

# Delirio commerciale e ideologia Open Source

di Luca Benazzi

*In che modo possiamo valutare e scegliere tra soluzioni proprietarie e soluzioni Open Source, seguendo un approccio pragmatico piuttosto che ideologico? Il concetto di Open Source nasce nel 1983, quando Richard Stallman, un abile programmatore con idee fuori dall'ordinario, annunciò di voler riscrivere in proprio una versione del famoso sistema operativo Unix che fosse interamente gratuita e i cui codici sorgenti fossero a disposizione di chiunque. La novità dell'approccio di Stallman consisteva nel fatto di voler rilasciare non solo i codici binari eseguibili, e leggibili solo dai computer, ma anche il codice di programmazione, in formato comprensibile agli esseri umani. Questo significava dare la possibilità ad altri di interpretare il codice apportando delle modifiche, cosa che di fatto avvenne quando uno studente di informatica finlandese, Linus Torvalds, adattò ai personal computer ciò che Stallman aveva creato per le realtà aziendali; nasceva così Linux, un sistema operativo gratuito il cui codice sorgente poteva a sua volta essere letto e modificato da altri.*

Luca Benazzi (www.lucabenazzi.eu) vive a Londra e lavora da libero professionista come progettista di interfacce. Ha seguito progetti in vari settori e in particolare per eBay, PayPal, T-Mobile, Unilever, Dubai Holding, Royal Mail. Si occupa di varie cose tra cui scrittura, arte, fisica e cosmologia, scienze naturali, linguaggi, comunicazione, interazione uomo-computer, web semantico, cucina.



Richard Stallman, l'iniziatore del movimento Open Source

## Open Source come condivisione della conoscenza

Stallman si fece portavoce di un modo nuovo di concepire il software come patrimonio collettivo, al pari di tante altre forme di creazione dell'intelletto umano. Un approccio che raccolse negli anni a seguire un ampissimo consenso e la partecipazione di migliaia di programmatori, uniti nell'intento comune di fornire all'umanità porzioni di software libero dal controllo delle aziende.

Il concetto del tutto nuovo era quello di offrire la possibilità di interpretare, studiare e se necessario modificare il codice sorgente, creando di fatto uno straordinario esempio di conoscenza condivisa e gratuita che aveva precedenti in altri settori dell'arte e della cultura, ma non nell'informatica<sup>1</sup>. Vi è, in questo approccio, una differenza sostanziale rispetto al software proprietario, che nasce da un'istanza di (iper)protezione della conoscenza, al fine di salvaguardare l'azienda produttrice. Il software proprietario viene rilasciato nel solo formato binario necessario per essere eseguito da una macchina, ma è completamente privo di informazioni comprensibili a un essere umano e, dunque, impossibile da interpretare e modificare. Volendo fare un paragone molto semplicistico ma efficace, mettere a disposizione il codice sorgente è un po' come offrire una pietanza che richiede una complessa preparazione insieme alla ricetta per prepararla, lasciando quindi agli altri la possibilità di preparare la stessa ricetta in altre occasioni, o di arricchirla con nuovi ingredienti.

<sup>1</sup> L'essere gratuito (o freeware) non è in realtà un requisito necessario dell'Open Source: un software 'open' può essere rilasciato a pagamento purché il codice sorgente sia reso disponibile. Esistono molte licenze che prevedono diversi gradi di libertà in termini di modifica dell'originale, utilizzo, diffusione. Non ci addentreremo nelle differenze tra le varie licenze, che sono piuttosto complesse e richiedono una certa conoscenza della terminologia giuridica. Ci accontenteremo di una definizione semplificata, secondo cui il requisito di un software Open Source è che il codice sorgente venga reso pubblicamente disponibile per essere studiato, modificato e ridistribuito. Per una definizione completa, si veda: <http://www.opensource.org/docs/osd>



## L'avanzata del modello Open Source e di quello proprietario

Il modello Open Source ebbe negli anni Ottanta e Novanta un successo sorprendente, e il lavoro collettivo di sviluppatori sparsi in tutto il mondo produsse prodotti altamente complessi, talvolta in grado di rivaleggiare ad armi pari con le alternative proprietarie.



Il successo dell'Open

Source fu una risposta all'abuso di posizione dominante da parte delle aziende

Il pinguino simbolo di Linux, il primo sistema operativo per personal computer interamente Open Source

produttrici di software, Microsoft in testa, e al disagio creato a milioni di utenti con prodotti che portavano con sé tanti indubbi benefici ma anche tanti difetti, causando perdita di tempo, di dati e situazioni di frustrazione anche profonda, che tutti noi conosciamo. Per anni la mancanza di alternativa fece sì che le maggiori software house approfittassero della loro situazione di monopolio quasi assoluto per incrementare i propri guadagni, senza doversi necessariamente fare carico delle problematiche generate da una cattiva progettazione o dall'impossibilità di poterle migliorare i prodotti.

La nascita e il successo del movimento Open Source fu indubbiamente un segnale di allarme per le multinazionali del software. Ciò nonostante, le problematiche sono ancor oggi ben lontane dall'essersi risolte e abbiamo ancora pochissime aziende che influenzano, attraverso le loro soluzioni 'chiuse', lo stile di vita di milioni di persone, imponendo agli utenti modelli monolitici e inflessibili, esercitando un potere al di là di ogni ragionevolezza. A differenza di tanti altri beni di consumo, la scelta di prodotti informatici si restringe purtroppo a poche o pochissime soluzioni: prendere o lasciare.

## I problemi dei prodotti proprietari

A differenza di molti altri prodotti di consumo, come per esempio le macchine fotografiche, le automobili o i vestiti, i prodotti informatici prevedono il passaggio di dati da un utente all'altro, da una macchina all'altra, da un ufficio all'altro. Una volta che un determinato software si diffonde, con esso si trasmettono anche formati di file a cui altre applicazioni simili non saranno in grado di accedere, oltre a standard, abitudini mentali degli utenti e via dicendo... si creano cioè una serie di vincoli che rendono di fatto impossibile cambiare direzione: si tratta di un circolo vizioso e, nei casi più estremi, per esempio nel settore dei sistemi operativi

(Windows, Apple), assistiamo al paradosso di un'umanità intera alle dipendenze di un manipolo di manager riuniti in un ufficio da qualche parte del mondo<sup>2</sup>.

Possiamo dunque affermare con sicurezza che se da una parte i prodotti proprietari offrono alle persone una serie di indubbi vantaggi pratici, dall'altra costringono a dolorosi compromessi e nascondono un certo numero di insidie nascoste e non dichiarate. Per esempio, il fatto che molti di tali prodotti consentono di salvare i dati solo in un formato proprietario. Il futuro di qualsiasi documento salvato in tale formato, poniamo per esempio un documento di Word, dipende dalle strategie dell'azienda produttrice. Quando il formato di un documento è proprietario, il più delle volte è necessario possedere il software distribuito dalla casa madre per accedervi, e nessun'altra software house può produrre, senza autorizzazione, un applicativo compatibile in grado di accedere a tali dati.

Questo è solo un esempio tra i tanti possibili; le problematiche sono numerose e varie, tutte legate al meccanismo di controllo che un'azienda esercita sulla propria utenza. Il problema è accentuato dalle leggi sui brevetti, che costituiscono un ulteriore freno all'innovazione, in quanto rendono difficile creare soluzioni simili a ciò che già esiste ma in certa misura migliorative, pena il rischio di trovarsi coinvolti in battaglie legali che possono talvolta concludersi con il fallimento di un'azienda.

Quando una casa produttrice di software lancia un prodotto di successo, può decidere di adottare pratiche di concorrenza sleale, vincolando gli utenti alle proprie scelte commerciali. I casi più estremi vengono oggi puniti dalla legge, ma di fatto le aziende hanno il potere di impadronirsi dei dati che le persone hanno creato con il loro tempo e la loro fatica, sia che si tratti delle mail custodite in una casella di posta elettronica, sia che si tratti di un account VoIp, un documento di testo, un filmato, un'immagine...<sup>3</sup>

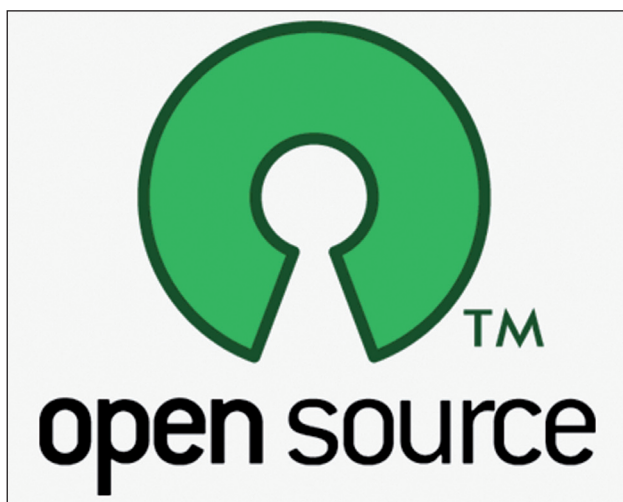
## Allora l'Open Source è meglio?

La risposta richiede cautela. Se il fine ultimo è quello di rendere un servizio alle persone, allora dobbiamo domandarci se la disponibilità di codice sorgente è veramente il fattore chiave. A un esame attento, dobbiamo riconoscere che certamente non è l'unico parametro da tenere e in considerazione, e che dunque l'Open Source non può essere considerato in linea di principio migliore delle soluzioni proprietarie.

Vediamo, senza la pretesa di essere esaustivi, alcuni dei parametri che influenzano la qualità di un prodotto informatico per quanto riguarda la capacità di garantire alle persone risposte affidabili, di qualità, con risparmio di tempo e rispetto delle libertà individuali (di condivisione dei contenuti, di salvaguardia dei propri dati personali, ecc.).

<sup>2</sup> Può sembrare un'esagerazione, ma di fatto questo è il modello che tende oggi a prevalere nella società capitalista.

<sup>3</sup> Per semplicità stiamo solo considerando il settore dei software, ma è chiaro che un discorso analogo riguarda gli applicativi su web. Per esempio, se accedo a un sito internet che mi consente di salvare i miei dati personali (es. la fotografia di un profilo online), i miei dati vengono salvati su un server e nessuno mi garantisce che un domani potrò ancora accedere a quei dati o esportarli.



### Semplicità di utilizzo

In alcuni dei numeri precedenti abbiamo ampiamente discusso della semplicità di utilizzo dei prodotti informatici. Abbiamo visto come la cosiddetta *usabilità* sia un parametro chiave da cui dipende la soddisfazione o la frustrazione degli utilizzatori finali. Ecco una prima ragione per essere cauti quando si mettono a confronto soluzioni Open Source e soluzioni proprietarie. A conti fatti, l'usabilità è un parametro troppo importante per poter passare in secondo piano, ed è lecito chiedersi: è meglio un prodotto Open Source con una pessima interfaccia o un prodotto proprietario semplice da utilizzare, che garantisce prestazioni ottimali e più ampie possibilità di personalizzazione? Lasciamo la risposta al lettore, dato che ci sono una serie di parametri che entrano in gioco di volta in volta e che possono prevalere in varia misura. Certamente è una domanda che è utile porsi, piuttosto che bollare il software proprietario come 'cattivo' e il software Open Source come 'buono' (stiamo generalizzando, ma in fondo non siamo lontani da quanto si sente spesso affermare in determinati contesti di informazione cosiddetta alternativa).

Sottolineiamo con fermezza che un prodotto Open Source può essere più ostile di un prodotto proprietario, in un determinato contesto di utilizzo, nella misura in cui non garantisce all'utente un'esperienza d'uso ottimale.

Ciò detto, va sottolineato come non manchino nel mondo Open Source esempi di prodotti che superano di molto le soluzioni proprietarie per quanto riguarda la semplicità di utilizzo, per esempio nel settore dei browser web, dove Firefox sta guadagnando un crescente consenso per i vantaggi che offre in termini di semplicità, estendibilità, intercomunicabilità.

### Formati aperti vs formati chiusi

Ciascuno dovrebbe essere padrone dei propri dati, che si tratti dei contatti di una rubrica, dei segnalibri di un browser, di un documento di testo o di una collezione di fotografie e registrazioni video. È una questione di

rispetto, e il fatto che un applicativo sia gratuito, e che il codice sorgente sia disponibile o meno, non ci dà garanzia che, passando a un altro software o a una diversa versione del medesimo, potremo conservare i nostri dati, esportandoli e reimportandoli a nostro piacimento. Certo, in linea teorica questo è possibile, ma all'atto pratico occorre che qualcuno se ne faccia carico.

Di fronte a tali considerazioni, la differenza tra software Open Source e software proprietario passa in secondo piano. L'apertura dei formati finali in cui vengono salvati i dati è spesso decisamente più importante degli strumenti in sé con cui i dati vengono salvati. Se una soluzione proprietaria consente di salvare i dati in un formato aperto e/o di esportare i dati verso altri applicativi che seguano standard internazionalmente riconosciuti, allora merita senz'altro un occhio di riguardo. Va comunque detto che il mondo Open Source è tendenzialmente a favore di formati aperti e condivisibili.

Esistono oggi moltissimi standard (in gran parte basati su Xml) che consentono agli applicativi di scambiare dati e impostazioni relativi ai contatti di una rubrica, ai segnalibri di un browser, alle lettere di un account di posta elettronica, ai contenuti di un documento di testo, e innumerevoli altri tipi di dati.

### Per lo sviluppo occorrono tempo, competenze e documentazione

È vero che codice aperto significa la possibilità per chiunque di potervi mettere mano per migliorare e personalizzare un prodotto in modo che possa evolvere. È però altresì vero che quanto più i prodotti diventano complessi, maggiore diventa il grado di organizzazione richiesto. Quando ciò avviene all'interno di un gruppo organizzato piuttosto che di un'azienda, è facile che si venga a creare una struttura di tipo gerarchico, che presenti una certa affinità con le grandi aziende per quanto riguarda il modo di gestire la complessità richiesta. Occorre dunque porsi la domanda se abbia senso, in un mondo in così rapida evoluzione come quello informatico, fare della disponibilità di codice sorgente una ragione di principio, prescindendo dal fatto che la libertà deve anche potersi concretamente esercitare, affinché sia una libertà effettiva. La libertà e la condivisione devono poi essere assicurate a tutti i livelli, altrimenti l'approccio cosiddetto libero perde il suo valore.

La documentazione, per esempio, è uno dei fattori fondamentali che entrano in gioco. Esistono prodotti Open Source la cui documentazione è di scarsissimo livello perché i programmatori non sono stati in grado di organizzarsi in modo da creare strumenti comunicativi efficaci a supporto del codice. E nella condivisione della conoscenza, relativamente a prodotti informatici complessi, la documentazione può essere addirittura più importante del prodotto.

Anche in questo caso, non mancano esempi vincenti che provengono dal mondo Open Source. Ad esempio, la piattaforma Wordpress per gestire blog offre degli



strumenti di supporto piuttosto validi e gode per questo di una straordinaria diffusione. Wordpress è un tool semplice da installare anche per coloro che non possiedono conoscenze informatiche avanzate; è potente, abbastanza ben documentato e in costante aggiornamento, e gode del supporto di una vasta comunità che rilascia gratuitamente una serie variegata di strumenti e servizi.

### Costi e assistenza

Per le aziende, questo è un altro parametro da tenere in considerazione. Ancora una volta, Open Source non significa necessariamente migliore o peggiore.



Il logo di Firefox, il browser web che sta facendo concorrenza al browser di Microsoft, Internet Explorer

Avere garantita un'assistenza qualificata in ogni momento è importante e richiede ovviamente dei costi aggiuntivi.

A tale proposito, ricordiamo che Open Source non significa necessariamente gratuito: alcuni prodotti vengono ceduti con codice aperto e gratuito, ma di fatto è richiesta una conoscenza così

approfondita dell'architettura interna e delle procedure di installazione da rendere indispensabile l'intervento di esperti specializzati, a pagamento.

### Qualità, potenza, prestazioni

Su questo fronte, è impossibile in poche righe trarre conclusioni significative perché è indubbio che sia sul fronte delle soluzioni proprietarie, sia sul fronte delle soluzioni Open Source, esistono esempi di prodotti consolidati che forniscono prestazioni ottimali. Microsoft, ad esempio, ha avuto la meglio nel settore dei prodotti per ufficio, creando prodotti molto potenti quali Word ed Excel; sul versante opposto, osserviamo che gran parte dei siti internet sono ospitati da server che girano su Apache, un prodotto Open Source di larghissimo successo.

### Spinta innovativa

Mentre il mondo Open Source ha saputo mettere in pratica idee vincenti e guadagnarsi alcune nicchie di mercato, in numerosi ambiti dominano incontrastate aziende come Adobe (con i suoi prodotti per l'elaborazione multimediale) e Microsoft (una delle più grandi innovazioni è stata .Net, una versatile, potentissima piattaforma per lo sviluppo di software e applicativi online).

In questo caso dobbiamo riconoscere che l'ago della bilancia pende a favore delle multinazionali del software, le quali non solo hanno a disposizione fondi incomparabilmente maggiori e risorse meglio organizzate, ma hanno beneficiato, negli anni, di un grave errore da parte delle comunità Open Source: quello di voler rivaleggiare con i prodotti proprietari emulandoli in tutto e per tutto, quando invece vi era ampio spazio per produrre interfacce più intuitive. Il problema nasce dal fatto che la comunità Open Source è stata per molti anni, sostanzialmente, una comunità di sviluppatori, persone notoriamente poco attente alla semplicità di utilizzo e poco consapevoli dei fattori che determinano una buona interazione uomo-macchina. L'Open Source si è così concentrato su cloni di Windows, Office e software di ogni sorta, concepiti da sviluppatori motivati da validi ideali ma carenti di idee originali. Si sono così ricalcati i medesimi errori e le medesime imposizioni ideologiche che costituivano i maggiori motivi di critica dei prodotti proprietari, prima fra tutte la prevalenza dei modelli gerarchici nell'archiviazione dei contenuti<sup>4</sup>.

### Conclusioni

Il movimento Open Source, concentrandosi su alcuni degli aspetti che caratterizzano la qualità dei prodotti e trascurandone altri, ha ottenuto molto meno di ciò che avrebbe potuto, pur producendo risultati di indubbio valore, talvolta concorrenziali o superiori ai corrispondenti prodotti proprietari.

In un mondo dove le persone hanno poco tempo libero perché sono impegnate a lavorare nelle aziende, e i software proprietari si possono ottenere gratuitamente in vari modi senza correre grossi pericoli, ha prevalso la pigrizia e la mancanza di alternative e, di fatto, i pesci grossi dell'informatica hanno protratto il loro monopolio pressoché assoluto.

L'imprenditore che deve scegliere tra soluzioni Open Source e proprietarie deve tenere conto di numerosi parametri legati a questioni contingenti, ma anche al futuro dei contenuti e dei dati. Per progetti destinati a durare in eterno, è probabile che una soluzione Open Source sia da preferire, purché la piattaforma scelta sia effettivamente di qualità e goda del supporto di un'ampia comunità di sviluppo. Al contrario, determinati tipi di business richiedono necessità di assistenza, documentazione, adattabilità in tempi stretti che una comunità Open Source non sempre è in grado di garantire.

La condivisione non può prescindere dalla semplicità di utilizzo e non può trascurare altri parametri che sono stati qui discussi, per questo introduciamo il termine di 'ideologia Open Source' in riferimento a quella convinzione, così poco vicina alla realtà dei fatti, che un segno, un nome, un principio possano garantire prodotti di qualità e a misura d'uomo.

<sup>4</sup> Il modello gerarchico impone che i dati siano racchiusi in file che, a loro volta, devono essere inseriti in cartelle e sottocartelle, ovvero in una struttura predefinita e inflessibile, senza la possibilità di creare associazioni di significato di diversa natura. È interessante notare come quest'organizzazione gerarchica si ritrovi sia a livello di dati, sia a livello di organizzazione del personale lavorativo.