

L'OpenSource

1. Licenze software: introduzione

Il software è un codice che è trattato dalle leggi sul diritto di autore, in maniera simile a quanto avviene per le opere letterarie. Tradizionalmente, il contratto che regola l'uso del software è la **licenza** ed è sempre importante conoscere i termini di questo accordo per il software con cui si intende avere a che fare.

La legge sul diritto di autore stabilisce già quali sono i diritti di chi produce l'opera e di chi la utilizza, ma nell'ambito del software si è introdotto nella pratica un contratto, non firmato, il cui scopo è spesso quello di limitare ulteriormente i diritti di chi ne fruisce. Questo contratto è la **licenza**.

Il software ha un proprietario (salvo il caso del software di dominio pubblico che viene descritto in seguito), che è tale in quanto «detiene i diritti di autore». Questo proprietario può essere l'autore originale, oppure un altro detentore che ne ha acquisito i diritti in base a un contratto. Il detentore dei diritti di autore è colui che possiede il **copyright**.

L'utilizzo del software può essere concesso gratuitamente o a pagamento, per le operazioni stabilite nel contratto di licenza, o in sua mancanza per quanto stabilito dalla legge. Il pagamento per l'«acquisto» di software, non si riferisce all'acquisizione dei diritti di autore, ma solo delle facoltà stabilite dalla legge, ovvero da quanto indicato nella licenza.

Le licenze software sono documenti legali allegati ai programmi. Senza una tale documento, *un programma non può essere distribuito né modificato senza l'esplicito consenso degli autori*.

All'opposto, una licenza di software libero permette lo studio, la distribuzione, la modifica e la redistribuzione del programma modificato. Questo è possibile in quanto i detentori dei diritti di sfruttamento economico del programma lo consentono, eventualmente a certe condizioni.

2. Software libero

Il concetto di software libero discende naturalmente da quello di libertà di scambio di idee e di informazioni. Negli ambienti scientifici, quest'ultimo principio è tenuto in alta considerazione per la fecondità che ha dimostrato; ad esso infatti è generalmente attribuita molta parte dell'eccezionale ed imprevedibile crescita del sapere negli ultimi tre secoli.

La libertà di scambio di idee non è tuttavia una questione puramente pratica: essa è anche alla base dei concetti di libertà di pensiero e di espressione. Analogamente alle idee, il software è immateriale, e può essere riprodotto e trasmesso facilmente. In modo simile a quanto avviene per le idee, parte essenziale del processo che sostiene la crescita e l'evoluzione del software è la sua libera

diffusione. Ed ogni giorno di più, come le idee, il software permea il tessuto sociale e lo influenza, produce effetti etici, economici, politici e in un senso più generale culturali

Il **software libero** è software che fornisce il permesso per chiunque di utilizzarlo, copiarlo e distribuirlo, in forma originale, o anche dopo averlo modificato, sia gratuitamente che a pagamento. Il software libero può essere tale solo se viene messo a disposizione assieme al codice sorgente, per cui, a questo proposito, qualcuno ha detto: «se non è sorgente, non è software» (*if it's not source, it's not software*).

È importante sottolineare che la «libertà» del software libero non sta tanto nel prezzo, che eventualmente può anche essere richiesto per il servizio di chi ne distribuisce le copie, ma nella possibilità di usarlo senza vincoli, di copiarlo come e quanto si vuole, di poterne distribuire le copie, di poterlo modificare e di poterne distribuire anche le copie modificate.

In alcuni ambienti si preferisce utilizzare la definizione «Open Source» per fare riferimento al software libero nei termini che sono stati descritti, per evitare ambiguità nella lingua inglese. Sotto questo aspetto, nella lingua italiana, come in molte altre lingue, è più appropriato l'uso della definizione «software libero».

Si distinguono quattro punti fondamentali, necessari perché il software possa essere considerato «libero»:

Livello	Descrizione
libertà 0	la libertà di eseguire il programma per qualunque scopo;
libertà 1	la libertà di studiare come funziona il programma e di adattarlo alle proprie esigenze (in tal caso, deve essere disponibile il sorgente);
libertà 2	la libertà di ridistribuire copie del programma;
libertà 3	la libertà di migliorare il programma e di distribuire tali miglioramenti (anche per questo è necessario disporre dei sorgenti).

Un'elegante definizione di licenza libera fu data per la prima volta da Richard Stallman negli anni Ottanta. Una [definizione più dettagliata](#), equivalente alla prima, fu data negli anni Novanta dalla Open Source Initiative.

Una licenza per il software libero permette a chiunque riceva un programma ad essa soggetto di usarlo per qualunque scopo, di copiarlo, modificarlo e ridistribuirlo. Che la distribuzione sia o meno gratuita non è rilevante. Il codice sorgente deve essere accluso o può essere richiesto al prezzo di costo del trasferimento. Se chi riceve il software può ridistribuirlo solo entro gli stessi termini di licenza, si parla di licenza con [permesso d'autore](#) (*copyleft* in inglese); in caso contrario si parla di licenza *senza permesso d'autore* (*non-copyleft*).

Il software che non può essere commercializzato, pur soddisfacendo i punti precedentemente elencati, viene considerato «software semi-libero», come viene chiarito in seguito.

Vale la pena di elencare alcune definizioni riferite al software libero.

- **Software di dominio pubblico.** Il software di dominio pubblico è software senza copyright. Di per sé, questo tipo di software è libero, solo che, mancando chi può difenderlo, qualcuno può riuscire ad accamparvi dei diritti; pertanto, alcune copie, o varianti di questo software

possono non essere più libere.

- **Software protetto da copyleft.** La parola *copyleft* («permesso d'autore») è un'invenzione e vuole rappresentare il copyright di chi, mentre difende il proprio diritto di autore, vuole difendere la libertà della sua opera, imponendo che questa e le sue derivazioni restino libere. In pratica, una licenza appartenente alla categoria «copyleft» impedisce che chi ridistribuisce il software (originale o modificato che sia) possa aggiungere delle restrizioni ulteriori. Il classico esempio di licenza di questo tipo è la licenza pubblica GNU-GPL.
- **Software libero non protetto da copyleft.** Il software libero non è necessariamente di tipo copyleft e ciò accade quando la licenza non vieta espressamente l'aggiunta di restrizioni da parte di chi lo ridistribuisce. Quando si utilizza software di questo tipo, non è possibile generalizzare: occorre accertarsi dei termini del contratto che riguarda la copia particolare della quale si è venuti in possesso.

Software GPL (General Public License). La **GPL** è il primo e più importante esempio di licenza con [permesso d'autore](#). Chi modifichi software con permesso d'autore e ne distribuisca le modifiche è tenuto a farlo secondo gli stessi termini in cui è distribuito il programma originale. Detto altrimenti, bisogna dare agli altri le stesse libertà che si sono ottenute. Più precisamente, la distribuzione di ogni opera derivata da un programma con licenza GPL deve a sua volta essere soggetta alla licenza GPL o ad una licenza che dia le stesse libertà. La GPL si preoccupa di proteggere il buon nome degli autori imponendo che le versioni modificate di un programma siano chiaramente indicate come tali, se distribuite. È comunque possibile effettuare modifiche per uso personale, o uso interno in un'organizzazione, senza sottostare a formalità particolari. In definitiva, se lo scopo della licenza è di assicurarsi che tutte le copie del programma, comprese quelle modificate, siano accompagnate dal codice sorgente o sia comunque facile procurarselo, allora è bene usare la [GNU-GPL](#).

La licenza **GNU-GPL** è l'esempio più importante di licenza che protegge il software libero con il copyleft. Quando si parla di «software GPL» si intende fare riferimento a software protetto con la licenza pubblica **GNU-GPL**. Il [Progetto GNU](#) è stato lanciato nel 1984 per sviluppare un sistema operativo Unix-compatibile completo che fosse [software libero](#): **il sistema GNU**. GNU è un acronimo ricorsivo per “GNU's Not UNIX”; (GNU Non è Unix). Varianti del sistema operativo GNU, che utilizzano il kernel Linux, sono ora ampiamente utilizzate; anche se a questi sistemi ci si riferisce spesso come “Linux”, essi vengono chiamati con più precisione [sistemi GNU/Linux](#).

Quello che segue è un riassunto dei *Termini e condizioni per la copia, distribuzione e modificazione* contenuti nella versione più recente, la 2, della GNU General Public License.

1. La Licenza si applica a qualsiasi programma od altro lavoro che contenga una nota in cui il legittimo possessore del copyright dichiara esplicitamente che può essere distribuito nei termini della GNU General Public License. La Licenza si applica non solo al programma o lavoro, ma anche a qualsiasi altro lavoro che lo contenga o ne contenga una porzione, anche testuale, modificata o tradotta.

La Licenza si riferisce solamente ad attività come il copiare, il distribuire ed il modificare, altre attività sono al di fuori dello scopo della Licenza.

2. È possibile copiare a distribuire con ogni mezzo copie testuali del codice sorgente del programma così come lo si riceve, non senza però aver provveduto a pubblicare in ogni copia una appropriata nota di copyright e di esonero da garanzie. Vanno conservate intatte le note che si riferiscono alla Licenza e all'assenza di garanzia e va consegnata a chi riceve il programma una copia della Licenza.

È possibile richiedere una somma in cambio dell'atto fisico di trasferire una copia, nonchè per l'offerta di una garanzia che però resta una libera iniziativa e non rientra nei termini della Licenza.

3. Si possono modificare intere copie o porzioni del programma cui è stata applicata questa Licenza e distribuire le modifiche o il programma modificato, a patto che siano rispettati i termini di cui al punto 1 e che contestualmente si verifichino le seguenti condizioni:
 1. I file modificati devono contenere delle note che dichiarino le modifiche, l'autore e la data in cui sono avvenute.
 2. Al lavoro che deriva in tutto o in parte da un programma sotto questa Licenza, deve essere applicata la Licenza stessa senza oneri per alcun terzo.
 3. Se possibile il programma modificato deve indicare, all'avvio, una nota di copyright e di esonero da garanzie (a parte quelle prestate personalmente dall'autore e nel rispetto delle leggi vigenti), l'indicazione della possibilità di redistribuire il programma stesso sotto le presenti condizioni nonchè i modi per ottenere una copia della Licenza.

La Licenza non si applica a parti di un lavoro che si possano chiaramente identificare come non derivate dal programma in questione, ma quando tali parti sono distribuite come parte integrante di un lavoro considerato nella sua interezza come derivato dal programma, la Licenza si applica all'intera distribuzione. L'intento in questo caso è di esercitare il diritto di controllo su distribuzioni di lavori che derivano da o contengono programmi sotto questa Licenza.

4. È possibile copiare e distribuire il programma cui è stata applicata questa Licenza, od il lavoro da esso derivato, in codice oggetto o in forma eseguibile nei termini di cui ai punti 1 e 2 se contestualmente si verificano le seguenti condizioni:
 1. Che sia allegato il codice sorgente corrispondente, anch'esso distribuito secondo i termini di cui ai punti 1 e 2; oppure,
 2. Che sia presente un'offerta scritta, valida tre anni, di rilasciare a qualunque terzo al prezzo massimo del costo effettivo il codice sorgente, sempre nei termini di cui ai punti 1 e 2; oppure,
 3. Che sia allegata l'offerta di cui sopra se se ne è stati a propria volta destinatari.
5. Non è possibile compiere azioni quali copiare, modificare, distribuire, in modi che non rispettino quanto dettato da questa Licenza. Chi contravviene ai termini dettati non potrà più godere dei diritti garantiti dalla Licenza. Ovviamente ciò non vale per chi avrà ricevuto in buona fede programmi sotto tale licenza.

6. Nessuno è obbligato ad accettare i termini di questa Licenza finchè non l'ha sottoscritta, tuttavia essa permette azioni quali copiare, modificare e distribuire che altrimenti sarebbero proibite dalla legge.
7. Ogni volta che viene distribuito un programma cui è stata applicata questa Licenza, od il lavoro da esso derivato, il ricevente acquisisce il permesso dal licenziatario originale di copiarlo, modificarlo o distribuirlo secondo i presenti termini e condizioni. Non è possibile apporre successive restrizioni a questi diritti né forzare terze persone ad agire in conformità con questa Licenza.
8. Il fatto che siano poste a qualcuno delle condizioni a causa di decisioni della magistratura o accordi privati, non lo esime dal seguire i termini di questa Licenza. Ciò nell'intento di proteggere il sistema di distribuzione del software libero.
9. Se per problemi di brevetti o copyright la distribuzione del programma sotto questa Licenza subisce restrizioni in alcune nazioni, l'autore deve aggiungere alla Licenza delle clausole che escludano la distribuzione del programma in quelle nazioni.
10. La Free Software Foundation può in ogni momento produrre delle revisioni a questa Licenza che saranno conformi allo spirito della presente versione, ma potranno differire in alcuni particolari per affrontare nuovi problemi o istanze.
11. Chi volesse includere parte di un programma sotto questa Licenza in altri programmi liberi ma sottoposti ad altre condizioni di distribuzione, deve chiedere il permesso all'autore od al possessore del copyright. Nel caso questo sia la Free Software Foundation, essa si impegna a decidere caso per caso seguendo due principi: preservare la libertà di cui gode il proprio software e promuovere la condivisione ed il riuso del software in generale.

Per rendere libero un programma bisogna allegare al codice sorgente un file (tipicamente di nome `LICENSE.TXT`) che contenga il testo della licenza scelta. Inoltre, è consigliabile aggiungere all'inizio di ogni file sorgente una riga di copyright. Ad ogni modifica del file bisogna aggiungere l'anno di modifica alla lista degli anni riportata nella riga di copyright. Se risultasse difficile riportare il carattere ©, questo può essere sostituito dai tre caratteri (C). Dopo la riga di copyright bisogna indicare che i termini della licenza di copyright sono contenuti nell'allegato file `LICENSE.TXT`.

Scrivere ex novo una licenza di copyright per il software è una pratica da evitare. È molto difficile scriverne una nuova, anche per un legale specializzato in questa materia. Inoltre, l'uso di una delle licenze il cui uso è consolidato ha diversi vantaggi:

- gli utilizzatori del software le conoscono già, e ne sanno valutare le implicazioni;
- sono scritte da legali;
- hanno provato la loro validità sul campo;
- facilitano lo scambio di codice fra progetti liberi diversi.

3. Software non libero

Il software non è libero tutte le volte che non sono soddisfatti tutti i requisiti necessari per poterlo essere. È bene ricordare che il prezzo non è un fattore che limita la libertà, mentre altri dettagli sono più importanti. Anche in questo caso, vale la pena di elencare alcune definizioni che in generale riguardano software non libero.

- **Software semi-libero.** Il software semi-libero è software che permette agli individui di usarlo, copiarlo, modificarlo e distribuirlo, anche modificato, per qualunque scopo, escluso quello di trarne profitto. In altri termini, si potrebbe dire che si tratta di software libero a cui è stata aggiunta la limitazione per la quale questo non può essere usato e distribuito per trarne profitto.
- **Software proprietario.** Il software proprietario è quel software che non è né libero, né semi-libero. Di solito, per «avere a che fare» con questo software è necessario ottenere un permesso speciale, che spesso si limita a concedere l'uso su un elaboratore, o su un gruppo ben determinato.
 - **Software freeware.** Il termine *freeware* non è abbinato a una definizione precisa, ma viene inteso generalmente come software gratuito, del quale non viene reso pubblico il codice sorgente, che può essere usato e copiato senza poterlo modificare. In questo senso, il prefisso «free» serve solo a evidenziare la gratuità della cosa, ma non la libertà che invece richiede altri requisiti. Un esempio molto noto è il visualizzatore di documenti PDF.
 - **Software shareware.** Con il termine shareware si fa riferimento a software proprietario che può essere ridistribuito, ma per il quale viene richiesto espressamente il pagamento dopo un periodo di prova. Alcuni di questi programmi smettono di funzionare dopo il periodo di prova o forniscono funzionalità ridotte fino all'acquisto di una chiave che li sblocchi.

4. Software commerciale

In base alle classificazioni viste in questo capitolo, il **software commerciale** è tale solo in quanto viene venduto per profitto. Lo sviluppo e la diffusione del software libero dipendono anche dalla possibilità di venderne delle copie, originali o modificate, per trarne profitto. Pertanto, il software che pur offrendo le quattro libertà fondamentali, non consente la commercializzazione per trarne profitto, non viene considerato «libero» in modo completo. In questo senso, è importante evitare di confondere il software proprietario con il software commerciale, perché non sono la stessa cosa.

Si osservi che può esistere anche del software non-libero, che non è nemmeno commerciale.

5. Alcuni miti da sfatare

Il software libero è gratuito

È falso: la libertà del software non ha nulla a che vedere con il suo prezzo. Benché gran parte del software libero più diffuso sia distribuito gratuitamente, ci sono programmatori che vivono della vendita e della manutenzione dei programmi liberi da loro creati.

Il software gratuito è libero

È falso. Molti programmi proprietari vengono distribuiti gratuitamente.

Il software libero è privo di copyright

È falso. Benché si possa rinunciare al copyright su un proprio programma e renderlo così di pubblico dominio, la gran parte del software libero è distribuito con una licenza. Per esempio, sono licenze di copyright la licenza BSD e la GNU GPL, anche se per qualificare quest'ultima spesso si parla di permesso d'autore (copyleft).

L'introduzione del software libero nella scuola e nella pubblica amministrazione, ma anche nei paesi poveri, ridurrebbe i costi relativi al software

Potrebbe essere vero, ma una seria valutazione dei costi è molto difficile. Qualunque tipo di software, se usato in ambito non domestico, ha dei costi di manutenzione che sono solitamente maggiori del suo prezzo di acquisto. I motivi per sostenere l'uso del software libero, specie in ambiti pubblici, riguardano anzitutto la libertà, non il prezzo.

6. Il movimento OpenSource

Nel 1998 Bruce Perens, Eric Raymond e altre personalità nel campo del software libero si convinsero che i principi di libertà associati ad esso fossero malvisti nel mondo degli affari, a causa della loro carica ideologica. Decisero perciò di evitare accuratamente ogni riferimento a considerazioni politiche o di principio, e di lanciare una campagna di promozione del software libero che ne mettesse in luce i numerosi [vantaggi pratici](#), come la facilità di adattamento, l'affidabilità, la sicurezza, la conformità agli standard, l'indipendenza dai singoli fornitori. A tal fine scrissero la [Open Source Definition](#), il documento fondamentale del movimento [open source](#).

Il movimento open source fu un successo, e contribuì a sdoganare il concetto di software libero in campo aziendale, dove era guardato con sospetto o condiscendenza. Un esempio di questo successo è l'atteggiamento dell'IBM, l'azienda che ha fatto di gran lunga i maggiori investimenti nel campo del software libero, la quale parla esclusivamente di open source, mai di software libero.

La voluta neutralità del movimento open source verso gli aspetti etici e politici del software libero è la caratteristica sostanziale che lo distingue dalla filosofia del software libero, che al contrario pone l'accento sulle motivazioni ideali. Parlare di software libero piuttosto che di open source è una questione politica piuttosto che pratica; i due movimenti concordano infatti sulle licenze considerate accettabili, ed hanno obiettivi e mezzi comuni.

7. Le dinamiche di mercato

L'economia dei paesi avanzati si sta completamente ristrutturando e riorganizzando. La risorsa informazione è il cardine del cambiamento ed è sotto gli occhi di tutti la straordinaria accelerazione che la tecnologia ed il software in quanto strumento hanno impresso a questa evoluzione. Le imprese devono restare al passo o persino precedere il cambiamento per mantenere il vantaggio competitivo acquisito o per conquistarlo. In molti settori, soprattutto dell'area hi-tech, il cliente vuole prodotti demassificati, adattati alle sue specifiche esigenze ed in tempi sempre più brevi. Aziende estremamente flessibili ed in grado di coinvolgere il cliente nella progettazione tendono a lasciarsi alle spalle i concorrenti organizzati in modo tradizionale.

Molte *parole d'ordine* circolano già da alcuni anni: *Just-in-time*, *produzione flessibile*, *ottimizzazione organizzativa*, *qualità totale*, *produzione di massa a misura del cliente*, ecc. Per reggere in questa economia della velocità, le linee di confine tra azienda, fornitore e cliente (singoli o altre aziende) devono essere sfumate e compenetrarsi ed occorre essere in grado di individuare e gestire beni o servizi già esistenti che formeranno il prodotto finale prima ancora che questo venga creato. L'*Azienda Virtuale* di Davidow e Malone [Davidow] congiunge tutte queste caratteristiche. L'aumento della tecnologia informatica ha fatto sì che la forza e la produttività di un'impresa, anche tradizionale, si basino sempre di più sulla risorsa informazione e sulla sua creazione. Internet ha favorito la nascita di un nuovo tipo di impresa che agisce in una realtà virtuale omogenea ed interconnessa, dove la competizione si svolge sul piano della velocità, della qualità e dell'efficienza in una corsa sfrenata alla soddisfazione dei bisogni e delle necessità, anche quelle più velleitarie.

Le imprese che si dedicano all'open source rientrano solo in parte nel modello dell'Azienda Virtuale. Utilizzano gli stessi media e si basano sulla stessa risorsa principale: *l'informazione*; tuttavia differiscono dal modello di Davidow e Malone per le caratteristiche e la stessa natura del prodotto su cui si basano. Il software libero ed open source ed il loro modello di sviluppo, il Progetto Open Source, non obbediscono alle leggi ed ai tempi del mercato. Come vedremo in seguito, i concetti di *scadenza* e di *dovere aziendale* sono in parte estranei ai componenti delle comunità open source. Ciò non vuol dire che su questo tipo di prodotto sia impossibile creare un business, anzi, tuttavia è necessario tenere in considerazione tutti i suoi aspetti per trarre vantaggio dai suoi punti di forza ed evitare di essere danneggiati dai suoi punti deboli. Le Imprese Open Source hanno inoltre una caratteristica in più: si basano su un patto etico che tutte le parti devono rispettare considerando che la *materia prima* è donata all'azienda dalla comunità e che la logica del dono prevede che questo venga ricambiato.

La struttura dei costi e dei ricavi delle aziende fondate sul software libero o comunque open source è completamente diversa rispetto a quelle tradizionali: la soglia di ingresso è estremamente bassa e basso il costo di produzione, ma il bene offerto può essere reperito facilmente e a bassissimi costi o quasi dai possibili clienti.

In realtà ciò che viene offerto sul mercato non sono beni ma servizi, il valore di vendita è indiretto. Ad esempio le distribuzioni Linux commerciali come SuSE, Red Hat, Caldera, ecc., non vendono i singoli bit del sistema operativo ma il valore aggiunto nell'assemblaggio dei diversi software che lo compongono e le garanzie di funzionamento e compatibilità con altri sistemi della stessa marca, oltre all'assistenza gratuita per un certo periodo di tempo ed opzioni per l'estensione dell'assistenza stessa. I distributori Linux, inoltre, sono costretti a competere con modalità che apportano benefici ai consumatori ed al mercato in generale: infatti, la Licenza GPL li obbliga a condividere le migliorie e le nuove caratteristiche via via sviluppate che prontamente vengono inserite nei propri pacchetti dalle altre distribuzioni ed evita le frammentazioni.

Alcune aziende fondate sull'open source vendono solo servizi (supporto, ricerca, sviluppo e formazione) come Linuxcare oppure documentazione, ad esempio i manuali sul software open

source della casa editrice americana O'Reilly & Associates o della giovane ed italianissima Hops Libri.

Un'altra caratteristica dei software open source a vantaggio degli utenti, è che questi potranno essere sempre supportati anche se l'azienda che li ha distribuiti è scomparsa o se i programmatori che li hanno sviluppati all'interno di un'azienda non fanno più parte del suo organico: senza la disponibilità dei sorgenti ed il permesso di modifica ciò non sarebbe possibile. Anche la ripartizione del rischio, quindi, può far optare le aziende e gli utenti per questo tipo di software, non solo la ripartizione dei costi di sviluppo.

Non sempre, però, la scelta open source è la più conveniente per l'impresa. In assenza di alternative funzionali e dell'effetto rete è meglio tenere i propri bit proprietari e scegliere la soluzione commerciale come accade ad esempio per un software innovativo che copre una nicchia di mercato non occupata da altri.

Alla luce di tutto ciò le aziende tradizionali possono avere diversi tipi di adattamento a seconda del posizionamento sul mercato loro e dei loro prodotti. La Microsoft alla comparsa di Linux ha preferito rispondere con tattiche *FUD* (Fear, Uncertainty, Doubt) ovvero tendenti ad alzare i costi di transazione della concorrenza, oggi comincia a dare i primi segni di apertura rispetto ai metodi open in considerazione del cambiamento riscontrato nel mercato. Altre società si sono già *convertite*, distribuendo i propri prodotti completi dei sorgenti e con licenze open source, come è avvenuto per Netscape che nel '98 ha reso disponibili i sorgenti del suo web browser avviando il progetto *Mozilla* e riguadagnando fette di mercato sottratte da Internet Explorer di Microsoft. Lo stesso sta accadendo per Sun Microsystems che ha rilasciato StarOffice^{4.1} completo dei sorgenti e sotto licenza GPL con l'ultima versione.

8. Redditività del software

Un quesito ricorrente che sorge spontaneo nell'approccio al mondo dell'open source è se questo modello, in genere efficace a livello di qualità del software prodotto, lo sia anche a livello economico, se produce dei profitti.

Generalmente si è portati a pensare al modello proprietario come all'unico in grado di garantire un ritorno economico, grazie alle royalties incassate dalle copie vendute, ma questo è vero solo in parte. Non sono effettivamente molte le aziende, in genere molto grandi, che possono basare le loro fortune quasi esclusivamente sulla vendita di copie di software proprietari da loro prodotti. Non sono molte proprio perchè la pratica di mantenere segreti i codici sorgenti dei loro prodotti ha portato ad un progressivo sfoltimento della concorrenza ed all'assunzione di posizioni monopolistiche non sempre corroborate da una effettiva maggiore qualità.

Su questo aspetto, la concorrenza, il software open source può forse offrire una maggiore possibilità di confronto ad armi pari.

Anche se a prima vista ad un addetto ai lavori può non sembrare, il software - in generale - è un prodotto estremamente complesso, soprattutto nella fase dello sviluppo. Dietro la sua realizzazione esistono una lunga serie di costi che giustificano gli ingenti investimenti attuati in genere dalle tradizionali imprese che producono e mantengono software proprietario, costi che vengono ammortizzati in seguito con i ricavi della sua vendita.

Non irrilevante è il problema dei costi per chi tale software lo acquista. L'impresa che intende acquistare un prodotto di supporto alla sua attività deve fare i conti non solo con il suo prezzo d'acquisto, ma anche con tutti i costi futuri collegati a tale decisione

Il prodotto software, a causa della sua complessità, risulta oneroso. Di seguito sono riportati alcuni tra i più significativi costi relativi al software, cui va aggiunto, ovviamente, il *costo d'acquisto*.

- *Pianificazione*. L'attività di pianificazione comporta costi connessi all'utilizzo di risorse, prevalentemente umane, sia interne che esterne. Al lavoro del personale interno che viene dedicato alla valutazione e definizione di quali sono gli effettivi bisogni dell'azienda, viene affiancato in genere il lavoro di consulenti esterni che poi saranno anche i fornitori del software.
- *Adeguamento dei sistemi*. Spesso per poter installare il nuovo software sono necessari ritocchi più o meno significativi ai sistemi esistenti, come ad esempio l'aggiunta di componenti hardware, nuove infrastrutture od altro software.
- *Efficienza operativa*. L'uso di un software, anche dopo la sua installazione, richiede nelle sue varie fasi una serie di risorse, da quelle umane ad aggiornamenti hardware, che comportano dei costi. La scarsa efficienza operativa dei processi connessi ad un software sono dunque fonte di un deciso innalzamento dei costi; si pensi ad esempio alla presenza di errori nel software o ad una incompatibilità con gli standard usati per i dati o addirittura una scarsa intuitività o eccessiva complessità dell'interfaccia utente.
- *Training*. L'introduzione di nuovo software in una azienda spesso porta una certa spesa per l'aggiornamento o l'insegnamento ex-novo delle conoscenze necessarie agli utenti - impiegati - per usarlo. Ciò significa corsi e/o manuali d'uso.
- *Servizi*. Comprendono la manutenzione del software installato da parte dell'azienda fornitrice e l'intervento in casi d'emergenza come crash dei sistemi o malfunzionamento. Di solito tali servizi fanno parte del "pacchetto" completo acquistato con il software, ma sono comunque a pagamento.
- *Aggiornamenti del software*. In genere anche il software più testato contiene degli errori o comunque necessita di ritocchi dovuti alla sua normale evoluzione tecnica. Sono necessari dunque aggiornamenti, frequenti o meno, per poter disporre di un prodotto adeguato e funzionante e, normalmente, si pagano a parte.
- *Costi incidentali*. Il software installato opera in un ambiente complesso come quello aziendale e spesso il suo uso porta a dei costi non direttamente imputabili ad esso. Tali costi in genere hanno una loro giustificazione: il fattore umano. Ecco che allora una svista o un uso improprio del software possono portare all'accidentale cancellazione di dati anche importanti; o una errata valutazione soggettiva può scatenare delle conseguenze non imputabili al software in se, si pensi ad un macchinario che, nell'eseguire correttamente i movimenti dettati dal software che lo governa, provoca un danno materiale dovuto ad un suo errato posizionamento.

Un'importante osservazione sulla recente evoluzione del mercato del software è che il business si sta spostando radicalmente dalla vendita dei programmi alla vendita dei *servizi correlati*.

Ciò non vuol dire che ci sia una drastica diminuzione della vendita di copie di software proprietario, si è ancora lontani da questo, ma i *costi correlati* per chi lo acquista stanno diventando sempre più onerosi, assumendo un'importanza sempre maggiore. Ciò porta a constatare la possibilità per le aziende di evitare di caricare il cliente del costo per copia del software venduto, nel momento in cui vengono generati abbastanza profitti dai soli servizi correlati.

Il margine di profitto per copia, tolta la quota attribuita dei costi sostenuti per lo sviluppo, può essere distribuito sul ricavo derivante dai servizi, che in genere sono forniti durante un arco temporale anche pluriennale.

Se consideriamo inoltre che le aziende che utilizzano software libero o comunque open source devono sopportare costi di sviluppo molto minori in virtù di quanto già detto nei precedenti paragrafi, capiamo il potenziale vantaggio competitivo a livello di prezzo di cui possono godere.

9. L'Open Source come business

Visto da vicino non c'è niente di strano o magico nel metodo di sviluppo open source dal punto di vista economico e non dovrebbe essere additato come impraticabile.

Abbiamo visto che non c'è un solo modello di sviluppo ben definito da seguire pedissequamente. Il fenomeno open source, inteso come fenomeno generale comprendente tutte le sue varie anime, è semplicemente un nuovo modo di sviluppare, distribuire e licenziare il software. Alle imprese che comprendono i fattori economici, culturali e, volendo, politici che risiedono in una effettiva strategia open source, questo modello offre l'aspettativa di un business vitale con le forti premesse di un futuro caratterizzato da una domanda crescente.

Saranno rispettate le aspettative? Ci sono realmente tali premesse? Funziona davvero questo modello nel "freddo" mondo del business o è destinato a restare relegato tra le accoglienti e protettive braccia della comunità? Prima di analizzare dei casi reali di Impresa Open Source, vediamo più nel dettaglio quali sono gli elementi che la caratterizzano.

Perfino Richard M. Stallman, il fondatore della Free Software Foundation nel suo *GNU Manifesto*^{4.2} non ha mai sostenuto che lo sviluppo del software non si dovesse pagare o comunque fosse un'attività non-profit. Semmai proponeva e propone l'abbandono del modello for-profit basato sul trattare il software come proprietà intellettuale, in favore di un modello di business che vedesse l'attività di sviluppo come una attività professionale.

Molte grosse imprese del settore hanno cominciato già da alcuni anni, e con un trend crescente, ad usare software libero ed open source come base per i propri prodotti commerciali e, in alcuni casi, hanno contribuito allo sviluppo di questo tipo di software attraverso donazioni in denaro alla Free Software Foundation o ai singoli progetti, hardware o tempo dei propri impiegati.

Eventi come il recente rilascio da parte di Netscape del codice sorgente del suo web browser *Communicator* hanno attirato l'attenzione generale sull'importanza del software libero ed open source sia per gli utenti che per i produttori, grazie in parte anche all'entrata in scena del concetto di *open source* che ha contribuito a smorzare la diffidenza nata dalle incomprensioni sul concetto di *free software*^{4.3}.

Quali sono i motivi oggettivi, a parte le convinzioni personali sulla bontà del modello, che possono far decidere di abbracciare l'open source ad una impresa commerciale? Delle aree o necessità aziendali che possono trarre giovamento dall'adozione di una strategia open source ben applicata, tra le più significative si devono annoverare le seguenti [Hecker]:

- *Evoluzione del prodotto.* Il modello di sviluppo open source ben si adatta ad un'impresa che vuole ed ha bisogno di creare nuovi prodotti o migliorare gli esistenti. I costi connessi a

queste attività, indispensabili nel mondo del software, potrebbero subire una drastica riduzione.

- *Qualità del prodotto.* Si collega al punto precedente l'osservazione della maggior qualità, già dalle prime versioni, di un software sviluppato correttamente con il modello open source che garantisce, tra l'altro, tempi di debugging minori.
- *Mantenimento del prodotto.* Un software sviluppato come Progetto Open Source può godere dell'appoggio della comunità dei suoi utilizzatori che, anche se piccola, può contribuire a tenerlo aggiornato.
- *Reclutamento e fidelizzazione dei collaboratori.* La comunità open source è ricca di individui dalle notevoli capacità tecniche e può fungere, con le dovute garanzie del rispetto degli ideali *free*, da capace serbatoio di giovani talenti o comunque di onesti tecnici.

Queste caratteristiche non solo possono giovare al prodotto in termini di qualità e funzionalità, ma possono anche aumentarne il valore, che è poi quello che i consumatori decidono di pagare se lo riconoscono.

Il codice sorgente è visto oggi da molte delle imprese che producono software come qualcosa di simile ad una risorsa preziosa da proteggere ad ogni costo con licenze restrittive o altri mezzi^{4.4} per impedirne la sottrazione per usi scorretti o quantomeno per non dare una mano alla concorrenza.

Probabilmente, il problema del possesso esclusivo del codice sorgente va ridimensionato in quanto, almeno in alcuni ambiti, irrilevante. Il codice sorgente acquisisce valore commerciale solo se lo si può utilizzare per creare prodotti e/o servizi da vendere, ma ciò si verifica con successo in genere nel caso di imprese dal marchio sufficientemente conosciuto e dalla buona posizione di mercato - come per esempio Microsoft.

Il fatto che un'impresa rilasci, sotto adeguata licenza, il codice sorgente di un suo prodotto, non vuol necessariamente dire che i concorrenti lo useranno con lo stesso successo o che influenzeranno negativamente i suoi affari. Il successo di un prodotto software è il risultato di molto più delle sole caratteristiche del codice, ma comprende fattori come, tra gli altri, la fiducia e la reputazione derivanti dalla qualità dei servizi forniti e l'efficacia dell'area commerciale e dei canali distributivi.

La disponibilità del codice sorgente di un prodotto software può aumentare direttamente il suo valore per i clienti a cui è diretto aiutandoli a risolvere una serie di problemi:

- I clienti possono trarvi una sorta di protezione del loro investimento nel caso chi ha loro fornito il software decida di non continuarne la produzione e l'assistenza o smetta semplicemente di esistere. In questo caso la disponibilità del codice sorgente del software permette di correggerlo e svilupparlo adeguandolo alle situazioni contingenti o di farlo fare ad altri.
- I clienti hanno la possibilità di capire meglio come funziona il software nel caso la documentazione fornita dal venditore risulti carente.
- I clienti possono apportare o fare apportare al software le correzioni necessarie ai propri bisogni e che il venditore non è in grado o non intende eseguire.
- I clienti possono adattare il software a nuovi sistemi operativi o nuovi hardware non supportati dal venditore.
- I clienti possono creare versioni personalizzate del software o versioni rinnovate di esso. Possono anche creare intere nuove applicazioni utilizzando parte del codice in loro possesso, evitando di svilupparle ex-novo.

Ma la stessa disponibilità del codice sorgente può anche aumentare indirettamente il valore del software per i clienti:

- Chi compone manuali e chi insegna l'uso del software in questione può creare una documentazione più completa se per esempio quella fornita dal venditore risulta carente. E tale documentazione può essere di aiuto ai clienti stessi.
- Consulenti ed integratori di sistema possono acquisire familiarità con la tecnologia e le tecniche su cui è stato sviluppato il software divenendo esperti nell'implementare sistemi che si basino su di esso o che lo incorporino. Questo permette di creare una base di persone che possiedono le abilità per sviluppare applicazioni complesse basate sul software d'origine. Ciò significa che i clienti potranno avere la possibilità di sviluppare sistemi meno costosi o più funzionali sfruttando la competizione che si viene a creare tra chi ha acquisito le abilità necessarie.
- Terze persone interessate allo sviluppo del software in questione possono, in via più o meno autonoma, trovare degli errori nel codice o delle inconsistenze nella sicurezza, creando delle correzioni. Le correzioni possono in seguito essere inserite nel software originale migliorandolo ed aumentandone il valore per i clienti, che si trovano con un prodotto che genera meno costi di mantenimento.
- Altri produttori di software possono riutilizzare il codice del software in questione per creare modificazioni o aggiunte di caratteristiche non previste dal prodotto originale. Ciò incrementa certamente il valore del software per i clienti.

Il Prodotto Open source può consistere in varie tipologie:

- Software prodotto da un'impresa che decide di renderne pubblico il codice sorgente secondo una licenza libera od open source.
- Software open source prodotto da un'impresa che riutilizza il codice sorgente reso disponibile da un'altra azienda.
- Software prodotto da un Progetto Open Source classico e riutilizzato da un'impresa per un proprio prodotto.
- Software prodotto da un Progetto Open Source promosso e coordinato da un'impresa anziché da singoli individui.

In linea di massima il Progetto Open Source promosso o sostenuto in via principale da un'impresa che intenda utilizzarlo come prodotto e come base per i suoi servizi, non si discosta molto dalle linee generali di un Progetto Open Source tipico. La differenza sta nel fatto che è appunto l'impresa che lo fa partire o lo sostiene rendendo disponibile il codice sorgente del software di cui vuole stimolare lo sviluppo sulle basi open source. Se tale software risulta interessante e offre buone prospettive di sviluppo, e l'impresa offre le dovute garanzie di rispettare i principi della comunità, non avrà difficoltà a trovare membri della comunità che affiancheranno il lavoro degli sviluppatori interni. È da notare inoltre che ciò offre l'opportunità all'impresa di reperire, all'occorrenza, buoni collaboratori da inserire nel suo organico.

10. Impatto pratico del software libero

La [rilevanza economica](#) del software libero è ancora molto ridotta, ma è in fortissima crescita ormai da alcuni anni, e tutto consente di supporre che tale crescita [continui nel prossimo futuro](#), anche grazie ai [vantaggi tecnici ed economici](#) del software libero.

Ad oggi, il software libero è ampiamente diffuso in ambito accademico, industriale e fra gli appassionati di calcolatori, soprattutto grazie ai sistemi GNU/Linux. Questi sistemi liberi sono disponibili a costi molto bassi, 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.

In ambito accademico viene molto apprezzata la possibilità di personalizzare ogni parte del sistema, visto che i programmi liberi sono liberamente modificabili (libertà numero uno). In ambito industriale, si apprezza l'affidabilità dei sistemi liberi, dovuta al fatto che quando un utente corregge un errore in un programma solitamente rende disponibile la correzione agli altri utenti (libertà numero tre). Gli appassionati di calcolatori apprezzano lo spirito di condivisione esistente fra gli utenti di software libero.

Ma le implicazioni dell'uso del software libero non sono soltanto tecniche ed economiche, perché il software da tempo ormai è avviato ad occupare un ruolo di primo piano nella nostra vita quotidiana, ed è destinato a cambiare in maniera profonda la società.

È per queste ragioni che la nostra libertà futura dipenderà anche dalla capacità di ognuno di noi di controllare il software. È per queste ragioni che ai tradizionali principi di libertà sessuale, di culto, di movimento, di espressione deve essere affiancata la libertà del software. È per queste ragioni che la nostra libertà futura dipenderà anche dall'uso di software libero.