

[Home Page](#) [Informazioni](#) [Aiuto](#)

Perchè gli sviluppatori di software beneficiano dell'Open Source?

http://www.vbsimple.net/news/news_09.htm

Riassunto generale

Partecipare allo sviluppo di software Open Source è altamente ricompensante, sia personalmente che finanziariamente. Inoltre gli studi recenti sottolineano che investire nel software Open Source accresce enormemente, non decresce, le opportunità per il successo economico in tecnologia. La natura innovativa del software Open Source crea nuovi interi mercati per i prodotti commerciali, nonché altre opportunità commerciali quale il supporto, l'impacchettamento, la distribuzione, l'addestramento, la consulenza di servizi, ed i contratti di sviluppo.

Introduzione

La maggior parte delle persone che non hanno esperienza nello sviluppo di software si aspettano che i guadagni sul software commerciale tradizionale siano enormemente maggiori rispetto al lavoro sul software Open Source. È incomprensibile perchè un programmatore debba sgobbare su un progetto software, solo per darlo via al pubblico, senza guadagnarci nulla e rilasciarlo *senza nessuna restrizione d'utilizzo*. Sembrerebbe più logico che scrivere un software e rilasciarlo ad un prezzo alto riporti maggiori ricompense professionali e finanziarie.

Molte migliaia di programmatori preferiscono invece scrivere software Open Source; così facendo le loro carriere ed i loro curricula tendono a migliorare. La maggior parte di loro lo fa nel **tempo libero**, in modo da guadagnare preziosa esperienza e fama; molti altri mantengono software Open Source che si rivela vitale al loro lavoro pagato. Ancor di più, programmatori ricevono commissioni o viene loro pagato uno stipendio per scrivere o migliorare software Open Source; questi "lavori da sogno" sono scelti dai migliori e più preparati nella professione di programmazione piuttosto che per impieghi tradizionali a stipendi molto più alti. *Perchè sta lavorando su un software Open Source piuttosto che su un'opzione così attraente?*

Un esempio personale

La mia più semplice risposta riguarda la mia esperienza. Ho lavorato per diversi anni su un prodotto software per una grossa compagnia. Insieme a molta altra gente sveglia ho, ingenuamente, effettuato un'attenta pianificazione ed ho duramente lavorato in questo progetto. In breve tempo la mia compagnia è stata acquistata da un rivale maggiore e l'atmosfera è cambiata; entro l'anno successivo il nostro codice iniziò a marcire nella proverbiale cantina, perché non usato da nessuno. Certo, io ero pagato per il mio tempo, ma questo mi accontentava? È veramente questo ciò che desideravo fare per vivere? Ed il

mio curriculum era migliorato solo di poco, avendo prodotto un software molto buono che tuttavia nessuno userà o rivedrà mai più.

Qualche anno più tardi, spesi pochi minuti nella scrittura di un semplice script ed in un semplice programma in linguaggio C e li spedii allo sviluppatore di una delle maggiori distribuzioni Linux. Entrambi erano ciò che serviva a quel tempo, e fu incluso nelle distribuzioni Linux usate da quel giorno in poi **da milioni di persone**. Certo, erano poca cosa, e poche persone seppero che li scrissi io, ma questi 2 piccoli pezzi di codice, scritti in meno tempo di una passeggiata con il cane, mi hanno dato compensi personali maggiori e più durevoli degli anni di lavoro che avevo fatto in precedenza.

Poco dopo diventai il maintainer o l'aiutante maintainer di alcuni pacchetti software Open Source ampiamente utilizzati. Questo lavoro si presenta come un punto fantastico in un curriculum. Con il tempo ho imparato nuove tecnologie che sono più tardi diventate rilevanti al mio lavoro giornaliero facendomi avanzare professionalmente ed introducendomi in nuove ed eccitanti aree nelle quali non sarei potuto entrare diversamente. Questo approccio ha richiesto un modesto investimento di tempo libero, ma mi ha ricompensato largamente nel lungo tempo.

Un senso di comunità

Un'altro problema con lo sviluppo di software tradizionale è che incoraggia lo sviluppo di codice "inutile". Ogni azienda spende grosse risorse *"inventando nuovamente la ruota"*, sviluppata già dai loro concorrenti -- questo è un lavoro insoddisfacente e frustrante che deve essere fatto prima ancora che l'innovazione possa cominciare. I giovani programmatori vedono soltanto esempi di insegnamento (sui libri ad esempio) ed il codice dei colleghi nella compagnia. L'accesso ai codici dei colleghi è generalmente limitato alla squadra di progetto, le cui elevate diversità di viste ed esperienze spesso comportano problemi di *"incrocio"* e design scadenti.

L'ampia disponibilità di software Open Source ed i modelli di sviluppo basati su Internet aggiungono un contesto di sociale ed un elemento di comunità allo sviluppo del software. Il miglior codice scritto dal migliore dei programmatori senior diventa un punto di partenza ed un pregevole strumento di apprendimento anziché un ostacolo che deve essere reimplementato. L'accedere a codice di diffusione mondiale permette una progettazione ed implementazione di alta qualità con uno sforzo minimo. I conflitti di personalità possono ancora essere un problema, ma la competizione è principalmente basata sui risultati -- ricompensando quindi quelli che investono gli sforzi e l'intelletto per effettuare il lavoro nella maniera corretta.

Il mito del codice indispensabile

Molti programmatori promettenti ma senza esperienza pensano di poter scrivere da soli un software così buono da non poter essere copiato da nessun altro. Per questo unico prodotto possono chiedere molto danaro, che la gente pagherà perché ha bisogno di quelle caratteristiche. Ma non è così che il mercato funziona. Semplicemente ci sono moltissimi

programmatore svegli per ogni programmatore che detiene il monopolio su una buona idea. È assurdo ed arrogantemente folle presumere che un singolo individuo può creare un prodotto che l'intero mondo pagherà a caro prezzo perché basato sul suo merito.

Anche il codice più complesso che qualcuno può sperare di scrivere ha una forza di mercato davvero miserevole. Per esempio, un gruppo di brillanti programmatori hanno crackato, decodificato (reverse engineering) e duplicato gli algoritmi di sicurezza Microsoft più segreti in pochi mesi del 1997 e **nel loro tempo libero**. E, quando un anno dopo la Unisys ha cominciato a richiedere royalties (pagamento sui diritti d'autore) sull'uso commerciale del formato grafico GIF, è stato sviluppato ed implementato un nuovo formato grafico, più piccolo, più comprensivo e più efficiente partendo da zero. Con qualche serio incentivo, interi sistemi operativi possono essere (e sono stati) rimpiazzati e le loro caratteristiche cruciali, replicate o aggirate a discapito di brevetti o copyright che impediscono un diretto approccio di copiatura o clonazione.

Il merito intrinseco di un dato prodotto software è responsabile solo in minima parte al suo successo o valore economico. Anche il prodotto più insignificante e ricco di errori può riscuotere un tremendo successo economico se viene proposto forzatamente alla gente che ne ha bisogno (o che crede di averne bisogno) -- vedi ad esempio le prime versioni di DOS. Il contesto del mercato, il suo tempo e la pubblicità sono i più importanti determinanti del successo rispetto ai meriti tecnici del prodotto software. Ma introducendo un prodotto con una licenza estremamente generosa, specialmente se rilasciato con il suo codice sorgente, i costi e le barriere burocratiche (del mercato) sono eliminati, rendendo il software un probabile successo ampiamente utilizzato ed apprezzato, cosa che non accadrebbe se venisse rilasciato senza codice sorgente o sotto dei termini di licenza restrittivi.

Perché il software Open Source non eliminerà il software commerciale?

L'avvocato del diavolo punterà il dito sul fatto che lo sviluppo di software commerciale per un sistema operativo quale Linux, nel quale sono compresi tanti tanti software Open Source, non può avere una buona posizione rispetto allo sviluppo per un sistema monopolista quale Microsoft Windows per il quale sono presenti moltissimi software commerciali. In entrambi i casi, il rischio è che un prodotto e la fonte di reddito vengano superati da qualcosa disponibile a tutti **senza costi aggiuntivi**. Così, cambiando barca da Microsoft Windows a Linux, uno sviluppatore di software commerciale si porta dalla padella alla brace (almeno Microsoft impiega migliaia di programmatori). La semplice ed ampia disponibilità di licenze Open Source porterebbe quindi un incremento di perdite di lavoro nella professione del software.

L'analogia, tuttavia, è un'idea teorica senza basi su un fatto empiricamente economico. Al contrario, la Microsoft spesso ed intenzionalmente punta i prodotti più redditizi dei suoi concorrenti. In alcuni casi essa sviluppa o acquista direttamente prodotti concorrenti utilizzando le sue tremende risorse di marketing e potere commerciale ed impossessarsi così del loro mercato; in altri casi essa acquista l'intero concorrente; e nel peggiore dei casi include le funzionalità del prodotto direttamente all'interno di Windows allo scopo di eliminare l'intera categoria di prodotto e l'altrui fonte di guadagno.

La comunità del software libero ha considerevoli risorse, ma non reagisce ai prodotti commerciali per annullare o indebolire il mercato. La maggior parte dei software liberi è spinta dal soddisfare un bisogno non soddisfatto, nella forma di prodotti che semplicemente prima non esistevano o provvedendo un prodotto specializzato per il quale non esiste un incentivo commerciale nello svilupparlo, oppure provvedendo un'alternativa ad un prodotto software commerciale di bassa qualità o dal prezzo esorbitante che manca delle caratteristiche necessarie ai suoi clienti. A differenza della Microsoft, gli sviluppatori di software libero non hanno motivo di distruggere i venditori di software commerciale per diminuire la competizione. Ad essi mancano anche gli enormi investimenti pubblicitari e forza di mercato rispetto a Microsoft, senza i quali non possono porsi come seria minaccia ai prodotti software commerciali.

Il software Open Source è anche molto guidato dai clienti, *perché i clienti direttamente contribuiscono e controllano il suo sviluppo*. Microsoft ed altri grossi venditori di software commerciale sono invece guidati dal massimizzare i profitti, e soddisfare i bisogni dei loro clienti è soltanto uno degli ultimi scopi. Certamente alcuni sviluppatori di software commerciale sono invece altamente ricettivi e benèfici verso i loro clienti; ma quando queste aziende mancano di forte competizione, gli altri obiettivi aziendali presto prendono la priorità ed i clienti soffrono per questo.

Nel caso di un prodotto commerciale di buona salute ed alta qualità che compete in ragione dei suoi meriti, l'incentivo a sviluppare un'alternativa gratuita è troppo debole per motivare un'ampia forza di sviluppatori volontari a spendere un considerevole investimento di tempo e di risorse intellettive. In altre parole, la comunità di sviluppatori di software libero può solo fare preda prodotti commerciali deboli che sono diventati *grassi, pigri* e costosi per una mancanza di competizione (e pertanto non soddisfano più i bisogni dei clienti in maniera adeguata). Inoltre gli sviluppatori di software libero sono molto più interessati ad essere pionieri in nuovi mercati piuttosto che a competere in quelli esistenti. Questi nuovi mercati a loro volta spingono enormemente la crescita di nuove aziende per software commerciali. La storia delle tecnologie internet rappresenta una drammatica dimostrazione dell'importanza di quest'ultima motivazione.

Infine ci sono molte classi di software nelle quali gli sviluppatori Open Source sembrano non avere molto interesse per occuparsene, perché non forniscono una sfida tecnica interessante o perché richiedono uno sforzo concentrato di un team ampio e centralmente coordinato. Gli esempi includono gli elaboratori di testi (word processor), i fogli di calcolo, il software finanziario, e molti altri tipi di software nei quali la qualità (lustratura) commerciale, la sicurezza di qualità, la documentazione ed il supporto sono somma cosa.

Invece il modello di software Open Source è più forte nei ruoli di infrastruttura essenziale come nei sistemi operativi dove può costituire un vero campo di gioco per gli sviluppatori di applicazioni commerciali, integratori di sistemi e consulenti. Esso è anche un eccellente veicolo di riferimento per implementazioni e prove di concetti su protocolli di rete, strumenti di integrazione ed altre entusiasmanti tecnologie. La libera disponibilità del codice sorgente è un fattore chiave di compatibilità tra le successive implementazioni in competizione, incluse quelle commerciali.

Un ringraziamento particolare all'autore dell'articolo: [Niemi di tux.org](http://Niemi.di.tux.org)
Documento originale: Why Software Developers Benefit from Open-Source Software
Traduzione di [Fibia FBI](http://Fibia.FBI)
4 Aprile 2002



[Torna all'indice degli Articoli](#)
