

ECONOMIA E MANAGEMENT DELL'OPEN SOURCE SOFTWARE. ALCUNE NOTE DI RICERCA*

Paola Giuri* e Salvatore Torrisi†

*Scuola Superiore Sant'Anna, Pisa

†Università di Camerino e Scuola Superiore Sant'Anna, Pisa

1. Introduzione

Il Laboratorio di Economia e Management della Scuola Sant'Anna sta lavorando ad un progetto di ricerca sull'economia e il management dell'Open Source Software (OSS). Questo paper riassume alcuni risultati preliminari derivanti dalla prima fase dell'indagine. In questa fase di ricerca abbiamo effettuato una ricognizione dei problemi più rilevanti dell'OSS: l'organizzazione e la performance del processo di sviluppo del software, gli incentivi all'innovazione in un contesto di OSS, i principali modelli di licenza e di business basati sull'OSS. Le fasi successive della nostra ricerca si concentreranno sulla raccolta di nuova evidenza empirica circa l'uso di OSS e di software proprietario nell'industria manifatturiera e nella pubblica amministrazione in Italia.

La prima fase del progetto ha avuto come obiettivi la ricognizione della letteratura esistente, l'elaborazione di dati relativi alla diffusione di alcuni prodotti OSS e il confronto tra diversi modelli di licenza in ambiente OSS. Una versione completa dei risultati di questa prima fase di indagine è riportata in Giuri, Rocchetti e Torrisi (2002).

Questo paper solleva una serie di problemi che riguardano la sostenibilità di lungo periodo dell'OSS come modello economico, nella consapevolezza che una valutazione approfondita di tali problemi richiederebbe un'analisi teorica ed empirica che a nostro avviso è ancora in uno stato embrionale nella letteratura economica. Anziché fornire delle vere e proprie conclusioni quindi questo scritto vuole offrire spunti di riflessione e indicare alcune direzioni di ricerca futura.

Questa prima rassegna del fenomeno OSS ha messo in evidenza alcuni aspetti importanti che vale la pena anticipare:

- 1) l'OSS rappresenta un modello di organizzazione della produzione e della commercializzazione del software che contiene elementi di novità rispetto al software proprietario. L'aspetto più originale riguarda l'accesso al codice sorgente e la possibilità da parte degli utilizzatori di modificare, adattare e ridistribuire il software originale senza alcun corrispettivo di licenza ('copyleft'). La disponibilità/modificabilità del codice sorgente da parte degli utilizzatori accentua la natura 'evolutiva' e 'collettiva' tipica del processo di sviluppo del software;
- 2) esistono diversi modelli di licenza dell'OSS che si differenziano tra loro per il livello di protezione della 'libertà' d'uso del codice sorgente; la forma più estrema di protezione del diritto di libera distribuzione contro eventuali comportamenti opportunistici da parte di produttori di software proprietario è rappresentata dalla GPL (General Public Licensing), secondo la quale qualunque lavoro che deriva da (o semplicemente contiene) software coperto da GPL deve essere distribuito alle stesse condizioni di licenza ('viral clause'). L'adozione della GPL è

* Si ringraziano Giuseppe Attardi, Università di Pisa, Alfonso Fuggetta, Politecnico di Milano e Guido Scorza, CIRFID, Università di Bologna, per aver condiviso con noi le loro conoscenze tecniche e legali sull'open source software. Si ringraziano inoltre Icube S.r.l., Ksolutions S.p.A., Microsoft Italia, ed MLX S.r.l. (MadeinLinux) per aver fornito informazioni e discusso alcuni risultati preliminari della ricerca.

piuttosto problematica dal punto di vista dei produttori commerciali e questo ha spinto la comunità OSS a sviluppare modelli meno restrittivi come la LGPL (Library o 'Lesser' GPL), la quale permette di produrre software proprietario a partire da software aperto (es. librerie di routine);

- 3) dal punto di vista tecnologico non sono ancora emersi significativi esempi di innovazioni radicali introdotte in ambiente OSS - molti prodotti, come Linux, rappresentano solo innovazioni incrementali rispetto a innovazioni radicali prodotte precedentemente da produttori di software proprietario (come AT&T e Xerox); non è del tutto chiaro quindi se il sistema di incentivi proprio di questo nuovo 'paradigma' basato su un modello di sviluppo distribuito sia adeguato per la creazione di innovazioni complesse e costose quali quelle prodotte in passato dai produttori di software proprietario o da laboratori di ricerca pubblica;
- 4) il numero elevato di contributori e utilizzatori che possono visionare il codice sorgente viene ritenuto dai sostenitori dell'OSS un fattore che rende questo tipo di software generalmente più robusto, affidabile e sicuro rispetto al software proprietario; l'evidenza disponibile a questo riguardo tuttavia è ancora molto scarsa e non conduce a risultati univoci;
- 5) a differenza di Microsoft, che contrappone nettamente il proprio modello di business e di tutela della proprietà intellettuale a quello dell'OSS, alcune imprese informatiche affermate, come IBM, Sun e SAP, hanno adottato una strategia di 'ibridazione' che punta a sfruttare la convivenza tra OSS (es. il sistema operativo Linux o il database MySQL) e i loro prodotti proprietari (es. gli application servers di IBM e la piattaforma di servizi on-line MySAP). Come diremo, queste strategie riflettono la diversa posizione di ciascuna impresa affermata nel mercato del software, e specialmente in quello dei sistemi operativi.

2. Nascita ed evoluzione storica dell'OSS

La nascita ufficiale del movimento del software libero o aperto risale ai primi anni '80, nonostante prima di questo periodo esistevano forme di libera circolazione di codice sorgente tra sviluppatori. Nel 1983 Richard Stallman del MIT ha fondato la Free Software Foundation (FSF) per promuovere lo schema di licenza GNU/General Public Licensing (GPL), con l'obiettivo di proteggere il software libero dall'appropriazione da parte delle imprese commerciali, garantendo l'accesso al codice sorgente e la possibilità per gli utilizzatori di farne copie, di modificarlo e di ridistribuirlo senza pagare né far pagare licenze d'uso. Ispirandosi a questo principio, nell'agosto 1991 Linus Torvalds iniziò un processo condiviso di sviluppo del software mettendo in rete una versione free del kernel Unix e chiedendo contributi alla comunità per individuare e correggere eventuali 'buchi'. Dopo poche settimane era pronta la versione di Linux 0.01. Nel 1998 un gruppo di fautori del software libero - tra cui Todd Anderson (Foresight Institute), Eric Raymond (autore dell'ormai celebre libro "The Cathedral and the Bazaar"), John Hall e Larry Augustin (Linux International) - hanno creato la Open Source Initiative (OSI), che ha fissato dei criteri generali da applicare alla distribuzione di software aperto. Tali criteri sono contenuti nella Open Source Definition e nell'OSI Certification Mark. L'obiettivo della OSI era di rendere il software libero più visibile nella comunità di produttori commerciali, aumentandone quindi il mercato potenziale, al di là della ristretta cerchia di utilizzatori pionieri (Raymond, 2001). Uno dei cardini di questo nuovo approccio consiste nella rinuncia alla clausola di viralità propria della GPL ("the license must not insist that all other programs distributed on the same medium must be open-source software", www.opensource.org/docs/definition.html).

Nonostante le differenze accennate, per semplicità d'ora in poi ci riferiremo all'OSS per intendere indifferentemente software libero o software aperto.

Come abbiamo già anticipato, per OSS si intende software distribuito con il codice sorgente. Il software aperto può essere distribuito gratuitamente sotto condizioni di licenza che garantiscono agli utilizzatori il diritto di usare, copiare, modificare e distribuire il codice sorgente. Tuttavia l'OSS non deve necessariamente essere distribuito gratuitamente. I distributori possono, infatti, fissare un prezzo che copre i costi di distribuzione del codice sorgente originario (es. il kernel di Linux), eventuali costi di

sviluppo di software addizionale (es. interfacce grafiche e driver per l'installazione di periferiche), i costi di assemblaggio del codice eseguibile e i servizi di supporto collegati. Ma nessun diritto di utilizzazione dell'opera può essere richiesto al cliente nell'ambito dell'OSS.

Come ha notato Eric Raymond, il processo di sviluppo dell'OSS ha le caratteristiche di un 'bazar' dove chiunque (sviluppatori, utilizzatori) può entrare in contatto con un progetto di sviluppo, scaricare il codice sorgente, apportare modifiche (ulteriore sviluppo di linee di codice, debugging, ecc.) e spedirle al project leader, senza ricevere un compenso monetario. Il compito di certificazione e distribuzione delle modifiche proposte è normalmente svolto da un limitato numero di *core developers* che affiancano i *project leaders*.

Quali sono le implicazioni di questa particolare organizzazione del processo di sviluppo per la qualità del software? Secondo i fautori dell'OSS questo processo di sviluppo permette maggiore controllo della qualità, stabilità e sicurezza del software. Oltre al numero potenzialmente elevato di sviluppatori che partecipano al processo di sviluppo, secondo i sostenitori dell'OSS un elemento che gioca a favore della qualità del software aperto consiste nelle forti motivazioni di sviluppatori e utilizzatori, che scelgono liberamente di contribuire all'ispezione e correzione di errori del software aperto secondo la logica del 'Bazar'. Nel modello tradizionale di sviluppo ('la cattedrale') invece la rigida divisione gerarchica del lavoro tra sviluppatori e il numero limitato di contributori ridurrebbe le possibilità di controllo e di correzione di errori e allungherebbe i tempi di sviluppo (Raymond, 2001). In realtà, non esiste una ben documentata evidenza empirica che dimostri la relazione tra qualità del software, tempi di sviluppo e modalità di sviluppo. Inoltre, anche nel campo del software proprietario, ormai da tempo si sono diffusi modelli di sviluppo che permettono una elevata flessibilità e creatività da parte degli ingegneri del software (es. il rapid prototyping, l'incremental e evolutionary development e il ciclo di vita a spirale) (Fuggetta, 2001). Inoltre, vi sono limiti organizzativi che frenano la crescita dimensionale dei progetti OSS e ne allungano i tempi di sviluppo (Cubranic e Booth, 1999).

Infine, sul piano della sicurezza, l'apertura del codice sorgente non è condizione sufficiente per garantire livelli più elevati di sicurezza. Per esempio, una recente indagine condotta da Netcraft conclude che 'essendo oltre il 50% i siti Internet potenzialmente vulnerabili, esistono le condizioni per un'epidemia di attacchi sia verso siti basati su Microsoft IIS che su siti Apache' (<http://www.netcraft.com/survey/>, 1 luglio 2002).

3. Diritti di proprietà intellettuale e modelli di licenza

Nell'industria del software la tutela dei diritti di proprietà intellettuale (IPR) è storicamente debole rispetto ad altre industrie come, per esempio, quella farmaceutica. Questo dipende dalla giovane età dell'industria del software e dalle stesse caratteristiche della tecnologia sottostante. Il copyright rappresenta la forma più comune di protezione legale del software. L'estensione della normativa sul copyright al software negli USA risale ai primi anni '80. Ma in Europa tale normativa è stata sviluppata solo negli anni '90 a livello comunitario e successivamente recepita dai maggiori paesi dell'Unione. Nel corso degli anni '80 e '90 negli USA si è sviluppata anche una giurisprudenza favorevole alla brevettabilità del software, pur in assenza di una esplicita normativa. In questo periodo in USA oltre 80 mila brevetti sono stati concessi a invenzioni collegate al software. In Europa è in fase di approvazione una direttiva che disciplina la brevettabilità del software per i paesi dell'Unione.

Il dibattito sulla tutela legale del software vede contrapposti i sostenitori di un regime di protezione legale forte, che sottolineano l'importanza degli incentivi all'innovazione, ed i sostenitori di una protezione legale debole, che enfatizzano i benefici sociali di elevati tassi di entrata di nuove imprese, della competizione fra standard tecnologici differenti e della diffusione tecnologica. Tra i sostenitori del regime di protezione debole si sostiene che in settori caratterizzati da forti esternalità di rete e nei quali la competizione fra standard è strutturalmente debole, come per esempio quello dei sistemi operativi, i diritti di proprietà intellettuale dovrebbero essere limitati per garantire l'entrata e la competizione di tecnologie complementari all'interno dello standard (per esempio tra software applicativi) (Merges, 1996; Cohen e Lemley, 2001). Secondo questa linea di pensiero una protezione forte non è condizione

necessaria per garantire un tasso di cambiamento rapido dal momento che molte innovazioni importanti nel software sono state introdotte prima degli anni '80, quando ancora il regime di tutela degli IPR in questo settore era molto debole (Merges, 1996). Inoltre un regime di protezione forte può produrre effetti negativi sul tasso di cambiamento tecnologico poiché molte innovazioni di rilievo in questo settore sono state introdotte da nuove imprese (Prusa e Schmitz, 1991).

Nell'attuale sistema legale, copyright, brevetti e segreto commerciale non sono esclusivi poiché sia negli Stati Uniti che in Europa, con delle differenze, la stessa invenzione può essere protetta da questi tre istituti giuridici. In particolare, l'assenza (con il copyright) o la debolezza (con i brevetti) dell'obbligo di rivelare il codice sorgente rende possibile coprire attraverso il segreto commerciale il codice sorgente e di prevedere schemi di licenza per l'uso temporaneo del codice eseguibile.

Il software proprietario è distribuito con contratti di licenza che in genere non permettono la possibilità di copiare, modificare e distribuire il software (la distribuzione non è consentita a meno che il licenziatario paghi un prezzo al detentore del copyright). Inoltre, il software proprietario è generalmente protetto dal segreto commerciale.

Lo sviluppo del mercato del software, specialmente dei pacchetti software, è stato favorito dallo sviluppo di un regime di proprietà intellettuale che, come abbiamo visto, ha assunto una sempre più precisa fisionomia negli ultimi decenni. Tale regime costituisce infatti il tessuto istituzionale entro cui la tecnologia software può essere trasferita tra individui e organizzazioni attraverso diverse forme contrattuali (schemi di licenza). Il software aperto mette in discussione questo tessuto istituzionale, contestando specialmente lo sfruttamento dei diritti di proprietà intellettuale mediante contratti di licenza che impongono condizioni restrittive agli utilizzatori. Su queste basi critiche, il movimento OSS ha costruito diversi nuovi schemi di licenza alternativi al modello tradizionale del software proprietario. Per un confronto approfondito tra diversi modelli di licenza e di distribuzione del software proprietario e non proprietario si rinvia a Giuri, Rocchetti e Torrisi (2002).

In questo paragrafo ci limitiamo ad accennare alcune caratteristiche dei modelli di licenza OSS più diffusi: GPL, LGPL, BSD e MPL/NPL.

Le caratteristiche essenziali della licenza GPL sono state esaminate prima. La licenza LGPL presenta le stesse caratteristiche della GPL ma, a differenza di quest'ultima, può essere utilizzata insieme a software commerciale proprietario che usa librerie LGPL purchè queste librerie non vengano modificate e non vengano incluse nel codice eseguibile. Quindi la clausola di 'viralità' in questo caso vale in forma limitata.

La licenza BSD (Berkeley Software Distribution) rappresenta forse il modello più permissivo nell'ambito delle licenze approvate dall'OSI. Infatti essa permette agli utilizzatori di modificare e distribuire il codice sorgente senza limiti e quindi permette anche che il software venga modificato per essere venduto secondo schemi di licenza tradizionali. L'unica condizione che viene imposta in questo caso è che il software deve riportare una nota di copyright che cita tutti i contributori.

Le licenze MPL (Mozilla Public License)/NPL (Netscape Public License) riguardano Netscape (AOL) e la comunità di sviluppo Mozilla.org (browser). Questi modelli di licenza, seppure con delle differenze, permettono in una certa misura a Netscape di combinare il software proprietario con software open e di distribuire eventuali modifiche dello stesso software open secondo modelli di licenza tradizionali.

La tabella 1 elenca i modelli di licenza più diffusi nell'ambito dell'OSS. Essa riporta la distribuzione dei progetti riportati nella banca dati SourceForce per tipo di licenza adottata. Dalla tabella risulta evidente che la stragrande maggioranza delle licenze adottate in questo ambito sono approvate dall'OSI. E' altrettanto evidente che il modello di licenza dominante è il GPL/GNU, quindi quello che presenta il maggior grado di protezione del principio di *copyleft*. Va chiarito che oltre l'80 per cento dei progetti contenuti nel database Sourceforge non ha ancora raggiunto lo stadio di produzione. Quindi i dati della tabella 1 rappresentano un indicatore molto parziale della effettiva diffusione commerciale di diversi modelli di licenza.

In che misura questi diversi modelli di licenza possono essere utilizzati da produttori e distributori di software commerciale? Per i produttori di software commerciale la licenza GPL impone vincoli troppo

stringenti. La “clausola virale” infatti impedisce di combinare software aperto con software proprietario. Licenze come la MPL e la BSD sono più flessibili. A differenza della licenze GPL le licenze BSD e MPL/NPL (e in parte anche la stessa LGPL) infatti non impongono clausole di contaminazione agli utilizzatori e come tali sono pienamente compatibili con il software proprietario. La licenza BSD in particolare non impone restrizioni sulla disponibilità del codice e quindi presenta il grado più basso di protezione della *openness* in quanto il codice distribuito può essere appropriato facilmente da sviluppatori o distributori di software proprietario. Quindi la BSD protegge troppo poco i produttori di software aperto da comportamenti opportunistici da parte di produttori di software commerciale.

Ma perché diversi produttori di software proprietario che hanno aperto alcuni loro prodotti ricorrono a schemi di licenza come la GPL e la LGPL? Questo tipo di licenze possono essere utili anche a produttori di software proprietario che intendono aprire il proprio software ma nello stesso tempo vogliono evitare che altri si appropriino della tecnologia per trasformarla in software proprietario. Un caso emblematico è rappresentato da SAP DB, distribuito come software aperto con la licenza GPL. Probabilmente SAP si aspetta che ulteriori significativi miglioramenti del prodotto possano venire principalmente dalla comunità di sviluppatori aperta anziché dai propri laboratori di R&S. Inoltre, come abbiamo accennato prima, questo prodotto non rappresenta la fonte principale di ricavi per SAP. Grazie alla clausola virale SAP DB e tutti i suoi miglioramenti futuri saranno protetti da possibili tentativi di appropriazione da parte di concorrenti commerciali. E, d'altra parte la maggiore diffusione di questo DB potrebbe offrire a SAP nuove opportunità di ricavi derivanti da servizi di supporto ai clienti e da prodotti complementari come l'ERP.

Tutti i modelli di licenza OSS prevedono la possibilità di fissazione di un prezzo per la distribuzione, e quindi offrono teoricamente delle opportunità di profitto centrate sulla vendita di servizi o prodotti complementari rispetto al software in quanto tale. Si tratta di modelli di business descritti dalla recente letteratura sull'internet economy – es. *Support Seller* (servizi) e *Widget Frosting* o *Loss Leader* (prodotti complementari) (Hecker, 2000).

Tab. 1. Diffusione dei principali modelli di licenza OSS

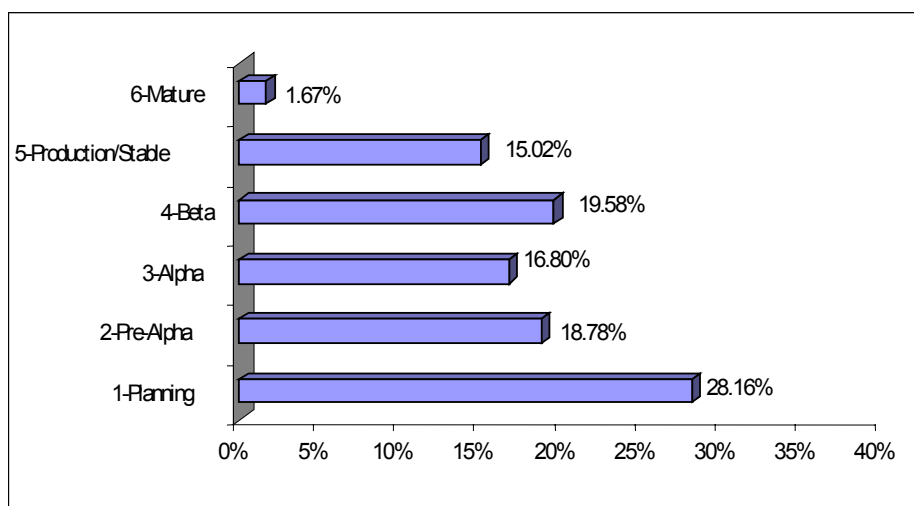
LICENZA	N°	%
Public domain	856	3,11
Other/Proprietary licence	549	1,99
OSI approved licenses	26124	94,90
GNU General Public License	19096	69,37%
GNU Library or Lesser General Public License (LGPL)	2653	9,64%
BSD License	1812	6,58%
Artistic License	767	2,79%
MIT License	420	1,53%
Apache Software License	335	1,22%
Mozilla Public License 1.0 (MPL)	227	0,82%
Mozilla Public License 1.1 (MPL 1.1)	160	0,58%
Python License (CNRI Python License)	158	0,57%
Qt Public License (QPL)	139	0,50%
zlib/libpng License	129	0,47%
Common Public License	46	0,17%
IBM Public License	37	0,13%
Sun Industry Standards Source License (SISSL)	28	0,10%
Jabber Open Source License	20	0,07%
Apple Public Source License	17	0,06%
Nethack General Public License	16	0,06%
Intel Open Source License	11	0,04%
Python Software Foundation License	11	0,04%
Ricoh Source Code Public License	6	0,02%
Eiffel Forum License	5	0,02%
MITRE Collaborative Virtual Workspace License (CVW)	5	0,02%
Nokia Open Source License	5	0,02%
Sleepycat License	4	0,01%
Sun Public License	4	0,01%
Open Group Test Suite License	3	0,01%
University of Illinois/NCSA Open Source License	3	0,01%
W3C License	3	0,01%
Zope Public License	3	0,01%
Vovida Software License 1.0	1	0,00%
Motosoto License	0	0,00%
X.Net License	0	0,00%
TOTALE	27529	100,00 %

Fonte: ns. elaborazioni su dati SourceForge (Giugno 2002)

4. Diffusione dell'OSS

L'elevato numero di progetti ospitati nel sito web SourceForge.net – oltre 40 mila – indica il considerevole interesse nel software aperto da parte degli sviluppatori. Tuttavia a giugno 2002 molti dei progetti erano ancora allo stadio preliminare, il 15,02 % aveva raggiunto lo stadio di produzione (prima release ufficiale) e solo l'1,67 % aveva raggiunto lo stadio di maturità (release successive) (cfr. Fig. 1).

Fig. 1. Stadio di sviluppo di progetti OSS



Fonte: Ns. Elaborazioni su dati www.Sourceforge.net.

Queste semplici statistiche tuttavia non forniscono informazioni sul tasso di sopravvivenza o sulla probabilità di nuovi progetti OSS di raggiungere stadi di sviluppo successivi - produzione e maturità. Inoltre, esse non forniscono alcuna evidenza circa l'effettivo successo commerciale degli stessi progetti.

Per avere un quadro approssimativo dell'importanza commerciale dell'OSS abbiamo raccolto da diverse fonti i dati sulla diffusione di alcuni prodotti OSS di maggiore successo e li abbiamo confrontati con i prodotti proprietari loro concorrenti. La tabella 2 fornisce un quadro generale. Per un'analisi più ampia si rinvia a Giuri, Rocchetti e Torrisi (2002). E' importante notare che i dati sulla diffusione di OSS sono molto difficili da reperire a causa delle peculiarità del sistema distributivo (compreso il fatto che il software installato su un computer può essere copiato e installato su altri computer).

Tabella 2 Diffusione di alcuni prodotti OSS

Sistemi operativi per server (<i>nuove installazioni</i>)	Sistemi operativi (computer che gestiscono siti web) (<i>base installata</i>)	Web server software (siti attivi) (<i>base installata</i>)	E-mail server (<i>base installata</i>)
(2002)	(2001)	(2002)	(2001)
Windows (59,9%)	Windows (a)(49,6%)	Apache (64,4%)	Sendmail (42%)
Unix (14,6%)	Linux (29,6%)	Microsoft (b) (27,2%)	Microsoft Exchange (18%)
Linux (13,7%)	Solaris (7,1%)	Zeus (1,1%)	Unix qmail (17%)
Netware (9,7%)	BSD (6,1%)	Netscape (1,7%)	Windows Ipswitch (6%)
Altro (2,1%)	Altri Unix (2,2%)	Altri (5,6%)	Unix Smap, Postfix, Exim (5%)
	Non-Unix (2,4%)		Altro (12%)
	ND (3,0%)		
N=4274372	N=n.d.	N= 16323607	N= 958
Fonte: IDC (2002)	Fonte: Netcraft (2001)	Fonte: Netcraft (2002a)	Fonte: Wheeler (2002)

(a) Windows 2000, NT4, NT3, Windows 95 e 98; (b) Microsoft IIS, Microsoft IIS-W, Microsoft PWS-95

Secondo stime IDC, nel mercato dei sistemi operativi per server, Windows risulta il leader del mercato con una quota di mercato del 59,9% (cfr. tab. 2). La quota di Unix è pari al 14,6% mentre quella di Linux è pari al 13,7% (IDC, 2002). Tali stime sono confermate anche da altri studi. Infatti, secondo un'indagine condotta nel 2002 su 395 utilizzatori di OSS in Germania, Svezia e Regno Unito la percentuale di sistemi operativi OSS (compreso Linux) installati è pari al 15,7% (Gosh et al., 2002).

Un'altra indagine commissionata da Suse Linux srl mostra che in un campione di 414 PMI industriali italiane oltre il 66,9% delle imprese utilizza Windows come server mentre solo il 3,4% utilizza Linux. La quota di Windows supera l'83% nel caso dei sistemi operativi client (Congenio, 2002).¹

Come mostra la Tabella 2, nel mercato del *Web Server Software* un prodotto OSS, Apache, ha una netta posizione di leadership (Netcraft, 2002a). Apache è entrato nel mercato nel 1995 guadagnando rapidamente la posizione di leadership precedentemente coperta dal software del National Center for Supercomputing Applications (NCSA) della Università dell'Illinois. Microsoft è entrata subito dopo con Windows IIS (Internet Information Server), diventando il *second player* nel 1998. Dopo un ingresso tempestivo ed una certa crescita iniziale, Netscape ha progressivamente perso quote di mercato mentre Zeus, entrato nel 1997, ha lentamente raggiunto la quarta posizione sui siti attivi e recentemente la terza posizione su tutti i siti. Apache è il leader in Europa, Russia, Canada e India, mentre Microsoft negli Stati Uniti ed in Cina. In Italia Apache e Microsoft hanno quote di mercato molto simili.

Anche nel mercato degli e-mail server troviamo un prodotto OSS, Sendmail, che detiene una posizione di leadership, seguito da Microsoft Exchange (cfr. tab. 2).

I segmenti di mercato esaminati sono tutti caratterizzati da prodotti per ambiente server che vengono utilizzati da utenti con elevate competenze tecniche. Inoltre, in Italia, gli utilizzatori sono ancora imprese di grandi dimensioni e gestori di servizi Internet (internet service providers e application service providers) (Netcraft, 2002b). La diffusione futura di OSS nel mercato degli utenti finali (client) è ancora molto limitata. Il futuro dell'OSS in questo segmento di mercato è legato all'evoluzione delle interfacce grafiche, come KDE (K desktop environment), un prodotto distribuito secondo la licenza GPL (mentre la libreria di strumenti di sviluppo è distribuita in LGPL), e allo sviluppo del software applicativo, come KOffice (un prodotto complementare delle interfacce KDE) e StarOffice/Open Office.

5. Open software e strategie commerciali

La comunità degli sviluppatori di OSS è una realtà molto eterogenea, con differenze marcate tra diverse categorie di partecipanti in termini di motivazioni a rivelare il codice sorgente. La maggior parte degli studi sul fenomeno dell'OSS enfatizzano gli incentivi di una particolare categoria di partecipanti rappresentata da alcuni leader dotati di grandi capacità tecnologiche e carisma, come Eric Allman, Bruce Perens e Linus Torvald. Questa categoria di contributori è spinta da un insieme di motivazioni tipiche delle comunità scientifiche e della cosiddetta 'gift economy' – altruismo, priorità, reputazione e 'consumo' (Lerner e Tirole, 2000; Raymond, 2000). Ma molti individui che partecipano a progetti OSS (compresi alcuni tra quelli appena citati) sono mossi anche da motivazioni economiche ben precise, quali la necessità di risolvere problemi tecnici specifici, la ricerca di un reddito immediato e la ricerca di un reddito futuro (attraverso la segnalazione delle proprie capacità nello sviluppo di software) (Lakhani e von Hippel, 2000; Lerner e Tirole, 2000). Infatti, diversi esperti hanno fatto leva sulla reputazione acquisita all'interno della comunità OSS per fondare nuove imprese di successo. Per esempio, Eric Allman ha fondato Sendmail, Inc. (servizi di supporto), William Joy ha creato Sun Microsystems (workstation e software) e Paul Vixie ha dato vita a Vixie Enterprises (servizi di engineering e consulenza). Altri, hanno trovato occupazione in imprese affermate del settore, come Brian Behlendorf (Collab.Net) e Larry Wall (O'Reilly & Associates) (Lerner e Tirole, 2000).

Gli incentivi delle imprese che partecipano allo sviluppo di OSS sono a loro volta differenti da quelli degli sviluppatori individuali. Numerose imprese sono entrate nel mercato a partire dagli anni '90 per assemblare diversi moduli di OSS, per sviluppare interfacce che rendono i programmi più 'amichevoli' o prodotti complementari come driver ed altre applicazioni, o per fornire supporto tecnico e servizi (tra

¹ Ricordiamo che questi dati si riferiscono alla base installata mentre i dati IDC sui sistemi operativi del 2002 si riferiscono alle nuove installazioni effettuate nell'arco di quell'anno.

queste Caldera, RedHat e Suse). Alcune di queste imprese hanno fondato i loro modelli di business sulla distribuzione di software molto popolari come Linux o Apache. In altri casi il modello di business è stato centrato su nuovo software aperto sviluppato dai fondatori dell'impresa (per esempio Sendmail). Alcune grandi imprese commerciali hanno adottato un approccio favorevole all'OSS, rivelando il codice sorgente di alcuni loro prodotti oppure contribuendo allo sviluppo di nuovi prodotti OSS. Questa strategia è perseguita da imprese che non possiedono uno standard proprietario che domina il mercato, quindi imprese che non hanno problemi di 'cannibalizzazione' dei propri prodotti, anzi cercano di utilizzare l'OSS per aumentare le vendite di prodotti proprietari complementari (hardware o software). Questo è chiaramente il caso di IBM, che sostiene Linux per ridurre la propria dipendenza dallo standard Windows di Microsoft, il caso di Sun, che ha aperto la propria suite di prodotti per l'ufficio (StarOffice/Open Office) per renderla più competitiva rispetto a Windows e ai prodotti di altri concorrenti (Corel e Lotus Smart Suite), e di SAP, che ha reso aperto il proprio database SAP DB perché molti utilizzatori del suo ERP usano database concorrenti (es. Oracle, IBM o Microsoft).

D'altra parte, Microsoft, che sicuramente ha molto da perdere dalla concorrenza con Linux e OpenOffice per via della sua posizione di leadership nel mercato del software per PC, ha adottato una strategia di reazione aggressiva. Nel 2001 Microsoft ha permesso ad alcune categorie di clienti di visionare il codice sorgente di alcuni prodotti lanciando la cosiddetta "Shared Software Initiative (SSI)". La SSI risponde parzialmente ad una critica di fondo mossa da più parti, che è quella di limitare l'accesso alle proprie interfacce di programmazione (API) ai produttori di software applicativo complementari o concorrenti, e di rinunciare alle opportunità di controllo della qualità/sicurezza offerte dalla visione del codice sorgente da parte dei clienti. Nell'ambito di questa iniziativa, Microsoft ha consentito a oltre 35 mila programmatori di accedere al codice sorgente della CLI (Common Language Interface) dell'ambiente di sviluppo di servizi web .Net (Judge, 2002). La SSI si differenzia dall'OSS perché non accetta il principio del copyleft - l'utente Microsoft non può modificare né redistribuire il codice ispezionato.

6. Prospettive future

L'evoluzione futura dell'OSS appare piuttosto incerta. Un elemento di incertezza riguarda la sostenibilità economica dei modelli di business legati all'OSS, specialmente quelli della distribuzione di Linux e Apache, centrati sullo spostamento della fonte dei ricavi dalle licenze (prodotti) ai servizi. Un altro aspetto critico riguarda gli effetti della coesistenza fra software proprietario e software aperto sul futuro avanzamento tecnologico. Come abbiamo visto, diverse imprese affermate hanno adottato una strategia che punta sulla convivenza con l'OSS. A favore della sostenibilità di uno scenario di convivenza c'è da considerare il fatto che l'OSS produce esternalità positive per i produttori di software proprietario dovute semplicemente al fatto di mettere in circolazione nuovi prodotti e nuove idee. Ma la comunità OSS produce anche esternalità negative nei confronti dei produttori di software proprietario dovute all'assenza di prezzi di licenza. I bassi prezzi di acquisto di prodotti OSS riducono la domanda di prodotti proprietari sostituiti e quindi riducono ex ante gli incentivi all'innovazione delle imprese di software proprietario. Poiché molti prodotti OSS oggi sul mercato sono nati ispirandosi a prodotti proprietari affermati (es. Linux e OpenOffice), la mancanza di incentivi all'innovazione dei produttori di software proprietario potrebbe avere conseguenze negative sulla crescita di lungo periodo (Saint-Paul, 2001).

D'altra parte, i sostenitori dell'OSS affermano che il progresso tecnologico in questa industria consiste in una sequenza di innovazioni incrementali interdipendenti che nel loro insieme rappresentano il frutto di un'azione inventiva collettiva. In generale è vero che molti nuovi prodotti sul mercato apportino un contributo originale limitato rispetto al bagaglio di conoscenze precedenti a cui attingono. Quindi siamo in presenza di un processo di sviluppo tecnologico incrementale e sequenziale. Inoltre, rispetto ad altre discipline come la chimica o la fisica dello stato solido, nel caso del software le conoscenze tecnologiche sono spesso documentate in modo molto approssimativo.

La sequenzialità e la scarsa documentazione disponibile riducono la brevettabilità del software, perché rendono difficile separare il contributo originale di una singola invenzione dallo 'stato dell'arte',

e spiegano la maggiore popolarità del copyright. Queste peculiarità della tecnologia software però non richiedono necessariamente l'abbandono del sistema di protezione legale tradizionale e l'adesione tout court al sistema del copyleft. Una via alternativa consiste nel ridurre l'ampiezza della protezione legale, per esempio permettendo un certo grado di *reverse engineering* del software brevettato (cfr. Cohen e Lemley, 2001).

Bibliografia

- Cohen J. E., Lemley M.A. (2001), "Patent Scope and Innovation in the Software Industry", <http://www.georgetown.edu>.
- Congenio (2001) "Indagine circa la propensione delle PMI industriali a migrare verso sistemi operativi alternative: il caso Linux", Congenio Srl, ricerca commissionata da SuSe Linux srl (http://linfe.it:9451/Aicalab/Aicadocs/doceuments/1035740863.5/doc_index.html).
- Cübranić D., Booth K. (1999), "Coordinating Open Source Software Development", *IEEE Proceedings*, 8th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, (WET ICE '99), pp. 61–66.
- Fuggetta A. (2001), "Open Source Software: an evaluation", Politecnico di Milano, mimeo.
- Giuri, P., Rocchetti, G. e Torrisi, S. (2002), "Open Source Software: From Open Science to New Marketing Models", *LEM working paper 2002/23*, Scuola Superiore Sant'Anna, Pisa, (<http://sssup1.sssup.it/~LEM/wplem.html>).
- Gosh R., A. Krieger, B., Glott, R. and Robles, G., (2002) *Free/Libre and Open Source Software: Survey and Study*, Infonomics, NL, June (<http://www.infonomics.nl/FLOSS/report>).
- Hecker F. (2000), "Setting Up Shop: The Business of Open-Source Software", <http://www.hecker.org/writings/setting-up-shop.html>.
- IDC (2000), *Server Operating Environment: 2000 Year in Review*, Bulletin#23731.
- Judge, P. (2002) "Open Source, Shared Source or Secret Sauce", ZDNet News, 23 settembre (<http://zdnet.com>).
- Lakhani K., von Hippel E. (2000), "How open Source software works: 'Free' user-to-user assistance", MIT Sloan school of Management WP No. 4117.
- Lerner J., Tirole J. (2000), "The Simple Economic of Open Source", NBER Working Paper n. 7600 (<http://www.nber.org/papers/w7600>).
- Merges, R. (1996) "A Comparative Look at Property Rights and the Software Industry", in D. Mowery (a cura di) *The International Computer Software Industry. A Comparative Study of Industry Evolution and Structure*, Oxford University Press, Oxford, pp. 272-303.
- Netcraft (2002a), *Web Server Survey*, <http://www.netcraft.com/survey/>.
- Netcraft (2002b), *Web Server Operating Systems in Italy*, Netcraft per Microsoft.
- Netcraft (2001), *Web Server Survey*, <http://www.netcraft.com/Survey/computers2001>.
- Prusa T.J., Schmitz J.A. (1991), "Are New Firms an Important Source of Innovation?", *Economic Letters*, 35; 339.
- Raymond E. S. (2000), "Homesteading the Noosphere", <http://tuxedo.org/~esr/writings/homesteading/homesteading/>.
- Raymond, E. S. (2001), *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly, www.o.reilly.com.
- Saint-Paul G. (2001), "Growth effects of non-proprietary innovation", CEPR Discussion Paper Series No. 3069, <http://cepr.org/pubs/dps/DP3096.asp>.
- Sirmi (2002), *Indagine Linux*, rapporto di ricerca per Microsoft, gennaio.
- Wheeler D. (2002), "Why Open Source Software/Free Software (OSS/FS)? Look at the Numbers!", http://www.dwheeler.com/oss_fs_why.html.