

Il software Open Source:

Renata Bortolin

Open Source e Free Software sono termini che evocano libertà e gratuità. E' veramente così? Il modello introdotto da LINUX, il più famoso progetto Open Source, trova applicazione per tutti gli strumenti che compongono il sistema informativo delle aziende?

1. OPEN SOURCE: UN NUOVO FENOMENO?

I mondi Free Software ed Open Source sono nati molti anni fa, ma solamente negli ultimi mesi hanno conosciuto una più ampia diffusione, spesso accompagnati da un altro termine molto in voga: LINUX. Da quando in questo nuovo contesto hanno fatto il loro ingresso le più importanti case fornitrici di hardware (come IBM, HEWLETT PACKARD, SUN e DELL), i loro nomi sono usciti dall'ambito riservato agli addetti ai lavori, apparendo anche sulla stampa non specializzata, oltre che in televisione, accrescendo così la curiosità e le attese degli utilizzatori di strumenti informatici.

A cosa si riferiscono esattamente? Sono davvero portatori di una rivoluzione, oppure rappresentano una qualche minaccia, come qualcuno asserisce? Sono fenomeni che porteranno qualche beneficio anche nelle imprese? Condurranno ad un'eliminazione dei costi dei programmi di gestione delle nostre aziende? E se così fosse, dove si nasconde il business per i fornitori? O si tratta piuttosto di argomenti accademici, da lasciare in mano ad un mondo di fanatici e di idealisti, perché non troveranno mai applicazione concreta? Chi è concorde con l'ultima affermazione, dovrà ricredersi. Se in un'azienda si fa uso della posta elettronica, se vengono forniti servizi via web, se si dispone di un proprio sito, è molto probabile che vengano usati strumenti software chiamati Apache e Sendmail, ed è altrettanto probabile che il server web funzioni grazie al sistema operativo LINUX;

questi sono tutti strumenti provenienti dal mondo Open Source.

2. IL SOFTWARE COMMERCIALE

Per comprendere i movimenti che hanno portato alle due definizioni prima citate, contrapposte a quella di software commerciale, è indispensabile fare un pò di storia.

In principio tutto il software era pubblico. I primi computer commerciali venduti su larga scala da IBM negli anni '60, erano corredati del software, che poteva essere liberamente scambiato tra gli utenti. Era disponibile il codice sorgente, che poteva essere modificato e adattato a piacere e l'hardware costituiva la quasi totalità del costo del sistema informativo. All'epoca, avevano accesso ai sistemi solo programmatori che dovevano continuamente modificare il software per svolgere il lavoro di tutti i giorni. Non esistevano i programmi che conosciamo oggi e gli utenti erano solo i tecnici.

A mano a mano che aumentavano le prestazioni delle macchine, e quindi il numero di elaborazioni che queste potevano eseguire, cresceva anche la complessità e la quantità del software, utilizzato sempre di più in aziende diverse, per compiti diversi. Il numero di utenti aumentava, ma non così il numero di sviluppatori. Diventò allora economicamente interessante sviluppare programmi per poi venderne la sola licenza d'uso. In questo modo, i programmi erano mantenuti chiusi, non modificabili dall'acquirente, al quale

Avvicinare il mondo delle aziende al modello di sviluppo del software libero per scoprire i principi di libertà del software: Libertà di Modifica, Libertà d'Uso, Libertà di Ridistribuzione. Perché le informazioni siano liberamente condivise e la libertà, grazie alla conoscenza ed al miglioramento continuo, sia la vera ed unica fonte d'innovazione.

dal mito alla realtà d'impresa

*libertà di pensiero,
di comunicazione,*



LINUX: libertà di pensiero, di comunicazione, di scelta.

non veniva venduto il codice sorgente (condizione necessaria per modificare i programmi), ma solo il diritto ad utilizzare il programma eseguibile. Si iniziò a parlare di proprietà intellettuale, di brevetti, di software come bene e non come servizio.

Alla fine degli anni '60, IBM iniziò a separare le vendite dell'hardware da quelle del software. Già alla metà degli anni '70, la vendita del software chiuso, non ridistribuibile, non accompagnato dal codice sorgente e non modificabile era diventata una consuetudine.

Nasceva così il Software Commerciale, che oggi rappresenta il mercato contrapposto a quello generato dal Software Open Source. Il software commerciale è interamente sviluppato da un'azienda con lo scopo di realizzare un profitto dalla vendita dello stesso.

Anche in ambito accademico, fino ai primi anni '80, i programmatori avevano continuato a realizzare sistemi operativi ed altre componenti, distribuendoli liberamente, senza curarsi degli aspetti inerenti alle licenze ed ai costi di sviluppo. Questi erano ben felici di cedere a terzi i loro programmi, così come erano felici di poter ispezionare il codice sviluppato da altri per comprenderne tecniche e funzionamento e riusarne alcune parti. Sono nati così interessanti strumenti, tutt'ora in uso, come il linguaggio di programmazione "C" ed il sistema operativo UNIX, quest'ultimo creato libero ed in seguito divenuto commerciale. Nelle università, lo sviluppo del software era considerato alla stessa stregua della ricerca. Per questi programmatori, la nascita del software commerciale costituì una grave violazione della libertà individuale e della libertà di comunicazione, un sistema per dividere e discriminare, un sistema per limitare la ricerca.

3. LA GENESI DELL'OPEN SOURCE E LE SUE CARATTERISTICHE

In questo scenario, tra la fine degli anni '70 e i primi anni '80, due diversi gruppi, che avevano visto minacciata la loro libertà, iniziarono a costruire le basi dei movimenti "Free Software" (Software Libero) e "Software Open Source".

Oggi, nonostante le due differenti definizioni, i concetti di Free Software ed Open Source sono spesso usati indistintamente per riferirsi al software modificabile (spesso sviluppato da una comunità liberamente organizzata), del quale sono disponibili

i sorgenti e la cui distribuzione avviene gratuitamente, tramite download da Internet. Il modello di riferimento è LINUX, l'esempio più famoso di software Open Source.

Le basi per la definizione del software libero furono gettate nel 1983, quando un programmatore del MIT (Massachusetts Institute of Technology), Richard Stallman, lanciò ufficialmente un progetto denominato GNU, atto a favorire la realizzazione di un sistema operativo libero che si proponesse come alternativa ad UNIX (quest'ultimo, nato come software libero, successivamente si è evoluto verso una moltitudine di UNIX incompatibili tra loro e sul quale più organizzazioni commerciali vantavano dei diritti). I principi a cui si ispirava questo progetto non erano commerciali, ma etici, e si basavano sul presupposto che le informazioni devono essere liberamente condivise e che la libertà è la vera ed unica fonte di innovazione, grazie alla conoscenza ed al miglioramento continuo. Così, secondo la sua tesi, chi modifica il software, per migliorarlo ed adeguarlo alle nuove necessità, lo deve rendere disponibile alla stessa comunità dalla quale lo ha ricevuto. Nel 1985, Stallman fondò la "FREE SOFTWARE FOUNDATION" ("FSF"), organizzazione finalizzata a sviluppare il progetto GNU. I principi di libertà del software, ossia la Libertà di Modifica, Libertà d'Uso, Libertà di Ridistribuzione, furono definiti attraverso un documento, la "GNU PUBLIC LICENSE" ("GPL"), ancora in uso per il software libero. Chiunque può modificare un software libero e cederlo a terzi, anche a fronte di un corrispettivo, ma senza dover riconoscere alcun compenso all'autore del software originale; tuttavia, non può in alcun modo farlo diventare un software proprietario. Il software libero, anche se modificato, non può comprendere componenti proprietarie e non può imporre limitazioni maggiori rispetto al software da cui deriva. Il progetto GNU fu portato a termine con la distribuzione di una licenza free di UNIX, ancora in uso. I principi di Stallman non trovarono però sufficiente adesione nel mondo delle imprese. Nel 1998, l'OPEN SOURCE INITIATIVE ("OSI"), fondata da due ricercatori, Raymond e Perens, entrambi attivi in progetti Free Software, elaborò una strategia volta a far avvicinare il mondo delle aziende al modello di sviluppo del software aperto. Lo scopo era di dare maggior credibilità al software libero, attraverso la definizione formale di Open Source e la

creazione di un marchio di certificazione per il software aperto. I fondatori percepirono che era indispensabile eliminare qualsiasi citazione filosofica, politica ed etica presente nella definizione del Free Software, e confermare nello stesso tempo i principi di adattabilità, sicurezza, indipendenza. Il termine free doveva essere sostituito poiché traeva in inganno: secondo la traduzione letterale del termine inglese, free significa libero, ma anche gratuito, cosa che appare poco credibile al mondo delle aziende.

Il software Open Source è caratterizzato dal fatto che i sorgenti sono aperti, ossia disponibili, consultabili e modificabili. Tuttavia, questo concetto è ancora confuso con quello che indica la gratuità del software, ed ha suscitato reazioni negative l'iniziativa recente di qualche società di richiedere un corrispettivo a fronte della distribuzione dell'ultima versione dei propri prodotti Open Source, anche se questo non è in contrasto con i principi dell'Open Source stesso.

Vendere una copia di un programma non è in

contrasto con la libertà. Lo stesso Stallman, una volta uscito dal MIT, trovò come ricavare quanto necessario per il suo sostentamento, facendo pagare 150\$ a chiunque gli richiedeva un nastro con il software da lui sviluppato. La Free Software Foundation accetta donazioni, ma i suoi introiti principali sono dovuti alle copie di software libero e servizi venduti (cd-rom con raccolte di programmi e manuali, per esempio).

L'attività di raccolta di tutti gli strumenti utili per una determinata utenza, il loro trasferimento su un unico cd-rom, la realizzazione di funzioni che automatizzano l'installazione, il tutto accompagnato dai manuali d'uso (talvolta anche stampati), è un servizio apprezzato dagli utenti, che non sono in questo modo costretti ad individuare e a scaricare dalla Rete i componenti a loro utili. Questa è l'attività eseguita dai distributori di LINUX, per esempio. L'unico vincolo da rispettare, imposto dalla licenza 'GPL', è che tutto il software libero non sia mai conglobato in software proprietario e che non siano posti limiti alla sua libera diffusione.

La tabella sotto riportata riassume, allo scopo di evidenziarne le caratteristiche, le principali categorie di software che abbiamo trattato fino ad ora ed anche altri modelli marginali di distribuzione del software, citati solo per completezza.

Tipo di programma	Descrizione	Gratuito	Ridistribuibile	Uso illimitato	Le versioni derivate sono libere	Diponibilità sorgenti
Software Free	È liberamente utilizzabile, modificabile e distribuibile.	Si	Si	Si	Si	Si
Software Open Source	Sono disponibili i sorgenti, che sono quindi consultabili e modificabili.	Non sempre	Si	Si	Si*	Si
Software Commerciale proprietario	Realizzato per trarre un profitto, attraverso la concessione delle licenze d'uso.	No	No	No	No	No raramente Si
Software Commerciale libero	I profitti sono generati da vendita di servizi.	Non sempre	Si	Si	Non sempre	Si
Software Shareware	Si tratta in genere di programmi per usi specifici e funzionalità limitate, che i programmatori hanno messo a disposizione su Internet, e spesso la licenza d'uso è di entità modesta, volta solo a coprire i costi di realizzazione.	No	Si	Si	No	No
Software Freeware	Da non confondere con il Free Software. È spesso messo a disposizione su Internet a scopo promozionale. Spesso ha funzionalità ridotte rispetto al corrispondente software commerciale.	Si	Si	Si	No	No

* secondo la licenza GPL esistono altre varianti

D'ora in poi ci riferiremo allo stesso concetto di disponibilità del software usando indifferentemente i termini Open Source, software aperto e software libero. Lo sviluppo del software libero continuò per tutti gli anni '80 e '90 di pari passo con Internet, secondo un modello auto-costituito e collaudato. Diversi progetti che si ispiravano ai principi di libertà, condivisione di informazione, indipendenza dall'hardware e dai fornitori, venivano avviati e conclusi anche senza riferirsi esplicitamente ad uno dei modelli citati. Un esempio significativo è quello riguardante il sistema 'X-WINDOW', sviluppato dal MIT nel corso degli anni '80, che è stato il primo ambiente client/server, con interfaccia grafica a finestre, in grado di assicurare l'indipendenza dei dispositivi di interfaccia come video, mouse e tastiera rispetto ai server. La versione completa, e vicina a quella tutt'ora in uso, è stata rilasciata nel 1988, molto prima della nascita di WINDOWS 95. Alla fine degli anni '80, nove produttori di hardware, ovvero Apollo, DEC, Hewlett-Packard, IBM, Bull, Nixdorf, Philips, Siemens e Hitachi, fondarono la "OPEN SOFTWARE FOUNDATION" ("OSF"). Il loro obiettivo era di creare un ambiente di elaborazione aperto, basato su UNIX e su X-WINDOW. Nell'ottobre 1990, la meta fu raggiunta con il rilascio di OSF/1, la prima versione del nuovo sistema aperto. X-WINDOW è stato il precursore degli ambienti a finestre, sul quale tuttavia pochi produttori di software applicativo hanno basato le loro soluzioni. La solidità del sistema è oggi attestata dal fatto che il cuore della gestione di tutti gli ambienti grafici di LINUX è basato su Xfree86, un'implementazione gratuita di X-WINDOW. In Italia, il primo ERP basato su X-WINDOW, MAGIA90, fu rilasciato nel 1991. Era già all'epoca in grado di funzionare su tutti i server UNIX dei principali fornitori di hardware, un esempio di libertà e di indipendenza unico per l'epoca! La cosa che maggiormente colpì pubblico e stampa specializzata fu il fatto che a realizzarlo non era stato un grande centro di ricerca internazionale, ma una piccola società di software di Pordenone, la DATA CONSULT Sistemi Informativi.

4. IL CASO PIÙ FAMOSO: LINUX

La data storica a cui si fa risalire la nascita del software libero è il 1991. Linus Torvalds, allora studente di informatica ad Helsinki, ansioso di poter

avere un valido sistema operativo sul proprio PC di casa senza dover spendere molti soldi, iniziò a lavorare alla realizzazione di un sistema Unix-compatibile completamente libero, che battezzò LINUX. Al suo progetto, grazie alla Rete, si interessarono da subito moltissimi programmatori, che iniziarono, di propria volontà, a collaborare con lui. Dopo un anno, il cuore del sistema operativo (kernel) era in grado di funzionare e poteva inoltre far funzionare anche gran parte degli strumenti sviluppati nel progetto GNU; si trattava per la maggior parte di strumenti utili allo sviluppo di software, come editor e compilatori. Mano a mano che le funzionalità aumentavano, crescevano anche gli sviluppatori che aderivano volontariamente al progetto. Le prime versioni di LINUX, costituite all'inizio da poche funzioni, furono rilasciate poco dopo l'avvio del progetto, ma il sistema iniziò ad affermarsi al di fuori dell'ambiente accademico nella seconda metà degli anni '90. Uno dei progetti più grandi e più complessi per lo sviluppo del software, la realizzazione di un intero sistema operativo, si era concretizzata sotto il coordinamento di una sola persona. Il fenomeno LINUX continua oggi la sua ascesa senza la collaborazione di Linus Torvalds. Quest'ultimo, chiamato nel 1994 all'Università di Helsinki per proseguire il suo lavoro su LINUX, decise nel '97 di trasferirsi in California. Linus Torvalds è ormai considerato l'informatico anti - Bill Gates per eccellenza, un suo temuto avversario. A chi gli chiede com'è riuscito a fare tanto risponde "I don't know", non lo so. Ancora si diverte ed afferma che i programmi per computer sono frutto delle idee migliori che nascono dalla passione: non parla mai di lavoro, sempre di passione. Oggi LINUX sta conquistando il predominio su tutti gli altri sistemi operativi per alcuni tipi server (web server), ma sta rapidamente affermandosi anche in alternativa a qualsiasi altra soluzione basata su WINDOWS NT o UNIX.

5. MODELLO DI PRODUZIONE DEL SOFTWARE OPEN SOURCE

L'organizzazione della comunità ed i suoi membri

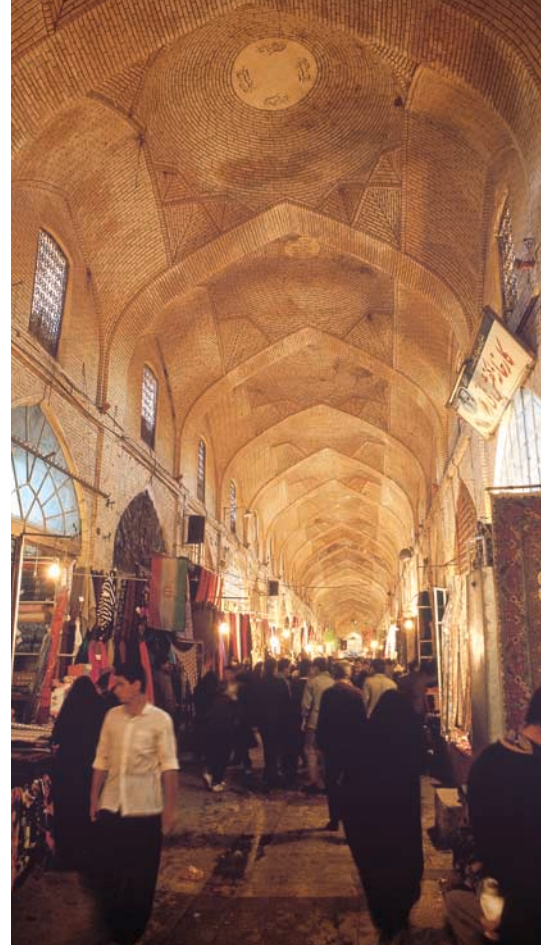
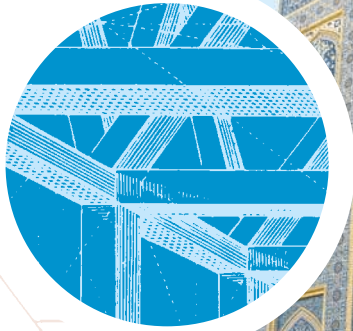
E' interessante capire come è organizzata la comunità di sviluppo Open Source e su quali principi è basata.

Il modello è stato illustrato in alcuni saggi, di cui il

più famoso è “La Cattedrale ed il Bazar”, scritto nel 1997 da Eric Raymond (uno dei due fondatori dell’OSI), un testo che ha impressionato i lettori a tal punto da spingere qualche fornitore di software a rendere disponibili i sorgenti dei propri prodotti

sviluppo del software Open Source nasce da un’esigenza di un programmatore, il quale decide di realizzare da solo gli strumenti a lui utili. Si è soliti parlare di un “prurito da scacciare”. Affinché possa nascere un progetto di successo, è necessario che

La cattedrale ed il bazar: due diversi stili di sviluppo. Il modello “cattedrale” in voga in gran parte del mondo commerciale si oppone al modello “bazar” del mondo Linux.



... nessuna cattedrale da costruire in silenzio e reverenza: la comunità LINUX assomigliava ad un grande e confusionario bazar, pullulante di progetti e approcci tra loro diversi. Un bazar dal quale soltanto una serie di miracoli avrebbe potuto far emergere un sistema stabile e coerente...

(come nel caso di Netscape per il proprio browser). Il Bazar è un luogo dove apparentemente sono affiancate diverse merci, senza un ordine precostituito. Nonostante questa disorganizzazione, la funzionalità del Bazar non viene meno; vi si trova tutto ciò che serve, tutto è facilmente individuabile e reperibile. Mentre lo sviluppo del software commerciale nasce a fronte di un piano di marketing, cui seguono le fasi di progettazione, implementazione, test e commercializzazione, lo

nella comunità, generata attraverso Internet, siano presenti molti programmatori con lo stesso prurito, tali da creare insieme un gruppo di dimensioni congrue per concretizzare l’idea iniziale. Il programmatore che ha lanciato il progetto sarà colui che ne coordinerà lo sviluppo, che deciderà se accogliere o meno tutte le implementazioni realizzate dalla comunità, per inserirle poi nel software originario. La validità dell’idea ne determinerà o meno il successo.

Alla base di qualsiasi progetto di sviluppo Open Source vi è Internet, infrastruttura indispensabile per lo sviluppo del software libero che lega il suo straordinario sviluppo proprio allo stesso software free, sul quale si basa. Solo attraverso la Rete le innovazioni possono essere condivise velocemente attraverso tutto il mondo. Le comunità di sviluppo sono autodisciplinate e non sussistono significativi problemi organizzativi, come si potrebbe invece immaginare. Si può infatti pensare al proliferare di software non sviluppato correttamente o di funzionalità inutili. I programmatori vedono il software che producono di loro iniziativa come una loro creatura, e la loro professione quasi come una forma d'arte. Ogni software è firmato dall'autore, condizione che consente all'autore stesso di ricercare prestigio all'interno della comunità mondiale dei programmatori. Le funzionalità inutili o scritte male non vengono semplicemente accolte. Ma chi paga i programmatori per questo lavoro? La risposta è semplice: i programmatori realizzano software per soddisfare esigenze legate alle loro attività e creano strumenti per supportare il loro lavoro quotidiano; quindi, queste mansioni rientrano in quelle svolte per le aziende con le quali collaborano, o, nel caso di ricercatori, rientrano nei rispettivi progetti di ricerca. Si può quindi dire che, salvo rari e limitati casi, si tratta di volontari pagati per svolgere i loro compiti. I programmatori che sviluppano individualmente, esclusivamente per passione ("hacker" autentici), dedicano a queste attività solo qualche ora del loro tempo libero ogni settimana. Società che hanno fondato il loro business sul software Open Source, come REDHAT, che distribuisce una versione di LINUX, pagano i loro collaboratori per la realizzazione di software Open Source. E' pressoché impossibile determinare la dimensione della comunità di sviluppatori Open Source. Un sito, sourceforge.net, cataloga i soli progetti Open Source registrati. A dicembre 2000 i progetti attivi erano 14.000 ed i programmatori registrati 110.000. Oggi, i progetti sono oltre 54.000 ed i programmatori registrati oltre 540.000. Per capire meglio chi sono i programmatori che si aggregano alla comunità e da che cosa sono motivati, bisogna comprendere il significato della parola hacker. L'evoluzione del termine è sfortunata, perché i mezzi di comunicazione lo hanno legato alla figura del pirata informatico capace di violare i

sistemi computerizzati altrui, provocando danni ingenti. Hacker deriva dal termine "to hack", fare a pezzi: l'hacker sminuzza il software, lo ripara, lo modifica, lo rimette in ordine, ricrea qualcosa di nuovo e di migliore, in continua sfida con se stesso. La garanzia di un ottimo risultato è costituita principalmente dal fatto che opera imitando i modelli migliori e dalla consapevolezza che il suo operato verrà giudicato dalla comunità.

Il bagaglio culturale dell'hacker non è costituito da teoremi complessi, quanto da competenze tecniche acquisite con la pratica e soprattutto da curiosità, il tutto mescolato ad una buona dose di genialità. L'atteggiamento psicologico dell'hacker, che motiva la costante ricerca verso il meglio, si è formato alla fine degli anni '80, quando le risorse a disposizione non erano distribuite così generosamente come oggi, e bisognava escogitare il sistema di far eseguire ciò che serviva a macchine con poca memoria e limitate capacità di calcolo. Riuscire a realizzare quanto ha in mente rappresenta quindi una sfida con una macchina. L'hacker aborrisce ciò che è progettato male, ed è per questo che viola i sistemi altrui, per dimostrarne l'inadeguatezza. Lo scopo dell'hacker è disegnare un bel software e farlo funzionare. Chi non è motivato da questa determinazione, non potrà mai essere un hacker, così come una persona che non ama la musica non potrà mai essere un compositore.

Le caratteristiche dei progetti Open Source

L'industria del software si pone oggi alcuni quesiti. Il nuovo modello potrà contribuire a colmare il divario che esiste tra la richiesta di soluzioni software e l'offerta? Potrà contribuire a ridurre i costi di sviluppo e di manutenzione del software? Uno degli aspetti critici per il software è la complessità dello stesso, che ne rende molto difficile la manutenzione e ne rende incerta l'affidabilità. Delegare alla comunità lo sviluppo e la manutenzione dei programmi consente di eliminare i costi derivanti dal test, test che è eseguito direttamente dalla stessa comunità, che si occupa dell'individuazione e della correzione degli errori. La teoria, supportata dalla pratica, è che se il numero di utilizzatori di un prodotto è alto, gli errori inevitabilmente presenti nel software verranno rilevati e corretti prima, e quindi questo potrà essere reso rapidamente disponibile in versione stabile, ovvero, utilizzabile senza errori. Requisito

indispensabile per supportare questo paradigma è che i primi utilizzatori siano tecnici in grado di realizzare e modificare il software, nonché individuare e correggere gli errori. I programmatori ed i primi utilizzatori devono essere pari, ovvero di cultura ed attitudini uguali.

Nel mondo nulla di grande è stato fatto senza passione.

Hegel, "Lezioni sulla filosofia della storia"

I progetti fino ad ora completati sono stati realizzati per soddisfare le esigenze delle persone che li hanno lanciati e completati. Le funzionalità desiderate da questi utenti, ed in primo luogo dai programmatori, sono omogenee: strumenti che consentono di trasportare gli stessi programmi su macchine diverse, che consentono di non riscrivere più volte le stesse funzioni, che facilitano il controllo dei programmi, strumenti per la comunicazione tra gruppi di lavoro, e così via. Di conseguenza, esiste oggi molto software libero destinato prima di tutto a sistemisti e programmatori. Anche le distribuzioni di LINUX e gli strumenti messi a disposizione da società che fanno di LINUX il loro business, come REDHAT o SUSE, sono utilizzati prima di tutto da informatici. L'altro ambito dove il software libero si sta affermando è costituito dalle applicazioni di tipo office e presenta le stesse caratteristiche di condivisione delle esigenze. Anche queste applicazioni nascono e si sviluppano per soddisfare le esigenze di una massa critica di utenti-programmatori in grado di alimentare i progetti. Ora, anche importanti produttori di hardware affidano la realizzazione di parti di sistemi operativi alla comunità. I sistemi operativi sono necessari per il funzionamento dell'hardware. I produttori di sistemi e componenti sono costretti ad investire cifre considerevoli per sviluppare gli strumenti in grado di farli funzionare e colloquiare con l'hardware di altri fornitori. Delegare lo sviluppo di questo software alla comunità Open Source può costituire un'alternativa valida. Oltre ad eliminare i costi di sviluppo, si possono ottenere benefici in termini di tempestività nel rilascio e qualità, per i motivi prima descritti.

6. LINUX CONTRO WINDOWS E LA MINACCIA AI SOFTWARE PROPRIETARI

La progressiva ed imponente affermazione di LINUX è stata, ed è tutt'oggi, determinata, oltre che dalla condivisione della filosofia che sottende allo sviluppo dello stesso, da risparmi significativi derivati da diversi fattori:

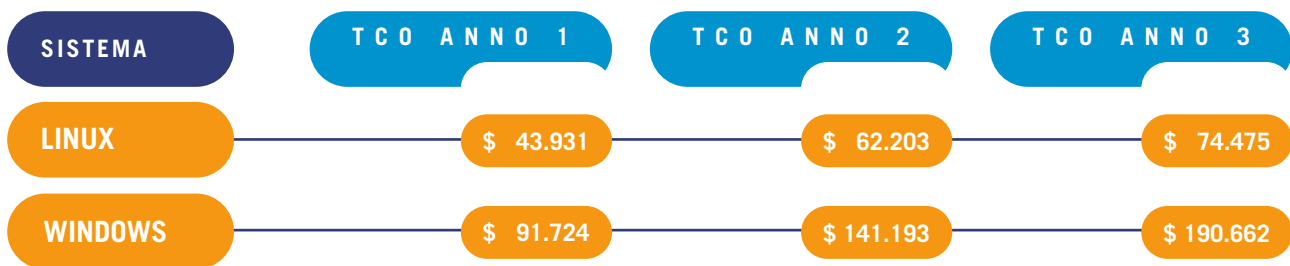
- Costo irrilevante delle licenze del software. Una distribuzione di LINUX completa di cd-rom e manuali costa pochi dollari, e può essere installata su un numero illimitato di server e posti di lavoro.
- Funzionamento su hardware meno potente di quello richiesto da WINDOWS e UNIX. LINUX garantisce, su un server basato su processore Intel, analoghe prestazioni rispetto a quelle offerte da UNIX su server di fascia superiore. Risulta vincente anche nel confronto con WINDOWS-NT.
- Riduzione dei costi di acquisizione e di gestione dei server. E' una conseguenza del fatto che è richiesta una potenza inferiore rispetto a quella richiesta dai sistemi concorrenti.
- Riduzione del personale impegnato per la gestione e manutenzione dei sistemi e del costo per la formazione del personale. Secondo una recente indagine di IDC, il personale EDP necessario è ridotto del 79% rispetto a quanto richiesto per i sistemi UNIX. Il costo per la formazione è ridotto complessivamente dell'87%, questo sia per il minor impegno che per la maggiore disponibilità di personale tecnico già formato.
- Garanzia di scalabilità, assicurata dalla disponibilità di LINUX e degli strumenti indispensabili, come i database, su server di tipo e dimensioni diversi. IBM, per esempio, consente di installare LINUX su tutta la gamma dei suoi server, dalla fascia d'ingresso basata sugli xseries ai mainframe. I database Oracle e DB2 affiancano queste soluzioni.
- Affidabilità e sicurezza che garantiscono continuità operativa. Il livello di affidabilità è tale che molti produttori di software hanno scelto LINUX come piattaforma più sicura: Oracle ad esempio fornisce una soluzione, con la versione 9i del suo database, denominata "Unbreakable LINUX" ("LINUX Indistruttibile"). La soluzione contempla il database relazionale, un ambiente di sviluppo completo e ricco di strumenti utili per la realizzazione di software object-oriented ed il relativo "application server".

Il software Open Source: dal mito alla realtà d'impresa

Molte aziende, come Amazon e Credit Suisse, hanno sostituito i loro server web basati su altri sistemi operativi con server LINUX realizzando risparmi di diversi milioni di dollari, soprattutto per le spese di gestione e manutenzione. Il risparmio stimato per Amazon è di una riduzione di spese di IT del 24% sul totale.

Nella figura seguente è rappresentato il raffronto tra "total cost of ownership" di soluzioni internet/intranet basate su LINUX e quello delle stesse soluzioni basate su WINDOWS, rilevati attraverso un'inchiesta condotta su medie/grandi aziende americane operanti in diversi mercati

di un collaboratore della stessa società di Redmond. Raymond, uno dei due fondatori dell'Open Source Initiative, contribuì alla sua diffusione. In questo documento, il così detto Halloween Document, Microsoft si interrogava sul fenomeno e definiva una strategia per fronteggiarlo, invitando ad utilizzare qualsiasi mezzo utile per raggiungere lo scopo. Un anno fa, James Allchin, Chief del sistema operativo WINDOWS, asseriva che "l'Open Source è distruttore della proprietà intellettuale, la peggior cosa che sia accaduta all'industria del software, una cosa che uccide l'innovazione".



(fonte: "RFC: Total Cost of Ownership for Linux in the Enterprise" - luglio 2002). Poiché le dimensioni dei progetti incontrati sono naturalmente molto diverse tra loro, i dati sono stati rapportati ad una soluzione che consente 100.000 accessi giornalieri ai servizi web.

All'incirca dal 1998, il movimento OPEN SOURCE ha iniziato ad assumere proporzioni talmente significative da non poter più essere considerato conseguenza dell'aggregazione di alcuni sostenitori fanatici. Nell'ultimo periodo, l'aumento della credibilità del paradigma ha ricevuto una spinta considerevole dall'ingresso in questo mondo di diverse società, tra le quali Dell, Sun, HP ed IBM. Il contributo maggiore è stato portato proprio da quest'ultima, notoriamente fautrice del software proprietario.

Per alcuni, il software libero rappresenta una minaccia verso il mondo "proprietario" e "commerciale". Il 31 ottobre 1998, un documento interno della Microsoft trapelò all'esterno, per opera

Ogni anno, più o meno nello stesso periodo, trapela all'esterno qualche documento simile al primo. Sembra che anche nel novembre 2002 Microsoft abbia preparato l'ormai annuale testo. I contenuti, teoricamente segreti, sono arrivati a Raymond (lo stesso diffusore del primo Halloween Document), il quale ha lasciato trapelare alcune indicazioni ed ha tratto la conclusione che le discussioni relative ai diritti sulla proprietà intellettuale hanno fatto molto parlare dei prodotti liberi, apportando un importante vantaggio alla comunità. Gli attacchi di Microsoft al mondo Open Source non solo sono falliti, ma hanno contribuito a diffonderne la conoscenza e a consolidarne l'esistenza. Non vale la pena per la comunità spendere energie per fronteggiare gli attacchi.

Da un'inchiesta eseguita su un campione di 100 IT manager di grosse multinazionali americane, emerge che LINUX è presente nel 39% di queste, con un numero di applicazioni sempre in crescita. La minaccia al software proprietario è dunque

effettivamente reale.

Il Dipartimento della Difesa Statunitense ha approvato l'uso di software libero in alternativa a quello proprietario, asserendo che questo tipo di applicazioni gioca un ruolo molto più importante di quello che viene percepito generalmente all'interno del Dipartimento stesso. Qualcuno ha invitato i membri del Dipartimento a non formulare apprezzamenti non richiesti e pericolosi nei confronti del software non proprietario; qualcun altro lo ha messo in guardia rispetto al rischio di possibili attentati: essendo a disposizione i sorgenti, qualsiasi malintenzionato può eseguire manomissioni. Quest'ultima ipotesi è fugata dai fautori dell'Open Source, che dimostrano invece come la conoscenza del codice sorgente, oltre alla sua diffusione, possa fare individuare i difetti e le manomissioni in tempi assolutamente non raffrontabili con quelli propri del software commerciale. E' stato infatti dimostrato come il tempo necessario per l'individuazione e la correzione di un errore in un software Open Source sia inferiore di almeno un ordine di grandezza rispetto a quello legato al mondo proprietario, e questo contribuisce alla sicurezza di un corretto funzionamento ed alla tempestiva individuazione di errori e di eventuali manomissioni.

Non è solo LINUX a minacciare il mondo Microsoft: ci sono anche le applicazioni di tipo office, destinate ai personal computer, gli strumenti collegati ad Internet, gli "application server", i programmi utili per le applicazioni di e-commerce.

La comunità LINUX non desidera rimpiazzare Microsoft facendo il suo stesso gioco. Desidera cambiare le regole del gioco. Desidera che il software sia conoscenza, che ci sia maggiore libertà. Il problema non è tanto chi sarà ad avere la supremazia, quanto poter avere libertà di scelta. Oltre ai movimenti già visti, esistono oggi molti gruppi che si impegnano per far sì che possa progredire un mercato aperto e libero. Diverse associazioni si battono affinché il software libero venga adottato nelle Pubbliche Amministrazioni e nelle scuole. Sotto queste spinte, anche la Comunità Europea e la nostra Pubblica Amministrazione hanno analizzato le nuove opportunità, per eliminare la dipendenza dall'unico fornitore attuale e ridurre i costi. Ragioni politiche consigliano di non abbandonare in mano ad un

unico operatore, privato e straniero, tutte le infrastrutture di comunicazione. Nel giugno 2002, il Ministro per l'Innovazione e le Tecnologie ha diffuso un documento denominato "Linee guida del Governo per lo sviluppo della società dell'informazione" in cui indica:

"Si diffonderanno gli standard aperti ed i software open source, cioè i software liberi, la cui proprietà non sia di un singolo fornitore ma governati da una licenza d'uso (libera) che ne garantisce la possibilità di libero utilizzo, scambio, studio e modificabilità." Microsoft ha reagito inviando al Parlamento, il 20 giugno 2002, una nota di cui riportiamo un passo significativo:

"L'innovazione del software è una forza motrice per il progresso economico, sociale e tecnologico. Permettere la competizione, sulla base dei propri meriti, tra lo sviluppo di software molteplici, di business e modelli soggetti a licenza (chiusa) costituisce sicuramente il modo migliore per promuovere l'innovazione del software e per assicurare ai clienti - privati e pubblici - un'ampia scelta nelle decisioni di acquisto del software".

Il movimento Open Source non nega alcuna delle libertà descritte nella nota, ed anzi, basa la sua battaglia anche sulla competizione e sui meriti. Anche il Giappone sta prendendo in seria considerazione la migrazione totale dai sistemi operativi Microsoft verso altre soluzioni, ed in particolare verso LINUX. I costi di licenza inesistenti e la disponibilità di sorgenti da adattare sono le maggiori attrattive, uniti alla sfida verso il monopolio di Microsoft, ed al desiderio di riportare competitività, di ricreare situazioni e fervore simili a quelli che hanno portato allo sviluppo di UNIX e di LINUX. Un altro fattore di successo del software libero, non ispirato a principi etici o filosofici, è la ricerca di un'alternativa che possa contribuire a salvaguardare gli investimenti.

Il predominio di Microsoft obbliga gli utilizzatori ad una continua rincorsa per l'aggiornamento dei propri sistemi. I nuovi prodotti rilasciati da Microsoft, o i nuovi hardware, funzionano solo con le versioni più recenti di WINDOWS. Questo obbliga ad acquisire continuamente gli aggiornamenti, il cui prezzo è mediamente pari a metà del prezzo d'acquisto della licenza, mentre in ambito LINUX per ogni nuovo dispositivo è disponibile il relativo driver, anche per le versioni più vetuste del sistema

operativo, senza considerare che l'aggiornamento di LINUX non costa nulla.

Un fornitore monopolista può realizzare applicazioni maggiormente integrate con il suo sistema operativo, proprio in quanto unico vero conoscitore dello stesso; può addirittura modificare il sistema operativo a favore delle sue applicazioni e a discapito di quelle di sviluppatori concorrenti. E' necessario evadere dal modello dove il produttore del sistema operativo ha il controllo sia sul sistema operativo che sull'hardware (come avviene con Microsoft, grazie a taciti accordi con Intel) e su tutti gli strumenti che colloquiano con il sistema operativo stesso. Questa è una minaccia per gli sviluppatori di software applicativo, che rischiano di ritrovarsi improvvisamente con soluzioni non funzionanti su nuove versioni di sistema operativo.

La vera caratteristica della libertà è l'indipendenza, mantenuta con la forza.

Voltaire

Rispetto ad altri sistemi operativi proprietari, la scelta di LINUX è giustificata anche dal fatto che lo stesso software applicativo, come un ERP (Sistema Informativo Gestionale Integrato), può essere portato su hardware diversi. Questo consente di disporre sempre del meglio in termini di tecnologia hardware, salvando gli investimenti riguardanti l'acquisto di software, lo sviluppo di software personalizzato e la formazione degli utenti, capitolo quest'ultimo di notevole peso nei progetti.

7. I MODELLI DI BUSINESS PER I FORNITORI DI PRODOTTI OPEN SOURCE

Quali sono i profitti per le società che hanno fatto dell'Open Source il loro business? Può un progetto Open Source essere remunerativo?

I primi programmatori pensavano che l'etica degli utilizzatori potesse essere sufficiente per riconoscere il valore di quanto utilizzato e devolvere liberamente agli autori quanto reputavano congruo per finanziarli. L'ipotesi non si è mai concretizzata. Le aziende che vivono con l'Open Source basano i loro profitti sulla vendita di distribuzioni e sulla

fornitura di servizi. Gli utenti, nonostante il software sia disponibile liberamente in Rete, apprezzano le raccolte già preconfezionate, suddivise per argomenti ed interessi, i programmi già compilati per un dato sistema, autoinstallanti, e sono disposti a pagare i pochi dollari richiesti. Tuttavia, poiché i margini sono minimi, per realizzare un profitto è necessario raggiungere un pubblico molto vasto. Le attività di consulenza, dal supporto alla personalizzazione dei programmi a quello necessario per formare gli utenti, costituiscono l'altra componente significativa dei ricavi. Molte aziende offrono corsi specifici, fino ad arrivare alla certificazione degli specialisti ed al supporto alle imprese per la migrazione dei propri applicativi sul sistema operativo LINUX.

A differenza di quanto avviene per il software proprietario, una buona parte del budget di investimento per il sistema informativo è spostata dall'acquisto di hardware e di licenze software a quello della formazione. Questo introduce maggiore flessibilità nella gestione del budget, in quanto i servizi, a differenza delle licenze, possono essere acquistati solo quando di proprio interesse e contrattati con la società erogatrice.

Va notato che la licenza GPL nega la possibilità di vendere licenze chiuse basate su software acquisito con la stessa licenza. Nessun autore potrà basare il proprio business sull'acquisizione di programmi liberi basati su questa licenza, sulla loro modifica, sull'integrazione in una propria soluzione chiusa e sulla vendita della licenza d'uso "chiusa". Ciò che è stato acquisito libero, deve essere diffuso come tale e non può essere utilizzato in contesti che ne limitino la libertà.

8. QUALI SONO I BENEFICI POSSIBILI PER LE IMPRESE ITALIANE?

Fino a questo punto, abbiamo esplorato i modelli di sviluppo del software libero, parlando sempre di sistemi operativi, strumenti per il web, per Internet, per i programmatori.

Per le imprese la scelta del sistema informativo utile per la propria gestione è stata fino a poco tempo fa basata quasi esclusivamente sul sistema di gestione dell'impresa ("ERP") e non sul sistema operativo o sul software di rete.

I quesiti da porsi ora sono almeno tre:

- Ha senso valutare la scelta di una soluzione

Open Source, ma non solo... La via di DATA CONSULT Sistemi Informativi nel mondo degli ERP

Gli ideali di libertà, passione ed innovazione propri delle comunità di sviluppo del software libero ispirano da sempre il nostro agire e più volte ci hanno indotto a chiederci se il modello Open Source fosse applicabile ai sistemi ERP. La risposta più naturale sembra essere un secco no, confermato anche dal fatto che non esistono soluzioni Open Source che possano essere classificate come ERP e non si intravedono evoluzioni significative in tal senso. I motivi tecnici sono diversi e conosciuti: complessità, necessità di integrazione tra i componenti, lunghi tempi di sviluppo, ingenti investimenti e complessità di collaudo.

Un'analisi più approfondita, e meno tecnica, fa emergere anche altri significativi elementi che limitano l'applicazione degli ideali sopra citati al mondo delle soluzioni gestionali per le imprese. I grandi produttori ERP internazionali utilizzano infatti ambienti di sviluppo creati ad hoc per realizzare le loro applicazioni. Di conseguenza, l'estensione delle soluzioni e l'utilizzabilità di questi strumenti sono ristrette al sistema per il quale sono state sviluppate. Come si applica allora il concetto di libertà ai sistemi informativi aziendali? E soprattutto, come si configura nel caso di imprese che si sanno affermare, il più delle volte anche a livello internazionale; che sono sufficientemente organizzate per non essere più "grossi artigiani"; che hanno quel minimo di struttura che, unita alla capacità ed all'inventiva, le rende protagoniste riconosciute (esprimendo la loro identità in prodotti e servizi solo apparentemente di nicchia)? Come si delinea insomma per quelle aziende che rappresentano la forza propulsiva del nostro sistema-paese?

Da quasi 25 anni, queste imprese rappresentano i nostri clienti: aziende che ci chiedono soluzioni per gestire, valorizzare, esaltare la loro tipicità, la loro capacità di proporsi sul mercato con prodotti e servizi eccellenti, soluzioni che in un sistema precostituito non possono esistere.

Per queste aziende, libertà non significa adottare un ERP internazionale, le cui logiche architetturali si basano su concetti quali: "il nostro sistema copre tutte le esigenze", "il nostro sistema è garanzia di integrazione tra tutti i suoi componenti", "noi conosciamo le best-practices delle più grandi aziende del mondo e le abbiamo inserite nel sistema". Questo approccio è valido in termini di

qualità, coerenza e disponibilità di funzioni solo per quelle strutture in cui il vantaggio competitivo è basato su efficienza, economie di scala, ottimizzazione, omogeneità, standardizzazione, controllo dell'organizzazione e dei processi ed imposizione di disciplina. La libertà viene quindi limitata alla scelta di cosa utilizzare tra ciò che è già presente e consiste nell'adeguarsi alle modalità previste da chi ha progettato il sistema (o nel percorrere l'onerosa e classica strada di realizzare software ad hoc). Per le medie imprese libertà significa invece disporre di soluzioni complete ed articolate, ma al tempo stesso facilmente utilizzabili, proprio per non costringere l'azienda a bruciare le proprie energie nell'implementazione e nella gestione del sistema. Al tempo stesso, significa avere la flessibilità e le potenzialità necessarie per adattarsi velocemente ed efficacemente alle criticità ed alle peculiarità attuali e future.

Se questa libertà non si ritrova nelle soluzioni "dei grandi" non si ritrova nemmeno nelle soluzioni "home made" (basate su architetture più o meno dipendenti da produttori di strumenti di sviluppo) o in soluzioni apparentemente nuove, ma basate su infrastrutture ormai datate. Le soluzioni intermedie poi (gli ERP italiani), se pur funzionalmente valide, generalmente limitano la libertà al "personalizzarsi o farsi personalizzare le applicazioni", uscendo una volta per sempre dallo "standard", accollandosi l'onere della realizzazione, della qualità tutta da dimostrare, della necessità di riadattamenti continui alla nuove release del prodotto. Non è così per DATA CONSULT. Libertà allora non significa poter disporre di un software in formato sorgente, ma avere a disposizione un sistema costantemente aggiornato ed innovato da una struttura che trova la sua stessa ragione di esistenza nel fornire soluzioni complete, tecnologicamente evolute, che garantiscono al cliente indipendenza nell'estenderne le funzionalità.

Da sempre, per DATA CONSULT libertà è MAGIA. MAGIA (Moduli Applicativi per la Gestione Integrata Aziendale), giunto alla terza generazione, offre alle aziende la concreta possibilità di rompere con il passato, di collocare realmente il sistema informativo tra i fattori strategici di successo. I concetti di sistema aperto, indipendenza, utilizzo di strumenti standard fanno parte del codice genetico

di MAGIA: una dimostrazione per tutte, MAGIA è su LINUX.

Ma la vera libertà è qualcosa di più: noi garantiamo alle aziende un ottimale, veloce, sicuro e completo utilizzo delle applicazioni ed al tempo stesso l'implementazione di quelle funzioni o modifiche atte a supportare i fattori distintivi d'impresa.

Offriamo la possibilità di realizzare queste estensioni, meglio definite "specializzazioni", in modo autonomo, con il nostro supporto o con quello di terze parti (ad indiscutibile scelta dei clienti), estensioni che non richiedono costosi adattamenti alle successive release, ma, anzi, possono automaticamente sfruttare le nuove potenzialità introdotte nella soluzione standard.

Anni di investimenti e di ricerca ci hanno condotto alla riprogettazione della nostra soluzione con logiche e strumenti realmente Object Oriented, dando in questo modo vita a [magia]³.

In questo caso, la libertà di "specializzazione" si concretizza attraverso strumenti assolutamente standard (Java, XML), di pubblico dominio e di ampia diffusione, che sono parte del bagaglio culturale di ogni professionista del mondo informatico.

[magia]³ è garanzia di qualità, continuità, innovazione e stimolo alla ricerca. Allo stesso tempo, permette di ospitare nella medesima piattaforma soluzioni di sviluppo collaborativo.

Componenti e strumenti di corredo al sistema ERP, specializzazioni e integrazioni con altri software, il cui codice sorgente è a disposizione della comunità degli utilizzatori di MAGIA, possono essere sviluppati, estesi e migliorati sia con il contributo di DATA CONSULT che con quello dei propri clienti. Per questo promuoviamo una comunità di professionisti attenti, attivi, interessati ed aperti che sanno apprezzare, migliorare ed estendere quanto mettiamo a disposizione e che non mancheranno di essere loro stessi portatori di nuove soluzioni.

[magia]³: un superamento della contrapposizione tra software commerciale ed Open Source, un modo nuovo per massimizzare gli sforzi e per trovare insieme con i nostri clienti nuove opportunità per far crescere le nostre aziende e le persone che operano all'interno di queste.

basata su server o infrastrutture dotati di sistema operativo libero?

- Cosa fare per i posti di lavoro? Esiste veramente l'alternativa a WINDOWS?
- Cosa fare per l'ERP, il sistema integrato di gestione aziendale?

Per la prima domanda, il lettore giunto sino a questo punto della trattazione, avrà già trovato le risposte. Un sistema operativo Open Source offre l'indipendenza dall'hardware e dal relativo fornitore, riduce i costi di acquisto, offre maggiori garanzie di sicurezza ed affidabilità, mette a disposizione una serie di strumenti per la gestione ordinaria che una soluzione commerciale non è in grado di proporre (strumenti per le attività di backup, strumenti per il web, sistemi di posta, firewall, solo per citare alcune soluzioni). La scalabilità, ossia la possibilità di passare ad hardware superiori o diversi, salvaguardando gli investimenti nel software e nella formazione, è un ulteriore vantaggio. Non meno importante è la constatazione che le competenze necessarie per la gestione dei sistemi sono più facilmente reperibili, a differenza di quanto avviene per i sistemi proprietari.

Non è oggi possibile fornire un'unica risposta alla seconda domanda. Molte soluzioni applicative utilizzate nelle imprese, i cosiddetti strumenti di 'front-office', richiedono ancora l'uso di WINDOWS. Molti fornitori stanno rapidamente migrando le loro soluzioni verso LINUX o verso applicazioni basate esclusivamente su Web, svincolandosi quindi da un sistema specifico. Alcuni produttori di PC hanno annunciato l'imminente offerta di macchine già pronte per l'uso con installato LINUX ed un'applicazione tipo-office.

L'appiattimento delle conoscenze informatiche, conseguente all'introduzione del sistema operativo WINDOWS, ha portato l'utenza ad identificare univocamente il Computer con il Personal Computer, il sistema operativo con WINDOWS, il programma utile per scrivere testi e quello per gestire un database con Microsoft Office. Va precisato che a questo hanno partecipato anche le scuole, le università, la stampa, ma ora, per fortuna, la tendenza è invertita. A differenza di quanto si crede, esistono da sempre alternative valide al più diffuso sistema di scrittura di testi, ed oggi ne esistono di Open Source che garantiscono la compatibilità e quindi lo scambio di testi con Word.

Prima di selezionare il sistema operativo dei "Client" (posti di lavoro), è indispensabile verificare se le applicazioni utilizzate richiedono ancora l'uso di WINDOWS. Per le applicazioni di tipo office, va valutato l'impatto per gli utenti finali, ma anche il risparmio derivante dalla mancata necessità di acquisire licenze commerciali e di sostenere i continui aggiornamenti dell'hardware e del software richiesti da WINDOWS e da Office. Va notato che esistono soluzioni di tipo office libere anche per WINDOWS, che consentono di fare il primo passo verso il mondo libero.

Ultimo, ma non per importanza e complessità, è il sistema di gestione dell'impresa, l'ERP. E' pensabile adottare un ERP libero, o avviare un progetto libero per lo sviluppo di un nuovo ERP? Una società che ha già sviluppato un proprio prodotto può affidarlo alla comunità per la manutenzione e rendere libere le licenze? Quali potranno essere i ritorni?

Gli aspetti da considerare sono molteplici. Alcuni sono legati alla natura dei problemi risolti dagli ERP, altri alla cultura degli sviluppatori, altri ancora alle caratteristiche stesse degli ERP.

Riservatezza

Le maggiori criticità relative alla riservatezza sono collegate al fatto che nell'ERP sono spesso racchiuse le informazioni proprie di un'azienda: i sorgenti dell'ERP sono una parte del sistema di conoscenza e di descrizione dei processi dell'azienda stessa. Affidare l'ERP alla comunità significa rendere espliciti i propri peculiari processi di produzione e la propria organizzazione, esplicitare insomma importanti dettagli sui quali appoggia il vantaggio competitivo dell'impresa. Molto diverso è il caso di un produttore di hardware, che non ha alcun interesse nel mantenere riservate le specifiche dell'hardware e del relativo software: il rischio di plagio è annullato dai vantaggi derivanti dalla tempestività di fornitura della soluzione, soluzione destinata comunque a durare sempre pochi mesi. E' più proficuo per il produttore convogliare le proprie risorse verso l'individuazione e progettazione di nuove soluzioni che verso la protezione di quelle ormai realizzate. Analogo paragone può essere trovato con i sistemi operativi: i concorrenti sono pochi e nessuno è interessato oggi a sviluppare un nuovo sistema. Le conoscenze tecniche in merito allo sviluppo di sistemi operativi, di linguaggi di programmazione e di protocolli di

Rete sono diffuse e la protezione dei sorgenti non è sufficiente per far fronte alla concorrenza.

L'offerta degli ERP è ancora vasta, e ci sono attori interessati a carpire la struttura di soluzioni esistenti per migliorare il proprio vantaggio competitivo sulla concorrenza.

Dimensioni del software

Non sono solo i problemi legati alla riservatezza a sconsigliare l'approccio 'open source'.

Il progetto GNU per la realizzazione di un ERP libero, denominato "GNU ENTERPRISE", è stato avviato nel 2000. Si trova ancora nella fase iniziale, ha accolto pochissime adesioni, nessun modulo applicativo è ancora stato rilasciato. Sulla Rete, si trovano disponibili poche soluzioni denominate ERP, liberamente scaricabili. Le funzionalità offerte da questi programmi, per lo meno quelli destinati al mercato italiano, coprono solo alcune aree e per poter essere utilizzate necessitano di ulteriori sostanziose implementazioni e personalizzazioni. Per analizzare questo 'insuccesso' è necessario un raffronto tra le caratteristiche di un ERP e quelle di un sistema operativo.

LINUX è costituito da un "micro-kernel", un nucleo piccolo al quale sono collegate centinaia di funzioni o pacchetti che svolgono ognuno un compito preciso. Ogni pacchetto può vivere anche una vita autonoma, non collegato agli altri. Queste caratteristiche, nate da una corretta visione di Torvalds, sono applicabili a qualsiasi software tecnico. Un ERP è invece un sistema complesso costituito da innumerevoli programmi tra loro strettamente connessi per supportare tutti i processi di un'impresa. Così, il programma di gestione delle scorte non può vivere di vita propria se non conosce esattamente i comportamenti dei programmi di gestione del magazzino, della produzione, delle vendite.... Le dimensioni di ogni programma sono sempre molto significative: la scrittura di un programma di gestione della movimentazione di magazzino può impegnare un programmatore a tempo pieno per mesi. Affinché un progetto affidato alla comunità abbia successo, è indispensabile che sia scomponibile in frazioni minime, tenendo presente che i programmatori liberi sono tanti, ma che ognuno dedica poche ore del proprio tempo alla settimana per lo sviluppo Open Source.

Complessità del collaudo

Una componente di un sistema operativo è scritta

sempre per uno scopo preciso e limitato. Affidare una componente, non appena scritta, alla comunità di utenti (costituita da programmatori e sistemisti) consente di concludere tempestivamente il test, attraverso l'utilizzo in ambienti diversi. I collaudatori sono tecnici, che hanno da una parte l'interesse a comprenderne il funzionamento, dall'altra l'interesse ad utilizzare subito la soluzione. Gli errori presenti nel software di questo tipo emergono sempre in modo esplicito, impedendo l'uso della funzione errata, o, nei casi peggiori, causando danni che sono immediatamente evidenti e quindi neutralizzabili e riparabili.

Nessun utente di un ERP ha interesse a provarne una sola parte. Una sola funzione gli è di utilità solo quando è completa ed è inserita in un contesto preciso. Per esempio, per l'utente non ha alcun senso provare la stampa dei libri contabili se non è in grado di inserire le registrazioni relative.

I tempi necessari per i test e le modalità con cui gli errori si possono individuare sono ben diversi rispetto a quanto richiesto dai programmi tecnici! Un errore in una funzione di inserimento dati può continuare a produrre i suoi danni per mesi, memorizzando per esempio valori errati, prima che gli effetti collaterali generati possano manifestarsi in modo evidente. E quando l'errore si manifesta, probabilmente i danni sono già irreversibili. Proprio per evitare situazioni del genere, è necessario pianificare un test esaustivo prima di rilasciare il software anche in versione "beta" a potenziali utenti. I costi per il test di un ERP superano spesso i costi sostenuti per la scrittura del codice.

Interfacce e progettazione

Gli hacker sono interessati a programmare bene ed ad "aprire" qualsiasi giocattolo (software) a loro disposizione per capirne il funzionamento, modificarlo, migliorarlo. Il software con il quale lavorano è in genere privo di interfacce utente, o al massimo dotato di interfacce molto semplici. Gli hacker conoscono i loro strumenti e vi accedono parlando il linguaggio degli strumenti stessi, attraverso la digitazione di comandi piuttosto che attraverso l'uso di mouse, icone, finestre. Non hanno alcun interesse a capire le esigenze dell'utente non tecnica, molto diversa da loro stessi. L'eleganza e la validità della soluzione, in ambito tecnico, coincide con l'eleganza della stesura del codice, ma nell'ambito delle soluzioni applicative

l'eleganza della soluzione equivale anche alla semplicità d'uso ed alla pulizia delle interfacce. Per questo motivo, le interfacce dei programmi desktop di LINUX non eguagliano quelle di WINDOWS. Per una soluzione applicativa, è necessaria un'accurata progettazione delle interfacce di input e di output, omogenee per tutto un ERP. La progettazione è eseguita da analisti in grado di comprendere da una parte le esigenze di facilità di comunicazione, dall'altra le problematiche da risolvere. Le competenze a loro richieste sono non solo tecniche, ma anche di organizzazione d'impresa, piuttosto che fiscali. Il ciclo di sviluppo rimane quello classico di analisi, progettazione, codifica, test, e coinvolge diversi reparti (dal marketing alla ricerca e sviluppo, dalla direzione tecnica all'assistenza clienti). Le fasi possono essere abbreviate fino ad eseguirle in parallelo, ma non possono essere eliminate.

Scadenze

La comunità di sviluppo Open Source non rispetta alcuna data di scadenza. Nel mondo Open Source il tempo non è denaro... L'uscita di uno strumento può non coincidere per nulla con la data in cui il mercato l'attende. Ogni cosa è rilasciata appena si ritiene sia usabile. Il regime fiscale, in continua evoluzione, e quello italiano in particolare, impone agli utilizzatori e quindi ai progettisti di cambiare software talvolta nel giro di pochi giorni. Gli sviluppatori di software commerciale sono organizzati per gestire questi eventi, ed una buona parte del prezzo riconosciuto per le licenze e per la manutenzione è destinato a coprire questi costi di organizzazione.

Dimensioni della comunità

Abbiamo già visto che affinché un progetto Open Source abbia successo è necessario che venga generata una massa critica di utenti-programmatori con la stessa esigenza da soddisfare. Se questo è vero per gli strumenti destinati a tecnici e programmatori, è pressoché impossibile che sia concretizzato per utenti di applicativi di gestione aziendale.

Per l'Europa, la 'localizzazione', ossia l'adeguamento del software alle diverse lingue e normative fiscali, impedisce ad un progetto di software applicativo, riguardante per esempio un ERP, di estendersi oltre ai confini nazionali, a differenza di quanto accade negli USA.

Le PMI italiane sono caratterizzate da dimensioni ridotte, flessibilità (e quindi ampie opportunità di sviluppo) e da fattori distintivi che sono la fonte del loro vantaggio competitivo. Sono per questo basate su processi ed organizzazioni molto peculiari, che richiedono, in ogni progetto di inserimento di un ERP, una quantità consistente di personalizzazioni ed adattamenti specifici, esclusivi per una singola impresa: la comunità Open Source può fare ben poco di fronte a questa esigenza. E' impensabile che un unico programmatore, sempre che sia a disposizione nell'impresa, possa affrontare da solo un intero progetto di personalizzazione del software e del suo mantenimento, con gli strumenti oggi a disposizione.

Riassumendo, quindi, per i programmi applicativi necessari alla gestione delle aziende italiane, la condivisione delle esigenze non è significativa per generare un movimento sufficientemente ampio. Inoltre, l'utente di un ERP non è un tecnico, in grado di intervenire sui programmi. Se questo problema fosse superabile, il software proprietario già esistente potrebbe, teoricamente, essere reso libero. I produttori di software non potrebbero in ogni caso sostenere da soli gli alti costi richiesti per lo sviluppo e basare il loro business esclusivamente sulla vendita dei servizi, essendo il loro mercato molto limitato.

A meno che non si raggiungano dimensioni di mercato estremamente ampie, come è avvenuto per le società che distribuiscono LINUX o che vendono servizi collegati allo stesso (REDHAT, SUSE, LINUXCARE), appare ormai chiaro che l'offerta di servizi legati ad un prodotto Open Source da sola non può portare al profitto una società. In alternativa, i servizi dovrebbero essere erogati a costi tali da coprire anche gli oneri derivanti dallo sviluppo e dalla manutenzione del software.

Conclusioni

Per i programmi di gestione integrata aziendale, destinati alle PMI, non esistono oggi soluzioni Open Source complete e non sembra possa apparire nulla di nuovo all'orizzonte. Anzi, Microsoft sta tentando di estendere la sua supremazia anche nel settore "business solutions", accelerando i tempi tramite l'acquisizione di società che operano nel settore, come la danese Navision.

Il movimento Free Software indica: "La libertà consiste nell'essere in grado di prendere decisioni

Un'esperienza di libertà

Quella che segue è una breve sintesi della storia di un prodotto e della nascita di una comunità.

La storia ebbe inizio nel 1981, quando un notaio ci affidò l'incarico di supportarlo nell'informatizzazione del suo studio. Fu formulato un progetto per automatizzare tutte le funzioni. Dopo poco tempo, fu installato il primo modulo software, su hardware PDP11/23 della Digital. Nel giro di pochi mesi l'informatizzazione era completa ed il notaio era già in grado di valutare il ritorno sugli investimenti. La soluzione era così pronta per essere applicata anche presso altri studi. Il software fu portato successivamente su macchine di caratteristiche superiori e, alla fine degli anni '80, su sistema operativo UNIX, introducendo quindi la possibilità di scegliere liberamente server di fornitori diversi (IBM, HP...).

Nel 1991, il prodotto fu portato in ambiente X-WINDOWS e fu introdotto il database relazionale per l'archiviazione e la gestione dei dati. Era nato così SINFON90 (Sistema Informativo Notarile per gli anni 90), primo programma notarile ad utilizzare terminali grafici e mouse e a conservare i dati del cliente in un database liberamente accessibile e, quindi, a completa disposizione per qualsiasi consultazione. Sempre nel 1991, un gruppo di notai della Lombardia affidò ad un docente del Politecnico di Milano (il prof. Augusto Celentano) l'incarico di selezionare il prodotto che fornisce le migliori garanzie in termini di affidabilità e completezza funzionale. Dopo un'attenta analisi comparativa, fu selezionato SINFON90. Lo stesso gruppo, guidato da un notaio

definito "il padre" dell'informatica notarile (il Dr. Angelo Gallizia), pose un requisito: la società uscente dalla selezione avrebbe dovuto realizzare il "compositore", ossia uno strumento che permettesse al notaio di scrivere gli atti secondo il proprio stile e secondo la propria interpretazione delle norme vigenti, garantendo nel frattempo la sicurezza dell'atto finale attraverso controlli da lui stabiliti e ripetuti dal sistema per ogni nuovo atto; il tutto avrebbe dovuto funzionare a regime senza necessità di ricorrere al fornitore del software per ogni modifica.

I notai committenti fondarono l'Associazione degli Utenti di SINFON, allo scopo di condividere le esigenze, di comunicare unitariamente con i fornitori, e soprattutto di sviluppare una libreria di schemi d'atto comune. Il modello dell'organizzazione è quello delle comunità di sviluppo del software libero: gli schemi che descrivono le regole seguite per redigere gli atti sono realizzati da un notaio, che ha in quel momento un'esigenza specifica, e sono quindi messi a disposizione di tutta la comunità.

Dopo dieci mesi dal ricevimento dell'incarico, in anticipo sui tempi pattuiti, il compositore fu rilasciato e superò brillantemente il collaudo. Ancora oggi, lo stesso compositore è al centro della soluzione.

Dal 1999, SINFON funziona anche in ambiente LINUX. Questa soluzione, dimostratasi assolutamente sicura ed affidabile, sta sostituendo via via tutte le precedenti. Nel 2001, l'applicazione è stata resa accessibile da web.

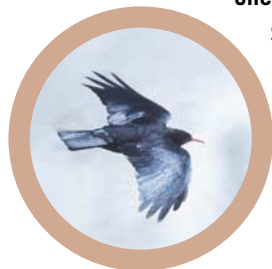
La "Comunità di Utenti di SINFON"

continua la sua missione. Ora che Internet è entrato negli studi notarili, i notai utilizzano il web per aggiornare e condividere gli "schemi d'atto". Senza averne la consapevolezza, da oltre 10 anni stanno utilizzando con successo lo stesso modello di organizzazione della comunità di sviluppo del software libero. Come avviene nella libera comunità di sviluppatori software, nella Comunità di Utenti di SINFON ogni notaio porta il suo contributo nelle aree di maggior interesse. Soprattutto negli ultimi mesi, i notai utenti di SINFON hanno percepito il grande valore della loro comunità, in quanto nei prossimi mesi il lavoro che li attende, tutto da condividere, sarà molto significativo, a causa delle rilevanti modifiche che verranno apportate alle norme del diritto societario. Grazie alla costante attenzione verso la sicurezza, l'affidabilità e la libertà del cliente, la storia continua...

(per chi desidera saperne di più: utentisinfon@itnotai.it)

Bibliografia

Per chi desideri approfondire gli aspetti filosofici che sottendono ai movimenti Free Software ed Open Source, citiamo:



"La Cattedrale ed il Bazaar" ed "Il Calderone Magico", di Eric Raymond (diverse traduzioni in italiano sono disponibili su Internet)

www.opensource.org: sito del movimento Open Source

www.fsf.org: sito del movimento Free Software

www.linux.org: informazioni generali, documentazione, curiosità, software e collegamenti a risorse disponibili per LINUX

che influiscono principalmente su se stessi". Ed ancora: "Le discussioni sui diritti e sulle regole relative al software si sono spesso concentrate sugli interessi dei soli programmatori".

Esistono alternative in grado di supportare il primo assunto e smentire il secondo?

Nel caso di soluzioni applicative, libertà significa indipendenza dal fornitore del prodotto e dai fornitori dell'hardware. Significa poterne ampliare le funzionalità, poterne modificare i comportamenti, senza dover ricorrere necessariamente all'autore dei programmi. Significa non dover rifare tutte le personalizzazioni ad ogni nuova release del software rilasciata dal fornitore.

Tre generazioni di MAGIA (Moduli Applicativi Gestione Integrata Aziendale) sono testimonianza che questa strada, anche se molto ardua, è percorribile.

Una delle maggiori preoccupazioni dei fautori del software libero è quella di assicurare ai clienti la continuità nella manutenzione. Il quesito posto è: cosa accade se il fornitore del software abbandona la manutenzione del prodotto, o, ipotesi peggiore, cessa l'attività? Per il software applicativo, il cliente può tutelarsi attraverso specifici contratti. Nei contratti proposti da DATA CONSULT, la stessa si impegna a mantenere il prodotto, il quale evolve continuamente in nuove release. Nel caso di cessazione dell'attività, i sorgenti sono messi a disposizione dei clienti, senza limitazione alcuna. E' quello che nelle comunità Open Source è stato definito 'liberare il futuro', ovvero rendere disponibile il sorgente nel caso di inadempienza o fallimento del fornitore.

In merito all'accusa mossa verso il movimento Free Software, in relazione all'ipotesi che in fase di sviluppo vengono tutelati esclusivamente gli interessi dei programmatori (i quali realizzerebbero solo funzioni che richiedono loro il minimo sforzo ed investimento), questo è senza dubbio un principio che non si accompagna allo sviluppo di [MAGIA]3. I programmatori di [MAGIA]3 infatti sono impegnati nella realizzazione di strumenti utili all'utente per specializzare i comportamenti dell'ERP.

Rispettare l'interesse dei programmatori significa anche garantire un adeguato ambiente di lavoro affinché ogni individuo, nell'espletare compiti estremamente creativi, ed anche difficili, possa portare il proprio contributo e derivarne

gratificazione, secondo le stesse motivazioni che portano gli sviluppatori ad aderire alle comunità di sviluppo del software libero. E questo, come avviene nel caso di DATA CONSULT Sistemi Informativi, può trovare applicazione anche in società che non fanno esplicito riferimento ad un manifesto o ad un movimento, e che, per i motivi prima dimostrati, non possono distribuire e licenziare una soluzione ERP sotto la licenza "GPL".

9. EVOLUZIONI

Quali saranno le evoluzioni future del software libero? Per le aziende, è ormai chiaro che in merito alle tecnologie e alle infrastrutture di base la strada giusta è quella basata sull'Open Source.

Il software verrà considerato alla stregua dei risultati della ricerca, i quali vengono messi a disposizione di tutti. Internet consentirà la creazione di nuove comunità. Il modello della comunità verrà applicato anche in altri settori e non solo in quello del software. Rimanendo nell'ambito dell'informatica, il fenomeno LINUX ha fatto finalmente esplodere la consapevolezza che lo standard non è quello definito dal fornitore che ha avuto fino ad oggi la supremazia, ma è quello definito dalla comunità che lo ha accolto, verificato, fatto diventare tale. Il popolo di coloro che hanno solo adesso assunto questa consapevolezza, si sta ribellando verso quello che ora appare come un tiranno.

Oracle e Redhat estenderanno la loro alleanza anche per migliorare il desktop e realizzare un ambiente unificato dove far girare le applicazioni Oracle basate sul Web. Anche Suse ha annunciato il proposito di diffondere LINUX nell'ambito dei desktop. Altri fornitori hanno messo in atto iniziative analoghe. Il software applicativo e le soluzioni verticali resteranno in gran parte proprietarie. Il valore del software, degli algoritmi, supera il valore dei servizi forniti. La diffusione capillare, anche attraverso Internet, non è sufficiente per creare una massa critica di utenti disposti a pagare servizi che possano coprire da soli i costi di sviluppo. E non esistono utenti pari agli sviluppatori, che possano integrare e mantenere autonomamente il software. La scelta delle imprese verterà quindi verso qualsiasi soluzione in grado di offrire maggiore libertà e salvaguardia dei propri investimenti.