

# **Codice, *Open source* e Diritti di Proprietà: Innovazione, Coordinamento e Diffusione**

---

Andrea Pontiggia\*, Francesco Bolici<sup>1</sup>

\*Università degli Studi di Cassino  
Dipartimento Impresa e Lavoro  
[andrea.pontiggia@eco.unicas.it](mailto:andrea.pontiggia@eco.unicas.it), [francesco.bolici@eco.unicas.it](mailto:francesco.bolici@eco.unicas.it)

<sup>1</sup>Università Luiss Guido Carli  
Centro di Ricerca sui Sistemi Informativi  
[fbolici@luiss.it](mailto:fbolici@luiss.it)

## 1. Introduzione

L'analisi si sofferma sull'esame e sul confronto delle diverse licenze *open source*. Questa indagine è la prima fase di un più articolato programma di ricerca.

Il termine *Open source* (OS) indica che i diritti di proprietà intellettuale sul codice sorgente di un software sono di pubblico dominio. Il codice può quindi essere utilizzato e modificato senza che venga richiesto alcun pagamento, come avviene invece nel caso dell'acquisto di una licenza. A questa caratteristica essenziale di proprietà pubblica dei diritti di proprietà intellettuale si è aggiunto nel tempo un secondo aspetto distintivo del fenomeno OS: un modello di produzione del software basato su una comunità di programmatori distribuita e geograficamente dispersa (Kogut e Metiu, 2001).

Il programma di ricerca, di cui la presente analisi costituisce una prima fase, distingue tre momenti nel processo di sviluppo di un software *Open source*: i) un individuo con un'idea (più o meno) innovativa decide di renderla pubblica chiedendo l'aiuto di altre persone interessate a svilupparla, costituendo così un gruppo di volontari; ii) si forma spontaneamente una comunità di programmatori, geograficamente distanti e spesso sconosciuti, che iniziano a collaborare per la realizzazione del progetto; iii) il software si diffonde (o meno) tra gli utilizzatori.

A ciascuna fase si possono associare delle domande di ricerca:

1. Quali sono i fattori motivanti per l'individuo con un'idea progettuale innovativa che possano spingerlo a renderla pubblica? Perché i programmatori scrivono in maniera volontaria linee di codice, impiegando il proprio tempo e le proprie risorse, senza che ci sia un sistema premiante per il loro lavoro? Quali sono le caratteristiche di questo modello di innovazione che motiva la partecipazione dei programmatori? Quali sono le opzioni di distribuzione tra cui l'innovatore può scegliere e quali sono le differenze significative tra questi meccanismi?
2. Può un elevato numero di programmatori (distribuiti e dispersi) collaborare in assenza di una tradizionale struttura gerarchia formalizzata? Quali meccanismi permettono di coordinare il lavoro di persone impegnate nella realizzazione di progetti spesso molto complessi in assenza di un controllo formale esercitato dalla tradizionale organizzazione fondata sulla proprietà? Quali sono le fondamenta di questi comportamenti di cooperazione e di reciprocità collettiva in una comunità di quasi sconosciuti?
3. Perché e come si diffondono i software *Open source* tra gli utilizzatori? Esistono fattori che giustifichino la loro esistenza? In che misura la presenza di standard proprietari impedisce o accelera la diffusione? Come si sviluppa una eventuale competizione tra licenze, tra livelli di proprietà differenti?

La prima fase del progetto di ricerca copre le domande del punto 1 e si concentra sulle motivazioni individuali e sul sistema di incentivi che spingono prima l'innovatore e quindi gli altri programmatori a contribuire al progetto di sviluppo. La

---

linea di ricerca si concentra sul *modello di innovazione aperto* (modello *Open source*) con particolare attenzione al sistema e contenuto delle licenze.

La seconda area di ricerca ha per oggetto i meccanismi di *coordinamento* nelle comunità *Open source*, realtà in cui gli obiettivi individuali devono essere allineati, gli incentivi definiti ed assegnati, il progetto scomposto in attività base ed i compiti distribuiti senza ricorrere alle tradizionali soluzioni basate su diritti di proprietà, contratti di subordinazione e strutture gerarchiche.

Il terzo punto riguarda i processi di *diffusione* dei prodotti *Open source*. Si mira a comprendere quali siano le condizioni per la diffusione di un software *Open source* soprattutto in casi in cui sia già affermato nel mercato uno standard proprietario che dovrebbe permettere dei rendimenti di adozione crescenti.

L'analisi empirica basata sui progetti della comunità SourceForge.net si concentra dapprima su differenti tipi di licenze OS indagando la frequenza e il ricorso a queste forme di protezione dei diritti di proprietà (Allegato 1). Dal confronto con i risultati dell'analisi dei dati relativi ai progetti OS si rilevano delle differenze nel contenuto che sembrano suggerire una modularità delle forme di protezione e di difesa dei modelli di business. Tale analisi è stata condotta a partire da un dataset pubblico relativo ai progetti *open source* integrato con dati specifici sulle modalità di sviluppo.

## 2. Caratteristiche del Modello di Innovazione Aperto

I processi di innovazione sono qui analizzati legando le motivazioni individuali e collettive alle diverse licenze oggi disponibili (Allegato 1); in particolare le licenze definiscono alcune delle modalità di distribuzione e la relazione tra incentivi, ricompense e modalità di distribuzione stesse (Bonaccorsi e Rossi 2002; Giuri et al. 2002).

### 2.1 Comunità e organizzazione del lavoro

Il modello di innovazione relativo ai progetti *open source* si fonda su un concetto alternativo rispetto al tradizionale processo di sviluppo software proprio delle aziende private. Il software OS è sviluppato all'interno di un sistema di volontari incuranti dei confini organizzativi e geografici, scarsamente coordinati tra loro e che collaborano via Internet per produrre un prodotto innovativo e di alta qualità (Lee e Cole, 2003).

Rispetto ai classici modelli di innovazione basati o sul modello di *investimento privato* --costruito sull'idea che l'innovazione sia supportata dalle risorse dei privati che si aspettano quindi un ritorno economico dal suo sfruttamento-- o su quello di *azione collettiva*--che si impernia sul concetto di bene pubblico caratterizzato dalla non escludibilità e non rivalità--, il fenomeno OS apre un nuovo scenario (von Hippel e von Krogh, 2003). Infatti, il modello di sviluppo basato sulla comunità *open source* risulta alternativo rispetto ai tradizionali processi commerciali sotto diversi aspetti (Fulk et al. 2004). I diritti di proprietà intellettuale sul codice sono pubblici, --anche se possono essere sfruttati dagli sviluppatori che tuttavia devono rendere disponibile anche il codice sorgente-- e non vengono detenuti nella loro totalità da un'azienda o

da un singolo. La partecipazione allo sviluppo del software è libera e non ristretta ai soli dipendenti selezionati dall'azienda per svolgere la specifica attività di programmazione. L'attività degli sviluppatori non è pagata direttamente da nessuna azienda essendo i programmatori dei volontari. La distribuzione della conoscenza generata dal processo di sviluppo software non incontra confini aziendali che ne limitino il trasferimento (Gatcher e Fehr, 1999). Infine le comunità OS hanno come canale di comunicazione privilegiato quello basato su Internet o altri mezzi tecnologici, rinunciando quasi per la totalità a rapporti faccia a faccia (Lee e Cole, 2003 cfr. Tabella 1).

GWENDOLYN K. LEE AND ROBERT E. COLE *From a Firm-Based to a Community-Based Model of Knowledge Creation*

**Table 1** The Community-Based Model vs. the Firm-Based Model of Knowledge Creation

Organizing Principles	The Firm-Based Model	The Community-Based Model
(1) Intellectual Property Ownership	Knowledge is private and owned by the firm.	Knowledge is public but can be owned by members who contribute it as long as they share it.
(2) Membership Restriction	Membership is based on selection, so the size of the firm is constrained by the number of employees hired.	Membership is open, so the scale of the community is not constrained.
(3) Authority and Incentives	Members of the firm are employees who receive salaries in exchange for their work.	Members of the community are volunteers who do not receive salaries in exchange for their work.
(4) Knowledge Distribution Across Organizational and Geographical Boundaries	Distribution is limited by the boundary of the firm.	Distribution extends beyond the boundary of the firm.
(5) Dominant Mode of Communications	Face-to-face interaction is the dominant mode of communication.	Technology-mediated interaction is the dominant mode of communication.

**Tabella 1** Confronto tra il Modello basato sulla Comunità e quello basato sulle aziende tradizionali (Lee e Cole, 2003).

## 2.2 Motivazione individuale e collettiva

Considerando il processo di sviluppo del software *open source*, basato sull'appartenenza ad una comunità di sviluppatori volontari, un primo interrogativo che ci si può porre è se questo modello sia sostenibile nel tempo e quindi se gli sviluppatori continueranno a dare il proprio contributo al "bene pubblico". La questione posta dalle licenze *open source* può quindi essere sintetizzata nella domanda se ci siano sufficienti incentivi per gli sviluppatori a contribuire all'innovazione (Kogut e Metiu, 2001).

Una prima risposta è che gli sviluppatori contribuiscano per un senso di altruismo e di disinteressata generosità. Diverse ricerche sullo studio del comportamento economico evidenziano il fenomeno per cui le persone non si basano solo su puri calcoli economici nelle loro decisioni, anzi spesso questi elementi non vengono presi in considerazione se non in un secondo momento. Ad esempio, gli esperimenti condotti da Kahneman et al. (1986) e Bies (1993) sottolineano il ruolo della correttezza, sulla base della quale le persone distribuiscono una ricompensa fissa. Le persone inoltre sem-

bra si comportino in maniera più corretta di quanto sia prevedibile attraverso modelli fondati su prospettive come quelle alla base del dilemma del prigioniero. Inoltre, la non correttezza del comportamento diminuisce drasticamente in presenza della comunicazione o di semplici sistemi di screening (Ostrom et al., 1992; Frank, 1998). L'importanza della reciprocità negli scambi tra i membri di una comunità *open source* è stata recentemente documentata da Lakhani e von Hippel (2000). Il loro studio sui gruppi di supporto per i web server Apache mostra che la ragione più importante che spinge le persone a condividere le proprie risposte all'interno dei gruppi Usenet è il desiderio di aiutare gli altri perché loro stessi sono stati in precedenza aiutati, oppure perché si aspettano di essere aiutati in futuro. Secondo questa prospettiva, il fenomeno *open source* viene spesso accostato alla prospettiva dell'"economia del dono" (Fehr e Schmidt, 1999). La gift economy non è semplicemente una generica forma di "altruismo", bensì un modello differente attraverso cui perseguire i propri interessi. Inoltre, le esternalità positive degli scambi di doni possono incrementarsi reciprocamente e quindi tendere ad espandersi.

Lerner e Tirole (2000) propongono una spiegazione del modello di innovazione aperto che non si basa solamente sui concetti di altruismo o di identificazione in una comunità. Nel loro studio, sostengono che il contribuire ad un progetto *open source* abbia forti somiglianze con una competizione aperta in cui vengano segnalati i meriti dei diversi sviluppatori. Attraverso queste "segnalazioni" dei propri meriti, il singolo sviluppatore/contributore può ottenere un beneficio diretto per quanto riguarda la propria reputazione e posizione all'interno del mercato del lavoro.

Inoltre, l'attiva partecipazione ad un progetto *open source* permette di conseguire altri benefici, come sottolinea Raymond (1998, 1999): 1) una buona reputazione tra i pari; 2) maggiore attenzione e disponibilità alla cooperazione da parte degli altri soggetti; 3) uno status più elevato nell'economia degli scambi.

Possiamo quindi sintetizzare due possibili prospettive che spiegano la contribuzione di un individuo ad un progetto *open source*: l'una basata sul concetto del dono e l'altra della segnalazione dei meriti sul mercato del lavoro. Interessante notare che queste due spiegazioni possono essere associate a due diverse prospettive sulla motivazione individuale (Frey, 1997). Il concetto di "dono" riflette una motivazione di tipo intrinseco, dove l'individuo trova la sua ricompensa nel riconoscimento pubblico del proprio contributo. La prospettiva di segnalazione sul mercato del lavoro corrisponde ad una motivazione di tipo estrinseco che lega il contributo del singolo ad un ritorno di tipo economico. Entrambi le motivazioni possono operare in un gruppo, anche se sembra che le comunità basate su entrambi questi fattori spesso si dissolvano a causa del moltiplicarsi dei comportamenti opportunistici (*free-rider*).

L'importanza del distinguere tra queste due motivazioni è cruciale per i classici studi dell'economia del dono, all'interno dei quali Timuss (1971) scoprì che la ricompensa estrinseca di compensare per le donazioni di sangue faceva crescere l'offerta, ma aveva anche l'opposto effetto di diminuire la ricompensa intrinseca dei donatori volontari. Come risultato e conseguenza dell'analisi si otteneva la potenziale distruzione del mercato del sangue attraverso l'aumento dell'incertezza rispetto alla qualità (poiché pagando per la donazione si aumentava il numero di persone bisogno-

se che pur di ricevere la ricompensa economica procedevano alla donazione in qualsiasi condizione fisica si trovassero).

Si può affermare che la produzione tra pari si fonda sulla raccolta e sullo scambio decentralizzato di informazioni anche allo scopo di ridurre l'incertezza legata ai partecipanti. Questo offre il vantaggio, come processo informativo per l'identificazione e l'allocatione della creatività disponibile, di lavorare su risorse informative, culturali e cognitive. Il modello di produzione tra pari di una comunità è largamente dipendente da un'ampia aggregazione di individui che setacciano l'ambiente in maniera indipendente in cerca di opportunità per manifestare la loro creatività in misura più o meno ampia. Questi individui quindi si auto-selezionano per alcune attività e vi contribuiscono per una varietà di motivazioni.

Il problema motivazionale può essere affrontato attraverso due distinte conclusioni la prima è che a certe condizioni, un progetto di sviluppo può offrire delle ricompense sociali o psicologiche che possono attrarre alcuni sviluppatori, a cui una promessa di ricompense solo economiche non sarebbe stata sufficiente. Il secondo punto è che quando un progetto di qualsiasi dimensione viene diviso in piccole parti, --ognuna delle quali richiede un limitato impiego di tempo da parte dell'individuo per essere completata--, la quantità di motivazione che il singolo deve avere per contribuirvi è più bassa. I progetti devono essere divisibili in componenti, ciascuna delle quali si deve poter produrre indipendentemente dalle altre (moduli). Questo permette al processo di produzione di essere incrementale ed asincrono, migliorando gli sforzi di persone differenti, con distinte capacità e che sono disponibili in orari diversi. La granularità dei moduli è importante e si riferisce alla dimensione dei moduli del progetto che devono essere piuttosto fini o di piccole dimensioni. Ciò consente al progetto di catturare i contributi di un largo numero di persone il cui livello di motivazione non potrebbe sostenere nient'altro che piccoli sforzi di contribuzione al progetto.

### **3. Modelli di Innovazione e prospettiva economica sui diritti di proprietà intellettuale**

Nella prospettiva economica, le logiche *open source* interessano perché le persone cooperano mostrando un impegno per un periodo non breve nonostante l'assenza sia di una retribuzione, --o compenso diretto-- sia di diritti di proprietà sulla propria opera. Il contributo personale sebbene sia visibile, misurabile e in alcuni casi facilmente distinguibile da quello degli altri, non è collegato alcuna forma di compenso.

La domanda di software è fortemente influenzata dalla sua diffusione per la presenza di consistenti effetti rete e di esternalità. L'utilità per un utente di un certo software frequentemente aumenta al crescere del numero degli utilizzatori. La presenza di forti esternalità potenzialmente consentirebbe ad un'impresa in una posizione dominante nel mercato di costruire (o almeno tentare) un duraturo vantaggio competitivo impedendo o regolando indirettamente l'ingresso di nuovi concorrenti. La sostenibilità di questo modello è anche funzione dell'efficacia con cui i diritti di proprietà intellettuale sono sostenuti e difesi (Kogut e Metiu, 2001).

Un aspetto interessante da analizzare è la relazione tra la logica *open source* e gli studi dedicati all'innovazione tecnologica. Il processo di sviluppo *open source* è simile per molti aspetti al tipo di "innovazione user-driven" presente in molti altri settori. Tra gli altri, gli studi di Rosenberg (1982) sul settore degli strumenti meccanici e di von Hippel (1988) su quello degli strumenti scientifici hanno contribuito a evidenziare il ruolo che alcuni utenti possono giocare nell'accelerare e contribuire ai progressi tecnologici. In diversi casi le soluzioni sviluppate per le particolari esigenze di pochi utenti sono divenute nel tempo prodotti validi per la maggior parte degli utilizzatori. Tuttavia alcuni aspetti della tematica *open source* --si pensi ai meccanismi di motivazione o a quelli di coordinamento del lavoro-- si distinguono nettamente dai problemi affrontati negli altri settori.

La diffusione delle logiche *open source* ha quindi posto nuova attenzione su un argomento ancora dibattuto: quali sono i vantaggi sociali e i relativi costi dell'assegnare delle patenti sulle invenzioni? Mazzoleni e Nelson (1998) ci offrono una efficace classificazione che a tale proposito individuano quattro teorie: 1. la teoria della motivazione all'innovazione; 2. della rivelazione /disseminazione dell'innovazione; 3. dello sviluppo e della commercializzazione; 4. del controllo delle modalità di sviluppo ed esplorazione.

La più diffusa teoria che tenta di spiegare questo fenomeno (*invention-motivation theory*) è quella per cui la concessione di patenti incrementerebbe lo sviluppo di innovazioni utili e d'altro canto il costo consisterebbe nella restrizione all'accesso all'invenzione completa. Una possibile implicazione di questa teoria è che le patenti dovrebbero essere evitate in contesti in cui le innovazioni verrebbero ad essere prodotte anche in assenza di stringenti forme di protezione e diritti di sfruttamento dell'innovazione.

La seconda teoria (*invention-dissemination*) sostiene che le patenti spingono gli innovatori a rivelare le loro invenzioni che altrimenti tenterebbero di nascondere. Questo faciliterebbe la diffusione della conoscenza sull'uso dell'innovazione nella società.

La terza teoria (*induce-commercialization*) considera le patenti come utili al fine di indurre gli investimenti necessari per lo sviluppo e la commercializzazione dell'innovazione.

L'ultima teoria (*exploration-control*) riguarda non tanto la motivazione all'innovazione, quanto le patenti come un sistema efficiente per allocare le risorse per lo sviluppo delle innovazioni esistenti. Possedere la patente relativa ad un'innovazione garantirebbe al proprietario la possibilità di coordinare gli sforzi e gli investimenti per la ricerca di futuri sviluppi di quella innovazione, prevenendo possibili duplicazioni di sforzi e facilitando il flusso delle informazioni.

L'economia e la tradizione giuridica sui diritti di proprietà sostengono che le patenti siano la soluzione al problema della "tragedy of commons". In un celebre articolo Hardin (1968) conclude infatti che i beni pubblici sono soggetti ad essere sovrautilizzati quando troppi "proprietari" hanno il diritto di utilizzarli e nessuno ha il diritto di escludere un altro. Questo contributo ha anche spinto le politiche di privatizzazione

della proprietà comune attraverso accordi privati (Ostrom, 1990) o la concessione di patenti per scoperte scientifiche.

Recenti contributi, soprattutto provenienti dagli studi giuridici, hanno messo in discussione queste conclusioni. Heller e Eisenberg (1998) hanno infatti obiettato che i beni pubblici sono inclini ad essere sotto-utilizzati in quella che hanno chiamato “la tragedia delle cose non pubbliche” quando troppi individui detengono diritti di esclusione su una risorsa scarsa. L’esempio più classico riguarda la frammentazione dei diritti della ricerca bio-medica negli USA dove la necessità di avere un accesso multiplo a differenti input tutti con relative patenti può impedire un utente dallo sviluppare prodotti utili.

In anni recenti l’attenzione si è molto concentrata sui costi dei regimi di protezione della proprietà al fine di garantire una “ricompensa” per l’innovazione. Le implicazioni a cui si è arrivati, sotto certe condizioni e in specifici casi, è che una politica basata sulla mancanza di protezione può dominare un sistema che permetta di ottenere dei monopoli (diritti esclusivi) attraverso l’esercizio delle licenze e dei brevetti.

D’altra parte alcuni ricercatori negano che ci siano differenze sostanziali nel settore dello sviluppo software tra i metodi e gli strumenti open source e quelli del settore tradizionale commerciale. I progetti e i software *open source* non sembrano produrre una codice di qualità migliore per quanto riguarda la modularità o i *bug*. Si riscontra un’ampia variabilità nella qualità del codice così come nella documentazione o nelle fasi di controllo e di verifica. Inoltre, molti ricercatori non trovano giustificata l’idea resa popolare da Raymond secondo cui l’*open source* produca software di qualità migliore perché così tanti “occhi” possono accedere al codice sorgente. I dati indicano che, sebbene il numero di progetti *open source* attivi nel mondo sia molto elevato, la maggior parte dei progetti consiste di soli pochi sviluppatori e probabilmente anche di una piccola quantità di utilizzatori.

I nostri dati sembrano confermare che un basso numero di sviluppatori scrive la maggior parte del codice sorgente in circolazione sulla rete. Il fenomeno *open source*, sostiene Cusumano (2004), nasce perché un certo numero di programmatori molto capaci, alla cui testa troviamo Stallman e Torvalds, erano frustrati dal non poter migliorare le caratteristiche del software in commercio protetto. Non nasce per esigenze relative ad una migliore qualità del software o un interesse diffuso di tutti gli utenti a modificare il codice sorgente dei propri programmi.

Diviene centrale analizzare il modello di innovazione *open source* alla luce delle caratteristiche relative ai meccanismi di distribuzione che le licenze possono prescrivere. Le caratteristiche relative ai tipi di licenza individuati dagli innovatori, intorno a cui si aggrega la comunità degli sviluppatori *open source*, potrebbero dare un contributo significativo ad una più approfondita comprensione di questo fenomeno.

#### **4. Analisi condotta**

L’analisi è condotta a partire dai dati messi a disposizione della comunità di SourceForge.net (allegato 2) composta da più di 100.000 progetti di sviluppo *open source* attivi.



Dopo aver ri-classificato le diverse tipologie di licenze secondo dimensioni organizzative, si è calcolata la loro distribuzione e la loro rilevanza all'interno dei progetti di sviluppo di SourceForge.net.

L'analisi condotta permette di confrontare la diffusione dei diversi tipi di licenze in un arco temporale di tre anni (i dati sono relativi a due momenti: Giugno 2002 e Luglio 2005), evidenziando la creazione di nuove licenze, la loro evoluzione e i cambiamenti nel loro utilizzo e diffusione. Attraverso questi dati si procede ad un ulteriore approfondimento del rapporto tra tipologia di licenza scelta dagli sviluppatori e modello di innovazione e di business del progetto.

#### 4.1 Le modalità di distribuzione

Le licenze *open source* sono dei meccanismi di distribuzione del software che, pur potendo essere molto diverse l'una dall'altra, hanno delle caratteristiche comuni (*Open Source Definition* v.1.9, cfr. Allegato 3).

L'importanza delle licenze come meccanismo distributivo dell'innovazione ha spinto a focalizzare l'analisi sui diversi tipi di licenze *open source* esistenti e sulle loro caratteristiche. Lo studio si è incentrato sull'esame delle 58 licenze *open source* riconosciute ufficialmente alla data di Luglio 2005 dall'OSI (*Open Source Initiative*) 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.

Il passo successivo dell'analisi è stata una riclassificazione delle dimensioni significative delle licenze e la loro successiva classificazione.

Per quanto riguarda le dimensioni d'analisi, abbiamo innanzi tutto verificato la presenza esplicita di clausole che impongano la disponibilità del codice sorgente. Ulteriori dimensioni analizzate sono quelle riferite all'esplicita presenza dei diritti di utilizzo, modifica e distribuzione del prodotto coperto da licenza e loro condizioni d'impiego. Abbiamo inoltre provveduto a verificare la presenza di clausole di protezione dei diritti morali degli sviluppatori e di eventuali diritti "speciali" riservati agli sviluppatori iniziali. Quindi si è accertata la possibilità (o l'eventuale obbligo) di distribuire il prodotto in forma gratuita, oppure le condizioni sotto le quali si possa esigere un pagamento. Altra dimensione d'analisi adottata è stata la presenza o meno di clausole di garanzia del prodotto o di responsabilità dello sviluppatore relativamente al prodotto distribuito. Infine si è verificata la possibilità di combinare il prodotto coperto da licenza con altri software, sia di tipo *open source* che di tipo privato tradizionale. Data l'importanza della licenza GPL, che descriveremo più avanti, abbiamo anche esaminato la possibilità di combinare la licenza esaminata con un altro prodotto coperto dalla GPL. A conclusione dell'analisi dei diversi tipi di licenze OS disponibili abbiamo provveduto a classificare le 58 licenze secondo un grado di "livello di protezione della libertà", una dimensione che quindi rappresenti in che misura sia garantito il principio del copyleft.

Esaminiamo in dettaglio le caratteristiche principali di quattro licenze *open source*: la GPL, la LGPL, la MPL e la BSD.

La General Public License (GPL) è stata introdotta dalla Free Software Foundation nell'ambito del progetto GNU e garantisce la massima espressione di copyleft non ponendo alcun limite o discriminazione alle libertà di utilizzo, modifica e distribuzione dei prodotti. Inoltre, la GPL non permette l'aggregazione del proprio codice con altri software che non siano coperti dalla stessa licenza. La forma più estrema di protezione del diritto di libera distribuzione contro eventuali comportamenti opportunistici è rappresentata dalla GPL, secondo la quale qualunque lavoro che derivi o semplicemente contenga una parte di software coperto da GPL deve essere distribuito alle stesse condizioni di licenza ("viral clause"). In pratica, un qualunque software che contenga anche solo una riga di codice coperta da GPL diventa immediatamente e interamente *open source* e viene coperto automaticamente dalla stessa licenza GPL. L'adozione della GPL è quindi problematica per i produttori commerciali e questo ha spinto la comunità OS a sviluppare modelli meno restrittivi come la LGPL (Library o 'Lesser' GPL), la quale permette di produrre software proprietario a partire da software aperto (permettendo in pratica la condivisione di librerie di routine senza che si debba applicare la "clausola virus").

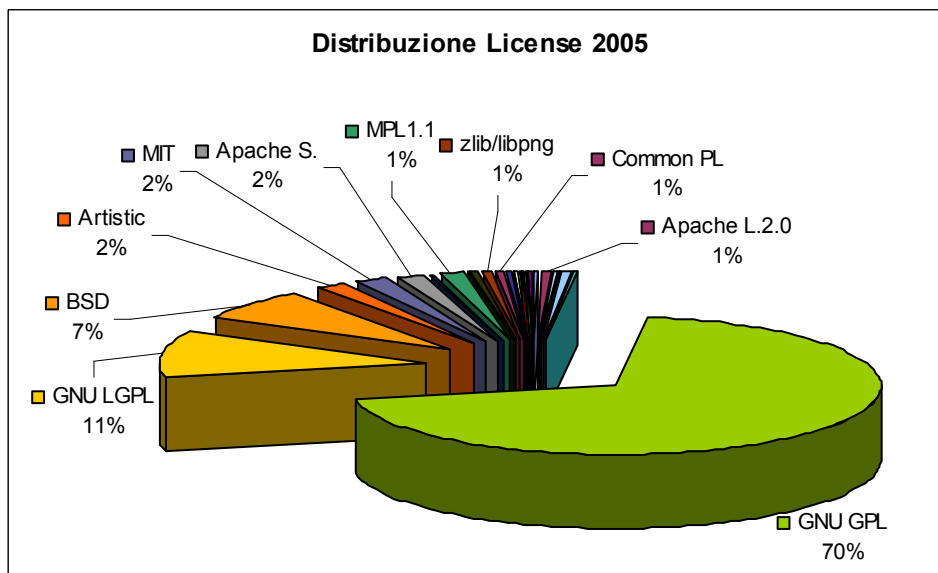
La LGPL è quindi una diretta evoluzione della GPL, la differenza è che consente anche al software proprietario di utilizzare librerie *open source* senza per questo essere costretto a modificare i termini della propria licenza. Questa modifica è stata necessaria perché le librerie di codice necessitano di essere utilizzate e richiamate anche all'interno di altri prodotti non necessariamente coperti dalla GPL.

La MPL (Mozzila Public License) è stata proposta da Netscape come evoluzione della NPL (Netscape Public License). La NPL prevede infatti dei privilegi riservati all'impresa Netscape,--come la possibilità di mantenere private parti di codice sviluppate in ambito *open source*--. Naturalmente questa clausola impediva che questa licenza potesse essere considerata *open source*. Eliminando questa clausola di salvaguardia, Netscape ha allora sviluppato la MPL, che per le altre caratteristiche è simile alla NPL, consentendo anche di integrare il software da essa coperto in prodotti commerciali.

La Berkeley Software Distribution (BSD) pur soddisfacendo la *open source* definition, permette di mantenere private le sole modifiche applicate ai codici originali. Non vi è quindi l'obbligo di distribuire il nuovo codice o di applicarvi la stessa licenza del software originale. Importante ribadire che la possibilità di privatizzare il codice è limitata alle sole parti modificate privatamente senza l'aiuto esterno della comunità *open source* (altrimenti si contraddirebbe la *open source* definition). La licenza concede quindi maggiori libertà di adattare e miscelare i prodotti con altro codice (anche privato), ma vi è anche la possibilità che alcuni sviluppatori non rilascino alla comunità le modifiche da essi introdotte.

Come si può notare dalla Figura 1 sotto, la netta maggioranza dei progetti sviluppati in ambito *open source* all'interno di SourceForge.net risultano essere coperti dalla GPL. Una possibile spiegazione di questo fenomeno potrebbe consistere proprio nell'estrema protezione delle libertà che questa licenza garantisce, oltre anche

all'effetto che la clausola virus ha di modificare le licenze del codice con cui entra a contatto.

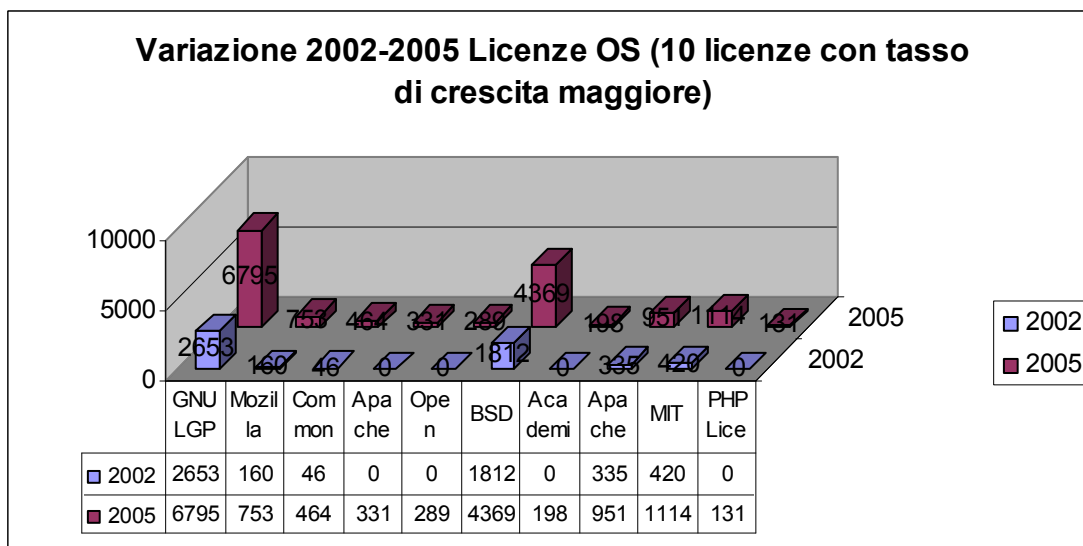


**Figura 1** Distribuzione delle Licenze OS tra i progetti presenti in SourceForge [dati Luglio 2005]

D'altra parte, il proliferare del numero di licenze diverse nel corso del tempo potrebbe dipendere anch'essa dalla necessità di differenziare e di modificare secondo termini diversi la estrema protezione delle libertà attuata attraverso la GPL. Ragione di questa tendenza, così come scopo dell'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" (Giuri e Torrisi, 2004).

L'analisi dei dati relativi all'allegato 1 ci permette di confermare questa tendenziale aumento del numero di licenze. Si noti, infatti, che nel 2005 sono presenti 26 nuovi tipi di licenze che nella precedente rilevazione non esistevano.

La prima analisi che possiamo compiere sulle frequenze di impiego delle licenze è proprio quella relativa ai tassi di crescita nell'utilizzo dei diversi tipi di licenza nel triennio 2003-2005. Da questo studio si nota che tra le 10 licenze con maggiore tasso di crescita, solo quattro sono licenze non presenti nel 2002 (cfr. Figura 2). Questa considerazione ci permette di affermare che, nonostante lo sviluppo di nuovi tipi di licenze, quelle tradizionali conservano un forte interesse ed utilizzo tra i programmatori.

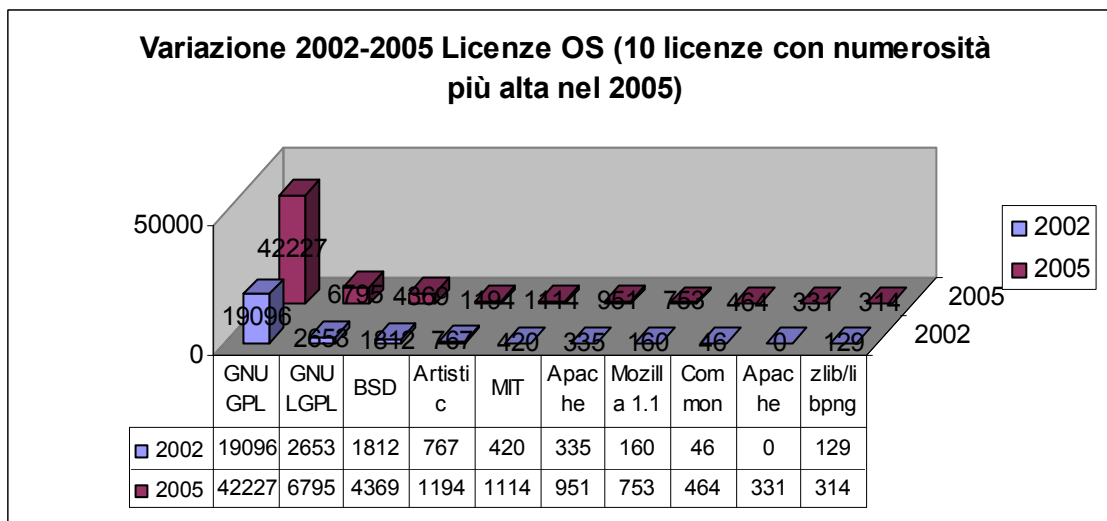


**Figura 2** Variazione 2002-2005 della distribuzione delle licenze con maggior tasso di crescita (dati SourceForge.net).

In particolare si noti come il miglior risultato per una “nuova” licenza sia quello di Apache License 2.0 che nel 2005 viene ad essere utilizzata in trecentotrentuno progetti all’interno della comunità SourceForge.net. La crescita maggiore in assoluto la realizza invece la LGPL che in tre anni quasi triplica il numero di progetti in cui viene utilizzata.

La distribuzione delle licenze (per tipo) presenti in SourceForge.net (Figura 3) mostra la netta prevalenza della licenza GPL, che con i suoi oltre quarantaduemila progetti nel 2005 distanzia nettamente la seconda licenza per numerosità che si ferma a quasi settemila (si tratta inoltre della LGPL, del cui particolare legame con la GPL abbiamo già discusso).

Altra caratteristica da notare è che l’unica licenza non presente nel 2002 ad essere utilizzata in buona misura dalla comunità è la Apache License, che figura, oltre che tra le prime dieci licenze per tasso di crescita, anche tra le prime dieci per numerosità assoluta. Tra le licenze già presenti nel 2002 invece ben cinque (LGPL, BSD, MIT, Apache Software, MPL 1.1) possono vantare sia una buona copertura dei progetti che un buon tasso di crescita.



**Figura 3** Variazione 2002-2005 della distribuzione delle licenze con maggiore numerosità (dati SourceForge.net)

## 5. Primi risultati

Dall'analisi della riclassificazione delle licenze riconosciute dall'OSI emergono alcuni tratti interessanti. In particolare per tutte e 58 le licenze risultano presenti delle clausole che richiedono espressamente la disponibilità del codice sorgente, la possibilità del suo utilizzo e della modifica. Le clausole di distribuzione sono quelle che presentano le differenze più interessanti da licenza a licenza e che definiscono la vera natura della licenza. Per la maggior parte, la distribuzione può avvenire purché la nuova versione renda il codice sorgente sempre disponibile, venga distribuita sotto la stessa licenza e vengano riportate le notifiche delle modifiche apportate al codice originale. Alcune licenze prevedono dei limiti di tempo in cui il codice sorgente debba rimanere a disposizione della comunità, con la possibilità in seguito di non renderlo disponibile.

Altro tratto comune a tutte le licenze riguarda le clausole che limitano completamente la responsabilità degli sviluppatori relativamente a un malfunzionamento o qualunque altro tipo di problema legato al software. Anche la garanzia del software non è prevista nelle licenze. Inoltre è sempre specificato che ad ogni re-distribuzione del software questa clausola venga sempre allegata in modo da tutelare sia gli ultimi sviluppatori che quelli iniziali. Interessante notare che una delle possibilità offerta da molte licenze (non tutte) per poter ottenere il pagamento di una fee per la distribuzione del software è proprio quella di assumersi l'onere della garanzia sul prodotto.

Per quanto riguarda i diritti speciali garantiti agli sviluppatori iniziali e la protezione dei diritti morali degli sviluppatori, queste clausole sono esplicitamente e dettagliatamente regolate solo in poche licenze. Inoltre generalmente le licenze più recenti non considerano direttamente questi diritti.

La dimensione d'analisi relativa alla gratuità merita attenzione. La maggior parte delle licenze utilizza una clausola standard, con cui si afferma che il software può essere ridistribuito e venduto in diverse forme e sotto determinate condizioni. Generalmente è obbligatorio che non si debba pagare una ulteriore fee per ottenere il codice sorgente se si è già in possesso dell'eseguibile (a meno di ragionevoli costi di accesso). Vi sono comunque delle licenze che prevedono espressamente la gratuità del software distribuito, anche se in larga misura risultano avere dei campi di applicazione particolari (specie nel settore educational).

La licenza GPL risulta poi essere l'unica dotata della clausola virus, e questo sarebbe in linea con l'idea che i diversi tipi di licenza siano stati sviluppati per ottemperare alla forte protezione copyleft della licenza sviluppata dalla FSF.

La maggior parte delle licenze inoltre (tranne in pratica la GPL) consente, anche se a condizioni diverse e più o meno restrittive, di combinare software con altro codice sia *open source* che proprietario (in quest'ultimo caso alcune licenze presentano significative differenze).

L'ultima dimensione d'analisi che abbiamo utilizzato per classificare i diversi tipi di licenza riguarda un indice di difesa del copyleft che la licenza è in grado di assicurare (cfr. figura 4).

Licenze 2002 e 2005 - Indice di "protezione"

		Indice di "protezione"				Totale
		Basso	Medio-basso	Alto-medio	Alto	
Licenze 2002	N.	2	19	6	2	29
	Percentuale riga	6,9%	65,5%	20,7%	6,9%	100,0%
	Percentuale colonna	22,2%	61,3%	60,0%	50,0%	53,7%
Licenze 2005 (nuove)	N.	7	12	4	2	25
	Percentuale riga	28,0%	48,0%	16,0%	8,0%	100,0%
	Percentuale colonna	77,8%	38,7%	40,0%	50,0%	46,3%
Total	N.	9	31	10	4	54
	Percentuale riga	16,7%	57,4%	18,5%	7,4%	100,0%
	Percentuale colonna	100,0%	100,0%	100,0%	100,0%	100,0%

**Figura 4** Indice di Protezione Aperta assicurata dalle diverse licenze

La GPL, che è la licenza più restrittiva e che quindi garantisce le più ampie libertà di modifica, utilizzo e distribuzione, è stata quindi considerata una licenza ad alto li-

vello di protezione della libertà. Secondo questa dimensione abbiamo classificato le licenze secondo quattro possibili valori: protezione alta, medio-alta, medio-bassa e bassa. Come risultato della nostra analisi abbiamo constatato che la maggior parte delle licenze si attestano su una posizione medio-bassa. Questo probabilmente è dovuto al fatto che molte delle licenze che cadono in questa categoria sono state sviluppate da delle società o comunque al fine di entrare in contatto anche con software commerciale-tradizionale. In questa categoria rientrano poi anche molte licenze estremamente simili tra loro tranne che per alcune particolarità relative spesso al legame privilegiato con i soggetti che hanno contribuito a svilupparle (es. la clausola per cui il nome della società deve essere sempre citato).

Dall'analisi dell'allegato 1 possiamo invece concludere che la licenza in assoluto più diffusa all'interno della comunità di SourceForge.net è la GPL, in grado di garantire il massimo della protezione delle libertà. Questa considerazione si collega inoltre all'evidenza che molte licenze sono adottate da un numero molto basso di progetti (solo diciassette licenze sono utilizzate da un numero di progetti superiore ai 100). L'analisi diviene ancora più puntuale se si considera che solo quattro tipi di licenze sviluppatasi dopo il 2002 raggiungono i 100 progetti, mentre la maggior parte viene utilizzata da meno di dieci progetti. Questa constatazione può essere spiegata dal fatto che le società che decidono di partecipare per parte delle loro attività allo sviluppo di software *open source*, preferiscono sviluppare delle proprie licenze piuttosto che utilizzare quelle già esistenti. Infatti dall'analisi delle clausole di queste licenze, risulta spesso emergere la loro perfetta similitudine ad altri tipi più diffusi di licenze in cui di solito viene ad essere modificata o aggiunta una sola clausola nella quale si richiede di citare sempre il nome dell'azienda stessa, --per evidenti motivi di marketing-- oppure si vuole impedire che il nome dell'azienda possa essere utilizzato senza il consenso della stessa.

Questi risultati sembrano essere in linea con la tendenza che si inizia a registrare recentemente di ridurre il numero di licenze (o quantomeno di non crearne di nuove) e di semplificarne il contenuto restando comunque la possibilità per chiunque di sviluppare la licenza adatta per il proprio progetto. In questo senso interessante la strada intrapresa da qualche tempo dalla comunità di Creative Commons che attraverso dei moduli base delle licenze che possono essere combinati in maniera diversa dagli sviluppatori, consente di ottenere delle licenze fortemente personalizzate in base alle proprie esigenze ma allo stesso tempo rende di facile lettura e interpretazione le differenze tra i diversi tipi di licenza.

## 6. Conclusioni

Il fenomeno *open source* pone una serie di interrogativi che riguardano la regolamentazione della competizione e i diritti di proprietà intellettuale. Il dilemma tradizionale per la regolamentazione anti-trust e per la legge è il trade-off tra il garantire incentivi per l'innovazione permettendo profitti monopolistici e impedire l'accesso

alla proprietà intellettuale che potrebbe servire come elemento complementare per ulteriori innovazioni.

Una prima possibile implicazione di policy che deriva dall'analisi dell'*open source* potrebbe essere che regimi di protezione della proprietà intellettuale troppo restrittivi possono impedire l'implementazione di modelli di produzione innovativi che Internet e le nuove tecnologie rendono oggi più vantaggiosi che in passato. Internet è infatti essa stessa il risultato di una combinazione di incentivi pubblici e privati. Poiché la sua utilità è il fornire informazioni, che sono contemporaneamente input e output per future innovazioni e ricerche, la propagazione delle informazioni risulta un elemento altamente positivo.

L'esperienza *open source* costituisce una sfida avvincente ai tradizionali dubbi e idee sugli effetti di politiche di protezione dei diritti di proprietà intellettuale lasche o deboli sull'innovazione. Infatti il fenomeno *open source* sembra rappresentare un meccanismo endogeno di innovazione su scala globale che riesce a garantire un efficiente modello di produzione sostenibile nel tempo.

L'applicazione delle logiche *open source* non è confinabile, a nostro giudizio, solo al settore della produzione del software; i primi effetti sono stati di avere riaperto il dibattito nelle politiche economiche di sviluppo offrendo un differente modo di interpretare la relazione tra protezione dei diritti di proprietà intellettuale e innovazione. Si hanno peraltro già i primi segnali di applicazioni in altri settori e discipline, spesso a partire dalle scienze e dal mondo della ricerca che "contamina" altri settori dell'attività. In modo enfatico si sostiene, è chiara la forzatura, che come la catena di montaggio è stata l'immagine del modello industriale fordista, l'*open source* può essere l'essenza di un'economia della conoscenza e dell'innovazione. Con la catena di montaggio le macchine, invece che i lavoratori, guidavano la produzione. Se poco si condivide di questa visione, non si può però negare la complessità e la centralità delle motivazioni e dei modelli di collaborazione che le esperienze di *open source* offrono. Creare condizioni favorevoli alla collaborazione intellettuale sembra essere uno dei più grandi vantaggi del modello *open source*.

La protezione dei diritti di proprietà intellettuale attuale persegue scopi conosciuti in particolare di tutela del proprietario; il copyright e le licenze danno agli innovatori il diritto di impiegare le proprie innovazioni per un certo periodo di tempo. In seguito le innovazioni diventano di pubblico dominio. Se licenze possono proteggere i prodotti, in un'economia della conoscenza devono proteggere anche le idee e i concetti. La sfida è di sviluppare un diritto di proprietà intellettuale in grado di far condividere l'innovazione con un più ampio numero di soggetti. Questo è in fondo il tentativo che le logiche *open source* cercano di compiere. Le decine di licenze, dalla GPL di Stallman a quelle proposte dalla Creative Commons, sono un importante sforzo in questa direzione. Grazie a queste licenze, la proprietà è qualcosa che deve essere distribuita piuttosto che protetta, e i proprietari sono più guardiani che guardie.



## Bibliografia

- Bies R., T. Tripp e M. Neale, 1993, "Procedural Fairness and Profit Seeking: The Perceived Legitimacy of Market Exploitation", *Journal of Behavioral Decision Making*, 6, 243-56.
- Bonaccorsi, A. e C. Rossi (2002) "Why *Open source* Software can Succeed?", *LEM working paper series, Sant'Anna Pisa*.
- Cusumano M.A., 2004, "Reflections on Free and Open Software", *Communication of the ACM*, Ottobre, vol.47, n.10.
- Falk A., E. Fehr e U. Fischbacher 1999, "On the Nature of Fair Behavior", *working Paper n.17, University of Zurich*.
- Fehr W. e K. Schmidt, 1999, "A Theory of Fairness, competition and cooperation", *Quarterly Journal of Economics*, pp.817-868.
- Frey B., 1997, "A Constitution for knavers crowds out civic virtues", *Economic Journal*, pp.1043-1053.
- Fulk J., Heino R., Flanagan A.J. e P.R. Monge, (2004) "A Test of the Individual Action Model for Organizational Information Commons", *Organization Science*, Vol.15, n.5, settembre – ottobre.
- Gatcher S. e E. Fehr, (1999), "Collective Action as Social Exchange", *Journal of Economic Behavior and Organization*, 39, pp.341-369.
- Giuri, P. e S. Torrisi (2004) *Economia e Management dell'Open source Software*. alcune Note di Ricerca. *LEM working paper series*.
- Giuri, P., G. Rocchetti e S. Torrisi (2002) *Open source Software: From Open Science to New Marketing Models*. *LEM working paper series*, n.23.
- Hardin, G., 1968, "The tragedy of the commons", *Science*, 162, pp. 1243–1248.
- Heller, M. and R. Eisenberg (1998). "Can Patents Deter Innovation? The Anticommons in Biomedical Research." *Science* 280, 698-701.
- Kahneman D., J.L. Knetsch e R.H. Thaler, 1986, "Fairness as a Constraint on Profit Seeking: Entitlement in the Market", *American Economic Review*, 76, 728—41.
- Kogut B. e A. Metiu, 2001, "Open-Source Software Development and Distributed Innovation", *Oxford Review of Economic Policy*, vol.17, n.2, pp.248-264.
- Lakhani, K.R. e E. von Hippel, 2003, "How *Open source* Software Works: Free User-to-user Assistance", *Research Policy*, n.32, pp.923-943.
- Lee Gwendolyn e R.E., Cole, 2003, "From a Firm-Based to a Community-Based model of Knowledge Creation: The Case of Linux Kernel Development", *Organization Science*, Vol.14m n.6.
- Lerner, J. e J. Tirole (2001) Some Simple Economics of *Open source*. *Journal of Industrial Economics*, n. 50, pp.197-234.
- Mazzoleni, R. e R.R. Nelson (1998). Economic Theories about the Benefits and the Costs of the Patents, *Journal of Economics Issues*, n.4, pp.1031-1052.
- Muffatto, M. e M. Caldani, 2004, *Open source. Strategie, organizzazione, prospettive*, Il Mulino, Bologna.
- Ostrom, E., 1990, *Governing the Commons: The Evolution of Institutions for Collective Action*, Cambridge University Press, Cambridge, MA.
- Raymond, E.S., 1998, *The Cathedral and the Bazaar*. *First Monday*, 3, 3.
- Raymond, E.S., (1999), *The Cathedral and the Bazaar: Musings on Linux and Open source by an Accidental revolutionary*, O'Reilly & Associates, Sebastopol, CA.
- Rosenberg, N. (1982), *Inside the Black Box: Technology and Economics*. Cambridge University Press, New York.

- Titmuss, R. M. (1971), *The Gift Relationship: From Human Blood to Social Policy*, New York, Pantheon.
- von Hippel, E., and G. von Krogh. 2003. *Open source* software and the 'privatecollective' innovation model: Issues for organization science. *Organization Science* 14 (2):209-223.

## Allegato n.1

Analisi dei diversi tipi di licenze sotto cui sono distribuiti i progetti presenti nel database della comunità SourceForge.net. I dati si riferiscono a due rilevazioni, la prima compiuta nel Giugno 2002 e la seconda nel Luglio 2005 (si noti la presenza nel 2005 di alcune licenze non ancora sviluppate nel 2002 --le licenze presenti in entrambe le rilevazioni sono in corsivo).

LICENZA	2002		2005	
	N°	%	N°	%
<i>Public Domain</i>	856	3,11	1633	2,58%
<i>Other/Proprietari</i>	549	1,99	1040	1,65%
<b>Licenze Approvate OSI</b>	<b>26124</b>	<b>94,90</b>	<b>60520</b>	<b>95,77%</b>
Dettaglio licenze approvate OSI				
<i>GNU GPL</i>	19096	69,37%	42227	66,82%
<i>GNU LGPL</i>	2653	9,64%	6795	10,75%
<i>BSD</i>	1812	6,58%	4369	6,91%
<i>Artistic</i>	767	2,79%	1194	1,89%
<i>MIT</i>	420	1,53%	1114	1,76%
<i>Apache Software</i>	335	1,22%	951	1,50%
<i>Mozilla 1.1 (MPL1.1)</i>	160	0,58%	753	1,19%
<i>Common PL</i>	46	0,17%	464	0,73%
<i>Apache License 2.0</i>	0	0,00%	331	0,52%
<i>zlib/libpng</i>	129	0,47%	314	0,50%
<i>Open Software License</i>	0	0,00%	289	0,46%
<i>Mozilla 1.0 (MPL1.0)</i>	227	0,82%	228	0,36%
<i>Qt PL</i>	139	0,50%	215	0,34%
<i>Academic Free License</i>	0	0,00%	198	0,31%
<i>Python (CNRI)</i>	158	0,57%	140	0,22%
<i>PHP License</i>	0	0,00%	131	0,21%
<i>Python Software FL</i>	11	0,04%	126	0,20%
<i>IBM PL</i>	37	0,13%	81	0,13%
<i>Apple PSL</i>	17	0,06%	54	0,09%
<i>wxWindows Library L.</i>	0	0,00%	50	0,08%
<i>SUN Industry SSL</i>	28	0,10%	47	0,07%
<i>SUN PL</i>	4	0,01%	47	0,07%
<i>Jabber OSL</i>	20	0,07%	39	0,06%
<i>Eclipse PL</i>	0	0,00%	37	0,06%
<i>Univ. of Illinois</i>	3	0,01%	36	0,06%
<i>Zope PL</i>	3	0,01%	34	0,05%
<i>W3C L</i>	3	0,01%	30	0,05%
<i>Nethack GPL</i>	16	0,06%	28	0,04%
<i>Intel OSL</i>	11	0,04%	24	0,04%
<i>Sleepycay</i>	4	0,01%	23	0,04%
<i>Attribution Assurance L.</i>	0	0,00%	20	0,03%

Eiffel FL 2.0	0	0,00%	16	0,03%
Reciprocal PL	0	0,00%	16	0,03%
Historical Permission ND	0	0,00%	15	0,02%
<i>OGTSL</i>	3	0,01%	15	0,02%
Fair L	0	0,00%	8	0,01%
<i>Nokia OSL</i>	5	0,02%	8	0,01%
<i>Ricoh SCL</i>	6	0,02%	7	0,01%
<i>Eiffel FL</i>	5	0,02%	7	0,01%
Common Devel. Distrib. L	0	0,00%	6	0,01%
<i>Vovida SL</i>	1	0,00%	6	0,01%
<i>X.Net</i>	0	0,00%	6	0,01%
NASA OS	0	0,00%	3	0,00%
CUAL OPL 1.0	0	0,00%	2	0,00%
Intessa	0	0,00%	2	0,00%
Lucent 1.02	0	0,00%	2	0,00%
Sybase Open Watcom PL	0	0,00%	2	0,00%
<i>Mitre</i>	5	0,02%	2	0,00%
Adaptive Public License	0	0,00%	1	0,00%
EU Data Grid	0	0,00%	1	0,00%
Frameworkx OL	0	0,00%	1	0,00%
Lucent PL (Planet9)	0	0,00%	1	0,00%
Naumen PL	0	0,00%	1	0,00%
OCLC RPL 2.0	0	0,00%	1	0,00%
RealNetworks PL	0	0,00%	1	0,00%
<i>Motosoto</i>	0	0,00%	1	0,00%
Computer Ass. Trusted OSL	0	0,00%	0	0,00%
Educational Community L	0	0,00%	0	0,00%
<b>Totale</b>	<b>27529</b>	<b>100,00%</b>	<b>63193</b>	<b>100,00%</b>

## Allegato n.2

### SourceForge.net

L'analisi è condotta a partire dai dati messi a disposizione della comunità di SourceForge.net che raccoglie più di 100.000 progetti di sviluppo *open source* attivi. Lo studio si è quindi basato su un dataset pubblico relativo ai progetti *open source* integrato con dati specifici sulle modalità di sviluppo. L'accesso ai dati è avvenuta nell'arco di diversi mesi, e l'ultimo accesso per la raccolta dei dati riferiti a questo contributo è avvenuto nel Luglio 2005.

### Allegato n.3

#### **Open source Definition**

- Libertà di Re-distribuzione: la licenza non dovrebbe restringere alcuna parte dal vendere o distribuire il software come componente di una distribuzione di software aggregati contenente programmi provenienti da diverse fonti. La licenza non dovrebbe prevedere il pagamento di una royalty o di altre fee per questo tipo di transazione;
- Codice Sorgente: il programma deve includere il codice sorgente, e deve permettere la distribuzione del codice sorgente così come del codice compilato. Nel caso il prodotto non fosse distribuito con il codice sorgente, deve esserci un modo ben pubblicizzato di ottenere il codice sorgente per non più del ragionevole costo della sua riproduzione e preferibilmente questo si deve poter scaricare da Internet senza ulteriori costi. Il codice sorgente deve essere nella forma preferibile in cui un programmatore possa modificare il programma. Celare deliberatamente il codice non è permesso.
- Opere Derivate: la licenza deve permettere modifiche e opere derivate, e deve permettere che siano distribuite sotto gli stessi termini della licenza del software originale;
- Integrità del Codice Sorgente dell'Autore: la licenza può restringere la distribuzione del codice sorgente in forma modificata SOLO se la licenza consente la distribuzione di "patch file" con il codice sorgente allo scopo di modificare il programma. La licenza deve esplicitamente assicurare la distribuzione del software costruito su delle modifiche al codice sorgente. La licenza può richiedere che le opere derivate abbiano un nome o un numero di versione differente dall'originale.
- Nessuna Discriminazione di Persone o Gruppi: la licenza non può discriminare alcuna persona o gruppo di persone;
- Nessuna discriminazione di Ambiti di Applicazione: la licenza non deve limitare alcuna persona dall'utilizzare il software in uno specifico ambito di applicazione;
- Distribuzione della Licenza: i diritti collegati al programma devono applicarsi a tutti coloro a cui il programma è ridistribuito senza la necessità della stipula di una nuova licenza tra quelle parti;
- La licenza non deve essere Specifica ad un Prodotto: i diritti collegati al programma non devono dipendere dal fatto che il programma faccia parte di una specifica distribuzione di software. Se il programma è estratto dalla distribuzione e usato o distribuito con i termini della licenza del programma stesso, tutte le parti cui il programma è ridistribuito devono avere gli stessi diritti garantiti congiuntamente con la distribuzione software originale;

- La licenza non deve essere restrittiva di altro software: la licenza non deve contenere restrizioni su altro software che viene distribuito insieme con il software protetto dalla licenza stessa. Per esempio, la licenza non può pretendere che tutti gli altri programmi distribuiti attraverso lo stesso mezzo debbano essere software *open source*;
- La licenza deve essere neutrale per la tecnologia: nessuna restrizione può essere prevista dalla licenza relativamente a una specifica tecnologia o interfaccia.