

UNIVERSITA' DEGLI STUDI DI TRIESTE

**Facoltà di Scienze della Formazione**

**Corso di Laurea in Scienze della Comunicazione**

Tesi di Laurea

in

Sociologia della Comunicazione

**IL SOFTWARE LIBERO/OPEN SOURCE**

Una dimensione sociale

Relatore: Prof. FRANCO FILENI

Correlatore: Dott. MASSIMO SBARBARO

Laureando: ANDREA TODON

Matricola: n. 45000894

**Anno Accademico 2003-2004**

Il presente documento è disponibile all'indirizzo:

<http://toddylog.altervista.org/works.html>



L'intero lavoro è sottoposto a licenza **Creative Commons - Attribuzione - Non commerciale - Non opere derivate 2.0 Italia**, consultabile all'indirizzo

<http://creativecommons.org/licenses/by-nc-nd/2.0/it/>

## INDICE

<b>1. Introduzione.....</b>	<b>3</b>
<b>2. Storia ragionata del software libero/open source.....</b>	<b>6</b>
2.1 Una storia non così recente.....	6
2.2 I primi hacker del software.....	7
2.3 La nascita della rete.....	12
2.4 La cultura Unix.....	14
2.5 La nascita del personal computer.....	15
2.6 Software libero: contesto e nascita.....	19
2.7 Linux.....	22
2.8 Apache.....	25
2.9 La Open Source Definition.....	26
2.10 Software aperto per l'azienda.....	28
2.11 Software aperto per tutti.....	29
2.12 Il punto della situazione.....	31
<b>3. Software libero/open source: caratteristiche generali.....</b>	<b>34</b>
3.1 Cos'è il software libero/open source.....	34
3.2 Le licenze.....	39
3.3 Debolezze del modello free software/open source.....	43
3.4 Sostenibilità economica del modello.....	49
<b>4. Il modello di sviluppo.....</b>	<b>57</b>
4.1 Un nuovo modo di sviluppare software.....	57
4.2 I partecipanti.....	61

4.3 Le motivazioni dei partecipanti.....	63
4.4 Le comunità del movimento free software/open source.....	69
4.5 Usabilità .....	80
<b>5. Software libero e partecipazione sociale.....</b>	<b>87</b>
5.1 La società dei saperi.....	87
5.2 Gli strumenti telematici.....	91
5.3 Intelligenza collettiva e modello free software/open source.....	97
5.4 Oltre il software: sperimentazioni.....	102
5.4.1 Creative Commons.....	102
5.4.2 Wikipedia.....	106
5.4.3 Open Directory Project (ODP) .....	110
<b>6. Software libero/open source e Pubblica Amministrazione.....</b>	<b>113</b>
6.1 Utilità del software libero/open source nella Pubblica Amministrazione.....	113
6.2 Il contesto europeo.....	118
6.3 La situazione italiana.....	121
<b>7. Software libero/open source e Paesi in Via di Sviluppo.....</b>	<b>128</b>
7.1 Il digital divide: un quadro introduttivo.....	128
7.2 Software libero/open source nei PVS: progetti e usi concreti.....	130
7.3 Software libero/open source: opportunità per i PVS.....	133
<b>8. Conclusioni.....</b>	<b>137</b>
<b>Bibliografia.....</b>	<b>142</b>
<b>Sitografia.....</b>	<b>152</b>

## 1. Introduzione

Il fenomeno del software libero/open source ha attirato l'attenzione di un pubblico relativamente vasto solo negli ultimi anni, sebbene le sue origini ufficiali risalgano alla prima metà degli anni ottanta e le pratiche di condivisione del software siano rintracciabili fin dalle origini della scienza informatica. Molto spesso interpretato erroneamente come un movimento anti-Microsoft, sovente considerato un modello di sviluppo capace di produrre solamente software di qualità amatoriale e molte volte confuso con il software freeware, gratuito ma non "libero", recentemente è stato ampiamente rivalutato da aziende e istituzioni nonché dalla letteratura accademica, che lo ha reso oggetto di studi approfonditi in svariati campi.

In particolare, la ricerca economica ha cercato di spiegare la viabilità di un modello apparentemente assurdo dal punto di vista economico, provando a capire le motivazioni delle aziende che lo hanno scelto per le loro attività; la psicologia sociale si è chiesta invece quali siano le ragioni che spingono un così alto numero di programmatori a contribuire in forma volontaria ai progetti open source; alcuni autori hanno cercato di spiegare il fenomeno dal punto di vista giuridico, interrogandosi sul modo in cui esso riarticola l'idea di proprietà intellettuale; altri, sulla scia delle intuizioni del programmatore Eric S. Raymond, hanno tentato un paragone tra il movimento del software libero/open source e le società arcaiche, richiamandosi agli studi sul dono e contestualizzandoli in un quadro economico post-scarità; infine, alcuni studi di tipo più marcatamente ingegneristico hanno cercato di spiegare il fenomeno dal punto di vista dell'architettura del codice.

Questa tesi si pone due obiettivi: da un lato, abbozzare un quadro complessivo, nel tentativo di fare chiarezza e delineare un punto della situazione in merito ad un fenomeno variegato e multifaccettato che è stato spesso interpretato confusamente; dall'altro, cercare dei nodi di congiunzione tra le interpretazioni del fenomeno proposte dai diversi ambiti accademici in un'ottica multidisciplinare, giustificata dalla stessa natura caleidoscopica dell'oggetto studiato.

Rimandando ad altre sedi una completa revisione critica della letteratura prodotta sul software libero/open source, mi sono limitato a formulare un discorso che metta in evidenza i legami e le intersezioni possibili tra le diverse discipline in merito ad un fenomeno che può essere considerato, a tutti gli effetti, di "*frontiera*".

Nel corso di questo lavoro ho ritenuto utile affiancare allo studio della letteratura accademica un'intensa attività di relazione in Internet, attraverso la partecipazione a forum e mailing list. Di particolare stimolo è stata la partecipazione saltuaria a Wikipedia e l'attività di editore nell'Open Directory Project, due progetti ispirati al modello di sviluppo free software/open source.

Inizio la trattazione di questa tesi ripercorrendo la storia della nascita e dello sviluppo del software libero/open source, cercando di mettere in luce le continuità tra le prime pratiche di condivisione del software, agli albori della scienza informatica e le loro più recenti evoluzioni.

In un secondo momento propongo una definizione del software libero/open source, evidenziandone le caratteristiche principali, i limiti al suo sviluppo e alla sua diffusione e descrivendo i modi in cui si articola il suo modello economico.

Verranno poi affrontate le motivazioni dei partecipanti al modello di sviluppo, analizzando le consuetudini degli sviluppatori alla luce del dibattito attualmente in corso in ambito sociologico e psicologico sulle comunità virtuali.

Nel quinto capitolo, riferendomi alle teorie di più ampio respiro sulla società dell'informazione, con il proposito di indagare sulle capacità del modello di stimolare un rapporto più partecipato e consapevole nei confronti degli strumenti telematici, propongo un parallelo tra il modello di intelligenza collettiva descritto da Levy nella sua opera più celebre e il modello free software/open source discutendone l'espansione ad altri ambiti oltre a quello del software.

Nei capitoli sesto e settimo, infine, fornisco un quadro dei possibili ruoli assegnabili al software libero/open source nelle Pubbliche Amministrazioni e nei cosiddetti Paesi in Via di Sviluppo, con particolare attenzione ai processi di sviluppo locale e ai processi in corso verso l'*eGovernment*, e l'*eDemocracy*.

## **2. Storia ragionata del software libero/open source**

### 2.1 Una storia non così recente

Quando si parla del movimento free software/open source lo si considera spesso come un movimento recente, un nuovo modo di sviluppare e distribuire software diverso da quello proprietario.

In realtà il fenomeno non è assolutamente nuovo e si posiziona lungo una linea di continuità che ci porta direttamente alle origini della scienza informatica.

Le pratiche di libera distribuzione del software, insieme al suo codice sorgente, e quelle di condivisione delle modifiche apportate ai programmi erano comunissime fin dagli anni '60. Si può ben dire che nei primi anni della nascita dell'informatica esistesse solo software libero. Si può assistere ad una prima diffusione del software proprietario soltanto a partire dalla metà degli anni '70, quando i computer iniziarono a diffondersi su una scala più ampia. Fu proprio alla fine di questo decennio che nacquero le prime avvisaglie del movimento che viene oggi chiamato Open Source.

Si può guardare al movimento Free Software non come a un fenomeno nato dal nulla, ma come a un movimento che, in reazione alle politiche di chiusura e segretezza adottate da alcune aziende produttrici di software, cerca di tenere in vita, riadattandole in chiave nuova, quelle istanze etiche, filosofiche e tecnologiche che fin dalle origini dell'informatica hanno permesso la nascita di numerose innovazioni e la diffusione di una cultura tecnica comune a moltissimi programmatori.

Ripercorreremo alcune tappe dello sviluppo della tecnologia informatica cercando di rintracciare quegli elementi che attestano una continuità con il movimento del software non proprietario.

In una seconda fase poi, analizzando la storia recente del movimento, cercheremo di capire come questi elementi si articolino nel dar vita a qualcosa di nuovo che sta modificando non solo l'economia del campo informatico ma anche la società nel suo complesso.

## 2.2 I primi hacker del software

Se vogliamo andare alla ricerca delle prime avvisaglie di quell'etica, legata strettamente ad un ambito tecnico specifico, che ha fornito le basi per l'attuale fenomeno del software libero/open source, dobbiamo fare un salto all'indietro negli Stati Uniti dei primi anni '60.

La storia è ambientata a Cambridge, e precisamente all'interno del Massachusetts Institute of Technology. Fu proprio qui che nacque il Tech Model Railroad Club, un club di modellismo in cui si radunarono diversi appassionati di scienza, tecnologia e informatica. Il club era il luogo adatto per fare esperienza sui circuiti elettrici, in attesa di poter mettere le mani sui computer mainframe a schede perforate dell'Ibm, a quel tempo riservati a un ristretto "clero"<sup>1</sup> di tecnici.

Fu proprio presso il TMRC che nacque il termine "hack", che in futuro avrebbe conquistato grande successo e si sarebbe anche prestato a numerosi fraintendimenti e slittamenti di significato. Negli anni '80 il vocabolo hacker ha assunto per l'opinione pubblica un valore dispregiativo, legato al concetto di "pirata informatico". Per i

---

<sup>1</sup> S. LEVY, *Hackers. Gli eroi della rivoluzione informatica*, ed. it., Milano, ShaKe Edizioni Underground, 2002 (quarta edizione), p.23.

protagonisti degli eventi di cui ci stiamo occupando, un “hack” era la brillante risoluzione di un problema di qualsiasi tipo (non solo tecnologico o legato all’informatica) in maniera creativa, arguta e soprattutto gioiosa. Quello che rendeva speciale un “hack” era la passione della scoperta, la voglia di partecipare attivamente ad un processo creativo, l’amore per la sperimentazione. Il fine non era la risoluzione del problema in sé, ma l’atto stesso di cercare di risolverlo. Il fulcro dell’attenzione era cercare di scoprire i segreti dell’elettronica, testare i limiti delle macchine per capirne il loro funzionamento, usando spesso tecniche non convenzionali. Era un modo di lavorare che rifuggiva dalle guide e dai manuali e che si basava sull’apprendimento spontaneo, pratico e antiburocratico.

Quando al MIT arrivò il Tx-0, il primo computer a transistor, alcuni membri del TMRC si lanciarono a provarlo, essendo quest’ultimo circondato da una minore burocrazia dell’Ibm 704. Lavorando in maniera inconsueta rispetto agli utenti “normali” di questo computer, che lo usavano prevalentemente per applicazioni, questi appassionati iniziarono a scrivere diversi programmi per facilitarne l’utilizzo, come *assembler*<sup>2</sup> e *debugger*<sup>3</sup>.

Nel passare numerose ore di fronte alla console del Tx-0, questi amanti dei computer stavano andando inconsapevolmente a costituire una cultura che avrebbe prosperato e si sarebbe arricchita negli anni, consistente di principi etici, linguaggio comune, leggende folkloristiche e che sarebbe stata raccolta, a partire dagli anni ‘70 all’ interno del Jargon File, “un compendio del dialetto hacker che illumina diversi aspetti della tradizione, del folklore e dell’umorismo degli hacker”<sup>4</sup>.

---

<sup>2</sup> Un programma che traduceva il linguaggio macchina, composto da numeri binari, in linguaggio assembly, che usa abbreviazioni simboliche per dare istruzioni alla macchina.

<sup>3</sup> Un programma di correzione degli errori di programmazione, detti *bug*.

<sup>4</sup> E. S. RAYMOND (a cura di), *The Jargon File version 4.4.7*, 2003. Reperibile in <http://www.catb.org/~esr/jargon/>.

Il modo in cui operavano questi membri del TMRC lasciava trasparire un'etica e una visione dell'informatica tutta particolare. Per questi appassionati i computer dovevano essere accessibili ogni volta ce ne fosse stato bisogno da chiunque li volesse usare. Per capire le macchine bisognava “metterci su le mani”<sup>5</sup>, aprirli e smontarli per capire quali fossero le loro potenzialità. Lo spirito di ricerca e sperimentazione li spingeva a rifiutare qualsiasi forma di burocrazia. Vi era la convinzione che per ottenere i migliori risultati e le più grandi innovazioni il processo di creazione dovesse essere lasciato libero e spontaneo, non pianificato e privo di regole arbitrarie. Le regole erano utili solo se efficaci per una situazione specifica. Un'altro principio importante, che sarebbe stato poi fondamentale per la nascita del movimento open source, stabiliva che l'informazione dovesse essere libera.

Questo era ancora più vitale in un'epoca in cui l'informatica era ancora agli inizi e i computer erano visti come qualcosa di misterioso. Ecco che i primi hackers iniziarono a far vivere una prassi in cui chiunque apportasse dei cambiamenti a un programma rendesse il proprio lavoro disponibile agli altri utilizzatori, in modo che potessero lavorarci sopra senza essere costretti a “reinventare la ruota”.

Questo punto derivava sì da motivi tecnici legati all'arretratezza e alla giovinezza della scienza informatica di quei tempi ma dipendeva in gran parte dall'attitudine anticonformista e sperimentatrice di questi utilizzatori.

Lo spirito che li accompagnava era ben diverso da quello dell'IBM, che invece di puntare sull'innovazione faceva riferimento a concetti di stabilità e a meccanismi ampiamente testati e collaudati, senza spingersi in maniera incauta oltre certi limiti.

Le macchine dell'IBM di quegli anni erano infatti caratterizzate da alta affidabilità e stabilità, ma anche da un livello molto basso di interattività con l'utente.

---

<sup>5</sup> S. LEVY, *Hackers*, cit., p. 21.

Si rivelavano poco adatte ad uno stile di programmazione basato sull'improvvisazione, sulla curiosità e sull'esplorazione.

Già ai suoi albori la cultura hacker mostrava caratteri squisitamente meritocratici, secondo i quali si veniva considerati dal gruppo unicamente per le proprie azioni e non secondo criteri quali il ceto, l'età, il genere, la razza o la confessione religiosa.

A chiudere il quadro vi era inoltre la convinzione che il codice di un programma dovesse essere considerato come un testo a tutti gli effetti. Il codice sorgente possedeva una sua bellezza intrinseca e poteva essere guardato nello stesso modo in cui si guarda un'opera d'arte. Un programma non doveva semplicemente assolvere bene la sua funzione, ma doveva anche essere scritto con un certo stile. Nacquero anche particolari forme di competizione, in cui si cercava di scrivere una stessa funzione impiegando il numero minore di righe di codice. Questo modo di concepire la programmazione ha, a nostro avviso, avuto un ruolo decisivo nella esportazione dell'etica hacker ad ambiti esterni a quello informatico e ha costituito la piattaforma per la nascita di diverse pratiche sociali. Concepire il programma come un testo pone infatti le basi per l'estensione dell'"etica hacker" a diversi temi, quali la libertà dell'informazione, la proprietà intellettuale, il diritto all'educazione e all'apprendimento.

Nel 1961 fece il suo ingresso una nuova macchina, il Pdp-1, regalata al Research laboratory of electronics (Rle) del MIT dalla Digital Equipments Corporation (DEC).

Questa si rivelò fin dall'inizio la macchina più adatta agli hackers, poiché era stata progettata per un utilizzo più interattivo, pensata com'era più per la ricerca che per l'elaborazione numerica. Gli hackers iniziarono subito a scriverci dei programmi per poterla usare al meglio e, potendola utilizzare più spesso del Tx-0, cominciarono a servirsene nei modi più svariati e impensabili: a ritmo serratissimo nacquero

assemblatori, compilatori musicali, giochi di scacchi professionali e giochi di guerre spaziali.

Il modo di lavorare degli hackers iniziava intanto a essere riconosciuto e al MIT ormai numerosi professori avevano capito che per imprimere un maggiore sviluppo alla scienza informatica era necessario permettere l'utilizzo dei computers a più gente possibile. Per questo motivo cercarono di costruire un sistema *time-sharing*<sup>6</sup>, che avrebbe permesso a più persone di utilizzare contemporaneamente lo stesso computer. Ma i sistemi time-sharing esistenti non soddisfavano le pretese degli hackers, così i planner del IA lab diedero il compito di crearne uno nuovo ad alcuni di loro. Crearono l'Incompatible Time Sharing System. Questo sistema incarnava in sé vari principi dell'etica hacker: aveva una struttura aperta, e stimolava la collaborazione e lo scambio reciproco tra gli utenti. Questo sistema operativo multi-utente, comune a molti utilizzatori, si impose soprattutto per le possibilità di continua implementazione che offriva agli utenti, che iniziarono a lavorarci sopra in modo cooperativo per migliorarlo costantemente.

Alla fine degli anni '60 l'etica hacker si stava diffondendo anche in altri luoghi, soprattutto in California, dove si stava creando una comunità di programmatori con molti punti in comune con quella del MIT. Iniziavano a svilupparsi i laboratori della Stanford University e della Carnegie-Mellon e una nuova infrastruttura, basata sull'integrazione di diverse tecnologie, stava per avvicinare i programmatori dei diversi centri di ricerca: Arpanet.

---

<sup>6</sup> Un sistema time-sharing è costituito da un solo computer (detto mainframe), molto potente, e da molti terminali che sfruttano contemporaneamente la sua potenza di elaborazione. La potenza di calcolo del computer centrale viene ripartita equamente tra tutti i terminali collegati e attivi in un dato periodo di tempo.

## 2.3 La nascita della rete

Arpanet andò on-line per la prima volta nel 1969. Era il prodotto della convergenza di svariate ed eterogenee ricerche effettuate da diversi poli universitari e finanziate dal Dipartimento della Difesa americano. I finanziamenti verso la ricerca informatica partirono alla fine degli anni '50, dopo lo shock del lancio dello Sputnik sovietico. Da quell'episodio il Dipartimento della Difesa decise di adottare una politica più spregiudicata e diede vita all'A.R.P.A. (Advanced Research Project Agency), autorizzata ad offrire finanziamenti per i progetti più innovativi, portati avanti da ingegneri brillanti e visionari.

Le ricerche si focalizzavano sui campi più svariati. L'adozione al Mit di sistemi time-sharing permise di fare esperimenti sulle reti di computer. Se era possibile infatti, connettere diversi terminali a un'unico elaboratore in uno stesso edificio, forse lo sarebbe stato anche per distanze maggiori.

In altri centri di ricerca, come lo Stanford Research Institute, il Lincoln Lab, l'Università dello Utah, solo per citarne alcuni, si conducevano altri studi: interfacce uomo-computer, computer grafica e comunicazioni di gruppo.

In comune tra tutti i ricercatori che guidavano questi progetti c'era la convinzione che i computer avrebbero potuto fornire degli strumenti utili alla gente. Il desiderio era quello di permetterne l'utilizzo a più persone possibili, anche non esperte.

L'idea di base per lo sviluppo di Arpanet venne fornita da Paul Baran della RAND Corporation nel 1964. Cercando di studiare un sistema che potesse assicurare la persistenza della comunicazione anche nel caso di una catastrofe nucleare, Baran suggerì di creare un sistema senza un centro di controllo. I nodi della rete dovevano essere indipendenti uno dall'altro. Lo strumento utilizzato sarebbe stato la

*commutazione di pacchetto*: i messaggi sarebbero stati divisi in pacchetti di uguali dimensioni e ogni pacchetto poteva seguire lungo i nodi della rete il percorso più conveniente per poi andare a ricostruire il messaggio originale una volta giunti a destinazione.

Bisogna aggiungere che se la creazione di un sistema di comunicazione decentrato utilizzabile a scopi militari era l'obiettivo ufficiale e dichiarato dei progettisti, non era l'unico. Essi, infatti, già sognavano un mondo in cui le persone avrebbero potuto comunicare le une con le altre attraverso dispositivi di diverso tipo situati in ogni parte del mondo, scambiandosi informazioni e dando vita a culture basate su interessi condivisi.

I primi nodi della rete furono UCLA, Stanford Research Institute, University of Santa Barbara e University of Utah. In breve tempo si collegarono alla rete anche altri centri di ricerca, come il MIT, BBN, RAND Corporation e Carnegie-Mellon, solo per citarne alcuni.

Inizialmente la rete era stata pensata come un sistema di redistribuzione della potenza di calcolo tra i sistemi time-sharing dei vari nodi: la capacità di elaborazione sarebbe stata disponibile dove ci fosse stato maggior bisogno.

Ma già in queste prime fasi, nonostante l'accesso fosse permesso ai soli ricercatori dei dipartimenti che collaboravano con il Dipartimento della Difesa si potevano intuire le potenzialità di questa tecnologia d'essere utilizzata per creare, mantenere e sviluppare delle comunità.

Ancora prima del '69 alcuni programmatori, intuendone i possibili vantaggi, svilupparono dei sistemi di posta elettronica per i sistemi time-sharing.

Quando diversi nodi furono in rete, i ricercatori iniziarono a usare la posta elettronica per scambiarsi informazioni, per spedirsi materiale su cui lavorare, per rendere manifeste le scoperte fatte.

Cosa ancor più interessante, iniziarono a utilizzare la posta elettronica per fini non attinenti a quelli della ricerca e alle finalità del Dipartimento della Difesa: essa si rivelò uno strumento perfetto per la socializzazione, la messaggistica personale, la chiacchiera. Inoltre, grazie alla capacità di adattarsi sia alla comunicazione “uno a uno” che “uno a molti”, si rivelava ottima per le discussioni pubbliche e quindi per la creazione di comunità.

Un esempio celebre è la prima mailing-list, SF-LOVERS, un elenco degli indirizzi di posta elettronica di alcuni ricercatori dell'ARPA appassionati di fantascienza.

La rete aveva già dimostrato di poter diventare uno strumento utilizzabile anche per scopi non inerenti alla ricerca informatica e di poter diventare uno strumento utile a tutti.

## 2.4 La cultura Unix

In questi anni si stavano affacciando nuovi contributi che avrebbero formato una nuova cultura tecnica legata ai computer, che sarebbe stata significativa per il futuro sviluppo del software libero.

Il sistema operativo Unix iniziò ad essere sviluppato nei Bell Laboratories dell'AT&T da Ken Thompson e Dannie Richie. Sempre da questi laboratori nacque il linguaggio “C”, con cui venne scritta la prima versione veramente innovativa di Unix. Scrivere tutto un sistema in C garantiva la possibilità di portare il sistema operativo su macchine di diverso tipo, elemento chiave per la sua futura diffusione.

Nella prima metà degli anni '70 UNIX fu adottato dall'Università di Berkeley e diversi utilizzatori iniziarono a svilupparlo e ad aggiungervi diverse funzionalità, così uscì la Berkeley Software Distribution, una distribuzione Unix sviluppata dall'università omonima. Le due distribuzioni, quella dell'AT&T e quella di Berkeley continuarono a uscire in varie release<sup>7</sup>, sfruttando l'una le innovazioni dell'altra. Grazie a una funzionalità del sistema operativo, la UUCP ( Unix to Unix Copy Protocol ), creata da un gruppo di studenti di due università americane escluse da Arpanet, i computer Unix, che iniziavano a diffondersi in varie università, poterono creare una rete a bassa velocità.

Nacque Usenet, un sistema di teleconferenze per computer Unix che mise in contatto tutti i suoi utilizzatori, che diedero vita a delle comunità. Nel 1983, con la decisione della DEC di passare dalla linea PDP-10 alla linea VAX e con la conseguente decisione di vari centri di ricerca di passare dal sistema ITS al sistema Unix che garantiva maggiore portabilità, la cultura formatasi intorno a Unix e quella storica dei dipartimenti legati all'ARPA si integrarono tra loro.

Nello stesso anno prendeva vita anche Fidonet, un sistema di bacheche elettroniche per pc che faceva uso del modem e di uno speciale software.

## 2.5 La nascita del personal computer

Contemporaneamente alla nascita di Internet, nella costa Ovest degli Stati Uniti, era in corso un'altra rivoluzione, che mirava alla creazione di microcomputer, accessibili alle tasche di tutti e non soltanto agli istituti accademici.

Alla fine degli anni '60 a Berkeley diversi gruppi di attivisti informatici, come il Community Memory e la People's Computer Company, convinti che il potere dei

<sup>7</sup> Termine usato per indicare una particolare versione di un software.

computer dovesse essere diffuso tra la gente comune, misero in atto diversi happening e progetti per mettere in contatto e formare i cittadini all'uso dei computer. Tra queste iniziative vi fu la posa di un terminale accessibile a chiunque all'interno di un negozio di dischi. Il terminale fungeva da bacheca elettronica e ospitava i messaggi e le inserzioni dei cittadini, oltre a notizie e informazioni sulle attività della città. Fu un'evento curioso più che un servizio con reali potenzialità ma ebbe il merito di avvicinare la gente comune ai computer, rappresentandoli come oggetti meno spaventosi e di possibile utilità per tutti.

Nel 1975 alcuni dei propulsori di questi progetti crearono un'associazione, la Homebrew Computer Club, attraverso la quale settimanalmente ingegneri, hobbyisti e appassionati si incontravano in un garage di Berkeley per scambiarsi informazioni e lavorare intorno al loro sogno più grande: un computer da tavolo.

L'input per la nascita di questo gruppo lo diede Ed Roberts, un imprenditore del New Mexico che vendeva un kit per costruire un computer da tavolo primordiale, l'Altair 8080.

Quando futuri membri del gruppo di appassionati di Berkeley vennero a sapere, attraverso la rivista "Popular Electronics", di questo kit, fecero di tutto per ottenerlo. Alla prima riunione del club Steve Dompier, che era riuscito dopo alcuni giorni di lavoro a costruirlo, lo mostrò agli altri membri.

La memoria dell'Altair era ridottissima, di soli 256 bytes, ma gli hackers iniziarono a lavorarci sopra cercando di sfruttare al meglio la sua limitata capacità.

Il passo seguente fu la costruzione di una scheda di memoria da aggiungere all'Altair, costruita da Bob Marsh, per espandere la capacità di elaborazione della macchina. Da quel momento in poi si fece strada un modo sinergico di lavorare a quel progetto

comune, in cui ogni progetto e ogni elemento aggiunto da un membro stimolava e forniva delle basi per lavorare agli altri hackers. La consuetudine voleva che venissero lasciati a disposizione di tutti le specifiche e i disegni di ogni scheda e componente elettronica costruita e, se si trattava di software, il rilascio agli altri del club del codice sorgente del programma.

Di pari passo con la diffusione, seppure ancora germinale, dei computer da tavolo, sempre più persone iniziarono a pensare alla possibilità di guadagnare attraverso la vendita di programmi. Nel 1975, Bill Gates e Paul Allen, con ambizioni di lucro, scrissero una versione del programma Basic per l'Altair. L'azienda di Roberts, la Mits, fece un accordo con i due programmatori per la vendita del linguaggio di programmazione.

Ad un raduno di appassionati il Basic finì tra le mani degli hackers dell'Homebrew, che ne fecero delle copie e iniziarono a distribuirlo tra gli amatori.

Nella sua "Lettera aperta sulla pirateria" Bill Gates accusava gli hobbisti di avere commesso un furto nei suoi confronti e portava per la prima volta alla luce una nuova concezione, che rifiutava l'idea del software condiviso. Metteva inoltre l'accento sul fatto che senza un adeguato compenso ai programmatori nessuno si sarebbe mai lanciato nell'impresa di scrivere del buon software professionale.

Il Basic, comunque, proprio per il modo in cui veniva indebitamente distribuito dagli hackers, si diffuse come uno standard, facendo paradossalmente la fortuna della neonata azienda di Gates, la Microsoft, alla quale si rivolsero tutte le aziende che avevano bisogno di qualche adattamento o perfezionamento del programma.

Molti dei membri dell'Homebrew continuarono a lavorare come avevano sempre fatto, ridistribuendo i risultati dei loro sforzi e chiedendo in cambio dei contributi volontari in denaro e che gli altri programmatori riservassero loro lo stesso trattamento.

Nel 1975 l'azienda di Marsh, la Processor Technology, insieme a Lee Felstein, fondatore del Community Memory, costruirono insieme un computer completo, fornito di involucro e di monitor, il Sol. Nel 1976 per la prima volta un personal computer finiva in televisione, nel Tomorrow Show di Tom Snyder. Da "mostri" enormi per addetti ai lavori, i computer erano divenuti strumenti meno spaventosi, e iniziavano finalmente a diventare accessibili alle persone comuni, non soltanto ai ricercatori e agli ingegneri.

Un altro membro dell'Homebrew Computer Club, Steve Wozniak, progettò un computer che sarebbe stato destinato ad avere grande successo, l'Apple. Attraverso l'opera di pianificazione dell'amico Steve Jobs, che aveva coinvolto nella neonata azienda dei validi manager e designer industriali, crearono una versione del primo computer di Wozniak diretta a un pubblico più ampio, l'Apple II, un computer da tavolo completo dotato anche di un aspetto accattivante e definito.

Una fiera di computer organizzata da alcuni membri dell'Homebrew, situata al Civic Auditorium di San Francisco, con le sue 13.000 presenze, consacrò l'ingresso dei computers nella schiera degli elettrodomestici.

Fu da questo punto in poi che le questioni sollevate da Gates due anni prima cominciarono a venire a galla.

## 2.6 Software libero: contesto e nascita.

Sul finire degli anni '70 ci si trova in uno scenario in cui gli utilizzatori sono organizzati intorno alle diverse culture descritte e si intravedono grandi possibilità di aumento del numero degli utenti di computer. Le varie reti stavano andando a unificarsi. Contemporaneamente alla crescita degli utenti della rete crescevano anche le prospettive di commercializzazione del software e crescevano i segreti delle compagnie e l'atmosfera di irrigidimento dei rapporti tra le aziende.

La diffusione dei personal computer favorì la nascita di numerosi hacker DOS e Mac. La mancanza di sorgenti disponibili su cui poter lavorare creò una situazione in cui questi appassionati di computer “non riuscirono mai a sviluppare una cultura consapevole”<sup>8</sup>.

Sempre secondo Eric Raymond:

*Il ritmo dei cambiamenti era talmente veloce che ben cinquanta diverse culture tecniche nacquero e cessarono di esistere con la rapidità di una farfalla, senza mai raggiungere la stabilità necessaria allo sviluppo di un gergo, di un folklore e di una storia propri<sup>9</sup>.*

Agli inizi degli anni '80 la cultura hacker si ritrovò in un periodo difficile: le macchine Unix erano costose e l'indisponibilità di materiale libero per creare una base comune su cui lavorare minacciava le comunità e non ne favoriva la nascita. E' in questi anni che spuntano i primi tentativi, vivi ancora oggi, di ricreare quello spirito di collaborazione e

---

<sup>8</sup> E. S. RAYMOND, *Breve storia sugli hackers*, in *Open Sources. Voci dalla rivoluzione Open Source*, a cura di Ch. Di Bona - S. Hockman - M. Stone, ed. it., Milano, Apogeo 1999, p. 27.

<sup>9</sup> E.S. RAYMOND, *Breve storia sugli hackers*, in *Open Sources. Voci dalla rivoluzione Open Source*, cit..

lavoro cooperativo che aveva costituito la linfa vitale e fulcro dell'innovazione della scienza informatica.

La storia ufficiale del movimento Free Software inizia nel 1984, anno in cui Richard Stallman scrisse il Manifesto GNU e fondò la Free Software Foundation.

L'obiettivo ultimo del Progetto GNU<sup>10</sup> era quello di costruire un intero sistema operativo, corredato dei programmi principali, che fosse libero da influenze proprietarie. Secondo la filosofia di Stallman, nessuna azienda doveva condizionare l'attività creativa degli sviluppatori, inoltre i programmi dovevano essere accessibili a tutti, in modo che chiunque potesse apportarvi dei cambiamenti in modo da poterli migliorare e aggiungervi sempre nuove funzionalità. Quello che spinse Stallman a creare la Free Software Foundation fu l'insofferenza nei confronti di una situazione in cui gli sviluppatori, spinti dalle loro rispettive aziende, si trovavano in condizione di non poter collaborare con i loro colleghi delle aziende concorrenti, in quanto tenuti a non rivelare i segreti industriali. Stallman lavorava da anni nel laboratorio di Intelligenza Artificiale del Massachusetts Institute of Technology, permeato per tradizione dall'etica hacker. Per lui questa situazione era un grande perdita di energie, in quanto utili informazioni che potevano rendere il lavoro più semplice non potevano essere trasmesse tra persone che svolgevano compiti simili. Non solo, egli vedeva questa situazione come la disgregazione di una fiorente comunità, che aveva sempre avuto rapporti collaborativi e aveva creato intorno a sé una cultura fatta di simboli, abitudini e lessico condiviso.

Stallman poi decise di basare GNU sul modello del sistema operativo Unix per ragioni connesse alla filosofia architettonica di questo sistema.

---

<sup>10</sup> Il termine GNU sta per "GNU's Not Unix", acronimo ricorsivo tipico della tradizione hacker. Il nome stava a significare che GNU sarebbe stato basato su un sistema Unix, ma che non sarebbe stato un sistema proprietario come quest'ultimo.

In Unix infatti esistono due principi fondamentali:

1. Ogni cosa è un file.
2. Qualsiasi cosa venga costruita, bisogna cercare delle soluzioni che assolvano in modo efficace ed efficiente ad una sola funzione.

Questa struttura modulare garantisce la possibilità di lavorare separatamente ad un'unico elemento, senza doversi preoccupare troppo delle interazioni con gli altri.

La scelta si rivelò adeguata, in quanto permetteva a sviluppatori di tutto il mondo di lavorare sul progetto che preferivano e poi, solo in un secondo momento, preoccuparsi di come associarlo agli altri elementi.

Richard Stallman alcuni anni prima di dar vita al progetto GNU, scrisse un editor<sup>11</sup> che si diffuse moltissimo nei dipartimenti di informatica statunitensi: Emacs. Seguendo i principi dell'etica hacker, distribuì gratuitamente Emacs, rendendone disponibile il codice sorgente, in modo che gli utenti, che a quei tempi erano prevalentemente programmatori o esperti di informatica, potessero personalizzarlo senza limite. L'unica condizione che mise era che tutte le estensioni apportate al programma dovessero essere disponibili, in modo da migliorarlo continuamente. Questa filosofia si incarnò poi nel 1985 nella licenza GPL (General Public License), ideata da Stallman con l'aiuto di un avvocato, che garantisce all'autore la libertà di fornire agli utenti pieni diritti di distribuzione e modifica, con la garanzia che il software rimanga sempre liberamente distribuibile e modificabile.

---

<sup>11</sup> Un programma che facilita la stesura di testi o, nel caso dei programmatori, la scrittura di codice di programmazione.

Un'altra delle tecnologie che permisero la diffusione del software GNU fu il GCC (GNU C Compiler), un compilatore<sup>12</sup> sviluppato dalla Free Software Foundation e distribuito come software libero.

## 2.7 Linux

All' inizio degli anni '90, il progetto di Stallman di creare un kernel<sup>13</sup> libero, GNU HURD, era ancora molto lontano dal suo completamento, anche se erano stati creati diversi programmi fondamentali per la costruzione di un sistema operativo completo.

A Berkeley inoltre, le distribuzioni di freeBSD, una versione libera di Unix sovvenzionata attraverso fondi statali, stavano attraversando momenti difficili per via delle cause legali con l'AT&T.

A completare il contesto, tra la seconda fase degli anni '80 e i primi anni '90, la rete stava vivendo un periodo di rapida crescita accompagnata da numerose innovazioni che ne permetteranno l'utilizzo a fasce di utenza sempre più larghe.

Nacquero in questi anni tecnologie di configurazione dinamica dei router che permisero la gestione di numeri altissimi di nodi e indirizzi differenti, fattore cruciale per l'espansione di Internet (Arpanet aveva ormai formato una rete a sé chiamata Milnet, riservata alle applicazioni militari). Le innovazioni di questi anni sono state determinate da un lavoro collettivo che ha come polo di coordinamento la Internet Engineering Task Force (IETF), una associazione di studiosi, ricercatori e ingegneri aperta a tutti gli interessati a contribuire allo sviluppo della rete.

---

<sup>12</sup> Un compilatore è un programma che traduce le istruzioni scritte in un linguaggio di programmazione (in questo caso il C) in linguaggio macchina, ovvero in numeri binari interpretabili dal computer. Senza di esso è impossibile far interpretare dal computer il codice scritto dagli sviluppatori.

<sup>13</sup> E' il cuore di un sistema operativo, costituisce la parte del sistema che fa "girare" le applicazioni.

Caratteristico di questo organo di coordinamento è il suo processo aperto e trasparente di documentazione nella messa a punto degli standard.

Il processo stesso di formazione dei gruppi di lavoro è decentrato: i gruppi si formano sulla base di esigenze individuali di singoli ricercatori che si uniscono ad altri per lavorare su un singolo tema. I documenti IETF sono di libero accesso e sono regolati da forme di copyright tese ad assicurare la loro disponibilità e la possibilità di lavorarci sopra ulteriormente.

La rete giocherà un ruolo chiave nello sviluppo del prodotto più importante del modello di sviluppo del software libero/open source: il sistema operativo GNU/Linux.

Nel 1991 Linus Torvalds, uno studente di informatica di Helsinki, per imparare a scrivere software complesso, si mise al lavoro per creare un kernel libero che potesse girare sul suo 386. Per iniziare utilizzò il codice di Minix, scritto dal professore di informatica Tanenbaum, disponibile agli studenti. Torvalds si avvale dei consigli di alcuni utenti del gruppo di utenti di Minix (su Usenet) e cercò qui di attirare l'attenzione sul suo progetto.

Molte persone iniziarono a lavorarci sopra, mentre Torvalds ridistribuiva le modifiche con rapidità e cercava di mantenere alto l'entusiasmo intorno al progetto, facendo sentire utili e parte di un gruppo tutti quelli che contribuivano. Gli sviluppatori di queste prime versioni, ancora instabili, di Linux (così venne soprannominato il sistema), si raggrupparono intorno al newsgroup [alt.os.linux](#).

Il kernel di Linux era perfetto per integrarsi con il sistema GNU: avrebbe costituito il kernel che secondo il progetto della FSF avrebbe permesso la costituzione di un sistema operativo libero a tutti gli effetti. Questo sistema completo si sarebbe chiamato

GNU/Linux. Le due componenti lavorano in sinergia: i programmi hanno bisogno di un kernel per poter girare, un kernel senza programmi è inutile. Moltissimi sviluppatori utilizzavano il GCC di Stallman per sviluppare Linux: la sinergia tra i due progetti era già delineata prima che vi fosse un' alleanza formale.

Nel 1992 Linux fece un primo passo per la sua adozione da parte di un numero maggiore di persone: lavorando sui codici del sistema X Window, gestiti dall'X Consortium e resi disponibili sotto la licenza X del MIT, nacque Xfree86, un'interfaccia utente a finestre che rendeva più semplice e intuitivo l'utilizzo del sistema.

Sempre in quest'anno iniziarono a fiorire delle iniziative che preannunciavano quello che sarebbe stato il modello economico del software libero/open source: le distribuzioni.

Numerose aziende iniziarono a vendere delle versioni di Linux ripulite dai bug e orientate a usi particolari del sistema: venivano inseriti programmi e funzionalità aggiuntive, già inserite nel pacchetto proposto. Le distribuzioni nacquero per facilitare l'utilizzo del prodotto agli utenti meno esperti e per risparmiare lavoro aggiuntivo di assemblaggio e compilazione dei pacchetti. Insieme ai pacchetti software le aziende proponevano anche servizi di assistenza sul prodotto.

Tra le varie distribuzioni spiccava la Debian, che nacque con il proposito di creare una distribuzione Linux di elevata qualità e per rendere credibile il modello a codice aperto, che poteva essere screditato da distribuzioni avventate e distanti dallo spirito comunitario che stava alla base di questo modo di fare software. La Debian ufficializzò questo intento con un manifesto, le "Debian Free Software Guidelines", scritto da Bruce Perens. Il documento fissava quali dovessero essere le caratteristiche della licenza d'uso di un programma perché potesse essere considerato "libero". Le distribuzioni, tra cui Red Hat, SuSe, Mandrake, Turbo Linux, hanno il merito di essersi preoccupate non solo

dell'evoluzione del sistema ma soprattutto della sua fruibilità da parte dell'utente finale, fattore chiave per la diffusione di Linux.

Un'altra apertura nei confronti degli utenti comuni, voluta da Torvalds e dai "suoi" sviluppatori, è stata la scelta di sviluppare Linux in due versioni: una sperimentale, detta Beta, riservata ai programmatori che intendevano aggiungere funzionalità e correggerne i bug, l'altra testata e stabile, destinata principalmente alle applicazioni.

Linux nella prima metà degli anni '90 continuava a raggiungere sempre più utenti, aiutato anche dalla diffusione a macchia d'olio di Internet.

Nel frattempo era nato, ad opera degli studiosi del Cern di Ginevra Tim Berners Lee e Robert Calliau, il World Wide Web.

In analogia con lo sviluppo contagioso di Linux, la decisione di rendere di pubblico dominio il protocollo http e la possibilità di vedere il codice HTML sottostante alle pagine web ha permesso un'apprendimento veloce e diffuso del linguaggio, contribuendo all'arricchimento delle rete, facendola percepire come una risorsa importante a sempre più persone.

## 2.8 Apache

Nell'immaginario comune il software libero/open source è sempre associato al sistema operativo Linux, ma è nel campo del web che è stata giocata una partita fondamentale per lo sviluppo successivo di questo modello.

Il server web<sup>14</sup> HTTPd (HTTPdaemon) era stato sviluppato nel 1993 dall' NCSA (National Center of Supercomputing Applications ) e reso disponibile gratuitamente.

---

<sup>14</sup> Un server web è un software installato su un computer connesso alla rete Internet che ha il compito di soddisfare le richieste ricevute dai browser. Generalmente il server attende la richiesta del client, rintraccia la risorsa richiesta, applica l'azione scelta dal client e gli risponde. I server web di ultima generazione sono costruiti per fornire servizi di vario tipo, dall'autenticazione degli utenti all'accesso ai database.

Quando nacque Netscape Communications numerosi tecnici dell'NCSA passarono a questa azienda per lavorare sulla loro linea di server, e iniziarono a non aggiornare più il software e a non rispondere più alle patch di debug<sup>15</sup> inviate dagli utenti. Così gli utenti di WWW\_Talk, una mailing list che collezionava queste patch, iniziarono ad autorganizzarsi per riempire il vuoto lasciato dall'NCSA. Behlendorf, al tempo amministratore di sistema della rivista on-line Hot-Wired, si offrì volontario per attuare il progetto di un server web libero.

Lo sviluppo di questo software era gestito da un'equipe abbastanza ristretta che riuscì però a mantenere un costante *feedback* con gli utenti, che mandavano continue segnalazioni sui bug e suggerimenti sulle funzionalità che desideravano fossero supportate dal server.

Apache viene rilasciato nella sua versione stabile nel 1995 e da quel momento, in virtù della sua stabilità e dell'alto livello qualitativo, nonché della possibilità di scaricarlo gratuitamente, diventa il server web più utilizzato al mondo. Bisogna poi sottolineare che la disponibilità dei sorgenti di Apache ha costituito una garanzia per il mantenimento di standard unitari nel web, allontanando il pericolo di corruzione degli standard da parte di aziende proprietarie.

Sempre nel campo del web, i linguaggi di scripting Perl, scritto da Larry Wall nel 1987 e poi sviluppato da un numero sempre maggiore di programmatori, e PHP, hanno permesso la creazione di sistemi web dinamici, che offrono all'utente una maggiore interattività, permettendo l'interrogazione di database e la fornitura di pagine personalizzate in relazione alla piattaforma in uso dall'utente.

---

<sup>15</sup> Le patch di debug sono frammenti di codice volti a correggere gli errori di programmazione preesistenti.

## 2.9 La Open Source Definition

Nella seconda metà degli anni '90 il software libero, grazie alla creazione di programmi molto importanti e alla diffusione di Internet tra un numero sempre maggiore di utenti non specializzati, si stava diffondendo in maniera crescente.

La nascita di riviste come Linux Journal e di iniziative editoriali specialistiche come quelle della O'Reilly & Associates, che pubblica manuali di programmi, linguaggi e tecnologie software a codice aperto, in aggiunta ad alcuni saggi sul modello di sviluppo del software libero, tra i quali spicca “La cattedrale e il bazaar” di E. S. Raymond, hanno contribuito a donare credibilità e serietà al movimento, prima considerato soprattutto dalle aziende come qualcosa di amatoriale, inadatto agli sviluppi commerciali e alle esigenze delle grandi imprese.

Molte società che lavoravano principalmente nel software libero iniziarono a sentire l'esigenza di potersi espandere nel campo dei servizi alle aziende ed erano in cerca di una strategia per rendere loro appetibile il software libero.

L'iniziativa venne da Eric Raymond e Bruce Perens, che pensarono che per rendere più accattivante il software libero alle aziende sarebbe stato necessario correggerne il nome.

Il termine “free software”, nella lingua inglese, può infatti prestarsi a equivoci e ambiguità, perché significa sia “libero” che “gratis”. Molte imprese si facevano così un'idea sbagliata di che cosa fosse realmente il software libero, concentrandosi di più sulla sua gratuità che sulla disponibilità del codice sorgente.

Nacque così la Open Source Definition, un documento basato sulle precedenti “Debian Free Software Guidelines” che definiva quali licenze avevano le caratteristiche adatte a rendere Open Source un determinato software. Le direttive di questo documento non si

discostano molto da quelle fornite dalla Free Software Foundation per definire il software libero, ma tra le due posizioni vi è diversità di punti di vista.

L'Open Source Initiative, l'associazione nata per gestire la campagna Open Source e per certificare il software, ha posto l'accento sui vantaggi pratici derivanti dall'apertura del codice sorgente dei programmi e dall'adozione di questo tipo di software nelle aziende.

Ha avuto il merito di analizzare e far conoscere il modello economico alla base dello sviluppo free software/open source, mostrandone la sua sostenibilità.

La Free Software Foundation, invece, vede il fatto di mettere in risalto i vantaggi pratici dell'utilizzo di software a codice aperto come qualcosa di rischioso: il pericolo è che ci si dimentichi dei valori e dell'etica del software libero, basata sul concetto di libertà dell'utente.

## 2.10 Software aperto per l'azienda

La mossa di Raymond e Perens si rivelò una scelta appropriata: numerose aziende di software e hardware iniziavano a considerare il modello open source come qualcosa da valutare e considerare attentamente.

La prima mossa fu fatta da Netscape, che nel 1998 decise di liberare il sorgente del suo browser web Communicator.

Netscape licenziò il codice di Communicator attraverso due licenze distinte: la Netscape Public Licence, che conferiva alla Netscape alcuni privilegi e la Mozilla Public Licence, licenza priva dei privilegi della prima e volta a promuovere la comunità open source e il suo supporto al progetto del browser web Mozilla.

La nascita del browser Mozilla, caratterizzato da un processo di sviluppo aperto guidato dalla comunità di Internet, è stata un momento di svolta per il raggiungimento di standard nel web, mettendo fine alla guerra dei browser tra Microsoft Explorer e Netscape Communicator, che aveva frammentato gli utilizzatori del web in due poli distinti.

Sempre nel 1998, IBM annunciò di utilizzare Apache come software per il suo Websphere Application Server e che avrebbe fornito assistenza aziendale per il free software.

Nacquero software che potevano soddisfare esigenze aziendali, come i database server: PostgreSQL, inizialmente sviluppato da Ingres II, e MySQL. Aziende leader nel campo dei database come Oracle e SAP iniziano a fornire il supporto per GNU/Linux nei loro prodotti.

Il 1999 è l'anno dei produttori hardware: IBM, HP, Dell, Compaq iniziano a supportare il sistema GNU/Linux anche per macchine di alto livello e contribuiscono ad aumentare il livello di credibilità dei prodotti free software/open source.

Le collaborazioni strette con queste grandi aziende fecero la fortuna di distribuzioni come Red Hat, SuSe (diffusa nel mercato europeo) e TurboLinux (mercato giapponese).

## 2.11 Software aperto per tutti

Il sistema GNU/Linux e in generale il software libero traggono le loro origini dalla cultura hacker e dalle prassi di condivisione dei sorgenti dei programmatori.

Questa tradizione prevalentemente tecnica ha determinato un panorama in cui la stragrande maggioranza di programmi "free" era costituita da applicazioni per sviluppatori software e esperti informatici, da strumenti per fare strumenti.

La consuetudine e la facilità dell'utilizzo del computer da parte degli sviluppatori attraverso riga di comando ha di gran lunga rallentato la creazione di software user-friendly, facilmente utilizzabile anche da utenti inesperti.

Fino alla prima metà degli anni '90 gli applicativi per gli utenti finali e i progetti di GUI<sup>16</sup> erano pochissimi nell'arena del software libero, perché gli utilizzatori della piattaforma GNU/Linux erano ancora in gran parte programmatori e tecnici informatici di qualche azienda.

Uno dei primi progetti per un'applicazione rivolta agli utenti finali è stato lanciato nel 1996 da due studenti di Berkeley, Kimball e Mattis. Il progetto si chiamava GIMP (General Image Manipulation Program), un software di editing di immagini, simile al proprietario Photoshop. Inizialmente il programma fu sviluppato utilizzando il toolkit<sup>17</sup> grafico di Unix, Motif. In seguito i due crearono Gtk (GIMP Tool Kit), che iniziò ad essere usato da molte persone che contribuirono poi al suo sviluppo.

Oggi GIMP è un programma di livello professionale, che è stato utilizzato anche per grandi progetti di animazione e grafica internazionale.

Quando uscì Windows 95 il ritardo del sistema GNU/Linux per quanto riguarda il desktop per l'utente finale era molto accentuato.

Ettrich, un programmatore norvegese iniziò a sviluppare un progetto di questo tipo, chiamato KDE (K Desktop Environment), utilizzando un tool kit chiamato Qt, rilasciato gratuitamente e con il sorgente aperto da un'azienda chiamata Trolltech.

Ma Qt non era rilasciato sotto GPL perché la Trolltech voleva mantenere il controllo sulla tecnologia.

---

<sup>16</sup> GUI: Graphical User Interface

<sup>17</sup> Un insieme di librerie.

La comunità del free software (quella che gravitava intorno alla Free Software Foundation) era impaurita dal fatto che un prodotto software non coperto da GPL potesse diventare l'unica interfaccia utente di GNU/Linux e così alcuni sviluppatori iniziarono a sviluppare una GUI interamente libera.

Il tool kit grafico utilizzato era quello di GIMP, Gtk+. Il progetto di interfaccia si chiamava GNOME e vi contribuirono anche alcuni programmatori della Red Hat, particolarmente interessata al progetto. Gli esponenti della comunità free software cercavano intanto di fare pressioni alla Trolltech perché rilasciasse Qt sotto GPL.

La Trolltech non accettò ma rispose creando la KDE Free Qt Foundation, che assicurava che avrebbe sempre reso disponibile una versione libera delle librerie in questione.

Un passo ulteriore in direzione della comunità free software/open source fu compiuto con l'istituzione della Q Public Licence, una licenza open source creata con il contributo di Raymond e Perens.

Nel 2000 le questioni si chiusero quando Trolltech fornì la possibilità di licenziare Qt anche sotto GPL.

Durante la scrittura di questo lavoro i due sistemi di interfacce, KDE e GNOME, convivono sulla stessa piattaforma e differiscono nei risultati grafici, entrambi divenuti molto accessibili e usabili per l'utente finale, che può scegliere tra le due impostazioni.

## 2.12 Il punto della situazione

Nel corso di questo capitolo abbiamo cercato di mettere in luce quegli elementi storici fortemente collegati con la nascita del movimento free software/open source.

Abbiamo cercato in particolare di mostrare come le pratiche di condivisione, rilascio delle modifiche, ridistribuzione e copia dei codici sorgenti dei programmi fossero presenti fin dall'inizio della scienza informatica e che una corrente di questo modo di sviluppare software è giunta fino ai giorni nostri, fino a rappresentare un'ottima alternativa ai sistemi proprietari per quanto riguarda diversi aspetti:

- come modello economico per le aziende che si occupano di software, che possono guadagnare spostando le loro attività dalla vendita dei prodotti ai servizi sui prodotti.
- per le aziende in genere, che possono utilizzare del software gratuito e pagare i servizi forniti dalle aziende di software, senza rischiare di rimanere ancorati alla stessa azienda se questa non fa un buon lavoro.
- per gli utenti finali, che possono disporre di software gratuito spesso di alto livello
- per le pubbliche amministrazioni, che grazie al software a codice sorgente aperto possono soddisfare le loro esigenze di indipendenza dai privati e di trasparenza nel trattamento delle informazioni.

In questa storia breve del software libero abbiamo cercato di spiegare come le istanze originarie si siano modificate e adattate di volta in volta ai nuovi scenari che si presentavano e di come siano riuscite a sopravvivere a diversi momenti critici, come i tentativi di chiudere i software e di istituire segreti industriali nel momento iniziale della commercializzazione di massa dei personal computer e l'ingresso nel mondo dei pc di una folta schiera di utilizzatori inesperti e bisognosi di sistemi semplici da usare e intuitivi.

Negli ultimi anni poi, il movimento free software/open source ha creato una nuova breccia nel muro rappresentato dal monopolio Microsoft nel campo dei sistemi operativi e negli applicativi per l'utente finale assicurando la compatibilità dei documenti scritti su piattaforma GNU/Linux con il sistema operativo Microsoft e mettendo a punto diversi programmi che vanno in questa direzione.

Tra i più importanti ricordiamo Samba, un file server che permette l'interoperabilità di sistemi Windows e Linux all'interno di reti locali (LAN<sup>18</sup>) e Wine, un software che permette di fare girare programmi scritti per piattaforma Microsoft anche su Linux, in maniera semplice e intuitiva.

Le intuizioni di Raymond e Perens, che quando hanno creato la Open Source Initiative prospettavano uno scenario futuro in cui software libero e proprietario potessero convivere su piattaforme comuni, sono sempre più vicine al loro avverarsi.

---

<sup>18</sup> Local Area Network

### 3. Software libero/open source: caratteristiche generali

#### 3.1 Cos'è il software libero/open source

Prima di andare a definire quali caratteristiche siano utili a identificare un determinato software come software libero/open source dobbiamo chiarire alcuni punti oscuri legati alla terminologia che hanno permeato di ambiguità la sua definizione fin dalla sua comparsa.

Il termine inglese “free”, a causa del suo duplice significato ( può essere inteso sia come “libero”, con riferimento alle libertà che garantisce agli utenti, che come “gratis”) ha portato a numerosi fraintendimenti e a diverse proposte di nuove terminologie, tra le quali il diffusissimo termine “open source”, che si focalizza sull’apertura del codice sorgente.

In questo lavoro utilizzeremo unitamente entrambi i termini, in quanto li consideriamo espressione di due filosofie diverse, anche se legate da molti punti in comune, all’interno dello stesso universo.

In particolare la corrente del Free Software, rappresentata dalla Free Software Foundation<sup>19</sup>, è convinta dell’importanza di utilizzare il termine “free”, in quanto mette in luce il concetto di libertà degli utenti. La Open Source Iniziative<sup>20</sup> invece, sostenitrice del termine “open source”, è maggiormente interessata ad evidenziare gli aspetti economici, tecnici e innovativi del software a codice aperto.

E’ importante chiarire che la gratuità del software non è una caratteristica basilare del software libero/open source, e non va confuso con il software *freeware*, gratuito ma

---

<sup>19</sup> <http://www.fsf.org>

<sup>20</sup> <http://www.opensource.org>

non necessariamente “libero”. Con il termine freeware ci si riferisce infatti a software distribuito in forma gratuita ma esclusivamente in formato binario, senza concedere la visibilità dei sorgenti, condizione necessaria per permettere la creazione di versioni modificate e la loro redistribuzione. Generalmente questo tipo di software viene rilasciato gratuitamente per attrarre i clienti verso altri programmi (a pagamento) della stessa casa produttrice.

La possibilità di vendere il software libero/open source è sempre stata contemplata, fin dalla costituzione del Progetto GNU negli anni ottanta e ora sono numerose le aziende che basano la loro attività commerciale sulla vendita di software a codice aperto e di servizi correlati ad esso.

*Indipendentemente dal fatto che si siano ottenute copie di software GNU a pagamento o gratuitamente, si ha sempre la libertà di copiare e cambiare il software, e anche di venderne copie.*

*"Software libero" non vuol dire "non-commerciale". Un programma libero deve essere disponibile per uso commerciale, sviluppo commerciale e distribuzione commerciale.<sup>21</sup>*

Il software libero/open source si identifica per mezzo delle libertà concesse all'utente al momento dell'acquisto o della ricezione da terzi di una copia del programma.

Le principali libertà di cui dispone un utente di questo tipo di software sono le seguenti:

- ogni utente ha la *libertà di eseguire il programma* per qualsiasi fine. Non sono ammesse discriminazioni di alcun tipo sulla nazionalità, confessione religiosa, età, genere o posizione politica dell'utente.

---

<sup>21</sup> FREE SOFTWARE FOUNDATION, *Cos' è il software libero?*, ed. It, Boston, 2001 Reperibile in: <http://www.gnu.org/philosophy/free-sw.it.html>

- Il software deve permettere all'utente la sua *modifica* per rispondere ai propri bisogni individuali. Questo significa che deve essere possibile correggerne gli errori, aggiungere o togliere alcune funzioni, personalizzarlo a proprio piacimento e studiarlo per capire il suo funzionamento. Per poter disporre di questa libertà il prerequisito fondamentale è disporre del codice sorgente. Ogni utente è poi libero di apportare delle modifiche al programma, se lo desidera e ne è capace, oppure può commissionarle ad altri, gratuitamente o a pagamento.
- Deve essere possibile *ridistribuire* copie del software di cui si è in possesso, indipendentemente dal fatto che questo sia stato acquistato oppure copiato da terzi. Lo stesso vale per le modifiche apportate: ogni utente ha la possibilità di ridistribuire pubblicamente i miglioramenti che ha aggiunto, permettendo così ad altri utilizzatori di goderne.

Un requisito fondamentale di questo software è *l'accessibilità del codice sorgente*, il codice scritto dall' autore (o dagli autori) del programma, molto più gestibile del codice binario (linguaggio macchina).

Questa caratteristica innesca una serie di conseguenze positive sull'innovazione.

In primo luogo, fa sì che un programma possa essere costantemente migliorato e consegna il suo futuro alla comunità degli sviluppatori/utilizzatori.

Il futuro di un programma non è nelle sole mani di un'azienda che ne detiene il controllo completo, ma è frutto di un dialogo tra le parti interessate al suo sviluppo.

La direzione che prenderà una determinata tecnologia software è il prodotto di una contrattazione tra i diversi attori: sviluppatori, utenti, aziende, istituzioni.

L'apertura del codice sorgente, in un contesto caratterizzato da un'infrastruttura aperta e distribuita come quella di Internet, può inoltre garantire una maggiore durata della vita

di un software. Anche quando un programma cessa di avere un uso “di massa” o quando i suoi principali sviluppatori smettono di aggiornarlo, tutti gli interessati, aziende o privati, possono organizzarsi per lanciare un progetto per continuare a tenerlo aggiornato e in vita.

E’ la disponibilità dei sorgenti che agevola la portabilità di un programma da una macchina ad un’altra. Questa caratteristica è di importanza sempre maggiore ai giorni nostri, in cui il software sta penetrando in un numero sempre crescente di dispositivi: dagli strumenti casalinghi (software “embedded” o incastonato) al software per l’automazione, utilizzato per eseguire numerose lavorazioni industriali.

Oltre a tutto questo, l’apertura del codice sorgente è anche garanzia di trasparenza.

Il codice di un programma è innanzitutto informazione: attraverso di esso possiamo conoscere le azioni che compie un programma.

Man mano che le principali attività economiche e culturali tenderanno a spostarsi sull’infrastruttura digitale, diventerà sempre più importante sapere *come* un programma compie il trattamento dei dati.

Lo studio del codice di un programma, oltre ad essere un importante strumento di apprendimento per gli studenti di informatica, è un documento a tutti gli effetti e la sua disponibilità pubblica è una garanzia di trasparenza nel prevenire intromissioni nella privacy dei cittadini.

La possibilità di sottoporre i programmi ad una verifica di migliaia di utenti e a un dibattito pubblico attraverso mailing list e forum di discussione è una eccellente garanzia sulla loro sicurezza. Il codice aperto permette infatti di correggere più facilmente rispetto a un software proprietario i bachi del sistema.

La possibilità, grazie all'infrastruttura di Internet che unisce utenti e programmatori di tutto il mondo, di testare i programmi in una moltitudine di contesti diversi per macchine utilizzate, tipo e modo di utilizzo, fornisce una ulteriore garanzia di sicurezza. In poche parole, i programmi vengono per così dire "testati sul campo" e inseriti fin da subito (anche se inizialmente attraverso versioni sperimentali, dette "beta") in contesti di utilizzo reali, più ricchi dei contesti di laboratorio in cui viene testato il software di tipo proprietario.

E' il caso del sistema operativo Linux, che viene rilasciato in due versioni: una stabile e ampiamente testata, destinata al pubblico più vasto ed una sperimentale, spesso piuttosto instabile, in cui vengono inserite le ultime novità, dedicata agli sviluppatori e a tutti quelli che desiderano testarla o farne il debugging.

La filosofia insita nel software libero/open source restituisce ai programmi di computer una dimensione "a tutto tondo": non viene considerata solo la visione utilitaristica, secondo la quale la cosa importante di un programma è ciò che permette di fare, ma anche la dimensione sociale, considerando il modo in cui lo fa e mettendo in luce la questione del *chi* detiene il controllo del suo sviluppo.

Questa stessa dimensione sociale del software a codice aperto è anche alle origini di un possibile rischio: è difficile sapere, soprattutto alla sua nascita, se un certo progetto raggiungerà uno stato di maturità e un buon livello di diffusione. Lo sviluppo di un progetto dipende infatti dalla capacità dei suoi promotori di riuscire a coinvolgere un buon numero di programmatori interessati a collaborare.

Il software libero/open source, infine, si rivela ottimale in uno scenario tecnologico che ci sta abituando a cambiamenti sempre più rapidi e ad una convergenza tra tecnologie differenti sempre più presente.

La visibilità del codice sorgente dona al software una maggiore flessibilità nell'adattarsi a nuove tecnologie, fattore fondamentale per assicurare al tempo stesso stabilità e dinamicità all'infrastruttura informatica odierna.

### 3.2 Le licenze

All'interno del mondo del software libero/open source le licenze con cui i programmi vengono distribuiti ricoprono un ruolo di importanza fondamentale. E' attraverso le licenze che si può stabilire se un determinato programma sia da considerarsi libero o proprietario e da esse derivano i vantaggi di questo modo di fare e distribuire software.

*Strictly speaking, the terms of the distribution licence are the only factor that determines whether or not software is open source.*<sup>22</sup>

Le licenze tradizionali per il software sono basate sulle normative riguardanti il diritto d'autore e generalmente sono finalizzate a difendere gli interessi degli autori e delle case produttrici. Le licenze di software proprietario sono in genere progettate per definire le limitazioni d'uso sul software che viene venduto, concedendo in prestito agli utenti determinati diritti sul software e restringendone altri. La maggioranza di esse solitamente proibiscono la copia a terzi, la modifica del programma, pongono delle restrizioni sull'uso in differenti computer e talvolta delle restrizioni temporali.

Esse sono infatti studiate affinché le case produttrici possano mantenere il controllo, in particolar modo economico, sulla tecnologia in questione.

Il software libero/open source si contraddistingue invece dalla scelta di cedere agli utenti il controllo sul programma, dotandoli di diverse libertà sull'uso del software.

---

<sup>22</sup> E. E. KIM, *An Introduction to Open Source Communities*, Blue Oxen Associates, 2003, p.1, in <http://www.blueoxen.org/research/00007/>

Paradossalmente, anche se in questo caso non si tratta di difendere tanto i diritti dell'autore quanto quelli degli utenti, per garantire queste libertà sono pur sempre necessarie delle licenze. Rilasciare il software come prodotto di dominio pubblico può essere rischioso e finire a lungo andare per restringere i diritti degli utenti: il software di dominio pubblico può infatti essere nuovamente licenziato sotto una licenza proprietaria. Le licenze per software libero/open source, invece, tutelano i diritti degli autori di concedere determinate libertà agli utenti; nel pubblico dominio gli autori, rinunciando a questi diritti, fanno perdere anche le libertà agli utenti.

Le licenze per software libero/open source differiscono per il diverso grado con cui enfatizzano alcuni aspetti rispetto ad altri: protezione dell'apertura del software derivato, riconoscimento dell'autore, mantenimento o meno di alcuni diritti proprietari e compatibilità rispetto a licenze proprietarie e a licenze open source.

Avendo deciso in questo lavoro di considerare il movimento del free software e il movimento open source come due correnti all'interno dello stesso macro-movimento, includeremo nelle nostre considerazioni sia le licenze utili a classificare del software come software libero sia quelle utili a classificarlo come open source.

Diremo, comunque, che i criteri indicati dalla Open Source Definition per definire le licenze per software open source sono meno restrittive di quelle della Free Software Foundation, alcune volte concedendo, pur di rendere appetibile l'idea dell'open source al mondo delle aziende, alcuni privilegi speciali alle aziende coinvolte.

La più usata licenza per software libero è la GNU GPL (GNU General Public License). Nata nel 1989 dalla collaborazione tra Richard Stallman e una équipe di avvocati è tuttora la licenza meglio congegnata e che più delle altre stimola il circolo virtuoso dello sviluppo del software libero/open source. Essa viene anche chiamata *copyleft* (permesso

d' autore), con un'ironico capovolgimento del termine copyright. Il suo funzionamento si basa sul seguente meccanismo: l'autore del programma si assicura i diritti attraverso una semplice nota di copyright e tramite la licenza GNU GPL dà il permesso legale alla copia, distribuzione e modifica del suo lavoro. Tutte le modifiche che vengono apportate ai programmi vanno notificate, mostrando con chiarezza quali files sono stati modificati, da chi e quando.

Una caratteristica della GPL, comune a tutte le licenze per software libero, è la notifica che il programma coperto da licenza viene fornito senza alcuna garanzia. Questa è una clausola essenziale poiché gli autori di software libero, che spesso contribuiscono a titolo volontario e gratuito, smetterebbero di scrivere programmi liberi se si dovessero sobbarcare le responsabilità di funzionamento di un software che viene sviluppato e modificato continuamente da programmatori di tutto il mondo.

La GPL è una licenza congegnata principalmente per fini non commerciali e per incentivare il più possibile la diffusione di programmi liberi. Una delle sue caratteristiche principali è la "viralità": non è permesso infatti rendere private le modifiche al codice di un programma, che devono essere a loro volta "licenziate" sotto GPL. La stessa incorporazione di codice coperto da GPL all'interno di altro codice comporta che il codice ottenuto sia a sua volta coperto da questa licenza.

Inoltre, dal punto di vista della compatibilità questa licenza è molto restrittiva: non è infatti possibile inserire del codice coperto da questa licenza all'interno di software proprietario. Questa caratteristica, se da un lato alimenta l'espansione del software libero a detrimento di quello proprietario, esclude dall'altro i possibili contributi a progetti di software libero da parte di aziende interessate a incorporare il software ottenuto nelle loro proposte commerciali.

La licenza LGPL (Lesser General Public License<sup>23</sup>) invece, studiata in un primo momento per le librerie software, mantiene le caratteristiche di *viralità*<sup>24</sup> della GPL, obbligando a rendere pubbliche le modifiche ma rende possibile l'utilizzo del codice all'interno di software proprietario.

Le licenze X, BSD (Berkeley Software Distribution) e Apache, simili tra loro, permettono di fare del software e del codice di un programma qualsiasi cosa si desideri. E' possibile ad esempio mantenere private le modifiche effettuate al software e licenziare nuovamente il software ottenuto sotto una licenza proprietaria.

La licenza BSD è nata per coprire progetti finanziati da fondi pubblici, per i quali i cittadini avevano già pagato attraverso i versamenti delle tasse.

La licenza BSD, anche se non incentiva le aziende a contribuire a un progetto aggiungendovi del codice proprio e le incoraggia invece a sfruttare semplicemente il codice già disponibile è però ottima se si vuole «promuovere l'uso di un corpus di codice di riferimento che implementi un protocollo o un servizio comune»<sup>25</sup>.

In poche parole una licenza utile quando si è più interessati a diffondere degli standard e protocolli comuni che a diffondere in forma libera un programma specifico.

La Mozilla Public Licence (MPL) indica che le modifiche effettuate ai file distribuiti nel codice sorgente vengano rilasciate sotto lo stesso copyright. Questo fa sì che tutti i miglioramenti effettuati al codice originario tornino a essere utili per il progetto di partenza ma permettendo l'incorporazione di caratteristiche a valore aggiunto in prodotti soggetti a licenze proprietarie. Questa clausola invoglia le aziende a partecipare

---

<sup>23</sup> Originariamente chiamata Library General Public License.

<sup>24</sup> Con l'uso di questo termine ci riferiamo al fatto che il software rilasciato sotto questa licenza obbliga il rilascio sotto la medesima di tutto il software derivato.

<sup>25</sup> B. BEHLENDORF, *Open Source come strategia commerciale*, in *Open Sources. Voci dalla rivoluzione Open Source*, a cura di Ch. Di Bona - S. Hockman - M. Stone, ed. it., Milano, Apogeo 1999, p. 178.

a progetti open source, potendo allo stesso tempo curare i loro fini commerciali e contribuire alla comunità di sviluppatori formatasi intorno a un determinato progetto.

Un'altra clausola di questa licenza fornisce degli strumenti per scongiurare la sempre presente minaccia dei brevetti sul software: il soggetto che contribuisce, attraverso la modifica del codice, a un progetto è tenuto a rinunciare a qualsiasi pretesa di diritti di brevetto. Attraverso questa clausola si prevencono quei casi in cui alcuni sviluppatori inseriscono di proposito del codice già brevettato per poter poi richiedere un pagamento per il suo utilizzo.

Esistono moltissime altre licenze per software libero/open source ma, rifacendosi tutte in qualche modo alle caratteristiche che abbiamo qui sopra menzionato, non le tratteremo in questo lavoro<sup>26</sup>.

Bisogna infine ricordare che dalle licenze qui elencate ne sono derivate altre adatte a coprire altri tipi di artefatti culturali, quali libri, brani musicali, immagini.

Citiamo a questo riguardo le licenze Creative Commons<sup>27</sup>, nate dal lavoro dell'avvocato americano Lawrence Lessig, che assieme a collaboratori da tutto il mondo sta lavorando anche ad un progetto di licenze per le scoperte scientifiche, le Science Commons.

### 3.3 Debolezze del modello free software/open source

Il modello che abbiamo descritto fin qui in termini generali, se da un lato lascia intravedere grandi vantaggi e notevoli capacità di affermazione all'interno delle infrastrutture della comunicazione futura, non è immune da pericoli per il suo sviluppo e dalla possibilità di vedere stemperare i principi che lo hanno reso così interessante agli

---

<sup>26</sup> Per approfondimenti, Vedi:  
FREE SOFTWARE FOUNDATION, *Licenze varie e commenti relativi*, ed. it., in  
<http://www.gnu.org/licenses/licenses.it.html>

OPEN SOURCE INITIATIVE, *Approved licenses*, in <http://www.opensource.org/licenses/>

<sup>27</sup> Il sito ufficiale di Creative Commons è reperibile all' indirizzo <http://creativecommons.org/>

occhi di molti osservatori di vari campi: informatici, economisti, psicologi, sociologi, studiosi dell'innovazione.

Bisogna sottolineare come alcune caratteristiche di questo modello, se da un lato sono responsabili dei suoi vantaggi, per loro stessa natura possono anche essere fonte di debolezza.

Il fatto che i progetti possano nascere e prosperare in maniera indipendente dagli interessi economici di qualche azienda, grazie a un lavoro in gran parte volontario di tutti gli interessati, può essere un motivo di debolezza per le sorti future del progetto stesso.

*...it is not possible to know if a project will ever reach a usable stage, and even if it reaches it, it may die later if there is not enough interest<sup>28</sup>*

Per superare la fase più critica dello sviluppo di un progetto, quella iniziale, la strategia consiste nel coinvolgere il maggior numero di aziende e istituzioni interessate a finanziare parte del progetto o a fornire il lavoro dei propri programmatori, in modo che l'alto numero iniziale di programmatori cooperanti spinga altri ad unirsi.

Per cause affini, è molto spesso difficile sapere quali progetti sono in fase di svolgimento in un dato momento, perché è raro che vengano stanziati fondi per campagne pubblicitarie di un prodotto open source. Esistono però diversi punti nodali dove si possono ottenere informazioni a riguardo dei progetti in corso, nei quali i programmatori possono consultare lo stato dei diversi progetti e decidere di collaborare in varie maniere. I sistemi web più importanti a questo riguardo sono FreshMeat<sup>29</sup> e Source Forge<sup>30</sup>.

---

<sup>28</sup> WORKING GROUP ON LIBRE SOFTWARE, *Free Software/Open Source: Information Society Opportunities for Europe?*, Version 1.2, 2000. Reperibile in: <http://eu.conecta.it/paper/paper.html>

<sup>29</sup> <http://freshmeat.net>

<sup>30</sup> <http://sourceforge.net>

Oltre a questo, esistono diverse compagnie di informatica che hanno spostato il loro interesse dalla vendita del software alla vendita di servizi e consulenza sul software, basata su un'intensa attività di informazione sullo stato dei progetti in corso e sul *testing* dei programmi. Queste aziende, generalmente, oltre a fare attività di ricerca e di prestazione di servizi, dedicano parte delle loro energie alla cooperazione con altre aziende e sviluppatori, dando dei contributi alla comunità.

Le aziende che non operano in questo modo spesso perdono di credibilità di fronte agli sviluppatori, che non sono più stimolati a collaborare ai loro progetti.

Uno dei maggiori pericoli per il software libero/open source arriva dall'esterno ed è rappresentato dai brevetti sul software.

Per trattare questo tema bisogna definire che cosa si intende con il termine "brevetto" e in cosa differisce dalle licenze copyright.

Brevettare del software consiste nel richiedere dei diritti esclusivi su degli algoritmi o delle tecniche usate nel suo sviluppo. Non si parla qui di diritti su parti di codice, ma del modo in cui quella parte di codice è stata sviluppata.

E' spesso capitato che venissero brevettati algoritmi utilizzati diffusamente dai programmatori come conoscenze acquisite: soprattutto in questi casi è evidente come i brevetti costituiscano un ostacolo alla ricerca informatica e un freno all'innovazione tecnologica, obbligando i programmatori a pagare una quota per l'utilizzo di quell'algoritmo o a creare delle tecniche nuove per aggirare la protezione del brevetto.

Gli effetti prodotti dalle patenti sul software sono stati analizzati in numerosi studi economici. Bessen e Hunt<sup>31</sup> nel loro lavoro dipingono un quadro generale di come si è evoluta la legislazione statunitense al riguardo, evidenziando come la rapida ascesa

---

<sup>31</sup> J. BESSEN, R.M. HUNT, *An Empirical look at software patents*, 2004 Reperibile in:  
<http://swpat.ffii.org/vreji/minra/sisku/>

dell'utilizzo di brevetti sul software sia stata causata da nuove leggi tendenti ad ammorbidire i criteri di cessione delle licenze sul software, rafforzando contemporaneamente i dispositivi per farle rispettare. Affermano inoltre che il maggiore uso di brevetti viene fatto da grandi aziende, non necessariamente produttrici di software, perseguendo finalità strategiche, con l'obiettivo di procurarsi vantaggi sui concorrenti. In particolar modo si oppongono all'idea che una più vasta applicabilità dei brevetti sul software possa portare ad un maggior numero di investimenti in Ricerca e Sviluppo. Al contrario, sostengono che il largo utilizzo dei brevetti a scopo strategico rallenti l'innovazione.

Dallo studio di Tang, Adams e Parè<sup>32</sup> emerge che la maggior parte delle piccole e medie imprese preferiscono affidarsi, per proteggere i loro lavori, al semplice copyright, in aggiunta a sistemi di protezione digitali e a strategie ben congegnate di penetrazione all'interno del mercato. Il vasto progetto britannico sulla proprietà intellettuale<sup>33</sup>, conclude che i brevetti sul software non sono utili ad incentivare l'innovazione, essendo in gran parte utilizzate dalle grandi imprese manifatturiere. Le piccole e medie imprese preferiscono affidarsi ad altri metodi di protezione, più informali, tendendo a privilegiare tempestività di ingresso nel mercato e rapporto con la clientela.

Se l'impatto dell'introduzione di brevetti sul software sembra sollevare delle ambiguità al riguardo della sua utilità nello stimolare l'innovazione nel mondo del software proprietario, per quanto riguarda il software libero/open source gli effetti sono ancora più dannosi.

---

<sup>32</sup> P. TANG, J. ADAMS, D. PARE<sup>2</sup>, *Patent protection of computer programmes. Final report.*, ECSC-EC-EAEC, 2001, Brussels. Reperibile in: <http://swpat.ffii.org/vreji/minra/sisku/>

<sup>33</sup> INTELLECTUAL PROPERTY INITIATIVE, *Background and Overview of the Intellectual Property Initiative*, 2000. Reperibile in: <http://info.sm.umist.ac.uk/esrcip/background.htm>

Il pericolo per il software libero/open source sta nella stessa apertura del codice sorgente: la sua visibilità permette ai detentori di brevetti di localizzare in maniera semplice le tecniche utilizzate all'interno del codice. Inoltre, essendo il codice scritto cooperativamente da migliaia di sviluppatori, è sempre presente il rischio che aziende interessate ad un rallentamento nello sviluppo di un determinato programma possano inserire al suo interno parti di codice coperte da brevetto.

Se la situazione era preoccupante negli Stati Uniti, dove il software è brevettabile, problemi nuovi stanno sorgendo anche in Europa, dove fino al 2003 non si potevano richiedere brevetti sul puro software, ma soltanto su invenzioni che utilizzavano anche del software per funzionare.

Recentemente, in contrasto con le direttive del Parlamento Europeo, il Consiglio dei Ministri europeo sta attuando delle proposte per l'estensione della brevettabilità del software ad una più vasta gamma di programmi.

In una situazione europea in cui l'industria del software è spinta da imprese piccole e medie e dove l'open source si sta qualificando come un modello economico in crescita, queste posizioni sembrano controproducenti per la stessa economia dell'Unione.

Un'altra minaccia al modello open source arriva dall'esterno, incarnata da strategie di "marketing-guerriglia" tese a screditare l'immagine del software open source. Una delle tecniche più utilizzate è quella di associare il movimento free software/open source con gli hackers, usando il termine nell'accezione di "pirata informatico". Spesso si è tentato di screditare il software libero/open source associandolo a caratteristiche di bassa affidabilità e sicurezza, indicandolo come software non professionale.

Alcuni critici del modello free software/open source indicano poi tra i problemi legati a questo tipo di software la possibilità di una frammentazione in una varietà di software

incompatibili tra loro. Si intravede nel software libero/open source la possibilità di una “balcanizzazione” simile a quella che ha interessato i sistemi Unix nella seconda metà degli anni ottanta, in cui si assisteva alla proliferazione di almeno una cinquantina di sistemi Unix spesso incompatibili gli uni con gli altri. Ripercorrendo la storia dei sistemi Unix possiamo però scorgere diverse fasi la cui analisi potrebbe gettare luce su questo problema. Ricordiamo innanzi tutto che Unix, a partire dagli anni della sua nascita nei Bell Laboratories negli anni settanta, veniva rilasciato insieme ai codici sorgenti per ragioni legate ad una causa in corso nei confronti della Bell per violazione della legislazione statunitense sui monopoli. E’ stato proprio questo fattore a stimolare una larga adozione di Unix nei centri di ricerca e a garantire l’ampia portabilità di Unix su macchine di diverso tipo. La disponibilità dei sorgenti del sistema operativo contribuì a dare vita ad una relazione collaborativa tra i centri di ricerca che adottavano Unix, attraverso lo scambio delle patch e la condivisione delle nuove funzionalità. Con la prima commercializzazione di Unix, iniziata nella prima metà degli anni ottanta, vennero alla luce numerosissime versioni proprietarie del sistema operativo. Cercando di ottenere un vantaggio competitivo rispetto alle distribuzioni concorrenti, ogni sistema Unix proprietario cercava di tenere nascoste le proprie innovazioni, portando alla creazione di numerosi sistemi Unix divergenti tra loro. Si tratta del meccanismo opposto a quello sottostante allo sviluppo di Linux: quando una distribuzione introduce delle nuove soluzioni, tutte le altre distribuzioni saranno portate ad adottarla se queste incontrano popolarità e sono legalmente autorizzate a farlo. Il modello free software/open source “crea una pressione unificante verso un punto di riferimento

comune di fatto, uno standard aperto e rimuove le barriere della proprietà intellettuale che altrimenti ostacolerebbero questa convergenza<sup>34</sup>.

### 3.4 Sostenibilità economica del modello

Per anni il software libero è stato guardato dal mondo delle imprese con occhi particolarmente scettici. Sembrava assurdo che un modello basato sul contributo volontario di soggetti potesse raggiungere un livello qualitativo professionale e ancora più assurda pareva l'idea che una qualche azienda potesse guadagnare rendendo liberi i propri programmi. La Open Source Initiative ha avuto proprio il merito di avvicinare le imprese a questo modello, mostrando loro come potesse rivelarsi conveniente in diverse situazioni. Oggi imprese come Cygnus, basata su servizi di consulenza e assistenza su software libero/open source, Red Hat, azienda che commercializza la sua distribuzione di Linux e offre servizi su di essa, e Suse, una distribuzione tedesca, godono di ottima salute sul mercato. Aziende come IBM, HP, Corel hanno lanciato diversi progetti open source per soddisfare alcuni loro interessi commerciali. Netscape ha liberato il codice sorgente del suo Communicator quando la diffusione di Explorer guidata da Microsoft aveva ristretto drammaticamente la sua quota di mercato.

Green sostiene che il modello sia perfettamente spiegabile dall'analisi economica:

*The economics of Open Source software do not need any concepts other than those supplied by free market economies<sup>35</sup>.*

---

<sup>34</sup> R. YOUNG, *Regalato! Come Red Hat Software si trovò tra le mani un nuovo modello economico e contribuì a migliorare un'industria*, in C. DI BONA, S. OCKMAN, M. STONE, *Open Sources*, cit., p.134.

<sup>35</sup> E. L. GREEN, *Economics of Open Source Software*. Reperibile in: <http://badtux.org/home/eric/editorial/economics.php>

Bonaccorsi e Rossi<sup>36</sup> cercano di sfatare l'apparente assurdit  di un modello che si sviluppa in un'industria caratterizzata da un'economia di scala sia da parte dell'offerta sia dal lato della domanda (la cosiddetta *esternalit  di network*<sup>37</sup>).

La diffusione del modello in un contesto dominato dalla Microsoft, almeno nel mercato dei desktop, pu  essere spiegata attraverso un fenomeno di *esternalit  locale*<sup>38</sup>: un utente di un programma non   tanto interessato al numero di utilizzatori complessivi di un programma quanto al numero degli utilizzatori all'interno della sua cerchia di conoscenze. Prova di questo fatto   la presenza forte nel modello di una tecnica di pubblicizzazione peculiare: l'*advocacy*<sup>39</sup>, strategia basata sulla fiducia interpersonale, in cui un'utilizzatore entusiasta del software libero introduce un conoscente in questo mondo, trasformandolo a sua volta in un potenziale "avvocato".

Comunque, non dobbiamo cadere nell'errore di considerare il modello economico del software libero/open source come antagonista e in opposizione con il classico modello commerciale. Esso, infatti, non   un modello ottimale per tutte le aziende e la scelta di inversione di strategia non   priva di rischi e va valutata caso per caso.

In particolare il modello sembra essere ottimale per le piccole e medie imprese dotate di risorse limitate sia sul piano degli investimenti che su quello del personale, che si trovano a competere con imprese pi  grandi. Rendere open source il proprio software pu  voler dire per queste aziende beneficiare del contributo gratuito di bravi

---

<sup>36</sup> A. BONACCORSI-C. ROSSI, *L'Economia degli Standard e la Diffusione delle Tecnologie. L' Open Source non   un Assurdo Economico*, Pisa, Laboratorio di Economia e Management. Scuola di Studi Superiori Sant'Anna, 2001.

<sup>37</sup> M. L. KATZ-C. SHAPIRO, *Network Externalities, Competition and Compatibility*, in *American Economic Review*, 1985. Con il termine si indica il fenomeno per cui un utilizzatore trae maggiore valore da un bene quanto pi  questo si diffonde.

<sup>38</sup> J. M. DALLE – N. JULLIEN, *NT vs Linux or Some Explanation into Economics of Free Software*, saggio presentato a "Applied Evolutionary Economics", Grenoble 1999, cit. in A. BONACCORSI – C. ROSSI, *L'Economia degli Standard e la Diffusione delle Tecnologie. L' Open Source non   un Assurdo Economico*.

<sup>39</sup> R. C. PAVLICEK, *Keys to Effective Linux Advocacy Within Your Organization*, 1999. Reperibile in: [http://opensource.hp.com/the\\_source/linux\\_papers/als-fullpaper-1999.html](http://opensource.hp.com/the_source/linux_papers/als-fullpaper-1999.html).

programmatori di ogni parte del mondo, rendendo possibile il supporto per le versioni sorpassate così come la creazione di nuovi prodotti di qualità. L'adozione del modello può anche essere utile per stimolare i propri dipendenti, inserendoli in una comunità informale di programmatori, caratterizzata da un'approccio alla programmazione creativo e anti-burocratico.

I motivi che spingono un'azienda, un'organizzazione o un privato a lanciare un progetto open source sono diversi, ma tutti "economici" a loro modo.

Come sottolinea Green:

*The "market" is a handy short cut for the basis of free market economies: the supply and demand of a particular good or service. Note that the concept of "market" does not necessarily involve the concept of "money". A market merely requires that there be a demand for a product, a supply of that product and some medium of exchange which may be yet another good or service<sup>40</sup>.*

Sono infatti numerosi i benefici che può trarre un'azienda dal partecipare ad un progetto open source.

In primo luogo, la libera distribuzione del codice sorgente può essere considerata un valore aggiunto al prodotto, nella misura in cui la sua disponibilità può essere utile al cliente per risolvere alcune sue problematiche.

Un'azienda o un privato che sceglie di utilizzare un prodotto open source, ad esempio, è rassicurata dal fatto che può sempre rivolgersi a terzi se l'azienda con cui aveva stipulato il contratto fallisce o fa un cattivo lavoro. Questa possibilità permette di mettersi al sicuro dagli altissimi costi necessari per migrare da un software ad un altro, legati alla necessità di cambiare le procedure organizzative dell'azienda se queste sono

---

<sup>40</sup> E. L. GREEN, *Economics of Open Source Software*, cit., p.2.

guidate da un software e alla formazione dei dipendenti, abituati ad utilizzare lo stesso strumento da parecchi anni. Un altro vantaggio decisivo garantito dalla disponibilità dei sorgenti è la possibilità di adattare lo stesso software a nuove piattaforme hardware, consentendo all'azienda di aggiornare la tecnologia senza cambiare gli strumenti in uso. L'adozione di un prodotto open source può inoltre garantire il beneficio di disporre di una comunità di sviluppatori volontari che costantemente testano il programma, lo ripuliscono dai bachi, creano nuove funzionalità e lo tengono aggiornato.

Se il codice sorgente rappresenta un valore aggiunto per i clienti, capace di spingerli all'adozione di un determinato prodotto, è necessario capire come una azienda di software possa riuscire a trarre profitto dai clienti raggiunti.

La Open Source Initiative indica quattro modelli principali per guadagnare o trarre vantaggio dal software libero/open source: "Support Sellers", "Loss Leader", "Widget Frosting" e "Accessorizing"<sup>41</sup>.

- Il modello chiamato "Support Sellers" consiste nella vendita di servizi sul software, che viene ceduto gratuitamente. I servizi comprendono in genere supporto generico per l'installazione e la manutenzione, consulenza, formazione dei dipendenti, *customizzazione*<sup>42</sup> del software e vendita di documentazione in formato cartaceo.

Per la riuscita di una attività basata su questo modello gioca un'importanza fondamentale la capacità dell'azienda di imporre il proprio nome attraverso adeguate strategie di marketing. La competizione tra le aziende che si occupano di servizi sul software è spesso basata sull'affidabilità del servizio offerto e sul prezzo. Aziende di questo tipo, come Red Hat, spesso offrono delle distribuzioni

---

<sup>41</sup> OPEN SOURCE INITIATIVE, *Open Source Case for Business*, 2004. Reperibile in: [http://www.opensource.org/advocacy/case\\_for\\_business.php](http://www.opensource.org/advocacy/case_for_business.php)

<sup>42</sup> Personalizzazione e adattamento del software alle esigenze del cliente.

dei programmi studiate per essere più semplici da utilizzare per il cliente. Storicamente, la prima compagnia che ha preso questa strada è stata Cygnus Solution, che offriva supporto per software GNU.

- La strategia “Loss Leader” prevede il rilascio di un software sotto licenza open source come “apri-pista” per la vendita di altro software sottoposto a copyright. Il codice ottenuto attraverso il lavoro della comunità di sviluppatori può essere poi utilizzato dalla compagnia per migliorare la stabilità e la funzionalità del loro software commerciale. Questa strategia è utile anche al fine di rafforzare la reputazione della compagnia, far conoscere il nome dell’azienda e abituare gli utenti all’intera linea software dell’azienda. Per l’attuazione di questo modello è importante la scelta della licenza. L’utilizzo di licenze di tipo copyleft (come la GPL) va evitato perché richiede che il codice derivato sia nuovamente distribuito sotto la stessa licenza libera.

La viabilità di questo modello è stata testata per la prima volta da Netscape.

- Il modello chiamato “Widget Frosting”, invece, è tipicamente utilizzato da compagnie che vendono principalmente hardware. I programmi rilasciati da queste compagnie sono generalmente drivers e codice di interfaccia. Rilasciando in versione aperta questo software, sono maggiori le possibilità di allargare la fascia d’utenza dei dispositivi hardware che lo utilizzano. Un altro motivo che spinge le aziende ad adottare questa strategia è la possibilità di veder migliorare le caratteristiche del proprio prodotto hardware attraverso una ottimizzazione e un adattamento del software alle sue caratteristiche.
- Con il termine “Accessorizing” si indica il modello utilizzato da aziende non direttamente coinvolte nella vendita di software o servizi su di esso ma che si

occupano di vendita di materiale legato al software, come libri e documentazione cartacea. Queste aziende sono interessate all'espansione del fenomeno per poter guadagnare sulla vendita di accessori.

Oltre a questi modelli principali, indicati programmaticamente dalla Open Source Initiative, Frank Hecker<sup>43</sup> ne individua altri:

- “Service Enabler”: in questo caso il programma che viene rilasciato sotto licenza libera serve come supporto ad un servizio web attraverso il quale l'azienda guadagna tramite pubblicità o quote di registrazione al servizio. Lo sviluppo libero garantisce una continua personalizzazione e aggiornamento del software, contribuendo a rendere allo stesso tempo più interessante e utile il servizio web offerto.
- Il modello “Sell it, Free It” è composto da due fasi distinte: in un primo momento l'azienda distribuisce il software in forma proprietaria, per poi “liberarlo” in una seconda fase quando i benefici derivati da questa scelta superano gli introiti derivanti dalla vendita delle licenze.
- “Brand Licensing”: un'azienda rende open source il suo software detenendo però i diritti sul nome del prodotto e sui marchi registrati. Le aziende intenzionate a creare versioni derivate del software usando questi marchi registrati devono pagare i diritti all'azienda che li detiene.
- Il modello “Software Franchising” trae ispirazione dal modello precedente. Il software è distribuito in forma libera, ma l'azienda guadagna autorizzando, previo pagamento dei diritti, altri sviluppatori a creare aziende di consulenza e supporto sul software utilizzando il suo nome.

---

<sup>43</sup> F. HECKER, *Setting Up Shop: The Business of Open-Source Software*, pp. 22-25, 2000. Reperibile in: <http://www.hecker.org/writings/setting-up-shop.html>.

In linea generale, comunque, bisogna notare che la via dell'open source non rappresenta la strada migliore per tutte le aziende di software. L'adozione di un modello di distribuzione open source sembra essere particolarmente indicata per compagnie che detengono quote di mercato non molto alte e che vogliono allargare il loro bacino di utenza. In particolare il modello open source fornisce una possibilità in più per le piccole aziende che spesso si trovano in difficoltà a contrastare le risorse economiche, umane e di *know-how* delle grandi compagnie. Attraverso il modello open source, le piccole aziende non si trovano più sole a contrastare le grandi, ma possono contare sulla collaborazione con altre realtà. Infatti,

*All of them share the same code base(...),and whenever one of them fixes a bug, or improves a package, all its competitors have access to the source code, and thus to the same fix or improvement. Therefore, projects and companies in the open source world are compelled to collaborate by the open source model<sup>44</sup>.*

Questa caratteristica sembra essere vitale soprattutto per le economie arretrate nel campo dell' ICT (Information and Communication Technology) e può dare un notevole contributo alla riduzione del Digital Divide. Le aziende produttrici di software proprietario infatti, che si trovano a spendere cifre altissime per pagare i programmatori e per varare un progetto software, possono decidere di non investire tempo e soldi per implementare funzioni utili soltanto a paesi dove non si può contare nella presenza di economie di scala abbastanza robuste.

---

<sup>44</sup> WORKING GROUP ON LIBRE SOFTWARE, *Free Software/Open Source: Information Society Opportunities for Europe?*, p. 14, Version 1.2, 2000. Reperibile in: <http://eu.conecta.it/paper/paper.html>.

Attraverso il modello open source si può contare su una diffusione maggiore e meglio distribuita dell'ICT, fattore che nel tempo produrrà la nascita di nuove sfere di mercato non strettamente legate al campo informatico.

## 4. Il modello di sviluppo

### 4.1 Un nuovo modo di sviluppare software

La crescita del fenomeno del software libero/open source e in particolare la creazione di programmi di qualità nati dalla collaborazione remota e decentralizzata di sviluppatori di diversi paesi ha rovesciato alcuni assunti classici dell'ingegneria informatica, secondo i quali il software di qualità sarebbe ottenibile soltanto attraverso una gestione centralizzata dei compiti dei programmatori e un rigido controllo dell'accesso ai sorgenti.

Uno dei primi tentativi di far luce sulle caratteristiche di questo modello è rintracciabile nel saggio di Raymond *“La cattedrale e il bazaar”*<sup>45</sup>. Il sistema tradizionale di sviluppo del software viene qui paragonato allo stile di costruzione delle cattedrali medioevali, caratterizzato da un forte controllo centrale, in contrapposizione allo “stile di sviluppo a bazaar”, in cui la divisione dei compiti viene effettuata attraverso comunicazioni informali tra programmatori e buona parte del codice è costituito da piccoli moduli indipendenti sviluppati spontaneamente dagli interessati.

E' necessario sfatare il mito che tutto il software libero sia basato su questo modello così come l'illusione che esso derivi in linea diretta dalla strategia di licenziamento “libera”.

Allo stesso tempo bisogna notare che i progetti open source di maggiore successo sono scaturiti proprio dall'applicazione di questo modello, in ogni caso inapplicabile nei progetti proprietari.

---

<sup>45</sup> E. S. RAYMOND *La cattedrale e il bazaar*, ed. it., 1998. Reperibile in: <http://www.apogeeonline.com/openpress/doc/cathedral.html>

Attraverso un'attenta analisi dello sviluppo di Fetchmail (un programma di gestione della posta elettronica) e basandosi sull'esperienza di Linus Torvalds nella creazione del kernel Linux, Raymond evidenzia alcune caratteristiche interessanti dello stile "bazaar". Molti progetti open source nascono dalla necessità personale di un programmatore, in cerca di una particolare funzionalità per svolgere alcuni compiti. Diversi studi pongono l'accento sul desiderio di risolvere un particolare problema di programmazione come molla scatenante di un progetto.

Il punto di partenza generalmente è costituito da una certa quantità di codice già disponibile ma sporco o incompleto, che costituisce la base su cui iniziare a lavorare.

Questa base di codice deve essere portata dall'iniziatore del progetto ad un livello base di usabilità, in modo da attirare utenti desiderosi di vedere crescere il programma.

E' qui importante far notare che storicamente i sistemi basati su Unix sono utilizzati prevalentemente da programmatori e tecnici informatici. Su questi sistemi ogni utente è un potenziale co-sviluppatore.

Analizzando la strategia di coordinamento di Torvalds, Raymond fa notare come il rapido successo di Linux sia dipeso dalla tendenza a trattare tutti i suoi utenti come co-sviluppatori, rilasciando nuove *release*<sup>46</sup> a ritmo sostenuto, in modo da incoraggiarli a rispedire con frequenza le *patch*<sup>47</sup> di debug. Gli utenti venivano ricompensati dalla soddisfazione di prendere parte ad una azione collettiva che produceva risultati costanti. Non venivano coinvolti soltanto gli sviluppatori veri e propri ma anche gli utenti che segnalavano dei bug o che fornivano dei semplici suggerimenti.

---

<sup>46</sup> Una nuova versione del programma.

<sup>47</sup> Termine utilizzato per indicare una porzione di grandezza variabile di codice.

Le possibilità di successo di un progetto dipendono dalla capacità del leader-manutentore di ascoltare i propri utenti e di farli sentire importanti rispetto alla buona riuscita del progetto.

Uno stile di sviluppo di questo tipo è reso possibile dalle caratteristiche del medium Internet, adatto alla collaborazione volontaria, aperta, distribuita e flessibile.

Gli strumenti primariamente utilizzati dagli sviluppatori per comunicare tra loro sono molteplici: mailing list, forum, chat, sistemi di file-sharing per lo scambio di documenti e sistemi di telefonia via Internet.

La capacità di un progetto di attirare intorno a sé un grande numero di utenti co-sviluppatori dipende anche dalla struttura iniziale che viene data al codice del programma.

*Il kernel Linux ha subito mostrato chiaramente che quanto ci serve è un sistema il più possibile modulare. Il modello di sviluppo open source lo vuole, per non avere gente che lavori simultaneamente sugli stessi punti<sup>48</sup>.*

La modularità permette una ottimale distribuzione dei compiti ed è fondamentale per assicurare ad ogni potenziale sviluppatore la possibilità di poter lavorare su ciò che più lo entusiasma senza interferire su altre parti di codice.

La suddivisione in moduli permette non solo l'aggiunta di nuove funzionalità senza interferire sull'intero sistema, ma anche la ripartizione del lavoro tra più sviluppatori.

Baldwin e Clark<sup>49</sup> considerano anche le proprietà "opzionali" dei moduli, ovvero la capacità di ogni modulo di supportare un nuovo design senza essere obbligato ad adottarlo. L'opzionalità genera una situazione in cui ogni contributo (anche quando non

---

<sup>48</sup> L. TORVALDS, *Il vantaggio di Linux*, in *Open Sources. Voci dalla rivoluzione Open Source*, a cura di Ch. Di Bona - S. Hockman - M. Stone, ed. it., Milano, Apogeo 1999, p. 118

<sup>49</sup>C.Y. BALDWIN, K.B. CLARK, *The Architecture of Cooperation: How Code Architecture Mitigates Free Riding in the Open Source Development Model*, Harvard Business School 2003. Reperibile in: <http://opensource.mit.edu/papers/baldwinclark.pdf>

verrà adottato) accresce il valore dell'intero sistema senza togliere valore agli altri contributi individuali.

Se da un lato la modularità permette l'aggiunta di nuove funzionalità e la sperimentazione senza rischi, da un'altro lato, come fa notare Bezroukov<sup>50</sup>, essa può portare ad un approccio conservatore dell'architettura di un programma già sviluppato.

Il grande successo di alcuni progetti open source e in particolare il grande numero di sviluppatori coinvolti in alcuni di questi (uno per tutti il kernel Linux) hanno contribuito a diffondere l'idea che il software libero sia sempre creato da una vasta comunità di programmatori.

Questa concezione, seppur capace di descrivere ottimamente alcuni progetti, non riflette il reale processo di creazione di un gran numero di programmi rilasciati sotto licenze libere.

Krishnamurthy<sup>51</sup>, in uno studio basato sull'analisi di 100 progetti open source, rileva come gran parte di essi siano stati guidati da un numero ridotto di partecipanti, tanto da suggerire la sostituzione del termine comunità, sovente usato per indicare l'insieme degli sviluppatori di un particolare programma, con quello di associazione volontaria.

Dobbiamo però notare come il processo collettivo di creazione del software open source non venga alimentato solamente dagli sviluppatori che contribuiscono concretamente attraverso donazioni di codice: parte integrante del lavoro è svolto da un numero maggiore di persone che si occupano di fissare i banchi e da un numero ancora più ampio di utenti che li individuano.

---

<sup>50</sup> N. BEZROUKOV, *Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism)*, 1999 First Monday, 04 (11). Reperibile in: [http://www.firstmonday.dk/issues/issue4\\_10/bezroukov/](http://www.firstmonday.dk/issues/issue4_10/bezroukov/).

<sup>51</sup> S. KRISHNAMURTHY, *Cave or Community? An Empirical Examination of 100 Mature Open Source Projects*, First Monday, volume 7, number 6 (June 2002), Reperibile in: [http://firstmonday.org/issues/issue7\\_6/krishnamurthy](http://firstmonday.org/issues/issue7_6/krishnamurthy)

Lo studio di Mockus, Fielding e Herbsled<sup>52</sup>, incentrato sul processo di sviluppo del web server Apache e del browser Mozilla, indica come le persone che fissano i banchi siano di un ordine di grandezza superiore rispetto ai componenti del team principale e quelle che li individuano siano di un ordine di grandezza superiore ai primi.

Una visione soddisfacente del processo di sviluppo open source si può ottenere soltanto includendo nell'oggetto dell'osservazione tutti i suoi partecipanti, cercando di definirne i rispettivi ruoli e gettando luce sul grado di mobilità esistente tra i differenti livelli di coinvolgimento.

## 4.2 I partecipanti

I dati statistici più attendibili sulle caratteristiche degli sviluppatori di software open source sono stati raccolti da tre grandi studi:

- *Who is doing it? A research on Libre Software Developers (WIDI)*<sup>53</sup>.
- *Free/Libre and Open Source Software: Survey and Study (FLOSS)*<sup>54</sup>.
- *The Boston Consulting Group Hacker Survey (BCG/OSDN)*<sup>55</sup>.

Dalla comparazione dei risultati dei tre studi emergono le seguenti caratteristiche:

*Genere*: tutti gli studi visionati concordano nell'attribuire uno scarso apporto da parte di sviluppatori di genere femminile. Il 98% dei partecipanti ai questionari è costituito maschi.

---

<sup>52</sup> A. MOCKUS, R.T. FIELDING, J.D. HERBLEB, *Two Case Studies of Open Source Software Development: Apache and Mozilla*, Avaya Labs Research 2002. Reperibile in: <http://www.research.avayalabs.com/techreport/ALR-2002-003-paper.pdf>

<sup>53</sup> G. ROBLES, E. SHEIDER, I. TRETOKOWSKY, N. WEBER, *Who is doing it?(WIDI) A research on Libre Software Developers*. Fachgebiet für Informatik und Gesellschaft, TU-Berlin 2001. Disponibile in: <http://widi.berlios.de>

<sup>54</sup> R. A. GOSH, R. GLOTT, B. KRIEGER, G. ROBLES, *Free/Libre and Open Source Software: Survey and Study (FLOSS) Part 4: Survey of Developers*, International Institute of Infonomics, University of Maastricht, 2002. Disponibile in: <http://www.infonomics.nl/FLOSS/report/>

<sup>55</sup> J. BATES, K. R. LAKHANI, B. WOLF, *The Boston Consulting Group Hacker Survey*. The Boston Consulting Group 2002. Disponibile in: <http://www.bcg.com/opensource/BCGHACKERSURVEY.pdf>

*Età:* gran parte (70%) dei partecipanti allo sviluppo di programmi open source hanno un'età compresa tra i 22 e i 37 anni. L'età media è di 27 anni.

*Concentrazione geografica:* più dell'80% dei partecipanti vivono negli Stati Uniti o in Europa. I partecipanti che vivono in Europa sono circa il 40%, così come quelli che vivono negli Stati Uniti. In Europa il numero maggiore di partecipanti si riscontra in Germania.

*Educazione e professione:* l'80% dei partecipanti è coinvolto nel campo dell'ICT. Tra questi un terzo è costituito da ingegneri informatici. Il secondo posto è occupato da studenti del ramo dell'ICT, seguiti da programmatori e consulenti informatici.

Gran parte dei partecipanti possiede un impiego. I disoccupati costituiscono una componente irrisoria se includiamo tra i lavoratori anche gli studenti che svolgono un lavoro non pagato.

Secondo WIDI<sup>56</sup>, il 20% degli intervistati viene pagato per sviluppare software libero. E' interessante notare che un numero simile di intervistati è stato pagato in precedenza per svolgere questo compito e un terzo di quelli che non sono mai stati pagati conta di poterne beneficiare in futuro.

L'università occupa un posto speciale nel panorama del software libero/open source: secondo WIDI<sup>57</sup> un terzo degli sviluppatori è collegato all'università.

Il livello di istruzione dei partecipanti è elevato: un numero superiore al 50% è in possesso di un attestato universitario, mentre il 15% possiede un attestato post-universitario.

*Coinvolgimento:* il software libero/open source viene sviluppato principalmente con un impegno part-time da parte dei partecipanti. Più del 50% dei partecipanti dedica al

---

<sup>56</sup> G. ROBLES, E. SHEIDER, I. TRETOKOWSKY, N. WEBER, *Who is doing it? (WIDI)* cit., p.33.

<sup>57</sup> Ivi, p.32.

software open source meno di 10 ore settimanali. FLOSS<sup>58</sup> mette in luce il fatto che gli sviluppatori che si dedicano per più di 40 ore settimanali al software open source sono i più anziani. Un'analisi ulteriore indica come un numero elevato di ore di partecipazione sia correlato con un'alta competenza nel campo dell'ICT, rivelando come il forte interesse in questo campo sia un importante fattore di stimolo alla partecipazione.

### 4.3 Le motivazioni dei partecipanti

Due degli studi sopra citati, FLOSS<sup>59</sup> e BCG/OSDN<sup>60</sup>, indagano le motivazioni che spingono gli sviluppatori a partecipare ad un progetto open source.

Secondo entrambi gli studi gli intervistati indicano come ragione principale della partecipazione ad un progetto open source la possibilità di apprendere nuove abilità e espandere e condividere le proprie conoscenze.

Discreta importanza, anche se meno marcata, viene attribuita all'opportunità di partecipare ad una nuova forma di cooperazione, alla convinzione che il software non debba essere un bene proprietario e alla necessità di risolvere un problema tecnico non disponibile con le applicazioni già esistenti.

Gran parte degli intervistati considera la partecipazione ad un progetto open source un'attività creativa e stimolante.

I benefici derivanti dalla partecipazione sono principalmente l'accrescimento delle proprie basi di conoscenza e il senso di gratificazione personale derivante dall'aver contribuito alla crescita del progetto.

---

<sup>58</sup> R. A. GOSH, R. GLOTT, B. KRIEGER, G. ROBLES, *Free/Libre and Open Source Software: Survey and Study (FLOSS) Part 4: Survey of Developers* cit., p. 24.

<sup>59</sup> Ivi.

<sup>60</sup> J. BATES, K. R. LAKHANI, B. WOLF, *The Boston Consulting Group Hacker Survey*.

Il tema della motivazione degli sviluppatori di software libero/open source è stato analizzato da vari studiosi del fenomeno, attraverso approcci differenti.

La psicologia motivazionale offre uno schema attraverso il quale classificare le differenti motivazioni degli sviluppatori.

Deci e Ryan<sup>61</sup> distinguono nel campo della motivazione umana *motivazioni intrinseche* e *motivazioni estrinseche*. Le prime vengono così definite:

*...the doing of an activity for its inherent satisfactions rather than for some separable consequence.*<sup>62</sup>

Riguardano attività che vengono svolte indipendentemente da fattori esterni, soltanto per la gioia di farlo, per accrescere in qualche modo la propria autostima.

Le altre invece riguardano le attività motivate dalla speranza o dalla certezza di ottenere in cambio una ricompensa di qualsiasi tipo oppure per adempiere a degli obblighi.

Gli sviluppatori di software libero/open source sono spinti, all'interno di questa struttura, da entrambi i tipi di motivazioni.

Una forte rilevanza sembrano avere le motivazioni intrinseche basate sul puro divertimento, che spingono gli sviluppatori a contribuire per soddisfare la curiosità intellettuale, mettere alla prova le proprie capacità e confrontarsi con nuovi problemi.

L'importanza di questa istanza è confermata dalle stesse origini del software libero<sup>63</sup>, che indicano come la curiosità e il desiderio intellettuale di scoperta rappresentassero dei grossi stimoli per i primi hackers.

Anche se lo stimolo intellettuale e la passione per la programmazione rappresentano senza dubbio una ragione fondante della partecipazione ai progetti open source, non

---

<sup>61</sup> E. L. DECI, R. M. RYAN, *Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions*, in *Contemporary Educational Psychology*, January 2000, vol. 25, no. 1, pp. 54-67 Elsevier Science. Reperibile in:

<http://www.psych.rochester.edu/SDT/publications/documents/2000RyanDeciIntExtDefs.pdf>

<sup>62</sup> Ivi, p. 56.

<sup>63</sup> Vedi Capitolo 2: Storia ragionata del software libero/open source.

sono capaci di spiegare cosa spinga i programmatori a condividere le proprie realizzazioni e a preservare l'apertura del proprio codice.

Alcune risposte possono essere trovate nella categoria delle motivazioni intrinseche basate sullo spirito di appartenenza ad una comunità: i programmatori contribuirebbero per il piacere personale di partecipare ad un progetto condiviso e soddisfare così un bisogno di relazione. A fornire una base per la creazione di uno spirito comunitario intervengono convinzioni etiche basate, nel caso del software libero/open source, sull'idea che il codice rappresenti un veicolo informativo e che la sua consultazione debba essere disponibile a tutti.

Il rispetto di tali motivazioni etiche viene assicurato attraverso l'applicazione delle licenze. Nella prospettiva di Camp e Syme<sup>64</sup>, codice e licenze forniscono una struttura attraverso la quale governare l'informazione. Inoltre, differenti licenze sottendono a differenti visioni della società. Le licenze permissive garantiscono ad ogni partecipante il diritto di non vedere sprecato il proprio sforzo nei confronti della comunità. In poche parole rendono più semplice l'atto del donare.

Raymond<sup>65</sup> attribuisce grande importanza a questo genere di motivazioni intrinseche e assume che il loro attualizzarsi sia da attribuirsi ad una forma di economia del dono, piuttosto che dello scambio. Descrivendo la cultura open source come un'adattamento all'abbondanza, favorita dalla sempre più ampia disponibilità di spazio su disco rigido e di ampiezza di banda, individua nel dono lo strumento attraverso il quale si attualizza il gioco della reputazione tra i pari.

---

<sup>64</sup> L. J. CAMP, S. SYME, *The Governance of Code: Code as Governance*, Social Science Research Network Electronic Papers Collection, 2001. Reperibile in: <http://papers.ssrn.com/abstract=297154>.

<sup>65</sup> E. S. RAYMOND, *Colonizzare la noosfera*, Open Press 1998. Reperibile in: <http://www.apogeeonline.com/openpress/doc/homesteading.html>

L'importanza delle motivazioni intrinseche basate sul senso di appartenenza ad una comunità è attestata dalla crescente tendenza attuale, favorita dalle reti di comunicazione distribuite, a costituire aggregazioni basate su principi etici e relazionali, in reazione ad una predominante tendenza all'atomizzazione e all'assottigliamento dei legami sociali.

*... la comunità virtuale rappresenta, in una società fortemente tecnologica, differenziata e segmentata, segnata dal valore "mercato", il ritorno di un'esperienza di socialità più vicina alla dimensione dell'agire solidaristico, un ritorno a modalità/opportunità di ricostruzione di legami sociali ed intersoggettivi non utilitaristici.<sup>66</sup>*

Oltre alle motivazioni intrinseche, che sembrano ricoprire molta importanza nel mondo del software libero, dobbiamo segnalare anche diversi tipi di motivazioni estrinseche.

Segnalata già nel celebre saggio di Raymond, particolare importanza sembra rivestire la motivazione nota come "*scratching an itch*"<sup>67</sup>. Molti progetti open source nascono dalla necessità di qualche programmatore di risolvere un problema tecnico personale o dalla curiosità di affrontare un nuovo problema di programmazione. E' il caso del kernel Linux, nato dalla necessità di Torvalds di poter disporre di un sistema Unix sul proprio 386 e di molti altri progetti. Facciamo notare che le motivazioni degli sviluppatori nella partecipazione ad un progetto variano in relazione a diversi parametri: coinvolgimento in termini di tempo ed esperienza dei partecipanti, stato di sviluppo del progetto e possibilità di evoluzione dello stesso.

La necessità di una certa applicazione sembra essere un motivo scatenante in particolare per il lancio di un progetto. Gli sviluppatori sono invogliati a liberare il proprio codice

---

<sup>66</sup> A. C. FRESCHI, *La società dei saperi. Reti virtuali e partecipazione sociale*. Carocci, Urbino 2002.

<sup>67</sup> Può essere resa in italiano come "togliersi un tarlo".

dalla convinzione di poter disporre di un software funzionante lavorando meno, grazie all'aiuto dei co-sviluppatori.

Questa istanza è particolarmente importante se si pensa che molti degli sviluppatori open source provengono dal mondo accademico, ambito in cui l'innovazione e la ricerca occupano un posto di primo piano.

Attuando un'analisi economica delle motivazioni dei partecipanti, basata sul calcolo dei costi e dei benefici, Lerner e Tirole<sup>68</sup> suggeriscono che i partecipanti siano spinti a collaborare quando i benefici derivanti dall'esterno superino i loro costi (in particolare in termini di tempo ed energie impiegate). Tra i *benefici immediati*, oltre alla soddisfazione della necessità di nuove funzionalità, vi è la retribuzione.

Nonostante il carattere largamente volontaristico del fenomeno open source, sono molte le aziende che pagano dei programmatori per partecipare ad un progetto, in modo che esso raggiunga in breve uno stato di usabilità, per influenzarlo in alcune direzioni cruciali per l'azienda o per garantirsi un'alto *know-how* dei propri dipendenti su quel prodotto.

Tra i *benefici posticipati* troviamo un'istanza legata a quest'ultima: molti sviluppatori utilizzano il mondo dell'open source come vetrina per rendersi appetibili alle aziende di software. Luoghi virtuali di accentrimento come Sourceforge e Freshmeat, che collezionano le specifiche dei vari progetti open source, forniscono una rampa di lancio internazionale per i programmatori con più esperienza.

La possibilità di fare esperienza e migliorare le proprie competenze può essere considerato un ennesimo beneficio posticipato, con il quale guadagnarsi una retribuzione futura. L'ambiente di sviluppo open source offre la possibilità non soltanto

---

<sup>68</sup> J. LERNER, J. TIROLE, *The Simple Economics of Open Source*, Social Science Research Network Electronic Paper Collection, 2000. Reperibile in: <http://ssrn.com/abstract=224008>

di imparare a programmare, ma permette di introiettare uno stile di lavoro collaborativo e interattivo, adatto al lavoro in squadra, sempre più richiesto dalle aziende.

Nonostante i tentativi di diversi autori di ridurre le motivazioni dei partecipanti ad un unico comune denominatore, crediamo che nessun modello a sé stante riesca a rendere l'eterogeneità delle motivazioni degli sviluppatori, che variano sia per diversità di competenze tecniche degli stessi, per coinvolgimento all'interno dei singoli progetti e per anni di esperienza all'interno del mondo open source.

Le motivazioni, il tipo e la quantità dei contributi di un singolo partecipante variano nel tempo e sono collegati ad un processo di socializzazione all'interno della comunità.

In accordo con Tzouris<sup>69</sup>, crediamo che per comprendere appieno il fenomeno dell'open source sia necessario adottare un approccio multi-disciplinare.

L'analisi economica, il modello organizzativo delineato dalle licenze, l'analisi motivazionale e le teorie sull'economia del dono, contestualizzate all'interno di un ambiente digitale, si complementano tra loro nella spiegazione di un fenomeno complesso e per certi versi nuovo.

Il nucleo di questo fenomeno che tocca temi quali l'organizzazione in rete, la proprietà intellettuale e il dono va ricercato nel concetto (discusso e utopistico quanto utile) di comunità, che

*...si presenta come un concetto ponte tra i vari saperi incrociando differenti ambiti teorico/disciplinari quali la sociologia, la psicologia, la psichiatria, gli studi urbani, la politica<sup>70</sup>.*

---

<sup>69</sup> M. TZOURIS, *Software Freedom, Open Software and Participants Motivation. A Multidisciplinary study*. The London School of Economics and Political Science, 2002. Reperibile in: <http://opensource.mit.edu/papers/tzouris.pdf>

<sup>70</sup> B.R. GELLI, *Comunità: dalla metafora al concetto*, p.21, in B.R. GELLI (a cura di), *Comunità, rete, arcipelago. Metafore del vivere sociale*, Carocci, Roma 2002.

#### 4.4 Le comunità del movimento free software/open source

Prima di applicare il termine comunità al fenomeno open source dobbiamo fare una premessa che ci metta al riparo da interpretazioni errate.

Esso, infatti, come indicato da Bagnasco<sup>71</sup>, può nascondere delle trappole se si corre il rischio di considerarlo come un termine analitico piuttosto che come una metafora. Il suo uso nelle diverse scienze umane l'ha poi reso estremamente duttile e inclusivo, tanto da rappresentare contesti sociali molto differenti tra loro.

Nella sua accezione originaria, elaborata da Tönnies<sup>72</sup>, il termine si riferisce a realtà sociali basate sul territorio, caratterizzate da legami forti (primo fra tutti quello di sangue), non scelti ma al contrario già dati. Le comunità così definite si caratterizzano come insiemi relativamente chiusi, dove è molto difficile entrare e altrettanto difficile uscire e il divario tra chi vi appartiene e chi non vi appartiene è molto sentito.

Le comunità virtuali, categoria in cui si inserirebbero le comunità legate al software libero/open source, rappresentano un concetto per alcuni versi opposto a questo.

*La comunità virtuale, sostenuta dal computer e da e-mail, è una comunità che si crea artificialmente nella quale si può entrare e dalla quale uscire molto facilmente, dandosi regole a piacere, per fini scelti e contrattati, esattamente il contrario di quello che diceva Tönnies parlando della comunità<sup>73</sup>.*

D'altro canto, l'utilizzo del termine in svariate discipline ha contribuito a renderlo meno specifico, più duttile e flessibile. Proprio questa flessibilità lo rende adatto a descrivere un fenomeno nuovo, variegato e complesso come l'open source.

---

<sup>71</sup> A. BAGNASCO, *Il concetto di comunità: utilità e trappole*, pp. 37-42, in B. R. GELLI (a cura di), *Comunità, rete, arcipelago*, cit..

<sup>72</sup> F. TÖNNIES, *Comunità e società*, Ed. di Comunità, Bologna 1985.

<sup>73</sup> A. BAGNASCO, *Il concetto di comunità: utilità e trappole*, cit., p.40.

In un contesto sociale in cui i legami con il territorio tendono ad affievolirsi e le comunità tradizionali si sfaldano in molti soggetti autonomi e indipendenti, muta anche l'accezione del termine stesso.

Taurino propone la combinazione del concetto di comunità con quello di *rete*: le comunità non devono essere intese come gruppi sociali, compatti e omogenei, ma come

*...espressione di una specifica forma di rapporto sociale che, svincolato dall'ancoraggio al territorio, si realizza attraverso una struttura relazionale di tipo reticolare, fondata sull'azione del soggetto e sulla sua libertà di scelta e movimento<sup>74</sup>.*

La comunità post-moderna non si struttura rispetto al territorio e a rapporti ereditati e tradizionali ma rispetto a legami che vengono creati sulla base di affinità di pensiero e pratiche comuni.

Quello che rimane della precedente concezione è il *senso di comunità*, basato sulla percezione di un'identità comune, sulla reciprocità e sulla fiducia tra gli appartenenti.

Un ruolo chiave è occupato in questo contesto dalla condivisione di simboli, storie, linguaggi. Gli artefatti collettivi costituiscono il polo intorno al quale strutturare le comunità, in mancanza di rapporti faccia a faccia prolungati e in presenza di legami relativamente deboli, mutabili e non coercitivi. Nel caso delle comunità del free software/open source le norme che regolano le comunità sono in genere norme riguardanti gli artefatti e l'approccio da adottare nei loro confronti.

L'apertura delle comunità virtuali, diversamente dalle comunità tradizionali, dà vita a comunità multiple, dai confini incerti, che si strutturano su più livelli: la struttura reticolare sottostante permette un dialogo costante tra singole comunità (nel caso del

---

<sup>74</sup> A. TAURINO, *Comunità e networks sociali. L'importanza del costruito di rete nell'interpretazione della comunità come fenomeno relazionale*, p.108, in B. R. GELLI (a cura di), *Comunità, rete, arcipelago. Metafore del vivere sociale*, Carocci, Roma 2002.

software libero/open source quelle che si formano attorno ad un unico progetto), comunità immaginate, presenti più come idea che come reale interazione (nel nostro caso, la comunità del free software e la comunità open source) e comunità a base territoriale che si intersecano con le precedenti (si pensi ai gruppi di supporto agli utenti Linux, i Linux Users Groups, detti LUGs).

Proporremo un'analisi degli usi e delle norme che regolano le comunità del software libero/open source, descrivendone le consuetudini e i modi in cui i progetti nascono e si sviluppano e cercando di capire come avviene il processo di socializzazione tra i partecipanti, per poi situare concettualmente queste specifiche comunità all'interno del contesto più generico delle comunità virtuali.

Ogni progetto nasce dall'iniziativa di un singolo o di un gruppo di sviluppatori, che contribuiscono rilasciando una prima porzione di codice. Essi vengono considerati a tutti gli effetti proprietari del programma e denominati *manutentori*.

Come nota Raymond,

*...la cultura open source possiede una serie di usi e costumi sulla proprietà, elaborati ma essenzialmente non dichiarati. Essi stabiliscono a chi e in quali circostanze sia consentito modificare il software, e (soprattutto) chi abbia il diritto di ridistribuire nuovamente le versioni modificate alla comunità.<sup>75</sup>*

Nonostante le licenze indichino che chiunque può ridistribuire versioni modificate di un programma, nella pratica questo è un privilegio riservato ai manutentori del progetto.

Sono accettate versioni modificate di un programma sviluppate per soddisfare esigenze individuali da persone non appartenenti al gruppo dei manutentori, ma sono scoraggiate le distribuzioni dirette all'intera comunità.

---

<sup>75</sup> E. S. RAYMOND, *Colonizzare la noosfera*, cit..

Leggi informali regolano le procedure di acquisizione della “proprietà” di un progetto abbandonato: l’interessato deve rendere pubblica la sua intenzione di prendersi in carico un progetto attraverso forum e mailing lists, far passare del tempo in modo che il manutentore precedente abbia la possibilità di manifestare le sue volontà, (tra le quali la possibilità di delegare un successore), infine dimostrare attraverso donazioni di codice la sua competenza tecnica.

Un grande sforzo viene fatto da tutti i programmatori per annotare i nomi di tutti i contribuenti e per gestire la cronologia delle modifiche.

L’omissione o la cancellazione del nome di uno sviluppatore nei crediti del programma è ritenuta un’azione molto grave, denunciata dai partecipanti con molta forza e avviene raramente.

Il *forking*<sup>76</sup> è considerato negativamente da tutta la comunità perché provoca una duplicazione del lavoro collettivo. Esso viene accettato solo quando la scissione dà corpo a due progetti estremamente differenti tra loro, che coprono diversi campi applicativi.

Nelle discussioni attraverso mailing list e forum emerge anche la tendenza diffusa nei programmatori “anziani”, cioè quelli che hanno maggiore visibilità e considerazione all’interno della comunità open source, a esprimersi con umiltà ed evitare atteggiamenti auto-celebrativi. I leader ottengono la fiducia dei partecipanti soltanto assicurando che le scelte riguardanti le porzioni di codice da inserire all’interno della versione ufficiale di un programma vengono presi in base a criteri condivisi riguardanti la qualità.

La selezione basata sul merito è fondamentale per garantire il carattere di apertura della comunità.

---

<sup>76</sup> La divaricazione di un progetto.

Queste regole informali, alle quali gran parte degli hackers open source si attengono, sembrano svolgere, a detta di Raymond, la funzione di

*...stimolare maggiore responsabilità pubblica, maggiore attenzione collettiva e maggiore cura nel preservare i nomi dei partecipanti e le cronologie delle modifiche in ogni progetto, così da stabilire, tra l'altro, la legittimità dei proprietari correnti<sup>77</sup>.*

Nell'ottica dell'autore, le norme precedentemente descritte servirebbero a incentivare la collaborazione attraverso il gioco della reputazione tra pari.

Questo punto mette in luce il forte aspetto relazionale delle comunità free software/open source. Non si contribuisce solamente per motivazioni utilitaristiche, benché queste siano sempre presenti, ma anche per mettersi in relazione con un gruppo di persone che condivide interessi, etiche e pratiche comuni.

Il fatto di basarsi su donazioni di codice conferisce alle comunità virtuali free software/open source caratteristiche peculiari. Infatti,

*L'ingresso in una relazione di scambi di doni implica un' indebitamento perenne, un coinvolgimento in relazioni personali che limitano la libertà di interrompere a proprio piacimento la relazione<sup>78</sup>.*

L'instaurazione di un rapporto di reciprocità basato sul dono conferisce nuove caratteristiche alle comunità virtuali di tipo free software/open source.

I legami che contraddistinguono le comunità virtuali sono spesso indicati come deboli, specialmente in relazione ai legami di tipo primario: la grande libertà lasciata ai singoli individui può spingerli a uscire ed entrare molto facilmente.

---

<sup>77</sup> E.S. RAYMOND, *Colonizzare la noosfera*, cit. p.6.

<sup>78</sup> M. BERRA, A.R. MEO, *Informatica solidale. Storia e prospettive del software libero*, p.150, Bollati Boringhieri, Torino 2001.

Tuttavia, in linea con Mannarini<sup>79</sup> crediamo che la presenza di norme ed etiche condivise e l'attribuzione di grande importanza alla dimensione relazionale e dello stare insieme possa contribuire a rendere più stabili i legami virtuali.

Nel caso delle comunità free software/open source i legami sono ulteriormente rafforzati dalla presenza del dono, che costituisce la base intorno alla quale si costruisce la comunità. E' infatti attraverso di esso che le comunità riescono a limitare il fenomeno del *free riding*.<sup>80</sup>

Se poi consideriamo le peculiarità del digitale, comprendiamo come il software, e soprattutto il codice sorgente del software, costituisca un tipo di dono del tutto particolare. I bassi costi di riproduzione del software, la non deperibilità del bene in questione e la possibilità di lavorare servendosi di un'infrastruttura decentrata e aperta rendono economico e conveniente lo scambio di doni. Esso appare ancora più conveniente in un campo come l'informatica in cui il livello dell'innovazione e della complessità è molto alto. Il pieno accesso alle conoscenze reciproche che si ottiene attraverso la collaborazione è centrale. Le licenze free software/open source offrono poi l'ulteriore garanzia che i doni rimarranno disponibili a tutti, andando a costituire una sorta di bene pubblico.

Le comunità free software/open source riescono a conciliare l'apertura, la propensione volontaristica e la libertà individuale tipica delle comunità virtuali con un senso di identità di gruppo e un richiamo all'obbligatorietà della partecipazione più vicino (anche se molto differente) a quello delle comunità tradizionali di tipo amicale<sup>81</sup>.

---

<sup>79</sup> T. MANNARINI, *Comunità virtuali. Figure del legame sociale*, pp. 115-123, in B. R. GELLI (a cura di), *Comunità, rete, arcipelago. Metafore del vivere sociale*, cit..

<sup>80</sup> Il termine si riferisce ai casi in cui alcuni membri di un team di sviluppo lavorano meno degli altri, conquistandosi comunque i meriti della realizzazione del progetto.

<sup>81</sup> Ci riferiamo qui alle relazioni di tipo secondario, non legate a legami di sangue ma caratterizzate da una certa persistenza nel tempo.

Grandi sono i legami che congiungono le comunità free software/open source con il mondo accademico: la collaborazione e lo scambio frequente di conoscenze sono fondamentali per gestire la complessità dei sistemi e i frequenti cambiamenti cui sono sottoposti.

Edwards propone il paragone tra le comunità free software/open source e le comunità epistemiche descritte da Haas<sup>82</sup>, elencando le seguenti caratteristiche comuni:

- 1) *A shared set of normative and principled beliefs, providing a value based rationale for the social action of community members;*
- 2) *Shared causal beliefs, which are derived from their analysis of practice leading or contributing to a central set of problems in their domain and serving as the basis for elucidating the multiple linkages between possible policy actions and desired outcomes;*
- 3) *Shared notions of validity – that is, intersubjective, internally defined criteria for weighing and validating knowledge in the domain of their expertise;*
- 4) *A common policy enterprise – that is, a set of common practices associated with a set of problems to which their competence is directed, presumably out of the conviction that human welfare will be enhanced as a consequence<sup>83</sup>;*

Nel nostro caso, i programmatori free software/open source hanno una serie di principi comuni, quali la convinzione che il software proprietario restringa la libertà degli utenti

---

<sup>82</sup> P.M. HAAS, *Introduction: Epistemic Communities and International Policy Coordination*, in *International Organization* 46 (Winter 1991), pp. 1-36.

<sup>83</sup> K. EDWARDS, *Epistemic communities, Situated Learning and Open Source Software Development*. Department of Manufacturing Engineering and Management, Technical University of Denmark, reperibile in <http://opensource.mit.edu/papers/kasperedwards-ec.pdf>.

e la loro possibilità di scelta e l'opposizione ai brevetti sul software, considerate un grande ostacolo per i programmatori e particolarmente nocive per l'innovazione.

Condividono criteri di scelta derivati dagli studi (generalmente nell'area tecnico-informatica) e dalla conoscenza del progetto a cui partecipano. Da questi criteri dipenderà poi il futuro sviluppo del progetto stesso.

La presenza di comuni criteri di validità rende più facile la scelta delle patch da includere nelle nuove release e contribuisce ad attenuare i conflitti tra gli sviluppatori, desiderosi di vedere i propri contributi inseriti nelle distribuzioni ufficiali.

I criteri di validità riguardano anche il linguaggio utilizzato e il modo di proporsi alla comunità, che dipendono anche dallo status reputazionale dello sviluppatore stesso.

La presenza di obiettivi comuni, come lo sforzo di creare programmi liberi che costituiscano un'alternativa al software proprietario crea un quadro di riferimento che rende più dirette e semplici le conversazioni tra sviluppatori che non si conoscono e suggerisce temi di discussione.

Le istanze comuni qui descritte sono decisive nel contesto di interazione in remoto, creando un quadro di riferimento che può essere dato per scontato, riducendo in tal modo le possibili ambiguità che possono nascere nella comunicazione tra i partecipanti.

Nella visione di Haas le comunità epistemiche emergono in presenza di alti fattori di incertezza. Il paragone tra queste e le comunità del free software/open source sembra calzante visto il contesto informatico, caratterizzato da enorme complessità e necessità di aggiornamenti continui.

Ci interessa qui capire come i partecipanti sviluppino le competenze, i valori, i metodi di valutazione e i linguaggi sopra evidenziati in un contesto particolare, basato su di una infrastruttura decentrata.

Possiamo suddividere i partecipanti al processo di sviluppo di un progetto in almeno tre categorie principali:

1) *Manutentori*: sono gli sviluppatori autorizzati a rilasciare le versioni ufficiali del programma. Sono le persone che hanno maggiore voce in capitolo sulle scelte riguardanti le direzioni che il progetto prenderà. Si occupano, oltre a contribuire con il proprio codice alla scrittura del programma, di rivedere, correggere e inserire nelle release il codice proveniente dai collaboratori. Generalmente questa categoria è costituita da persone con grandi abilità di programmazione ed esperienza consolidata nel mondo dell'open source.

2) *Collaboratori*: si tratta di persone coinvolte in svariati modi in un progetto open source. Contribuiscono attraverso donazioni di codice, fissaggio o semplice segnalazione dei bug, ma anche attraverso meno impegnative discussioni all'interno dei forum o attraverso mailing lists. Possono in qualche modo essere considerati collaboratori anche le persone che cercano informazioni e chiedono agli utenti più esperti informazioni sull'uso del programma, in quanto danno il loro contributo mettendo in luce possibili problemi di usabilità del software.

3) *Utilizzatori*: sono quelle persone che utilizzano il software ma non partecipano in alcun modo al processo di sviluppo, nemmeno lasciando commenti o domande nei forum.

Quest'ultima categoria di utenti è di vitale importanza per i progetti open source, perché spesso costituisce il bacino dal quale nascono nuovi sviluppatori.

Alcuni utilizzatori, inizialmente percepiti come esterni dalla comunità, intraprendono un processo di socializzazione e di apprendimento che li porterà a diventare parte di essa, interiorizzandone le istanze comuni sopra descritte.

Alcuni di essi, infatti, spinti spesso dal desiderio di risolvere un problema riscontrato nell'uso del programma, iniziano a postare delle domande nei forum, cominciando ad interagire con gli utenti più esperti e maturano il desiderio di inserirsi all'interno della comunità.

I concetti sviluppati da Lave e Wenger<sup>84</sup>, poi applicati al contesto open source da diversi autori<sup>85</sup>, di *communities of practice* e di *apprendimento contestuale*, ci offrono un'ulteriore spunto per comprendere come avvenga il processo che porta i normali utenti a diventare parte della comunità di sviluppatori.

Le communities of practice sono gruppi di persone che si costituiscono sulla base di interessi e pratiche comuni. L'apprendimento all'interno di questi gruppi si attualizza attraverso l'interazione tra i partecipanti. Il sapere è strettamente connesso con il contesto in cui viene usato ed è un tutt'uno con le relazioni sociali che si creano nell'interazione tra i partecipanti e l'ambiente in cui si sviluppa.

I principianti non vengono a contatto con conoscenze astratte ma con indicazioni estremamente mirate al compito che stanno svolgendo.

---

<sup>84</sup> J. LAVE, E. WENGER, *Situated Learning: Legitimate Peripheral Participation*, Cambridge University Press, 1991.

E. WENGER, *Communities of Practice: Learning, Meaning and Identity*, Cambridge University Press, 1998.

<sup>85</sup> I. TUOMI, *Internet, Innovation, and Open Source: Actors in the Network*, in *First Monday* Volume 6, Number 1 (July 2002), reperibile in [http://www.firstmonday.dk/issues/issues6\\_1/tuomi/](http://www.firstmonday.dk/issues/issues6_1/tuomi/)

S. FARAJ, M.M. WASKO, *The Web of Knowledge: An Investigation of Knowledge Exchange in Networks of Practice*, 2001 reperibile in <http://opensource.mit.edu/papers/Farajwasko.pdf>

Nonostante le osservazioni di Tzouris<sup>86</sup>, che notano come la teoria di Lave e Wenger non si riferisca ad un contesto di lavoro remoto e distribuito, ci sembra plausibile considerarla adatta ad analizzare le comunità free software/open source.

Se ogni sviluppatore lavora in solitudine davanti al proprio computer, è vero anche che ha a propria disposizione una serie di strumenti di interazione in tempo reale e non, quali IRC<sup>87</sup>, forum, mailing list e programmi telefonici peer-to-peer<sup>88</sup>, che rendono il contesto lavorativo più ricco e simulano la situazione del lavorare insieme.

Gli utenti divengono collaboratori a tutti gli effetti attraverso un'apprendimento che ha come punto di partenza l'inserimento immediato nella comunità in qualità di principianti. Gli appartenenti alla comunità accettano la presenza della *legitimate peripheral participation* descritta da Lave e Wenger<sup>89</sup>, in cui i principianti vengono accettati come membri periferici all'interno di una comunità con il ruolo di apprendisti. Attraverso questo apprendimento situato nel contesto comunitario gli apprendisti acquisiscono non solo abilità di programmazione, ma anche un linguaggio specifico, uno stile comunicativo e un particolare approccio al lavoro di gruppo.

Nel processo open source gli utenti rappresentano la risorsa potenziale attraverso la quale reclutare nuovi partecipanti.

La trasparenza e l'apertura degli artefatti e dei processi nel modello free software/open source danno vita ad un nuovo tipo di rapporto tra progettista e utente finale, che si attualizza mediante il livellamento delle barriere comunicative tra le due figure.

---

<sup>86</sup> M. TZOURIS, *Rethinking Communities of Practice in the Networked Organisation – an Organisational Learning Study*, reperibile in: <http://www.geocities.com/tzmnlaos/cop/cop.pdf>

<sup>87</sup> Internet Relay Chat: sistema di messaggistica in tempo reale.

<sup>88</sup> Con i quali è possibile telefonare da un computer all'altro.

<sup>89</sup> J. LAVE, E. WENGER, *Situated Learning: Legitimate Peripheral Participation*, cit.

Bollier<sup>90</sup> guarda al movimento free software/open source come ad un movimento dei consumatori, in cui gli utenti di computer riacquistano gradualmente il controllo dei mezzi che usano. Non sono più semplici fruitori costretti a scegliere tra una relativamente ampia varietà di prodotti messi a punto dalle aziende, ma parte integrante e attiva del processo di indirizzamento e creazione degli strumenti adatti a soddisfare i propri bisogni.

Il software si trasforma quindi da semplice prodotto che viene creato, venduto e comprato, in “*a new kind of knowledge and community-building infrastructure*”<sup>91</sup>.

#### 4.5 Usabilità

Abbiamo descritto il modello open source come un processo comunitario di creazione del software, in cui sviluppatori e utilizzatori fanno parte della stessa comunità.

Questo è particolarmente vero nei primi anni di nascita e diffusione del software libero/open source. Esso trae le sue origini dalla cultura hacker, estremamente tecnica e dedita a costruire strumenti altamente innovativi e sofisticati, destinati a divenire a loro volta risorse della comunità. Il software libero muove i primi passi attraverso la creazione di programmi scritti da sviluppatori per gli sviluppatori.

I programmi open source di maggiore successo, quali il server web Apache, il database server Mysql, l'editor Emacs, il linguaggio Perl, sono concretamente utilizzati da amministratori di sistemi e programmatori più che da utenti medi.

---

<sup>90</sup> D. BOLLIER, *The Power of Openness. Why Citizens, Education, Government and Business Should Care About the Coming Revolution in Open Source Code Software. A Critique and a Proposal for the H20 Project*, 1999. Reperibile in <http://www.opencode.org/h20>.

<sup>91</sup> Ivi, p.3.

Nelle prime fasi dello sviluppo del movimento l'usabilità dei programmi non rappresentava un problema, in quanto questi venivano progettati avendo in mente degli utilizzatori con caratteristiche simili a quelle dei programmatori stessi.

*Concettualmente, l'usabilità di un prodotto, e più specificamente di un prodotto software, misura la distanza cognitiva fra il modello del progettista (modello del prodotto e delle sue modalità d'uso, possedute dal progettista e incorporate nel prodotto) e il modello dell'utente (modello di funzionamento del prodotto che l'utente si costruisce e che regola l'interazione col prodotto): quanto più i due modelli sono vicini tanto meno l'usabilità costituirà un problema<sup>92</sup>.*

Soltanto nella seconda metà degli anni '90 iniziarono a nascere programmi a codice aperto per gli utenti finali: programmi di fotoritocco (GIMP), editor di testo (OpenOffice), lettori multimediali (XMMS).

In questo periodo le comunità free software/open source iniziarono a percepire l'ambiente desktop come un fattore cruciale per la diffusione del software a codice aperto e per l'estensione del suo uso a fasce d'utenza non specialistiche. I progetti di interfaccia grafica per l'utente finale GNOME e KDE costituirono un grande passo avanti in questa direzione, fornendo un desktop grafico sulla scia di Windows e Macintosh.

Fino a questo momento, la maggior parte dei progetti a codice aperto erano applicazioni non esistenti nella versione commerciali, nati per colmare dei vuoti di mercato.

Il desiderio da parte delle comunità e delle aziende che supportano il software open source di raggiungere ampie fasce di utenza, fornendo una reale alternativa al software

---

<sup>92</sup> D. ZIGGIOTTO, *Usabilità del WWW: metodi di valutazione*, p.2, reperibile in: <http://www.hyperlabs.net/ergonomia/ziggiotto>

commerciale, può realizzarsi soltanto attraverso una discussione e un impegno sui temi dell'usabilità.

Secondo l'ISO 9241<sup>93</sup> l'usabilità è valutabile attraverso i livelli di efficacia, efficienza e soddisfazione con cui gli utenti raggiungono i loro obiettivi in determinati contesti.

E' importante sottolineare che l'usabilità non deve essere considerata una caratteristica del prodotto, ma il risultato di un processo di interazione contestuale tra l'utente, il prodotto e i fini dell'utente stesso.

Il problema dell'usabilità si è posto nel software commerciale a metà degli anni '80, quando i computer sono entrati negli uffici e nelle abitazioni, utilizzati da utenti non esperti. Le compagnie di software hanno così cercato di affrontare il problema assumendo esperti dell'interazione uomo-computer, che iniziarono a lavorare fianco a fianco con i programmatori.

Nel caso dell'open source l'emergere di una nuova utenza di non esperti è un fenomeno nuovo, e deve essere ancora affrontato in maniera completa.

Una delle prime ragioni consiste nel fatto che gli sviluppatori open source sono spesso motivati a contribuire dalla necessità di disporre di funzionalità avanzate non ancora presenti e sono meno interessati a svolgere del lavoro per degli utenti meno esperti.

Altri fattori, quali il ridotto apporto di investimenti sul software open source da parte delle aziende per costituire team multidisciplinari, la più generale difficoltà di individuazione dei problemi di usabilità rispetto a quelli di funzionamento, la propensione degli sviluppatori a lavorare sul codice in maniera interattiva e non pianificata, senza un progetto alla base, e non ultima la preferenza (derivata dalla cultura hacker) degli sviluppatori per delle interfacce che non nascondano all'utente i

---

<sup>93</sup> ISO 9241, *Ergonomic Requirements for office work with visual display*, 1997.

meccanismi di funzionamento della macchina, ci restituiscono un quadro più completo della situazione dell'usabilità nel modello open source.

Negli anni recenti, la progettazione degli artefatti informatici si è avvicinata sempre di più ad una prospettiva che metta al centro dell'analisi l'utente nel suo contesto. L'interazione uomo-computer viene concepita come un processo di adattamento reciproco e iterativo, non come un adattamento dell'uomo alla macchina.

Il conflitto uomo-computer, secondo Mantovani, può essere ricondotto a quello tra ruoli socio-professionali differenti:

*E' possibile considerare gli artefatti come la posta in un gioco di cooperazione-conflitto tra gruppi socio-professionali che se ne disputano il controllo, al fine – non necessariamente consapevole, anzi di solito non esplicitamente presente – di influenzare, tramite tali strumenti, l'ambiente sociale e fisico<sup>94</sup>.*

E' sulla scia di riflessioni di questo genere, nel tentativo di accorciare la distanza tra progettista e utilizzatore, che iniziano a fare il loro ingresso nel campo dell'HCI<sup>95</sup> figure professionali provenienti dall'antropologia, dall'etnologia e dalla sociologia.

A partire dalla seconda metà degli anni '90, nell'intento di andare ancora più a fondo nel processo di avvicinamento tra le due figure contrapposte, inizia a farsi strada un'approccio all'usabilità detto *design partecipativo*.

Esso prevede il diretto coinvolgimento dell'utente nelle varie fasi del processo di creazione del prodotto, del quale è ritenuto corresponsabile insieme al progettista.

*La produzione del software non è più un processo lineare, ma un processo iterativo in cui si perviene al risultato finale attraverso aggiustamenti*

---

<sup>94</sup> G. MANTOVANI, *L'interazione uomo-computer*, Il Mulino, Bologna 1995, pp.72-73.

<sup>95</sup> Human-Computer Interaction.

*successivi guidati dalla continua verifica delle esigenze e delle necessità dell'utente finale*<sup>96</sup>.

Il parallelo tra questo approccio e il modello free software/open source, dove gli utenti (per ora soltanto quelli dotati di elevate capacità tecniche) partecipano attivamente al processo di costruzione collettiva di un progetto in cui gli obiettivi e le necessità vengono stabilite *in itinere* attraverso l'interazione tra i partecipanti, sembra essere calzante oltre che fruttuoso per l'individuazione di punti deboli e potenzialità ancora non sfruttate.

Il fattore chiave per il miglioramento dello stato di usabilità del software libero/open source sembra essere il coinvolgimento nel processo di sviluppo degli utenti finali.

Questo risultato può essere raggiunto combinando diversi approcci, attraverso un'impegno delle comunità free software/open source in varie direzioni.

La prima consiste nel fornire agli utenti delle semplici interfacce, automatizzate per raccogliere dati sul tipo di operazione effettuata dall'utente, per comprendere il contesto in cui si è verificato il problema di usabilità. L'utente deve essere fornito di strumenti per comunicare la proprie osservazioni agli sviluppatori sia in forma testuale che in forma grafica, per esempio scattando fotografie del desktop.

Attraverso il sistema dell'*advocacy*, in cui persone interessate all'usabilità e studiosi di HCI svolgono studi di usabilità su persone con cui intrattengono relazioni sociali di tipo amicale o familiare, si possono ottenere rapporti su problemi specifici, che vanno poi raccolti e interpretati. I collaboratori del team dedicato all'usabilità finiscono per costituire delle figure di tramite tra gli sviluppatori e gli utenti finali, creando un collegamento tra i due estremi.

---

<sup>96</sup> D. ZIGGIOTTO, *Usabilità del WWW: metodi di valutazione*, cit., p.6.

Il lavoro distribuito di tipo modulare applicato dai programmatori può essere riproposto anche per gli studi sull'usabilità, permettendo agli utenti e agli esperti di usabilità di lavorare su ambiti molto circoscritti e riservando l'attività di raccolta, organizzazione delle informazioni e coordinamento del lavoro agli analisti più esperti.

Come per la fasi di sviluppo, gli sforzi dei partecipanti al progetto usabilità devono essere segnalati all'intera comunità, stimolando un meccanismo basato sulla reputazione.

Nichols e Twidale mettono in evidenza come gli strumenti per rendere più semplice la collaborazione da parte degli utenti finali non siano sufficienti:

*Creating a technological infrastructure to make it easier for usability experts and users to participate will be insufficient without an equivalent social structure. These new entrants to OSS projects will need to feel welcomed and valued, even if (actually because) they lack the technical skills of traditional hackers<sup>97</sup>.*

Il miglioramento dello stato di usabilità del software open source per gli utenti finali deve passare attraverso un processo di apertura della stessa comunità hacker, incoraggiando l'ingresso all'interno delle comunità di nuovi gruppi sociali e occupazionali.

La nascita di gruppi femmili all'interno dei progetti open source (quali *KDE-Women*, gruppo di partecipanti al progetto K Desktop Environment<sup>98</sup> e *System*, una mailing-list di informatiche impegnate nell'open source) è il segno dell'affacciarsi di nuove voci all'interno della cultura hacker e del desiderio di farsi vedere ed essere riconosciuti.

---

<sup>97</sup> D.M. NICHOLS, M.B. TWIDALE, *Usability and Open Source Software*, University of Waikato 2002, reperibile in <http://www.cs.waikato.ac.nz/~daven/docs/oss-wp.html>.

<sup>98</sup> Un progetto di Graphical User Interface per GNU/Linux.

La nascita di comunità in rete basate sul libero scambio di conoscenze, quali Wikipedia (un'enciclopedia "libera" sul modello del software libero), l'Open Directory Project (un catalogo del Web redatto da partecipanti volontari), Opsound (un sito che raccoglie materiale audio rilasciato sotto licenze libere), costituisce una possibilità di allargamento delle comunità free software/open source.

Queste comunità basano infatti la loro interazione su programmi di gestione dei contenuti (*Content Management System*<sup>99</sup>) di tipo open source e spesso dalle discussioni che emergono tra i partecipanti nascono esigenze e proposte di nuove funzionalità che vengono presentate agli sviluppatori.

Gli sviluppatori che partecipano a queste comunità costituiscono il tramite tra gli utenti e i programmatori stessi, contribuendo ad avvicinarne la distanza.

La partita dell'usabilità e della stessa estensione del modello del software libero/open source si giocherà anche sul piano del locale, nei rapporti tra i gruppi locali di utilizzatori di software a codice aperto e le pubbliche amministrazioni.

---

<sup>99</sup> Un programma destinato all'automatizzazione dei progetti di creazione, gestione e distribuzione dei contenuti sul web. Permette agli autori dei contenuti di non preoccuparsi della loro visualizzazione nella pagina, affidata al manutentore del sito, che si occupa della forma.

## 5. Software libero e partecipazione sociale

### 5.1 La società dei saperi

Nella sua opera “L’intelligenza collettiva<sup>100</sup>” Pierre Levy individua lo spiraglio per la nascita di quello che chiama un nuovo “*spazio antropologico*”, dominato dall’importanza cardinale del sapere e della conoscenza.

Con il termine “spazio antropologico” si fa riferimento ad

*...un sistema di prossimità (spazio) proprio del mondo umano (antropologico) e dunque dipendente dalle tecniche, dai significati, dal linguaggio, dalla cultura, dalle convenzioni, dalle rappresentazioni e dalle emozioni umane.*

Caratteristica di questi spazi è la tendenza a non sostituirsi agli spazi precedenti ma a sovrapporvisi, rimodellandoli e ri-articolandoli in base alle nuove istanze dominanti.

Lo spazio della Terra, basato sul rapporto degli uomini con il cosmo, lo spazio del territorio, segnato dalla nascita dell’agricoltura, della scrittura e delle unità statali, lo spazio delle merci, dominato dai flussi di materie prime, mano d’opera e informazioni, si ristrutturano in funzione del nuovo spazio, incentrato sulla padronanza delle informazioni e sulla capacità di coordinarle.

Gli indizi della nascita del nuovo spazio del sapere andrebbero rintracciati nella crescente terziarizzazione dell’economia, nella necessità sempre più sentita dalle aziende di connettersi ad altre per scambiarsi informazioni e riallacciarsi ad una serie di servizi complementari, nell’esigenza continua di rinnovamento e creazione di nuovi saperi per far fronte ad una realtà sempre più complessa.

---

<sup>100</sup> LEVY P., *L’intelligenza collettiva. Per un’antropologia del cyberspazio*

Gli elementi di novità di questo spazio sono la velocità con cui si evolvono i saperi e le conoscenze tecnico-scientifiche, il grande numero di persone chiamato a produrre nuove conoscenze e i nuovi strumenti basati sul digitale utilizzati per il trattamento delle informazioni e per la loro trasmissione.

Lo Spazio del sapere individuato da Levy non rappresenta un'analisi della società odierna ma acquisisce, piuttosto, valore progettuale. Per l'autore la crescente difficoltà di gestione di una società in cui la necessità di coordinamento e organizzazione delle informazioni e delle conoscenze specialistiche dei campi più diversi è sempre più sentita, deve attualizzarsi attraverso la messa in comune delle esperienze individuali in *immaginanti collettivi* all'interno dei quali *“le potenzialità sociali e cognitive di ciascuno possano svilupparsi e ampliarsi reciprocamente”*<sup>101</sup>.

Il sapere diventa gestibile soltanto attraverso una partecipazione piena degli individui ad una collaborazione che prevede il rispetto della diversità delle esperienze dell'altro e l'inserimento in un processo di apprendimento reciproco e continuativo.

L'intento di Levy è indicare una strada possibile per la risoluzione di alcuni problemi cruciali: il progressivo assottigliarsi della sfera pubblica e del legame sociale sotto la spinta individualizzante del mercato, in un contesto in cui gli stati nazionali stanno gradualmente perdendo la loro capacità di influenza rispetto a quest'ultimo; la crisi del lavoro, sia sotto le spinte della crescente automazione delle funzioni produttive che nei termini di una maggiore difficoltà nello stare al passo con la richiesta, stimolata dal vorticoso processo di innovazione, di competenze sempre più specialistiche dal punto di vista tecnico e relazionale.

La proposta di Levy deve essere considerata una potenzialità, non certo come una premonizione.

---

<sup>101</sup> LEVY P., *L'intelligenza collettiva*, cit., p.31.

Rifkin<sup>102</sup>, pur essendo concorde con le teorie sulla società dell'informazione nell'attribuire un peso sempre più rilevante alla conoscenza e all'informazione rispetto ai beni materiali e ponendole al centro del processo di trasformazione sociale in corso, evidenzia come la cultura stia attraversando un processo di marcata mercificazione. Rileva come la proprietà intellettuale venga raramente scambiata, rimanendo in possesso del fornitore, che in cambio di un pagamento in denaro ne permette l'accesso in forma temporanea o comunque limitata.

*...la proprietà di beni materiali diventa sempre più marginale per l'esercizio del potere economico, mentre quella di beni immateriali si sta trasformando nella forza dominante di un sistema economico fondato sull'accesso. Idee sotto forma di brevetti, diritti sulle opere dell'ingegno, marchi, segreti industriali e relazioni sono utilizzate per creare un nuovo tipo di potere economico in cui pochi "mega-fornitori" controllano estese reti di utenti<sup>103</sup>.*

Si assiste al passaggio da una logica economica basata sulla dicotomia venditori/compratori ad una incentrata sulla coppia fornitori/utenti.

La prospettiva delle aziende si sposta dalla produzione al marketing, in una logica che vede l'utente al centro di un processo di personalizzazione che mira alla costruzione di relazioni economiche a lungo termine.

L'instaurazione di una relazione con il cliente è la chiave d'accesso per l'ingresso in un rapporto economico teso a coprire l'intera esperienza di vita di quest'ultimo. Il cliente, e la capacità di relazionarsi, costituisce in sé stesso un valore economico, tanto

---

<sup>102</sup> RIFKIN J., *L'era dell'accesso. La rivoluzione della new economy.*, Milano, Arnoldo Mondadori Editore 2000.

<sup>103</sup> Ivi, p.79.

maggiore quanto più può estendersi lungo l'intero arco della vita (*lifetime value*, come viene chiamato dagli specialisti di marketing).

Secondo Rifkin "l'era dell'accesso si definisce, soprattutto, attraverso il crescente asservimento delle esperienze alla sfera economica<sup>104</sup>" mediante una progressiva infiltrazione di quest'ultima in porzioni sempre più ampie della vita quotidiana degli individui.

Il quadro dipinto da Rifkin, pur definendo il sapere, i beni immateriali, l'informazione e la componente relazionale come risorse strategiche nell'attuale contesto sociale, ci mostra un paesaggio notevolmente distante dal progetto delineato da Levy.

La cardinalità del sapere all'interno del processo di trasformazione in corso non ci dice come i vari attori in gioco si relazioneranno in rapporto a quest'ultimo.

Nel progetto di Levy ci troviamo di fronte ad un rapporto con il sapere esercitato in maniera partecipativa, vissuto attraverso un processo di costruzione collettiva da parte dei cittadini. D'altra parte non possiamo non considerare le tendenze in corso descritte da Rifkin, di crescente accentramento del controllo da parte di grandi multinazionali sulla fruizione culturale dei cittadini, sempre più relegati al ruolo di ricettori di pacchetti confezionati su misura.

E' indubbia, nel contesto di una società sempre più incentrata sulla capacità di creazione, gestione e distribuzione della conoscenza, l'importanza rivestita dalle tecnologie basate sul digitale. Salutate da molti entusiasti come tecnologie di libertà, capaci di rinvigorire le fondamenta delle istituzioni democratiche stimolando processi di maggiore partecipazione dei cittadini alla sfera pubblica e allo stesso tempo considerate da altri uno strumento nelle mani delle grandi imprese per accrescere il loro controllo sulle vite e sulle esperienze individuali, le nuove tecnologie dell'informazione e della

---

<sup>104</sup> RIFKIN J., *L'età dell'accesso*, cit., p.131.

comunicazione si prestano ad utilizzi molto differenti e il loro potenziale è suscettibile a sviluppi molto diversi.

## 5.2 Gli strumenti telematici

Nel primo libro<sup>105</sup> della sua trilogia intitolata “L’età dell’informazione”, Castells delinea i punti distintivi del nuovo paradigma della tecnologia dell’informazione.

Esso sarebbe caratterizzato dalla presenza dell’informazione in quanto materia prima, sulla quale agiscono le nuove tecnologie e dalla *pervasività* degli effetti di quest’ultime.

Cruciale è la tendenza all’instaurarsi di una *logica a rete* nei sistemi e nelle relazioni che utilizzano le nuove tecnologie. L’adozione di quest’ultime e quindi l’ingresso in un sistema reticolare diventa tanto più conveniente in proporzione allo sviluppo della rete stessa, mentre aumenta lo svantaggio per chi ne è escluso.

Un’ulteriore caratteristica è la *flessibilità*, che permette una “riconfigurazione”<sup>106</sup> costante della struttura delle relazioni tra le componenti in gioco.

La flessibilità dei sistemi basati sul digitale è stata attentamente analizzata da Negroponte<sup>107</sup>. Nel mondo dei bit vi è la possibilità di non vincolarsi ad una scelta definitiva sul loro impiego, tenendosi aperti a diverse possibilità di evoluzione e di riarrangiamento contestuale. Una caratteristica saliente del digitale è la possibilità di disporre di bit che “parlano” di altri bit, permettendo la catalogazione e il filtraggio delle informazioni. In merito a questo aspetto Negroponte parla di uno spostamento dell’“intelligenza” (rispetto ai meccanismi presenti nell’emissione televisiva o radiofonica) dall’emittente al ricevente.

---

<sup>105</sup> CASTELLS M., *La nascita della società in rete*, Milano, Università Bocconi Editore 2002.

<sup>106</sup> Ivi, p.76.

<sup>107</sup> NEGROPONTE N., *Essere digitali*, ed. it., Milano, Sperling & Kupfer Editori 1995.

*La tecnologia digitale cambierà la natura dei mass media, nel senso che da una situazione in cui i bit vengono “sospinti” verso l’utente, si passerà ad una in cui sarà quest’ultimo (o il suo computer) a “tirarli” a sé<sup>108</sup>.*

E’ innegabile che a partire dagli anni ’80 si sia assistito - con l’introduzione di nuove tecnologie quali il videoregistratore, il registratore a cassette, le videocamere, e poi negli anni ’90, in cui si sviluppano la tv via cavo e la trasmissione via satellite – ad un processo di “de-massificazione” dell’audience televisivo.

Bisogna però notare che questo processo di decentralizzazione, diversificazione e personalizzazione, intensificatosi con l’avvento del digitale, non ha comportato “una perdita del controllo sulla televisione da parte di grandi imprese e governi.”<sup>109</sup>

Non sembra così confermata la previsione di Negroponte secondo la quale “gli imperi monolitici dei mass media si stanno frantumando in una miriade di piccole imprese.”<sup>110</sup>

Al contrario, “mentre il pubblico si segmentava e si diversificava, la televisione è diventata più commercializzata che mai e, sempre più, un’industria oligopolistica.”<sup>111</sup>

Se il processo di personalizzazione della televisione non ha portato necessariamente ad un maggiore controllo e indipendenza dei cittadini sui mezzi di comunicazione, non dobbiamo rischiare di commettere lo stesso errore nel considerare la rete Internet.

Come suggerito da Castells,

*...risulta essenziale mantenere la distanza tra la valutazione della nascita di nuove forme e processi sociali, in quanto indotti e resi possibili dalle nuove tecnologie, e la deduzione delle potenziali conseguenze di tali sviluppi per la società e le persone(...)<sup>112</sup>.*

---

<sup>108</sup> N. NEGROPONTE, *Essere digitali*, cit., p.83.

<sup>109</sup> M. CASTELLS, *La nascita della società in rete*, cit., p.393.

<sup>110</sup> N. NEGROPONTE, *Essere digitali*, cit. p.54.

<sup>111</sup> M. CASTELLS, *La nascita della società in rete*, cit., p.396.

<sup>112</sup> M. CASTELLS, *La nascita della società in rete*, cit. p.76.

Le varie forme di comunicazione mediata dal computer nate sulla base della piattaforma Internet sono state spesso indicate come tecnologie di libertà.

Questo punto di vista tradisce a nostro avviso una prospettiva deterministica, secondo la quale le tecnologie in sé sarebbero capaci di cambiare (in meglio) persone e organizzazioni. Si considera in questo modo soltanto una parte del processo di innovazione, senza fare attenzione ad un altro aspetto, riguardante il “quadro d’uso”<sup>113</sup>.

Non ci si chiede infatti come le tecnologie vengano cambiate e riadattate dagli utilizzatori per far fronte alle proprie esigenze.

*Dobbiamo chiederci quale tipo di interazione si crei di volta in volta tra le componenti tecniche del sistema e le sue componenti umane, perché abbiamo a che fare con una causazione circolare, non una causalità unidirezionale in cui le nuove tecnologie agirebbero sulle organizzazioni migliorandole*<sup>114</sup>.

Nel prendere in considerazione la tecnologia di Internet, consapevoli di trovarci di fronte ad un vasto insieme di tecnologie diverse che convergono su un’unica piattaforma, dobbiamo innanzitutto considerare il processo di costruzione dell’immaginario collettivo che ha fatto da sfondo alla nascita e allo sviluppo della tecnologia stessa.

Nell’analizzare il processo di creazione dell’immaginario che è andato a costituire un quadro di riferimento per la nascita e lo sviluppo del personal computer, Flichy<sup>115</sup> concentra il suo interesse sui gruppi di informatici che, dando vita a circoli, club e riviste, hanno formulato un primo quadro d’uso di questo strumento.

---

<sup>113</sup> P. FLICHY, *L’innovazione tecnologica. Le teorie dell’innovazione di fronte alla rivoluzione digitale*, Milano, Feltrinelli 2001.

<sup>114</sup> G. MANTOVANI, *Comunicazione e Identità. Dalle Situazioni Quotidiane agli Ambienti virtuali*, Bologna, Il Mulino 1995.

<sup>115</sup> P. FLICHY, *L’innovazione tecnologica*, cit., pp. 202-208.

Il personal computer probabilmente non si sarebbe mai sviluppato e diffuso se non fosse stato sostenuto da quell'ideologia comunitaria, strettamente legata alla corrente contro-culturale dei primi anni '70, che credeva nella possibilità di mettere il potere dei computer al servizio delle persone e nell'utilizzo di questi in senso antiburocratico.

Flichy evidenzia come anche nel passaggio ad una vera e propria industria del personal computer il progetto di un computer "conviviale e decentralizzato"<sup>116</sup> abbia rappresentato un punto di riferimento importante.

La costruzione di un immaginario tecnologico è una attività profondamente sociale, in cui diversi gruppi di attori si confrontano dando vita ad una base culturale comune che costituisce un punto di partenza per gli sviluppi futuri.

*All'origine della tecnologia troviamo (...) un complesso di luoghi di socializzazione dove i progetti sono presentati e dove, dal confronto fra le diverse ipotesi, nasce un primo quadro di riferimento.*<sup>117</sup>

Approcciandoci a Internet dobbiamo quindi valutare innanzi tutto quali gruppi sociali abbiano preso parte alla costruzione dell'immaginario tecnologico sottostante e quali istanze culturali abbiano proposto, ricordando come i quadri di riferimento non siano cornici statiche ma sistemi dinamici sempre suscettibili di ristrutturazione in relazione agli attori che partecipano alla loro costituzione e al loro sviluppo.

L'idea che Internet sia di per sé una tecnologia emancipante, democratizzante e intrinsecamente pluralistica rischia di mascherare il suo carattere squisitamente sociale tendendo a nascondere i processi costantemente in corso al suo interno.

Dobbiamo però ammettere, giustificando in parte chi fa queste affermazioni, che fino ai giorni nostri Internet ha rappresentato e continua a rappresentare un canale di

---

<sup>116</sup> Ivi, p.206.

<sup>117</sup> Ivi, p.207.

comunicazione relativamente libero, pluralista, e che permette agli utenti di essere non solo fruitori di risorse ma anche produttori. Questo è dovuto in particolar modo alla presenza di standard aperti e all'ampia diffusione di programmi freeware e open source, che permettono ai piccoli produttori di disporre di strumenti utili per raggiungere i loro obiettivi con costi contenuti.

Le prime soluzioni tecniche adottate nella progettazione della rete originavano da una concezione visionaria che immaginava la nascita di una "rete galattica"<sup>118</sup> che collegasse ogni luogo e permettesse l'accesso a informazioni e programmi da qualsiasi punto.

Queste concezioni, condivise dai progettisti e dai primi fruitori, agivano come quadro di riferimento per il concepimento degli artifici tecnici. E' a partire da queste concezioni che si cercò di costruire una piattaforma che potesse essere il più indipendente possibile dalle caratteristiche dei computer ad essa collegati, fornendo degli standard universali che permettessero il collegamento ad essa dei dispositivi più diversi.

Non si trattava soltanto di costruire un dispositivo decentrato a fini difensivi per assicurare la continuità delle comunicazioni in caso di attacco ma, nelle concezioni dei suoi progettisti, di costruire una rete che permettesse la comunicazione tra tutti i cittadini del globo.

L'influenza di queste visioni sui primi fruitori dei network che andavano ad aggiungersi ad Arpanet spingeva verso una concezione della rete come un sistema aperto, che stimolasse l'interoperabilità tra sistemi e il libero scambio di informazione.

Anche con l'ingresso di entità commerciali all'interno della rete le componenti competitive sono sempre state affiancate da una forte componente di cooperazione tra identità differenti. La prima cultura della rete abbracciava un punto di vista volto ad incentivare la proliferazione dei saperi, la contaminazione e l'arricchimento reciproco

---

<sup>118</sup> Concetto proposto nel 1962 da J.C.R. Licklider del Massachusetts Institute of Technology.

tra entità differenti, piuttosto che stimolare la chiusura identitaria e la compartimentazione in unità differenti.

Sarebbe però cecità non notare, a fianco di queste istanze, tuttora presenti nell'ecologia della rete, la presenza di strategie aziendali mirate al controllo e alla segmentazione degli utenti attraverso l'introduzione di standard proprietari.

Una visione deterministica del processo di innovazione tecnologica rischia di farci perdere di vista i processi in corso senza renderci consapevoli del fatto che il futuro degli strumenti telematici, in materia di democrazia e partecipazione sociale, dipende strettamente dall'azione di singoli, gruppi, aziende e istituzioni.

Nell'idea di società dell'informazione, che distinguiamo da quella di società della comunicazione, "si compie la nostra posizione di exteriorità rispetto alla tecnica, similmente a quanto già avvenuto rispetto alla scienza, elemento decisivo per cui si possa realizzare quella dinamica del tutto autocontenuta di mercato e tecnologia."<sup>119</sup>

La tendenza a proporre tecnologie "invisibili", apparentemente semplici da usare e allo stesso tempo annoverare tra i vantaggi primari delle nuove tecnologie digitali la possibilità di fornire servizi "*on demand*" nasconde la volontà di porre l'utilizzatore in una posizione di passività, concependolo come un semplice consumatore di servizi preconfezionati.

Il software libero/open source si fa al contrario portavoce di un punto di vista che rimette in discussione la tecnologia presentandola come qualcosa di inscindibile dalla società, che deve diventare oggetto di dibattito pubblico attraverso la libertà di accesso al suo stesso processo di sviluppo.

---

<sup>119</sup> A. C. FRESCHI, *La società dei saperi*, cit., p.40.

L'apertura del processo di sviluppo rende possibile la verifica, l'adattamento ai bisogni collettivi e permette l'apprendimento, andando a costituire una grande risorsa per l'istruzione e la formazione, in ambito scolastico e aziendale.

L'apertura del codice sorgente permette in qualsiasi momento il *porting*<sup>120</sup> di un programma a qualsiasi piattaforma, impedendo la formazione di "blocchi" tecnologici che legherebbero gli utenti a determinate aziende con aspirazioni monopolistiche.

L'insieme di queste considerazioni dimostra l'importante ruolo del software libero/open source nella creazione di quegli strumenti, auspicati da Levy, che aiuteranno a "filtrare i flussi di conoscenze, a navigare nel sapere e a pensare insieme piuttosto che a trasportare masse di informazioni."<sup>121</sup>

### 5.3 Intelligenza collettiva e modello free software/open source

L'intelligenza collettiva descritta da Levy è "un'intelligenza distribuita ovunque, continuamente valorizzata, coordinata in tempo reale, che porta ad una mobilitazione effettiva delle competenze"<sup>122</sup>. Il fine di questo tipo di intelligenza sta nel "riconoscimento e arricchimento reciproco delle persone<sup>123</sup>", non nell'appiattimento delle individualità in una collettività massificata, uniformata e uniformante.

Non si tratta della costituzione di una grande mente onnicomprensiva e inglobante, quanto della valorizzazione di tutte le competenze individuali, alla ricerca della ricchezza della diversità dell'"altro", attraverso uno spirito collaborativo proteso a "costruire insieme" a partire dalle differenze individuali.

---

<sup>120</sup> L'adattamento di un programma ad una diversa piattaforma.

<sup>121</sup> P. LEVY, *L'intelligenza collettiva*, cit., p.31.

<sup>122</sup> P. LEVY, *L'intelligenza collettiva*, cit. p.34.

<sup>123</sup> Ibidem.

*Lungi dal fondere le intelligenze individuali in una sorta di magma indistinto, l'intelligenza collettiva è un processo di crescita, di differenziazione e di mutuo rilancio delle specificità*<sup>124</sup>.

Levy intravede la possibilità, grazie ai mezzi forniti dalle nuove tecnologie digitali, di attuare una gestione razionale delle qualità umane, che permetta una minimizzazione degli sprechi nel campo delle competenze degli individui. Le nuove tecnologie, che egli etichetta come molecolari, permettono un controllo gene per gene nel campo della scienza genetica, atomo per atomo a livello microscopico, bit per bit nel campo dell'informazione e del controllo dei flussi dei messaggi.

L'applicazione di queste tecnologie nel campo della regolazione dei gruppi umani permetterebbe quindi una valorizzazione del patrimonio umano "qualità per qualità"<sup>125</sup>.

Gli individui si organizzerebbero autonomamente in base alle proprie competenze in gruppi tesi alla risoluzione di temi specifici, attraverso discussioni in tempo reale, interattive, basate sulla costruzione di un legame sociale tra i partecipanti.

Il progetto di Levy viene volutamente presentato in modo molto generale e vuole offrire soprattutto una traccia per un possibile sviluppo di nuove modalità di accesso al sapere basate sulla condivisione e sulla collaborazione reciproca.

Sul piano reale, molte sono state le applicazioni degli strumenti telematici che hanno cercato di percorrere questa direzione.

---

<sup>124</sup> Ivi, p.37.

<sup>125</sup> Ivi, p.69.

Usenet<sup>126</sup> e i primi BBS<sup>127</sup>, che si diffusero a macchia d'olio nei primi anni '80, rappresentarono un primo esempio su larga scala di luogo virtuale di discussione retto da contribuzioni volontaristiche.

Questi primi sistemi hanno in particolare permesso il coordinamento su scala globale di gruppi locali di ogni parte del mondo legati da principi e valori comuni. Questi traevano dalle reti orizzontali due tipi di vantaggi:

- la possibilità di discutere su progetti comuni e proporre iniziative coordinate.
- la capacità di acquisire maggiore visibilità presso l'opinione pubblica.

Con lo sviluppo del World Wide Web a partire dagli anni '90 e l'integrazione in esso dei vari sistemi di messaggistica sono aumentate sempre di più le possibilità per le associazioni no-profit di far sentire la loro voce a livello globale, proponendo temi di discussione e facendo emergere informazioni spesso trascurate dai media tradizionali.

Il modello organizzativo proposto dal movimento free software/open source rappresenta la punta di diamante di questo modo di concepire le reti telematiche in senso orizzontale e connettivo.

Gli elementi di novità di questo approccio rispetto ai precedenti vanno ricercati nei seguenti punti:

- creazione di un corpus di artefatti collettivi che forniscono la base per il lavoro futuro e allo stesso tempo incorporano i principi regolatori del lavoro collettivo.
- dotazione degli strumenti legali atti a preservare la dimensione collettiva degli artefatti (le licenze "libere").

---

<sup>126</sup> Un sistema che permetteva lo scambio di dati tra utenti Unix.

<sup>127</sup> Un Bulletin Board Systems è costituito da un computer che fa girare un software che permette agli utenti di collegarsi ad esso attraverso la linea telefonica usando un programma per terminale. Permette lo scaricamento di software e dati, la possibilità di inserire dati, leggere news e scambiare messaggi con altri utenti.

- circolo virtuoso che trasforma gli artefatti derivanti dal lavoro collettivo in risorse per l'azione futura. Le soluzioni ad un determinato problema, rimanendo sempre accessibili e disponibili a tutti, possono essere direttamente riutilizzate senza svolgere ulteriore lavoro.
- adozione di uno stile di collaborazione incentrato sulla modularità che, oltre a permettere ad ogni individuo o gruppo di lavorare su un problema specifico senza interferire sul lavoro altrui, consente a ciascun partecipante di lavorare nell'ambito in cui si sente più ferrato.
- possibilità di adattamento degli artefatti a problemi e bisogni specifici.

Il modello di cooperazione comunitaria proposto dal movimento free software/open source si avvicina, per quanto in un'ambito di interesse molto ristretto, alla soddisfazione dei punti indicati da Levy nella definizione di intelligenza collettiva.

L'intelligenza delle comunità open source è in primo luogo *distribuita*, in quanto messa in atto da individui provenienti da ogni parte del mondo. Il limite all'universalità di questa distribuzione è innanzi tutto di tipo politico economico ed è esprimibile in termini di divario digitale e di possibilità di accesso a Internet tra paesi del Sud e del Nord del mondo. Il problema non si esaurisce al puro accesso (dal punto di vista della presenza di infrastrutture tecnologiche e del costo per i cittadini) ma, come sottolinea Freschi, "è necessario anche promuovere un accesso competente, consapevole (...) e critico – altrimenti questo si rivela addirittura controproducente - proprio rispetto alle specificità del mezzo, che possono favorire tanto un uso attivo e interattivo, che un uso passivo di tipo nuovo (...), definibile in termini di interpassività"<sup>128</sup>.

L'intelligenza del movimento free software/open source è *continuamente valorizzata*, nella misura in cui ognuno può contribuire, grazie all'approccio modulare, in base alle

---

<sup>128</sup> A.C. FRESCHI, *La società dei saperi*, cit. p.45.

proprie competenze, affrontando i problemi che lo interessano maggiormente. La partecipazione può avvenire a vari livelli e anche i semplici suggerimenti o osservazioni degli utenti costituiscono una risorsa importante. Inoltre, molti progetti open source si stanno ulteriormente aprendo ad una visione a tutto tondo del software e stanno nascendo gruppi di lavoro che si occupano, oltre alle traduzioni e alla documentazione, anche del marketing del prodotto e della grafica.

E' un'intelligenza *valorizzata in tempo reale*: i ruoli e le attività di ogni singolo partecipante sono ridefinibili giorno per giorno, in base ad un quadro contestuale che tiene conto delle esigenze e delle scelte di ognuno, delle decisioni prese a riguardo degli sviluppi futuri del progetto, dell'ingresso di nuovi partecipanti e dell'affacciarsi di nuovi problemi con i quali confrontarsi.

L'apertura del processo permette allo stesso tempo un'apprendimento continuo attraverso una condivisione ed uno scambio di conoscenze che oltrepassa le barriere verticali legate a ruoli rigidi e immobilizzati.

Il modello free software/open source si relaziona quindi alla società dell'informazione e della comunicazione in maniera duplice:

- sul versante tecnico mette in atto una vera e propria apertura della scatola nera tecnologica, praticando un livellamento della frattura tra tecnologia e società e favorendo una riflessione critica sugli strumenti da parte degli utilizzatori.
- da una prospettiva più ampia, fornisce un modello di lavoro collettivo e di condivisione della conoscenza applicabile a tutti i campi del sapere, non soltanto a quello del software.

La metodologia organizzativa del software libero, considerata nei suoi aspetti giuridici, economici e comunitari, è stata infatti presa ad esempio da una serie di iniziative e progetti che si stanno sviluppando e concretizzando in diversi ambiti.

## 5.4 Oltre il software: sperimentazioni

Descriveremo in questo paragrafo alcuni progetti esplicitamente ispirati alle esperienze del movimento free software/open source che non hanno a che vedere con lo sviluppo software e che vengono spesso indicati come esempi di “open source intelligence”<sup>129</sup> o “open content”.

Cercheremo di mettere in luce il modo in cui questi progetti articolano in diversi modi alcune istanze dello stile organizzativo open source alle esigenze di altri ambiti, quali la produzione musicale, audiovisiva, letteraria e scientifica.

### 5.4.1 Creative Commons

Il progetto Creative Commons<sup>130</sup> nasce nel 2001 dall’iniziativa di un gruppo di persone guidato dai giuristi Lawrence Lessig, James Boyle e Michael Carrol, insieme con l’esperto informatico Hal Abelson e l’editore Eric Eldred, e viene sostenuto da finanziamenti della “Stanford Law School” e dal “Center for the Public Domain”.

Creative Commons è un’organizzazione no-profit che ha l’obiettivo di fornire licenze legali destinate a opere dell’ingegno che permettano all’autore la possibilità di riservarsi alcuni diritti e allo stesso tempo rilasciarne altri ai fruitori.

Le licenze proposte da Creative Commons prendono diretta ispirazione dalla licenza GNU GPL messa a punto da Richard Stallman, fondatore della Free Software

---

<sup>129</sup> F. STALDER, J. HIRSH, *Open Source Intelligence*, First Monday, volume 7, number 6 (June 2002).  
Reperibile in: [http://firstmonday.org/issues/issue7\\_6/stalder/index.html](http://firstmonday.org/issues/issue7_6/stalder/index.html).

<sup>130</sup> <http://creativecommons.org>

Foundation. Esse propongono diversi gradi di “libertà” per quanto riguarda l’uso e la fruizione delle opere rilasciate in questa forma e tutelano diversi tipi di opere: registrazioni audio, video, testi, immagini e risorse per l’educazione.

Il meccanismo di queste licenze si basa sul copyright: l’autore si assicura attraverso di esso la paternità dell’opera garantendo ai fruitori svariati diritti, quali la possibilità di ridistribuire l’opera in forma completa e la possibilità di distribuirne versioni derivate (previa citazione della fonte). Esistono licenze che autorizzano, pur conservando intatta la possibilità di copia e redistribuzione per tutti, gli usi commerciali di un’opera e altre che invece li restringono.

Una delle licenze CC (Creative Commons) più diffuse, denominata “Attribution-ShareAlike-2.0”, è la trasposizione della GNU GPL al di fuori del campo del software. Essa permette la distribuzione dell’opera al pubblico, la creazione di opere derivate, l’utilizzazione a scopi commerciali a patto che si rispettino due condizioni: la paternità dell’opera deve essere riconosciuta all’autore originario e tutte le opere derivate devono essere rilasciate sotto la stessa licenza.

Contemporaneamente alla diffusione delle licenze CC sono nati diversi circuiti di aggregazione che rilasciano l’intero materiale raccolto sotto quest’ultime. Siti web come Opsound<sup>131</sup>, comunità di musicisti in rete e netlabels (ovvero etichette di produzione musicale sul web) collezionano un grande quantitativo di musica coperta da licenze Creative Commons. Alcune di queste, come Magnatune<sup>132</sup>, hanno elaborato una vera e propria strategia commerciale attraverso la vendita online e per via postale delle registrazioni ma consentendone la copia e la redistribuzione purchè a scopo non commerciale e richiedendo invece il pagamento di una quota per l’utilizzo a scopo

---

<sup>131</sup> <http://opsound.org>

<sup>132</sup> <http://magnatune.org>

commerciale. Il vasto Internet Archive<sup>133</sup>, invece, fornisce spazio web per tutte le opere rilasciate sotto dominio pubblico o sotto licenze Creative Commons.

Recentemente il progetto Creative Commons ha guadagnato ampia visibilità grazie all'appoggio di personalità del mondo dello spettacolo e dell'industria discografica. Nel novembre 2004 il Ministro della Cultura e artista brasiliano Gilberto Gil, durante la conferenza del "Software Livre" di Porto Alegre, ha presieduto al lancio della versione brasiliana delle licenze. La rivista americana di tecnologia e cultura Wired ha distribuito "The Wired CD: Rip. Mix. Sample. Mash. Share", un cd audio contenente registrazioni create da artisti conosciuti in ambito internazionale che hanno deciso di rilasciare le loro creazioni sotto licenze CC, incoraggiando esplicitamente la produzione di opere derivate da parte del grande pubblico. La BCC, nel maggio 2004, ha annunciato la nascita del "Creative Archive", un'archivio di materiale audiovisivo interamente basato sul sistema di licenziamento Creative Commons. L'intento dichiarato di questa iniziativa è quello di stimolare la crescita della cultura creativa in Inghilterra e allo stesso tempo incoraggiare la produzione di programmi derivati che potranno essere riutilizzati all'interno della programmazione dell'emittente.

Il quadro legale fornito dalle licenze Creative Commons pone le basi per un modello di condivisione della conoscenza mirante alla creazione di un vasto repertorio di opere dell'ingegno, fruibile e (spesso) modificabile da tutti, che andrebbe a costituire una risorsa importante soprattutto per piccole case di produzione video e audio, Internet radio, creatori di pagine e contenuti per il web, associazioni no-profit ed enti educativi.

Il movimento free software/open source non ha rappresentato soltanto una semplice ispirazione per il progetto Creative Commons: tra i due movimenti si è infatti creato un rapporto di tipo sinergico e collaborativo. L'esempio più lampante è costituito dalla

---

<sup>133</sup> <http://www.archive.org>

nascita di numerosi programmi che facilitano l'applicazione delle licenze CC ai propri lavori creativi, la pubblicazione di materiali in rete e la ricerca di materiale coperto da tali licenze. Ogni autore intenzionato ad utilizzare le licenze CC per i propri lavori può, ad esempio, inserire dei metadati che segnalino agli utenti la licenza utilizzata per quel lavoro attraverso ccTag e inserire il materiale all'interno dell'Internet Archive (che offre hosting gratuito per il materiale "libero") usando ccPublisher. Il motore di ricerca ccSearch, infine, consente la ricerca di questi materiali filtrandoli per tipo di licenza utilizzata e formato (audio, video, testi, immagini, siti web).

L'intento del progetto portato avanti da Lessig è multiplo. Da un lato, fornire uno strumento legale a persone, enti pubblici e privati interessati a condividere i propri materiali; dall'altro, stimolare una vera e propria proliferazione dei saperi consentendo allo stesso tempo la *navigabilità* in essi. La disponibilità di una grossa base di materiale creativo e informativo andrebbe ad avvantaggiare la ricerca storica, sociale, artistica e permetterebbe l'adattamento dei materiali a diversi scopi, attraverso adattamenti e trasposizione in diversi formati e in diverse lingue. Una piena accessibilità dei saperi dipende, oltre che dalla disponibilità fisica di infrastrutture informative adeguate, dalla possibilità di adattare i saperi alle esigenze di tutti, a partire dalle categorie spesso escluse in quanto minoritarie. Si tratta di proporre un modello opposto a quello della "comunicazione di massa" e allo stesso tempo distinto da una fornitura di servizi personalizzati basata su una prospettiva esclusivamente utilitaristica, andando a costruire una piattaforma partecipata che vada incontro ai bisogni reali delle persone. La proliferazione e moltiplicazione dei saperi però non garantisce da sola una reale *inclusione* nella società dell'informazione, se non è accompagnata dallo sforzo di rendere questo sapere navigabile e fruibile. Il progetto Creative Commons mira proprio

a questo, fornendo un sistema di etichettatura del materiale creativo che permetta ad ognuno di conoscere con chiarezza quali sono i suoi diritti sugli artefatti. Sarebbe ingenuità pensare che obiettivi come quelli dell'accessibilità e dell'inclusione sociale possano essere raggiunti semplicemente attraverso l'utilizzo di uno strumento quale il progetto CC: la costruzione di una società dei saperi realmente partecipata dipende dal coinvolgimento in questa direzione di tutte le parti sociali a livello globale e richiede enormi sforzi sia in campo economico che in campo educativo.

Per quanto riguarda l'Italia, solo recentemente il progetto ha iniziato ad acquistare una certa popolarità. Le traduzioni italiane delle licenze CC, frutto di un lavoro coordinato dal dipartimento di Scienze giuridiche dell'Università di Torino e dall'Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni (IEIIT-CNR), sono state infatti presentate il 16 dicembre 2004 a Torino. iCommons Italy ha recentemente lanciato una iniziativa denominata "Scarichiamoli!" che prevede il lancio di un progetto di legge che stabilisca che tutte le opere dell'ingegno finanziate esclusivamente attraverso fondi pubblici debbano essere sottoposte alla condizione di pubblico dominio o comunque sotto una licenza "libera". Viene inoltre proposta l'istituzione di un portale italiano dedicato alla raccolta di questo materiale.

## 5.4.2 Wikipedia

Wikipedia<sup>134</sup> è una enciclopedia multilingue basata su Internet, liberamente consultabile e modificabile da chiunque. Nel gennaio 2005 contava, nella versione inglese, 450.000 voci. Wikipedia nasce nel 2001, come progetto laterale di Nupedia, un'altra enciclopedia libera che disponeva di editori pagati dai fondatori del progetto.

---

<sup>134</sup> <http://en.wikipedia.org>

Attualmente tutti i contributori del progetto sono volontari.

Tutto il materiale dell'enciclopedia è rilasciato sotto la GNU Free Documentation License, una licenza originariamente concepita dalla Free Software Foundation per essere utilizzata nella documentazione del software (guide, tutorials, referenze). Il contenuto di Wikipedia può essere quindi usato, modificato, copiato e ridistribuito liberamente. Il suo database può essere utilizzato da chiunque per costituire dei mirrors<sup>135</sup> o delle versioni modificate. Il sistema web è retto da MediaWiki, un software basato su database che, a partire dal 2002, si è arricchito attraverso l'opera di debugging e l'aggiunta di funzionalità messa in atto da diversi sviluppatori.

La principale caratteristica di Wikipedia è che permette a tutti i lettori di correggere, ampliare, modificare l'articolo che stanno leggendo, sia in forma anonima che registrandosi nel sito e dandosi quindi un'identità. Il software sul quale è basata permette la conservazione delle pagine nella forma precedente alle modifiche, consentendo in qualsiasi momento di ritornare alle versioni precedenti nel caso in cui siano state inserite informazioni non corrette o brani che sottostanno ad un regime di copyright incompatibile con la strategia di licenziamento prevista.

Le motivazioni dei partecipanti a Wikipedia sono per certi versi molto simili a quelle degli sviluppatori di software libero open source: il fatto di compiere un'attività di per se stessa stimolante, la soddisfazione personale legata alla reputazione dei pari, la convinzione politica che l'informazione debba essere "libera" e soprattutto l'idea di prendere parte ad una attività collettiva.

Rispetto alle enciclopedie tradizionali, Wikipedia si differenzia soprattutto per un'alto grado di diversità nella qualità e nell'approfondimento degli articoli. Basandosi su una filosofia che lascia totale libertà di scelta agli editori su quali voci ampliare, troviamo in

---

<sup>135</sup> Copie speculari del sito.

essa articoli molto approfonditi, spesso in grado maggiore rispetto alle enciclopedie tradizionali, così come articoli di bassa qualità o a basso livello di approfondimento.

D'altro canto, troviamo in essa tematiche e voci spesso sottosviluppate nelle normali enciclopedie, così come news e articoli generalmente presenti nelle riviste specializzate.

Un altro problema di questo tipo di “intelligenza open source” è legato al vandalismo. Questo è il tema generalmente più dibattuto dai critici di Wikipedia, in quanto la possibilità di editare semplicemente gli articoli anche in forma anonima aprirebbe le porte agli eventuali disturbatori. Questo problema si è rivelato meno imponente del previsto. Secondo una ricerca condotta dall'IBM<sup>136</sup>, la maggior parte del materiale danneggiato dagli atti di vandalismo viene generalmente ripristinato nel giro di cinque minuti. Ciffolilli<sup>137</sup> spiega il basso livello di vandalismo (quantomeno in relazione a quello che ci si aspetterebbe), con i bassi costi di “costruzione creativa” rispetto ai più alti costi di “distruzione creativa” permessi dalla modalità di funzionamento del software Wikimedia, che permette appunto di ripristinare la precedente condizione di una pagina con un solo “click”.

Nonostante questo, dobbiamo notare come l'organizzazione “anarchica” di Wikipedia la distanzi da quelle caratteristiche del modello free software/open source in gran parte responsabili dell'alto livello di qualità e affidabilità di quest'ultimo.

Nel modello di sviluppo del software libero/open source i partecipanti vengono introdotti nella comunità di sviluppatori attraverso quella che abbiamo chiamato *legitimate peripheral participation*, attraverso la quale possono apprendere i valori, le pratiche, il linguaggio e lo stile di lavoro della comunità. Gli aspiranti a far parte di un

---

<sup>136</sup> IBM History Flow. Reperibile in: <http://researchweb.watson.ibm.com/history/>

<sup>137</sup> A. CIFFOLILLI, *Phantom authority, self-selective recruitment and retention of members in virtual communities: The case of Wikipedia*, First Monday, volume 8, number 12 (December 2003).  
Reperibile in: [http://firstmonday.org/issues/issue8\\_12/ciffolilli/index.html](http://firstmonday.org/issues/issue8_12/ciffolilli/index.html).

progetto devono mostrare le loro competenze attraverso l'invio di patch di debug, correzioni, suggerimenti ma, finchè non avranno raggiunto un'adeguato livello di maturità, non potranno accedere al CVS<sup>138</sup>. Prima di essere diffuso, il loro lavoro dovrà passare attraverso la revisione degli sviluppatori più esperti.

In Wikipedia, questo genere di partecipazione periferica non è presente: le modifiche effettuate da un utente vengono pubblicate istantaneamente, senza alcuna assicurazione che i nuovi partecipanti abbiano realmente introiettato le regole condivise dalla comunità, le consuetudini stilistiche e le direttive guida.

Il processo editoriale di Wikipedia, se ha il vantaggio di attirare un grande numero di contributori, attirati dalla semplicità d'uso e dalla velocità con cui possono vedere le proprie modifiche messe in atto, non garantisce un'affidabilità e qualità degli articoli paragonabile a quella delle enciclopedie tradizionali. D'altra parte permette la discussione partecipata dei temi, riflettendo gli interessi reali dei partecipanti alla comunità. Alla ricerca di una possibile soluzione al problema, affinché si possa mirare alla costruzione di una risorsa che sia partecipata ma allo stesso tempo garanzia di affidabilità, qualità e autorevolezza, avanziamo la proposta, in analogia con il modello di sviluppo del software libero/open source, di costituire due versioni dell'enciclopedia.

Una, sperimentale, consentirebbe la modifica e l'aggiunta di materiale in maniera istantanea da parte di tutti gli utenti, permettendo l'aggiornamento e l'inserimento continuo di nuove voci che verranno sviluppate in futuro.

La seconda, stabile, conterrebbe tutte le voci della versione sperimentale che abbiano raggiunto un'adeguato livello di maturità. La possibilità di modificare le voci di questa versione sarebbe riservata soltanto agli utenti con un'adeguato livello di esperienza, che avrebbero il ruolo di controllare e supervisionare la qualità degli articoli. In questo

---

<sup>138</sup> Current Versioning System.

modo, la partecipazione al processo editoriale della versione sperimentale costituirebbe il terreno di prova per gli aspiranti editori della versione stabile.

### 5.4.3 Open Directory Project (ODP)

L'Open Directory Project<sup>139</sup> è un progetto teso alla costruzione di una *web directory* interamente costruita attraverso l'apporto di editori volontari. Con il termine *web directory* intendiamo una lista ordinata, divisa per categorie e completa di descrizioni sintetiche e accurate di un'ampio numero di siti web, scelti in base a caratteristiche di qualità, funzionalità, utilità e originalità. L'esempio più popolare di *web directory* di tipo commerciale è rappresentato dalla Yahoo Directory<sup>140</sup>, costruita attraverso l'apporto di centinaia di editori retribuiti.

Gli albori dell'Open Directory Project vanno rintracciati nell'iniziativa del programmatore Rick Skrenta, che nel 1998 diede avvio a GnuHoo, una *directory* costruita con l'apporto di editori volontari.

Similmente a quanto avviene nel processo editoriale "anarchico" di Wikipedia, GnuHoo permetteva la partecipazione di chiunque in maniera totalmente libera. Chiunque volesse partecipare poteva farlo scegliendo una categoria e cominciando ad aggiungere i siti che preferiva. Non vi erano al tempo né gerarchie che differenziassero lo status di ogni editore, né direttive e regole precise a cui attenersi. Frequenti erano i conflitti tra editori, spesso più interessati all'inserimento di siti per i quali nutrivano specifici interessi che alla creazione di una *directory* di qualità. La mancanza di direttive conferiva una bassa omogeneità nello stile editoriale e la qualità degli inserimenti non era sempre garantita. Comunque, pochi mesi dopo il suo lancio GnuHoo, che aveva

---

<sup>139</sup> <http://dmoz.org/>

<sup>140</sup> <http://dir.yahoo.com/>

cambiato il suo nome in NewHoo, sotto pressioni della Free Software Foundation spaventata dalla possibilità che la directory potesse essere confusa con il progetto Gnu, sperimentava un sempre maggiore successo di pubblico.

Nel 1999 NewHoo viene acquisito da Netscape per il suo sito Netcenter e cambia il suo nome in Open Directory Project. L'ODP Social Contract<sup>141</sup>, attraverso il quale Netscape mette in chiaro i suoi obblighi nei confronti della comunità degli editori, garantisce lo status di libera fruibilità della directory, l'apertura al pubblico delle direttive editoriali e la disponibilità della base dati per il libero riutilizzo da parte di chiunque sia interessato. Da questo momento in poi, le direttive di inserimento dei siti e di descrizione annessa si fanno sempre più precise e dettagliate, generando una sempre maggiore qualità ed omogeneità editoriale.

Poco tempo dopo verranno introdotte delle gerarchie tra editori, basate sull'anzianità (in termini di esperienza all'interno del progetto) e sulla competenza. In termini generali, gli editori più alti in gerarchia possono editare anche le categorie spettanti a editori di grado minore e hanno il compito di esaminare le richieste di partecipazione degli aspiranti nuovi editori e degli editori desiderosi di essere promossi ad una categoria superiore o di poter editare nuove categorie. Similmente a quanto avviene nel modello free software/open source le richieste di partecipazione al progetto richiedono l'inserimento di nuovi siti, completo di descrizione, per testare in maniera pratica le competenze del nuovo arrivato.

Il processo dell'Open Directory Project è completamente comunitario, basato sull'interazione tra gli editori attraverso numerosi forum, divisi per categorie di interesse e creati in base alle esigenze della comunità stessa.

---

<sup>141</sup> <http://dmoz.org/socialcontract.html>

A partire dal 2002 poi, viene istituito il Forum Pubblico ODP, di libero accesso al pubblico più ampio (e non solo agli editori, come avviene per i forum interni) che permette finalmente un'interazione tra gli utenti del servizio e gli editori, in un'ottica di sempre maggiore trasparenza. Grazie ad esso è inoltre possibile la messa in atto di quella *legitimate peripheral participation* che abbiamo indicato come punto di forza per l'apprendimento delle regole e delle pratiche da parte dei nuovi partecipanti. Attraverso il forum pubblico gli utenti possono infatti segnalare malfunzionamenti, carenze, fornire suggerimenti e proporre nuovi siti da inserire, facendosi così un'idea del processo editoriale e delle sue regole.

In piena sinergia con il modello free software/open source, sono nate anche numerose applicazioni sviluppate da alcuni partecipanti che forniscono degli strumenti per semplificare il lavoro degli editori, quali validatori automatici dei link e strumenti per tenersi aggiornati su particolari forum.

Attualmente l'Open Directory Project costituisce la più ampia e completa directory presente sul web, con la presenza di 590.000 categorie per circa quattro milioni di siti selezionati e 65.000 editori volontari.

La base dati di ODP è attualmente utilizzata da numerosi portali quali America On Line(AOL), che predispose una versione dei dati customizzata sulle esigenze dei suoi utenti e Google, che combina in Google Directory il proprio motore di ricerca con i dati ODP. L'unica regola cui devono sottostare i siti che utilizzano la directory ODP è la citazione della fonte attraverso l'inserimento di un banner finalizzato al reclutamento di nuovi volontari.

## 6. Software libero/open source e Pubblica Amministrazione

### 6.1 Utilità del software libero/open source nella Pubblica Amministrazione

Il software libero/open source ha richiamato negli ultimi anni sempre maggiore interesse da parte delle Pubbliche Amministrazioni (PA) europee. Si sono rivelate decisive in merito allo sviluppo di questa attenzione tre istanze principali:

- a) la nascita di applicativi open source di qualità in numerose aree di interesse delle PA .
- b) alcune caratteristiche cruciali intrinseche al modello di sviluppo stesso di questo tipo di software.
- c ) le potenzialità insite in esso di stimolare l'innovazione nel campo dell'ICT.

Allo stesso tempo, la maturità raggiunta da alcuni progetti software e la loro crescente diffusione, insieme agli sforzi di tipo divulgativo messi in atto da organizzazioni quali la Free Software Foundation, la Open Source Iniziative e testate quali LinuxJournal, hanno contribuito a diradare quell'atmosfera di *fear, uncertainty and doubt (FUD)*<sup>142</sup> che si respirava nei primi anni della diffusione del software non proprietario.

Già diversi anni fa il panorama del software libero poteva vantare ottimi programmi nel campo del *software d'infrastruttura*<sup>143</sup>. Nell'ambito delle networking segnaliamo il web server Apache (il più diffuso al mondo), il browser Mozilla, l'*application server*<sup>144</sup> Zope, il *server DNS*<sup>145</sup> Bind e il server di posta Sendmail.

---

<sup>142</sup> FUD (fear, uncertainty, doubt), termine utilizzato per indicare un insieme di tecniche utilizzato da compagnie di software proprietario, spaventate dalla possibile perdita di quote di mercato, mirate a gettare discredito sul software a codice aperto circondandolo di un'atmosfera, appunto, di paura, incertezza e dubbio riguardante le sue caratteristiche tecniche.

<sup>143</sup> Si tratta di software che fornisce dei servizi ma non viene effettivamente usato dagli utenti.

<sup>144</sup> Un software dedicato a fare girare alcuni applicativi software in un ambiente di rete.

<sup>145</sup> Un programma che permette di assegnare dei nomi di dominio agli indirizzi IP.

Tra i sistemi di gestione di basi di dati (DBMS<sup>146</sup>) spiccano invece Mysql e PostgreSQL<sup>147</sup>.

Bisognerà aspettare la seconda metà degli anni '90 affinché il software a codice aperto possa costituire un'alternativa appetibile per le PA nel campo degli applicativi per l'utente finale. Fu solo da questo momento che vennero sviluppate le prime *graphical user interfaces* per Linux, Gnome e Kde, che hanno permesso l'utilizzo di questo sistema operativo anche agli utenti inesperti. Nel campo dell'*office automation* l'alternativa viene offerta da Sun Microsystems, che nel 2000 dà vita a Open Office, una versione open source della sua suite da ufficio Star Office, che allo stato attuale si è rivelato un programma di ottima qualità, ricco di funzionalità e basato su uno standard aperto, XML.

I benefici percepiti nell'adozione di software libero/open source da parte delle pubbliche amministrazioni vanno rintracciati soprattutto nel basso costo iniziale di azione, l'indipendenza dai fornitori, la sicurezza, la flessibilità e l'interoperabilità.

I *costi iniziali* dell'adozione di software open source si abbassano soprattutto in merito ai costi delle licenze e degli aggiornamenti. Molto spesso gli aggiornamenti del software proprietario vengono adottati non tanto perché apportano delle vere migliorie nel programma ma piuttosto per incompatibilità con le versioni precedenti.

Nel confrontare le spese di adozione da parte delle PA di software libero o proprietario non basta fare riferimento soltanto ai costi iniziali ma è bene valutare il TCO (*Total Cost of Ownership*). Con questo termine si intende prendere in considerazione non soltanto i costi delle licenze ma anche i servizi di consulenza e supporto, la formazione

---

<sup>146</sup> Data Base Management Systems.

<sup>147</sup> Per una descrizione più ampia dei software open source disponibili nei vari campi di applicazione rimandiamo a: Ministro per l'Innovazione e le Tecnologie, *Indagine conoscitiva sul software a codice a sorgente aperto nella Pubblica Amministrazione*, Roma, maggio 2003, reperibile in: [http://www.innovazione.gov.it/ita/comunicati/open\\_source/indagine\\_commissione\\_os.pdf](http://www.innovazione.gov.it/ita/comunicati/open_source/indagine_commissione_os.pdf).

degli impiegati, i costi di gestione e, non ultimi, i costi di migrazione. Questi in particolare possono essere molto onerosi in quanto spesso richiedono una ristrutturazione dei modi operativi dell'ente e investimenti sulla formazione del personale. Il software libero si rivela un'ottima soluzione soprattutto nei casi in cui si necessita di software *custom*, ovvero realizzato su misura da qualche azienda per l'Amministrazione che ne fa richiesta. I costi di sviluppo in questo caso si abbassano molto proprio grazie alla contribuzione volontaria di sviluppatori da ogni parte del mondo (nella misura in cui il progetto riesca effettivamente ad attrarre un alto numero di partecipanti).

L'adozione di software a codice aperto può rivelarsi strategica in merito alle esigenze di *indipendenza* delle PA. L'apertura dei sorgenti dà alle Amministrazioni la possibilità di affidarsi per il supporto alle aziende che preferisce, evitando di rimanere legata a tempo indeterminato all'azienda produttrice.

L'apertura dei sorgenti può allo stesso tempo fornire maggiori garanzie di *sicurezza*, rendendo possibile effettuare controlli sulla presenza di banchi e *back doors*.

Allo stesso tempo il software libero è generalmente dotato di una maggiore *flessibilità*, in quanto permette di apportare in maniera semplice personalizzazioni, estensioni delle funzionalità e adattamento ad altri sistemi.

Il software a sorgente aperto garantisce generalmente un alto grado di *interoperabilità*, termine con cui si indica la capacità di un sistema di condividere gli stessi dati con sistemi eterogenei. L'interoperabilità si raggiunge generalmente attraverso l'utilizzo di *formati standard aperti*. Con "formato aperto" intendiamo una "modalità di rappresentazione dei dati in forma elettronica, deliberatamente resa pubblica,

completamente documentata e utilizzabile da chiunque”<sup>148</sup>. Esso diventa “standard” quando è di fatto condiviso da un’ampia comunità. Non è scontato che tutte le aziende che sviluppano software proprietario utilizzino formati chiusi anche se è un comportamento molto diffuso, messo in atto al fine di avere un maggiore controllo sugli utenti nel corso del tempo. Alcuni formati chiusi proposti da programmi molto utilizzati sono diventati nel tempo standard di fatto, come i formati della suite Microsoft Office e il formato PDF dell’Adobe. Spesso una non perfetta compatibilità dei programmi open source con questi formati viene percepita come un problema di interoperabilità di quest’ultimi. E’ quindi pieno interesse delle Pubbliche Amministrazioni poter disporre di programmi che facciano uso di standard aperti, che assicurano indipendenza dai fornitori, alta interoperabilità e maggiore libertà di scelta per gli utenti. Inoltre, “i formati testo aperti standard (...) comportano l’ulteriore beneficio della *persistenza*, caratteristica importante per la tutela del patrimonio informativo nel tempo a fronte del mutamento tecnologico”<sup>149</sup>. Il formato XML utilizzato dalla suite Open Office rende ancora più evidente questo beneficio attraverso l’associazione ad ogni dato di una descrizione in linguaggio naturale (metadato).

Nel giugno 2002, tra gli svantaggi derivanti dell’adozione di software open source da parte delle PA, l’Autorità per l’Informatica nella Pubblica Amministrazione<sup>150</sup> segnalava l’instabilità del mercato, la carenza di applicazioni business e la carenza di drivers.

Crediamo che tali svantaggi non debbano essere considerati come punti deboli del software libero in sé, ma vadano interpretati come conseguenza della relativa giovinezza del modello. Se nei primi anni del suo sviluppo il modello dava vita

---

<sup>148</sup> MINISTRO PER L’INNOVAZIONE E LE TECNOLOGIE, *Indagine conoscitiva sul software a codice sorgente aperto nella pubblica amministrazione*, cit., p. 18.

<sup>149</sup> MINISTRO PER L’INNOVAZIONE E LE TECNOLOGIE, *Indagine conoscitiva sul software a codice sorgente aperto nella pubblica amministrazione. Rapporto della Commissione.*, cit., p. 18.

<sup>150</sup> AIPA, *Il software Open Source (OSS) scenario e prospettive*, i Quaderni, n°10 giugno 2002.

soprattutto a programmi destinati agli sviluppatori e agli amministratori di sistemi, recentemente sono nate molte applicazioni di tipo business per merito del crescente interesse delle aziende. Se è vero che fino a poco tempo fa i produttori di periferiche hardware non fornivano i driver per Linux, dobbiamo far notare come in tempi recenti diversi fornitori abbiano iniziato a garantire supporto al sistema operativo open source. Esempi celebri sono IBM, Intel, AMD. Il problema dei driver è strettamente legato al grado di diffusione del software open source ed è destinato a scomparire in misura proporzionale allo sviluppo del fenomeno. Non vi sono invece motivi per ritenere che il mercato del software open source sia più instabile di quello del software proprietario. Il mercato del software a codice aperto ha dimostrato la sua sostenibilità e in particolari situazioni si è dimostrato una scelta ottimale. Non si può parlare quindi di maggiore o minore stabilità del mercato open source rispetto a quello tradizionale, quanto di viabilità dei rispettivi modelli in merito a condizioni specifiche e contingenti.

Da un punto di vista economico non è possibile stabilire a priori se l'adozione di software libero/open source da parte delle PA comporti spese più contenute: ogni singola Amministrazione dovrà considerare la sua situazione specifica, variabile in base alle infrastrutture hardware a disposizione, ai software di cui necessita, al grado di informatizzazione degli impiegati, alle spese necessarie per la formazione e alla possibilità di ripartire le spese con altre Amministrazioni.

Da una prospettiva più ampia basata su considerazioni di principio, attraverso l'adozione di software a codice aperto le Pubbliche Amministrazioni potranno beneficiare di una maggiore trasparenza, di una più ampia indipendenza dai fornitori, della possibilità di rimettere in circolo il software creato su misura, della possibilità di stimolare un circolo virtuoso di collaborazione tra Amministrazioni e istituzioni

scolastiche e formative, ponendo il software utilizzato al centro dell'analisi e dello studio universitario e scientifico.

## 6.2 Il contesto europeo

Uno studio IDA<sup>151</sup> (Interchange of Data between Administrations) guidato dalla Commissione Europea fa il primo punto della situazione sull'uso e sulle direttive di adozione del software open source da parte delle Pubbliche Amministrazioni dei paesi europei.

Nel 2001 l'uso di software a codice aperto nel settore pubblico si concentrava in gran parte nel campo dei server. In particolare, si registrava un ampio uso del server web Apache e del sistema Linux negli Internet hosts<sup>152</sup>, mentre in ambito di reti intranet l'uso di componenti open source nei server si limitava ad un 8%, generalmente in sostituzione a soluzioni Unix proprietarie. L'uso di programmi open source in ambiente desktop risultava invece molto limitato (2%).

Nei casi in cui sono state adottate soluzioni open source, le caratteristiche percepite come determinanti per la scelta da parte dei managers IT del settore pubblico sono l'interoperabilità, la sicurezza, il rispetto degli standard e la funzionalità. La caratteristica del basso costo non emerge da questo studio come quella dominante, come si sarebbe potuto pensare.

Lo studio IDA<sup>153</sup> analizza la diffusione del software open source in sei paesi dell'Unione Europea: Belgio, Francia, Germania, Italia, Spagna e Svezia. Francia e Germania sono i paesi che fanno maggiore uso di questo genere di software all'interno

---

<sup>151</sup> INTERCHANGE OF DATA BETWEEN ADMINISTRATIONS, *Study into the use of Open Source Software in the Public Sector*, European Commission, DG Enterprise, June 2001.

<sup>152</sup> Computer connessi alla rete che ospitano dei siti web.

<sup>153</sup> INTERCHANGE OF DATA BETWEEN ADMINISTRATIONS, *Study into the use of Open Source Software in the Public Sector*, cit..

del settore pubblico. In entrambi l'appoggio dei governi all'adozione di software libero è forte e si è concretizzato in una serie di direttive volte ad incoraggiarne l'uso, numerosi dossier e istruzioni per l'adozione, siti di informazione e centri di consulenza e intermediazione.

Le direttive francesi incoraggiano fortemente l'uso di formati aperti riconoscendo all'open source il merito di essere un'importante promotore di questi ultimi. Il rapporto Carcernac<sup>154</sup> del 2001 suggerisce esplicitamente l'adozione di software open source da parte delle Pubbliche Amministrazioni francesi e auspica il rafforzamento del ruolo dell'MTIC, un'organo di coordinazione inter-ministeriale per lo sviluppo dell'ICT nelle amministrazioni, molto attento al mondo del software a codice aperto.

In Germania il KBSt, un centro di competenza del Governo federale per le tecnologie dell'informazione e della comunicazione nelle Pubbliche Amministrazioni, ha diffuso una lettera in cui il software open source viene indicato come una valida alternativa alla situazione presente nelle amministrazioni al momento della sua redazione. Il Governo federale supporta esplicitamente il BerliOS Project<sup>155</sup>, una struttura di mediazione per l'open source che si pone l'obiettivo di creare un momento di incontro tra sviluppatori open source, distributori, aziende e settore pubblico.

In Spagna il progetto Virtual MAP ha messo in atto la sostituzione di vecchi sistemi Unix proprietari con delle piattaforme Linux in numerose reti interne di competenza delle Pubbliche Amministrazioni. Dobbiamo far notare come in questo paese l'uso di software open source ha preso piede, seppur in maniera limitata, in assenza di incoraggiamenti formali del Governo.

---

<sup>154</sup> T. CARCERNAC, *Pour une administration électronique citoyenne*. Reperibile in : <http://www.internet.gouv.fr/francais/index.html>

<sup>155</sup> Il sito del progetto: <http://www.berlios.de>

Nel 2001 si attesta un bassissimo uso da parte di Belgio, Italia e Svezia (nonostante l'altissimo numero di gruppi locali di utenti Linux) di software a codice aperto nelle amministrazioni. Fino a questa data l'attenzione dei governi in questo ambito è stata praticamente nulla.

Lo studio IDA<sup>156</sup> fornisce una serie di indicazioni sulle possibili iniziative governative in favore dell'adozione di software libero/open source. Se da un lato il supporto diretto, attraverso finanziamenti pubblici, viene indicato come un'iniziativa controversa, viene fortemente consigliato il supporto indiretto. Esso consisterebbe nella promozione di standard aperti, strettamente legati al movimento open source, nel finanziamento indiretto degli sviluppatori attraverso i finanziamenti alla ricerca scientifica e in particolare nella creazione di un quadro legale che protegga le libertà degli sviluppatori. L'azione governativa dovrebbe concentrarsi nel limitare gli effetti negativi delle leggi riguardanti la proprietà intellettuale. Andrebbe soprattutto evitata l'introduzione dei brevetti sul software, che andrebbero a danneggiare il tessuto delle piccole imprese e in particolare il mondo del software libero/open source, ponendo delle barriere spesso insormontabili alla libertà degli sviluppatori.

Un sostegno indiretto al software libero è rappresentato anche da quelle iniziative tese a rinforzare gli effetti delle leggi sull'antitrust in presenza di casi di eccessiva dominanza.

Un esempio esemplare ci viene fornito dalla recente condanna alla Microsoft, confermata il 22 dicembre 2004, per abuso di posizione dominante. Oltre ad una ingente multa e all'obbligo di rilasciare una versione di Windows priva del lettore multimediale Windows Media Player, Microsoft è stata costretta a fornire, al fine di correggere la sua posizione, informazioni sui codici di interoperabilità che permettano ai software di altre

---

<sup>156</sup> INTERCHANGE OF DATA BETWEEN ADMINISTRATIONS, *Study into the use of Open Source Software in the Public Sector*, cit..

aziende di comunicare con Windows. La sentenza nasce da un lavoro di collaborazione tra il Tribunale UE insieme al Samba Team, costituito dagli sviluppatori di Samba<sup>157</sup>, e alla Free Software Foundation Europe, che hanno fornito la loro competenza tecnica e legale.

IDA suggerisce inoltre l'istituzione di un vasto archivio di pubblico accesso che collezioni il software creato dai governi per finalità interne, allo scopo di facilitare la nascita di versioni del software in uso migliorate e adattate alle esigenze più diverse.

Nel 2003 diversi paesi dell'Unione Europea hanno seguito la strada indicata dagli studi guidati dalla Commissione Europea e dagli esempi indicati da Francia e Germania.

La linea generalmente seguita da questi paesi, quali Finlandia, Slovenia, Danimarca, Olanda e Italia è quella di favorire e in alcuni casi imporre standard aperti per i documenti, richiedere la disponibilità dei sorgenti (quantomeno ad uso delle Pubbliche Amministrazioni, se non per tutti) e infine considerare l'open source un'alternativa valida al modello proprietario.

### 6.3 La situazione italiana

L'apertura di un dibattito italiano sul tema del software libero/open source per le Pubbliche Amministrazioni ha origine da una lettera aperta del 19 ottobre 2000 inviata al dipartimento della funzione pubblica e pubblicata su Interlex<sup>158</sup> all'Autorità per l'Informatica nella Pubblica Amministrazione (AIPA) e al Ministro del Tesoro intitolata "Soggezione informatica dello Stato italiano alla Microsoft<sup>159</sup>", in cui si denunciano

---

<sup>157</sup> Un software che permette il dialogo tra il sistema operativo della Microsoft e Gnu/Linux.

<sup>158</sup> [www.interlex.it](http://www.interlex.it)

<sup>159</sup> *Soggezione informatica dello Stato italiano alla Microsoft*, lettera aperta, 19/10/2000. Reperibile in: <http://www.interlex.it/pa/letterap.htm>

l'utilizzo massiccio all'interno del settore pubblico di software di un'unica azienda e l'ampio utilizzo di standard proprietari come se si trattasse di standard aperti.

Nel 2001 uno studio di Angelo Raffaele Meo del Politecnico di Torino, intitolato "Proposta di un programma nazionale di ricerca sul tema freeware"<sup>160</sup> pone allo scoperto la debolezza del settore informatico italiano e presenta il software libero come una risorsa per accelerare l'innovazione, per diminuire l'importazione di prodotti software e infine "accrescere quella cultura collettiva delle strutture pubbliche e private della ricerca che è stata, ed è, uno dei fattori di successo più importanti dei sistemi industriali dominanti"<sup>161</sup>.

Pochi mesi dopo, in linea con la proposta del senatore Millio, il senato redige una raccomandazione, accolta in sede di discussione della legge finanziaria del 2001, che mette in luce i possibili vantaggi derivanti dall'adozione di software open source, soprattutto in relazione alle alte spese degli investimenti informatici e chiede all'AIPA un'impegno all'esame e alla messa a punto di progetti per l'introduzione di questo tipo di software nelle PA.

Il 31 ottobre 2002 viene istituita dal Ministro per l'Innovazione e le Tecnologie la "Commissione per il software a codice sorgente aperto nella Pubblica Amministrazione", presieduta da Angelo Raffaele Meo.

Il lavoro della Commissione<sup>162</sup> viene alla luce nel maggio 2003 e si concretizza in numerose proposte. In primis la proposta di basarsi sul criterio del *value of money* al

---

<sup>160</sup> A. R. MEO, *Proposta di un programma nazionale di ricerca sul tema freeware*, 2000. Reperibile in: <http://www.interlex.it/pa/prognaz.htm>. Dobbiamo qui segnalare l'incorrettezza nel nominare il software libero/open source, di cui si parla all'interno del documento, con il termine freeware, con cui si indicano generalmente programmi gratuiti ma disponibili solamente come eseguibili e privi dei sorgenti.

<sup>161</sup> A. R. MEO, *Proposta di un programma nazionale di ricerca sul tema freeware*, cit.

<sup>162</sup> MINISTRO PER L'INNOVAZIONE E LE TECNOLOGIE, *Indagine conoscitiva sul software a codice sorgente aperto nella pubblica amministrazione. Rapporto della Commissione*, cit.

momento della scelta di soluzioni software per le PA, senza discriminazioni nei confronti dei prodotti open source. Va incoraggiato il riuso del software di tipo *custom* di proprietà non esclusiva delle PA e il trasferimento dei pacchetti tra PA. Inoltre, si propone l'istituzione dell'obbligatorietà dell'uso di almeno un formato aperto per i documenti prodotti dalle PA. Viene oltretutto messa in evidenza l'utilità del software open source per la distribuzione dei prodotti risultanti dai progetti di ricerca e innovazione tecnologica finanziati attraverso fondi pubblici.

In risposta allo studio della Commissione nel novembre 2003 viene emanata dal Ministro per l'Innovazione e le Tecnologie una direttiva per l'open source nella Pubblica Amministrazione. Si richiede alle PA di adottare criteri tecnici ed economici nella scelta del software da acquisire privilegiando i criteri di indipendenza da un singolo fornitore, della disponibilità dei sorgenti per l'ispezione, dell'interoperabilità, del riuso e della possibilità di esportare i documenti in più formati di cui almeno uno aperto. Nonostante la notevole portata della direttiva va osservato come questa non faccia menzione ai caratteri partecipativi del modello open source. Ci si concentra in particolare sui benefici economici e sulla necessità per le PA di conoscere i sorgenti e detenerne la proprietà. Si considerano in misura molto minore la trasparenza e le possibilità di modifica e riutilizzo da parte dei cittadini.

La trasparenza dei codici sorgenti ad uso delle PA non assicura che il software sia classificabile come open source. Per poter godere di questo appellativo il software deve permettere la visibilità del codice a chiunque desideri conoscerlo, siano essi privati o istituzioni, senza discriminazioni di alcun tipo.

Uno sviluppo pieno delle potenzialità del modello open source all'interno delle Pubbliche Amministrazioni implicherebbe non soltanto il controllo interno sui

programmi e la redistribuzione dei pacchetti tra PA ma la creazione di un vero e proprio rapporto sinergico e collaborativo tra Pubbliche Amministrazioni, piccole aziende produttrici di software, associazioni di telematica, gruppi di utilizzatori di software libero<sup>163</sup>, scuole, istituti di ricerca e singoli.

Un simile approccio è oltretutto incoraggiato dalle stesse linee guida per la “Società dell’Informazione e della Comunicazione”: il Consiglio Europeo di Lisbona del 2000, al fine di sostenere l’innovazione e dare vita ad una Società dell’Informazione aperta a tutti, auspica la costruzione di una società aperta all’innovazione attraverso l’incentivazione di un “dialogo aperto tra ricerca, impresa, Governo, e opinione pubblica”<sup>164</sup>.

Nelle Linee Guida del Governo per la Società dell’Informazione, inoltre, si aggiunge:

*“Un sistema di eGovernment nei suoi sviluppi più avanzati rappresenterà anche un potente strumento di coinvolgimento e partecipazione dei cittadini ai processi decisionali, evolvendo verso modelli innovativi di eDemocracy.”*<sup>165</sup>

Un sistema di governo elettronico volto a stimolare una piena partecipazione dei cittadini, oltre ad assicurare una completa accessibilità ai soggetti a maggiore rischio di esclusione, potrà avvalersi del modello open source per aprire un dibattito pubblico a partire dalle stesse scelte tecnologiche che sottostanno al suo sviluppo.

L’esperienza organizzativa del modello free software/open source potrebbe inoltre fornire gli strumenti intellettuali per il passaggio da una democrazia di tipo rappresentativo ad una sua forma più diretta.

---

<sup>163</sup> Segnaliamo tra gli altri i Linux User Groups e i Perl Mongers, una comunità di utilizzatori del celebre linguaggio di programmazione inventato da Larry Wall.

<sup>164</sup> MINISTRO PER L’INNOVAZIONE E LE TECNOLOGIE, *Linee guida del Governo per lo sviluppo della Società dell’Informazione nella legislatura*, Roma, giugno 2002.

<sup>165</sup> Ivi, p. 20.

Nella prospettiva indicata da Levy, l'utilizzo più proficuo delle nuove tecnologie telematiche non consisterebbe nel semplice snellimento e razionalizzazione della burocrazia ma nella sperimentazione di "forme di organizzazione o trattamento dell'informazione innovative, decentralizzate, più duttili e interattive"<sup>166</sup>, che permettano "una autentica *socializzazione delle soluzioni dei problemi*"<sup>167</sup>. In particolare l'autore sottolinea la differenza tra una politica molare, dove le soluzioni rispondono a una logica uniformante e centralizzata che appiattisce le differenze e una politica molecolare, in cui "maggioranze e minoranze (...) devono essere declinate al plurale perché non si riferiscono più a un *programma* di governo molare ma a *problemi* emergenti più o meno duraturi". Il ruolo delle minoranze non si ridurrebbe ad una parziale esclusione dai processi decisionali ma consisterebbe nella sperimentazione di soluzioni alternative.

Le esperienze di *open source intelligence* descritte nel quinto capitolo di questo lavoro, derivate direttamente dal modello di sviluppo open source, costituiscono un primo passo nell'affinamento degli strumenti utili alla costruzione di una sistema di governo più diretto e partecipato.

Sul piano pratico, la sensibilità verso il fenomeno del software libero/open source e la sua adozione da parte delle Pubbliche Amministrazioni sta gradualmente prendendo piede nel territorio italiano. In questo ambito, la provincia italiana che ha compiuto gli sforzi maggiori è Pisa. Indicata da molti come capitale italiana dell'open source, non si è limitata ad adottare questo tipo di software in molti dei propri computer, ma si è fatta promotrice dell'importante convegno "Open Source e Pubblica Amministrazione", svoltosi nel marzo 2003. Il convegno, che ha preso nell'anno successivo la forma di un

---

<sup>166</sup> P. LEVY, *L'intelligenza collettiva*, cit., p.74.

<sup>167</sup> Ibidem.

forum annuale intitolato “SALPA – Sapere Aperto e Libero nella Pubblica Amministrazione”, ha avuto il merito di occuparsi non soltanto degli aspetti tecnici della questione ma anche di far riflettere sugli aspetti sociali e formativi del fenomeno, giungendo ad occuparsi anche dell’open content (o open source intelligence).

In generale si può dire che la massima diffusione di software libero si ha nel campo dei server web, dei file server e dei database server, in cui le spese per la formazione dei dipendenti sono ridotte, in quanto queste applicazioni funzionano indipendentemente dall’utente finale. Sistemi web realizzati con tecnologie di tipo open source, quali server web (Apache), DBMS (Mysql, PostgreSQL) e Content Management Systems (Plone, Mambo, Open CMS, PostNuke, PhpNuke) sono stati adottati da numerosi comuni e province, come Ampezzo, Udine, Ferrara, Argenta, Padova, Cremona e Imperia.

La Provincia di Cremona si è spinta ancora oltre adottando il sistema operativo Linux, il programma di automazione d’ufficio Open Office, il programma per la posta elettronica Sendmail, Twiggi per il groupware e Putty per l’emulazione di terminale.

La rete telematica ProFeTa, che collega enti e uffici della Provincia di Ferrara, è stata realizzata con un grande apporto di tecnologie open source.

Tra i progetti di respiro più ampio citiamo Progettoconsiglio.it<sup>168</sup>, proposto dal Consiglio Comunale di Venezia, che oltre ad essere un sito di consulenza per l’introduzione delle nuove tecnologie informatiche nelle PA, ha realizzato un software di tipo open source per l’aggiornamento dei contenuti, richiesto da molti comuni dell’area.

La Provincia di Prato, da anni interessata al fenomeno del software libero/open source in virtù della sinergia con alcune associazioni telematiche operanti nel territorio, ha realizzato un servizio di protocollo open source liberamente disponibile in Internet.

---

<sup>168</sup> [www.progettoconsiglio.it](http://www.progettoconsiglio.it)

PA-Flow, nato dalla collaborazione tra l'AIPA e la Scuola Superiore S.Anna di Pisa, è un progetto per la realizzazione di un protocollo informatico per la gestione documentale.

Care2x, infine, nasce dall'iniziativa di Elpidio Latorilla ed è un software pensato per la gestione ospedaliera, progettato per l'integrazione di vari sistemi informativi all'interno di una unica piattaforma basata sul web.

Fornire un panorama veramente esaustivo delle esperienze in atto nelle province e nei comuni italiani non è obiettivo di questo lavoro, sia per la velocità dei cambiamenti in atto che per l'ampia prospettiva adottata: diremo comunque che l'adozione di tecnologie di tipo open source nel territorio italiano è in continua crescita ed è molto probabile che continui in questa direzione.

Il terreno più accidentato rimane, in Italia come altrove, quello del desktop, in cui le spese per la formazione dei dipendenti sono molto alte e possono costituire un ostacolo non indifferente.

## 7. Software libero/open source e Paesi in Via di Sviluppo

### 7.1 Il digital divide: un quadro introduttivo

Nell'affrontare il problema della penetrazione informatica nei Paesi in Via di Sviluppo (PVS) non possiamo non introdurre il concetto, ormai giunto alle orecchie di tutti, di *digital divide*. Con questo termine si fa riferimento al divario presente tra chi può e chi non può accedere (possedendone o meno le capacità) alle tecnologie informatiche. Esso è un fenomeno tipico non soltanto dei PVS, ma presente, anche se in forma meno accentuata, nei paesi che hanno visto una maggiore penetrazione delle tecnologie informatiche.

La questione del divario tecnologico può essere quindi suddivisa in due problematiche distinte:

- un *digital divide interno*, riguardante le differenze tra individui all'interno dello stesso paese. Questo tipo di divide si articola in diversi fattori. Il maggiore o minore accesso alle tecnologie informatiche e alla rete Internet varia in base al reddito (ma il divario è più ampio nei PVS), alla razza (fattore molto presente in Sudafrica), alla dislocazione territoriale, soprattutto nei termini di un forte scarto tra zone urbane e rurali, al livello di istruzione, all'età (in merito alla quale gli anziani sono generalmente svantaggiati), al sesso (in questo caso il gap dipende più dalle tendenze socio-culturali di ciascun paese che da ragioni legate al reddito) e all'handicap (che pone il problema dell'accessibilità degli artefatti informatici).

- un *digital divide internazionale*, riguardante le differenze in termini percentuali di penetrazione delle tecnologie informatiche e di alfabetizzazione informatica tra diversi paesi e generalizzabile, in maniera imprecisa, nel divario tra paesi del nord e del sud del mondo. Si distinguono a questo proposito tre tipologie di paesi: *paesi leader*, caratterizzati da una infrastruttura informatica matura e da alti livelli di interconnettività; *paesi adottanti*, caratterizzati da bassa interconnettività ma da un buon livello di interconnessione; *paesi ritardatari*, in cui il livello di interconnessione è talmente basso da non permettere un'incremento del livello di interconnettività.

In entrambi i casi si configura come un problema non meramente economico, ma anche e soprattutto riguardante l'ambito della formazione e dell'insegnamento.

Consapevoli del fatto che gran parte dei Paesi in Via di Sviluppo deve ancora affrontare il problema ben più urgente della soddisfazione dei bisogni primari (la fame, l'acqua, la sanità, l'istruzione, l'elettricità), crediamo che la questione del divario tecnologico sia di importanza cruciale in uno scenario globalmente interconnesso e in presenza di un grande cambiamento all'interno di molte società riguardante il campo produttivo e organizzativo, in buona parte indotto dall'introduzione delle tecnologie informatiche.

Parlare delle "nuove" tecnologie informatiche significa innanzi tutto parlare della capacità di queste di immagazzinare, catalogare, selezionare, trattare enormi quantità di informazioni, con evidente influenza in qualsiasi campo dell'agire umano.

Il divario tecnologico non si riassume semplicemente nell'accesso fisico alle tecnologie, in termini di presenza di computer e connettività disponibili per l'utilizzo da parte di tutti in tutte le aree geografiche, ma anche in termini di acquisizione di competenze

nell'uso degli strumenti, istanza che necessita di grossi investimenti di tempo e denaro nella formazione.

In questa sede non ci sarà possibile approfondire il tema complesso e variegato del digital divide: ci limiteremo ad evidenziare le prospettive tecniche, economiche e sociali derivanti dall'adozione di software libero/open source da parte dei PVS nell'ottica di una riduzione del divario tecnologico rispetto ai paesi maggiormente sviluppati.

## 7.2 Software libero/open source nei PVS: progetti e usi concreti<sup>169</sup>

Negli ultimi anni, sempre maggiore interesse è stato dimostrato da parte dei governi di alcuni paesi, in ritardo dal punto di vista dello sviluppo tecnologico, nei confronti dell'adozione di software libero/open source. Inoltre, numerosi investimenti in questo campo sono stati fatti dal settore privato, sia da operatori "puri" del software libero, quali le distribuzioni Red Hat, Suse e TurboLinux, sia da produttori di hardware e sviluppatori di DBMS, quali Hewlett-Packard, IBM, Sun, LG, Oracle e SAP.

In Brasile, su iniziativa del Presidente Luiz Ignacio Lula da Silva, è stata varata una raccomandazione che prevede l'adozione di software libero/open source in sostituzione al software proprietario da parte delle dipendenze del settore pubblico. Nel 2003, a questo proposito, è stato pubblicato un rapporto dal titolo "Free Software Implementation Guidelines" mirante a fornire indicazioni sulle modalità della migrazione. Nella città di Sao Paulo, il *Telecenter Project* mira alla creazione all'interno della città di un centinaio di centri telematici, ognuno dei quali dovrebbe essere in grado di servire circa tremila utenti, di libero accesso a tutti i cittadini.

---

<sup>169</sup> Una discreta parte delle informazioni fornite in questo paragrafo provengono da:  
P. DRAVIS, *Open Source Software. Perspectives for Development*, Information for Development Program (InfoDEV), The World Bank, Washington 2003. Reperibile in: <http://www.infodev.org>.

La “Electronic Government Initiative” alla quale sono affidati i compiti di creazione e gestione dei centri, conta nell’abbassamento dei costi derivanti dall’uso di piattaforme open source (Debian GNU/Linux e Linux Terminal Server) su un’infrastruttura hardware che prevede la presenza in ogni centro di un server e di venti terminali privi di hard disk. In questo caso, l’uso di tecnologia free/open source consente, oltre al taglio dei costi per l’acquisizione di licenze proprietarie, un’abbassamento dei prezzi complessivi attraverso il taglio delle spese per l’hardware.

Caratteristica interessante del progetto è la previsione di fornire costanti attività di formazione e assistenza. Sono infatti previsti corsi di formazione nell’ambito della grafica computerizzata, nella creazione di sistemi web, alfabetizzazione informatica di base e l’istituzione di un team tecnico di supporto al servizio degli utenti. E’ incoraggiata la produzione di articoli sulla vita della comunità a cui il centro fa riferimento e la creazione di materiale di supporto per l’apprendimento e la creazione di nuove prospettive occupazionali.

Numerose iniziative si stanno affacciando anche nel sud-est asiatico. In India, la *Linux India Initiative*, promossa dal governo, si sta occupando di organizzare gruppi di studio sul software libero/open source, centri di ricerca e gruppi di supporto all’adozione. Il Goa Schools Computer Project (GSCP), nato nel 1996, si è occupato della messa in funzione, all’interno delle scuole dell’isola, di circa quattrocento computer riciclati dotandoli di software completamente libero/open source, risparmiando sull’acquisto delle licenze e potendo godere del supporto dei Linux User Group locali. In Malesia è stato istituito un centro di riferimento per l’open source con compiti di assistenza e supporto all’adozione, alla certificazione del software e alla formazione. E’ inoltre in

previsione lo stanziamento di fondi per compagnie informatiche indipendenti che si occupano di software a codice aperto.

Nelle Filippine il governo si è impegnato nella messa a punto di pacchetti di software libero/open source disegnati sulla base delle esigenze locali. Tra questi, una versione semplificata di Linux chiamata Bayanihan Linux. In Thailandia, il Ministro per l'ITC ha esplicitamente incoraggiato l'uso di software libero/open source nelle agenzie amministrative.

Cina, Giappone e Sud Corea hanno annunciato una collaborazione, da attuarsi insieme al settore privato, mirante allo sviluppo di sistemi software *embedded*<sup>170</sup> basati su Linux per l'uso all'interno di cellulari, macchine digitali e sistemi di navigazione satellitare.

In Sudafrica, attraverso un'approccio simile a quello adottato dall'Unione Europea, si raccomanda, mediante il rapporto governativo "Using Open Source Software in the South African Government"<sup>171</sup>, l'adozione di criteri economici e di merito nella scelta tra software libero e proprietario, suggerendo di preferire l'utilizzo di software libero quando il bilancio tra i vantaggi e gli svantaggi diretti tra software proprietario e aperto sia da considerarsi equiparabile. Si sottolineano inoltre i vantaggi del software open source in merito all'indipendenza dai fornitori e in relazione alla maggiore sicurezza.

In Tajikistan, l'assenza di traduzioni delle distribuzioni proprietarie nel linguaggio locale ha stimolato la nascita di un progetto, guidato da Khujand Computer Technologies, teso allo sviluppo di una traduzione della distribuzione Mandrake Linux che potrà in seguito essere utilizzata nelle scuole, nelle amministrazioni e nel settore privato.

---

<sup>170</sup> Software che viene incorporato all'interno di utensili di uso comune o strumenti per l'automazione.

<sup>171</sup> GOVERNMENT INFORMATION TECHNOLOGY OFFICERS' COUNCIL, *Using Open Source Software in the South African Government*. Reperibile in:  
[http://www.oss.gov.za/docs/OSS\\_Strategy\\_v3.pdf](http://www.oss.gov.za/docs/OSS_Strategy_v3.pdf)

### 7.3 Software libero/open source: opportunità per i PVS

In vantaggi derivanti dall'adozione di software libero/open source nei Paesi in Via di Sviluppo sono pressapoco gli stessi che si produrrebbero in paesi caratterizzati da una maggiore informatizzazione, ma crediamo che possano essere in qualche modo più consistenti, anche se variabili da situazione a situazione.

Un primo vantaggio consiste nella facilità di accesso al software a costi ridotti. Il software libero/open source è disponibile in forma gratuita e non necessita il pagamento per ogni copia concessa in licenza. I prodotti possono essere scaricati direttamente da Internet, convertiti in formato CD e ridistribuiti nel paese al solo costo del supporto. I costi per l'accesso ai prodotti software si abbassano notevolmente, permettendo la concentrazione degli sforzi monetari sulla customizzazione e l'adattamento alle esigenze locali, sulla formazione e sui servizi.

Lo stesso vale nel caso di finanziamenti da parte di enti pubblici o privati nei confronti di questi paesi: il trasferimento tecnologico può essere effettuato a minore costo, permettendo una maggiore effettività dei fondi risparmiando nell'acquisizione delle licenze. Nei paesi a maggiore penetrazione informatica il Total Cost of Ownership è costituito da un'alta percentuale di costi di formazione del personale dovuti alla migrazione ad una nuova tecnologia software e spesso questo fattore è decisivo nel determinare la scelta di continuare a rivolgersi allo stesso fornitore proprietario e a non effettuare la migrazione. Nei PVS a bassa penetrazione informatica questo fattore ha meno peso nel rapporto tra software a codice aperto e software proprietario, in quanto i costi di formazione, nei casi diffusissimi di analfabetismo informatico, sarebbero uguali per entrambe le soluzioni.

Un aspetto molto importante è la possibilità per i PVS di avere un accesso diretto alla tecnologia software e non soltanto al suo utilizzo. L'accesso al software è infatti molto vantaggioso, sia in termini educativi che in termini di diritti da pagare in caso di utilizzo del software per la creazione di prodotti derivati.

Il modello free software/open source è in grado di ridurre lo scarto tra paesi sviluppati e non, attraverso l'accesso a costi molto ridotti di tecnologie avanzate di alta qualità, creando una base comune dalla quale tutti possono partire per costruire nuove risorse. Inoltre vi è la possibilità di legarsi ad un circuito internazionale di sviluppatori attraverso il quale i PVS potrebbero beneficiare di collaborazioni su progetti specifici e garantirsi una maggiore visibilità a livello mondiale delle aziende ICT locali.

Ovviamente la riduzione del divario sarebbe solo relativamente notevole, in quanto continuerebbero a pesare le carenze infrastrutturali (elettricità, telecomunicazioni, hardware, connettività) e formative.

Il software libero/open source è inoltre indicato per quelle zone dei Paesi in Via di Sviluppo in cui per questioni demografiche l'assenza di una scalabilità dei costi di sviluppo software costituisce un deterrente alla creazione di prodotti studiati per le esigenze di quei territori. Questo fenomeno si verifica spesso nel caso delle traduzioni. L'esempio del Tajikistan, citato nel paragrafo precedente, ne è un esempio lampante.

Il software a codice aperto si rivela quindi ottimale per la riduzione del gap soltanto in alcuni dei numerosi aspetti in cui si articola la questione del divario tecnologico: permette la riduzione dei costi per le licenze software, che costituisce soltanto una minima parte dei costi complessivi (tra i quali citiamo i costi per la formazione, i servizi, l'hardware e l'accesso a Internet), la riduzione dei costi di aggiornamento dei

programmi, la localizzazione dei programmi(adattamento dei programmi alle esigenze locali) e la loro traduzione.

Sembra in particolare indicato a contrastare l'obsolescenza dei programmi: nel mondo del software proprietario le aziende tendono ad aggiornare spesso i programmi smettendo di supportare le versioni precedenti. Questo rappresenta generalmente un problema per molti PVS spesso dotati di macchine superate. Grazie all'apertura del codice sorgente, il *porting* di un programma su di una macchina datata è sempre possibile.

In merito a questo è importante citare il progetto RULE (Run Up to-date Linux Everywhere)<sup>172</sup>, che mira allo sviluppo di una distribuzione basata su Red Hat pensata per essere particolarmente leggera in modo di potersi adattare alle macchine con bassa CPU e poca RAM.

Un'aspetto particolarmente importante del modello free software/open source è di carattere economico. Nel rapporto della Digital Opportunity Task Force<sup>173</sup> si sottolinea l'importanza del settore ITC nel creare opportunità di sviluppo, attraverso un migliorato accesso alle informazioni e più ampie capacità di comunicazione.

Il modello free software/open source restituirebbe ai paesi adottanti maggiore autonomia e padronanza locale nel settore ICT. Diversamente dal software proprietario, legatissimo alle istanze del mercato, il software libero garantisce una maggiore indipendenza da esso, stimolando maggiormente lo sviluppo di soluzioni destinate alle esigenze locali. Potrebbe inoltre permettere la nascita di un settore ICT autonomo, riducendo le spese di importazione di software dai paesi leader.

---

<sup>172</sup> <http://www.rule-project.org/en/>

<sup>173</sup> Digital Opportunity Task Force, *Digital Opportunities for All: Meeting the Challenge*, 2001

La nascita di Linux User Groups locali, caratteristici in presenza di una buona diffusione del software libero/open source, costituirebbe una ulteriore opportunità per creare dei legami di tipo locale/globale tra associazioni, singoli, privati e istituzioni e potrebbe rivelarsi importante a scopo formativo.

## 8. Conclusioni

Nel 1991 Linus Torvalds, studente universitario di Helsinki, decide di mettersi al lavoro per creare un sistema operativo “libero” di tipo Unix che possa funzionare sul suo pc 386.

Le motivazioni che lo spingono in questa direzione sono svariate: la necessità di disporre di un sistema operativo su cui studiare senza spendere troppo per pagare le licenze di un sistema Unix proprietario, il desiderio di mettere in pratica le proprie abilità di progettista di sistemi operativi imparando “sul campo”, la voglia di mettersi alla prova e soprattutto di divertirsi in una delle attività per lui più stimolanti: la programmazione.

Inizia il suo lavoro partendo dal codice sorgente di Minix, il kernel di un semplice sistema operativo sviluppato dal professor Tanenbaum e rilasciato in forma “libera” per gli studenti.

Decide innanzi tutto di non fare tutto da solo, ma di appoggiarsi alla collaborazione di tutti i possibili interessati al progetto. E' così che, sul newsgroup di Minix, dà l'annuncio del suo progetto agli utilizzatori di quest'ultimo, iniziando fin da subito a rilasciare alcune parti di codice e invitando gli interessati a collaborare correggendo i bug del codice sottoposto e aggiungendovi funzionalità.

Alcuni sviluppatori, incuriositi, iniziano a prendere parte al progetto, lavorando al codice del software e dando dei suggerimenti. Il progetto inizia ad essere seguito da un buon numero di persone, tanto che in breve Torvalds si toglie dal newsgroup Minix e ne crea uno interamente dedicato al suo progetto: alt.os.linux.

Lo stile di leadership di Linus Torvalds, al tempo unico manutentore del progetto, è esemplare: risponde velocemente ai commenti e alle proposte ricevute, si affretta a correggere e a rispedire a tutti i partecipanti al newsgroup le patch di debug ricevute e le nuove aggiunte, cercando di tenere vivo l'interesse al progetto, stimolando gli sviluppatori e responsabilizzandoli.

Nel dicembre 2001 esce la versione 0.11 del kernel Linux, la prima funzionante autonomamente e a partire dalla 0.12 viene rilasciato sotto licenza GNU GPL, ideata da Richard Stallman.

Da quel momento in poi il kernel Linux, insieme agli applicativi GNU, attirerà sempre maggiore interesse e verrà sviluppato da un numero sempre maggiore di sviluppatori, che potranno lavorare a parti di codice senza interferire gli uni con gli altri grazie al requisito della modularità del codice sorgente, fortemente voluto da Torvalds.

Lo sviluppo seguente di Linux è incrementato in proporzione diretta con la diffusione di Internet e la sua evoluzione ha riflesso in buona sostanza le istanze di tutti gli attori che hanno gradualmente iniziato a prendere parte al suo sviluppo, dai singoli programmatori, alle aziende, alle istituzioni.

La storia della nascita di Linux, nel suo piccolo, rappresenta una sintesi degli aspetti più interessanti del fenomeno del software libero/open source, dalle sue caratteristiche più feconde ai suoi limiti.

Ci parla di come nascono i progetti: dalla necessità di risolvere un problema individuale, dal piacere di programmare e dallo stimolo intellettuale derivante dall'impegno in un progetto ambizioso.

Ci parla anche dei diversi fattori necessari a garantire ad un progetto il raggiungimento dello stato di stabilità. Un delicato equilibrio fatto di strumenti legali, atti a preservare la

proprietà collettiva degli artefatti; di architettura del codice, che attraverso la modularità permette il lavoro individuale degli sviluppatori senza interferenze reciproche; di regole di comportamento informali, che permettano l'innescò del meccanismo della reputazione tra pari attraverso lo scambio di "doni" dei quali viene resa nota la paternità a tutta la comunità, e che donino visibilità a quegli sviluppatori che considerano i progetti di software libero/open source una rampa di lancio verso il mondo del lavoro; di accortezza dei manutentori, che devono saper trattare i propri utilizzatori come delle risorse, facendoli sentire costantemente importanti per permettere il passaggio dalla partecipazione periferica dei semplici utilizzatori allo status di manutentori, attraverso un apprendimento graduale che prevede l'acquisizione di capacità tecniche e relazionali. Alla base di questo equilibrio vi è l'infrastruttura di Internet, con la quale il movimento del software libero/open source ha da sempre costruito un rapporto sinergico. Senza la rete, che abbatta i costi - sia monetari che in termini di tempo - del donare e permette l'incontro di persone di ogni parte del mondo legate da interessi affini, il modello organizzativo open source non avrebbe mai visto la luce. D'altra parte la rete, privata dell'apporto di quest'ultimo, che ha stimolato l'ampia adozione di standard aperti e il fiorire di tecnologie multiplatforma contrastando la frammentazione in sistemi proprietari, non sarebbe come la conosciamo: flessibile, aperta, distribuita.

Dalla storia della nascita di Linux capiamo anche che il movimento free software/open source ha rappresentato, fin dalle sue origini, un movimento di programmatori per programmatori, in cui la partecipazione al lavoro collettivo veniva messa in atto da persone con forti competenze informatiche e aveva il principale scopo di costruire strumenti per creare strumenti o programmi che risolvessero problematiche avanzate non ancora disponibili sul mercato. Dai primi tempi ad oggi abbiamo assistito ad una

apertura della comunità ad un numero più ampio di attori, come testimonia la nascita di software gestionali per le aziende, di programmi di editing audio/video, dedicati ai designers e di numerosi programmi rivolti all'utente finale, con un notevole incremento dello stato di usabilità di quest'ultimi.

Nel chiederci se l'applicazione del modello free software/open source possa stimolare una maggiore consapevolezza ed un rapporto più partecipativo nei confronti degli strumenti telematici e possa essere estesa ad altri campi oltre a quello del software, suggeriamo che una variabile è sicuramente rappresentata dalla capacità delle comunità free software/open source di aprirsi il più possibile ad altri gruppi socio-professionali, dando forma ad un dialogo interdisciplinare, creando una sorta di sistema a vasi comunicanti che informi costantemente il pubblico più ampio delle istanze portate avanti dalla comunità stessa e, viceversa, permetta un ritorno alla comunità dei tecnici/sviluppatori delle esigenze degli utenti finali.

La nascita di movimenti affini al software libero/open source ma legati ad un'ambito meno tecnico, indicati con il nome di "open source intelligence" o "open content" va in questa direzione, lasciando aperti numerosi quesiti su come ri-articolare la forza organizzativa del modello in questione con le caratteristiche di questo tipo di artefatti.

In conclusione dobbiamo evidenziare che il fenomeno del software libero/open source, se da un lato offre dei grandi vantaggi relativi alla sicurezza, alla trasparenza, all'interoperabilità, alla promozione di standard aperti, comporta anche dei rischi: uno fra tutti la possibilità, in un contesto di proliferazione di soluzioni applicative adattate a contesti diversi, che venga a crearsi una situazione di dispersione e confusione dovuta alla mancanza di infrastrutture che permettano agli interessati il ripperimento delle soluzioni adeguate.

In questo contesto si rivela importantissimo il ruolo delle istituzioni nel promuovere un'utilizzo consapevole alle tecnologie informatiche e nel creare delle strutture che sappiano informare e costituirsi come guide per la comprensione di un fenomeno variegato, complesso, in costante evoluzione.

Questa tesi ha cercato di rincorrere il fenomeno cercando di focalizzarne i punti chiave, sperando che possa contribuire alla creazione di una base solida per la ricerca futura.

## Bibliografia<sup>174</sup>

- ABOBA B., CERF V., 1993. *How the Internet Came to Be*. Reperibile in:  
<http://www.internetvalley.com/archives/mirrors/cerf-how-inet.txt>
- ADAMS J., PARE' D., TANG P., 2001. *Patent protection of computer programmes. Final report.*, ECSC-EC-EAEC, Brussels. Reperibile in:  
<http://swpat.ffii.org/vreji/minra/sisku/>
- AMADORI G., BAZZIGALUPPI G., BERNOCCHI W., GATTI M., 2002. *Sistemi operativi "Open Source": il caso Linux*, in *Mondo Digitale*, numero 1. Reperibile in:  
[http://www.mondodigitale.net/Rivista/02\\_numero\\_uno/Bazzigaluppi.pdf](http://www.mondodigitale.net/Rivista/02_numero_uno/Bazzigaluppi.pdf)
- AUTORITA' PER L'INFORMATICA NELLA PUBBLICA AMMINISTRAZIONE (AIPA), 2002. *Il software Open Source (OSS) scenario e prospettive*, i Quaderni, n°10, giugno. Reperibile in:  
[http://www.innovazione.gov.it/ita/intervento/normativa/allegati/oss\\_scenario.pdf](http://www.innovazione.gov.it/ita/intervento/normativa/allegati/oss_scenario.pdf)
- BADIALI B., *La storia di Unix. Come nacque e crebbe Unix e i suoi discendenti*. Reperibile in: [http://digilander.libero.it/bunix/it\\_hist.html](http://digilander.libero.it/bunix/it_hist.html)
- BALDWIN C.Y., CLARK K.B., 2003. *The Architecture of Cooperation: How Code Architecture Mitigates Free Riding in the Open Source Development Model*, Harvard Business School. Reperibile in: <http://opensource.mit.edu/papers/baldwinclark.pdf>
- BARBROOK R., 1998. *The High-Tech Gift Economy*, in *First Monday*, volume 3, number 12 (December). Reperibile in:  
[http://www.firstmonday.org/issues/issue3\\_12/barbrook/index.html](http://www.firstmonday.org/issues/issue3_12/barbrook/index.html)
- BATES J., LAKHANI K.R., WOLF B., 2002. *The Boston Consulting Group Hacker Survey*. The Boston Consulting Group. Disponibile in:  
<http://www.bcg.com/opensource/BCGHACKERSURVEY.pdf>

---

<sup>174</sup> Questa bibliografia contiene numerosi articoli tratti da Internet. Avvertiamo quindi che gli indirizzi qui indicati potrebbero in futuro non funzionare più o portare ad altre versioni (aggiornate rispetto a quelle a cui ci riferiamo) degli articoli. Dove è stato possibile, abbiamo provveduto ad annotare la versione del file.

BESSEN J., HUNT R.M., 2004. *An Empirical look at software patents*. Reperibile in:  
<http://swpat.ffii.org/vreji/minra/sisku/>

BEZROUKOV N., 1999. *Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism)*, *First Monday*, 04 (11).  
Reperibile in: [http://www.firstmonday.dk/issues/issue4\\_10/bezroukov/](http://www.firstmonday.dk/issues/issue4_10/bezroukov/).

BONACCORSI A., ROSSI C., 2001. *L'Economia degli Standard e la Diffusione delle Tecnologie. L' Open Source non è un Assurdo Economico*, Pisa, Laboratorio di Economia e Management, Scuola di Studi Superiori Sant'Anna. Reperibile in:  
[http://www.dvara.net/HK/I2001-02\\_0.pdf](http://www.dvara.net/HK/I2001-02_0.pdf)

BOSCAROL M., 2003. *Ecologia dei siti web. Come e perché usabilità, accessibilità e fogli di stile stanno cambiando il modo di realizzare siti internet*, Tecniche Nuove, Milano.

BLISSET L., 2000. *Totò, Peppino e la guerra psichica 2.0*, Einaudi, Torino.

CAMP L.J., SYME S., 2001. *The Governance of Code: Code as Governance*, Social Science Research Network Electronic Papers Collection,. Reperibile in:  
<http://papers.ssrn.com/abstract=297154>.

CARCERNAC T., *Pour une administration électronique citoyenne*. Reperibile in :  
<http://www.internet.gouv.fr/francais/index.html>.

CARLINI F., 1996. *Internet, Pinocchio e il Gendarme. Le prospettive della democrazia democrazia in rete*, Manifestolibri, Roma.

CARLINI F., 2002. *Divergenze digitali. Conflitti, soggetti e tecnologie della terza Internet*, Manifestolibri, Roma.

CASTELLS M., 2001. *Galassia Internet*, Feltrinelli, Milano.

CASTELLS M., 2002. *La nascita della società in rete*, ed. it., Università Bocconi Editore, Milano.

CEDERGREN M., 2003. *Open Content and Value Creation*, in *First Monday*, volume 8, number 8 (August). Reperibile in:

[http://firstmonday.org/issues/issue8\\_8/cedergren/index.html](http://firstmonday.org/issues/issue8_8/cedergren/index.html)

CERF V.(a cura di), 2002. *A Brief History of the Internet, version 3.32*, Internet Society.

Reperibile in: <http://www.isoc.org/internet/history/brief.shtml>.

CIFFOLILLI A., 2003. *Phantom authority, self-selective recruitment and retention of members in virtual communities: The case of Wikipedia*, in *First Monday*, volume 8, number 12 (December). Reperibile in:

[http://firstmonday.org/issues/issue8\\_12/ciffolilli/index.html](http://firstmonday.org/issues/issue8_12/ciffolilli/index.html)

CUKIERMAN H.L., PINHEIRO A.S., 2004. *Free software: Some Brazilian translations*, in *First Monday*, volume 9, number 11 (November). Reperibile in:

[http://firstmonday.org/issues/issue9\\_11/pinheiro/index.html](http://firstmonday.org/issues/issue9_11/pinheiro/index.html)

DAFERMOS G.N., 2003. *Blogging the Market. How Weblogs are turning corporate machines into real conversations*. Reperibile in:

[http://radio.weblogs.com/0117128/Blogpaper/blogging\\_the\\_market.html](http://radio.weblogs.com/0117128/Blogpaper/blogging_the_market.html)

DEBORD G., 2001. *La società dello spettacolo*, ed. it., Baldini&Castoldi, Milano.

DELIO M., 2003. *Developing World Needs Linux*. Reperibile in:

<http://www.wired.com/news/politics/0,1283,59334,00.html>

DIAMOND D., TORVALDS L., 2001. *Rivoluzionario per caso: come ho creato Linux (solo per divertirmi)*, ed. it., Garzanti, Milano.

DI BONA C., OCKAM S., STONE M. (a cura di), 1999. *Open Sources. Voci dalla rivoluzione Open Source*, ed. it., Apogeo, Milano.

DECI E.L., RYAN R.M., 2000. *Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions*, in *Contemporary Educational Psychology*, January, vol. 25, no. 1, Elsevier Science. Reperibile in:

<http://www.psych.rochester.edu/SDT/publications/documents/2000RyanDeciIntExtdefs.pdf>.

DRAVIS P., 2003. *Open Source Software. Perspectives for Development*, Information for Development Program (InfoDEV), The World Bank, Washington. Reperibile in: <http://www.infodev.org>.

DUTTON W.H., 2001. *La società online*, ed. it., Milano, Baldini&Castoldi.

EDWARDS K., 2000. *When Beggars Become Choosers*, in *First Monday*, volume 5, number 10 (October). Reperibile in: [http://firstmonday.org/issues/issue5\\_10/edwards/index.html](http://firstmonday.org/issues/issue5_10/edwards/index.html)

FIELDING R.T., HERBLEB J.D., MOCKUS A., *Two Case Studies of Open Source Software Development: Apache and Mozilla*, Avaya Labs Research 2002. Reperibile in: <http://www.research.avayalabs.com/techreport/ALR-2002-003-paper.pdf>

FILENI F., 1998. *Comunicazione cultura conoscenza*, Ed. Goliardiche, Trieste.

FILENI F., 2003. *Analogico e digitale*, Ed. Goliardiche, Trieste.

FLICHY P., 2001. *L'innovazione tecnologica. Le teorie dell'innovazione di fronte alla rivoluzione digitale*, ed. it., Feltrinelli, Milano.

FOPPA S.U., 2002. *Digital Divide, azioni e percorsi per combatterlo*, in *Mondo digitale*, numero2. Reperibile in: [http://www.mondodigitale.net/Rivista/02\\_numero\\_due/Foppa.pdf](http://www.mondodigitale.net/Rivista/02_numero_due/Foppa.pdf)

FREE SOFTWARE FOUNDATION, *Free Software Definition*. Reperibile in: <http://www.gnu.org/philosophy/free-sw.html>.

FREE SOFTWARE FOUNDATION, 2001. *Cos' è il software libero?*, ed. it., Boston. Reperibile in: <http://www.gnu.org/philosophy/free-sw.it.html>.

FREE SOFTWARE FOUNDATION, 2001. *Licenze varie e commenti relativi*, ed. it., Boston. Reperibile in <http://www.gnu.org/licenses/license-list.it.html>

GARNHAM N., 2001. *Information Society Theory as Ideology: a Critique*, in *Studies in Communication Sciences* 1.

- GIORELLO G., 1996. *Un mondo senza centro nella rete delle reti*, in *Telemà 3*.  
Reperibile in: <http://www.fub.it/telemà/TELEMA3/Giorell3.html>.
- GOSH R.A., GLOTT R., KRIEGER B., ROBLES G., 2002. *Free/Libre and Open Source Software: Survey and Study (FLOSS)*, International Institute of Infonomics, University of Maastricht. Reperibile in: <http://www.infonomics.nl/FLOSS/report/>.
- GOVERNMENT INFORMATION TECHNOLOGY OFFICERS' COUNCIL, *Using Open Source Software in the South African Government*. Reperibile in:  
[http://www.oss.gov.za/docs/OSS\\_Strategy\\_v3.pdf](http://www.oss.gov.za/docs/OSS_Strategy_v3.pdf)
- GREEN E.L., *Economics of Open Source Software*. Reperibile in:  
<http://badtux.org/home/eric/editorial/economics.php>
- HAUBEN M., *The history of ARPA leading up to the ARPANET*. Reperibile in:  
<http://www.deip.isep.ipp.pt/docs/arpa--1.html>
- HECKER F., 2000. *Setting Up Shop: The Business of Open-Source Software*. Reperibile in: <http://www.hecker.org/writings/setting-up-shop.html>.
- HIRSH J., STALDER F., 2002. *Open Source Intelligence*, in *First Monday*, volume 7, number 6 June. Reperibile in:  
[http://firstmonday.org/issues/issue7\\_6/stalder/index.html](http://firstmonday.org/issues/issue7_6/stalder/index.html).
- HIMANEN P., 2001. *L'etica hacker e lo spirito dell'età dell'informazione*, ed. it., Feltrinelli, Milano.
- HOFFMAN N., *Open Source Software*. Reperibile in:  
<http://public.kitware.com/VTK/pdf/oss.pdf>
- IBM, *History Flow*. Reperibile in:  
<http://researchweb.watson.ibm.com/history/index.htm>
- INTELLECTUAL PROPERTY INITIATIVE, 2000. *Background and Overview of the Intellectual Property Initiative*. Reperibile in:  
<http://info.sm.umist.ac.uk/esrcip/background.htm>

INTERCHANGE OF DATA BETWEEN ADMINISTRATIONS (IDA), 2001. *Study into the use of Open Source Software in the Public Sector*, European Commission, DG Enterprise.

ISO 9241, 1997. *Ergonomic Requirements for office work with visual display*.

JESIEK B.K., 2003. *Democratizing software: Open source, the hacker ethic, and beyond*, in *First Monday*, volume 8, number 10 (October). Reperibile in:  
[http://firstmonday.org/issues/issue8\\_10/jesiek/index.html](http://firstmonday.org/issues/issue8_10/jesiek/index.html)

KELTY C.M., 2001. *Free Software/Free Science*, in *First Monday*, volume 6, number 12 (December). Reperibile in:  
[http://firstmonday.org/issues/issue6\\_12/kelty/index.html](http://firstmonday.org/issues/issue6_12/kelty/index.html).

KRISHNAMURTHY S., 2002. *Cave or Community? An Empirical Examination of 100 Mature Open Source Projects*, in *First Monday*, volume 7, number 6 (June). Reperibile in: [http://firstmonday.org/issues/issue7\\_6/krishnamurthy](http://firstmonday.org/issues/issue7_6/krishnamurthy)

LERNER J., TIROLE J., 2000. *The Simple Economics of Open Source*, Social Science Research Network Electronic Paper Collection. Reperibile in:  
<http://ssrn.com/abstract=224008>

LEVY P., 1996. *L'intelligenza collettiva. Per un'antropologia del cyberspazio*, ed. it., Feltrinelli, Milano.

LEVY P., 1999. *Cybercultura*, ed. it., Feltrinelli, Milano.

LEVY S., 2002. *Hackers. Gli eroi della rivoluzione informatica*, ed. it., ShaKe Edizioni Underground, Milano.

MADANMOHAN T.R., NAVELKAR S., *Roles and Knowledge Management in Online Technology Communities: An Ethnography Study*, Reperibile in:  
<http://www.inderscience.com/storage/f921216851137104.pdf>

MANTOVANI G., 1995. *L'interazione uomo-computer*, Il Mulino, Bologna.

MEYROWITZ J., 1993. *Oltre il senso del luogo. L'impatto dei media elettronici sul comportamento sociale*, ed. it., Baskerville, Bologna.

MEO A.R., 2000. *Proposta di un programma nazionale di ricerca sul tema freeware*.  
Reperibile in: <http://www.interlex.it/pa/prognaz.htm>.

MEO A.R., 2002. *Software libero e "open source"*, in *Mondo Digitale*, n°2.  
Reperibile in: [http://www.mondodigitale.net/Rivista/02\\_numero\\_due/Meo.pdf](http://www.mondodigitale.net/Rivista/02_numero_due/Meo.pdf)

MINISTRO PER L'INNOVAZIONE E LE TECNOLOGIE, 2002. *Linee guida del Governo per lo sviluppo della Società dell'Informazione nella legislatura*, Roma.  
Reperibile in: [http://www.innovazione.gov.it/ita/documenti/socinfo11\\_06\\_02.pdf](http://www.innovazione.gov.it/ita/documenti/socinfo11_06_02.pdf)

MINISTRO PER L'INNOVAZIONE E LE TECNOLOGIE, 2003. *Indagine conoscitiva sul software a codice sorgente aperto nella pubblica amministrazione*, Roma. Reperibile in:  
[http://www.innovazione.gov.it/ita/comunicati/open\\_source/indagine\\_commissione\\_oss.pdf](http://www.innovazione.gov.it/ita/comunicati/open_source/indagine_commissione_oss.pdf).

MOODY G., 2002. *Codice ribelle*, ed. it., Hops Libri, Milano.

NEGROPONTE N., 1995. *Essere digitali*, ed. it., Sperling & Kupfer Editori, Milano.

NICHOLS D.M., TWIDALE M.B., 2002. *Usability and Open Source Software*, University of Waikato. Reperibile in <http://www.cs.waikato.ac.nz/~daven/docs/oss-wp.html>.

OPEN GROUP, 1995. *Unix past*. Reperibile in:  
[http://www.unix.org/what\\_is\\_unix/history\\_timeline.html](http://www.unix.org/what_is_unix/history_timeline.html)

OPEN SOURCE INITIATIVE, *Approved licenses*. Reperibile in:  
<http://www.opensource.org/licenses/>

OPEN SOURCE INITIATIVE, 2004. *Open Source Case for Business*. Reperibile in:  
[http://www.opensource.org/advocacy/case\\_for\\_business.php](http://www.opensource.org/advocacy/case_for_business.php).

PAVLICEK R.C., 1999. *Keys to Effective Linux Advocacy Within Your Organization*.  
Reperibile in: [http://opensource.hp.com/the\\_source/linux\\_papers/als-fullpaper-1999.html](http://opensource.hp.com/the_source/linux_papers/als-fullpaper-1999.html).

- PERENS B., *The Debian Free Software Guidelines (DFSG)*. Reperibile in:  
[http://www.debian.org/social\\_contract.html](http://www.debian.org/social_contract.html).
- PERENS B., 1997. *The Open Source Definition*. Reperibile su:  
<http://www.opensource.org/docs/definition.php>
- PRESS L., 2004. *The Internet in developing nations: Grand challenges*, in  
First Monday, volume 9, number 4 (April). Reperibile in:  
[http://firstmonday.org/issues/issue9\\_4/press/index.html](http://firstmonday.org/issues/issue9_4/press/index.html)
- RAYMOND E.S., 1998. *La cattedrale e il bazaar*, ed. it., Open Press. Reperibile in:  
<http://www.apogonline.com/openpress/doc/cathedral.html>.
- RAYMOND E.S., 1998. *Colonizzare la noosfera*, ed. it., Open Press. Reperibile in:  
<http://www.apogonline.com/openpress/doc/homesteading.html>.
- RAYMOND E.S., 1999. *Il calderone magico*, ed. it., Open Press. Reperibile in:  
[http://www.apogonline.com/openpress/doc/...](http://www.apogonline.com/openpress/doc/)
- RAYMOND E.S., 1999. *A response to Nikolai Bezroukov*, First Monday volume 4,  
number 11, November. Reperibile in:  
[http://firstmonday.org/issues/issue4\\_11/raymond/index.html](http://firstmonday.org/issues/issue4_11/raymond/index.html).
- RAYMOND E.S., 2002. *Come diventare un hacker*, ed. it. Reperibile in:  
<http://www.autistici.org/hackarena/etica/jargon.htm>
- RAYMOND E.S. (a cura di), 2003. *The Jargon File version 4.4.7*. Reperibile in  
<http://www.catb.org/~esr/jargon/>.
- REAGLE J.M., 2003. *Socialization in Open Technical Communities*. Reperibile in:  
<http://reagle.org/joseph/2003/socialization/voluntary.html>
- REIJSWOUD V., TOPI C., *Alternatives Routes in the Digital World*. Reperibile in:  
<http://opensource.mit.edu/papers/reijswoudtopi.pdf>
- RHEINGOLD H., 1994. *Le comunità virtuali*, ed. it., Sperling & Kupfer, Milano.

RIFKIN J., 2000. *L'era dell'accesso. La rivoluzione della new economy.*, ed. it., Arnoldo Mondadori Editore, Milano.

ROBLES G., SHEIDER E., TRETROWSKY I., WEBER N., 2001. *Who is doing it?(WIDI) A research on Libre Software Developers.* Fachgebiet für Informatik und Gesellschaft, TU-Berlin. Disponibile in: <http://widi.berlios.de>.

RODOTA' S., 1995. *La cittadinanza elettronica*, Telèma 1. Reperibile in: <http://www.fub.it/telema/TELEMA1/Telema1.html>.

SHIRKY V., 1998. *View Source... Lessons from the Web's massively parallel development.* Reperibile in: [http://www.shirky.com/writings/view\\_source.html](http://www.shirky.com/writings/view_source.html)

*Soggezione informatica dello Stato italiano alla Microsoft*, 2000. Reperibile in: <http://www.interlex.it/pa/letterap.htm>.

STALLMAN R., 1985. *The GNU Manifesto.* Reperibile in: <http://www.gnu.org/gnu/manifesto.html>.

STALLMAN R., 1998. *Copyleft: Pragmatic Idealism.* Reperibile in: <http://www.gnu.org/philosophy/pragmatic.html>.

STALLMAN R., 1998. *Why "free software" is better than "open source".* Reperibile in: <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

STALLMAN R., 1998. *Why free software should not have owners.* Reperibile in: <http://www.gnu.org/philosophy/why-free.html>.

STERLING B., 1993. *A short history of the Internet.* Reperibile in: <http://www.library.yale.edu/div/instruct/internet/history.htm>

TZOURIS M.. *Rethinking Communities of Practice in the Networked Organization. An Organizational Learning Study*, The London School of Economics and Political Science. Reperibile in: <http://www.geocities.com/tzmnlaos/cop/cop.pdf>

TZOURIS M., 2002. *Software Freedom, Open Software and Participants Motivation. A Multidisciplinary study*, The London School of Economics and Political Science. Reperibile in: <http://opensource.mit.edu/papers/tzouris.pdf>

*Unix history page*, 2002. Reperibile in:

[http://media.wiley.com/product\\_data/excerpt/36/04711648/0471164836.pdf](http://media.wiley.com/product_data/excerpt/36/04711648/0471164836.pdf)

VEALE K.J., 2003. *Internet Gift Economies: Voluntary payment schemes as tangible reciprocity*, in *First Monday*, volume 8, number 12 (December). Reperibile in:

[http://firstmonday.org/issues/issue8\\_12/veale/index.html](http://firstmonday.org/issues/issue8_12/veale/index.html)

WORKING GROUP ON LIBRE SOFTWARE, 2000. *Free Software/Open Source: Information Society Opportunities for Europe?*, Version 1.2, 2000. Reperibile in:

<http://eu.conecta.it/paper/paper.html>.

ZAKON R.H., *Hobbes' Internet Timeline v8.0*, Reperibile in:

<http://www.zakon.org/robert/internet/timeline/>

ZELTSER L., 1995. *The World-Wide Web: Origins and Beyond*. Reperibile in:

<http://www.zeltser.com/web-history/>

## Sitografia

<http://www.archive.org/> (sito dell'Internet Archive).

<http://www.annozero.org/> (sito italiano di notizie legate in particolare al software libero).

<http://www.apache.org> (sito ufficiale della Apache Software Foundation).

<http://www.berlios.de/> (sito del progetto tedesco Berlios, che funge da mediatore per il software open source tra sviluppatori, distributori e clienti).

<http://www.catb.org/~esr/> (sito personale di Eric S. Raymond).

<http://copydown.inventati.org/> (sito italiano dedicato al fenomeno del copyleft).

<http://www.crcitalia.it/> (sito dei Centri Regionali di Competenza per l'e-government e la società dell'informazione).

<http://www.creativecommons.it/> (sezione italiana del sito Creative Commons).

<http://www.creativecommons.org/> (il sito ufficiale del Progetto Creative Commons).

<http://cyberlaw.stanford.edu/index.shtml> (sito dello Stanford Center for Internet and Society).

<http://www.debian.org/> (sito della distribuzione Debian).

<http://www.dvara.net/HK/open.asp> (bibliografia sulla cultura hacker e sul software libero).

<http://www.eff.org/> (sito della Electronic Frontier Foundation).

<http://europa.eu.int/idabc/en/chapter/452> (sezione del sito dell'IDA-Interoperable Delivery of European eGovernment Services to public Administrations dedicata al software open source con particolare riferimento alle pubbliche amministrazioni)

**<http://fedora.redhat.com/>** (sito del Fedora Project, sponsorizzato da Red Hat).

**<http://www.firstmonday.dk>** (giornale in rete che raccoglie articoli relativi alle nuove tecnologie).

**<http://www.freego.it/>** (sito italiano di notizie sul software libero).

**<http://freshmeat.net/>** (sito che raccoglie numerosissimi progetti open source e costituisce un centro di aggregazione per gli sviluppatori).

**<http://www.fsf.org/>** (sito ufficiale della Free Software Foundation).

**<http://www.fsfeurope.org/>** (sito ufficiale del distaccamento europeo della FSF).

**<http://www.gimp.org/>** (sito ufficiale del progetto General Image Manipulation Program).

**<http://www.gnu.org/>** (sito ufficiale del progetto GNU).

**<http://www.gnutemberg.org/>** (sito del Progetto GNUtemberg).

**<http://www.kuht.it/>** (sito comunitario italiano dedicato al software libero, all'informatica e alla cultura hacker).

**<http://www.interlex.it>** (periodico in rete dedicato al diritto nell'ambito delle nuove tecnologie).

**<http://www.isr.uci.edu/research-open-source.html>** (sezione del sito dell'Institute for Software Research dell'Università della California dedicato al software open source).

**<http://www.lessig.org/>** (sito personale di Lawrence Lessig).

**<http://www.linux.it/>** (sito italiano della Italian Linux Society).

**<http://linux.kuht.it/risorse/freelosophy/freelosophy.php>** (bibliografia sulla cultura hacker e sul software libero).

**<http://www.mandrakelinux.com/it>** (versione italiana del sito della distribuzione Linux Mandrakelinux).

**<http://www.mysql.com/>** (sito del progetto MySQL).

**<http://www.mozilla.org/>** (sito ufficiale del progetto Mozilla).

**<http://www.newsforge.com/>** (sito di notizie su Linux e sull'open source).

**<http://www.ngvision.org/>** (sito di New Global Vision, progetto orientato alla raccolta e distribuzione di materiale video sotto licenze "libere").

**<http://www.novell.com/it-it/linux/suse/>** (versione italiana del sito della distribuzione tedesca Linux SuSe).

**<http://www.nosoftwarepatents.com/en/m/intro/index.html>** (sito della campagna contro l'introduzione dei brevetti sul software in Europa).

**<http://www.openlabs.it/>** (sito di un'associazione italiana che promuove il software libero).

**<http://www.openoffice.org/>** (sito ufficiale del progetto Open Office).

**<http://www.opensource.org/>** (sito ufficiale della Open Source Initiative).

**<http://www.opensourcedigest.org/>** (aspetti storici del movimento Opensource).

**<http://opensource.dschola.it/>** (sito italiano dedicato al software libero/open source con particolare attenzione al mondo della scuola).

**<http://opensource.mit.edu/home.html>** (sito della Free/Open Source Research Community promosso dal MIT, ospita un grande numero di articoli sulla materia).

**<http://opensource.provincia.pisa.it/opensource/cda/templates/index.jsp>** (sezione del sito della Provincia di Pisa dedicato al rapporto tra mondo open source e Pubblica Amministrazione).

**<http://www.plos.org/>** (sito del Public Library of Science).

**<http://www.pluto.linux.it/>** (sito del progetto Pluto, dedicato alla promozione del software libero).

**<http://www.redhat.com/>** (sito della distribuzione Linux Red Hat).

**<http://us2.samba.org/samba/>** (sito del progetto Samba).

**<http://slashdot.org/>** (sito comunitario di notizie sulla tecnologia).

**<http://www.softwarelibero.it/>** (sito dell'Associazione Software Libero, referente italiano per la FSF Europe).

**<http://sourceforge.net/>** (sito che funge da luogo di aggregazione per gli sviluppatori di software libero/open source).

**<http://www.stallman.org/>** (sito personale di Richard M. Stallman).

**<http://www.turbolinux.com/>** (sito in inglese della distribuzione Turbolinux).

**<http://www3.ware.it/>** (Osware, sito che raccoglie risorse utili sul mondo del software libero/open source).

**<http://www.winehq.com/>** (sito del progetto Wine)

**<http://www.wikipedia.org/>** (il sito dell'enciclopedia libera Wikipedia).

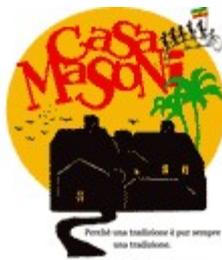
**<http://it.wikipedia.org/>** (pagina principale della sezione italiana di Wikipedia).

**<http://www.ziobudda.net/>** (portale italiano dedicato a Linux).

## RINGRAZIAMENTI

Si ringraziano in particolare mamma e papà per tutto quanto, gli zii Fiorindo, Renata, Bruna e Tano (che da quando ho iniziato l'università non ha mai smesso di incoraggiarmi), i cugini Gianlu e Alessandro e la Pierina.

Grazie al dott. Sbarbaro per l'interesse e l'aiuto dimostrato durante tutto il periodo della stesura del lavoro.



Casa Masoni: Masoni (la figura di riferimento, il capo e grande organizzatore), il Conte Gabro con cui ho condiviso la mia intera carriera scolastica (anche se non è venuto alla mia festa di Laurea), il Fede, vecchio viveur e confidente, Fabie, che devo rievocare ogni giorno in occasione dello spuntino notturno, TDK, con cui mi sposerò a 40 anni, Paola, supporto in questi anni universitari e che è uscita con due punti meno di me :-), Chiara a.k.a. Balute, che dopo un lungo percorso romagnolo è tornata in patria friulana, Cate, rappresentante della nuova onda di Casa Masoni, i pischelli Lore, Teo, Meri, i membri onorari Pinde, pluri-personaggio dell'anno e grande compagno di avventure nonché mio alter-alter, la Claire, una francese smutandata, Anna la Frikkessa, che presto farà un film sulle nostre avventure e tutte le persone che abbiamo ospitato nella nostra dimora.



Casa Pedi: Dr. Pedi, grande star del P-funk nonché mio capace personal trainer, Simonute, che ha sofferto con me in occasione della laurea ed è una fantastica

compagnia di “cuttigghio”, quando non le parte la liscia, Dario a.k.a. Punda a.k.a. Totò a.k.a. Cuervo Jones, con il quale metterò su un'azienda che gli permetterà finalmente di inserirsi nel florido nordest (e rubarci la pastasciutta).

Slalus, compagno di banco e amico saggio, Bigio, che ha condiviso con me le esperienze messicane, il compagno di radio Speed Banana, il Quaglia e lo Stief, dal liceo in poi con furore, Daniela a.k.a. Peli, la mia ex-amica, Gigi Sentimento, maestro di vita e di pesca, guru (beccatela!), Callister, Spagna, Felipe, Ari, Addi, i Fiori Malaticci e tutti quelli della combriccola, Eleonora, Giuli, Torquecasa, Turbofolk, Orcianna, Depistabile, Tano il Gatto, MasonBlanche, CasaBongiah, multinazionale Belpoggio, Casa Felice, Radio Fragola, Via dei Porta, Prod. Kaffeinico, Comfortstand per la condivisione, Rana per l'indifferenza e tutti quelli che ho dimenticato(vi inserirò nella versione 2.0).







