



ICT E DIRITTO

Rubrica a cura di

Antonio Piva, David D'Agostini

Scopo di questa rubrica è di illustrare al lettore, in brevi articoli, le tematiche giuridiche più significative del settore ICT: dalla tutela del *domain name* al *copyright* nella rete, dalle licenze software alla *privacy* nell'era digitale. Ogni numero tratterà un argomento, inquadrandolo nel contesto normativo e focalizzandone gli aspetti di informatica giuridica.

Licenze software e Open Source

David D'Agostini, Antonio Piva

1. LICENZE SOFTWARE

Fin dalla sua comparsa, il software ha posto delle serie problematiche riguardo alla sua tutela giuridica. Prima del suo avvento, due erano i principali istituti giuridici di tutela delle opere dell'ingegno umano, il diritto d'autore e il brevetto industriale: il primo concepito originariamente per le creazioni artistico-espressive, il secondo per le invenzioni di carattere tecnico-industriale. Mentre il diritto d'autore mira a proteggere solo la forma espressiva dell'opera, la tutela brevettuale ha per oggetto il contenuto dell'invenzione. Il software si presenta però come un'opera atipica in quanto si colloca proprio a cavallo delle due categorie di opere dell'ingegno: la sua vocazione di opera destinata alla soluzione di problemi tecnici, lo avvicinerebbe ineluttabilmente alla categoria delle invenzioni dotate d'industrialità; d'altro canto, però, il software risulta carente del requisito della materialità, fondamentale per la brevettabilità.

Nel 1980 il legislatore statunitense stabilì quale disciplina applicare al software, ovvero la tutela per mezzo del copyright. Dopo un decennio, il legislatore italiano, recependo con il d. lgs. 518/1992 la direttiva comunitaria 1991/250/CEE, inserì il software tra le opere sottoposte a tutela del diritto d'autore. Il diritto continentale europeo e in particolare quello italiano, avverte la distinzione tra gli inalienabili diritti morali d'autore e quelli patrimoniali, ovvero tutto il fascio di diritti passibili di una valutazione economica (e quindi alienabili) che ruotano attorno alla gestione e alla diffusione dell'opera. Gli ordinamenti di *common law*,

come gli USA non presentano una simile dicotomia, considerando il diritto d'autore come sempre rilevante dal solo punto di vista patrimoniale.

Data la sussistenza dei diritti esclusivi derivante dalla tutela d'autore, è necessario che l'autore del software stipuli un contratto con l'utente, nel quale definisca di quali prerogative si spogli e di quali quindi possa essere investito l'utente, ovviamente per contro di un corrispettivo. Dal momento dell'ingresso del software nella schiera dei beni di consumo, gli autori iniziarono a stilare dei contratti di portata generale in cui esprimevano i termini della distribuzione e della riproduzione del programma su cui essi vantavano i suddetti diritti esclusivi: nacque così il tipo contrattuale della licenza d'uso software. [1]

Il software, a seconda del grado di libertà che il titolare dei diritti di tutela decide di attribuire agli utenti della sua opera, può essere classificato in un prospetto di categorie; in una scala di valore che parte da un grado minimo di libertà per l'utente verso il grado massimo, troviamo le seguenti ipotesi.

□ *Software proprietario*

Prevede molteplici e sovrapposte forme di distribuzione, tra le quali le più diffuse sono:

- *commerciale*, è il software rilasciato da una azienda per essere venduto su licenza, con o senza periodi di prova. Prevede il pagamento per l'uso della licenza da parte dell'utente per poter cominciare l'utilizzo del pacchetto software e/o per poter entrare in possesso del supporto (contenente il software eseguibile). L'accettazione della licenza e delle clausole

contrattuali da parte dell'utente avviene, alcune volte, per il solo fatto di aprire la confezione esterna in plastica trasparente del package, che equivale alla lettura e accettazione delle clausole contrattuali leggibili all'esterno della scatola, ovvero, con l'accettazione esplicita delle clausole contrattuali durante l'installazione del prodotto. Il software commerciale non consente la distribuzione libera ad altri utenti, anche se interni alla stessa organizzazione, ma è possibile distribuire, o consentire l'uso del software ad altri utenti solo se vengono acquistate le relative licenze.

- **Demo** (o versioni di prova), è il software di tipo commerciale che il fornitore ritiene utile distribuire in versione dimostrativa, con alcune funzioni inibite, o con alcune limitazioni d'uso (quali, ad esempio, il divieto di utilizzare il software in ambienti di produzione, ma solo in ambienti di laboratorio, progetti prototipali o progetti pilota), rispetto alla versione commerciale (piena) che può essere utilizzata in produzione. Dipende, quindi, da caso a caso se questo tipo di software può essere usato, copiato e distribuito liberamente, o no, concordemente con le clausole contrattuali siglate dal cliente con il fornitore.

- **Shareware**, sono i pacchetti software che si possono ottenere gratuitamente, in cui il programma, a differenza delle demo, si presenta completo delle funzionalità, ma con la possibilità di essere provato per un limitato periodo di

tempo. Allo scadere del periodo di prova, occorre pagare una certa somma all'autore, altrimenti il programma cessa di funzionare.

- **Ad-ware**, ovvero software gratuito che per l'utilizzo visualizza banner pubblicitari o installa nel computer dell'utente programmi spyware (software che raccolgono e inviano dati per usi solitamente commerciali). I costi di licenza sono così coperti in via indiretta dalle aziende inserzioniste.

- **Freeware** (da non confondere con Free Software), cioè software commerciale completo che viene ceduto in uso gratuito¹. In questo caso, l'aggettivo "free" è usato nel solo significato di gratuito, a differenza che nell'espressione "free software".

□ **Software Libero e Aperto**

È questa la categoria del software della quale fanno parte le due denominazioni di "free software" e software "open source" e la sottocategoria del copyleft.

□ **Public Domain Software**

Programmi che, per concessione dell'autore, non contengono alcuna indicazione di copyright, tanto da occultarne anche la paternità (aspetto "morale" del diritto d'autore); è perciò possibile utilizzarli, riprodurli e diffonderli liberamente, ma anche eventualmente modificarli e inglobarli all'interno di altro software non public domain (anche commerciale), in quanto non sussistono i limiti derivanti da licenza [2].

Nella tabella 1 sono riportate a confronto di-

	Open Source o Free		Proprietario		
	Non Copyleft	Copyleft	Chiuso	Shareware	Freeware
Codice sorgente disponibile	Si	Si	No	No	No
Libertà di redistribuzione	Si	Si	No	Si	Si
Modifica	Si	Si	-	-	-
Restrizione	Si	No	No	No	No
Versioni modificate sono sempre libere	No	Si	-	-	-
Liberamente scaricabili	Si	Si	No	Si	Si
Limitazioni d'uso temporali	No	No	No	Si	No
Possibilità di ritorno economico	Si	Si	Si	Si	No

TABELLA 1

In tabella sono riportate alcune differenze fra le categorie di licenze Open Source o Free, nello spirito del copyleft o meno ed alcune licenze per software proprietario rispetto ad alcune libertà e/o limitazioni

¹ Nel Freeware viene concesso gratuitamente l'utilizzo del programma eseguibile mentre il codice sorgente rimane all'autore. Viene concesso in uso di solito per motivi promozionali, per esempio per puntare sull'aspetto della manutenzione e sulla personalizzazione.

verse tipologie di licenze rispetto ad alcune libertà e/o limitazioni.

2. COPYLEFT, FREE SOFTWARE E SOFTWARE OPENSOURCE

La parola Copyleft [3] è nata da un gioco di parole che deriva e rovescia il concetto di copyright. Un software copyleft viene dichiarato sotto copyright e poi vengono riversate le garanzie di libertà per l'utente all'interno della licenza, ribaltando così il ruolo della stessa e creando un vincolo di tipo legale fra la disponibilità del codice e le libertà di utilizzo, modifica e redistribuzione dello stesso.

Il copyleft consiste quindi nel convertire le licenze d'uso, da decalogo degli obblighi dell'utente, in una sorta di statuto dei suoi diritti, intoccabili nel tempo e invariati presso terzi.

In maniera più o meno sovrapposta a questa categoria di licenze, troviamo altre due definizioni di licenze software:

□ Software Libero (Free Software)

Tale dizione indica una categoria di licenze predisposte inizialmente da Richard Stallman e adottate dalla *Free Software Foundation* (FSF) [4]. Le caratteristiche principali di questa modalità di distribuzione del software sono descritte nella definizione di Free Software che recita testualmente: *L'espressione "software libero" si riferisce alla libertà dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. Più precisamente, esso si riferisce a quattro tipi di libertà per gli utenti del software:*

- *Libertà di eseguire il programma, per qualsiasi scopo (libertà 0).*
- *Libertà di studiare come funziona il programma e adattarlo alle proprie necessità (libertà 1). L'accesso al codice sorgente ne è un prerequisito.*
- *Libertà di redistribuire copie in modo da aiutare il prossimo (libertà 2).*
- *Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio (libertà 3). L'accesso al codice sorgente ne è un prerequisito.*

□ Software Open Source

Tale dizione è frutto di una rivisitazione delle proposte della Free Software Foundation operata nel 1998 da un gruppo di persone (fra cui Todd Anderson, Chris Peterson e Eric Raymond che guidò alla costituzione della *Open*

Source Initiative (OSI) [5]), con l'obiettivo di avvicinare lo sviluppo del software libero a quello sviluppato dalle imprese commerciali. Questo ha portato all'elaborazione della cosiddetta *Open Source Definition* (OSD), che definisce dei criteri generali per riconoscere le licenze e le distribuzioni Open Source. La OSD contiene 10 condizioni che disciplinano gli aspetti informatici, economici e giuridici della metodologia Open Source. Le prime tre riportate di seguito, servono a garantire le libertà fondamentali:

- *distribuzione Aperta (Libera e Gratuita): diritto di fare copie del programma e di redistribuirle liberamente;*
- *accesso Aperto (al Codice Sorgente): il codice deve essere disponibile a costo di distribuzione e non obnubilato;*
- *modificabilità Aperta: diritto di poter intervenire sul programma e modificarlo (non si obbliga né si vieta che la licenza si propaghi al lavoro derivato).*

Le rimanenti 7 garantiscono l'assenza di discriminazioni e definiscono i diritti e dignità degli autori. Queste sono:

- *integrità del codice sorgente dell'autore;*
- *nessuna discriminazione contro persone o gruppi;*
- *nessuna discriminazione contro campi applicativi;*
- *i diritti offerti dalla licenza si applicano automaticamente a tutti i destinatari;*
- *la licenza non deve essere specifica ad un prodotto (cioè legata ad una particolare forma fisica di distribuzione);*
- *la licenza non deve porre vincoli su altro software allegato a quello licenziato (esempio, vincoli sui software che possono o meno essere utilizzati assieme a quello licenziato);*
- *la licenza non deve richiedere particolari tecnologie di accesso (esempio, vincoli sui prodotti utilizzabili per accedere/utilizzare il software licenziato).*

Dunque, la OSD non è un modello di licenza per software Open Source, bensì una specifica di quanto è ammesso in una licenza software perché possa essere considerata Open Source. Attualmente la Open Source Initiative riconosce 59 licenze Open Source.

Dall'analisi delle definizioni di licenza Free Software e Open Source, si può notare come queste abbiano molti punti di contatto, tant'è che buona par-

te delle licenze Free Software risultano anche essere Open Source e viceversa. Tuttavia esistono licenze, anche se non molte, che sono Free Software ma non Open Source e viceversa [6].

Si presentano ora alcune tra le più importanti e diffuse licenze Open Source.

3. LICENZE OPEN SOURCE

3.1. GNU General Public License (GPL)

La GPL uscì nel 1989 ed esprime in sé stessa i principi del Free Software. La caratteristica principale della GPL è la sua “viralità”, vale a dire che ogni modifica a codice GPL e tutto il codice sviluppato per funzionare in associazione con software GPL, deve essere a sua volta coperto da licenza GPL. La licenza GPL è tale per cui si può affermare che il software regolato GPL è Open Source, inoltre, la sua viralità garantisce che le libertà dell’utente vengano mantenute nel tempo, quindi il software GPL fa anche parte della categoria del software copyleft.

Recentemente è stata pubblicata la terza versione della licenza GPL allo scopo di modificare alcuni aspetti della versione precedente, per meglio gestire in particolar modo l’uso di DRM (*Digital Right Management*) e del *tivoization*².

3.2. LGPL, Library General Public License

La viralità della licenza GPL pose un problema alla realizzazione di librerie di funzioni; infatti, una libreria protetta da GPL non sarebbe stata utilizzabile da sviluppatori di software proprietario e quindi la libreria avrebbe perso gran parte del suo potenziale; per questo motivo, nel 1991 uscì una seconda licenza, chiamata LGPL, *Library General Public License*. Questa è una licenza apposta per le librerie, che si presenta come una versione della GPL alleggerita di alcune restrizioni (per questo recentemente rinominata “Lesser GPL”). In tal modo, il software principale, che utilizza librerie LGPL, può essere rilasciato sotto qualsiasi licenza (anche proprietaria), ma le modifiche al codice LGPL sono comunque soggette a LGPL; tale particolarità pone la LGPL ai margini di ammissibilità fra le licenze copyleft.

3.3. BSD (*Berkley Standard Distribution*)

La licenza BSD (le cui prime versioni risalgono agli anni ‘80), è una licenza molto semplice, la cui applicazione si risolve nell’inserimento di una breve nota standard da inserire nei file che si intendono tutelare con la licenza. La nota deve riportare il nome di chi detiene il copyright, l’organizzazione cui appartiene e l’anno di realizzazione³. La licenza specifica che sono permesse la ridistribuzione e l’utilizzo del software tutelato in forma sorgente o binaria, con o senza modifiche, ma solo se vengono rispettate tre condizioni:

- le ridistribuzioni del codice sorgente devono mantenere la nota sul copyright;
- le ridistribuzioni in forma binaria devono riprodurre la nota sul copyright, l’elenco delle condizioni e la successiva avvertenza nella documentazione e nell’altro materiale fornito con la distribuzione;
- il nome dell’autore non potrà essere utilizzato per sostenere o promuovere prodotti derivati dal software licenziato, senza un apposito permesso scritto dell’autore.

La caratteristica principale di tale licenza è la possibilità di “chiudere” il software sotto qualsiasi altra licenza, anche proprietaria; lo stesso vale anche per le modifiche apportate al software. Questo aspetto la pone al di fuori del paradigma di “copyleft”, poiché non è garantita la trasmissione all’infinito dei diritti da essa concessi, i quali si interrompono nel momento in cui il codice (originale o modificato) viene chiuso sotto altra licenza. Simile alla BSD nella struttura e nello spirito generale è la MIT License, che prende il nome dal noto centro di ricerca del Massachusetts.

3.4. Mozilla Public License (MPL)

Sviluppata da Netscape nel 1998, la MPL è un ibrido tra le licenze GPL e BSD. Al suo interno distingue tra “Covered Code”, cioè il codice soggetto alla licenza, che è simile a GPL e “Larger Work” cioè il codice MPL che può essere combinato con altro codice (anche proprietario). In tal modo, la MPL si comporta come la GPL sul Covered Code e come la BSD sul Larger Work. All’interno di un progetto, la decisione sulla distinzio-

² Il termine indica la pratica di creazione di un sistema che incorpora software sotto i termini di una licenza copyleft, ma usa particolare hardware per evitare agli utenti la possibilità di eseguire versioni modificate del software su quell’hardware.

³ La BSD originale, prevedeva una clausola chiamata “pubblicitaria” che obbligava l’utente a “dare esplicito riconoscimento all’Università [Berkley] nelle eventuali inserzioni pubblicitarie previste per i programmi derivati”. Siccome la GPL proibisce ogni restrizione addizionale circa la ridistribuzione del software, la licenza BSD originale non era compatibile con la GPL. Nel 1999, quindi, tale clausola venne eliminata rendendo così la BSD, GPL-compatibile.

ne tra i due tipi di codice è lasciata agli sviluppatori. In essa, più che in altre licenze anteriori, sono presenti alcune rilevanti clausole mirate a tutelare gli sviluppatori del progetto Mozilla dalla malafede di coloro che avrebbero potuto approfittare della elasticità della licenza inserendo tacitamente parti di codice coperte da brevetto, per poi rivendicarne per vie legali il pagamento. La licenza Mozilla, tuttora usata per i browser Netscape, è incompatibile con la GPL⁴.

3.5. Q Public License

La QPL creata da Trolltech, cattura il significato generale della GPL, ma è incompatibile con essa

Licenze	Può essere miscelato con software commerciale?	Le modifiche possono essere mantenute private e non restituite all'autore originale?	Può essere ri-licenziato da chiunque?
GPL	no	no	no
LGPL	sì	no	no
New BSD	sì	sì	no
MPL	sì	sì	no
QPL	sì	sì	no
Public Domain	sì	sì	sì

TABELLA 2

Nella tabella sono riportate alcune differenze tra le licenze Open Source trattate nel presente articolo

Licenze	È una licenza di software libero?	È compatibile con la GPL?	È nello spirito del copyleft?
GPL	sì	sì	Sì
LGPL	sì	sì	non del tutto
New BSD	sì	sì	No
MPL	sì	sì	No
QPL	sì	no	non del tutto

TABELLA 3

Analisi delle licenze trattate in base ai parametri di software libero, compatibilità con la GPL e copyleft

nel senso che non è possibile unire assieme codice protetto da GPL e da QPL. Questa distingue tra il software originale e le modifiche apportate e come queste sono distribuite. Se le modifiche sono incorporate nel codice, allora la QPL si estende anche ad esse (come per la GPL). Se invece le modifiche sono distribuite sotto forma di patches (codice esterno), similmente a quanto accade nella BDS, queste possono essere soggette a qualsiasi licenza, anche proprietaria⁵. Nella tabella 2 e 3 sono riportate a confronto le licenze trattate rispetto ad alcuni parametri fondamentali.

Alla luce di quanto visto, si può procedere ad una classificazione delle licenze Open Source nel seguente modo:

□ **propagative** (nello spirito del copyleft), sono quelle licenze ispirate alla GPL, che richiedono che tutti gli sviluppi del codice siano integrati solo con codice Open Source e siano pubblicati sotto una licenza GPL-compatibile;

□ **persistenti ma non propagative** (non propriamente copyleft), sono quelle licenze che consentono un linking dinamico con codice proprietario, ma ogni modifica al codice originale deve essere soggetto alla stessa licenza (esempio, LGPL, MPL, QPL). Gli sviluppatori che modificano il codice originale o estendono librerie esistenti devono rilasciare il codice sotto la stessa licenza (persistenza), ma il codice esterno collegato al codice originale, può essere distribuito sotto altra licenza (non propagazione);

□ **permissive non propagative né persistenti** (non copyleft), sono quelle licenze che ammettono una combinazione di codice OS con codice proprietario (esempio, BSD, MIT) e consentono di porre il codice così ottenuto sotto altra licenza (anche proprietaria) mantenendo solo una nota riportante i nomi degli autori del codice OS.

Un aspetto degno di nota è quello delle garanzie sul software. Il software libero in genere, è ceduto senza alcuna garanzia. La ragione dell'assenza di garanzia è conseguenza del fatto che l'autore originario rischia di essere chiamato in cause legali anche per modifiche non apportate da lui. Garantire questo tipo di software diventerebbe, quindi, un ostacolo alla sua diffusione ed esporrebbe l'autore a rischi legali e finanziari non pre-

⁴ Alcune licenze derivate dalla MPL sono Apple PL (Mac OS X e Darwin (Covered Code) + Quartz e altre porzioni proprietarie), Nokia OSL, Sun PL.

⁵ Per questo motivo tale licenza non può essere considerata pienamente nello spirito del copyleft. Un esempio di software QPL sono le librerie Qt (che però sono state rilasciate anche sotto GPL).

vedibili a priori. È concessa garanzia, ma limitata - come avviene per il software proprietario - solo in caso di diffusione commerciale. In questo caso è l'azienda che realizza la "distribuzione" che esegue le verifiche ed i test sul software utilizzato e che si assume la responsabilità di fornire una garanzia "limitata". La limitazione scelta nella maggior parte dei casi fa sì che si debba solo risarcire il prezzo del pacchetto software, oppure obbliga alla sostituzione del supporto danneggiato. La OSD non fa riferimento a questo tipo di garanzie, pertanto in una licenza Open Source non vi è obbligo né divieto di inserirle.

4. UN CONTRO ESEMPIO: LA LICENZA MS SHARED SOURCE

Microsoft ha proposto otto licenze di tipo *Shared Source* [7] le quali permettono la lettura del codice sorgente, ma vietano di ridistribuire il codice o di condividere il sorgente con terze parti. Le licenze per enti commerciali o governativi sono di tipo "read only", ovvero non consentono la modifica neanche per uso interno (ma è possibile proporre le patch a MS).

Il ricevente è tenuto a trattare il sorgente come dati proprietari confidenziali di Microsoft. Pertanto uno sviluppatore che entra in contatto con il sorgente Microsoft, se successivamente sviluppa programmi competitivi, potrebbe essere citato per violazione di dati confidenziali.

5. CONCLUSIONI

Open Source è un modello di sviluppo innovativo, mirante alla qualità e l'affidabilità del software attraverso la revisione indipendente tra pari e l'evoluzione rapida del codice sorgente. La rilevanza economica del software Open Source è cresciuta rapidamente, soprattutto grazie al fatto che questi sistemi liberi sono disponibili a costi molto bassi, se non nulli, ma comunque ben inferiori a quelli di analoghi sistemi proprietari. Tuttavia, a causa delle loro caratteristiche, il loro uso richiede una buona cultura di base nel campo del software.

Grazie alle spinte all'utilizzo di software Open Source negli Enti Pubblici, il *Centro Nazionale per l'Informatica nella Pubblica Amministrazione* (CNIPA) [8] ha costituito l'Osservatorio Open Source [9] e molti sono i provvedimenti legislativi presi a livello sia nazionale che regionale che mi-

rano alla promozione e l'uso del software Open Source anche nella Pubblica Amministrazione.

Per essere Open Source, il software deve essere distribuito sotto una licenza che garantisca i diritti fondamentali espressi nelle 10 condizioni della *Open Source Definition* (OSD). È importante ricordare come questa definizione risulta abbastanza generale per cui non esiste un'unica licenza Open Source, ma è possibile scegliere (o proporre) la licenza Open Source più adeguata agli obiettivi e alle situazioni. La *Open Source Initiative* (OSI) mantiene sul suo sito ufficiale una lista delle licenze "Open Source compatibili".

Bibliografia

- [1] St. Laurent A. M.: *Understanding Open Source & Free Software Licensing*. O'Reilly, Agosto 2004.
- [2] Aliprandi S.: *Copyleft & opencontent l'altra faccia del copyright*. PrimaOra, marzo 2005.
- [3] Sito del Copyleft italiano: www.copyleft-italia.it
- [4] Sito Ufficiale della Free Software Foundation (FSF): www.fsf.org
- [5] Sito Ufficiale della Open Source Initiative (OSI): www.opensource.org
- [6] Baggi M., Piva A.: *Categorie di licenze per software libero*. Rivista di Informatica Giuridica, disponibile all'indirizzo: <http://217.141.103.211/alsiud/infogiur.htm>
- [7] MS Shared Source: <http://www.microsoft.com/resources/sharedsource/default.mspx>
- [8] Sito Ufficiale del CNIPA: <http://www.cnipa.gov.it/>
- [9] Osservatorio Open Source del CNIPA: <http://www.ossipa.cnipa.it/>

ANTONIO PIVA, laureato in Scienze dell'Informazione, *Vice Presidente dell'ALSI* (Associazione Nazionale Laureati in Scienze dell'Informazione ed Informatica) e Presidente della commissione di informatica giuridica. Docente a contratto di *diritto dell'ICT e qualità* all'Università di Udine. Consulente sistemi informatici e Governo Elettronico nella PA locale, valutatore di sistemi di qualità ISO9000 ed ispettore AICA.
E-mail: antonio@piva.mobi

DAVID D'AGOSTINI avvocato, master in informatica giuridica e diritto delle nuove tecnologie, collabora all'attività di ricerca scientifica dell'Università degli studi di Udine e ha fondato l'associazione "*Centro Innovazione & Diritto*". È componente della Commissione Informatica dei Consigli dell'Ordine del Triveneto, responsabile dell'area "*Diritto & informatica*" della rivista "*Il foro friulano*", membro dell'organo di Audit Interno di Autovie Venete SpA.

E-mail: studio@avvocatodagostini.it