

OpenSource: non solo risparmio.

Mettiamo solide basi

Evitiamo fraintendimenti: OpenSource è una buona cosa, anzi ottima, e merita attenzione. Ma ci sono miti da sfatare e concezioni erranee da correggere. OpenSource, freeware, shareware, Creative Commons, No-Nonsense License Agreement, GNU, Trial, Personal Edition, etc.

Nell'immaginario collettivo equivalgono ad una sola cosa: programmi o informazioni da usare gratuitamente senza preoccuparsi della licenza. Semplice, ma anche sbagliato. Non è qui la sede per spiegarne le differenze, ma due cose vanno precisate:

- 1) Il termine OpenSource si applica solo quando i dettagli costruttivi sono completamente resi pubblici (ad esempio, nel caso di un programma è distribuito anche il codice sorgente; nel caso di un dispositivo hardware sono distribuiti anche gli schemi CAD/CAE¹ e la programmazione del firmware);
- 2) Tutto ha una licenza, che stabilisce cosa si può fare e cosa no, e non necessariamente OpenSource significa che se ne può fare qualunque utilizzo (ad esempio alcuni Sistemi Operativi OpenSource sono gratuiti solo per utilizzazioni non-Business).

Per non complicare, limitiamoci a parlare di programmi per computer, ma teniamo a mente che un discorso analogo vale per l'hardware (ad esempio il progetto OpenMoko per un cellulare OpenSource) e per i documenti.

OpenSource = Risparmio

Vero. Ma allo stesso tempo: falso. Vero quando penso a programmi per il mio PC di casa. Falso quando considero il TCO² per una piccola PMI³ con poche competenze IT. Vero per una PMI di media grandezza o per una grande impresa, che ha un suo supporto IT interno.

Avere il programma è una cosa, farlo funzionare, tenerlo aggiornato e avere supporto in caso di problemi è un altro discorso.

Il risparmio c'è, ma va verificato e calcolato bene per organizzarsi in modo da poterne realmente beneficiare. Spesso il calcolo del TCO considera solo il costo iniziale. Elemento importante, ma c'è dell'altro.

Requisiti hardware. Attenzione! Requisiti consigliati, non quelli minimi: una Ferrari funziona anche con un serbatoio da 5 litri, ma... Un PC per Microsoft Vista è ben più assetato di risorse che un PC per Linux, a parità di funzionalità. Questo è un risparmio.

Longevità del computer. Un programma con minori requisiti hw ci garantisce anche una maggiore usabilità nel tempo. Cioè dovremo cambiare HW dopo un numero di mesi maggiore. Questo è un risparmio.

Supporto. La comunità OpenSource è attiva, ma non abbiamo un contratto di supporto commerciale: non ci sono metriche di servizio. Non significa che i tempi di risposta siano più lunghi e/o che la qualità delle soluzioni sia inferiore. Spesso è vero il contrario, dato che chi ci lavora lo fa per passione. Ma se le priorità del team di sviluppo sono altre, il nostro caso non riceve alcuna risposta. Senza appello. Punto. Un esercizio di Risk Management: quanto ci possiamo permettere di aspettare quando ci serve aiuto? A sua mitigazione, c'è che alcuni programmi OpenSource offrono due livelli di supporto (spesso legati a distribuzioni di software differenti): quello gratis (a best-effort), ed uno a pagamento (con maggiori garanzie di supporto). Attenzione! Se è vero che in molti casi di software commerciale ci possiamo permettere di non leggere la licenza e le relative condizioni di supporto, questo è un rischio da non correre se si investe nell'OpenSource per uso non personale. Qui ci può quindi essere un costo.

¹ CAD/CAE = Computer Aided Design/Engineering

² TCO = Total Cost of Ownership

³ PMI = Piccola o Media Impresa

Aggiornamenti. I principali SW OpenSource sfornano con regolarità gli aggiornamenti maggiori. Ma questo non ci interessa. Il punto rilevante è con gli aggiornamenti minori (detti patch): risoluzione di problemi specifici, aggiunta di driver per nuovo HW, risoluzione di falle della sicurezza. La possibilità di richiederne ad hoc c'è solo se esiste un servizio di supporto a pagamento. Controlliamo la frequenza di emissione di queste patch poiché indicano la rapidità con cui verremo supportati nel caso che in programma presenti dei problemi. Anche a livello di sicurezza. L'aspettativa è che la loro frequenza sia almeno pari a quella del software commerciale corrispondente (e spesso addirittura migliore). Se il programma è usato per attività aziendali di rilievo, è meglio garantirsi l'emissione degli aggiornamenti optando per un supporto a pagamento. Qui ci può quindi essere un costo.

Costo della sicurezza. Abbiamo mai calcolato i costi dovuti ad intrusioni o danneggiamenti da parte di malintenzionati, o danni dovuti a malfunzionamenti? Ci sono sorprese rilevanti: costi diretti (ripristino della funzionalità compromessa, con relativi straordinari e/o acquisto di consulenze) e indiretti (eventuali spese legali verso terzi danneggiati, perdita di informazioni non recuperabili, danno di immagine, etc.). Il rischio informatico è una funzione del tempo: un programma che non è regolarmente aggiornato vede accrescere in modo esponenziale la probabilità di essere compromesso o creare malfunzionamenti. Quindi il costo di assicurarsi aggiornamenti va considerato alla luce del danno a cui si è soggetti. Qui c'è un costo immediato e un risparmio a medio termine.

Il costo del rischio.

Se facciamo bene i conti riusciremo a risparmiare un (bel) po' di Euro sia come investimenti One-Off che come Running Costs. Fermiamoci un attimo: ma è sufficiente per compensare i rischi di una transizione completa?

A leggere bene i nostri calcoli, il risparmio è garantito se passiamo completamente all'OpenSource. E allora bisogna ricordarsi che vanno aggiunti altri costi:

- 1) gestione del cambiamento (in termini di gestione della resistenza, molto onerosa in termini di tempo e risorse);
- 2) formazione del personale;
- 3) migrazione dei dati (personali e aziendali);
- 4) aggiornamento dei protocolli di interscambio di informazioni con clienti e fornitori;
- 5) misure di backup e fallback (nel caso che qualcosa vada storto);
- 6) gestione ed implementazione del progetto di migrazione (le cose non accadono da sole o possono essere portate avanti a tempo perso).

Il punto 6 è sistematicamente sottostimato, con conseguenze disastrose. Il problema è maggiore per le aziende, ma anche il privato può andare incontro a costosi salassi per ripristinare una situazione compromessa. Il periodo di transizione deve essere corto per minimizzare il danno sull'efficacia aziendale ed il progetto deve essere gestito in modo professionale, con adeguato empowerment e sufficienti risorse, altrimenti quello che deve essere un risparmio, diventa un costoso fallimento.

Ma allora perché OpenSource?

A questo punto la maggior parte delle persone abbandona l'idea.

Il risparmio è normalmente ancora garantito, ma i conti vanno fatti bene per prendere le giuste decisioni, e questo spaventa. In effetti la transizione non è una passeggiata, ma è una strada da intraprendere. E' chiaro: la molla del risparmio economico, per quanto grande possa essere, non è sufficiente. Quindi la domanda è: perché migrare?

Le risposte ad oggi sono 3:

- 1) Perché è sinonimo di innovazione efficiente e funzionale;
- 2) Perché è sinonimo di sicurezza;

- 3) Perché promuove una cultura di collaborazione ed interscambio di know-how che ne favorisce ritenzione ed accrescimento efficace.

Innovazione efficiente e funzionale

La produzione di SW commerciale è guidata da scelte di marketing, strategia, desideri interni all'azienda. Molto spesso gli User Requirements sono scritti all'interno dell'organizzazione stessa, con poca voce in capitolo da parte della comunità di utenti. Nel mondo OpenSource la situazione è esattamente il contrario: gli utilizzatori sono non di rado i primi ad aiutare ad implementare quelle funzionalità che ritengono utili. La loro realizzazione può inoltre contare su di una piattaforma potenziale di cervelli così ampia che nessuna azienda potrebbe schierare, permettendo una realizzazione ottimale ed innovativa.

Sicurezza

Dareste le password del vostro conto in banca ad uno sconosciuto visto alla TV che arriva in casa tutto mascherato? Magari è un'onestissima persona, ma chi lo garantisce? Per i programmi commerciali (lo dico senza criminalizzare) la situazione è analoga. Ci dobbiamo fidare che le informazioni che affidiamo loro saranno gestite secondo le nostre aspettative e senza errori. Ma non è sempre così: programmi che includono Cavalli di Troia o Backdoor o Spyware, o con falle che portano alla perdita di informazioni. In tutti i casi, il rischio che questo accada in programmi OpenSource è molto ridimensionato, dato il numero di persone indipendenti che guardano il codice per i fini più vari. Sento obiettare che è anche più facile per i malintenzionati scoprire falle da utilizzare. Vero, ma i programmi OpenSource sono quelli con la percentuale di vulnerabilità 0-day minore e il tempo di risoluzione più rapido. Inoltre, il caso di alcune case commerciali importanti che sono state soggette a vicissitudini anche giudiziarie per l'utilizzo di informazioni raccolte da alcuni loro prodotti, non permette di dire che il nome è sempre garanzia di trasparenza. E sicurezza e trasparenza vanno in coppia.

Cultura di collaborazione ed interscambio di know-how

E' il punto fondamentale: nell'era dell'informazione sempre più aziende hanno in quell'asset intangibile che è il Know-How il loro bene più prezioso (ed anche un rilevante *point-of-failure*). Spendono molto per garantirne la ritenzione, spesso concentrato com'è in poche persone: sistemi di KM⁴, aumenti salariali fuori scala, costosi corsi di aggiornamento, etc. Senza dimenticare il problema di copertura durante le assenze. Sappiamo tutti quanto è difficile riuscire a far condividere anche una minima percentuale di queste conoscenze.

Abbracciare la filosofia OpenSource significa passare da una cultura aziendale/personale protezionistica e difensiva, ad un approccio aperto, in cui tutti danno quello che sanno fare meglio e tutti si arricchiscono imparando dal meglio. La condivisione diviene la regola e non più un'imposizione da cui guardarsi. Un salto culturale che deve permeare tutti i ruoli e livelli. E non basta: deve estendersi all'esterno, coinvolgendo altre realtà aziendali e non.

Una cultura del dare gratuito in cui ciascuno condividere il meglio di se, e ha a disposizione la possibilità di beneficiare del meglio messo in comune da altri.

E quando questi "altri" sono veramente tanti (così come oramai accade nella comunità OpenSource), l'arricchimento culturale di tutti è garantito, senza che nessuno si impoverisca, i problemi vengono risolti in modo più rapido e brillante, e la competizione si sposta dal livello personale ed aziendale, a quello di risoluzione congiunta di problemi e necessità che diventano comuni. E anche l'atmosfera lavorativa ne beneficia. Un tipico caso in cui il totale è maggiore della somma delle parti.

La storia del calabrone che non poteva volare

⁴ KM = Knowledge Management

Sento l'obiezione: ma se rendiamo pubblico il nostro core-know-how, cosa previene la perdita del vantaggio competitivo?

Questo mi ricorda la storia (vera) del calabrone che non poteva volare. Fino a tutto l'800 sono stati pubblicati numerosi studi di fisica atti a dimostrare che, alla luce della fisica Newtoniana (l'unica nota fino ad allora), il calabrone non poteva volare: il suo rapporto peso/estensione alare rendeva questa operazione impossibile. Eppure il calabrone vola. C'è voluta la fisica di Einstein per spiegare come mai il panciuto insetto vola amenamente da secoli, in barba a Newton.

La storia dell'OpenSource è analoga: si sono versati fiumi di parole per definirla come un approccio che non può funzionare, che è destinato al fallimento, etc. etc. Eppure continua a funzionare e ad espandersi. E che funziona lo hanno ultimamente capito anche le major di produzione software. Ditte che in teoria avrebbero tutto da perdere hanno già deciso di abbracciare la filosofia OpenSource (da SUNMicrosystems a HP, e sembra che anche Microsoft inizi a fare qualche passo).

E se l'Economia Politica, basandosi sul concetto dell'*Homo Economicus*, continua a sostenere che non può funzionare, l'Economia Civile, che riconosce alla natura umana un'attitudine *reciprocante*, predice invece che il *modus operandi* che sta dietro l'OpenSource è l'unica via percorribile per l'economia del futuro. Intanto il calabrone continua a volare alto.

Massimo Cardaci - info@edc-consulting.org