

Il FLOSS e i suoi modelli di business per le PMI

Flavia Marzano	Valentina Donno
Associazione UnaRete	Researcher / Consulente innovazione e new media
http://www.unarete.org	
Piazza Chiara Gambacorti 19	Via Vallarsa 16
56125 Pisa	00141 Roma
Cell. +39-348-5104754	Cell. +39-329.9776023
Email: flavia.marzano@gmail.com	Email: donno.valentina@gmail.com

Nota metodologica e Sommario

Questo documento si propone di condurre una disamina della dimensione strategica ed economica dell'open source che ne riconosca ed evidenzi, non tanto la valenza socio-politica, quanto le qualità e potenzialità di business.

Non s'intende tuttavia fornire una tassonomia dei "modelli di business" dell'open source, «per definire i quali occorrerebbe analizzare in dettaglio la strategia aziendale - di una software house o una ICT company - e come questa si differenzia dalla concorrenza, sia essa open o proprietaria (...)».¹

Sospendendo le motivazioni ideologiche, etiche e sociali alla base della genesi e dello sviluppo del software libero, la domanda a cui s'intende qui dare risposta è: a fronte della "liberazione" del codice, c'è un ritorno economico diretto o indiretto?

In altre parole: perché un'azienda, una piccola/media software house, o un singolo sviluppatore, dovrebbero adottare e/o sviluppare l'open source?

Se considerata in relazione al software proprietario, qual è la soluzione più conveniente? E soprattutto, su quali dimensioni va valutata la convenienza?

In sintesi: «Quali costi comporta adottare (e adattare) soluzioni libere? Quali costi invece permette di risparmiare? Quali sono (se ci sono) i limiti d'uso di questi programmi?»².

Analizzare le dimensioni critiche del processo di scelta di una soluzione open source è l'obiettivo ultimo di questo lavoro.

Si premette, con riferimento alla terminologia usata, che l'accezione scelta per definire l'open source è quella di "software a codice sorgente aperto"³, di seguito FLOSS: Free, Libre, Open source Software, laddove "free" va letto nel senso di "libero" e non di "gratuito".

Soggetto/oggetto d'attenzione, infine, sono le Piccole Medie Imprese (d'ora in poi PMI) attive nel settore del software e, fra queste, quelle che stanno

¹ R. Galoppini, "Modelli di business e Open source", in "Finalmente libero!" AA.VV., McGraw Hill, Milano 2008

² F. Pozzato, in "PC Open", "Corso Open source" - Lezione n°1, Marzo 2006

³ Come definito dall'Art. 2 lettera J del Codice dell'Amministrazione Digitale D. Lgs 82/2005

valutando di utilizzare soluzioni FLOSS e che necessitano di informazioni pratiche su opportunità e rischi.

Contro il permanere di luoghi comuni che ancora circondano il software libero, percepito più come una scelta ideologica, una “bella utopia”, che non come una soluzione economica concretamente praticabile, il *focus* d’analisi qui si sposta; si ri-orienta nel considerare il FLOSS come «il più importante fenomeno dell’industria del software dagli anni ‘80 ad oggi»⁴.

Come sottolinea G. Ruffatti «il software libero è ormai da tempo una realtà consolidata, visti i numeri del suo successo e le affermazioni e previsioni di diversi operatori del settore e analisti di mercato»⁵.

«Gartner, ad esempio, sostiene che nel 2006 il 13% della spesa IT è andata verso prodotti open source e prevede che nel 2011 questa quota salirà al 27%, mentre addirittura l’80% del software proprietario conterrà software open source entro il 2010».

Non a caso, le esperienze di successo nella produzione, uso e migrazione, di software a sorgente aperto si contano numerose.

Per citarne alcune, il FLOSS è stato scelto dall’*Academy of Defence* e da altri centri militari inglesi per le sue qualità di affidabilità e sicurezza. Sempre per ragioni di sicurezza, negli Stati Uniti la NASA è passata rapidamente all’open source dopo l’11 settembre. La medesima scelta, qui ispirata a ragioni di concorrenzialità e risparmio, è stata presa da paesi emergenti come il Brasile, la Malesia, Israele, le Filippine, l’India e da economie in crisi come l’Argentina, il Venezuela, il Sudafrica, la Bolivia.⁶

A livello europeo, il nuovo paradigma del FLOSS è stato portato avanti dall’*Open source Observatory - IDABC (Interoperable Delivery of European eGovernment services to Public Administration, Businesses and Citizens)*⁷; sul piano nazionale dalla “Direttiva Stanca”⁸ (a seguito dei lavori della cosiddetta “Commissione Meo”); a livello locale da leggi regionali e da atti di indirizzo adottati da diverse Amministrazioni⁹.

Spostando l’attenzione dalle politiche di governo alle quote di diffusione di molti progetti FLOSS si osserva che la presenza di prodotti liberi interessa tutte le tipologie fondamentali di software: da quello di sistema, con GNU/Linux che - stando ad un’indagine commissionata da IBM nel 2006¹⁰ - è il

⁴ R. Galoppini nel commento a uno studio IDC del 2006

⁵ G. Ruffatti “Open source e modelli di business: come creare valore per le PA – il system integrator e il caso Engineering” in “Finalmente Libero”, op. cit

⁶ M. Berra, “FLOSS e politiche dei governi”, in “Finalmente Libero!”, op.cit

⁷ <http://ec.europa.eu/idabc/>

⁸ http://www.cnipa.gov.it/site/_files/bs_Direttiva%20MIT%2019%20dicembre%202003_c.pdf

⁹ “FLOSS: gli indirizzi europei, la normativa italiana e le leggi regionali”, in “Finalmente Libero!”, op.cit

¹⁰ “La produzione del software” in “Finalmente Libero!”, op. cit, pag. 85

sistema operativo più diffuso al mondo, ai software di produttività individuale, come la suite Open Office, MySQL, PostgreSQL, sino ad applicazioni verticali come Compiere (software ERP)¹¹, Moodle (per l'elearning), PHPProjekt.

Tali esempi hanno dimostrato che il software libero può essere un ottimo prodotto, in grado di reggere la sfida con i programmi proprietari e perfettamente utilizzabile in qualunque ambito commerciale.

Ma quali sono le azioni strategiche alla base?

Quale potrebbe essere l'impatto economico dell'open source su una PMI?

¹¹ ERP sta per EnterpriseResource Planning (applicativi per la gestione d'impresa). In tale ambito "Compiere" copre ogni aspetto aziendale: il controllo di gestione, la contabilità generale, il ciclo attivo (listini e ordini del cliente), il ciclo passivo (listini e ordini ai fornitori), i movimenti di magazzino e l'inventario, l'analisi della performance aziendale.

Modelli di business a confronto

«Occuparsi della propria giornata significa occuparsi del proprio business (...) Il business non è semplicemente un fattore, una componente fra le molte che influiscono sulla nostra vita. Le sue idee costituiscono la trama e l'ordito fondamentali e imprescindibili su cui sono tessuti i modelli dei nostri comportamenti»

J. Hillman, 2002

La valutazione e l'adozione di un software è un'attività *time spending*, spesso impegnativa dal punto di vista economico. In molti casi, per un'azienda, una decisione in tal senso in luogo di un'altra può avere pesanti risvolti strategici. Che la finalità primaria - nonché condizione di sopravvivenza sul mercato - di qualsivoglia soggetto economico sia quella di "fare utile" è opinione corrente. Meno diffusa è invece l'idea per cui adottare software open source anziché software proprietario possa configurarsi come una scelta, puramente, di "convenienza economica".

Nella sua accezione più ampia, detta "convenienza economica" include sia aspetti legati a considerazioni inerenti il margine/fatturato rispetto ad un dato valore spesa/investimento, sia ricadute sul fronte organizzativo/gestionale.

Se ciò è valido come premessa, proviamo ad applicare tale accezione all'agire di una PMI attiva nel settore software.

Come ben illustrato da Federico Pozzato sulle pagine di "PC Open", una «software house (o uno sviluppatore) possono agire in modi diversi:

- a. proporre al cliente un programma proprietario già presente nel mercato;
- b. "personalizzare" per il cliente un software esistente;
- c. sviluppare un programma "ex novo"».

Di fronte all'interrogativo: «qual è la strada economicamente più conveniente?» i più risponderebbero, senza troppi dubbi, la prima.

Pozzato vi dedica una risposta più approfondita. «La prima soluzione - spiega - è la più semplice, ma non sempre percorribile. La software house diventa solo il partner commerciale dell'azienda produttrice del programma e si farà pagare dal cliente finale per il servizio di installazione, assistenza e formazione.

La seconda soluzione crea un legame maggiore fra la software house e il cliente finale (...). Anche in questo caso, come nel primo, la software house potrà trarre profitto dalle operazioni di installazione, assistenza e formazione.

La terza soluzione è la meno frequente: difficilmente, infatti, il cliente si accollerà l'intero costo di sviluppo (e successivo mantenimento) di un programma "ex novo"»¹².

Qualunque sia l'azione scelta, nel caso di adozione di software proprietario, emergono due aspetti essenziali:

- la software house (o il singolo sviluppatore) guadagna non tanto vendendo il software, quanto vendendo servizi aggiuntivi (installazioni, personalizzazioni, trasferimento dati, formazione); c'è poi il rischio che quanto sviluppato con un software proprietario sia utilizzabile solo da chi ha, a sua volta, quel programma (magari anche nella stessa versione);
- il cliente finale, dal suo lato, investe denaro per un software fornito con una licenza che non gli permette di fare nessuna modifica per adattarlo alle proprie esigenze o ottimizzarlo.

Entrambi, in sintesi, utilizzano qualcosa su cui non hanno un controllo diretto.

Entrambi, restano vincolati alle scelte del produttore del software.

Se le piccole e medie software house e gli sviluppatori - e qui veniamo alle conclusioni di Pozzato - fanno utili non tanto con la vendita del software quanto con la vendita dei servizi di installazione, personalizzazione, trasferimento dati, formazione e assistenza «per loro utilizzare un software libero o un software proprietario è indifferente dal punto di vista del fatturato in ingresso»¹³.

In più - conclude Pozzato - nel caso di adozione di FLOSS, «avranno comunque un guadagno visto che non dovranno spendere nulla per il software utilizzato».

A tali considerazioni, tuttavia, potrebbe aggiungersi quella secondo cui la vendita di software proprietario procura margini piuttosto bassi, ma in relazione a prodotti di largo consumo e «per il software enterprise vale esattamente l'opposto».¹⁴

Facciamo dunque un passo indietro; e proviamo a verificare l'"indifferenza" di cui ci dice Pozzato a partire da un'analisi comparata dei due modelli economici, supportata, nel paragrafo successivo, da una valutazione più ampia sui costi di produzione/acquisto del software.

Scriva Luca De Biase: «il modello di business del software proprietario si basa su forti investimenti iniziali per produrre software standard da rivendere in

¹² F. Pozzato, op. cit

¹³ F. Pozzato, op. cit

¹⁴ R. Galoppini, 2008

grandi volumi e a più riprese, con versioni successive che obbligano all'aggiornamento e all'ulteriore pagamento dei costi di licenza»¹⁵.

L'acquisto della licenza permette, in genere, di:

- installare il software su una macchina;
- utilizzare il software;
- effettuare una copia di *backup* per uso archivio.

Non è permesso, salvo esplicita concessione, decodificare, decompilare o disassemblare il programma.

Il prodotto generato è indifferenziato: cioè distribuito in versioni tutte uguali per spalmare il costo di realizzazione su numeri alti di copie rendendolo, così, economicamente appetibile.

La cadenza degli aggiornamenti e degli *step* di sviluppo sono decisi dai realizzatori, esclusivamente sulla base degli effetti economici di ciascuna decisione.

Il modello di business dell'open source, invece, riduce i costi di sviluppo iniziali mediante la condivisione della programmazione di base e di quella necessaria agli aggiornamenti: quello che si vende è tutto lo spettro dei servizi professionali correlati al prodotto realizzato.

“Non prodotti, ma servizi” - si sente dire - sintetizzando la questione sulla “res di compravendita”.

«Questo - sottolinea ancora De Biase - consente di lavorare con maggiore attenzione alle esigenze specifiche degli utenti sulla base di un patrimonio di software comune dal quale attingere».

Ciò - commenta - permette di tagliare il software su misura, quasi fosse un abito uscito dalla bottega di un sarto-artigiano.

In più, si può dire che, accanto ad un risparmio iniziale in termini di costi per le licenze e alla gestione di lungo periodo del software, il FLOSS garantisce:

- qualità. A suffragare questa affermazione è il successo di tutti i progetti liberi citati nell'introduzione (vedi Linux, Apache, Open Office, MySQL, Compiere, PHPProject);
- adattabilità e personalizzazione degli applicativi acquisiti;
- tempi di sviluppo e di miglioramento solitamente molto sostenuti, talvolta inarrivabili dai software proprietari. La vera novità di Linux, ad esempio, fu proprio il metodo di lavoro, caratterizzato dal rilascio di versioni aggiornate anche giornaliere¹⁶;

¹⁵ Luca De Biase su [javaopenbusiness.it](http://www.javaopenbusiness.it)

(<http://www.javaopenbusiness.it/Benvenuti+nel+portale+dell%27Open+Source+per+le+Piccole+e+Medie+Imprese/JobEditorial/newsCategory/ViewProcess.jsp?editorialID=3191>)

¹⁶ Sull'importanza della rapidità di aggiornamento Pozzato scrive «muovendosi entro un campo, quello dell'informatica, dove quel che vale oggi domani potrebbe essere messo in discussione da nuovi stravolgimenti, capiamo come velocità e sviluppo siano fattori determinanti per la vita di un software».

- sicurezza. L'apertura del sorgente, infatti, se in prima istanza espone il sistema a rischi, fa sì che il software libero sia oggetto di studio e di attenzione da parte di molti. Il che è una garanzia per la correzione di eventuali *bug*, nonché per la sicurezza complessiva del prodotto¹⁷.

A questi vantaggi si aggiunge quello “non quantificabile” riguardante la libertà - una libertà prescritta e regolamentata a mezzo licenza - di usare, distribuire (gratuitamente o dietro compenso) e modificare il programma per creare ulteriori programmi o migliorare i propri servizi.

Una libertà a dir poco fondamentale per una piccola/media software house, se considerato, come detto poc'anzi, che la sua attività principale gravita attorno alla fornitura di servizi.

C'è un ultimo aspetto, di non poco conto, da considerare. Il nostro paese «è capace di auto-produrre solo per il 10% del software richiesto dal mercato nazionale»¹⁸. Di conseguenza, «il passaggio anche parziale a soluzioni open source¹⁹ potrebbe ridirigere parte della spesa in tecnologie e servizi ICT verso società nazionali di sviluppo e manutenzione del software»²⁰.

Certamente, tali considerazioni vanno “pesate” caso per caso.

Ciò che invece costituisce un prerequisito sempre valido, è un'oculata valutazione del prodotto open da adottare.

È infatti indispensabile:

- evitare di ricorrere a software troppo recenti il cui sviluppo non sia ancora giunto ad una buona stabilità e maturità;
- conoscere il tipo di *community*²¹ alla base del progetto (è sempre attiva? risponde rapidamente ai problemi sollevati?);
- verificare l'esistenza di strumenti di supporto usati per garantire lo sviluppo del software. Ad esempio, la messa a disposizione di un *bugzilla*²² consente di verificare se esistono *bug* e in quanto tempo sono risolti;
- stimare la frequenza degli aggiornamenti e del rilascio di nuove versioni. Un progetto che si muove troppo lentamente può esser

¹⁷ Scrive Lessig: «i sistemi a codice chiuso possono dare l'impressione di un'esposizione minore, mentre l'esposizione reale... può essere molto maggiore (...) Alla lunga, l'apertura del codice aumenta la sua sicurezza... [e] permette agli utenti di fare scelte più informate sulla sicurezza di un sistema».

¹⁸ Berra M., Meo A.R., “Libertà del software, hardware e conoscenza – Informatica solidale 2”. Bollati Boringhieri, 2006, in “Finalmente Libero”, op. cit. pag. 66

¹⁹ “Indagine conoscitiva sul software a codice sorgente aperto nella pubblica amministrazione”, Commissione Meo, 2003

²⁰ In Brasile, ad esempio, il presidente Luiz Ignacio Lula da Silva, ha contraddetto uno studio della Microsoft che tendeva a dimostrare come il software open source costi di più di quello di proprietà dell'azienda di Redmond, sostenendo che anche se fosse vero, «preferirebbe un software che fa rimanere i soldi in Brasile a uno che li porta direttamente negli Stati Uniti».

²¹ “Comunità FLOSS”, Marzano F., Donno V., 2008

²² Il bugzilla è un sistema di trouble – ticketing utilizzato per segnalare i problemi e poi seguirne l'evoluzione su progetti di prodotti open e/o proprietari. Inizialmente sviluppato e usato nel team Mozilla, nel 1998 è stato rilasciato come software open source da Netscape-Communications.

sintomo di un basso numero di sviluppatori, quindi di una comunità poco operativa.

La sponsorizzazione del progetto da una qualsiasi importante azienda, costituisce poi un'ulteriore garanzia; e comunque, in genere, un buon progetto genera attorno a sé un indotto economico di servizi ben pubblicizzato e facile da verificare.

Eppure, al di là di ogni possibile vantaggio (con i relativi, necessari, accorgimenti), M. Berra sintetizza che «la principale ragione per l'adozione del FLOSS è di ordine economico (...) Il software libero consente notevoli risparmi sia nel capitolo degli investimenti sia in quello dei costi di esercizio»²³.

Dunque, laddove secondo Pozzato utilizzare un software libero o un software proprietario è indifferente dal punto di vista del fatturato in ingresso, per la Berra il software libero consente notevoli risparmi (in termini sia di investimenti che di costi d'esercizio).

Due opinioni apparentemente diverse. "Apparentemente" perché la prima è formulata assumendo il punto di vista dell'utilizzatore del FLOSS (sia esso l'utente finale o un'azienda cliente che decida di "adattarlo").

"Utilizzare", scrive infatti Pozzato.

La seconda, invece, si riferisce al produttore di software open.

Un confronto economico corretto dovrebbe essere ad ogni modo compiuto tra il TCO (*Total Cost of Ownership*) delle soluzioni open source e il TCO delle soluzioni proprietarie, considerando il rapporto tra funzioni offerte e il costo complessivo di utilizzo e gestione del pacchetto.

²³ M. Berra, "FLOSS e politiche dei governi", in "Finalmente Libero!", op.cit, pag. 54. L'affermazione è basata su uno studio condotto sulle politiche dei governi nella promozione e adozione del software libero di quasi tutti gli Stati del mondo; paesi che hanno individuato nel risparmio economico la molla principale per l'adozione del FLOSS.

«Il software open source non è “tutto uguale”,
ma anche i modelli di business nell’open source “non sono tutti uguali”»
G. Ruffatti

Tavola 1. ²⁴ MODELLI di BUSINESS OPEN SOURCE – Koenig²⁵

PUNTO di VISTA: chi sviluppa software open source	
MODELLO	DESCRIZIONE
PATROCINIO (<i>patronage</i>)	spingere la “commoditizzazione” di una soluzione. L’obiettivo non è il profitto ma “sostenere” l’adozione di software open source al fine di: accelerare l’adozione di uno standard, trasferire nel mercato i risultati della ricerca, bilanciare il ruolo dominante di un protagonista nel mercato.
OTTIMIZZAZIONE	spingere il risparmio dei costi per vendere valore aggiunto, in genere specifiche soluzioni proprietarie ottimizzate.
DOPPIA LICENZA (<i>dual licensing</i>)	schema ibrido che prevede sia il rilascio del software come licenza libera (spesso la GNU GPL), che la vendita dello stesso su licenza “proprietaria”. La versione di software distribuito con licenza libera e proprietaria può essere la stessa, ma vi sono soluzioni dove il software libero contiene meno funzionalità rispetto ad una versione “enterprise” o “professional” a pagamento, necessaria per la realizzazione di progetti industriali; in quest’ultimo caso l’open source è una caratteristica puramente promozionale.
CONSULENZA	l’open source è utilizzato per vendere servizi complementari, quali formazione, certificazione, supporto, integrazione.
SOTTOSCRIZIONE (<i>subscription</i>)	i ricavi derivano da servizi ricorrenti quali il packaging, il bundling (integrazioni di soluzioni open source in uno specifico stack), la manutenzione.
HOSTED	vendita di servizi (software as a service) basati su soluzioni open source.
EMBEDDED	prevede l’inclusione di software open source in hardware o altro software complesso; costituisce una leva per vendere una soluzione proprietaria.

²⁴ La tabella è costruita a partire dall’analisi dei sette modelli di business dell’open source di Koenig, illustrati da G. Ruffatti in *Finalmente Libero! Software libero e standard aperti per le pubbliche amministrazioni*, AA.VV., Mc Graw Hill, 2008

²⁵ Koenig J., *Seven open source business strategies for competitive advantage*, IT Manager’s Journal, May 2004

«L'adozione del software open source è influenzata dall'incontro di almeno due modelli di business: quello dell'azienda che lo produce e quello di chi lo adotta»
G. Ruffatti

Tavola 2.²⁶ MODELLI d'USO OPEN SOURCE – Letellier²⁷

PUNTO di VISTA: chi utilizza software open source	
MODELLO	DESCRIZIONE
RIUSO	utilizzo di software esistente a minor costo, rispetto al software proprietario, eventualmente per utilizzi episodici, anche con finalità di ricerca di una maggiore qualità.
DOPPIO SORGENTE	alternativa o sostituzione totale oppure parziale di software proprietario, alla ricerca di minore dipendenza o per forzare il fornitore di software proprietario ad una riduzione di prezzi.
OUTSOURCING	utilizzo della comunità open source come terza-parte cui delegare l'arricchimento e la manutenzione del software, con la volontà di rimanere sulla cresta della tecnologia, sfidare i concorrenti e sbarazzarsi di ciò che non costituisce più vantaggio competitivo.
R&D condivisa	sviluppo condiviso con la comunità, considerata al centro del processo di innovazione.

²⁶ La tabella è costruita a partire dall'analisi dei modelli d'uso di Letellier condotta da G. Ruffatti in *Finalmente Libero! Software libero e standard aperti per le pubbliche amministrazioni*, AA.VV., Mc Graw Hill, 2008 (pag. 209). Letellier ha commentato e contestualizzato i modelli di Koenig, aggiungendo, con i suoi "modelli d'uso", il punto di vista dell'utente.

²⁷ Letellier F., *Bringing Open Source Middleware to the Mainstream by Federating Business Ecosystems*, ObjectWeb Executive Committee INRIA, June 2005

Sul TCO (Total Cost of Ownership)

«Contrariamente alla credenza popolare che Open source significhi gratuito e sia un elemento che porta inevitabilmente all'abbattimento degli introiti, posso dire che i nostri ricavi sono aumentati del 13% ogni anno da quando siamo passati quasi completamente all'Open source».

R. Green

Una ricerca di CIO.com, (portale del gruppo IDG)²⁸ che ha analizzato i comportamenti delle aziende nei confronti dell'investimento in IT, rileva che il parametro più usato per valutarlo è il ROI (*Return On Investments*), seguito a distanza dal TCO o Costo Totale di Proprietà²⁹.

Stando alla stessa ricerca, la percentuale di utenti che vede nel ROI il parametro principe per una valutazione di convenienza va diminuendo (dal 68% del 2007 al 59% del 2008), mentre quelli che guardano al TCO sono sempre di più (sono cresciuti dal 32% al 41% nello stesso arco temporale)³⁰.

Questo perché «mentre il metodo del ROI cerca in maniera razionale l'espansione del ritorno economico adottando la soluzione che rende di più al fine dell'investimento (spendo molto per guadagnare di più), l'altro metodo (TCO) mira alla riduzione dei costi; e ogni manager (o persona che si occupa della gestione quotidiana o strategica di un'impresa) sa che le previsioni sui ricavi sono sempre molto più inaffidabili rispetto a quelle sui costi»³¹.

Inoltre la prima ottica «basata solo sul ritorno dell'investimento, risulta tanto meno analitica quanto più ci si trovi in un mercato turbolento e scosso da frequenti shock (...)». Di conseguenza, durante i momenti di crisi e di calo degli investimenti in IT, è prevedibile che sia maggiormente adottato il metodo del Costo Totale di Proprietà.

Già oggi, a differenza di quanto avveniva alla fine degli anni '90, «gli investimenti in IT (*Information Technology* ndr) vengono stanziati solamente di tanto in tanto e le aziende che resistono alla concorrenza sono poche»³².

Per questo, nella comparazione fra i costi del software libero e proprietario «le valutazioni di ordine economico sono oggetto di un dibattito controverso e complesso che riguarda proprio il TCO»³³.

²⁸ <http://www.cio.com/solutions/research-and-analysis>

²⁹ «ROI, Costo Totale di Proprietà e investimento in IT». Riferimento al link: <http://www.javaopenbusiness.it/ruolo+dell%27Open+source+nele+recessioni+economiche/JobEditorial/newsCategoryViewProcess.jsp?editorialID=6935>

³⁰ Per completezza d'informazione va detto che «nel mondo dell'informatica esistono decine di metodi di misurazione dei costi e delle prestazioni dei sistemi e dei programmi». G. Cosenza, ad esempio, presidente di Sinapsi (http://wiki.sinapsi.com/Snapsi_S.p.A.) ha proposto l'indice TAO (Total Account Ownership – Possesso Totale del Cliente) nel senso di dipendenza dei clienti dai fornitori di tecnologia.

³¹ «ROI, Costo Totale di Proprietà e investimento in IT», link cit.

³² «La chimera della New Economy e il budget IT», in <http://www.javaopenbusiness.it/ruolo+dell%27Open+source+nele+recessioni+economiche/JobEditorial/newsCategoryViewProcess.jsp?editorialID=6935>

³³ M. Berra in *Finalmente Libero!*, op. cit., pag. 55

Si tratta di un parametro che prende in considerazione tutti i costi legati al ciclo di vita delle soluzioni IT, inclusi quelli inerenti l'acquisto, l'installazione, la gestione, la manutenzione e la loro dismissione.

«Molti studi, effettuati tra il 2001 e il 2006, sostengono che il TCO del FLOSS (soprattutto GNU/Linux contro Windows, ma anche Web server e DBMS) è inferiore al software proprietario di una percentuale che tipicamente va dal 20% al 50%. Altri studi, per lo più finanziati da Microsoft, sostengono tuttavia il contrario»³⁴.

Senza entrare nel merito d'indagini o ricerche e fondatezza scientifica delle rispettive tecniche d'investigazione, ci si limita qui ad una constatazione d'ordine pratico: prima di adottare un software, sia esso libero o proprietario, sono due le cose – in termini prettamente economici – da valutare:

1. il TCO che, come detto, non s'identifica solo con il prezzo d'acquisto del programma, ma è la somma di tutte le spese e i costi, presenti e futuri, associati alla proprietà e all'uso di materiali e servizi;
2. il tipo di licenza che accompagna il software³⁵.

Ci si sofferma qui sul punto primo. Nel calcolo del TCO rientrano i costi di:

- acquisto del programma (o più precisamente, della licenza);
- adeguamento dei sistemi;
- installazione;
- formazione;
- personalizzazioni;
- manutenzione e aggiornamento;
- eventuale mancata aderenza a standard;
- eventuale cambiamento del programma o della software house partner.

Come si evince dalla lista, nel TCO di un software il costo di acquisto della licenza è soltanto una delle voci. In relazione alle quali - commenta Pozzato - «quello che le aziende clienti spesso non riescono a vedere sono le ultime quattro voci, che a lungo termine possono far lievitare il TCO a livelli impensabili»³⁶.

La conclusione a cui giunge l'editorialista di PC Open è che: «(...) a parità di caratteristiche, il TCO di un software proprietario rischia di essere ben più alto di quello di un software open source, e questa affermazione non dipende quasi per nulla dalla differenza inerente il prezzo d'acquisto».

Questo quantomeno considerando il lato utente/cliente.

³⁴ In Finalmente Libero!, op. cit., pag. 90

³⁵ Si ricordi che alcune licenze del mondo open source si adattano bene al modello di business delle software house tradizionali, poiché consentono di utilizzare il software libero all'interno di progetti proprietari e di conservarne la proprietà intellettuale.

³⁶ F. Pozzato, in "PC Open", "Corso Open source" - Lezione n°2, Marzo 2006

Dal punto di vista di una PMI, o di uno sviluppatore, che utilizza un programma open che non ha creato direttamente si determina, similmente, un abbassamento del TCO, oltre che l'eliminazione della dipendenza da terzi.

Scrive Pozzato «usando programmi liberi la software house abbassa il “suo” costo relativo al software, non solo perché non lo paga, ma anche (e specialmente) perché può lavorare liberamente su di esso senza dipendere contrattualmente dal produttore e dalle sue scelte»³⁷.

L'opinione è condivisa anche da M. Marchesi, dell'Università degli Studi di Cagliari, che precisa: «se il software open source è maturo, e se l'uso del software proprietario da sostituire non è legato fortemente ad altri sistemi software in uso e non sostituibili, il TCO del software open source è normalmente ben inferiore a quello del software proprietario, e tale vantaggio aumenta nel tempo, una volta ammortizzati i costi della transizione»³⁸.

Dalla “Guida Open Source”³⁹ pubblicata da Ancitel emerge, invece, una visione differente.

L'approccio al TCO, col confronto tra FLOSS e software proprietario, è definito come rispondente a «degli scenari di carattere generale che sono propri dell'America e che hanno una vaga somiglianza con i problemi reali(...)».⁴⁰

Inoltre, «il costo maggiore nello sviluppo in ambiente open source – si legge – sono le persone».

Sono cioè coloro che dovranno curare l'attività di sviluppo, siano essi programmatori o manager di progetto; sono coloro che gestiranno i livelli di supporto o ancora quelli che si occuperanno del mantenimento futuro dell'applicazione. “Le persone”, quindi.

Le stesse, dalle quali dipenderanno i costi di sviluppo complessivi.

Più avanti, infatti, la Guida di Ancitel riporta: «conviene sottolineare che il costo di sviluppo di un'alternativa in proprio può ridursi notevolmente, anche in forma drastica, se la strada intrapresa è quella di condividere i costi di sviluppo e di manutenzione dell'applicazione con altre amministrazioni locali».

E se ciò è vero per la PA, lo stesso vale per le piccole e medie imprese, a patto che venga rispettato quell'ipotetico “se”.

Nella tabella, di seguito inclusa⁴¹, vengono elencati e discussi i principali indicatori che formano il TCO del software di sistema e, in relazione a questi, un confronto tra FLOSS e proprietario.

³⁷ F. Pozzato, Lezione n°2.

³⁸ M. Marchesi, “Il modello open source e come le PPAA possono trarne vantaggio”, DIEE Università degli Studi di Cagliari

³⁹ Agenda Digitale Locale. “Guida Open Source. Guida pratica per strategie e servizi Open Source di e-Government nella P.A. locale, Ancitel, 2006

⁴⁰ “Guida Open Source”, op. cit, pag. 13

⁴¹ La tabella è tratta da “La produzione del software”, in “Finalmente Libero”, op. cit, pag. 89.

Tavola 3. Principali voci che formano il TCO del software di sistema, e confronto tra approccio FLOSS e proprietario

COSTO	FLOSS	SW PROPRIETARIO
COSTO INIZIALE	Gratuito o molto ridotto (ad es., distribuzioni GNU/ Linux a pagamento)	In genere mediamente o molto costoso
MANUTENZIONE e AGGIORNAMENTO	Gratuiti o molto ridotti	Dopo il primo anno, la manutenzione annuale vale circa il 15% del costo di acquisto. Ogni 2 - 5 anni il produttore realizza un nuovo prodotto e l'aggiornamento costa circa il 50%
GESTIONE delle LICENZE	Quasi nulli. I rischi di costi legali sono bassi	Occorre attenzione a seguire le clausole e mantenere tutte le prove di acquisto. Rischio di costi legali
COSTO dell'HARDWARE	Spesso (ma non sempre) il FLOSS funziona bene anche su hardware non recente	Sistemi operativi e applicativi spesso richiedono hardware moderno
AMMINISTRAZIONE del SISTEMA	Un singolo amministratore GNU/Linux riesce in genere a gestire il 30% di <i>client</i> in più rispetto a Windows	La disponibilità di amministratori Windows è superiore, anche se il costo stipendiale inferiore rispetto a GNU/Linux
APPRENDIMENTO e USABILITA	I corsi per applicativi FLOSS sono spesso più economici dei corrispondenti proprietari	Le interfacce del software proprietario sono spesso più facili da apprendere in un tempo minore
AFFIDABILITA e SICUREZZA	Se ben configurati, i principali applicativi FLOSS hanno un uptime molto elevato, un rischio di attacchi basso e tempi di ripartenza brevi	Uptime e tempi di ripartenza sono buoni. I costi potenziali dovuti ad attacchi alla sicurezza sono abbastanza elevati
DISPONIBILITA APPLICAZIONI	Sulla piattaforme FLOSS si trovano moltissime applicazioni FLOSS potenti e gratuite. Per problemi specifici, possono non essere disponibili applicazioni	Su Windows, esistono molteplici applicazioni, proprietarie e FLOSS, per qualunque tipo di esigenza

Giro di boa: aspetti critici e prospettive

«Questo movimento deve assolutamente far nascere un modello di business alternativo a quello fondato sui diritti d'autore e che sia in grado di sostenere la battaglia legale e culturale che si sta definendo»

L. Lessig

L'elenco delle "proprietà" del FLOSS (qualità, adattabilità, personalizzazione, tempi di sviluppo e miglioramento, indipendenza da fornitori terzi, sicurezza), unito alle considerazioni sul TCO, paiono definire un percorso tutto sommato lineare per l'adozione dell'open source da parte delle PMI.

Non è così.

Il *frame* di lettura, come pure il quadro complessivo, è ben più ampio. E, come sostiene De Biase, il gioco strategico va ancora definito.

Lo stesso De Biase, nel commento ad un intervento di L. Lessig all'*Open source Business Conference*, scrive: «regalare il software e far pagare il lavoro di progettazione, manutenzione e adattamento alle aziende che lo richiedono è un'idea. Ma potrebbe essere tanto meno redditizia quanto più cresce e migliora il software: perché, migliorando, diventa più facile e gestibile dai non addetti ai lavori, il che fa diminuire il valore dei tecnici che si occupano di installarlo»⁴².

Secondo B. Perens⁴³ i tradizionali modelli di business dell'open source, basati sui servizi, hanno in qualche modo segnato il passo, sono stati superati, e il loro maggior difetto è quello di non riuscire "a scalare". "Non scalano" perché «se un consumatore dovesse rivolgersi per ogni applicativo al *vendor* di quella soluzione sarebbe impossibile per il consumatore fruire davvero dei servizi di qualità che cerca».

In ambedue le opinioni c'è sicuramente del vero.

Eppure, laddove la "visione" di Perens somiglia abbastanza alla "situazione ideale" per un mercato estremamente polverizzato qual è quello italiano, in cui il 97% delle aziende ICT appartiene alla micro-impresa⁴⁴ (cioè conta meno di 9 addetti) dedite per lo più - come si è detto - alla fornitura di servizi professionali che non allo sviluppo, l'affermazione di Lessig, viceversa, si applica ancora molto difficilmente al contesto italiano, dato che crescita e miglioramento del software sono subordinati alla crescita delle community e/o ad un miglioramento della cooperazione fra le stesse.

⁴² De Biase, "Open source come ecosistema", 2007, su [javaopenbusiness](http://javaopenbusiness.it).

⁴³ Il commento è riportato da R. Galoppini, in un'intervista per [javaopenbusiness](http://javaopenbusiness.it). Riferimento al link: <http://www.javaopenbusiness.it/JobEditorial/newsCategoryViewProcess.jsp?editorialID=6360>

⁴⁴ R. Galoppini, "Modelli di business Open Source", in "Finalmente Libero!" op. cit, pag. 200

E qui veniamo forse, ad un primo, benché decisivo, collo di bottiglia dell'intera faccenda.

R. Galoppini, già presidente del Consorzio CIRS Consortium Italicum Ratione Soluta⁴⁵, sottolinea che «quello italiano è un contesto in cui mancano fornitori impegnati in esperienze di condivisione di standard», «né è possibile riscontrare la presenza di soggetti in grado di investire nel marketing di soluzioni open per promuoverne il marchio».⁴⁶

«La stessa riduzione dei costi di produzione - evidenziata dalla Berra - è raggiungibile - ammette Galoppini - ma richiede programmi di partnership in cui l'impresa si metta in gioco con la comunità, condivide con essa esigenze e opportunità (...)». Un'osservazione questa, che fa da eco o, ancor di più, da cassa di risonanza, a quanto letto anche nella "Guida Open Source" di Ancitel. Del resto, come ben illustra Marchesi, e qui s'introduce un aspetto volutamente sin'ora taciuto, le fonti di finanziamento allo sviluppo del software open source sono essenzialmente due:

1. «l'impegno volontario di singoli (programmatori) e talora di organizzazioni (finanziate per lo più con fondi pubblici - Università o Enti di ricerca);
2. un investimento da parte di una ditta (...)»⁴⁷.

Se l'uno, o l'altro, o entrambi, mancano, non c'è business, né gioco strategico alcuno.

Rimandando un approfondimento del primo punto all'oggetto del documento "Le Comunità FLOSS", consideriamo qui la seconda voce: «un investimento da parte di una ditta». In molti, a riguardo, si domanderebbero: «Ma che cosa dovrebbe *motivare* simile investimento?».

Se quanto detto sin qui non costituisce già materia di risposta sufficientemente esauriente, di seguito la risposta che fornisce Marchesi.

«In alcuni casi - spiega - si tratta di ritorni strategici a lungo termine, ad esempio supportando prodotti OS per indebolire un concorrente o per favorire prodotti di altro tipo, in sinergia col prodotto OS».

È il caso, ad esempio, del supporto fornito da Sun allo sviluppo di Open Office per, da un lato, tentare d'indebolire il monopolio di Microsoft nel settore dell'automazione d'ufficio e, dall'altro, avviare la parallela produzione di Star Office, prodotto compatibile con Open Office le cui vendite sarebbero aumentate di molto trainate proprio dall'incremento della quota di mercato di quest'ultimo.

Obiezione lecita: propositi strategici di lungo periodo possono valere per le grandi aziende ma non per le piccole. Verissimo.

⁴⁵ <http://www.consorziocirs.it> Primo consorzio italiano di aziende attive nel settore del software libero e dell'open source

⁴⁶ A tal proposito Galoppini riporta l'esempio relativo all'Open Solutions Summit cancellato per la mancanza di sponsor.

⁴⁷ M. Marchesi, *rif. cit.*

È a questo punto però che Marchesi conclude: «il modello di business OS vero e proprio, aperto anche a ditte medio-piccole, consiste nell'investire nella conoscenza e nello sviluppo di sistemi OS condividendone le spese con altri (la "comunità" di sviluppo), e rivendendo ai clienti servizi di installazione, configurazione, personalizzazione e addestramento, rinunciando a vendere licenze software. In tal modo, i costi sono minori sia per il fornitore che per il cliente, ma gli utili per il fornitore possono essere comunque ben commisurati ai costi».

“Investire in conoscenza” dunque, “condividere le spese di sviluppo”, “fare comunità”, (si potrebbe aggiungere) “promuovere le azioni svolte”: sono questi, in sintesi, gli *asset* strategici su cui una PMI dovrebbe puntare e agire.

Per una conclusione

Per tirare le fila di quanto detto sin qui, si richiamano ora le opinioni salienti descritte in relazione agli interrogativi d'apertura.

Alla domanda iniziale che chiedeva se, a fronte della “liberazione” del codice, vi è un ritorno economico diretto o indiretto, si è data - con Pozzato, Berra e Marchesi - risposta affermativa, nel sostenere il primo che «per una piccola e media software house utilizzare software libero o proprietario è indifferente dal punto di vista del fatturato in ingresso» oppure che «il TCO di un software proprietario rischia di essere ben più alto di quello di un software open source»; la seconda che «il software libero consente notevoli risparmi sia in termini di investimenti che di costi d'esercizio»; il terzo, infine, che «(...) il TCO del software open source è normalmente ben inferiore a quello del software proprietario».

Sempre in questi pareri può esser rintracciata una risposta anche all'altra domanda d'apertura, quella inerente la “soluzione più conveniente” da adottare.

Riguardo invece le “dimensioni” su cui valutare la convenienza, si ricordino i vantaggi elencati: qualità, adattabilità, personalizzazione, sicurezza, tempi di sviluppo e miglioramento sostenuti, e così via.

Per ciascuna di queste affermazioni, tuttavia, sembrerebbe essere emerso anche il suo contraltare.

Si è sottolineato, ad esempio, che se è vero che la vendita di software proprietario procura margini piuttosto bassi in relazione a prodotti di largo consumo, per il software *enterprise* vale esattamente l'opposto.

Si è poi precisato che se alcuni studi sostengono che il TCO del FLOSS è inferiore a quello del software proprietario, altri dichiarano apertamente il contrario.

È emerso, infine, che, secondo taluni, il costo maggiore nello sviluppo in ambiente open source sono “le persone”; per cui i costi di sviluppo di una alternativa in proprio possono sì essere ridotti notevolmente, ma solo a condizione di condividere i costi di sviluppo e manutenzione.

Questo proprio quando – come evidenziato con Galoppini – il contesto italiano è caratterizzato dalla mancanza di «fornitori impegnati in esperienze di condivisione di standard», come pure di «(...) soggetti in grado di investire nel marketing di soluzioni open (...)».

Si capisce che in un contesto simile «la riduzione dei costi di sviluppo iniziali mediante la condivisione della programmazione di base e di quella necessaria agli aggiornamenti» rischia di valere solo per alcune nicchie isolate; la

“garanzia” di sponsorizzazione del progetto da una qualsiasi grande azienda può venire facilmente meno; la stabilità o la maturità di un prodotto è destinata a languire per assenza di community stabilmente operative; e similmente può accadere per quel che attiene la frequenza d’aggiornamento o il rilascio di nuove versioni.

In tutte le osservazioni c’è del vero; e ciascuna altro non è che *una* di quelle “dimensioni critiche” del processo di scelta di una soluzione open source la cui analisi si è premissa come fine ultimo di questo lavoro.

Il gioco strategico va ancora definito.

Sul “come” e “perché” la risposta è demandata all’iniziativa e allo spirito imprenditoriale di molti.

Bibliografia su FLOSS e modelli di business

- [1] Afuah A., Tucci C. L., Modelli di e-business: acquisire vantaggi competitivi con organizzazioni internet-based, McGraw-Hill, 2002
- [2] Benkler Y., La ricchezza della rete. La produzione sociale trasforma il mercato e aumenta la libertà, Università Bocconi, 2007
- [3] Bertelè U., Chiesa V., Noci G., Creare valore con la rete: innovazioni gestionali e opportunità di business nel post New Economy, Il Sole 24 Ore, 2002
- [4] Berra M, Meo A. R., Informatica solidale. Storia e prospettive del software libero, Ed. Bollati Boringhieri, 2001
- [5] Berra M., Meo A. R., Libertà di software, hardware e conoscenza. Informatica solidale 2, Ed. Bollati Boringhieri, 2006
- [6] Bonaccorsi A., Rossi C. (a cura di), Economic perspectives on open source software. Intellectual property, knowledge-based communities, and the software industry, Franco Angeli, 2007
- [7] Bonaccorsi A., Rossi C., L'Economia degli standard e la diffusione delle tecnologie. L'Open Source non è un assurdo economico, Laboratorio di Economia e Management Scuola di Studi Superiore Sant'Anna, Pisa, 2002
- [8] Bonaccorsi A., Rossi C., Why Open Source software can succeed. Research Policy Special Issue on Open Source Software Development, Laboratorio Economia e Management Scuola di Studi Superiore Sant'Anna, Pisa, 2003 - <http://opensource.mit.edu/papers/rp-bonaccorsirossi.pdf>
- [9] Butera F., Il campanile e la rete : l'electronic business e le piccole e medie imprese in Italia, Editore Il Sole 24 Ore, 2001
- [10] Centro Nazionale per l'informatica nella Pubblica Amministrazione, (CNIPA) Rapporto conclusivo, Gruppo di lavoro "Codice sorgente aperto" ("Open Source"), 2004
- [11] CIPA (Convenzione Interbancaria per i Problemi dell'Automazione), La valutazione delle opportunità e dei costi connessi con l'adozione dei prodotti open source, 2004 - http://www.cipa.it/docs/rapporti/open_source/opensource.pdf
- [12] Corso Open Source. Lezione 1, 2, 3, di Pozzato F., in "PC Open", Marzo 2006
- [13] Daffara C., Esperienze europee e valutazione dell'OSS a livello economico. Innovaction, Udine, 2007
- [14] Di Bona, Ockman, Stone, Open Sources. Voci dalla rivoluzione Open Source, Ed. Apogeo, 1999
- [15] Formenti C., Mercanti di futuro, Einaudi, 2002
- [16] Formenti C., Not Economy, Etas, 2003

- [17] Ganzaroli A., Pilotti L., Natura e dinamica dell'open source (OS) tra creatività, ecologie del valore e fiducia, Quaderni AIDA, Giuffrè editore, 2005
- [18] Ganzaroli A., Pilotti L., Rileggere il marketing – strategie informative e gestione della conoscenza, CEDAM, 2006
- [19] Garfinkel S. L., Stallman R. M., Kapor M., Perché i brevetti sono dannosi per il software, Issues in Science and Technology - http://www.pluto.it/files/meeting1999/atti/nopatents/brevetti/docs/issues_it/issues_it.html
- [20] Green E. L., Economics of Open Source Software, 2002 - <http://badtux.org/home/eric/editorial/economics.php>
- [21] Guardagli M., Costo del software in ambiente Linux e Windows, Gruppo Tomware, 2003
- [22] IDABC, Migration to open source software – Beaumont Hospital Dublin, Ireland, 2004
- [23] IDABC, Open Source Software Study – OS on Vienna's Desktop Workplaces, 2004
- [24] IDABC, Open Source drives Italian eGovernment, 2004
- [25] Koenig J., Seven open source business strategies for competitive advantage, IT Manager's Journal, 2004
- [26] Locke J., Soluzioni open source per la PMI, Tecniche nuove, 2005
- [27] Marchesi, Concas, De Petra, Marzano, Zanarini, Finalmente Libero! Software libero e standard aperti per le pubbliche amministrazioni, Mc Graw Hill, 2008 - <http://finalmentelibero.ning.com/>
- [28] Marchesi M., Il modello open source e come le PPAA possono trarne vantaggio, DIEE Università degli Studi di Cagliari, 2007
- [29] Martin F., Modelli di business per Linux e open source. Il nuovo modello competitivo: scenari strategici e operativi, Ed. Personal Education, 2003
- [30] Meo R., Le potenzialità tecnologiche dell'open source. Open Source Software: lo Sviluppo delle Tecnologie e dei Mercati, convegno Pisa, dicembre 2000
- [31] Ministero per l'Innovazione e le Tecnologie, Indagine conoscitiva sul software a codice sorgente aperto nella Pubblica Amministrazione, rapporto della Commissione, 2003
- [32] Moody G., Codice ribelle, Ed. Hops, 2002
- [33] Muffatto M., Faldani M., Open source. Strategie, organizzazione, prospettive, Il Mulino, 2004
- [34] Open source e PMI: i risultati della ricerca TeDIS a cura di Marco Bettiol, aprile 2005
- [35] Osterwalder A., Pigneur Y., Tucci C., Clarifying Business Models: origins, present, and future of the concept, Communications of AIS, Vol.15, maggio 2005

- [36] Raymond E., La cattedrale e il bazar, Apogeo, 1999
- [37] Ruffatti G., Cazzin G., Il futuro dell'open source a supporto del Business delle aziende e il progetto SpagoBI, quaderni di management n.21, maggio-giugno 2006
- [38] Sanseverino G., Le licenze free e open source, Ediz. Scientifiche, 2007
- [39] Stallman R., Software libero, pensiero libero (Vol. I), Ed. Stampa Alternativa/Nuovi Equilibri, 2003
- [40] Stallman R., Software libero, pensiero libero (Vol. II), Ed. Stampa Alternativa/Nuovi Equilibri, 2004
- [41] Woods D., Guliani G., Open Source for the enterprise, O'Reilly, 2005
- [42] Julie Giera, The Costs And Risks Of Open Source, Forrester Research, aprile 2004
- [43] Zoppelletto M., Il modello open source vs il modello proprietario nel mercato del software, Copyleft-italia.it, 2005 - <http://www.copyleft-italia.it/pubblicazioni/zoppelletto.pdf>

Linkografia su FLOSS e modelli di business

- [a] <http://guide.conecta.it>
- [b] <http://www.istat.it>
- [c] <http://www.univiu.org/research/tedis>
- [d] <http://ec.europa.eu/idabc/> IDABC (Interoperable Delivery of Pan-European eGovernment Services to Public Administrations, Business and Citizens) è il programma di azione comunitaria volto a sostenere e promuovere servizi di governo elettronico, nonché l'uso dell'open source software
- [e] <http://www.paal2007.it> rende disponibili gli atti del convegno PAAL 2007, Pula (Cagliari)
- [f] <http://www.paal2008.it/> rende disponibili gli atti del convegno PAAL2008, Pula (Cagliari)
- [g] <http://www.pmi.it> sito per la piccola e media impresa e l'innovazione tecnologica
- [h] <http://www.distrettopmi.it> portale dedicato alle piccole e medie aziende
- [i] <http://www.ossipa.cnipa.it> osservatorio sull'open source del Cnipa (Centro Nazionale per l'Informatica nella Pubblica Amministrazione)
- [j] <http://www.javaopenbusiness.it> portale curato da Sun in collaborazione con l'Università di Venezia
- [k] <http://www.consorziocirs.it> primo consorzio italiano di aziende attive nel settore del software libero e dell'open source
- [l] <http://www.opensource.org> sito web ufficiale della Open Source Initiative (OSI) associazione no profit per la diffusione del modello basato sull'Open Source
- [m] <http://www.gnu.org> sito web ufficiale del progetto GNU avviato da Richard Stallman
- [n] <http://www.fsf.org> sito web della Free Software Foundation, fondata da R. Stallman per diffondere il software libero
- [o] <http://freshmeat.net> archivio di software open source per Unix e di software open source multiplatforma
- [p] <http://sourceforge.net> il più grande archivio ed ambiente di sviluppo per codice e applicazioni open source
- [q] <http://www.cnipa.gov.it> sito del Centro Nazionale per l'Informatica nella Pubblica Amministrazione
- [r] <http://flossmetrics.org> dal progetto FLOSSMETRICS che ha clusterizzato alcune macro-tipologie di approcci al FLOSS partendo da un insieme di 120 aziende
- [s] <http://www.tecnoteca.it/osz> open source zone del portale delle tecnologie informatiche. Tecnoteca srl è una azienda di Udine che sviluppa soluzioni Web Open Source

- [t] <http://www.freesoftwaremagazine.com> rivista online dedicata al software libero
- [u] <http://www.gartner.com> sito aziendale, Gartner è una delle organizzazioni leader nelle ricerche di mercato
- [v] <http://www.idc.com> sito aziendale, Idc è un'organizzazione americana indipendente, con sede a Framingham nel Massachussetts, che fa ricerche di mercato
- [w] <http://www.consulentelegaleinformatico.it> portale dedicato alle problematiche legali nel mondo dell'informatica
- [x] <http://www.netcraft.com> sito aziendale, la Netcraft svolge ricerche sui siti web attivi in collaborazione con i principali motori di ricerca
- [y] <http://www.softwarelibero.it> associazione senza scopo di lucro che ha come obiettivi principali la diffusione del software libero in Italia ed una corretta informazione sull'argomento

Ringraziamenti

- Si ringrazia Roberto Galoppini - <http://robertogaloppini.net/> - per i preziosi suggerimenti.
- Si ringraziano tutte le persone di cui abbiamo citato le opere: ci sono state preziose e da ognuna abbiamo imparato qualcosa.
- Questo documento è stato scritto con Open Office versione 2.1 per Mac OS X.

Indice delle Tavole

Tavola 1. MODELLI di BUSINESS OPEN SOURCE – Koenig.....	9
Tavola 2. MODELLI d'USO OPEN SOURCE – Letellier.....	10
Tavola 3. Principali voci che formano il TCO del software di sistema, e confronto tra approccio FLOSS e proprietario.....	14

*Questo documento, nelle sue parti originali, è coperto da licenza Creative Commons:
Attribuzione, Non commerciale, Condividi allo stesso modo*

<http://creativecommons.org/licenses/by-nc-sa/2.5/it/legalcode>

