

# La sfida dell'Open Source

Roberto "Jimmi" Franceschini

27 Novembre 2004

## Sommario

La filosofia Open Source ripropone un modello di sviluppo che supera il campo informatico: chi perde e chi guadagna da questa innovazione

## Indice

<b>1</b>	<b>Un nuovo modello di sviluppo</b>	<b>1</b>
<b>2</b>	<b>Un futuro che viene da lontano</b>	<b>3</b>
<b>3</b>	<b>Pro e contro del software aperto</b>	<b>4</b>
3.1	Generale . . . . .	4
3.2	Utenti . . . . .	5
3.3	Aziende . . . . .	5
<b>4</b>	<b>Software Libero: un'opportunità da non perdere</b>	<b>6</b>

## 1 Un nuovo modello di sviluppo

Cosa accomuna il sistema operativo GNU/Linux a OpenBSD, al server web Apache, ai linguaggi di programmazione Perl e Java?

Sono tutti software distribuiti con licenze Open Source, licenze cioè che garantiscono all'utilizzatore quattro libertà fondamentali:

- Libertà di eseguire il programma per qualunque scopo, senza vincoli sul suo utilizzo
- Libertà di studiare il funzionamento del programma, e di adattarlo alle proprie esigenze

- Libertà di redistribuire copie del programma
- Libertà di migliorare il programma e di distribuirne i miglioramenti

Sempre più spesso si sente definire Linux ed il software “Open Source” come nuovo modello economico. In un’articolo apparso sull’Economist nel giugno di quest’anno si parla dell’Open Source come l’iniziatore di un post-capitalismo che cambierà il modo di fare business. [1]

La filosofia “copyleft” che stà alla base di questo modo di pensare e creare il software libero, anche se descritta come anti-economica e accusata di essere addirittura un cancro che distrugge ricchezza, viene poi adottata con profitto sia da utilizzatori istituzionali come le amministrazioni di Francia, Germania, Brasile, Malesia, Israele, Messico, Perù, Cina, sia da grosse aziende produttrici e distributrici di software ed hardware come Sun, Dell, HP, oltre alle aziende che con l’Open Source sono nate, come RedHat, Mandrake, Suse e molte altre.

Si puo’ arrivare al paradosso che uno dei piu’ aggressivi monopolisti, IBM, sia oggi uno dei maggiori sponsor del software aperto, associandolo al suo hardware ed ai servizi correlati. [2]

Ma cosa rende questo modo di fare software un modello economico originale? Vi elencherò di seguito alcuni spunti di riflessione:

- Progetti Open Source ed organizzazioni non-profit utilizzano spesso gli stessi metodi manageriali delle grosse società a fini di lucro, ma non fanno del danaro il fulcro dei loro piani. La loro attenzione è focalizzata sull’efficacia del prodotto anziché sull’efficienza finanziaria, senza perdere di vista l’obiettivo economico, dato che avendo meno risorse non possono certo permettersi di sprecarle.[3]
- Lavorando per un progetto Open Source un programmatore riceve una gratificazione maggiore che lavorando per un software proprietario in quanto potrà concentrare la sua attenzione sull’oggetto del suo lavoro, il software appunto, senza essere assillato da problemi di marketing o opportunità finanziaria che poco hanno a che vedere con il prodotto.
- Per i motivi precedenti il modello “open” permette di ottenere non solo prodotti migliori in sé, ma anche in rapporto all’ambiente circostante. Avremo quindi software magari dalle prestazioni meno fantasmagoriche ma che interagiscono tra di loro, che si adattano all’utente, così come portando questo modello all’esterno potremo avere prodotti magari meno performanti, ma più attenti all’impatto sociale ed ambientale.

- Per gli attuali “signori” dell’economia molto spesso un buon prodotto non è anche la miglior scelta di marketing perchè si satura il mercato anzichè creare bisogno. Grosse aziende sono cresciute vendendo antivirus e software di protezione necessario a causa di software mal scritto, così come grosse case farmaceutiche prosperano vendendo farmaci resi necessari a causa di strutture sanitarie fatiscenti. È più “efficiente” la soluzione delle emergenze anzichè la gestione di sistemi complessi.[3]
- Nel modello “chiuso” le conoscenze e le idee vengono segregate all’interno delle aziende impedendone la libera circolazione per mezzo di leggi sui brevetti e sui copyright sempre più restrittive e che estendono la “proprietà intellettuale” per periodi di tempo sempre più lunghi. Questo rallenta la ricerca in molti campi e frena lo sviluppo di intere aree geografiche. Ad esempio uno degli effetti collaterali del Digital Millennium Copyright Act è stata la defezione da parte di molti studiosi da ricerche riguardanti la sicurezza dei software proprietari, rendendo nei fatti questi software meno sicuri.[4]
- Per quanto detto sopra il modello di sviluppo “aperto” contrasta il potere dei grossi monopoli oggi esistenti e permette l’ingresso nel mercato di un numero maggiore di entità favorendo la concorrenza. Il ministro francese della funzione pubblica Renaud Dutreil, annunciando l’intenzione di adottare software open source, ha dichiarato “. . . non stiamo iniziando una guerra contro Microsoft . . .” ma la stessa “. . . deve tornare ad essere un fornitore tra gli altri”. [2]

## 2 Un futuro che viene da lontano

In realtà il modello di sviluppo adottato nel progetto GNU/Linux, anche se può apparire rivoluzionario, non è per niente nuovo all’ambiente del software e della ricerca in generale. Fino alla fine degli anni settanta era anzi il modo normale di procedere. Si era agli inizi dello sviluppo di quello che sarebbe poi diventato il personal computer e le aziende di informatica facevano soldi soprattutto vendendo l’hardware. Il software aveva la funzione di accessorio funzionale ed in molti casi era pensato e scritto dagli stessi utilizzatori, per lo più scienziati e ricercatori. In quell’ambiente era naturale condividere la conoscenza, ed il software, considerato pura conoscenza strumentale, veniva liberamente diffuso e scambiato tra i colleghi. [2]

L’intelligenza di Bill Gates fu di capire che il software poteva diventare oggetto di consumo di massa per il grande pubblico. Una delle basi dello sfruttamento economico è il principio di esclusione: un bene può essere fonte

di profitto se è possibile interdire il suo utilizzo. Microsoft ed altre aziende cominciarono quindi a creare attorno al software quel complesso sistema di licenze che oggi conosciamo. Il futuro prossimo di questo sviluppo sembra siano licenze “a scadenza”, che ci permetteranno di utilizzare il software che acquistiamo per un periodo limitato di tempo.

In contrapposizione a questi processi di chiusura nacque negli anni ottanta il movimento del Free Software per opera di Richard Stallman. Le idee filosofiche si tradussero in pratica con la redazione della licenza d’uso *GPL* (General Public License) e la nascita del progetto *GNU*. Nei primi anni novanta la gran parte del software del progetto poteva dirsi completa, ma si era ancora in attesa di un kernel molto avanzato che la *FSF* stava sviluppando. Se questo fosse stato un progetto di tipo proprietario sarebbe quasi certamente fallito e nessuno avrebbe potuto utilizzare il lavoro compiuto fin lì. Ma era un progetto a sviluppo aperto, e Linus Torvald fu libero di costruire il suo personale, anche se meno avanzato kernel e di adottare i programmi GNU già scritti per ridistribuire un sistema usabile sotto licenza GPL. [5]

### 3 Pro e contro del software aperto

Ma in soldoni, cosa rende il software aperto vantaggioso per le aziende e per l’utente comune?

#### 3.1 Generale

Un dato di fatto è, come abbiamo visto sopra, che il software sviluppato con sistema aperto si è dimostrato qualitativamente migliore.

Altrettanto sicuro è che la sua diffusione ha avuto un effetto benefico sul mercato, abbassando i prezzi e correggendo situazioni di monopolio create negli anni.

Gli sviluppatori di software open source sono sempre stati molto sensibili all’utilizzo di standard aperti e comuni, per cui spesso è possibile trasmettere in modo “trasparente” dati ad utenti che utilizzano applicativi diversi, cosa che i software proprietari invece tendono ad evitare per legare l’utente al loro software.

Un’altro possibile vantaggio del software open source è la portabilità del codice su diverse piattaforme hardware, che di solito è più semplice rispetto a software proprietari.

Il fatto che il codice del software libero sia conosciuto a tutti viene proposto da alcuni come una diminuzione della sicurezza dell’applicativo. Questo ragionamento non tiene conto del fatto che la sicurezza di un programma non

viene garantita dal codice di programmazione ma dall'algoritmo utilizzato. Sarebbe come dire che le serrature sono insicure perchè tutti ne conoscono il principio di funzionamento, la sicurezza in realtà è garantita dal metodo di costruzione della chiave. [6]

Una promessa parzialmente mancata del software libero è invece il mito della sua leggerezza e parsimonia nell'uso delle risorse. Le distribuzioni più all'avanguardia hanno fame di risorse come e più di distribuzioni commerciali a pari livello. Se è pur vero che l'ambiente ha caratteristiche di scalabilità che non hanno paragoni in ambito proprietario, si deve comunque ricordare a quei newbie che pensano di riutilizzare il loro vecchio 486 si possono scordare le ultime features tecnologiche così come un pronto supporto.

### **3.2 Utenti**

Un vantaggio per l'utente comune e' dato dal basso od inesistente costo iniziale del software, nonché dalla facilità con cui si può ottenere supporto attraverso la vasta documentazione e la disponibilità delle comunità di sviluppatori ed utilizzatori.

Fino a poco tempo fa' utilizzando software aperti gli utenti comuni si trovavano di fronte ad interfacce grafiche abbastanza instabili e configurabili solo con mezzi piuttosto "ostici". Molto è cambiato negli ultimi anni e le interfacce attuali poco o nulla hanno da invidiare ai software più noti.

L'utente comune, abituato ad una interfaccia standard, si trova un pò spaesato di fronte alla quantità di soluzioni differenti che ha a disposizione, e la frustrazione per la mancata comprensione dei mezzi offerti può rivelarsi un ostacolo reale al loro utilizzo. D'altra parte questa molteplicità di offerte pecca di mancanza di omogeneità anche tra software simili, per cui anche le migrazioni risultano sempre in qualche modo "dolorose".

Un grosso ostacolo che il software libero incontra è la mancanza di disponibilità delle specifiche di molte periferiche di utilizzo comune o il ritardo con cui queste vengono rilasciate. Questo rende molto difficile, e in alcuni casi impossibile, l'utilizzo di schede grafiche, modem, stampanti, macchine fotografiche, palmari ed altro limitando, l'utilizzo di software liberi per chi da queste periferiche dipende.

### **3.3 Aziende**

Per le aziende che devono costruire un sistema ex-novo, se e' vero che i costi iniziali sono per i sistemi aperti piu' bassi, non e' sempre vero lo siano anche i costi di mantenimento. Numerosi sono gli studi effettuati per valutare il TCO (Total Cost of Ownership) di software liberi rispetto a quelli proprietari, ma i

risultati sono discordanti e, guarda caso, quasi sempre favorevoli a chi questi studi ha commissionato.

Le aziende che possiedono già un sistema funzionante devono invece confrontarsi con un costo di migrazione non trascurabile, considerando anche che le “curve di apprendimento” sono in molti casi frenate da ostacoli sia materiali che psicologici.

Qualche tempo fa un’azienda che avesse voluto adottare software aperto si trovava di fronte ad una scelta radicale “o di qui, o di là”. Oggi, anche grazie agli investimenti di grosse società di hardware e di software, sono disponibili un numero maggiore di soluzioni intermedie, per cui viene permesso all’utente un cambiamento più dolce, e soprattutto senza rischiare di trovarsi abbandonato in mezzo al guado.

Molti produttori di software paaventano scenari apocalittici con la scomparsa di posti di lavoro e distruzione di ricchezza causata dalla concorrenza “sleale” di prodotti gratuiti o con prezzi insostenibili per le aziende “serie”. Queste società in realtà non vogliono sostenere i costi di un adeguamento del loro sistema produttivo richiesto dai cambiamenti del mercato, con i quali presto o tardi dovranno fare i conti. Io ci vedo molte analogie con le industrie nazionali dell’auto e dell’acciaio.

Il software libero garantisce una continuità dell’esistenza di applicativi che possono costituire una risorsa strategica per l’azienda privata o pubblica. Qualora un produttore di software proprietario decidesse di ritirare dal mercato un applicativo perchè non più remunerativo o per altri motivi nessuno potrebbe più contare su aggiornamenti sul software stesso e gli utenti sarebbero forzati a cambiare, con possibili spese di transizione. Nel caso di software open source l’utente è libero di proseguire nello sviluppo dell’applicativo con altri mezzi.

In alcuni casi soluzioni software “chiave” per una fascia di utilizzatori, ad esempio software di manipolazione grafica o legati a particolari tipi di hardware, sono disponibili solo tramite software proprietari, e la sostituzione con alternative open source può risultare molto difficile od impossibile, anche a causa delle leggi sui brevetti o sulla proprietà intellettuale (copyright).

## **4 Software Libero: un’opportunità da non perdere**

Nel citato articolo dell’Economist il giornalista, pur dichiarando che “il modello Open Source non rimpiazzerà il capitalismo” ammette comunque che “la collaborazione tra larghi gruppi di persone che lavorano senza un compenso

per un fine comune, che la si chiami Open Source od in qualsiasi altro modo, può essere una forza potente di bene, e dunque deve essere benvenuta”. [1]

Che lo si guardi da un punto di vista tecnologico, ideologico o semplicemente secondo buonsenso il modello proposto da Richard Stallmann, Linus Torvald, Eric Raymond, Bruce Perens e molti altri è , come a me piace definirlo, un buon modo di fare le cose ed ha prodotto una quantità di effetti benefici non solo in ambito software.

Questo modello viene oggi invece osteggiato da un sistema di produzione che, ancorché si dichiari liberista, appare invece rigidamente ideologizzato e monopolizzato da poche grosse aziende decise a mantenere la loro posizione di privilegio. E le armi maggiormente utilizzate sono le leggi sui brevetti e sui diritti intellettuali. Tali leggi, un tempo create per difendere il singolo “genio” dallo strapotere delle grosse aziende, vengono oggi utilizzate per bloccare la competizione, e nei fatti creano barriere alla conoscenza ed allo sviluppo.

E si arriva al paradosso che gli investimenti maggiori siano orientati alla ricerca di mezzi con cui limitare la capacità del mezzo informatico di duplicare informazioni e renderle disponibili a costi bassissimi a miliardi di persone, o che la flessibilità, che è la migliore caratteristica del software, sia oggi sconosciuta a molti utenti in quanto le soluzioni offerte dai software proprietari offrono una rigida interfaccia esterna. La maggior parte degli utenti è vittima del proprio software e non se ne accorge nemmeno.[7]

Non è comunque mia intenzione creare ideologie di segno opposto, e chiunque volesse fare del modello open source una bandiera socio-politica renderebbe davvero un cattivo servizio alla comunità del software libero. Secondo me, e penso che il mio pensiero sia condiviso da quanti oggi offriranno il loro contributo per questo LinuxDay, il modello di sviluppo condiviso che ci viene proposto è una buona opportunità per lo sviluppo economico e sociale che deve potersi sviluppare senza discriminazioni o preconcetti. Spero che qualcuno fra di voi possa oggi cogliere questa occasione per conoscere meglio ed eventualmente adottare questi “bei pezzi di software”

## Riferimenti bibliografici

- [1] “Open Source” Economist, 10.06.2004
- [2] Stefano Chiarini, “Mondo Open Source” Il Manifesto, 30.06.2004
- [3] Pier Giorgio Perotto “Il Paradosso dell’Economia” Franco Angeli, 1993

- [4] “Unintended Consequences: Five Years under the DMCA”  
Electronic Frontier Foundation,  
[http://www.eff.org/IP/DMCA/unintended\\_consequences.php](http://www.eff.org/IP/DMCA/unintended_consequences.php)
- [5] Robert J. Chassell, “ Un’economia del software libero: vantaggi e pericoli” <http://www.softwarelibero.it/altri/economia-sl.shtml>,  
17.05.2001
- [6] Angelo Raffaele Meo, “Software Libero e OpenSource” Mondo digitale <http://www.osservatoriotecnologico.net/software/opensource.htm>,  
Giugno 2002
- [7] Alessandro Rubini, “Software libero e sostenibilità commerciale”  
<http://www.gnu.org/philosophy/software-libre-commercial-viability.it.html>, 12.11.1998

Ringrazio inoltre Ermanno “Cantharsis” per il suo contributo su “Integrazione desktop e enterprise di Linux”