “versione X o successive” [“version X or later”] come raccomandato nella GPL stessa e utilizzata da quasi tutti gli utenti GPL. L'uso della frase “versione X o successive” rende molto semplice il passaggio al testo aggiornato della GPL e garantisce la futura compatibilità sia che la GPL venga aggiornata o precisata, sia che sorgano nuove situazioni. Un altro testo nella GPL limita ciò che quelle versioni successive diranno e le nuove versioni della GPL con possono *ridurre* ciò che potete fare con il lavoro originale. La maggior parte delle licenze proprietarie odierne hanno clausole che anticipano le modifiche per quanto riguarda il passare degli anni - aggiungendo “o successiva” è solo un modo diverso per farlo. Tuttavia alcune società non riescono proprio a portarsi a dire “la versione 2 o successive”: in tali casi dovrebbero designare un rappresentante che sarà autorizzato a rilasciare il codice sotto una versione successiva della licenza, non appena quelle nuove versioni saranno disponibili, a stabilire un processo per esaminare tali versioni successive e a concedere il permesso una volta che il controllo abbia determinato che tutto va bene. Ma credo che l'approccio “versione X o successive” sia veramente il miglior approccio se scegliete di utilizzare la GPL: l'intero scopo della GPL è assicurare che possiate avere l'accesso a successivi miglioramenti ad un programma e le revisioni della GPL sono create specificamente per proteggere contro attacchi a quell'accesso.

Se scegliete la GPL, quale versione della GPL dovrebbe essere la vostra base? Per molti anni la GPL versione 2 è stata la sola vera scelta, ma nel 2007 è stata rilasciata la GPL versione 3. La GPL versione 3 ha un certo numero di miglioramenti se il vostro scopo è assicurare che i destinatari possano modificare il loro software: per esempio, semplifica l'applicazione internazionale (cambiando i termini USA-centrici in termini più ampi), aggiunge difese brevettuali migliori, è compatibile con più licenze (in particolare la Apache Public License e la Affero GPL), e neutralizza la “Tivoization” (gli utenti devono essere in grado di modificare il software sotto GPL che ricevono). Sono disponibili commenti più dettagliati sulla GPLv3, compresi quelli di [Glyn Moody](#), [Mark Radcliffe](#) e [Luis Villa](#). [Palamida sta rintracciando quali progetti si stanno aggiornando alla GPL versione 3](#), un processo che sembra [procedere bene](#). La “GPL versione 2 o successive” fornisce la compatibilità massima, ma dal momento che quasi tutti i progetti hanno la clausola “o successive”, per molta gente non è realmente molto più compatibile della “GPL versione 3 o successive”. Mi attendo che ci sarà una graduale migrazione di molti progetti dalla “GPL versione 2 o successive” alla “GPL versione 3 o successive”. Così per la maggioranza dei progetti la “GPL versione 3 o successive” [“GPL version 3 or later”] è probabilmente la scelta migliore dal momento che vi lascia cogliere il vantaggio dei miglioramenti della GPL versione 3.

In breve, accertatevi che il vostro software FLOSS sia compatibile GPL e di usare una delle poche licenze FLOSS standard (GPL, LGPL, nuova BSD, MIT e forse la licenza Apache 2.0). In caso contrario il vostro progetto FLOSS potrebbe non avere mai il supporto da voi sperato.

Appendice A: Il racconto ammonitore di XFree86

Se state cercando un racconto ammonitore su come le cose possono mettersi male a causa di una licenza incompatibile con la GPL, non occorre che andiate molto lontano alla tragica storia della morte di XFree86. Il progetto XFree86 storicamente ha portato allo sviluppo di un popolare server X e tradizionalmente la vasta maggioranza del suo codice ha fatto uso della semplice licenza open source "MIT/X", che è compatibile GPL. Un tempo, se avete utilizzato un sistema simil-Unix e un'interfaccia utente grafica, è molto probabile che abbiate usato XFree86 ... è così che si è diffuso.

Il presidente di XFree86, David Dawes, decise di cambiare la licenza di XFree86 con una che non era compatibile GPL, principalmente per dare maggior credito agli sviluppatori. Questo proposto cambiamento di licenza causò un grave tumulto. [Jim Gettys, uno sviluppatore molto rispettato e co-fondatore di X, si oppose strenuamente a questo mutamento di licenza di XFree86](#), sebbene non fosse un acceso sostenitore della GPL. [Richard Stallman chiese che fosse fatto qualcosa](#). Un articolo su [Linux Today](#) e una [discussione su Freedesktop.org](#) rivelarono che Red Hat, Debian, SuSE, Gentoo, Mandrake e OpenBSD avevano pianificato di abbandonare Xfree86 se fosse passata a questa nuova licenza.

Non tutte le parti sollevarono obiezioni a causa della incompatibilità con la GPL: l'obiezione di Theo de Raadt fu che la nuova licenza aveva reso il codice "meno libero", piuttosto che specificatamente sulla compatibilità con la GPL. Ma molti altri protestarono proprio per questa modifica proposta sul presupposto che una licenza incompatibile GPL è inaccettabile. Branden Robinson fece una dettagliata analisi della licenza XFree86 (disponibile negli elenchi postali di [XFree86](#) e [The Open Group](#): una versione più corta fu postata da [Debian](#)). In questa dettagliata analisi, supplicò che XFree86 rimediasse ai pochi problemi di licenza correntemente nella base del codice e di mantenere il programma compatibile GPL, dicendo: "Il percorso per l'eliminazione delle incompatibilità con GPL dovute alla clausola pubblicitaria BSD per l'intero albero dei sorgenti di XFree86 (come di XFree86 4.3.0, in ogni modo) sembra abbastanza breve. Dato ciò, sembra un peccato radicare una simile incompatibilità sia in estensione che in profondità". [L'elenco postale di XFree86 da Febbraio 2004](#) include molte dichiarazioni di persone affermant che la compatibilità GPL è per loro importante.

Dal momento che la gente di Xfree86 non voleva passare ad una licenza compatibile con la GPL, [la X.Org Foundation \(costituitasi nel gennaio 2004\) annunciò la sua personale versione di X il 6 aprile 2004](#). Uno dei suoi miglioramenti chiave fu che "soltanto i file senza la nuova licenza di Xfree86 1.1 erano stati inclusi nel rilascio X11R6.7.0". La [sede della fondazione X.org](#) rese ancora più chiaro che la compatibilità con la GPL era il problema chiave: disse che "l'archivio di X.Org è basato su XFree86 4.4 RC2. Appena prima del suo rilascio 4.4, XFree86 ha adottato una nuova licenza forse incompatibile con la GPL. Per tale motivo, abbiamo ricreato la sua struttura nel modo più simile possibile senza importare i file sottoposti alla nuova licenza".

[La maggior parte delle scissioni ebbero uno scarso supporto e presto morirono](#), ma la scissione della fondazione X.Org fu immediatamente approvata da molte organizzazioni chiave, comprese la SUSE di Novell, Red Hat, HP, TrollTech e FSF Europe fra le altre. [FreeBSD successivamente abbandonò XFree86](#), e come notato sopra, il leader di OpenBSD aveva già deciso che non avrebbe supportato l'approccio di XFree86. Dal [Luglio 2004, Linux Weekly News \(LWN\) poté riferire con sincerità che quasi tutti gli sviluppatori di XFree86 erano passati alla nuova scissione compatibile con la GPL](#), lasciando XFree86 quasi del tutto morto. Il principale argomento di discussione nella lista postale

della nuova scissione fu quali nuove caratteristiche e miglioramenti di progettazione dovessero essere fatti per primi, con la presunzione che qualsiasi cosa avesse fatto XFree86 fosse irrilevante. La transizione fu rapida e si può vedere esaminando la lista postale del “XFree86 forum”: l'archivio del [giugno 2003](#) di questa lista ebbe 20 messaggi su XFree86 (più 13 spam senza importanza) e tutti 20 furono delle discussioni tecniche relative al miglioramento del programma. Nel [giugno 2004](#) ci furono solo 13 messaggi di rilievo nell'intero mese (più 62 messaggi di spam) e invece di essere tutte discussioni tecniche, molti di quei messaggi furono ammissioni tra i membri della lista che XFree86 era stata abbandonata dalla maggior parte dei suoi sviluppatori e utenti, oppure ci fu una discussione su come FreeBSD sarebbe potuta eventualmente restare con XFree86 ([FreeBSD più tardi lasciò XFree86, azzerando questa speranza](#)). Per esempio, William M. Quarles rispedì una conversazione in chat affermando che “La maggioranza delle principali distribuzioni o hanno abbandonato XFree86 e sono passate a X.Org, oppure stanno [ancora] utilizzando XFree86 4.3 [la vecchia versione non più mantenuta]”.

Ora è vero che ci furono più problemi della incompatibilità con la GPL. Ci furono vari inviti a ristrutturare l'organizzazione degli sviluppatori e ad aprire (e accelerare) lo sviluppo. Ma nessuno di questo problemi determinò un repentino esodo degli sviluppatori ed utenti. Tutti i progetti (proprietary e FLOSS) discutono sulla loro struttura e progressi, e raramente fanno sì da condurre ad un abbandono immediato da parte di tutti gli utenti e gli sviluppatori in una volta. Il problema (o almeno l'ultima goccia) che determinò l'esodo di massa degli utenti da Xfree86 fu il tentativo di passare ad una licenza incompatibile con la GPL. E' chiaro che provare a passare un popolare programma FLOSS da una licenza compatibile GPL ad una incompatibile GP on la GPL può andare incontro ad una dura e rapida resistenza.

10 Appendice B: Informazioni sulla GPL e sulle altre Licenze.

Esistono molte fonti d'informazione sulla GNU General Public Licence (alias GPL). [La GPL stessa](#) è disponibile in rete. [Le domande poste frequentemente sulla GPL \(Frequently Asked Questions o FAQ\)](#) hanno più informazioni da parte del loro autore. [Groklaw mantiene una pagina di risorse sulla GPL](#). “[The Penguin Paradox: How the Scope of Derivative Works in Copyright Affects the Effectiveness of the GNU GPL](#)”, 85 B.U. L. Rev. 1439 (2005) di Mitchell Stoltz discute della GPL e delle definizioni legali circa le opere derivate.

Se state *utilizzando* o *modificando* del software rilasciato sotto la GPL, guardate [A Practical Guide to GPL Compliance](#) del [The Software Freedom Law Center \(SFLC\)](#). Come osservato in [How the GPL is enforced](#), un numero deprimente di società prende semplicemente il software sotto licenza GPL e lo utilizza in modi non consentiti dalla licenza, anche se esse vendono prodotti con licenze di gran lunga più complesse che si aspettano che i loro utenti osservino. “I numerosi casi in tempi recenti... hanno seguito tutti lo stesso schema: un produttore usa software licenziato con la GPL nel firmware del suo apparecchio, ma né avvisa gli acquirenti dello stesso circa i loro diritti sul software, né da loro accesso al codice sorgente, come richiede la GPL. Un contravventore viene inizialmente contattato in modo diretto, senza pubblicità, e viene fatto un tentativo per comporre la questione bonariamente... [e se non riesce, si sporge un reclamo] alla fine c'è di solito un accordo e nessuna sentenza giudiziale... Tutti gli accordi appaiono molto simili: la società ammette la violazione, informa retrospettivamente i suoi clienti, deve nominare una persona responsabile per la prevenzione qualsiasi violazione futura della GPL, e corrispondere un risarcimento... Le società in realtà non dovrebbero ricadere nella trappola di nuovo e di nuovo... [perché è disponibile una guida

come SFLCA]”.

Esiste un gran numero di licenze FLOSS, sebbene la vasta maggioranza del software sia coperta dalle cinque più comuni. Alcune sono "protettive", come la GPL o la Affero GPL – funzionano per proteggere il software dall'essere trasformato in un prodotto proprietario, anche come parte di un'opera più grande. Alcune sono "permissive", come la nuova BSD e la licenza MIT/X: esse consentono di trasformare il programma in un lavoro proprietario. E alcune licenze come la LGPL sono compromessi tra "protettive" e "permissive" - la LGPL, per esempio, protegge il componente (solitamente una libreria) ma consente il suo inserimento in un'opera proprietaria più grande. [La FSF mantiene un elenco delle licenze e di alcune analisi su di esse](#). La [lista delle licenze dell'Open Source Initiative](#) elenca licenze che ha stabilito essere licenze di software open source. [FOSSology](#) è uno strumento FLOSS che riporta le licenze identificate nel codice sorgente.

Appendice C: progetti GNU non sotto una licenza copyleft

La Free Software Foundation (FSF) sostiene l'utilizzo della GNU General Public License (GPL), ma accetta volentieri anche l'impiego di software non soggetto a GPL come parte di loro progetto GNU.

Quelli seguenti sono “pacchetti GNU” non sottoposti a licenza copyleft:

- Kawa [fsf.org]: licenziato sotto la Kawa License [gnu.org], che è una licenza in stile X11/MIT. Kawa è un ambiente Lisp Scheme/Emacs che gira in Java.
- GNU less [fsf.org]: il più recente tarball da ftp.gnu.org è GPL, ma le versioni più vecchie non sono copyleft.
- Ncurses [gnu.org], che è distribuito sotto una licenza stile X11.
- Proto [fsf.org], che è uno strumento di sviluppo software per trovare prototipi di funzioni C e C++, ed è di pubblico dominio.
- Quexo [fsf.org], implementazione di xquery che ricorre a Kawa per compilare java bytecode. Sotto la licenza Kawa, che significa una licenza stile X11.
- Speex [fsf.org], un codec di compressione audio per la voce, sotto la licenza Xiph.org (una BSD modificata) .

(I miei ringraziamenti a [Javier Candeira](#) che ha identificato molti di questi).

David A. Wheeler si diverte ad apprendere e scrivere di FLOSS e sicurezza IT, e trascorre anche troppo del suo tempo libero su di essi. E' l'autore di testi come [Why Open Source Software / Free Software \(OSS/FS or FLOSS\)? Look at the Numbers!](#), [How to Evaluate Open Source Software/Free Software \(OSS/FS\) Programs](#), [Open Source Software \(OSS\) in U.S. Government Acquisitions](#), [FLOSS License Slide](#), [Open Source Software/Free Software \(OSS/FS\) References](#), e del [Secure Programming for Linux and Unix HOWTO](#). Ha anche sviluppato alcuni programmi FLOSS come [flawfinder](#) e [sloccount](#). Suona pure il piano e la chitarra. Si rumoreggia che in realtà non dorma, il che spiegherebbe molte cose. Potete vedere la sua pagina web su <http://www.dwheeler.com> per una prova aggiuntiva di ciò.

