Appunti sul Software Libero

Storia

L'idea di software libero nasce agli inizi degli anni ottanta, quando lo sviluppo del software cominciò a passare di mano dalle università alle aziende (software proprietario), ponendo un pesante freno alla collaborazione che caratterizzava il lavoro di gran parte dei programmatori e dei sistemisti dell'epoca, soprattutto con i patti di non divulgazione che le aziende facevano firmare ai programmatori che assumevano.

Dal 1950 fino ai primi anni del 1970, era tipico per gli utenti di computer utilizzare software liberi associati a software gratuiti. Quindi il software "commerciale" esisteva da sempre, ma i costi elevati dell'hardware facevano sì che il business delle aziende non fosse concentrato sul software, che era considerato una parte naturale del prodotto, ed i cui codici sorgenti erano in genere pubblici. Per esempio venivano formate organizzazioni di utenti e fornitori per facilitare lo scambio di software. Inoltre poiché i software erano spesso scritti in un linguaggio ad alto livello, il codice sorgente veniva distribuito in riviste di computer (come Creative Computing, Softside, Compute!, Byte, ecc) e libri, come il bestseller BASIC Computer Games.

Dal 1970 la situazione cambiò: il software diventava sempre più complesso e difficile da realizzare e le aziende iniziarono a non distribuire i codici sorgenti e ad obbligare i propri dipendenti a non rivelare nulla per non avvantaggiare la concorrenza; inoltre con il crollo dei costi dell'hardware, lo sviluppo commerciale del software divenne un business notevole ed il codice sorgente divenne sempre più un investimento prezioso che poteva da un lato far acquisire una fetta di tale mercato in rapida crescita e dall'altro legare i propri utenti al proprio software mantenendo il segreto sui metodi utilizzati per lo sviluppo di sistemi e applicazioni.

L'industria del software iniziò così ad utilizzare misure tecniche (come ad esempio la distribuzione di copie di programmi per computer solo in binario) per impedire agli utenti di computer di essere in grado di studiare e adattare il software come meglio credevano.

Nel 1980, la legge sul copyright venne estesa anche ai programmi per elaboratore. In questo modo le aziende cominciarono ad utilizzare la legge sul diritto d'autore per impedire ai concorrenti di leggere e modificare i loro prodotti, assicurandosi il controllo dei propri clienti che, senza più poter vedere e modificare il codice sorgente del software, non potevano più adattarlo alle loro esigenze, ma dovevano chiedere alle aziende di farlo per loro.

Nel 1983 Richard Stallman, uno degli autori originali del popolare programma Emacs e membro di lunga data della comunità hacker presso il laboratorio di intelligenza artificiale del MIT, fondò il progetto GNU con l'intenzione di creare GNU: un sistema operativo completamente libero. Grazie alla collaborazione di molti sviluppatori volontari, all'uso di Internet per la coordinazione del progetto e al kernel Linux di Linus Torvalds, nel 1991 nacque GNU/Linux, un clone di Unix liberamente utilizzabile, modificabile e ridistribuibile.

Nella sua dichiarazione iniziale del progetto e del suo scopo, Stallman ha espressamente citato, come motivazione, la sua opposizione alla richiesta di accettare i vari accordi di non divulgazione e le licenze restrittive dei software che vietano la libera condivisione di software-sviluppo potenzialmente redditizio, un divieto che è direttamente in contrasto con la tradizionale etica hacker.

« I started the free software movement to replace user-controlling non-free software with freedom-respecting free software. With free software, we can at least control what software does in our own computers. »

« Ho avviato il movimento del software libero per rimpiazzare il software non libero che controlla l'utente con software libero rispettoso della libertà. Con il software libero, possiamo almeno avere il controllo su quel che il software fa nei nostri computer. »

Lo sviluppo dei software per il sistema operativo GNU è iniziata nel gennaio 1984, e la Free Software Foundation (FSF) è stata fondata nel mese di ottobre 1985. Ha sviluppato una definizione di software libero e il concetto di "copyleft", progettata per garantire la libertà software per tutti.

FREE SOFTWARE FOUNDATION

Il kernel Linux, iniziato da Linus Torvalds, è stato rilasciato come codice sorgente liberamente modificabile nel 1991. La prima licenza è una licenza di software proprietario. Tuttavia, con la versione 0.12, nel febbraio 1992, pubblicarono il progetto sotto la GNU General Public License. Essendo molto simile a Unix, kernel di Torvalds ha attirato l'attenzione di programmatori volontari.

Caratteristiche

La parola *libero* non implica la possibilità di utilizzare tale software in maniera indiscriminata: il software libero è comunque soggetto ad una *licenza d'uso*, a differenza ad esempio del software di pubblico dominio.

Esso si contrappone quindi al software proprietario ed è differente dalla concezione open source, incentrandosi sulla libertà dell'utente e non solo sull'apertura del codice sorgente, che è comunque un pre-requisito del software libero.

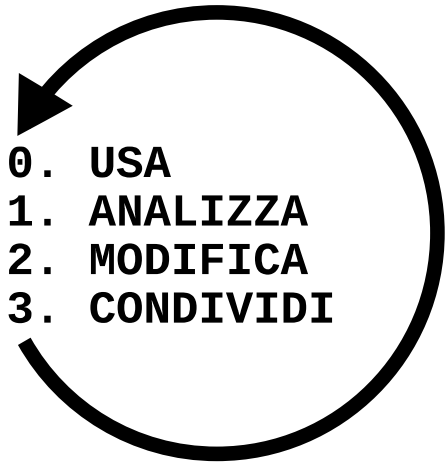
Rispetto al software proprietario, la licenza d'uso del software libero permette le "quattro libertà", ponendo in genere i seguenti vincoli:

- gli autori precedenti del software devono essere menzionati anche nelle versioni modificate, lasciando intatto il loro copyright;
- in seguito ad una modifica, non è possibile applicare una licenza d'uso incompatibile con la licenza originaria o che vada contro le norme della licenza stessa. Per esempio chiunque può riemettere del software pubblicato sotto LGPL usando la licenza GPL (tale operazione è anche chiamata *upgrade* della licenza), mentre non è possibile fare il contrario (naturalmente se non si è il detentore unico del copyright);
- normalmente, nella licenza, vi è una clausola che sancisce la non usabilità del software se non si rispetta la licenza d'uso o se una o più norme della stessa licenza non sono valide per termini di legge;
- quando si distribuisce un codice binario occorre o distribuire insieme anche i sorgenti o garantire per iscritto la possibilità a tutti gli utenti di venirne in possesso dietro richiesta ed al solo costo del supporto

Denominazione

La FSF consiglia di utilizzare il termine "software libero", piuttosto che "software open source", perché, come essa afferma in un documento sulla filosofia del software libero, quest'ultimo termine e la campagna di marketing associato ad esso si concentra sulle questioni tecniche di sviluppo del software, evitando la questione della libertà degli utenti. La FSF rileva inoltre che "Open Source" ha esattamente un significato specifico nell'inglese comune, vale a dire che "è

possibile guardare il codice sorgente." Stallman afferma tuttavia che il termine "software libero" può portare a due interpretazioni differenti, una delle quali è coerente con la definizione FSF di Software libero, così da esservi almeno qualche possibilità di essere inteso correttamente, a differenza del termine "Open Source".[1] Stallman ha anche affermato che, considerare i vantaggi pratici del software libero equivale a valutare i vantaggi pratici del non essere ammanettato, nel senso che non è necessario per un individuo considerare delle ragioni pratiche al fine di rendersi conto che essere ammanettato limita la propria libertà.[2] "Libre" è un termine spesso usato per evitare l'ambiguità dell'aggettivo "free" in lingua inglese e l'ambiguità con l'uso precedente del "software libero", come software di pubblico dominio.



Le "quattro libertà"

Secondo la Free Software Foundation, un software si può definire *libero* solo se garantisce quattro "libertà fondamentali"[4]:

- Libertà 0: Libertà di eseguire il programma per qualsiasi scopo. La libertà di usare un programma significa libertà per qualsiasi tipo di persona od organizzazione di utilizzarlo su qualsiasi tipo di sistema informatico, per qualsiasi tipo di attività e senza dover successivamente comunicare con lo sviluppatore o con qualche altra entità specifica. Quello che conta per questa libertà è lo scopo dell'*utente*, non dello *sviluppatore*; come utenti potete eseguire il programma per i vostri scopi; se lo ridistribuite a qualcun altro, egli è libero di eseguirlo per i propri scopi, ma non potete imporgli i vostri scopi.[5]
- Libertà 1: Libertà di studiare il programma e modificarlo.[6] L'accessibilità al codice sorgente è una condizione necessaria per il software libero, altrimenti non avrebbero senso neanche la libertà 0 e la 2.
- Libertà 2: Libertà di ridistribuire copie del programma in modo da aiutare il prossimo.
- Libertà 3: Libertà di migliorare il programma e di distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio. Questa libertà comprende quella di usare e rilasciare le versioni modificate come software libero. Una licenza libera può anche permettere altri modi di distribuzione; insomma, non c'è l'obbligo che si tratti di una licenza con copyleft. Tuttavia, una licenza che imponesse che le versioni modificate non siano libere non si può categorizzare come licenza libera.

Un programma è software libero se l'utente ha tutte queste libertà. In particolare, se è libero di ridistribuire copie, con o senza modifiche, gratis o addebitando delle spese di distribuzione a chiunque ed ovunque. Essere liberi di fare queste cose significa (tra l'altro) che non bisogna chiedere o pagare nessun permesso.

Richard Stallman aveva l'obiettivo di diffondere la libertà e la cooperazione, incoraggiando la diffusione del software libero in sostituzione al software proprietario:

« Rendo disponibile il mio codice affinché venga usato nel software libero, e non nel software proprietario, con lo scopo di incoraggiare chi programma a fare altrettanto. Ho capito che, poiché gli sviluppatori di software proprietario usano il diritto d'autore per impedirci di condividere il software, noi che cooperiamo possiamo usare il diritto d'autore per favorire coloro che come noi cooperano: possono usare il nostro codice. »
(Richard Stallman[7])

Licenze d'uso libere

La maggior parte del software libero è distribuito con queste licenze:

- GNU General Public License (GNU GPL)
- GNU Lesser General Public License (GNU LGPL)
- Licenze BSD (licenza originale, licenza modificata e licenza semplificata)
- Mozilla Public License
- Licenza MIT/Expat
- Licenza Apache
- Eclipse Public License (EPL)
- European Union Public Licence (EUPL)

Buona parte del software libero viene distribuito con la licenza GNU GPL, scritta da Richard Stallman e Eben Moglen per garantire legalmente a tutti gli utenti le quattro libertà fondamentali. Dal punto di vista dello sviluppo software, la licenza GPL viene considerata una delle più restrittive, poiché impone che necessariamente ogni prodotto software derivato - ovvero, che modifica o usa codice sotto GPL - venga a sua volta distribuito con la stessa licenza. Anche MediaWiki, il software usato per Wikipedia, è distribuito con licenza GPL.

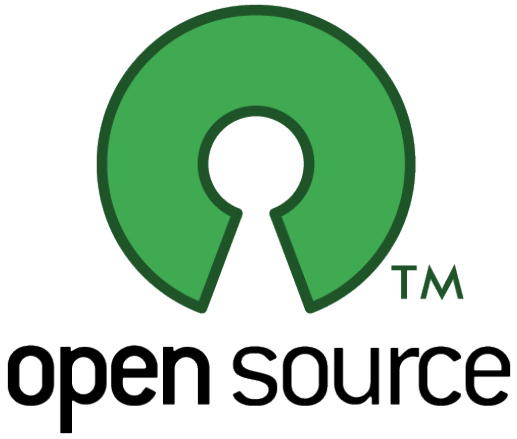
La GNU LGPL, simile ma meno restrittiva rispetto alla precedente, permette di utilizzare il codice anche in software proprietario, purché le parti coperte da LGPL - anche se modificate - vengano comunque distribuite sotto la medesima licenza. In genere è utilizzata per librerie software. Non tutte le licenze ritenute libere sono compatibili tra di loro, cioè in alcuni casi non è possibile prendere due sorgenti con due licenze libere ed unirle per ottenere un prodotto unico. Questo avviene quando non esista e non sia possibile creare una licenza che possa soddisfare i requisiti delle licenze originali.

Ad esempio la licenza BSD originale, pur essendo considerata licenza di software libero, è incompatibile con la GPL; per ovviare al problema è stato necessario creare una "licenza BSD modificata" compatibile con la GPL. Un'altra licenza degna di nota è la licenza Apache, stilata dalla Apache Software Foundation; la versione 2 di questa licenza è compatibile con la GPL versione 3 ma non con la GPL versione 2. L'Apache License considera un prodotto derivato alla stregua della LGPL, ma è più liberale nella concessione delle proprietà intellettuali.

Le varie licenze libere possono contenere ulteriori limitazioni per alcune situazioni particolari; per esempio la GPL prevede che si possa esplicitamente vietare l'uso del software nelle nazioni dove tale licenza non è valida o dove dei brevetti software impediscono la distribuzione di tale

software. Le licenze d'uso non vietano in genere di vendere software libero e non limitano il loro prezzo di vendita.

Differenze rispetto all'open source



Il termine open source indica criteri leggermente più deboli di quelli previsti per il software libero. Per quanto ne sappiamo, tutto il software libero esistente è anche open source. E anche quasi tutto il software open source che è stato rilasciato sotto forma di codice sorgente è anche software libero, ma ci sono eccezioni. Innanzitutto, alcune licenze open source sono troppo restrittive (ad esempio: “Open Watcom” non è libero perché la sua licenza non permette di realizzare una versione modificata e usarla in privato) e non si possono considerare libere, ma tali licenze sono poco usate.

Essendo la disponibilità del codice sorgente uno dei requisiti fondamentali che accomuna il software libero ed il software open source, spesso si è indotti a considerare i due concetti equivalenti, ma in realtà non lo sono. Un software è open source se i termini secondo i quali viene distribuito rispondono alla Open Source Definition dell'Open Source Initiative (OSI): in particolare, se una licenza rientra in tale definizione, allora tale licenza può essere dichiarata licenza open source.

La definizione potrebbe cambiare nel tempo (nessuno garantisce che questo non possa accadere) e quindi è possibile che una licenza attualmente open source non lo sia nel futuro o viceversa. OSI è anche l'organizzazione che su richiesta certifica con il relativo marchio registrato il fatto che una licenza sia effettivamente aderente alla Open Source Definition. Recentemente l'OSI ha posto un freno al proliferare delle licenze dichiarando che cercherà di limitare il numero di licenze che nel futuro saranno ritenute licenze open source. Questo potrebbe, in linea teorica, far sì che una licenza ritenuta libera non venga ritenuta open source.

Una licenza invece è libera (o meglio, una *versione* di una licenza è libera) solo se rispetta le quattro libertà fondamentali. Pertanto se una versione di una licenza è libera, allora lo sarà per sempre. Naturalmente è sempre complesso, almeno per un cittadino "normale" (non esperto di leggi), stabilire se una licenza è libera o meno perché entrano in gioco i termini legali utilizzati nella stessa. Il progetto GNU si occupa tra l'altro anche di indicare se una licenza è libera o meno e se è compatibile con le licenze GNU o meno.[12]

Il software libero inoltre non deve essere confuso con il software freeware, che è distribuibile gratuitamente, ma che non è né software libero né open source: il software libero infatti non è detto che sia gratuito ovvero può anche essere a pagamento e il termine Inglese *free* va inteso in Italiano appunto come *libero*, nel senso dei principi suddetti, e non *gratuito*. In ogni caso, gli insiemi di applicativi designati da software libero e open source coincidono a meno di poche eccezioni. La differenza fondamentale è nel tipo di approccio: parlando di software libero si pone l'accento sugli aspetti sociologici ed etici, che sono volutamente rimossi nella visione open source.

Regole sulla produzione dei pacchetti e sulla distribuzione

Vengono accettate regole su come pacchettizzare una versione modificata, purché non limitino in modo significativo la libertà di distribuire versioni modificate, o di produrre versioni modificate per uso interno. Quindi è accettabile, ad esempio, che la licenza vi obblighi a cambiare il nome della versione modificata, togliere un logo, etc. Regole che richiedano di distribuire il codice sorgente agli utenti delle versioni che avete pubblicamente distribuito sono accettabili.

Una questione particolare è quando una licenza richiede di cambiare il nome con cui il programma sarà chiamato da altri programmi. Questo impedisce di rilasciare la versione modificata in modo che possa sostituire l'originale quando chiamata dagli altri programmi. Questo tipo di richiesta è accettabile solo se c'è una funzionalità di "aliasing", cioè una funzionalità che permetta di specificare il nome originario del programma come alias della versione modificata.[13]

Obblighi degli utenti del software libero

Le licenze di software libero impongono una serie di obblighi a chi distribuisce il software in versione originale o modificata:

- rendere disponibile il software anche in formato sorgente (per es., la GPL e la MPL);
- includere informazioni sull'installazione del software (per es., la GPL e la EPL);
- se si modifica il software, rendere disponibile anche la versione originale (per es. la MPL e la GPL);
- non imporre all'utente ulteriori obblighi che limitino l'ulteriore distribuzione del software (per es. la GPL e la MPL);
- manlevare i contributori del software da eventuali danni conseguenti alla distribuzione di prodotti che includono il software stesso (per es., la EPL).

Aspetti filosofici e sociali

Nel software libero il significato della parola *libero* ha un'accezione particolare. Si è già sottolineato che la *libertà* del software libero non è incondizionata, perché è soggetta ai precisi vincoli della licenza d'uso, come qualsiasi altra licenza d'uso, solo che in questo caso l'autore si "espropria" di alcuni diritti per cederli agli utenti. Questi vincoli sono studiati in maniera tale da favorire il tipo di libertà cosiddetta copyleft, un metodo generico per rendere un programma (o altro lavoro) libero ed imporre che tutte le modifiche e versioni estese del programma siano anch'esse software libero, e hanno come obiettivo la *condivisione del sapere*.

Pertanto il software libero parte da considerazione sociali e per molti aspetti è una forma di filosofia.

Le implicazioni sociali del software libero sono notevoli. La condivisione del sapere non permette a un gruppo ristretto di persone di sfruttare la conoscenza (in questo caso

tecnologica) per acquisire una posizione di potere. Inoltre, è promossa la cooperazione delle persone, che tendono naturalmente ad organizzarsi in *comunità*, cioè in gruppi animati da un interesse comune.

Il modello del software libero si è naturalmente esteso ad altri campi del sapere. Chi crede nel modello copyleft pensa che questo possa essere applicato ad esempio alla musica o alla divulgazione. L'esempio più riuscito di applicazione di questo modello ad un campo differente dal software è oggi Wikipedia, che promuove la condivisione del sapere e la formazione di una comunità.

Differenti correnti di pensiero

Tra i sostenitori del software libero, e più in generale del copyleft, vi sono diverse correnti di pensiero, che spaziano da una visione radicale ad una più moderata. La visione più radicale tende ad un modello che si spinge molto oltre a quello del software libero, arrivando in alcuni casi ad auspicare una completa abolizione del software proprietario, considerato una limitazione inaccettabile della libertà e dei diritti umani. Questa ideologia è stata, erroneamente o almeno impropriamente, paragonata a correnti politiche quali il comunismo, sebbene solitamente i sostenitori del software libero non entrino in questioni politiche.

Chi è su posizioni più moderate considera il software libero un ideale a cui tendere, non negando la possibilità di esistere al software proprietario e più in generale allo sfruttamento commerciale del diritto d'autore, sfruttamento che può essere fatto anche usando software libero, come dimostrano vari casi di successo (es: MySQL).

La licenza LGPL è stata concepita per permettere una certa integrazione tra software libero e software non libero. C'è chi ritiene inopportuno un suo utilizzo perché permette l'integrazione, sotto determinate condizioni, di software libero da parte di software non libero;[15] tuttavia ogni autore di software può decidere liberamente che licenza scegliere e quindi sotto quali condizioni permettere l'uso del proprio lavoro.

Diffusione in altri campi

L'approccio della libertà del software e della collaborazione si è diffusa in altri ambiti produttivi come quella della creazione di contenuti. Un esempio è la definizione di opera culturale libera, che definisce i progetti a contenuto libero, come Wikipedia oppure pubblicazioni open data / open access, termini utilizzati rispettivamente per dati e per pubblicazione di articoli accademici fruibili liberamente.

Confronto fra software libero e proprietario

Sicurezza e affidabilità

C'è un grande dibattito riguardo alla sicurezza del software libero rispetto ad un software proprietario che gode di sicurezza tramite segretezza. Spesso software libero viene considerato sinonimo di poca sicurezza poiché, essendo il codice sorgente accessibile e migliorabile da tutti, è più facile trovare bug e punti deboli. Secondo Richard Stallman però l'accesso degli utenti al codice sorgente rende più difficile il rilascio di un software con bug oppure spyware rispetto ad un software proprietario. In merito a questo OpenBSD ha iniziato una campagna contro i blob binari all'interno dei kernel ovvero driver proprietari di hardware le cui case produttrici non hanno rilasciato il codice sorgente. Non essendo noti i Codici sorgente possono contenere bug

minando così la stabilità e la sicurezza del sistema. In merito a ciò è stata sviluppata in Irlanda gNewSense ovvero una distribuzione di Linux completamente priva di blob binari. Il progetto venne subito accolto e supportato dalla Free Software Foundation che stimolò la nascita del kernel Linux-libre.

Vantaggi del software libero

A prescindere dalle implicazioni sociali, secondo i suoi sostenitori il software libero presenta numerosi vantaggi rispetto al software proprietario.[16]

- Essendo possibile modificare liberamente il software, è possibile personalizzarlo ed adattarlo alle proprie esigenze.
- Il codice sorgente è sottoposto ad una revisione da parte di moltissime persone, pertanto è più difficile che contenga bug e malfunzionamenti. In ogni caso, è sempre possibile per chiunque tenere un indice pubblico dei problemi, in modo che gli utenti li conoscano.
- Se viene scoperto un baco o una falla di sicurezza, la sua correzione di solito è molto rapida.
- Essendo il sorgente liberamente consultabile, è molto difficile inserire intenzionalmente nel software backdoor, trojan o spyware senza che questi vengano prontamente scoperti ed eliminati, come invece è accaduto per alcune applicazioni commerciali (ad esempio il caso del database Firebird della Borland che conteneva una backdoor scoperta quando di tale software sono stati pubblicati i sorgenti).
- Non potendo esistere formati segreti, è molto più facile costruire software interoperabile, anche se resta la necessità di documentazione ecc. perché un formato possa dirsi un formato aperto.
- Permettere a chiunque di modificare i sorgenti garantisce che ogni nuova funzionalità o copertura di un bug possa essere proposta da chiunque e immediatamente applicata dagli sviluppatori. Questo permette di avere rapidamente a disposizione un software che rispetta le esigenze di chi ha richiesto le modifiche in caso di necessità.[17]
- Il software libero consente la collaborazione di molteplici enti (pubblici o privati) per lo sviluppo di una soluzione che soddisfi un'ampia gamma di esigenze.
- La complessità e le dimensioni di alcune applicazioni di software libero (ad esempio, dei sistemi operativi) è tale che è necessario il supporto commerciale di un'azienda; il software libero si presta a creare nuove opportunità di business nel campo della formazione e del supporto, oltre che della eventuale personalizzazione del software.
- Collaborando con sviluppatori volontari e utilizzando il lavoro della comunità, anche le piccole e medie imprese sono in grado di sviluppare e vendere prodotti di alta qualità, senza dover ampliare il loro organico.

Critiche al software libero

Secondo alcuni il software libero avrebbe delle limitazioni e degli svantaggi rispetto al software proprietario, specialmente nei casi in cui sia frutto di un lavoro volontario:

- lo sviluppo del software libero sarebbe più lento rispetto al software proprietario; tesi espressa da Bill Gates nella sua lettera aperta ai programmatori dilettanti[18]. Bill Gates ha inoltre particolarmente criticato la GPL definita come licenza "virale" e non economicamente sostenibile[19].
- alcune tipologie di software, soprattutto di nicchia, non sarebbero disponibili come software libero; infatti il software di nicchia non avrebbe abbastanza utenti per la creazione di una comunità che supporti lo sviluppo del software.

- lo sviluppo del software libero avrebbe una struttura anarchica, che porta a risultati incoerenti e ad una mancanza di uniformità e consistenza.
- nonostante il codice sorgente sia liberamente disponibile, non tutti sono in grado di apportarvi modifiche.

Impiego nella pubblica amministrazione

È largamente riconosciuto che per la pubblica amministrazione è preferibile usare software libero, a meno di specifici fattori che lo rendano impossibile o estremamente costoso per la specifica applicazione.[22][23] Questo consenso si riflette, soprattutto dagli anni 2000, in numerose norme quali il Codice dell'amministrazione digitale in Italia, le linee guida dell'Unione europea e centinaia di iniziative pubbliche internazionali, di cui un numero crescente (soprattutto in America latina) rende obbligatorio l'uso di software libero.[24]

Una sintesi dell'esperienza di 20 enti migrati al software libero ha mostrato che il software libero ha costi totali inferiori rispetto al software proprietario, posto che la migrazione sia ben pianificata e consideri tutti i costi del ciclo di vita del software (TCO comprensivo di ricerca, acquisizione, integrazione, uso e uscita).[25]

Per incrementare la diffusione del software libero all'interno della pubblica amministrazione, una strategia comune è che un ente centrale acquisti un servizio di assistenza per tutti gli enti periferici, garantendo così un elevato livello di qualità del servizio a un costo minimo.[26]

Il settore pubblico può contribuire alla sicurezza del software libero per tutti i suoi utenti, come ha fatto l'Unione europea a partire dal 2014.[27]

In Italia

In Italia, per effettuare la valutazione di possibilità o impossibilità, la Pubblica Amministrazione deve effettuare la "valutazione comparativa" prevista dall'art. 68 comma 1 bis[28]. Quest'ultima si basa su criteri nel contesto dell'acquisizione di software. I criteri sono, oltre al costo complessivo del programma o soluzione (costo di acquisto, di implementazione, di mantenimento e supporto, costo di uscita), il livello di utilizzo di formati di dati e di interfacce di tipo aperto nonché di standard in grado di assicurare l'interoperabilità e la cooperazione applicativa tra i diversi sistemi informatici della pubblica amministrazione e, infine, le garanzie del fornitore in materia di livelli di sicurezza, conformità alla normativa in materia di protezione dei dati personali, livelli di servizio tenuto conto della tipologia di software acquisito.

Applicazioni commerciali

Il software libero non deve necessariamente essere sviluppato a titolo gratuito o a fondo perduto. Purché si rispettino i vincoli della licenza d'uso, è possibile vendere software libero; all'interno dei documenti del progetto GNU, Stallman incoraggia la vendita di software libero. Stando alla GPL, però, il primo che compra un software libero ha il diritto di redistribuirlo gratis, è quello che succede ad esempio con REHL, CentOS, Suse, Canonical, ma semmai su servizi e assistenza. Il modello di business è quindi basato sul lavoro e non su licenze parassitarie.

Vi sono inoltre alcune aziende che adottano il modello di sviluppo del software libero per i propri prodotti commerciali. Il ritorno economico in questo caso può derivare dalla fornitura di assistenza e di know-how. Un caso diverso è quello di alcuni esempi di software che vengono pubblicati con un sistema di "licenze multiple". In pratica lo stesso software viene licenziato sia come proprietario, sia come software libero. La versione libera talvolta dispone di meno funzionalità, o è limitata ad un numero ristretto di piattaforme. Esempi celebri di software a doppia licenza sono il database MySQL, di cui esiste una versione "Pro Certified Server" a pagamento e una versione "Community Edition" pubblicata con licenza GPL, e la libreria Qt[29].

Vi sono poi aziende che sono strutturate integralmente per la vendita e l'assistenza di un determinato software libero: esempi classici sono alcune distribuzioni di GNU/Linux, come Red Hat o SUSE. Queste aziende utilizzano come base il software sviluppato dalla comunità, aggiungendo una serie di tool di configurazione o sviluppo, curando gli aspetti più tecnici e dando agli utenti finali un'assistenza mirata. Sfruttando le caratteristiche della licenza BSD, alcune aziende preferiscono invece partire da software libero per sviluppare un prodotto non libero. Per esempio il sistema operativo proprietario Microsoft Windows implementava, fino alla versione NT 4.0, lo stack di rete utilizzando codice sotto licenza BSD.