

Open source, software libero e altre libertà - Le licenze di software libero (open source)

Carlo Piana



Esportato da Wikisource il 8 marzo 2023. Segnala eventuali errori su it.wikisource.org/wiki/Segnala_errori

Le licenze di software libero (open source)

A volte alla domanda “sotto quale licenza è questo programma?” si sente rispondere “come che licenza?! Open source!”. Il che è come rispondere “nome maschile” alla domanda “come ti chiami?”. Come direbbero i francesi “*vive la petite différence!*” Una licenza è – al fondo delle cose – un documento legale. Tuttavia è anche una dichiarazione programmatica, un manifesto, ma siccome il software libero è un fenomeno giuridico, l’aspetto contrattuale o para-contrattuale è ciò che caratterizza, da molti punti di vista, la licenza. Ovvero, risponde alla domanda “che cosa ci posso fare con il software che sto analizzando?”.

Software libero, software non libero; software open source, software non open source

Torniamo alla prima risposta: “è software open source”. Uso il titolo di una presentazione che tenni qualche anno fa a Firenze, parafrasando un più noto *pamphlet*¹ di Paolo Rossi (il comico), “Si fa presto a dire open source”. Per dire che una cosa è open source, occorrerebbe prima sapere cosa distingue l’open source dal non open source, e il software libero dal non software

libero. Cosa distingue il software libero dal software open source è molto semplicemente l’approccio filosofico: in termini giuridici e pratici, invece, la differenza è pressoché nulla. Le due nomenclature sono operativamente intercambiabili, per cui non ci accapigliamo, ognuno usi quella che preferisce, io preferisco dire “software libero”. Altri dicono “open source”, altri dicono “Free and Open Source Software”, abbreviato nell’acronimo “FOSS”.

Però i due mondi fanno effettivamente riferimento a due differenti **definizioni**. La definizione di software libero è semplice: è software libero il software la cui licenza soddisfa tutte e quattro le libertà del software, di cui abbiamo già detto nel precedente capitolo, quando ci siamo occupati della storia del software libero. Se ne manca anche solo una, non è software libero. La definizione di cosa sia software open source è un po’ più complessa.

La Open Source Definition

Il termine stesso open source nasce successivamente al software libero. Abbiamo già detto che il primo globale e cosciente sforzo di creare software libero come tale appartiene a Stallman e alla Free Software Foundation, e consisteva nel sistema operativo GNU. Con l'avvento di Linux, GNU divenne un sistema operativo pienamente funzionante. Una delle prime e più "pure" distribuzioni di GNU/Linux fu Debian.

Cos'è una distribuzione? Una distribuzione è un assemblaggio di software proveniente da fonti diverse, da parte di un operatore che sceglie cosa inserire e come configurare tale software (normalmente, una versione di Linux, una certa quantità di software del progetto GNU, un sistema di gestione dei pacchetti per l'aggiornamento e l'installazione di componenti aggiuntive, applicazioni e così via), compila il software a partire dai sorgenti e lo distribuisce come un prodotto a se stante. Di qui il termine "distribuzione" o la sua versione slang abbreviata "distro". Tutto è GNU/Linux, ogni distribuzione è diversa dalle altre, ma anche simile alle altre.

Debian, appunto, è una distribuzione, che si caratterizza per due aspetti: un proprio sistema di gestione dei pacchetti (tutti con il suffisso .deb) e un relativo programma di gestione (apt), e una scelta piuttosto radicale su quali condizioni debbano rispettare le componenti per essere incluse. Si potrebbe dire "devono essere software libero", ma il progetto Debian preferì articolare meglio tale requisito e "spacchettare" le libertà in caratteristiche più puntuali, che inserì nelle "Debian Free Software Guidelines".

In uno sforzo di "vendere" meglio il concetto di software libero, e anche per risolvere una ambiguità del termine "software libero" (che in inglese si dice "Free Software", dove "free" può anche dire "gratis") alcuni attivisti e programmatori spinsero per usare un termine d'uso comune che non soffrisse dell'ambiguità "Free Software = software gratuito", ovvero "open source" ("sorgente aperto"). E venne a tal fine fondata la "Open Source Initiative" (OSI) che si diede il compito di "certificare" cosa fosse open source e cosa no. Come riferimento normativo vennero usate le Debian Free Software Guidelines, alle quali vennero rimossi i riferimenti specifici alla distribuzione Debian, che divennero la Open Source Definition: una sorta di decalogo di caratteristiche che una licenza deve rispettare per potersi fregiare del termine open source.

Proliferazione

OSI ha nel frattempo approvato qualcosa come quasi un centinaio di licenze diverse². Che non esauriscono affatto l'orizzonte delle licenze possibili ed esistenti (ed usate). È un numero enorme, ed averne così tante è un problema. Se avessimo poche licenze, con clausole in gran parte identiche e qualche clausola difforme, trovare una interpretazione solida e affidabile, che ricostruisca con una certa qual certezza le conseguenze giuridiche di ciascuna licenza e le condizioni alle quali tali licenze possano essere usate assieme, avremmo un mondo più semplice.

Invece abbiamo licenze che in larga parte sono simili tra loro (appartengono solitamente a una delle grandi famiglie, di cui diremo), ma con differenze che possono a volte determinare ambiguità, passare inosservate e – quel che è peggio – causare incompatibilità reciproche, soprattutto quando si passa al copyleft. Seguendo le discussioni sulle nuove proposte di licenze all'OSI ci si rende conto che in larga parte l'esigenza che tali licenze soddisfano sono o l'ego di chi le propone, per avere il proprio nome legato a una licenza (che poi non userà quasi nessuno) oppure... l'ego di chi le propone, che pensa che le centinaia di menti che collettivamente hanno sviluppato quelle esistenti e adottate dalla maggior parte del software siano incompetenti. Solo in alcuni casi, alcune licenze servono a soddisfare esigenze specifiche e sono in parte giustificabili, nella maggior parte dei casi la mia opinione è che **meno è meglio**. Nel dubbio se usare una licenza e scriverne una, contare fino a diecimila e poi comunque **abbandonare l'idea**.³

Le famiglie delle licenze: divisione tra vari livelli di copyleft

Dicevo, esistono grandi famiglie di licenze, che coprono da sole la gran parte del software realizzato e distribuito. Il resto è “coda lunga” (*long tail*), traducibile anche con “rumore di fondo”: poco rilevante, ma sempre rumore.

Uno degli aspetti più importanti delle licenze, il primo che ricerco in una nuova licenza, è: “quanto copyleft?”. Solitamente si definiscono tre livelli di copyleft, nel software: **copyleft forte**, **copyleft debole**, **nessun copyleft** (alle tre si aggiunga anche il cosiddetto “*network copyleft*”, di cui parleremo più avanti).. Nessuno è in grado di definire con precisione dove inizia uno e finisce l'altro; teniamo queste distinzioni come categorie di massima.

Licenze non copyleft

“**Nessun copyleft**” è abbastanza facile da capire: il codice sviluppato può essere preso, modificato, rilicenziato, senza che il codice così risultante debba soggiacere alla stessa licenza. Le licenze che seguono questo paradigma vengono dette anche “ultraliberali”, nel senso che consentono di fare quel che si vuole (una si chiama appunto “WTF – *What The Fuck [you want]* – license”: non traduco per decenza).

In ciò rientra anche rendere il software interamente proprietario, secondo la parabola dei primi UNIX, di cui abbiamo parlato nello scorso capitolo su cui non torneremo.

In tali licenze, che per la tradizione storica vengono anche dette “accademiche” abbiamo licenze legate appunto all’ambito universitario: la BSD (dove “B” sta per “Berkeley”), e, meno diffusa, la MIT. La BSD in realtà esiste in tre sfumature, a seconda di quante clausole è composta (la più utilizzata è senz’altro la three-clauses). La MIT, invece, esiste in una serie quasi infinita di varianti. Sono licenze molto semplici, in cui le principali condizioni sono quelle di riconoscere l’assenza di responsabilità dello sviluppatore e quella di indicarne il nome dell’autore quando si distribuisce il codice sorgente. Queste licenze sono molto popolari tra alcuni sviluppatori perché sono molto semplici e non vi abbonda il “legalese”.

Alle accademiche si accompagna un’altra licenza molto utilizzata, la **Apache Public License**, licenza usata dalla Apache Software Foundation, che si occupa del più famoso e utilizzato server web. La Apache, al contrario delle accademiche, è una licenza lunga, che comprende anche una clausola di risoluzione nel caso di uso aggressivo dei brevetti da parte di un licenziatario.

Licenze di copyleft forte

All’altro estremo abbiamo il **copyleft forte**. Tendenzialmente una licenza di copyleft forte tende a estendere il suo effetto vincolante il più possibile a qualunque “opera derivata” di software copyleft. Il confine di tale effetto è una delle questioni più aspramente dibattute tra coloro che si occupano della nostra materia. Il copyleft forte richiede anche una struttura normativa più complessa e stringente, perché essendo “restrittivo” (nel senso che tende a imporre condizioni più stringenti ed effettive al fine di mantenere le libertà del software), richiede che

tutti i “buchi” siano tappati, tutte le scappatoie siano evitate, in modo che il copyleft forte, e il tentativo di mantenere il software un common resista a chi cerca di evitarlo. Creare una licenza di copyleft forte è molto difficile, crearne una fatta bene è cosa a portata di ben pochi.

Siccome le licenze di copyleft forte hanno condizioni più stringenti e peculiari, è molto facile che una di esse sia incompatibile con qualsiasi altra licenza di copyleft forte, dimodoché combinare software sotto due licenze di copyleft forte è in pratica impossibile. Anche combinare copyleft forte con copyleft debole, e a volte anche con non copyleft, si rivela sovente impossibile. Se non si possono rispettare le condizioni di licenza – tutte le condizioni – non si può usare il software, non si possono trarre opere derivate.

La licenza di copyleft forte per antonomasia nonché la più antica è la **GNU General Public License**, o **GPL** della Free Software Foundation (FSF). Giunta alla terza versione, nella versione 2 è storicamente di gran lunga la più utilizzata licenza di copyleft forte, e la più utilizzata licenza in assoluto. Discutere di cosa c’è nella GPL richiederebbe un volume intero, non pochi paragrafi. La versione 3 è incompatibile con la versione 2, a meno che lo sviluppatore non abbia utilizzato la clausola di “aggiornamento” (tipo “o qualsiasi successiva versione”) che consente a chiunque di trasformare una versione di licenza più vecchia con una nuova.

Esistono altre licenze di copyleft forte, come ad esempio la EUPL, nata nell’ambito delle istituzioni dell’Unione Europea. Creare una nuova licenza di copyleft forte, lo si capirà da quanto detto, non è una buona idea, la EUPL non lo è stata. Fortunatamente la EUPL contiene una clausola di compatibilità che consente di utilizzare, per le opere derivate, una lista di altre licenze, tra le quali la GNU GPL, per cui le conseguenze dannose sono in via pratica evitate.

Licenze di copyleft debole

Il **copyleft debole** è più recessivo rispetto al copyleft forte. Come idea, il copyleft debole opera a livello di file; per cui se uno sviluppatore appone le proprie modifiche a quel file, quel file rimane sotto la stessa licenza. Mentre se il codice sorgente di quel file viene compilato (“linkato”) con altro codice sorgente per creare un file oggetto che li comprenda (un’opera più ampia), il file oggetto risultante – pur incorporando codice copyleft – non necessita di essere rilasciato sotto le stesse condizioni, dunque non c’è interferenza in caso i due file sorgente

siano sotto licenze incompatibili, e il prodotto risultante può anche essere sotto licenze proprietarie. Spesso infatti le licenze di copyleft debole sono usate per file destinati ad essere incorporati in altro software, tipicamente librerie (vedi sotto per la LGPL). Questo è detto in modo molto brutale, ma serve a dare l'idea.

Per il copyleft debole esistono due licenze paradigmatiche. La **GNU Lesser General Public License** (un tempo “Library General Public License”, in quanto nata per le librerie Glibc, le librerie del compilatore C del progetto GNU) o **LGPL**; e la **Mozilla Public License** o **MPL**, la licenza di Firefox, per intenderci, uno dei più diffusi browser web (poi adottata anche da LibreOffice). Entrambe sono compatibili con la GPL: anche la MPL, nella versione 2, in quanto contiene ha una espressa clausola di compatibilità con la GPL.

La caratteristica di essere compatibili con la GPL, vuoi per le condizioni, vuoi per un'espressa condizione di compatibilità, è una caratteristica molto importante delle licenze, alla pari, quasi, con il fatto di essere copyleft di un certo tipo. La FSF pubblica una lista di tali licenze, secondo la propria interpretazione, ovviamente.

Il copyleft “di rete”

Nella tripartizione or ora delineata (no copyleft, copyleft debole, copyleft forte), abbiamo preso in considerazione una versione tradizionale del copyleft. Avrete notato che è stato ripetuto il verbo “distribuire”, o “rilasciare” (che è sostanzialmente un sinonimo). Questo perché uno dei principi cardine di qualsiasi licenza copyleft, quale che ne sia l'intensità, è che l'uso personale del software, anche modificato, non richiede il rispetto di alcuna condizione. In altre parole, l'utilizzatore è **libero** di copiare, studiare e modificare il software a suo piacimento, senza costrizione alcuna. Per chi non distribuisca a terzi il software, ma lo utilizzi solo internamente, non vi è alcuna differenza a livello pratico tra le varie forme di licenza, il copyleft non è una faccenda rilevante.

Il copyleft scatta con la **distribuzione**.

Solo per chi distribuisce il software si pone dunque un problema effettivo di rispetto delle condizioni di licenza. Cosa accade, però, in quelle situazioni in cui non vi sia effettiva distribuzione, ma ciò non di meno il software venga messo a disposizione di terzi? In tal caso le licenze tradizionali nulla possono. Questo fenomeno è da tempo fonte di preoccupazione per alcuni osservatori, addirittura

da prima che per il cosiddetto “Software as a Service” divenisse una forma effettiva di utilizzo del software, e invece di “cloud” si parlava di “Application Service Provider” (fornitore di servizi di applicazioni, abbreviato “ASP”). Infatti al fenomeno si diede il nome di ASP loophole (trucco dell’ASP), ovvero la possibilità per uno sviluppatore di poter sfruttare software copyleft, metterlo a disposizione di terzi, ma senza consegnare il codice, essendo così libero dalle incombenze del copyleft (forte o debole) in quanto le condizioni non sono attivate. In questo modo il fornitore in cloud beneficia di un bene comune, lo sfrutta commercialmente, ma non restituisce al common alcunché.

Per tentare di ovviare a tale problema, Bradley Kuhn, un attivista americano, concepì una clausola aggiuntiva alla GNU GPL v.2 che venne conosciuta come “Afferro clause” (dal nome del progetto per cui Kuhn lavorava e per cui concepì la clausola). La Afferro è l’unica condizione di copyleft che viene attivata non dalla distribuzione, ma dall’uso del software in una particolare situazione: quella sfruttata per dare accesso tramite interfacce di rete al software modificato. In tali condizioni, chi modifica il software deve mettere a disposizione, tramite la stessa interfaccia di rete, l’intero codice sorgente corrispondente alla versione utilizzata. La clausola Afferro, nata come un’aggiunta alla GPL, ha poi trovato spazio in una licenza sua propria, denominata GNU AGPL, nella quale rappresenta la clausola 13.1.

Concludendo

Solo riassumendo le licenze di cui abbiamo fatto menzione, abbiamo:

- tre tipi di BSD;
- la MIT (multiforme);
- Due versioni di Apache;
- Due versioni di LGPL;
- Due versioni di Mozilla;
- Due versioni di GPL;
- La AGPL;
- Due versioni di EUPL.

Dunque quindici licenze. In più, ci sono in circolazione diverse versioni della GPL con “eccezioni”, studiate appositamente per rendere compatibile il software con altro software sotto licenza diversa. Ce n’è veramente a sufficienza per non

andare a cercare altre licenze e per dare materiale di studio ai legali. E questo senza contare la possibilità che il software sia sotto pubblico dominio.

Pensavate fosse facile? Nulla in questo campo è facile. Ma è tutto affascinante. Ora, almeno, avete una mappa ragionevolmente completa delle licenze di software. Nel prossimo capitolo ci occuperemo di alcuni aspetti delle licenze di software copyleft, del dual licensing e di alcuni aspetti peculiari dello sviluppo di software sotto tali licenze. In altre parole, dei modelli di business legati alle o consentiti dai diversi tipi di licenze.

1. ↑ Paolo Rossi, *Si fa presto a dire “pirata”*, Dalai Editore, Milano, 1997.
2. ↑ Per la precisione alla data di scrittura del presente libro risultano 80 *licenze* “current”, più 15 ritirate. Per la lista completa e aggiornata si veda <https://opensource.org/licenses/alphabetical>.
3. ↑ Parlo come autore di una licenza sottoposta e mai approvata da OSI: la licenza MXM (<https://www.linuxjournal.com/content/should-open-source-licence-ever-be-patent-agnostic>), sviluppata con Leonardo Chiariglione per MPEG.

Informazioni su questa edizione elettronica:

Questo ebook proviene da [Wikisource in lingua italiana](#)^[1]. Wikisource è una biblioteca digitale libera, multilingue, interamente gestita da volontari, ed ha l'obiettivo di mettere a disposizione di tutti il maggior numero possibile di libri e testi. Accogliamo romanzi, poesie, riviste, lettere, saggi.

Il nostro scopo è offrire al lettore *gratuitamente* testi liberi da diritti d'autore. Potete fare quel che volete con i nostri ebook: copiarli, distribuirli, persino modificarli o venderli, a patto che rispettiate le clausole della licenza [Creative Commons Attribuzione - Condividi allo stesso modo 3.0 Unported](#)^[2].

Ma la cosa veramente speciale di Wikisource è che **anche tu** puoi partecipare.

Wikisource è costruita amorevolmente curata da lettori come te. Non esitare a unirti a noi.

Nonostante l'attenzione dei volontari, un errore può essere sfuggito durante la trascrizione o rilettura del testo. Puoi segnalarci un errore a questo indirizzo:

http://it.wikisource.org/wiki/Segnala_errori

I seguenti contributori hanno permesso la realizzazione di questo libro:

- Alex brollo
- OrbiliusMagister

Il modo migliore di ringraziarli è diventare uno di noi :-)

A presto.

1. [↑ http://it.wikisource.org](http://it.wikisource.org)
2. [↑ http://www.creativecommons.org/licenses/by-sa/3.0/deed.it](http://www.creativecommons.org/licenses/by-sa/3.0/deed.it)