

## **CAPITOLO PRIMO**

### **Open Source come modello di business per le PMI: analisi critica e casi di studio**

*Sonia Bergamaschi, Francesco Nigro, Laura Po, Maurizio Vincini*

## **INTRODUZIONE**

Il software Open Source sta attirando l'attenzione a tutti i livelli, sia all'interno del mondo economico che produttivo, perché propone un nuovo modello di sviluppo tecnologico ed economico fortemente innovativo e di rottura con il passato.

In questo elaborato verranno analizzate le ragioni che stanno determinando il successo di tale modello e verranno inoltre presentate alcune casistiche in cui l'Open Source risulta vantaggioso, evidenziando gli aspetti più interessanti sia per gli utilizzatori che per i produttori del software.

Nella prima sezione vengono presentati i vantaggi immediati che l'utente trae dall'utilizzo del software open e come l'innovazione innescata dal software possa influenzare positivamente lo sviluppo economico locale.

Nella seconda sezione vengono presentati alcuni casi in cui l'Open Source è utilizzato con successo, facendo riferimento ad uno scenario globale, sintetizzando la direttiva Stanca sull'adozione del software nella pubblica amministrazione. Infine vengono analizzate brevemente le difficoltà che si possono incontrare introducendo software libero in azienda.

Nella terza sezione viene presentato il modello economico che sta alla base dell'Open Source, evidenziandone i principali meccanismi.

Nella quarta sezione vengono analizzati i forti cambiamenti organizzativi e di gestione necessari per portare avanti lo sviluppo di un progetto open, mostrando le principali difficoltà che si possono incontrare.

Infine nella quinta sezione sono riportate le conclusioni.

## I VANTAGGI DERIVANTI DALL'UTILIZZO DI SOFTWARE OPEN SOURCE

### 1.1 I vantaggi per i clienti dei prodotti open in un mondo IT sempre più complesso e alla ricerca di integrazione e riduzione dei costi

I vantaggi che il cliente trae dall'utilizzo di software Open Source rispetto a quello proprietario (chiamato anche privativo) si possono riassumere brevemente nei seguenti punti:

1. Minore *Total Cost of Ownership* (TCO)
2. Estrema personalizzazione
3. Svincolamento dal *Single Vendor Lock-In*
4. Protezione dall'inaccessibilità del fornitore
5. Migliore compatibilità e integrazione dei sistemi

Il costo totale sostenuto per l'adozione di un software, chiamato appunto Total Cost of Ownership, è dato da diversi fattori tra cui il costo iniziale di acquisto della licenza e il costo di mantenimento; benché il concetto di Open Source sia del tutto staccato dal concetto di gratuità del software è pur vero che una larga fetta di software Open Source non richiede alcuna spesa iniziale di acquisto, eliminando o riducendo fortemente la spesa per le licenze.

C'è da aggiungere inoltre che il software Open Source gode di un'ottima reputazione per quanto riguarda sicurezza e stabilità del codice; ciò è da imputare principalmente al metodo di sviluppo del software e in particolare alla revisione paritaria del codice.

Essendo il codice sorgente accessibile e visionabile da chiunque tendenzialmente il software Open Source ha una base di persone che testano e controllano il codice molto più ampia dei software proprietari; questo fattore si è dimostrato uno strumento eccezionalmente efficace per ottenere software di elevata qualità ed affidabilità ad un costo minore.

Oltre all'aspetto sicurezza, molto importante, questo si traduce inoltre in un minor numero di blocchi, meno interruzioni del lavoro, meno richieste di assistenza e quindi in definitiva una spesa minore; ciò contribuisce ulteriormente ad un minore TCO (*Total Cost of Ownership*).

Per fare un esempio concreto è possibile pensare al sistema operativo Linux, considerato uno tra i più importanti software Open Source, se non il più importante, che può essere utilizzato gratuitamente da chiunque ed in più è immune dai virus.

Al risparmio iniziale della licenza si somma quindi una maggiore affidabilità del sistema che può essere, a lungo andare, una voce di spesa molto più corposa della licenza stessa.

La disponibilità del codice sorgente consente inoltre una personalizzazione e una

raffinazione del prodotto estrema, permettendo di modificare e adattare ad esigenze specifiche ogni aspetto del software. Questa possibilità è spesso del tutto assente nel software proprietario o comunque incomparabilmente più limitata.

Il *Single Vendor Lock-In* è un fenomeno conosciuto in tutti i settori, non solo in quello del software, che porta ad essere legati in esclusiva ad un unico fornitore per alcuni servizi o prodotti.

Questa condizione di dipendenza diretta da un unico soggetto è particolarmente odiata perché sbilancia enormemente le capacità di contrattazione da parte del fornitore che può approfittare della posizione di esclusività per imporre al cliente condizioni inique.

Questa posizione di asimmetria contrattuale è la più ricercata dai fornitori di software che cercano attraverso determinati strumenti di imporla al cliente; gli strumenti comunemente utilizzati sono principalmente rivolti ad eliminare o a limitare fortemente l'interoperabilità con altri sistemi per impedire che altri concorrenti possano subentrare.

Questo obiettivo viene portato avanti utilizzando, ad esempio, formati per i dati segreti e incompatibili e nascondendo i dettagli tecnici del software impedendo la collaborazione con sistemi di terzi parti.

Per fare un esempio concreto e di uso comune basti pensare al formato .doc utilizzato da Word; tale formato è segreto, nessuno a parte Microsoft è conoscenza di come i dati, di proprietà dell'utente, vengono conservati all'interno di un file .doc, benché non ci sia nessun motivo tecnico né tanto meno alcun vantaggio per l'utente.

Una volta che l'utente abbia utilizzato Word per un po' di tempo, e abbia prodotto un certo numero di documenti, risultata molto difficoltoso e spesso antieconomico cambiare prodotto; i documenti risultano ingabbiati all'interno del formato .doc e l'utente con essi.

L'operazione di recupero dei dati da un formato chiuso, come il .doc, ha un costo generalmente basso sul singolo file ma che può diventare sostanzioso se il numero dei file da recuperare è rilevante; tale processo è inoltre non privo di imperfezioni, facendone aumentare il costo.

Questo solleva anche altre problematiche, oltre a quelle economiche, sulla effettiva disponibilità dei dati. Col passare del tempo la dipendenza da un unico prodotto può avere risvolti inaspettati e potenzialmente disastrosi: documenti importanti che necessitano di essere conservati per lunghi periodi di tempo, pensiamo ad esempio a tutta la documentazione creata per l'anagrafe dei cittadini, può andare involontariamente perduta perché il software proprietario utilizzato all'epoca è diventato obsoleto e non più utilizzabile.

Lo scenario appena presentato non è possibile con il software Open Source: avendo a disposizione il codice sorgente i dettagli interni del software sono automaticamente esposti così come il formato utilizzato per la memorizzazione dei dati, risolvendo alla radice il

problema.

Inoltre, nel momento in cui la casa madre non dovesse essere più disponibile o in grado di far fronte agli impegni, o, per un qualunque motivo, il cliente volesse cambiare fornitore, la disponibilità del codice garantisce la continuità del progetto oltre la vita stessa del fornitore originale.

Il caso di fallimento del proprio fornitore può risultare un evento particolarmente catastrofico nel caso di software chiuso, dopo un lasso di tempo variabile, ma generalmente non più lungo di qualche anno, il cliente si trova costretto a cambiare software per uno che venga aggiornato e per cui venga offerta assistenza; nel caso di software open il cliente può sempre pagare un'altra software house perché se ne occupi e ne continui lo sviluppo.

Il software Open Source permette intrinsecamente una migliore interoperabilità tra i sistemi, proprio perché ogni aspetto del software è aperto e ispezionabile; l'interoperabilità dei prodotti proprietari è invece comunemente garantita solo per il software della stessa casa madre.

E' evidente come per il cliente l'Open Source, anche a parità di costo iniziale, sia la strada più conveniente e affidabile nel tempo.

### *1.2 Opportunità per il territorio*

Al momento la quasi totalità del software chiuso utilizzato viene sviluppato da multinazionali di origine americana; in ultima analisi qualsiasi investimento in software è quasi completamente dirottato verso un paese estero e solo una piccola parte rimane sul territorio.

Questo vale sia per quello che riguarda l'aspetto economico ma anche le competenze; di norma infatti lo sviluppo del software viene portato avanti nella sede centrale della software house dove si concentrano tutte le competenze nonché l'offerta impiegatizia.

L'Open Source invece consente lo sviluppo di competenze locali sotto svariate forme:

1. la personalizzazione del software è un servizio che va effettuato a stretto contatto con il cliente ed è pertanto di fondamentale importanza la località del fornitore.
2. Come la personalizzazione anche l'assistenza post vendita richiede la presenza in azienda del consulente o del tecnico.
3. Come già detto la disponibilità del codice sorgente consente di studiare il software, favorendo lo sviluppo di professionalità e competenze di alto profilo e non solo limitate alla configurazione e installazione di un prodotto.
4. Il modello open consente alle piccole aziende di collaborare e fare sistema permettendogli di competere con i grandi colossi; questo modello si adatta particolarmente bene alla realtà italiana composta da una miriade di piccole e piccolissime aziende che difficilmente possono sostenere gli elevati costi di sviluppo

di un software complesso.

5. Il software Open Source sta diventando sempre più anche uno strumento didattico; in questa maniera gli studenti possono confrontarsi con software realmente utilizzato e diffuso. In questa maniera non solo gli studenti hanno un'idea concreta del mondo dello sviluppo del software ma acquisiscono anche competenze concrete che possono essere spese immediatamente una volta terminati gli studi. Questo innesca un circolo virtuoso tra il mondo accademico, che utilizza il software come strumento didattico, e il mondo economico che può trarre un forte impulso all'innovazione dall'attività di ricerca del mondo accademico.
6. La riduzione dei costi iniziali consente una riduzione delle barriere all'entrata favorendo lo sviluppo di nuove aziende.
7. La riduzione del costo iniziale di acquisto del software consente di dirottare i soldi che prima venivano spesi semplicemente per comprare un prodotto, verso altri settori come la formazione; anche a parità di costo è possibile trasformare una spesa in un investimento in nuove competenze per il personale.
8. Questo circolo virtuoso che si viene a innescare consente nel lungo periodo di diminuire la dipendenza dall'estero per tecnologie e sistemi che si rivelano ogni giorno di più strategici.

## CHI UTILIZZA IL SOFTWARE OPEN SOURCE

### 2.1 Utilizzatori istituzionali; La direttiva italiana sull'Open Source

Esistono alcuni ambiti in cui il software open possiede il predominio assoluto del mercato; non a caso uno degli ambiti più floridi è quello dei servizi per Internet, dove l'interoperabilità tra diversi sistemi è un requisito di partenza imprescindibile.

Apache è il server web che gestisce circa il 70% di tutti i siti web del mondo, il server BIND per il servizio di traduzione DNS degli indirizzi Internet detiene un altro primato con una quota di quasi l'80%.

Un altro ambito di grande successo e predominio assoluto è quello dei centri di calcolo ad alte prestazioni per i calcoli scientifici o simulativi, in cui non esistono praticamente concorrenti.

Tolti questi ambiti la vera sfida oggi per il software libero è sui servizi di uso comuni in ufficio, come l'office automation, oggi dominata apertamente da Microsoft.

Visti i grandi vantaggi che il software Open Source apporta si stanno moltiplicando di anno in anno i progetti, di sperimentazione prima e di migrazione poi, per la sua introduzione.

La mancanza di competenze informatiche, che spesso innesca anche diffidenza, è il maggiore ostacolo all'introduzione del software Open Source e questo porta ad una diffusione frammentata a seconda degli ambiti e disomogenea sul territorio.

Non è un caso che nel mondo i maggiori utilizzatori di software open siano tendenzialmente quei paesi in cui storicamente tale software è nato come gli Stati Uniti e, in Europa, Francia e Germania, dove si sono sviluppate due aziende tra le più importanti a livello mondiale per il mondo Linux: SUSE, recentemente acquisita da Novell, e Mandriva.

In ambito istituzionale gli utilizzatori preferenziali di prodotti Open Source sono in particolare tutti quegli enti dove la sicurezza ha un'importanza chiave come:

- La National Security Agency (NSA), che ha anche contribuito direttamente allo sviluppo del codice sorgente di Linux sviluppandone un'estensione, chiamata Security Enhanced Linux (SELinux).
- Le forze armate svedesi dal 2007 stanno portando avanti un piano di migrazione a Linux dell'infrastruttura IT.

- L'FBI utilizza diversi software Open Source, tra cui i più famosi sono Apache e MySQL, per il sistema di risposta alla emergenza per lo stato del Texas.
- La marina americana utilizza Linux per i sonar impiegati nei sottomarini nucleari.
- A Fairfax, Virginia, il dipartimento di giustizia ha migrato l'intero sistema di giudiziario che gestisce tutti il sistema dei processi, dalle udienze alla catalogazione delle prove e alla gestione dei bilanci, a software open.

É da notare come la migrazione a software open sia condizionata da molti fattori e non necessariamente solo da quello economico; in particolare la sicurezza, la flessibilità e le performance sono tra i motivi più importanti.

Tra gli altri grandi casi di successo di adozione di software Open Source in ambienti istituzionali ricordiamo:

- La NASA è uno degli utilizzatori storici di Linux e del software Open Source in generale; già dal 1997 il controllo degli esperimenti sullo space Shuttle è affidato a software open.
- Lo United States Departemend of Energy, che già nel 2002 possedeva il centro di calcolo Linux più potente del mondo.
- Il servizio postale americano utilizza software open dal 1997 per il riconoscimento automatico degli indirizzi sulle buste.
- La città di Monaco è impegnata dal 2002 nella migrazione dell'intera infrastruttura informatica comunale a software open.
- La città di Berlino è impegnato dal 2005 nella migrazione dell'intera infrastruttura del comune a software Open Source, operazione che coinvolge più di 58000 postazioni di lavoro e diverse migliaia di server.
- L'agenzia governativa per il lavoro tedesca ha già completato la migrazione di 13000 postazioni di lavoro a Linux.
- Il ministero dell'educazione francese ha già completato la migrazione dell'intera infrastruttura dei server utilizzati, svariate migliaia, a Linux.
- Il governo svizzero è impegnato dal 2005 nella migrazione dell'infrastruttura dei server utilizzati dagli enti pubblici a Linux.
- Il ministero degli affari interni giapponese è impegnato dal 2005 nella migrazione a software open delle infrastrutture chiave del paese con l'intento di diminuire la dipendenza del paese dall'estero e in particolare da Microsoft.



- L'agenzia governativa australiana che gestisce i sussidi utilizza software open per il pagamento di sussidi per 55 miliardi di dollari australiani.
- Il governo venezuelano è impegnato dal 2004 nella migrazione dell'intera struttura informatica governativa a software open.

A livello italiano il software open risulta distribuito, in ambito istituzionale, a macchia di leopardo, con alcune punte di eccellenza come la provincia autonoma di Trento che viene spesso citata come esempio di realtà virtuosa in questo ambito.

In ambito locale nel comune di Modena esistono diversi enti che già utilizzano software Open Source, come ad esempio il servizio ospedaliero locale, e in Comune si è avviato da qualche tempo un progetto per la migrazione graduale da Microsoft Office a OpenOffice.org.

## *2.2 La direttiva italiana sull'Open Source*

La direttiva Stanca “Sviluppo ed utilizzazione dei programmi informatici da parte delle pubbliche amministrazioni” del 2003, informalmente conosciuta anche come direttiva Open Source, pone le basi per le metodologie che la pubblica amministrazione dovrebbe utilizzare per la scelta del software da acquisire.

Oltre all'aspetto economico, come i costi di possesso e di uscita, la direttiva impone di valutare anche diversi fattori tecnici come:

1. assicurarsi che qualora il fornitore non sia più in grado di fornire supporto il codice sorgente e la documentazione relativa siano resi disponibili o ceduti.
2. Le soluzioni adottate devono garantire l'interoperabilità tra i diversi sistemi informativi.
3. I sistemi informativi non devono essere dipendenti da un unico fornitore.
4. I fornitori garantiscano la disponibilità del codice sorgente per l'ispezione e la tracciabilità del codice.
5. I dati siano esportati almeno in un formato aperto.
6. Il fornitore che sviluppi software personalizzato per la pubblica amministrazione ha l'obbligo di fornire servizi che consentano di trasportare il software su altre piattaforme e permettano il riuso del software.

Questa direttiva promuove l'utilizzo dell'Open Source nelle pubbliche amministrazioni ed è pertanto lecito attendersi una crescita costante e duratura della richiesta.

Gli ambiti in cui il software e i formati liberi sono destinati ad avere maggiore successo nella pubblica amministrazione risultano essere:

1. i servizi lato server, in cui storicamente il software libero eccelle, quindi tutti i servizi online; questo significa server web, posta elettronica, servizi di autenticazione degli utenti, sistemi di sicurezza e monitoraggio, servizi di backup, archiviazione e altro ancora.
2. La gestione documentale e l'office automation, di importanza strategica per la pubblica amministrazione così come i formati liberi.

### *2.3 L'Open Source in azienda.*

In ambito aziendale, dove non c'è la necessità di confrontarsi con i cittadini o di seguire normative sulla disponibilità e tracciabilità dei dati, i motivi che spingono un'azienda ad adottare una soluzione piuttosto che un'altra sono principalmente economici.

Tolti alcuni ambiti specifici, come quello dei centri di calcolo ad alte prestazioni dove praticamente non esistono alternative ai sistemi Linux / Unix, il minor costo delle licenze è senz'altro uno dei fattori di maggiore interesse per le aziende.

Le piccole o piccolissime aziende, che sono la stragrande maggioranza in Italia, hanno in genere necessità abbastanza comuni e standardizzate e le attività svolte si esauriscono in:

1. navigazione su Internet.
2. Posta elettronica.
3. Strumenti di base per l'office automation come scrittura di documenti, tabelle di calcolo e presentazioni.
4. Piccoli software gestionali per la fatturazione, gestione dei contatti con i clienti ed eventuale gestione del magazzino.
5. Sviluppo di semplici materiali grafici di promozione.
6. Semplici sistemi di reporting e statistiche.

Tutte queste attività possono essere compiute con l'utilizzo di comuni software open e spesso la migrazione di una piccola azienda si traduce nell'individuazione di software libero analogo nelle funzioni a quello proprietario utilizzato e nell'installazione sui computer.

La diffusione di una rete capillare sul territorio di competenze sul software open potrebbe incentivarne fortemente la diffusione tuttavia esiste anche un fattore umano di cui tenere conto.

Il maggior scoglio alla migrazione infatti può non derivare da problemi di natura tecnica bensì da fattori umani, il cambio d'interfaccia di utilizzo dei programmi può generare diffidenza e, all'inizio, frustrazione negli utenti e questo, unito alla inerziale resistenza al cambiamento, può effettivamente ostacolare l'adozione.

## IL MODELLO ECONOMICO DELL'OPEN SOURCE

### 3.1 Perché il modello economico proposto dall'Open Source funziona e come fanno le software house a sopravvivere se regalano i propri prodotti

Il termine inglese *Free Rinding* si potrebbe tradurre come utilizzo iniquo del lavoro o delle risorse altrui; ogni volta che qualcuno approfitta appunto di risorse che non gli appartengono senza dare nulla in cambio si parla di *Free Riding*.

Quando ci si avvicina per la prima volta al mondo Open Source si può incappare nell'errore di credere che l'Open Source favorisca se non addirittura incoraggi il *Free Riding*; questa è ovviamente una visione falsata. Così come per il software proprietario anche per quello open gli strumenti utilizzati per difendersi dal *Free Riding* sono le licenze d'uso con cui viene distribuito il software.

Nella maggior parte dei casi il software Open Source è coperto da licenze che garantiscono che il codice sorgente non possa essere reso proprietario; inoltre chiunque apporti delle modifiche al software e le ridistribuisca è obbligato a rendere pubbliche le modifiche sotto la stessa licenza e con gli stessi termini del programma originale; tali licenze mettono completamente al riparo dal *Free Riding*.

Il modello utilizzato dal software open ovviamente non è adatto a qualsiasi ambito, ma è soggetto a certe regole: innanzitutto è necessario dividere le aziende che producono software e che quindi per esse è il loro core business, dalle altre.

Le aziende infatti che non si occupano di software, ma che tuttavia si trovano a sviluppare codice per le proprie esigenze interne, hanno l'interesse primario di abbassare i costi di sviluppo ottenendo al contempo un software solido, testato e molto diffuso; tale situazione, abbastanza comune, può essere positivamente influenzata dall'utilizzo di una strategia open.

Per le di aziende il cui core business sia lo sviluppo di software è fondamentale analizzare il mercato nel suo complesso.

I mercati poco concorrenziali sono un ambiente in cui l'Open Source può risultare perdente; nel caso di predominio tecnologico in un settore con pochi concorrenti, o con concorrenti di livello nettamente inferiore, adottare una strategia Open Source risulta poco proficuo, il rischio è quello di bruciare un intero segmento senza ottenere particolari vantaggi.

Un'azienda che si trovi già in una situazione di predominio non potrebbe trarre nessun vantaggio dal condividere con la concorrenza il proprio *know how* tecnologico sviluppato internamente.

Altro punto da valutare sono le cosiddette caratteristiche differenzianti del prodotto: tutto quanto rende il proprio prodotto diverso dagli altri, quello che riesce ad offrire in più, è una caratteristica differenziante.

Nel caso in cui il software sia poco complesso e quindi le entrate per l'azienda derivino esclusivamente dalla vendita del software, condividere con i concorrenti una caratteristica differenziante, sviluppata internamente, sostenendo completamente i costi di ricerca e sviluppo, avvantaggerebbe i concorrenti senza portare nessun nuovo beneficio e vanificando l'investimento necessario allo sviluppo.

In linea di principio nel caso non sia possibile un modello basato sui servizi risulta pertanto vantaggioso cercare di condividere i costi e i rischi per lo sviluppo di tutte le caratteristiche non differenzianti, seguendo un modello aperto, e tenersi gelosamente strette quelle che invece lo sono.

Nel caso di software complessi o di ambienti di utilizzo particolarmente evoluti l'obiettivo invece è di tenere basso il costo di sviluppo e far evolvere il software il più possibile per poter offrire poi servizi e assistenza.

L'Open Source consente quindi di diminuire i costi iniziali di sviluppo, dividendoli tra più soggetti, e inoltre di ridurre i tempi, consentendo un *time to market* minore.

### 3.2 I nuovi meccanismi che stanno dietro a un nuovo mercato

Il software è un bene immateriale, una volta creato “l'originale” il costo di replica è praticamente nullo; questo ha un grandissimo impatto sulle strategie economiche che si possono adottare.

In altre industrie come quella dell'automobile risulta impraticabile un modello in cui il bene, in questo caso la macchina, venga ceduta a costo zero al cliente per concentrare tutti i guadagni sull'attività di assistenza post vendita; questo non è altrettanto vero per il software.

Nel modello utilizzato dall'Open Source il software viene ceduto gratuitamente all'utente e si spostano i guadagni verso altre fonti come:

1. Vendita di software accessorio
2. Consulenza
3. Assistenza alla prima installazione
4. Assistenza di mantenimento
5. Sviluppo di nuovo codice
6. Integrazione
7. Aggiornamenti
8. Certificazioni di sistema

9. Formazione
10. Documentazione
11. Personalizzazione
12. Hardware
13. Comodità
14. Doppia licenza
15. Gestione del marchio

Una prima possibilità è quella di offrire una base completamente aperta e gratuita per poter poi offrire software proprietario accessorio a pagamento; in questo caso ricadono ad esempio Novell e Red Hat che offrono a pagamento alcuni software molto specifici indirizzati al mercato *enterprise*.

La consulenza rimane una voce invariata dal passaggio a un modello open, quello che però permette di offrire in più al cliente rispetto ad un modello proprietario è la varietà e la possibilità di testare per intero il software prima di adottarlo, permettendo di fornire consulenza di maggior efficacia e a più ampio spettro.

Nel caso di software complesso o di installazioni massive una grossa fetta di guadagni, sia per il software open che close, viene dalla cosiddetta assistenza alla prima installazione in cui il cliente paga per avere il supporto e le competenze tecniche per installare e configurare il sistema.

Un ambito in cui questa voce ha un notevole peso è ad esempio quello dei sistemi ERP. Il mercato degli ERP è al momento dominato dall'azienda SAP che detiene una posizione dominante rispetto ai concorrenti; non risulta pertanto molto conveniente per SAP l'adozione di una strategia open che invece potrebbe risultare molto interessante per i suoi concorrenti.

L'assistenza alla prima installazione ovviamente non è, ovviamente, una voce di rilievo per software molto semplice che non richieda particolari conoscenze per essere installato.

L'assistenza di mantenimento è un aspetto molto importante per le aziende e che tende ad acquisire maggior peso con l'aumentare delle dimensioni dell'azienda in cui è richiesta una sempre maggiore garanzia di affidabilità.

Lo sviluppo di soluzioni ad hoc o la personalizzazione di software già esistente è una realtà abbastanza comune; il vantaggio competitivo che offre l'Open Source è quello di poter sviluppare a costi minori riutilizzando codice già disponibile e di diminuire inoltre i tempi di consegna potendo offrire al cliente una applicazione a valore aggiunto.

L'integrazione di sistemi è un'altra fonte di guadagno da non sottovalutare: l'integrazione è il valore aggiunto di per se, il fatto che il software sia open o meno non influenza il risultato. In particolare meno costa il software più risulta ampia l'offerta che si può proporre ai clienti.

Una maggior disponibilità porta tendenzialmente ad un maggiore utilizzo di software che a sua volta fa aumentare la richiesta di integrazione.

La scelta di far pagare gli aggiornamenti va valutata in base al tipo di clientela e di segmento che si considera ma può risultare una fonte di guadagno considerevole.

In alcuni ambiti, come quello della sicurezza, un software non aggiornato risulta sempre meno efficace mano a mano che il tempo passa, quindi il valore dell'applicativo è fortemente determinato dagli aggiornamenti.

Un'altra strada potrebbe essere di offrire a pagamento le patch o le nuove versioni in anteprima o con tempi differenti ai clienti che sottoscrivano un abbonamento; questa è la strada intrapresa qualche tempo fa dalla distribuzione Linux Mandriva.

La certificazione di sistema è un altro tema molto importante in ambiente *enterprise* in cui per l'adozione di un software è richiesta una certificazione che un determinato software funzioni su un determinato hardware o in collaborazione con un altro software.

Questo aspetto è molto importante, in particolare, nell'ambito dei centri di calcolo e di grosse installazioni, in cui la mole di hardware e software installato è considerevole, e sono richiesti importanti investimenti economici.

La formazione è un'esigenza generata da tutto il software; il vantaggio competitivo di chi offre software Open Source e di poter spostare l'attenzione dalla licenza alla formazione e di poter offrire ai propri clienti più di un semplice prodotto o della licenza ma nuovo *know how* da far entrare in azienda.

La documentazione, in particolare, se si considera insieme alla formazione, può essere un'altra voce di guadagno. Grandi società come Novell offrono a pagamento non solo il classico manuale d'uso per il sistema operativo e simili ma anche corposi, nonché costosi, kit di autoapprendimento per le certificazioni professionali che offre.

Quello in cui ovviamente il software Open Source eccelle è la flessibilità ed è relativamente facile e immediato offrire soluzioni personalizzate per il cliente come servizio a pagamento.

In particolare l'Open Source ha, in generale, l'interesse a far sviluppare e progredire il software per offrire soluzioni più avanzate; infatti più il software è complesso e articolato maggiore sarà la richiesta di servizi.

I produttori di hardware hanno tutto l'interesse ad offrire il software a bassissimo costo o a costo zero per poter abbassare i costi di utilizzo e vendere così più hardware.

Questo vale sia per i grandi sistemi per i centri di calcolo che per il mondo *embedded* che sta vedendo Linux in forte crescita. La chiave per il successo di una piattaforma *embedded* dipende fortemente anche dalla sua diffusione e adottare Linux può essere la chiave per

catalizzare l'attenzione verso una piattaforma che ha potenzialmente a disposizione migliaia di sviluppatori già pronti.

La comodità è una motivazione da non sottovalutare affatto; molte aziende non si interessano di informatica e pertanto non sono motivate nell'approfondire le problematiche sollevate dai sistemi informativi. In questo caso non fa alcuna differenza che il software sia liberamente scaricabile e installabile, la soluzione “chiavi in mano” è l'unica accettata. Il cliente è disposto a pagare per non doversi preoccupare della gestione dei sistemi informativi e non ha alcun orientamento verso un modello aperto o chiuso.

Per il fornitore offrire software aperto, e che quindi ha un costo minore, permette di avere margini di profitto più elevati e di offrire soluzioni a costi minori essendo al contempo più concorrenziale.

Chi detiene il *copyright* sul codice può decidere di rilasciare il software sotto due diverse licenze, una open e l'altra proprietaria, e decidere di differenziare le due versioni; il motivo per tenere questi due rami separati potrebbe essere la decisione di vendere a terzi il codice, per farlo integrare in altri programmi commerciali più ampi, o semplicemente per offrire due versioni con caratteristiche diverse.

Tipicamente la versione proprietaria offre caratteristiche indirizzate all'ambiente *enterprise* mentre quella open è più adatta per un uso in ambienti meno complessi.

Per potersi comunque avvantaggiare del modello open è necessario rilasciare la maggior parte del codice e tenere solo alcuni moduli proprietari da sviluppare completamente internamente.

Anche la gestione del marchio è una possibilità che il software open ha in comune con quello proprietario e nasce dall'interesse di aziende minori a farsi associare al marchio di una azienda prestigiosa, cosa che accade indipendentemente dal fatto che offra software open o meno. I programmi di partnership sono una delle possibilità che possono offrire nuovi guadagni all'azienda.

Ovviamente è possibile mescolare tutte queste possibilità per trovare la formula più adatta al prodotto.

Tipicamente vengono presentati quattro modelli, così come formalizzati dall'*Open Source Initiative*:

*Support Sellers*: chi appartiene a questa categoria offre gratuitamente il software per poter poi offrire tutti i servizi post vendita a pagamento, come assistenza, integrazione ecc.

*Loss Leader*: il software open serve per preparare la strada ad altri prodotti proprietari



dell'azienda

*Widget Frosting*: tipico esempio è la vendita di hardware, il software è open per consentirne la più ampia diffusione possibile e abbassare i costi

*Accessorizing*: l'offerta non è incentrata sul software ma su tutti i servizi accessori, dai manuali al *merchandise*.

Un altro aspetto che offre vantaggi sia pratici che “politici” e che sfugge un po' alle logiche viste fino ad ora è la diffusione di un prodotto come valore in se:

Poter vantare un certa base di installazioni garantisce una maggior credibilità davanti a potenziali investitori, il che può decidere il successo o meno di un progetto

Più la cerchia di utenti si allarga maggiore è il numero di ambienti e gli scenari in cui il software viene testato rendendolo di conseguenza più solido; testare il software in ambienti simulati è un procedimento lungo e oneroso e a volte non particolarmente conclusivo, avere dei test reali è una grossa fonte di informazioni, oltre che un grosso risparmio di tempo e denaro

Come spesso è accaduto in altri progetti open, avere una larga base di utilizzo stimola l'adattamento e a volte addirittura il *porting* del software verso nuovi scenari di utilizzo inizialmente non previsti, allargando a sua volta il numero potenziali utenti

Potenzialmente significa anche avere più sviluppatori: oltre che per un semplice calcolo statistico, questo è dovuto all'aumento di interesse, anche da parte di terzi, verso il prodotto; più è alta la domanda maggiore sarà la probabilità che altri si interessino al progetto aprendo la strada a nuove collaborazioni e mercati

Più clienti utilizzano una soluzione più saranno i nuovi clienti che prenderanno in considerazione quella soluzione per i propri problemi

Avere molti clienti vuol dire anche essere investiti di un'autorevolezza che può essere effettivamente spesa con i clienti e i partner e che può essere una fonte di guadagno. Questo aspetto in particolare ricade sotto la gestione del marchio: se si ha un marchio forte e riconosciuto si possono, ad esempio, attuare delle politiche di partnership a pagamento.

## **IL RADICALE CAMBIAMENTO NEI PROCESSI AZIENDALI APPORTATI DAL SOFTWARE OPEN**

### *4.1 Come viene modificato il normale processo di sviluppo, la gestione delle risorse e la comunicazione nel momento in cui si affronta un progetto open*

Normalmente in un progetto proprietario la struttura che sta dietro lo sviluppo di un software è rigida e ben definita; un piccolo e chiaramente circoscritto gruppo di persone pagate dall'azienda si concentrano nella scrittura del codice sorgente, il gruppo di management decide la *roadmap* del progetto, un altro gruppo si occuperà del marketing e della comunicazione, un altro del sito internet, un altro del supporto e così via; più l'azienda è grossa più queste figure saranno distinte e slegate tra loro mentre nelle aziende di ridotte dimensioni solitamente queste figure tendono a sovrapporsi ampiamente.

I processi aziendali a loro volta sono strutturati, esiste una gerarchia, un superiore di riferimento e i mezzi di comunicazione sono formali e definiti.

Al contrario in un progetto open di solito vi è un ristretto gruppo di persone che rappresentano "l'autorità" nel progetto, e non necessariamente sono legate all'azienda che lo ha iniziato, e un grosso volume di persone che entrano ed escono dal progetto dando piccoli contributi.

Il compito di chi gestisce un progetto open è proprio quello di tenere basse le barriere all'ingresso per favorire l'accesso di nuove persone.

Le relazioni sono paritarie e avvengono attraverso mezzi di comunicazione indiretti come l'email o un forum e si deve far fronte a differenze culturali tra persone geograficamente distanti.

Lo sforzo maggiore richiesto a chi si avvicina per la prima volta ad un progetto Open Source è proprio di cambiare radicalmente ottica, passando da un modello di sviluppo che potremmo definire privato, ad uno pubblico, davanti agli occhi di tutti; quello che a prima vista può sembrare caotico in realtà segue semplicemente leggi diverse, dove l'autorevolezza che una persona riesce ad esprimere è determinata dalle capacità e dal grado di coinvolgimento più che dalla posizione che si detiene in azienda.

L'obiettivo di chi gestisce un progetto open in ultima analisi è quello di riuscire a creare interesse e attirare il maggior numero di utenti e sviluppatori al progetto; questo ecosistema di persone che gravitano intorno al progetto partecipandovi più o meno attivamente è comunemente identificato con il termine *community* o, in italiano, comunità.

Come spiegato nel paragrafo sul *Free Riding* l'Open Source non è un modo per approfittare del lavoro altrui ed è pertanto di vitale importanza essere completamente onesti con la comunità; scelte ambigue o poco chiare genereranno inevitabilmente malcontento e daranno

cattiva fama al progetto. Questo implica inevitabilmente che la quasi totalità dei processi decisionali vengano resi pubblici e affrontati insieme alla community, evitando discussioni private e decisioni unilaterali, facendo ovviamente sempre le dovute distinzioni.

Questo può spaventare perché in parte implica perdere il controllo su alcuni aspetti che prima erano di esclusiva pertinenza dei creatori del progetto; i partecipanti della community diventano un soggetto con cui dialogare e di cui tenere conto, ma questo è proprio ciò che ci si aspetta dalla community!

Il risvolto positivo di tutto questo è che il prodotto sarà sempre vicino alle aspettative dei clienti. Lo sviluppo del codice diventa poi inaspettatamente un nuovo punto di contatto con il pubblico; se prima infatti il codice rimaneva segreto, ed eventualmente l'utente poteva solo constatare un malfunzionamento del software, adesso la "qualità" del codice è sotto gli occhi di tutti e può contribuire a creare una buona o cattiva reputazione del progetto.

#### *4.2 Molte fasi dello sviluppo e dei processi aziendali, prima interni all'azienda stessa e riservati, diventano di dominio pubblico, ponendo nuove opportunità ed esigenze*

Il processo di sviluppo e i programmatori stessi, prima totalmente sconosciuti all'esterno dell'azienda, entrano in contatto direttamente con la comunità; se si rende open un progetto precedentemente proprietario i programmatori in particolare potrebbero trovarsi a disagio a doversi confrontare con soggetti esterni che analizzano il proprio lavoro.

Anche il modello dei rilasci del software viene cambiato: il modello classico è organizzato come una sorta di scala con bruschi cambiamenti in concomitanza dei rilasci della *major release*.

Una release maggiore è seguita da un ciclo ben scadenzato nel tempo di *patch* per arrivare quindi a una nuova release minore di mantenimento che sintetizza grosso modo tutte le *patch* rilasciate fino a quel momento.

Il rilascio della *major release* è solitamente dettato da esigenze commerciali e quando si sfiorano i tempi si ha la tendenza a rilasciare il software anche se effettivamente non completo, scaricando lo sviluppo non ancora completato nelle release minori.

Al contrario nel mondo open c'è tendenzialmente un susseguirsi frequente di rilasci di minore entità, molto fitti nel tempo se paragonati al caso precedente, con una crescita del progetto costante e graduale.

Il problema è di origine principalmente organizzativa: se infatti nello sviluppo sono impiegate poche persone il contatto e lo scambio di idee tra esse può essere informale e avvenire ad esempio direttamente a voce; questo ovviamente non è più possibile se le persone coinvolte sono anche solo qualche decina.

I problemi organizzativi che la gestione di grandi gruppi solleva è ovviamente valida sia per

il software proprietario quanto per quello Open Source; il punto di svolta è la tecnologia oggi a disposizione che consente di portare l'organizzazione ad un nuovo livello.

Grazie agli strumenti disponibili oggi e a Internet è possibile organizzare e gestire una infrastruttura composta siti internet, *blog*, *wiki*, *chat* e altro che abilitano la comunicazione e lo sviluppo condiviso a distanza e servono appunto per aggirare, o quanto meno mitigare, i problemi gestionali, permettendo di automatizzare processi e regolamentazioni necessarie all'organizzazione.

## CONCLUSIONI

Come si è visto il software Open Source è già utilizzato comunemente da parecchio tempo ed anzi talvolta è in condizione di predominio assoluto sui concorrenti.

Chiunque utilizzi oggi giorno un computer utilizza software Open Source ma nella maggior parte dei casi l'utente comune non ne è consapevole; i servizi che consentono oggi a Internet di funzionare utilizzano, per la stragrande maggioranza, software open, ma questi servizi sono impliciti, invisibili all'utente.

Tutti questi servizi dove il software open eccelle sono lato server, quindi ben conosciuti agli amministratori di sistema ma quasi del tutto ignoti agli utenti; chi utilizza Internet difficilmente sarà consapevole della complessa struttura che la fa funzionare, così come chi utilizza la posta in ufficio difficilmente si sarà chiesto qual'è il programma che la gestisce o se il database aziendale è open.

La grande scommessa per il software open è ora il segmento Desktop, quello dei computer degli utenti, attualmente dominato dal monopolio di Microsoft.

In questo segmento i software di maggior successo risultano: Firefox, concorrente di Internet Explorer, OpenOffice.org, suite per l'ufficio paragonabile a Microsoft Office, e ovviamente Linux, concorrente di Windows.

La diffusione di questi tre programmi è in forte crescita e fino ad oggi sono stati gli unici che siano riusciti a scalfire il monopolio Microsoft acquisendo quote di mercato; tutti i software proprietari che hanno provato a concorrere sono spariti, pensiamo al caso Netscape - Internet Explorer o Microsoft Office StarOffice o ancora Windows - OS/2, il che è già un gran successo dell'Open Source.

In passato l'accusa che veniva mossa all'Open Source era di essere un fenomeno per appassionati, amatoriale, non adatto alle esigenze del mercato aziendale; oggi la situazione è radicalmente diversa, numerose sono le aziende che offrono soluzioni completamente open in grado di far fronte anche alle esigenze delle aziende più strutturate e complesse: Novell, RedHat, Oracle, IBM, HP e SUN sono solo alcuni dei nomi più rinomati del campo dell'IT che forniscono soluzioni open.

In ambiente Desktop ci sono ancora ampissimi margini di miglioramento ed è da attendersi che la diffusione di strumenti come OpenOffice.org aumenti soprattutto in quei settori, come la pubblica amministrazione, in cerca di nuove soluzioni più vantaggiose e che garantiscano l'integrità delle informazioni nel lungo periodo.

Come visto esiste già un modello economico consolidato e sostenibile a sostegno dell'Open Source, brevemente catalogato nelle quattro tipologie: Support Sellers, Loss Leader, Widget Frosting e Accessorizing.

Tutti questi modelli considerano il software più un servizio che un prodotto in senso stretto,

non essendo il software un oggetto materiale, e risultano particolarmente efficaci in ambienti estremamente competitivi.

Una software house che decide di intraprendere la strada dell'Open Source ha già a disposizione numerosi esempi e casi pratici da seguire; è fondamentale che però capisca a fondo cosa comporta tale scelta e sia pronta ad affrontare i cambiamenti imposti nelle attività aziendali che porta lo sviluppo open.

Una maggiore diffusione della cultura informatica e una maggiore consapevolezza degli utenti potrebbero essere facilitatori alla diffusione del software open; il proliferare di soluzioni integrate, come i subnotebook, come l'EEE pc di ASUS o l'OLPC di Negroponte, e gli smartphone saranno molto probabilmente il nuovo fronte su cui il software open andrà a concorrere.

In Italia, a differenza di altri paesi europei, non c'è ancora per gli utenti la libertà di scelta in ambito informatico; sull'acquisto di un portatile ad esempio, a parte alcune rare eccezioni, è inclusa una licenza per Windows che l'utente è obbligato a pagare e che non può scorporata, rimanendo nascosta all'interno del prezzo di acquisto.

Una più attenta difesa del consumatore da parte del legislatore e attività di contrasto della pirateria informatica più incisive potrebbero essere altri fattori rilevanti per la diffusione del software open.