



UNIVERSITÀ TELEMATICA DELLE CAMERE DI COMMERCIO ITALIANE

UNIVERSITAS MERCATORUM

FACOLTÀ DI ECONOMIA

CORSO DI LAUREA IN SCIENZE GIURIDICHE

TESI DI LAUREA

IN

DIRITTO PRIVATO

**OPEN SOURCE E FREE SOFTWARE: DIFFERENZE OPERATIVE E DUBBI
INTERPRETATIVI**

Relatrice

Chiar.ma Prof.ssa Silvia PICCININI

Laureando

Ugo LOPEZ

ANNO ACCADEMICO 2020/21

Ringraziamenti

Quando è iniziato questo brutto momento, ho avuto il terrore di perdere il lavoro e, in conseguenza di questo, tutto il resto: per questo ho iniziato a lavorare molto più del solito e a cercare strade che mi avrebbero consentito di lavorare anche da casa con risultati soddisfacenti.

Mentre mi lambiccavo il cervello, la mia Compagna mi ha detto: oltre a lavorare, perché non inizi a studiare giurisprudenza, dato che ti è sempre piaciuta? Lo ha fatto per distrarmi e per costringermi a fare qualcosa per me stesso, in un momento così difficile. Mai consiglio fu più giusto, per cui ti ringrazio, **Marilena**.

Devo poi ringraziare i miei figli, **Giovanni** e **Francesca**, che hanno sopportato senza lamentarsi il tempo che ho dovuto sottrarre per dedicarmi agli studi e terminare in regola.

Questo sforzo, però, è dedicato al mio cane **Ciop**, compagno di molte notti di studi e passeggiate antelucane: non ha mai chiesto niente ma ha dato molto. Dovunque tu sia, ti voglio bene.

In ultimo, un ringraziamento al Linux Group di Bari, alla Italian Linux Society, alla Commissione dell'Informazione degli Ingegneri della Provincia di Bari e a **Roberto Guido** per aver accettato il mio talk sull'argomento nel Linux Day 2020, nonché all'avvocato **Filippo Bianchini**, specialista della materia, per i preziosi suggerimenti: grazie alla passione che ne è nata, si è originato questo lavoro di tesi.

Sommario

CAPITOLO 1 – Tutele sottese al software	4
1.1 - La tutela del diritto d'autore nell'ordinamento italiano	5
1.1.1 – Cenni storici	5
1.1.2 – Tutele sottese alle opere d'ingegno: diritti morali, patrimoniali e connessi	7
1.1.3 – La tutela dei programmi per elaboratore: “copyright” o brevetto?	14
1.2 - Le licenze software come opere d'ingegno: panoramica delle principali tipologie	17
1.2.1 – Introduzione alle licenze d'uso per i software	17
1.2.2 - Panoramica delle principali tipologie di licenze software	20
CAPITOLO 2 – Le licenze FLOSS e di Pubblico Dominio	27
2.1 – La Free Software Foundation	27
2.1.1 – Cenni storici	27
2.1.2 – Il software libero	31
2.1.3 – Il copyleft	33
2.1.4 – Le licenze free software	36
2.2 – La Open Source Initiative	39
2.2.1 – Cenni storici	39
2.2.2 – La definizione di Open Source	41
2.3 – Casi di studio	46
2.3.1 – Il caso RHEL	46
2.3.2 – Il caso Microsoft Visual Studio Code	47
2.3.3 – Il caso Android con licenza GNU GPL 2	49
CAPITOLO 3 – Giurisprudenza	50
3.1 – Giurisprudenza Open Source e Free Software	50
3.1.1 – Jacobsen c.\ Katzer – il “trenino della discordia” (2004-2010)	50
3.1.2 – Novell c.\ CTPN (2011)	51
3.1.3 SCO c.\ IBM et altera (2003-2010)	52
CAPITOLO 4 – CONCLUSIONI	54
SITOGRAFIA	56

CAPITOLO 1 – Tutele sottese al software

L'idea alla base di questa tesi è nata quasi per caso, poco dopo l'inizio del lockdown. Avevo ricevuto il compito, assieme ad altre persone, di scrivere una breve relazione che indicasse alcune soluzioni utili ed efficaci per lo smart working di alcuni dipendenti di una Pubblica Amministrazione. In una delle relazioni pervenute ho letto questo stralcio di frase:

“software open source (proprietario e non)”

Questa frase mi ha immediatamente fatto sorridere: come può un software open source essere proprietario?

Per fugare ogni dubbio, ho riportato questo stralcio ad alcuni informatici forensi molto esperti (alcuni project manager di distribuzioni Linux) e ho avuto la prima sorpresa: non tutti ritenevano questa frase infondata, ognuno aveva un'opinione diversa sulla questione.

Allora ho deciso di approfondire e ho sottoposto la questione agli amici del LUG¹ a cui appartengo, quello di Bari: anche qui l'argomento è stato oggetto di un acceso dibattito, con opinioni molto distanti tra loro.

Ed è a questo punto che ho deciso di proporre un intervento su tale argomento al Linux Day 2020² e di chiedere alla prof.ssa PICCININI di poterlo utilizzare come base per la mia tesi di laurea.

¹ Linux User Group

² <https://www.linuxday.it/2020/programma/talk.php?slug=open-source-e-free-software-quali-differenze>

Se qualcuno pensa che questa tesi parli di informatica, si sbaglia: questa tesi parla di **libertà**, di pensiero, di arte. E, proprio per questo, parla di diritto.

1.1 - La tutela del diritto d'autore nell'ordinamento italiano

1.1.1 - Cenni storici

La proprietà intellettuale, artistica e letteraria in Italia viene tutelata attraverso il diritto d'autore, le cui origini vanno ricercate nella **L. 633/41**, "*Protezione del diritto d'autore e di altri diritti connessi al suo esercizio*", oggi nota col nome di **Legge sul Diritto d'Autore** (di seguito indicata come L. d. A.). Detto impianto normativo, nel corso degli anni, è stato ripetutamente modificato ed integrato da varie disposizioni legislative tra cui, in ultimo, la **L. 37/19³** che, a seguito di procedura d'infrazione⁴ concernente il mancato recepimento di una direttiva europea⁵, integra l'art. 71-bis del Testo Unico introducendo delle deroghe legislative per categorie speciali di soggetti svantaggiati⁶ anche in merito all'accessibilità delle opere e dei soggetti autorizzati in deroga, la **L. 128/04** che converte il **D.L. 72/04** (che introduce "*misure di contrasto alla diffusione telematica abusiva di opere d'ingegno*"), la **L. 43/05** che converte, con modificazioni, il **D.L. 7/05**, nonché in sanatoria degli effetti dell'**art. 4, I comma, D.L. 280/04** (contenente "*disposizioni urgenti per l'università e la ricerca, per i beni e le attività culturali, per il completamento di grandi opere*

³ Capo V "Disposizioni in materia di diritto d'autore", art. 15,
<https://www.gazzettaufficiale.it/eli/id/2019/05/11/19G00044/sg>

⁴ Procedura d'infrazione n.2018/0354, <http://www.senato.it/leg/18/BGT/Schede/docnonleg/37472.htm>

⁵ Direttiva UE 2017/1564, <https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32017L1564&from=IT>

⁶ Non vedenti e ipovedenti, con disabilità percettiva di lettura, con disabilità fisica che impedisce di tenere un libro

strategiche, per la mobilità dei pubblici dipendenti, nonché per semplificare gli adempimenti relativi a imposte di bollo e tasse di concessione”), il D.Lgs. 118/06 (in merito alla “attuazione della direttiva 2001/84/CE, relativa al diritto dell’autore di un’opera d’arte sulle successive vendite dell’originale”), il D.Lgs. 140/06 (in merito alla “attuazione della direttiva 2004/48/CE sul rispetto dei diritti di proprietà intellettuale”), dal D.D.L. S1861, approvato dal Senato il 21.12.2007 (contenente “disposizioni concernenti la Società italiana degli autori e degli editori”).

A detto impianto normativo, così creato e modificato, si aggiungono gli **artt. 2575-2583 c.c.**, contenuti nel libro V (“*del lavoro*”), titolo IX (“*diritti sulle opere d’ingegno*”), capo I (“*del diritto di autore sulle opere dell’ingegno letterarie e artistiche*”). A tal proposito, si noti come la stessa collocazione delle norme all’interno del codice di merito ben distingue il diritto d’autore (libro V, titolo IX, capo I) dal brevetto (libro V, titolo IX, capi II e III), simili per alcuni aspetti (da cui la medesima dislocazione nel libro V, titolo IX) ma ben distinti per ambito di applicazione: il primo relativo ad opere dell’ingegno, il secondo a invenzioni industriali, modelli di utilità e di registrazione per disegni e modelli.

La L. 633/41, in realtà, è l’epilogo di una serie di norme di tutela, volute durante il fascismo, principiate con la L. 562/26, in conversione del Regio Decreto Legge n. 1950 del 7 novembre 1925 (“*disposizioni sul diritto d’autore*”) che andavano sostanzialmente ad uniformare tale tutela a quella minima prevista dalla “**convenzione di Berna per la protezione delle opere letterarie e artistiche**”

del 1886⁷, successivamente modificata più volte fino al 1971 a Parigi, a cui l'Italia aderì formalmente il 20 giugno 1978 con ratifica attraverso la **L. 399/78** (*“ratifica ed esecuzione della convenzione di Berna per la protezione delle opere letterarie ed artistiche, firmata il 9 settembre 1886, completata a Parigi il 4 maggio 1896, riveduta a Berlino il 13 novembre 1908, completata a Berna il 20 marzo 1914, riveduta a Roma il 2 giugno 1928, a Bruxelles il 26 giugno 1948, a Stoccolma il 14 luglio 1967 e a Parigi il 24 luglio 1971, con allegato”*).

Più recentemente, tra le opere d'ingegno tutelate sono rientrati anche i programmi per elaboratori elettronici (c.d. *software*), a cominciare dal **D.Lgs. 518/92** (*“attuazione della direttiva 91/250/CEE relativa alla tutela giuridica dei programmi per elaboratore”*), successivamente modificato dal **D. Lgs. 205/96** (*“modificazioni al decreto legislativo 29 dicembre 1992, n. 518, in materia di tutela giuridica dei programmi per elaboratore”*), in recepimento di preciso indirizzo europeo, espresso attraverso la **direttiva 91/250/CEE**.

Più recentemente, il nostro ordinamento ha introdotto nuove norme a contrasto della pirateria informatica e contraffazione del software, attraverso la **L. 248/00** (*“nuove norme di tutela del diritto d'autore”*).

1.1.2 - Tutele sottese alle opere d'ingegno: diritti morali, patrimoniali e connessi

Il nostro ordinamento giudiziario orienta la tutela delle opere dell'ingegno alla loro paternità, producendo pertanto due precisi ambiti di applicazione, ovvero

⁷ <http://www.interlex.it/testi/convberna.htm>

la sfera dei diritti morali e la sfera dei diritti patrimoniali, legati allo sfruttamento economico.

I **diritti morali** sono tutelati all'interno del Titolo I, "*Disposizioni sul diritto d'autore*", Capo III, "*Contenuto e durata del diritto d'autore*", Sezione II, "*Protezione dei diritti sull'opera a difesa della personalità dell'autore (Diritto morale dell'autore)*", del vigente testo della L. d. A., artt. 20 e ss. In particolare, l'autore ha diritto a:

- ✓ Rivendicare la paternità dell'opera, opporsi a qualsiasi deformazione (**diritto all'integrità dell'opera**, *infra*), mutilazione od altra modificazione, ed a ogni atto a danno dall'opera stessa pregiudizievole del proprio onore o reputazione⁸;
- ✓ Esercitare i diritti di cui sopra indipendentemente dai diritti esclusivi di utilizzazione economica dell'opera ed anche dopo la cessione dei diritti stessi⁹;
- ✓ Rivelarsi e far conoscere in giudizio la propria qualità di autore anche in caso di opera anonima o pseudonima e ottenere che gli aventi causa ne indichino il nome nelle pubblicazioni, riproduzioni, trascrizioni, esecuzioni, rappresentazioni, recitazioni e diffusioni o in qualsiasi altra forma di manifestazione o annuncio al pubblico, nonostante qualunque precedente patto contrario¹⁰;

⁸ Art. 20 L. d. A., I comma

⁹ Art. 20 L. d. A., I comma

¹⁰ Art. 21 L. d. A.

- ✓ Godere dell'inalienabilità dei menzionati diritti, fatti salvi i casi in cui l'autore abbia conosciuto ed accettato le modificazioni della propria opera, nel cui caso non è più ammesso ad agire per impedirne l'esecuzione o per chiederne la soppressione¹¹;

Inoltre:

- ✓ Dopo la morte dell'autore, i diritti di cui all'art. 20 L. d. A. possono essere fatti valere *sine die* dal coniuge e dai figli o, in assenza, dai genitori e dagli altri ascendenti e discendenti diretti o, in ulteriore assenza, dai fratelli, sorelle e loro discendenti. Qualora l'azione rivesta finalità pubbliche, può essere esercitata dal Ministro della Cultura sentita l'associazione sindacale competente¹²;
- ✓ Gli eredi o legatari dell'autore godono del diritto di pubblicare opere inedite, salvo esplicite diffide o differenti volontà dell'autore stesso¹³;
- ✓ Le opere inedite non possono essere pubblicate prima di una eventuale scadenza fissata dallo stesso¹⁴;
- ✓ In caso di pluralità dissenziente di eredi e/o legatari, la decisione viene presa dall'autorità giudiziaria, sentito il Pubblico Ministero, nel rispetto della volontà del *de cuius*¹⁵;

¹¹ Art. 22 L. d. A.

¹² Art. 23 L. d. A.

¹³ Art. 24 L. d. A., I comma

¹⁴ Art. 24 L. d. A., II comma

¹⁵ Art. 24 L. d. A., III comma

- ✓ I diritti di utilizzazione dell'opera *post mortem* sono regolati dal Titolo III, "Disposizioni comuni", Capo II, "Trasmissione dei diritti di utilizzazione", Sezione II, "Trasmissione a causa di morte", artt. 115 e ss¹⁶.

Rientrano poi tra i diritti morali, oltre a quelli descritti (che la letteratura giuridica raggruppa spesso sotto il nome di **diritto alla paternità dell'opera**):

- ✓ Il **diritto di inedito**, secondo il quale neanche gli eredi possono violare la volontà dell'autore¹⁷ in merito alla pubblicazione dell'opera e che trova l'unica limitazione nell'interesse dello Stato¹⁸;
- ✓ Il **diritto di "pentimento"**, ovvero la possibilità di ritirare l'opera dal commercio per gravi ragioni (diritto personale e non trasmissibile)¹⁹, anche attraverso l'Autorità giudiziaria²⁰;
- ✓ Il **diritto all'integrità dell'opera** (in duplice veste di diritto morale e patrimoniale), già parzialmente descritto *supra*, con alcune eccezioni nell'ambito architettonico²¹ ed altre qualora l'autore fosse a conoscenza ed avesse accettato le modificazioni, come *supra* riportato, ai sensi e per l'effetto dell'art. 22, II comma, L. d. A.

Ai diritti morali così diffusamente descritti, si affiancano poi i **diritti patrimoniali**, riassumibili principalmente secondo il seguente schema:

¹⁶ Art. 24 L. d. A., IV comma

¹⁷ Già citato art. 24 L. d. A.

¹⁸ Art. 112 L. d. A.

¹⁹ Art. 142 L. d. A.

²⁰ Art. 143 L. d. A.

²¹ Art. 20 L. d. A., II comma

- ✓ Il **diritto di pubblicazione**, ovvero il diritto esclusivo dell'autore a pubblicare la propria opera²² e il diritto esclusivo di utilizzo economico²³;
- ✓ Il **diritto di riproduzione**, il quale ha per oggetto la moltiplicazione in copie dell'opera²⁴, con specificità e limitazioni²⁵;
- ✓ Il **diritto di trascrizione**, ovvero la protezione delle opere anche qualora in forma orale²⁶, riconoscendo all'autore il diritto esclusivo alla trascrizione²⁷;
- ✓ Il **diritto di esecuzione, rappresentazione o recitazione in pubblico**, relativo ad opere musicali, drammatiche, cinematografiche o che, comunque, possa costituire oggetto di pubblico spettacolo²⁸;
- ✓ Il **diritto di comunicazione al pubblico**, ovvero il diritto esclusivo di comunicazione al pubblico attraverso mezzi di comunicazione a distanza²⁹;

²² Art. 12 L. d. A., I comma

²³ Art. 12 L. d. A., II comma

²⁴ Art. 13 L. d. A.

²⁵ Titolo I "*Disposizioni sul diritto d'autore*", Capo V "*Eccezioni e limitazioni*", Sezione I "*Reprografia ed altre eccezioni e limitazioni*" L. d. A. – artt. 65 e ss.

²⁶ Art. 2 L.d.A., I comma, punto 1)

²⁷ Art. 14 L. d. A.

²⁸ Art. 15 L. d. A.

²⁹ Art. 16 L. d. A.

- ✓ Il **diritto di modificazione e elaborazione dell'opera**, ovvero il diritto esclusivo di tradurre, elaborare, pubblicare in raccolta e modificare l'opera³⁰, i cui esiti sono analogamente protetti³¹;
- ✓ Il **diritto di noleggio e diritto di prestito**, ovvero il diritto esclusivo di cedere in uso gli originali, le copie o i supporti di opere per un periodo limitato di tempo, con o senza ottenimento di benefici patrimoniali, cedibile a terzi anche dietro equa remunerazione, con alcune eccezioni tipizzate³²;
- ✓ Il **diritto di seguito**, ovvero il diritto ad un compenso sul prezzo di ogni vendita successiva alla prima cessione per gli autori di opere d'arte e di manoscritti³³.

Il diritto esclusivo dell'autore a pubblicare l'opera e utilizzarla economicamente è altresì disciplinato dall'**art. 2577 cc**, mentre i diritti esclusivi sopra menzionati sono fra loro indipendenti e sono riferibili all'intera opera nel suo insieme ed a ciascuna delle sue parti³⁴. I diritti patrimoniali, a differenza dei diritti morali, **sono rinunciabili e trasferibili**³⁵ e hanno una durata limitata nel tempo, ovvero sino a settanta anni dopo la morte dell'autore³⁶.

³⁰ Art. 18 L. d. A.

³¹ Art. 4 L. d. A.

³² Art. 18-bis L. d. A.

³³ Art. 144 L. d. A.

³⁴ Art. 19 L. d. A.

³⁵ Art. 107 L. d. A. e art. 2581 cc

³⁶ Art. 25 L. d. A.

Esiste, in ultimo, una terza sfera, quella relativa ai **diritti connessi**, ovvero quelli relativi agli artisti interpreti ed artisti esecutori (attori, cantanti, musicisti, ballerini e le altre persone che rappresentano, cantano, recitano, declamano o eseguono in qualunque modo opere d'ingegno)³⁷, i quali consentono a dette categoria di:

- ✓ Autorizzare la fissazione delle loro prestazioni artistiche³⁸;
- ✓ Autorizzare la riproduzione diretta o indiretta, temporanea o permanente, in qualunque modo o forma, in tutto o in parte, della fissazione delle loro prestazioni artistiche³⁹
- ✓ Autorizzare la comunicazione al pubblico delle proprie prestazioni artistiche, con varie ulteriori specificazioni⁴⁰;
- ✓ Autorizzare la messa a disposizione del pubblico *on demand* della fissazione delle proprie prestazioni artistiche e relative riproduzioni⁴¹;
- ✓ Autorizzare la distribuzione della fissazione delle proprie prestazioni artistiche⁴²;

³⁷ Titolo II “Disposizioni sui diritti connessi all’esercizio del diritto di autore”, Capo III “Diritti degli artisti interpreti e degli artisti esecutori”, Artt. 80 e ss L. d. A.

³⁸ Art. 80 L. d. A., Il comma, lett. a)

³⁹ Art. 80 L. d. A., Il comma, lett. b)

⁴⁰ Art. 80 L. d. A., Il comma, lett. c)

⁴¹ Art. 80 L. d. A., Il comma, lett. d)

⁴² Art. 80 L. d. A., Il comma, lett. e)

- ✓ Autorizzare il noleggio o il prestito delle fissazioni delle loro prestazioni artistiche e delle relative riproduzioni⁴³.

1.1.3 - La tutela dei programmi per elaboratore: “copyright” o brevetto?

Come già accennato, all'interno di questo ampissimo corpo normativo, viene tutelato anche il software. L'argomento si rivela essere abbastanza complesso sia per la giovane età dei meccanismi di tutela, inizialmente riservati solo all'hardware, sia per la dibattuta natura del software, da una parte visto come bene immateriale (e, quindi, protetto attraverso meccanismi collegati alla proprietà intellettuale), dall'altra visto come ritrovato tecnico (e, quindi, protetto attraverso meccanismi propri del brevetto).

All'esito di questo articolato dibattito, iniziato negli anni Settanta, la comunità si è orientata verso la prima delle due forme di tutela, probabilmente perché la protezione attraverso brevetto sarebbe stata eccessivamente forte e avrebbe potuto ostacolare la concorrenza. A parte le tutele specifiche, il software gode anche delle tutele a favore delle opere letterarie⁴⁴, così come previsto dalla **L. 399/78** in ratifica ed esecuzione della **Convenzione di Berna**, già citate in precedenza.

Oggetto della tutela sono *“i programmi per elaboratore, in qualsiasi forma espressi purché originali quale risultato di creazione intellettuale dell'autore. Restano esclusi dalla tutela accordata dalla presente legge le idee e i principi che*

⁴³ Art. 80 L. d. A., Il comma, lett. f)

⁴⁴ Art. 1 L. d. A., Il comma

*stanno alla base di qualsiasi elemento di un programma, compresi quelli alla base delle sue interfacce. Il termine programma comprende anche il materiale preparatorio per la progettazione del programma stesso*⁴⁵.

Nel principale *corpus juris* a protezione del diritto d'autore, poi, una sezione specifica è dedicata alla tutela dei programmi per elaboratore⁴⁶. In esso sono disciplinati i diritti specifici⁴⁷ per *“effettuare o autorizzare la riproduzione, permanente o temporanea, totale o parziale, del programma per elaboratore con qualsiasi mezzo o in qualsiasi forma”*⁴⁸, *“la traduzione, l'adattamento, la trasformazione e ogni altra modificazione”*⁴⁹, *“qualsiasi forma di distribuzione al pubblico, compresa la locazione, del programma per elaboratore originale o di copie dello stesso”*⁵⁰, nonché talune limitazioni⁵¹ relativamente alla non necessità di autorizzazione per quanto previsto alle lettere a) e b) del I comma dell'art. 64-bis⁵², al diritto alla copia di riserva⁵³, alle modalità consentite di reverse engineering⁵⁴, alla interoperabilità con altri software e banche dati⁵⁵ e, in ultimo, alle regole di interpretazione degli articoli specifici⁵⁶.

⁴⁵ Art. 2 L. d. A., I comma, num. 8)

⁴⁶ Titolo I *“Disposizioni sul diritto d'autore”*, Capo IV *“Norme particolari ai diritti di utilizzazione economica di talune categorie di opere”*, Sezione VI *“Programmi per elaboratore”*, artt. 64-bis e ss L. d. A.

⁴⁷ Art. 64-bis L. d. A.

⁴⁸ Art. 64-bis L. d. A., I comma, lett. a)

⁴⁹ Art. 64-bis L. d. A., I comma, lett. b)

⁵⁰ Art. 64-bis L. d. A., I comma, lett. c)

⁵¹ Art. 64-ter e 64-quarter L. d. A.

⁵² Art. 64-ter L. d. A., I comma

⁵³ Art. 64-ter L. d. A., II comma

⁵⁴ Art. 64-ter L. d. A., III comma. Una definizione di reverse engineering è disponibile qui:

https://it.wikipedia.org/wiki/Ingegneria_inversa

⁵⁵ Art. 64-quarter L. d. A., commi 1, 2 e 3

⁵⁶ Art. 64-quarter L. d. A., IV comma

Che la tutela brevettuale non sia applicabile al software viene anche chiarito dalla **Convenzione sulla concessione di Brevetti Europei (CBE)**, nel punto in cui, nella parte seconda “*Diritto dei brevetti*”, capitolo I “*Brevettabilità*”, art. 52 “*Invenzioni brevettabili*”, II comma, lett. c), si chiarisce che “*Non sono considerate invenzioni [omissis] i programmi informatici*”, qualora però “*nella misura in cui la domanda di brevetto europeo o il brevetto europeo concerna uno solo di tali oggetti o attività, considerati come tali*”⁵⁷, nonché nella c.d. **Legge invenzioni**⁵⁸ nel punto in cui viene precisato che “*Non sono considerate come invenzioni [omissis] i programmi per elaboratori*”⁵⁹, sempre e solo “*in quanto tali*”⁶⁰.

Ciononostante, la “scappatoia” fornita dalla limitazione alla brevettabilità delle opere “in quanto tali” ed un mutato orientamento giurisprudenziale, ha portato l’Ufficio Europeo Brevetti (EPO) e vari uffici nazionali a concedere migliaia di brevetti per invenzioni attuate a mezzo software⁶¹, proprio invocando l’escamotage citato. In particolare, nella decisione in nota, afferma l’EPO che se un software presenta “*effetti tecnici ulteriori o che vadano al di là della normale interazione software-hardware*” non può, *ipso facto*, essere escluso dalla brevettabilità.

⁵⁷ Art. 52 CBE, III comma

⁵⁸ R.D. 1127 del 29 giugno 1939 e s.m.i.

⁵⁹ Art. 12 R. D. 1127/39, II comma, lett. b, così come sostituito dall’art. 7 D. P. R. 338 del 22 giugno 1979

⁶⁰ Art. 12 R. D. 1127/39, III comma, così come sostituito dall’art. 7 D. P. R. 338 del 22 giugno 1979

⁶¹ Ex plurimis, decisione T1173/97, <https://www.epo.org/law-practice/case-law-appeals/recent/t971173ex1.html>

A tale conclusione si perverrebbe anche dall'analisi dell'art. 27 del TRIPs⁶², accordo internazionale nato con l'intento di creare un "testo unico internazionale" per la tutela commerciale della proprietà intellettuale, a mente del quale "*possono costituire oggetto di brevetto le invenzioni [omissis] in tutti i campi della tecnologia, che siano nuove, implicino un'attività inventiva e siano atte ad avere un'applicazione industriale*"⁶³.

In conclusione, quindi, qualora un software presenti un consistente contributo allo stato dell'arte in un settore tecnico, lo stesso può essere protetto a mezzo di brevetto e, qualora questo non accada, il programma per elaboratore gode comunque della tutela fornita dalle norme sul diritto d'autore.

1.2 - Le licenze software come opere d'ingegno: panoramica delle principali tipologie

1.2.1 - Introduzione alle licenze d'uso per i software

La tutela giuridica associata al software si concretizza comunemente attraverso lo strumento della licenza contrattuale, spesso definito **EULA**⁶⁴.

Sulla natura giuridica della licenza d'uso vi è, ad oggi, un aperto dibattito dottrinale: da un lato chi vede un contratto non tipizzato e interpretabile attraverso un'estensione analogica del contratto di locazione⁶⁵, dall'altro chi lo

⁶² "the agreement on Trade Related aspects of Intellectual Property Rights", trattato internazionale promosso dall'Organizzazione Mondiale del Commercio (OMC), ratificato dall'Italia con L. 747 del 29 dicembre 1994, https://www.uibm.gov.it/attachments/Accordo_trips.pdf

⁶³ Art. 27 TRIPs, I comma

⁶⁴ End User License Agreement

⁶⁵ Artt. 1571 e ss cc

intende come una vendita di un bene immateriale, anche in virtù della tipizzazione della fattispecie nelle già citate norme sul diritto d'autore. Tale ultima visione esegetica trova riscontro in una recente e non superata sentenza CGUE⁶⁶ la quale, interpretando il c.d. **principio di esaurimento comunitario**⁶⁷, chiarisce che “*An author of software cannot oppose the resale of his ‘used’ licences*”⁶⁸ (“un autore di un software non può opporsi alla rivendita delle sue licenze ‘usate’”).

Il contratto sotteso alla licenza si perfeziona, pertanto, in una delle seguenti tre fasi:

- ✓ Subito **prima dell’installazione del software**, quando viene richiesto se si vuole accettare la licenza;
- ✓ **Prima di scaricare il software** disponibile online;
- ✓ **Prima di aprire la custodia** con i supporti di massa contenenti il software.

Mentre, nei primi due casi, si parla in genere di **licenza a click** (traduzione del termine inglese **click-wrap license**), nel terzo si parla invece di **licenza a strappo** (**shrink-wrap license**).

⁶⁶ Sentenza CGUE caso C-128/11, c.d. “sentenza Oracle”, reperibile qui:

<https://curia.europa.eu/juris/document/document.jsf?docid=124564&doclang=IT>

⁶⁷ Principio proprio dell’ordinamento comune europeo secondo il quale il titolare di uno o più diritti di proprietà industriale o intellettuale perde le relative facoltà di privativa a seguito della prima messa in commercio del bene nel territorio UE

⁶⁸ <https://curia.europa.eu/jcms/upload/docs/application/pdf/2012-07/cp120094en.pdf>

Il perfezionamento del contratto avviene ai sensi e per l'effetto dell'**art. 1341 cc**, con particolare ma non esclusivo riguardo alle **clausole vessatorie**⁶⁹ e, nel caso di licenze a click, ci si pone il problema dell'equivalenza tra click e forma scritta, la cui analisi esonda dagli scopi dello scritto ma che è sintetizzabile nell'applicazione di due impianti normativi: il **Codice dell'Amministrazione Digitale**⁷⁰ e relative disposizioni attuative e il **Regolamento eIDAS**⁷¹.

Qualora il software sia solamente scaricabile (spesso noto con l'acronimo inglese **ESD - Electronic Software Distribution**), va ulteriormente considerata la norma che regola il commercio elettronico⁷², la quale concordemente precisa che *"le norme sulla conclusione dei contratti si applicano anche nei casi in cui il destinatario di un bene o di un servizio della società dell'informazione inoltri il proprio ordine per via telematica"*⁷³.

Sulla natura extracontrattuale di alcune licenze software, invece, si esprime parte della dottrina internazionale: il prof. MOGLEN, uno dei fondatori del movimento per il software libero e avvocato della Free Software Foundation, parlando di licenze GPL sostiene che *"A licence is a unilateral permission to use someone else's property [omissis]. A contract, on the other hand, is an exchange of obligations"* ("una licenza è un permesso unilaterale di usare la proprietà di qualcun altro, un contratto è uno scambio di obbligazioni")⁷⁴.

⁶⁹ Art. 1341 cc, II comma

⁷⁰ Noto come CAD, D. Lgs. 82 del 7 marzo 2005 e smi

⁷¹ Regolamento UE 910/2014

⁷² Direttiva 2000/31/CE, recepita in Italia con D. Lgs. 70 del 9 aprile 2013

⁷³ Art. 13 D. Lgs. 70/2013, I comma

⁷⁴ "The GPL is a License, not a Contract", <https://lwn.net/Articles/61292/>

Che il diritto d'autore ed i brevetti generino responsabilità extracontrattuali e illeciti penali è pacifico in diritto⁷⁵, dottrina⁷⁶ e giurisprudenza⁷⁷.

In differente contesto, ampia analisi meriterebbe anche la recente introduzione di meccanismi c.d. **DRM**⁷⁸, tesi ad applicare direttamente le previsioni contrattuali accettate, nonché a sanzionare il contraente che le disattenda, attraverso complessi meccanismi informatici basati sull'utilizzo di crittografia a chiave pubblica.

1.2.2 - Panoramica delle principali tipologie di licenze software

Esaurita quest'ampia panoramica sulla protezione dei programmi per elaboratore e sulla natura giuridica degli strumenti di protezione associati, passiamo a vedere quali sono le principali licenze software attualmente disponibili.

Volendo inizialmente dividere il panorama in macrocategorie, potremmo procedere come segue:

- ✓ **Software proprietario/closed source;**
- ✓ **Software FLOSS (Free/Libre and Open-Source Software).**

⁷⁵ Art. 171-bis cp

⁷⁶ In punto quo, B. BOCCHINI, La tutela del diritto d'autore, in Cod. resp. civ. e RC auto, a cura di Bonilini – Carnevali – Confortini, Milano, 2015, 1284 ss.

⁷⁷ Cass. Pen. n. 25104 del 19 giugno 2008, sulla pirateria informatica, in cui si precisa che *“Per la configurabilità del reato del reato di cui all'art.171 bis non è richiesto, infatti, che la riproduzione dei software sia finalizzata al commercio, essendo sufficiente il fine di profitto, come contestato, né il dolo specifico del fine di lucro”*

⁷⁸ Digital Rights Management

La principale differenza tra questi due ambiti non è, come molti pensano, la gratuità o meno del bene immateriale ma, bensì, il novero di diritti garantiti all'utente.

Una prima semplice articolazione delle licenze software è reperibile all'interno delle pagine⁷⁹ del progetto GNU⁸⁰, nella quale è ben visibile come alcune licenze non hanno una vera e propria categoria di appartenenza ma possono essere considerate, a seconda di alcuni dettagli, software FLOSS o proprietario.

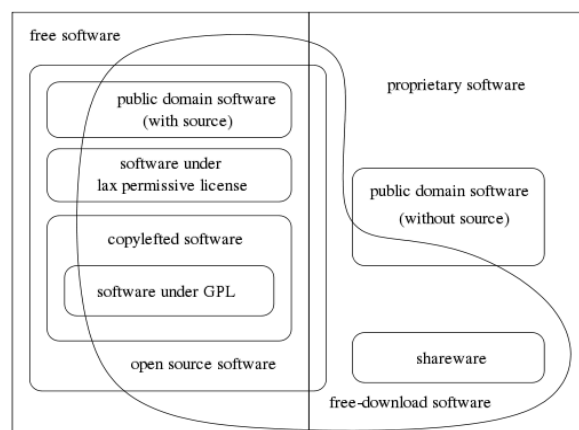


Figura 1 - Licenze software secondo Choa-Kuei

Va precisato come, in letteratura, non esista una vera e propria tipizzazione delle licenze software, per cui l'analisi che segue è da intendersi meramente indicativa e non esaustiva.

Stando alle definizioni della FSF, il **software proprietario** viene anche definito "**software non libero**", ovvero quando *"il suo utilizzo, la redistribuzione o*

⁷⁹ <https://www.gnu.org/philosophy/categories.en.html>, tradotta in italiano qui: <https://www.gnu.org/philosophy/categories.it.html>

⁸⁰ GNU's Not Unix, acronimo ricorsivo

*modifica sono proibiti o richiedono un permesso o sono sottoposti a tali vincoli che in pratica non si possono fare liberamente*⁸¹. Specifica poi come, in passato, si sia ricorsi alla distinzione tra “*software semilibero*’ (che consentiva modifica e redistribuzione non commerciale) e ‘*software proprietario*’ strettamente detto, che non la consentiva”⁸².

All’interno dello stesso documento, inoltre, viene distinto **il software free dal software freeware**: in questo caso si specifica che “*il termine ‘freeware’ non ha una definizione comunemente accettata, ma è comunemente utilizzato per i pacchetti software che possono essere ridistribuiti ma non modificati (ed il loro codice sorgente non è disponibile). Questi pacchetti non sono software libero, perciò non usate ‘freeware’ per indicare il software libero*”.

All’interno di questa categoria rientra anche il c.d. **software adware**, la cui assenza di corrispettivo economico viene garantita dalla visualizzazione periodica di “*messaggi pubblicitari non richiesti*”⁸³ che, a volte, contengono persino degli spyware⁸⁴.

Variante nella variante, il **software nagware** (detto anche **nag screen**, **begware** o **annoyware**) mostra finestre pop-up con inserzioni pubblicitarie che possono essere rimosse attraverso una donazione⁸⁵. In letteratura tecnica,

⁸¹ <https://www.gnu.org/philosophy/categories.it.html>

⁸² <https://www.gnu.org/philosophy/categories.it.html>

⁸³ Definizione adware dal dizionario Oxford Languages

⁸⁴ Software nascosti programmati per tracciare alcune azioni dell’utente

⁸⁵ https://it.wikipedia.org/wiki/Nag_screen

il software nagware viene spesso catalogato anche come variante dello shareware (*infra*).

E vengono considerati freeware anche il **software cardware** (detto anche **postcardware**), **mailware** o **emailware**⁸⁶ e **beerware**⁸⁷, la cui gratuità deve essere bilanciata dall'invio di una cartolina o una mail all'autore ovvero, nell'ultimo caso, dall'offrire volontariamente una birra all'autore.

Più recentemente, in tale categoria è stato introdotto il **freemium software** (contrazione dei termini free e premium), ovvero software freeware con la possibilità di sbloccare funzionalità aggiuntive a pagamento⁸⁸. Sebbene tale modalità di distribuzione del software sia disponibile già dagli anni '80, l'origine del termine freemium è assai più recente e può essere fatta risalire al 2006⁸⁹.

Sempre al di fuori del novero del software libero (e, in passato, anche di quello semilibero) si colloca lo **shareware**, licenza che consente *“di redistribuire copie, ma impone a chiunque continui ad usarne una copia di pagarne la licenza d'uso”*: le motivazioni vanno ricercate sia nell'indisponibilità del codice sorgente che nell'assenza del *“permesso di farne una copia ed installarlo senza pagare una licenza d'uso, neanche per chi lo utilizzi per un'attività senza fini di lucro”*⁹⁰.

⁸⁶ <https://it.wikipedia.org/wiki/Cardware>

⁸⁷ <https://it.wikipedia.org/wiki/Beerware>

⁸⁸ <https://it.wikipedia.org/wiki/Freemium>

⁸⁹ <https://vowe.net/archives/007069.html>

⁹⁰ <https://www.gnu.org/philosophy/categories.it.html>

Variante dello shareware è il **demoware** o **demo**, ovvero “*versione dimostrativa di un programma, presentato per lo più, a scopo di campagna pubblicitaria, in forma semplificata e parziale*”⁹¹.

Ulteriore variante dello shareware è rappresentata dal **software crippleware** o **trialware**, software le cui funzioni vengono limitate (dall’inglese “crippled”) qualora la licenza non venga acquistata entro un certo lasso di tempo⁹².

Variante meno diffusa, invece, è il **software careware** (oppure **charityware**, **helpware** o **goodware**)⁹³, ovvero software ceduto in cambio di una donazione liberale⁹⁴.

Il **software privato**, invece, può essere considerato software libero solo nel caso in cui l’utente “*gode di piene libertà sul programma privato*”, definizione non estendibile alle sue copie qualora “*l’utente distribuisce copie agli altri e non assegna a quelle copie le quattro libertà*”⁹⁵ [sulle quattro libertà del software libero, *infra*].

In ultimo, all’interno del già citato documento sul sito del progetto GNU, si parla di **software commerciale**, chiarendo che “*‘commerciale’ e ‘proprietario’ non sono la stessa cosa*”, in quanto il primo aggettivo è riferito alle finalità di lucro con cui viene sviluppato il software e non alle libertà ad esso associate; per cui,

⁹¹ Dizionario Treccani, https://www.treccani.it/vocabolario/demo_res-eceee31b-0017-11de-9d89-0016357eee51/

⁹² <https://en.wikipedia.org/wiki/Crippleware>

⁹³ <https://en.wikipedia.org/wiki/Careware>

⁹⁴ <https://www.collinsdictionary.com/dictionary/english/careware>

⁹⁵ <https://www.gnu.org/philosophy/categories.it.html>

sebbene *“la maggior parte del software commerciale è proprietario”, “c’è del software libero commerciale, e c’è software non commerciale non libero”*⁹⁶.

Categoria a sé è quella del **software abandonware**, ovvero software obsoleto (pubblicato da almeno 5 anni, stando al “Lessico del XXI Secolo” Treccani) che, pur non essendo più di interesse commerciale, mantiene un interesse storico o individuale ed è ancora protetto da diritto d’autore. L’utilizzo, tecnicamente costituirebbe una forma di pirateria informatica⁹⁷.

Volendo categorizzare il software sulla base della durata della licenza, invece, bisogna distinguere tra **licenza perpetua** e **abbonamento**: sebbene dal punto di vista normativo ci siano poche differenze (per lo più legate ad aspetti fiscali) e si ricada, in entrambi i casi, sostanzialmente nelle categorie già esaminate, dal punto di vista operativo le formule di abbonamento presentano notevoli vantaggi, particolarmente legati alla gestione dell’obsolescenza e all’abbattimento dei costi fissi a favore di quelli variabili.

Tali formule contrattuali, inoltre, sono in crescita costante, grazie anche all’incremento nell’utilizzo di software in modalità cloud SaaS⁹⁸, diretta conseguenza dello smart working ed e-learning che accompagnano questo periodo di pandemia: la tendenza a cui assistiamo, in ultima analisi, è quella di una rapida transizione dalla civiltà del possesso a quella dell’accesso.

⁹⁶ <https://www.gnu.org/philosophy/categories.it.html>

⁹⁷ <https://it.wikipedia.org/wiki/Abandonware>

⁹⁸ Software as a Service

Il termine perpetua, inoltre, non deve trarre in inganno, perché alcuni dei servizi direttamente legati alla licenza non hanno vita eterna (supporto tecnico, compatibilità, etc.).

Una via di mezzo tra i due approcci è costituita dai **programmi di maintenance**, acquistabili assieme alla licenza perpetua, i quali consentono opzioni flessibili quali l'aggiornamento alle nuove versioni senza costi aggiuntivi, la disponibilità per l'acquirente delle ultime versioni del software, a scelta (tipicamente le ultime due), di bloccare o ridurre il prezzo per l'acquisto di ulteriori licenze durante il periodo di validità del programma, supporto tecnico dedicato, tipicamente più rapido e specifico di quello generalmente fornito.

CAPITOLO 2 – Le licenze FLOSS e di Pubblico Dominio

Un capitolo a parte meritano le licenze “non commerciali”, oggetto di una lunga ed interessante storia incentrata sul concetto di “libertà” applicato al software e a ciò che attorno ad esso gravita (e.g. documentazione).

2.1 – La Free Software Foundation

2.1.1 – Cenni storici

Richard Matthew Stallman⁹⁹, hacker conosciuto negli anni settanta con lo pseudonimo di “rms”, inizia ad approcciare all’informatica subito dopo il suo diploma, tra il finire degli anni sessanta e l’inizio dei settanta. Pioniere dell’intelligenza artificiale, che iniziò a praticare presso il MIT¹⁰⁰ sin dal 1971, si laureò in fisica nel 1974. Quando nel 1977 il laboratorio presso il quale lavorava come programmatore¹⁰¹ attivò un sistema di accesso protetto da password, Stallman mostrò immediatamente la propria contrarietà a questo tipo di approccio inviando un messaggio agli altri utenti del laboratorio con la loro password decodificata, invitandoli a modificarla con una stringa vuota, così da consentire l’accesso anonimo al sistema: questa azione rivelò immediatamente le doti di leadership del giovane Richard, posto che il 20% degli utenti seguì il suo consiglio.

⁹⁹ https://it.wikipedia.org/wiki/Richard_Stallman

¹⁰⁰ Massachusetts Institute of Technology, Boston

¹⁰¹ LCS – Laboratory for Computer Science presso il MIT

A partire dal finire degli anni settanta, la maggior parte dei produttori di software smise di distribuire il codice sorgente e iniziò ad utilizzare sistema di licenza a protezione del proprio software (in americano, **copyright**), grazie all'introduzione nell'ordinamento americano della principale norma di protezione attualmente in vigore, sebbene modificata rispetto alla versione iniziale (1976), il **Copyright Act**¹⁰².

Per mero tuziorismo, bisogna precisare come l'accezione italiana del termine (diritto d'autore) sia molto più ampia di quella americana: questo è dovuto al fatto che, mentre il nostro ordinamento protegge -come visto- sia i diritti morali che quelli patrimoniali, l'ordinamento statunitense è fortemente sbilanciato verso questi ultimi, accentrandone l'attenzione anche nel nome dello strumento giuridico utilizzato (diritto di copia). Ciononostante, non è infrequente trovare i due vocaboli utilizzati come sinonimi.

Quando, nel 1979, **Brian Keith Reid**¹⁰³, uno dei padri di internet, inserì dei meccanismi di protezione informatica al suo software **Scribe** per impedirne l'uso illegittimo¹⁰⁴ e venderle alla Unilogic (che avrebbe successivamente assunto il nome di Scribe System), Richard Stallman parlò di "*crimine contro l'umanità*"¹⁰⁵.

¹⁰² <https://www.copyright.gov/title17/title17.pdf>

¹⁰³ [https://en.wikipedia.org/wiki/Brian_Reid_\(computer_scientist\)](https://en.wikipedia.org/wiki/Brian_Reid_(computer_scientist))

¹⁰⁴ Inserì delle bombe a tempo che, dopo l'installazione, ne disabilitavano le funzionalità in assenza di licenza

¹⁰⁵ "Codice libero. Free as in freedom. Richard Stallman e la crociata per il software libero", Sam Williams, edito da Apogeo, 1.1.2003, ISBN 9788850321087

Un anno dopo, Stallman lavorò al software di gestione della prima stampante laser mai creata (Xerox 9700 “Dover”): grazie al suo contributo, ottenuto attraverso prove effettuate su una vecchia stampante, la stampante era in grado di avvisare l’utente del termine della stampa. La Xerox, però, rifiutò di condividere il codice completo¹⁰⁶: Stallman avrebbe voluto introdurre un’ulteriore modifica che avrebbe allertato l’utente nel caso di congestioni della stampa, circostanza estremamente utile all’epoca in cui le stampanti erano poche, condivise e a costi altissimi.

Questa serie di eventi contribuì a formare nel giovane Richard l’idea che sarebbe stato necessario dare la possibilità agli utenti del software di modificarlo secondo le proprie esigenze.

A partire dallo stesso anno e fino al 1983, molti dei programmatori del LCS fondarono vari assetti societari, formando alleanze e contrapposizioni, con l’obiettivo di fornire computer da laboratorio. Questi movimenti portarono alla creazione di due società: la Lisp Machines, Inc. (LMI) e la Symbolics. Stallman, contrario alla politica di entrambe (che prevedeva un uso massiccio di NDA¹⁰⁷ per impedire la divulgazione del codice sorgente e informazioni tecniche), provò a clonare il lavoro della Symbolics per dividerlo con altri, in piena linea con i propri principi espressi fino a quel momento: questo evento segna l’inizio della creazione della licenza **GNU GPL**, prima licenza di tipo **copyleft** (*infra*).

¹⁰⁶ Fenomeno che prende il nome di **software hoarding**

¹⁰⁷ NDA – Non-Disclosure Agreement, accordi di non divulgazione

Ed è proprio *in limine* a questa esperienza che, nel settembre del 1983, egli fonda il **progetto GNU**, il cui primo manifesto¹⁰⁸ sarà poi pubblicato nel 1985 (seppur riprendendo annunci proprio del 1983) che verrà poi modificato fino al 1987 e al quale verranno ancora successivamente aggiunte svariate note in calce¹⁰⁹, a chiarimento dei vari punti controversi.

Tale progetto, come il nome stesso suggerisce, mirava a creare un sistema operativo “libero” (nell’accezione inglese del termine, *infra*) compatibile con Unix. E fu nel 1991 che uno studente finlandese, **Linus Torvalds**, realizzò il kernel di un sistema operativo realizzato in conformità al progetto GNU: tale sistema operativo prese il nome proprio da quello studente, quello che tutti oggi chiamiamo **Linux**.

Subito dopo aver pubblicato la prima versione dello GNU Manifesto, Stallman fondò la Free Software Foundation¹¹⁰, associazione no-profit ancora oggi molto attiva nel campo della promozione del software libero promuovendone gli aspetti legali, politici e filosofici.

¹⁰⁸ <https://www.gnu.org/gnu/manifesto.en.html>, tradotto in italiano qui: <https://www.gnu.org/gnu/manifesto.it.html>

¹⁰⁹ A partire dal 1993

¹¹⁰ <https://www.fsf.org/>

2.1.2 - Il software libero

Il concetto di software libero e le licenze ad esso associate sono ben descritte all'interno della documentazione relativa al progetto GNU¹¹¹. In primo luogo, viene chiarita l'accezione del termine "free" che, in lingue diverse dall'inglese, si presta a molteplici interpretazioni. Nella pagina dedicata del progetto GNU si legge che il focus del software free è la libertà e non il prezzo: "*To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer'. We sometimes call it 'libre software', borrowing the French or Spanish word for 'free' as in freedom, to show we do not mean the software is gratis*". La parola "free", quindi, deve essere Intesa come sinonimo di libertà e non di gratuità ed è per questo che, in italiano, l'espressione *free software* è stata tradotta con *software libero* e non *software gratis*.

Passando ad una prima sintetica definizione, gli utenti del software libero "*hanno la libertà di eseguire, copiare, distribuire, studiare, modificare e migliorare il software*".

Entrando maggiormente nello specifico, una licenza può essere considerata di software libero se rispetta le **quattro libertà essenziali**, ovvero, citando testualmente¹¹²:

- o Libertà di eseguire il programma come si desidera, per qualsiasi scopo (originariamente non presente, introdotta successivamente);

¹¹¹ <https://www.gnu.org/philosophy/free-sw.html.en>, tradotto in italiano qui: <https://www.gnu.org/philosophy/free-sw.it.html>

¹¹² "Cos'è il software libero – Le quattro libertà essenziali", <https://www.gnu.org/philosophy/free-sw.it.html>

- 1 Libertà di studiare come funziona il programma e di modificarlo in modo da adattarlo alle proprie necessità. L'accesso al codice sorgente ne è un prerequisito;
- 2 Libertà di ridistribuire copie in modo da aiutare gli altri;
- 3 Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti da voi apportati (e le vostre versioni modificate in genere), in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito.

Si noti come, conformemente agli “standard informatici” comunemente utilizzati, la numerazione delle citate libertà inizi proprio dal numero 0.

All'interno della stessa pagina, successivamente, viene meglio esplicitata ciascuna delle quattro libertà e viene ribadito che anche il software con scopi commerciali può essere libero, qualora conforme.

Tra i chiarimenti, trova particolare rilevanza quello relativo al **software di pubblico dominio**, ovvero “*software privo di copyright*”¹¹³: non essendo assoggettato ad alcuna condizione di licenza, possiamo considerare questo tipo di approccio accettabile per il mondo del software libero quando il codice sorgente è disponibile, non conforme quando non lo è. In ogni caso, tale software non è mai assoggettato a copyleft (*infra*). Da un punto di vista giuridico, a partire dalla già analizzata Convenzione di Berna e successive

¹¹³ “Classificazione del software libero e non libero – Software di pubblico dominio”, <https://www.gnu.org/philosophy/categories.it.html>

norme derivate, **i programmi per elaboratore sono automaticamente coperti da diritto d'autore** e pertanto, affinché il software sia considerato di pubblico dominio, deve legalmente e per iscritto disconoscerne i diritti.

2.1.3 - Il copyleft

Sul finire degli anni '70, il programmatore statunitense Li-Chen Wang¹¹⁴ realizzò uno dei tanti interpreti per il neonato linguaggio di programmazione BASIC¹¹⁵, il **Palo Alto Tiny BASIC**, sviluppato per processore Intel 8080. Nel 1976, il giovane **William Henry Gates III**, meglio noto con il nome di Bill Gates¹¹⁶, creatore di un precedente interprete per il linguaggio BASIC¹¹⁷, scrisse una famosa lettera¹¹⁸ agli appassionati di computer dell'Homebrew Computer Club¹¹⁹, introducendo probabilmente per la prima volta il concetto di **pirateria informatica**, la quale avrebbe causato ingenti danni alla sua società, la Microsoft, attraverso la diffusa pratica di copia del software.

¹¹⁴ https://it.wikipedia.org/wiki/Li-Chen_Wang

¹¹⁵ Beginner's All-purpose Symbolic Instruction Code

¹¹⁶ https://it.wikipedia.org/wiki/Bill_Gates

¹¹⁷ Altair BASIC, https://it.wikipedia.org/wiki/Altair_BASIC

¹¹⁸ "Open Letter to Hobbyist",

https://it.wikipedia.org/wiki/Lettera_aperta_agli_hobbisti#/media/File:Bill_Gates_Letter_to_Hobbyists.jpg,

disponibile in formato editabile qui:

https://www.digibarn.com/collections/newsletters/homebrew/V2_01/gatesletter.html

¹¹⁹ https://it.wikipedia.org/wiki/Homebrew_Computer_Club

“Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?”

(“Perché questo? La maggior parte degli hobbisti deve essere consapevole che molti di voi rubano il vostro software. L'hardware va pagato, ma il software sembra sia qualcosa da condividere. A chi gliene importa se le persone che ci hanno lavorato sopra vanno pagate?”)

Per rispondere a questa lettera, Li-Chen decise di inserire, nei listati del suo compilatore, la frase **@Copyleft - All Wrongs Reserved**, scherzosamente opposta a **@Copyright - All Rights Reserved**¹²⁰ che appariva sul software proprietario: pare sia stata proprio questa la prima volta che il vocabolo è stato utilizzato.

```
*****  
;   
; TINY BASIC FOR INTEL 8080  
; VERSION 2.0  
; BY LI-CHEN WANG  
; MODIFIED AND TRANSLATED  
; TO INTEL MNEMONICS  
; BY ROGER RAUSKOLB  
; 10 OCTOBER, 1976  
; @COPYLEFT  
; ALL WRONGS RESERVED  
;   
*****
```

Figura 2 - Primo utilizzo della parola copyleft

¹²⁰ Formalità da indicare nella licenza, inizialmente prevista dalla Convenzione di Buenos Aires del 1910

Secondo alcuni, inoltre, la parola copyleft, oltre all'evidente gioco di parole con il termine copyright, voleva essere un chiaro rimando agli ambienti di sinistra, spesso contrari al diritto d'autore.

Un'altra interpretazione, invece, vorrebbe attribuire le origini del copyleft al **discordianesimo**¹²¹, religione satirico/parodistica fondata nel 1963: all'interno del principale testo di riferimento, "*Principia Discordia*", vi si troverebbe il termine **Copyleft - All Rights Reversed**. Il termine rights (diritti) sarebbe stato utilizzato per la sua assonanza con il termine rites (riti).

In ogni caso, successivamente fu anche ideato un simbolo per il copyleft, in realtà privo di valore legale ma, ancora una volta, con intenti satirici perché simile ma opposto a quello del copyright (simbolo del copyleft).

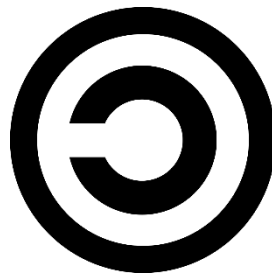


Figura 3 - Simbolo del copyleft

Quale che fosse la sua origine, la FSF introdusse il concetto di copyleft nel progetto GNU, specificando come "*il software con copyleft è software libero le cui condizioni di distribuzione impongono che tutte le copie di tutte le versioni siano*

¹²¹ <https://it.wikipedia.org/wiki/Discordianesimo>

*distribuiti sotto condizioni simili*¹²². Come vedremo a breve, il copyleft segnerà un netto e sostanziale distacco tra la FSF ed un movimento simile, la OSI¹²³.

2.1.4 - Le licenze free software

Il principale tipo di licenza che troviamo nel progetto GNU è la **licenza GNU GPL**¹²⁴, attualmente arrivata alla versione 3. Tale licenza, nella sua prima versione, è il primo chiaro esempio di applicazione del concetto di copyleft ad una licenza software.

Va però precisato come non tutte le licenze della FSF usino questo concetto: esistono alcune licenze (e.g. Apache web server, BSD) che, pur essendo approvate da FSF, proprio per la mancata conformità al copyleft, sono proprie della filosofia Open Source più che di quella del free software.

Tornando al copyleft, nel tempo, le licenze si sono moltiplicate, venendo raggruppate a seconda delle specificità:

- ✓ **Copyleft debole**, non tutte le opere derivate ereditano la licenza copyleft (esempi sono GNU Lesser General Public License - LGPL, Mozilla Public License - MPL);
- ✓ **Copyleft forte**: tutte le opere derivate e le librerie collegate ereditano la licenza copyleft (GPL);

¹²² “Classificazione del software libero e non libero – Software con copyleft”, <https://www.gnu.org/philosophy/categories.it.html>

¹²³ Open Source Initiative, <https://opensource.org/>

¹²⁴ GNU General Public License, <https://www.gnu.org/licenses/gpl-3.0.txt>

- ✓ **Cloud (o network) copyleft:** è necessario condividere il sorgente anche se disponibile SaaS (Software as a Service) – Affero GPL (AGPL, come la GPL ma con una sezione aggiuntiva per l'utilizzo in rete), European Union Public License (EUPL, prima licenza europea FLOSS, Free, Libre, Open Source Software);
- ✓ **Copyleft completo:** modifica illimitata dell'opera tranne la stessa licenza;
- ✓ **Copyleft parziale:** possibili modifiche solo di alcune parti dell'opera.

Una particolare categoria di licenze copyleft è costituita dalle licenze “**share alike**”¹²⁵: questo tipo di licenze prevede la non modificabilità dei diritti previsti dall'opera originale nelle derivate (e.g. alcuni tipi di licenza **Creative Commons**¹²⁶, la quale per prima ha introdotto questo tipo di licenza). Secondo parte della dottrina specializzata, il copyleft completo sarebbe un particolare tipo di licenza share alike (ma non il contrario). All'interno delle licenze share alike viene riportata spesso la dicitura “**Share alike – some rights reserved**”.

Come già accennato in precedenza, il concetto di copyleft non è applicabile solo a programmi per elaboratore ma è assai più ampio. Di seguito sono indicati alcuni esempi di licenze copyleft in tal senso:

¹²⁵ <https://en.wikipedia.org/wiki/Share-alike>

¹²⁶ <https://creativecommons.it/chapterIT/>

- ✓ **Design Science License**¹²⁷: copyleft forte applicabile a dati generici purché non siano software, documentazione o arte (non raccomandata da FSF);
- ✓ **Against DRM license**¹²⁸: copyleft per lavori artistici pubblicata dalla Free Creations (il cui sito è attualmente reperibile su “The way back machine”)¹²⁹ che prevede la nullità o annullabilità delle particolari condizioni di licenza se il lavoro utilizza una protezione di tipo DRM¹³⁰;
- ✓ **Creative Commons (o licenze CC)**¹³¹: licenza per opere d’ingegno generica molto interessante per l’utilizzo del license chooser¹³² che consente di determinare con quale licenza proteggere la propria opera sulla base dei requisiti richiesti;
- ✓ **GNU Free Documentation Licence (GNU FDL, GFDL o FDL)**¹³³: usata principalmente per documentazione;
- ✓ **Licenza Arte Libera (Licence Art Libre – LAL o Free Art License – LAL)**¹³⁴: licenza francese per opere artistiche.

¹²⁷ https://it.wikipedia.org/wiki/Design_Science_License

¹²⁸ https://en.wikipedia.org/wiki/Against_DRM_license

¹²⁹ <https://web.archive.org/web/20170327160130/http://www.freecreations.org/>

¹³⁰ https://it.wikipedia.org/wiki/Digital_rights_management

¹³¹ <https://creativecommons.it/chapterIT/>

¹³² <https://chooser-beta.creativecommons.org/>

¹³³ https://it.wikipedia.org/wiki/GNU_Free_Documentation_License

¹³⁴ https://it.wikipedia.org/wiki/Licence_Art_Libre

2.2 – La Open Source Initiative

Accanto e parallelamente alla storia della Free Software Foundation si sviluppa la storia di un altro grande movimento legato al mondo delle licenze software e alla distribuzione della conoscenza in questo settore intesa come dovere morale e filosofico, la Open Source Initiative (OSI)¹³⁵, nata in realtà alcuni anni dopo.

2.2.1 – Cenni storici

La nascita del termine Open Source associato al software viene fatta risalire al 3 febbraio 1998: **Christine Peterson**¹³⁶, analista dati americana, durante una conferenza a Palo Alto, California, subito dopo l’annuncio del rilascio del codice sorgente del noto browser **Netscape**¹³⁷, parlò della necessità di trovare un nome più corretto rispetto a free software per evitare di ingenerare confusione, in quanto “*those new to the term ‘free software’ assume it is referring to the price*” (coloro i quali sono nuovi al termine free software pensano sia riferito al prezzo). Nell’intervento spiegò che, passate in rassegna varie alternative in vari incontri con altri esperti (quali “*freely distributable*”, “*cooperatively developed*” o “*sourceware*”), sebbene presentasse alcuni punti da chiarire, lei preferisse il termine “**Open source software**”.

¹³⁵ <https://opensource.org/>

¹³⁶ https://en.wikipedia.org/wiki/Christine_Peterson

¹³⁷ https://it.wikipedia.org/wiki/Netscape_Navigator

Nello stesso febbraio del 1998, qualche giorno dopo, **Eric Raymond** e **Bruce Perens** (due delle tante persone che avevano contribuito alla nascita di quel nuovo termine) fondarono la Open Source Initiative.

L'interesse per il mondo del software non proprietario, in quegli anni, era altissimo, tant'è che il 10 agosto dello stesso anno il giovanissimo Linus Torvalds viene citato da Forbes¹³⁸, evidenziando come dai 10 utenti che scaricarono la versione 0.01 del kernel all'inizio del 1991 (di cui 5 lo aiutarono a risolvere alcuni problemi) e dalle 10.000 linee di codice iniziali si passò rapidamente ai 7.500.000 utenti della versione 2.1.110 del 1998 (centodecimo aggiornamento della versione 2.1, circa 10.000 programmatori coinvolti nel progetto) ed al 1.500.000 di righe di codice.

La mission della OSI, invece, fu comunicata successivamente, ad aprile del 1998, durante il Free Software Summit, ed era quella di educare, spiegare e proteggere l'etichetta Open Source.

Le prime attività dell'associazione sono invece datate al 1999, anno in cui OSI supporta una petizione per incoraggiare l'uso del software Open Source¹³⁹. Nelle citate pagine che descrivono la storia dell'OSI, dal punto di vista del movimento, viene precisato come le attività sociali si svolgono più sotto la forma della calma persuasione che dell'attivismo pubblico, segnando anche in questo una netta differenza rispetto alla FSF.

¹³⁸ <https://www.forbes.com/global/1998/0810/0109044s1.html?sh=321f14a44aa9>

¹³⁹ <http://web.archive.org/web/19990117101010/http://opensource.org/>

2.2.2 - La definizione di Open Source

Va precisato anche in questo caso come il termine **Open** sia da intendersi in accezione assai ampia: la mera visibilità del codice sorgente, come vedremo meglio in seguito, non soddisfa da sola i requisiti necessari a classificare il software come Open Source.

In un lungo articolo¹⁴⁰, però, Richard Stallman spiega il punto di vista FSF sull'Open Source, chiarendo anche la questione terminologica e precisando **“Perché l'Open Source manca l'obiettivo del Software Libero”**. L'articolo spiega con dovizia di particolari le differenze tra i due movimenti, differenze riassumibili in un passaggio chiave:

“The two now describe almost the same category of software, but they stand for views based on fundamentally different values. For the free software movement, free software is an ethical imperative, essential respect for the users' freedom. By contrast, the philosophy of open source considers issues in terms of how to make software “better”—in a practical sense only. It says that nonfree software is an inferior solution to the practical problem at hand.

For the free software movement, however, nonfree software is a social problem, and the solution is to stop using it and move to free software”

*(“I due termini descrivono all'incirca la stessa categoria di software. Ma si basano su valori fondamentalmente diversi. **L'open source è una metodologia di sviluppo; il software libero è un movimento sociale.** Per il movimento per il*

¹⁴⁰ “Why Open Source misses the point of Free Software”, <https://www.gnu.org/philosophy/open-source-misses-the-point.html.en>, tradotto in italiano qui: <https://www.gnu.org/philosophy/open-source-misses-the-point.it.html>

software libero, il software libero è un imperativo etico, il rispetto essenziale della libertà degli utenti. Al contrario la filosofia dell'open source pensa a come "migliorare" il software soltanto da un punto di vista pratico. Dice che il software non libero è una soluzione non ottimale. Spesso le discussioni sull'"open source" non considerano quel che è giusto o sbagliato, ma solo il successo e la popolarità. Per il movimento per il software libero, tuttavia, il software non libero è un problema sociale e la soluzione è passare al software libero").

Tale interpretazione viene peraltro condivisa dalla stessa Open Source Initiative che, rispondendo alla domanda *"What is free software and is it the same as open source?"* ("Che cos'è il software libero ed è la stessa cosa del software open source?")¹⁴¹, spiega che *"software freedom was primarily a practical issue rather than an ideological one"* ("la libertà del software era per lo più un problema pratico piuttosto che ideologico").

In merito al termine utilizzato, scrive Stallman che *"il significato ovvio della espressione 'software open source' è 'Puoi guardare il codice sorgente' e la maggior parte delle persone sembra attribuire questo significato al termine. Questo è un criterio molto più debole che non quello di software libero e molto più debole della definizione ufficiale di open source. In questo criterio rientrano molti programmi che non sono liberi né open source"*.

Venendo al punto di vista OSI, la definizione di Open Source si basò inizialmente sulle Debian Free Software Guidelines, parte del Debian Social Contract¹⁴². La

¹⁴¹ <https://opensource.org/faq#free-software>

¹⁴² https://www.debian.org/social_contract.html

versione 1.0, la cui prima bozza fu realizzata proprio da Bruce Perens (il primo vicepresidente dell'associazione), è datata al 5 luglio 1997, poi aggiornata alla versione 1.1 il 26 aprile 2004, considerata di estrema importanza per l'aggiunta della clausola di neutralità tecnologica. L'attuale versione¹⁴³ è la 1.9 del 22 marzo 2007, disponibile sul sito OSI anche in versione commentata¹⁴⁴.

In questo caso, i principi su cui si deve basare una licenza sono dieci (che iniziano dal punto 1) e sono i seguenti:

- 1 Ridistribuzione libera;
- 2 Codice sorgente disponibile e distribuibile e non offuscato. Forme intermedie (come quelle generate dall'output del preprocessore) non sono consentite;
- 3 Possibili modifiche ai lavori derivati, ridistribuiti alle stesse condizioni;
- 4 Integrità del codice sorgente dell'autore originale anche attraverso il versioning;
- 5 Principio di non discriminazione di persone o gruppi;
- 6 Principio di non discriminazione rispetto a specifici campi di applicazione;

¹⁴³ <https://opensource.org/osd>

¹⁴⁴ <https://opensource.org/osd-annotated>

- 7 Principio di distribuzione della licenza, ovvero i diritti connessi alla licenza vengono automaticamente estesi a tutti coloro i quali il programma viene redistribuito;
- 8 La licenza deve essere agnostica rispetto al prodotto o alla distribuzione software di cui il prodotto fa parte, anche in caso di estrazione di un software da una distribuzione più ampia;
- 9 La licenza non deve porre limitazioni ad altri software distribuiti assieme al software Open Source;
- 10 La licenza deve essere agnostica rispetto alla tecnologia utilizzata.

Volendo analizzare la principale differenza con i principi FSF da un punto di vista ontologico, si ravvisa una differente interpretazione del concetto di libertà: mentre la OSI ritiene una limitazione della libertà il fondamentale principio espresso dal copyleft, ovvero la modifica della licenza originaria nelle opere derivate, la FSF risulta essere di opinione diametralmente opposta, definendo come libertà proprio la diffusione del sapere.

E, difatti, è proprio la Open Source Initiative ad introdurre le cosiddette licenze **non copyleft** o **permissive Open Source**¹⁴⁵, secondo le quali il software che sia stato rilasciato sotto questo tipo di licenza può essere usato come parte di programma distribuito sotto altre licenze, ivi incluse licenze proprietarie non Open Source.

¹⁴⁵ <https://opensource.org/faq#permissive>

In merito alla sola ispezionabilità del codice sorgente, OSI chiarisce la sua posizione ancora nella FAQ page, spiegando che “*open source specifically entails not mere inspection access but also conveying to recipients the perpetual right to fork covered code and use it without additional fees*” (“open source consente ai destinatari specificatamente non solo la mera ispezione ma anche il diritto perpetuo di creare opere derivate ed utilizzare il codice sorgente senza dover pagare null’altro”).

Un elenco completo delle licenze conformi alla OSD è disponibile sul sito OSI¹⁴⁶. Affinché una licenza possa essere inserita in questo elenco, deve superare un rigido processo¹⁴⁷, al termine del quale, se conforme, riceve l’approvazione dell’associazione. A partire dai primi anni duemila, a causa dell’eccessiva proliferazione delle licenze Open Source¹⁴⁸, tale processo è diventato assai più rigido e, nel 2006, è stato pubblicato il “**Report of License Proliferation Committee and draft FAQ**”¹⁴⁹, che chiarisce e specifica alcune caratteristiche che le licenze devono avere per superare il vaglio Open Source (non devono duplicare licenze esistenti, devono essere scritte in maniera chiara, semplice e comprensibile, devono essere riutilizzabili). Tale rapporto, inoltre, propone una precisa classificazione delle licenze, dividendole in licenze popolari, licenze per scopi speciali, licenze ridondanti (specificando anche le licenze rispetto alle quali lo sono), licenze non riutilizzabili, altri tipi di licenze, licenze superate e,

¹⁴⁶ <https://opensource.org/licenses/alphabetical> (alfabetico), <https://opensource.org/licenses/category> (categoria)

¹⁴⁷ <https://opensource.org/approval>

¹⁴⁸ <https://opensource.org/proliferation>

¹⁴⁹ <https://opensource.org/proliferation-report>

in ultimo, licenze ritirate volontariamente: tali gruppi, per il futuro, costituiranno ulteriore criterio per l'approvazione di nuove licenze.

2.3 – Casi di studio

Vediamo ora alcuni casi di studio che hanno diviso il mondo dell'Open Source e quello del Free Software sull'applicabilità delle proprie tipologia di licenza ai prodotti indicati.

2.3.1 – Il caso RHEL

All'interno della community Open Source, fa molto discutere il caso di una distribuzione Linux commerciale (Red Hat Enterprise Linux, RHEL) e della licenza connessa¹⁵⁰: sebbene il codice sia ispezionabile, ci sono alcune limitazioni all'utilizzo del codice di natura commerciale, in particolare sulla ridistribuibilità e, pertanto, stando all'interpretazione del blog OpenLogic¹⁵¹, *“While the code may be open, there's a lack of freedom. So according to the ideology of open source software, Red Hat Enterprise Linux is not open source”* (“Sebbene il codice sia aperto, ci sono delle limitazioni di libertà. Per questo, in accordo all'ideologia Open Source, RHEL non è Open Source”). Tale interpretazione, quindi, andrebbe in contiguità con quella appena citata

¹⁵⁰ https://www.redhat.com/licenses/Enterprise_Agreement_Webversion_EMEA_Italy_Italian_20180312.pdf

¹⁵¹ <https://www.openlogic.com/blog/red-hat-enterprise-linux-open-source>

relativamente alle licenze compatibili con la OSD, nonché con la circostanza che la licenza citata non appare nell'elenco delle licenze compatibili OSD¹⁵².

2.3.2 – Il caso Microsoft Visual Studio Code

Altro caso interessante, citato proprio da Stallman nella citata lettera¹⁵³, è quello di Microsoft Visual Studio: nel paragrafo *“Practical Differences between Free Software and Open Source”*, il fondatore di FSF porta ad esempio la licenza Microsoft come evidente differenza tra la filosofia Open Source e quella del Free Software (parte non tradotta in italiano): *“when a program's source code carries a weak license, one without copyleft, its executables can carry additional nonfree conditions. Microsoft does this with Visual Studio Code, for example. If these executables fully correspond to the released sources, **they qualify as open source but not as free software**. However, in that case users can compile the source code to make and distribute free executables”* (“quando il codice sorgente di un programma ha una licenza debole, una licenza senza il copyleft, i suoi eseguibili possono essere assoggettati a condizioni aggiuntive che qualificano la licenza come non libera. Microsoft fa questo con Visual Studio Code, per esempio. Se questi eseguibili corrispondono pienamente ai sorgenti rilasciati, vengono qualificati come open source ma non come software libero”).

¹⁵² <https://opensource.org/licenses/alphabetical>

¹⁵³ “Why Open Source misses the point of Free Software”, <https://www.gnu.org/philosophy/open-source-misses-the-point.html.en>, tradotto in italiano qui: <https://www.gnu.org/philosophy/open-source-misses-the-point.it.html>

In particolare, la parte della licenza di Visual Studio Code¹⁵⁴ a cui Stallman fa riferimento è quella relativa alle limitazioni d'uso, indipendenti dalla disponibilità del sorgente. Si legge nelle condizioni, al punto 5 “*Scope of license*”, di una serie di limitazioni relative al reverse engineering, decompilazione e disassemblaggio, derivazione del codice sorgente, rimozione, minimizzazione, modifica o blocco di qualsiasi avviso di Microsoft compaia, condivisione, pubblicazione, noleggio, leasing o vendita che non sono compatibili con la logica del software libero (libertà 1, 2 e 3).

5. **SCOPE OF LICENSE.** This license applies to the Visual Studio Code product. Source code for Visual Studio Code is available at <https://github.com/Microsoft/vscode> under the MIT license agreement. The software is licensed, not sold. This agreement only gives you some rights to use the software. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it in certain ways. You may not

- reverse engineer, decompile or disassemble the software, or otherwise attempt to derive the source code for the software except and solely to the extent required by third party licensing terms governing use of certain open source components that may be included in the software;
- remove, minimize, block or modify any notices of Microsoft or its suppliers in the software;
- use the software in any way that is against the law;
- share, publish, rent or lease the software, or provide the software as a stand-alone offering for others to use.

Figura 4 - Art. 5 licenza Visual Studio Code

Tali limitazioni, invece, appaiono perfettamente in linea con la filosofia Open Source: infatti Microsoft, nella pagina della licenza di Visual Studio Code, rende disponibile il sorgente, indicandone il repository¹⁵⁵ e ulteriore materiale aggiuntivo¹⁵⁶ tra cui la licenza che, come specificato, è la licenza MIT (che rientra nella categoria permissive Open Source), “*A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed*

¹⁵⁴ <https://code.visualstudio.com/License/>

¹⁵⁵ <https://github.com/Microsoft/vscode>

¹⁵⁶ Licenza: <https://github.com/microsoft/vscode/blob/main/LICENSE.txt>, FAQs: <https://code.visualstudio.com/docs/supporting/faq>

under different terms and without source code”, licenza che appare in elenco tra quelle approvate dalla Open Source Initiative¹⁵⁷.

2.3.3 – Il caso Android con licenza GNU GPL 2

Continuando con l’analisi, ci dice Stallman che *“molti prodotti che contengono computer controllano l’integrità dei loro programmi eseguibili per impedire all’utente di installare eseguibili diversi; solo una specifica azienda può produrre eseguibili che funzionino sul dispositivo e che ne possano sfruttare tutte le capacità. Chiamiamo questi dispositivi ‘tiranni’ e questa pratica ‘tivoization’, dal nome del primo prodotto (Tivo) in cui l’abbiamo incontrata. Anche se questi eseguibili vengono da codice sorgente libero, gli utenti non possono eseguirne versioni modificate, quindi l’eseguibile è non libero. I criteri per l’open source non riconoscono questo problema; guardano solamente la licenza del codice sorgente. Quindi questi eseguibili non modificati, quando prodotti a partire da codice sorgente come Linux che è open source e libero, sono open source ma non liberi. Molti prodotti basati su Android contengono eseguibili Linux non liberi e ‘tivoizzati’*”. La versione originale inglese della lettera, assai più precisa, specifica come, in realtà, buona parte di questo software sia stato rilasciato sotto licenza GNU GPL versione 2, circostanza che ha portato questa licenza ad evolvere nella versione 3, introducendo una nuova ulteriore differenza tra il mondo Open Source e quello del Free Software.

¹⁵⁷ <https://opensource.org/licenses/alphabetical>

CAPITOLO 3 – Giurisprudenza

Tutto quanto visto fino a questo momento, sebbene interessante, non dimostra (se non in alcuni punti) l'applicabilità in sede giudiziaria della dottrina espressa, con particolare riguardo alle licenze libere e aperte: l'assenza di costi è di per sé una scriminante giuridica alla commissione di un illecito? E, qualora venga commesso un illecito, quale tipo di violazione si configura?

3.1 – Giurisprudenza Open Source e Free Software

I primi casi giudiziari relativi a queste nuove licenze sono ovviamente riscontrabili nel continente americano, a partire dalla seconda metà della prima decade del nuovo millennio.

3.1.1 – Jacobsen c.\ Katzer – il “trenino della discordia” (2004-2010)¹⁵⁸

Robert Jacobsen, professore di fisica, coordinatore del progetto Java Model Railroad Interface (JMRI), cita Matthew Katzer, imprenditore, per aver usato parte del progetto JMRI in un prodotto commerciale, ridistribuendo il programma e rimuovendo il copyright (copyleft) originale. Katzer propone domanda di riconvenzione. Al termine del giudizio di prime cure, viene stabilita una responsabilità contrattuale da parte del Katzer ma non una violazione del Copyright Act, non prevedendo la licenza dei costi. Robert Jacobsen impugna in appello il parziale accoglimento delle sue richieste ottenendo piena ragione dal

¹⁵⁸ https://en.wikipedia.org/wiki/Jacobsen_v._Katzer

secondo giudicante, il quale stabilisce, oltre ad una responsabilità contrattuale, la violazione di norme codicistiche. Determinato l'*an*, in attesa di determinazione del *quantum*, i due protagonisti trovarono un accordo stragiudiziale il 18 febbraio 2010, perfezionatosi con pagamento il successivo 5 marzo. Tutti gli atti del procedimento sono pubblici e consultabili, collegati alla pagina Wikipedia in nota.

A favore del prof. Jacobsen si schierano OSI¹⁵⁹ (dopo la sentenza di prime cure), Creative Commons in numerose circostanze¹⁶⁰, Software Freedom Law Center (SFLC)¹⁶¹ e molti altri.

3.1.2 - Novell c.\ CPTN (2011)

Nel 2011, assieme alla Free Software Foundation¹⁶², OSI¹⁶³ fornisce un prezioso contributo alla battaglia antitrust per una massiccia acquisizione di licenze Novell proprietarie da parte del consorzio CPTN¹⁶⁴ (guidato da Microsoft, Apple, Oracle e altri), portando all'acquisizione di dette licenze secondo lo standard GNU Public License (GPL) v. 2 e Open Invention Network (OIN). L'evento ebbe eco internazionale tanto da essere ribattuto anche dai media italiani¹⁶⁵.

¹⁵⁹ <https://opensource.org/node/360>

¹⁶⁰ <https://creativecommons.org/tag/jacobsen-v-katzer/>

¹⁶¹ <https://www.softwarefreedom.org/news/2009/jun/15/jacobsen-amicus-brief/>

¹⁶² <https://www.fsf.org/blogs/licensing/good-news-about-the-novell-cptn-deal>

¹⁶³ <https://opensource.org/history/CPTN>

¹⁶⁴ <https://www.justice.gov/opa/pr/cptn-holdings-llc-and-novell-inc-change-deal-order-address-department-justices-open-source>

¹⁶⁵ <https://www.punto-informatico.it/novell-e-il-consorzio-dei-brevetti/>

3.1.3 SCO c.\ IBM et altera (2003-2010)

Particolare rilievo, poi, rivestono le vicende legali avviate dall'azienda SCO Group. SCO (Santa Cruz Operations) Group, tra gli anni '80 e '90, acquisì una versione proprietaria Microsoft del sistema operativo Unix (variante XENIX¹⁶⁶) e la rese compatibile con i processori Intel 386¹⁶⁷. Inizialmente, SCO partecipò attivamente e supportò la nascita di Linux, variante libera di Unix, al fine di togliere fette di mercato a Microsoft e Windows, salvo poi accorgersi che questo la portò a perdere essa stessa terreno a favore della propria variante gratuita.

Fu così che, il 7 marzo 2003, SCO citò in giudizio IBM con una richiesta di 1 miliardo di dollari, aumentata poi a 3, per supposte violazioni di licenza in merito al proprio sistema operativo¹⁶⁸.

Nello stesso anno, a partire dal mese di maggio, iniziò ad inviare lettere ad aziende citate da Fortune 1000 e Global 500 diffidandole dall'utilizzo di Linux, quantomeno in assenza di acquisto di una licenza SCO UNIX.

Anche in questa circostanza, il già citato avvocato Moglen, Richard Stallman e Bradley Kuhn, in nome e per conto della FSF, tra il 2003 e il 2004 rilasciarono varie dichiarazioni in cui, in buona sostanza, accusavano SCO di confondere il termine Linux con il termine software libero, disconoscendo il complesso apparato noto con il nome di GNU/Linux¹⁶⁹. Come risultato, il 5 novembre 2003

¹⁶⁶ https://it.wikipedia.org/wiki/Microsoft_XENIX

¹⁶⁷ <https://it.wikipedia.org/wiki/Unix>

¹⁶⁸ <http://www.groklaw.net/staticpages/index.php?page=20031016162215566>

¹⁶⁹ <https://www.gnu.org/philosophy/sco/sco.html>, nella quale alcuni degli articoli sono stati tradotti in italiano

SCO cita in giudizio anche FSF chiedendo la produzione di svariati documenti, molti dei quali riservati.

Questo lungo iter giudiziale e stragiudiziale si concluse il 30 marzo 2010, data in cui la Corte Distrettuale dello Utah rigettò ogni richiesta presentata dalla ricorrente¹⁷⁰.

¹⁷⁰ https://it.wikipedia.org/wiki/Causa_legale_tra_SCO_e_IBM, oppure <http://www.groklaw.net/article.php?story=20100330152829622>

CAPITOLO 4 – CONCLUSIONI

Le idee apportate per un verso dalla FSF e per un altro dalla OSI hanno dato un forte impulso all'odierna società, non solo americana, da molti punti di vista.

Il 23 marzo 2017, la Commissione Europea adotta il **quadro europeo di interoperabilità**¹⁷¹, finalizzato alla *“rimozione delle barriere esistenti per la realizzazione del mercato unico digitale in Europa”* attraverso *“un’integrazione completa e l’automazione, un uso migliore delle fonti di informazione attendibili e la pubblicazione in formato aperto dei dati pubblici, garantendo allo stesso tempo che i record dei cittadini e delle imprese siano trattati nel rispetto delle norme sulla protezione dei dati per accrescere la fiducia”*.

Che tale tematica sia propria del mondo FLOSS lo dimostra l'**esposto** presentato nel dicembre 2020 da Roberto Guido, presidente della Italian Linux Society, **alla Corte dei Conti**, Sezione di Controllo sugli Enti e Sezione Centrale di Controllo sulla Gestione delle Amministrazioni dello Stato, *“in cui si invita l’Autorità ad intervenire nei confronti del suddetto fornitore per ottenere una maggiore integrazione con i formati liberi e aperti raccomandati dall’Agenzia per l’Italia Digitale*¹⁷², *al fine di raggiungere - con una singola azione - una più alta adesione agli standard tecnici e operativi definiti dalle norme”*¹⁷³ perché tale fornitore favoriva l'utilizzo di alcuni software proprietari rispetto ad altri

¹⁷¹ <https://eur-lex.europa.eu/legal-content/IT/TXT/HTML/?uri=CELEX:52017DC0134&from=LV>

¹⁷² https://trasparenza.agid.gov.it/archivio19_regolamenti_0_5385.html

¹⁷³ <https://www.ils.org/2020/12/20/formati-digitali-esposto-corte-conti.html>

(anche FLOSS) attraverso l'imposizione di standard non aperti, richiamando le succitate raccomandazioni AgID e la normativa nazionale.

Questo è solo uno degli esempi che la rivoluzione digitale, attraverso questi e altri illustri pensatori, sta portando nel mondo dei brevetti e del diritto d'autore, i cui esiti finali sono ancora ben lungi dall'essere noti.

SITOGRAFIA

- https://bibliotecadigitale.cab.unipd.it/bd/per_chi_pubblica/archivio/diritto-dautore-e-copyright;
- https://it.wikipedia.org/wiki/Diritto_d%27autore_italiano;
- <https://www.filodiritto.com/tutela-del-software-diritto-dautore-brevetto-e-principio-dellesaurimento>;
- <https://www.siae.it/it/diritto-dautore/>;
- http://eprints.biblio.unitn.it/1788/1/Licenze_software_caso_moscon_05_03_2009.pdf;
- [https://it.wikipedia.org/wiki/Licenza_\(informatica\)](https://it.wikipedia.org/wiki/Licenza_(informatica)).