

Alma Mater Studiorum – Università di Bologna

---

FACOLTÀ DI ECONOMIA  
Corso di Laurea in economia Aziendale

**OPEN SOURCE: UN'ANALISI DEI NUOVI MODELI DI  
BUSINESS E DELLE PROSPETTIVE DI APPLICAZIONI NEI  
CAMPI NO SOFTWARE**

Tesi di Laurea di:  
ILARIA ZINGARIELLO

Relatore:  
Chiar.mo Prof. VINCENZA ODORICI  
Strategia d'Impresa

Sessione III

---

Anno Accademico 2004/2005

*Dedicata a Papà Mamma  
Antonio Claudia Anja e Joele....*

*.....e a tutti coloro  
che sussurrano parole di libertà!*

Copyright © 2006 Ilaria Zingariello  
<ilaria.zingariello@gmail.com>

Quest'opera è stata rilasciata sotto la licenza Creative Commons Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/2.5/it/> o spedisci una lettera a Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Data di stesura Marzo 2006



# INDICE

Introduzione..... pag.9

## Capitolo 1

### RADICI STORICHE DEL MOVIMENTO OPEN SOURCE

1.La storia dell'Open Source..... pag.14

2.Il progetto GNU e la Free Software Foundation..... pag.18

3.La tutela giuridica del software..... pag.21

4.Free Software diventa Open Source..... pag.27

5.La svolta di Linux..... pag.30

6.Il software Open Source e la situazione attuale..... pag.32

7.Una barriera all'Open Source:Il caso Palladium..... pag.36

8.Conclusioni..... pag.39

## Capitolo 2

# L'OPEN SOURCE COME NUOVO MODELLO DI BUSINESS

- 1.La struttura organizzativa della comunità Open Source. pag.42
- 2.La comunità Open Source: le motivazioni..... pag.46
- 3.Il problema del free riding ..... pag.50
- 4.Il problema della diffusione del software Open e le esternalità di rete..... pag.52
- 4.a).Rendimenti crescenti ed il fenomeno del lock-in..... pag.55
- 5.Il concetto di Massa Critica..... pag.60
- 6.Il coordinamento: Modularità del software e il ruolo della rete Internet..... pag.62
- 7.Il sottogruppoefficace..... pag.67
- 8.Conclusioni..... pag.70

## CAPITOLO 3

### COMPETIZIONI E STRATEGIE OPEN SOURCE

1.Open Source Economy.....	pag.73
2 L'emergenza dei nuovi modelli di business Pure Player e Ibridi.....	pag.77
2 a) I Pure Player.....	pag.79
2 b) I modelli di Business Ibridi.....	pag.82
3 Le diverse strategie applicate al modello di sviluppo Open Source.....	pag.86
4. La concorrenza nel mondo Open Source.....	pag.90
5. Le Gare a progetto e la portabilità dei software.....	pag.92
6. L'autoformazione.....	pag.95
7.Lo stile.....	pag.98
8.Conclusioni.....	pag.100

## Capitolo 4

# L'OPEN SOURCE NELLA PUBBLICA AMMINISTRAZIONE

1.Il coinvolgimento della Pubblica Amministrazione.....	pag.101
2.Motivazioni all'adozione dell' Open Source nella Pubblica Amministrazione.....	pag.103
3.L'Open Source fa breccia nella Pubblica Amministrazione europea.....	pag.107
4.Criteri di valutazione del ruolo della Pubblica Amministrazione.....	pag.111
5.Il caso più eclatante: Monaco e il progetto LiMux.....	pag.117
6.Un caso reale: il CNIPA lancia l'Osservatorio Open Source.....	pag.121
7.Open Source e Pubblica Amministrazione: il comune di Udine insegna.....	pag.125
8.Conclusioni.....	pag.130

## CAPITOLO 5

### APPLICAZIONI OPEN SOURCE NEI CAMPI NO SOFTWARE

1.Un nuovo concetto di Produzione.....	pag.133
2.Nuovi campi di applicazioni.....	pag.136
3.Il progetto Wikipedia.....	pag.138
4.Il movimento Openaccess.....	pag 142
5.L'illusione del dominio pubblico.....	pag.144
6.Il personal biotech.....	pag.148
7.Biologia Open Source: Bios.....	pag.152
8.Open Source e Open Formats nei processi di ricerca Archeologica.....	pag.156
Conclusioni finali.....	pag.158



## Introduzione

Come tema della mia tesi ho deciso di trattare l'argomento Open Source.

Questo argomento ha catturato la mia attenzione sin da subito: mi ha incuriosito molto l'idea di sviluppo di un business attraverso una comunità volontaria cooperante, fatta di soggetti eterogenei, che vanno da studenti a ingegneri informatici, da appassionati di informatica a veri e propri programmatori, che più o meno quotidianamente si incontrano in rete per sviluppare, modificare o creare prodotti software, destinati ad essere di pubblico dominio.

Da buona studentessa di Economia, abituata agli insegnamenti di corsa ai profitti “senza nessuna pietà”, il suono di parole del tipo comunità, cooperazione, gratuità, copyleft (termine abbastanza sarcastico in contrapposizione al copyright) e pubblico dominio, inerenti allo sviluppo di un business, hanno rimbombato ripetutamente nella mia mente, e conoscendo un po' la reputazione di Linux, che mi sembrava abbastanza convincente, ho sentito immediatamente l'esigenza di approfondire.

Ho cercato di analizzare il fenomeno, paragonando la realtà Open a quella close dei software proprietari, ed ho cercato di capire, dal punto di vista economico, quali potessero essere le differenze.

Il fenomeno Open Source infatti, costituisce un'importante innovazione nel processo di produzione del software in contrapposizione al tradizionale modello proprietario del mondo commerciale.

Inoltre ho cercato di dimostrare come il software open non solo può convivere con quello proprietario, ma può anche originare una serie di nuove forme di organizzazione economica nelle quali elementi open e proprietari si combineranno variamente.

Per analizzare bene l'argomento ho cercato di dare una struttura alla tesi, che seguisse un certo ordine logico e tradizionale.

Sono partita con l'analisi delle radici storiche del movimento Open Source, descrivendo nel primo capitolo la dinamica storica e le radici ideologiche del fenomeno, dalle origini della scienza informatica, fino ai giorni nostri.

Sono entrata anche in merito all'analisi giuridica del fenomeno, ovvero alla tutela dell'opera software e all'alternativa delle licenze copyleft, descrivendo inoltre i primi prodotti Open Source: vale a dire la General Public Licence (GPL) dal punto di vista giuridico, e Linux dal punto di vista tecnico.

Nel secondo capitolo, ho cercato di analizzare la struttura organizzativa di una comunità Open, ricercando le motivazioni che spingono una moltitudine eterogenea di persone a prendere parte al lavoro di comunità.

Successivamente, l'analisi del fenomeno continua sulla base di una curiosità circa gli effetti economici e sociali indotti dall'introduzione di questo nuovo modello di business. Infatti il modello Open Source è basato su alcune diversità presenti nella comunità di sviluppo, vale a dire l'eterogeneità dei soggetti, l'indipendenza e la volontarietà, applicate ad un sistema autoregolato e in grado di garantire efficienza ed efficacia nei processi, e quindi di produrre risultati apprezzabili e di qualità.

Ho parlato poi delle conseguenze di fenomeni emergenti quali la diffusione di Internet, da cui ne deriva un forte sviluppo di economie di rete e quindi di

esternalità di rete, del problema del free-riding e della modularità del software.

La terza parte è di natura interpretativa: si analizza la nascita di una vera e propria Open Source Economy che si sviluppa attorno al nuovo paradigma, e la si paragona alla Close Economy delle più grandi software house. L'analisi inoltre continua sull'impatto dell'Open Source su competizioni e strategie, e sulla nascita di nuovi modelli aziendali che adattano il loro business al nuovo paradigma.

Il quarto capitolo riguarda l'evoluzione del fenomeno Open Source: spiega l'entrata in scena di nuovi potenti soggetti, come la Pubblica Amministrazione. In questa parte si analizza la rilevante influenza che ha la PA circa l'adozione del software libero, sottolineando come l'Open Source produca effetti di rilievo non solo dal punto di vista economico, ma anche dal punto di vista sociale.

Interessante inoltre è la diffusione dei concetti portanti dell'Open Source applicati a campi non software: nel quinto capitolo infatti, si parlerà delle applicazioni Open in campi tipo l'editoria, il settore musicale, il campo biotecnologia e farmaceutico.

L'obiettivo principale di queste opere è la diffusione di informazioni attraverso la messa a disposizione, da parte di tutti, del proprio sapere: ne è un esempio eclatante l'enciclopedia on-line Wikipedia.

Ho cercato, insomma, di approfondire e analizzare come era possibile, economicamente parlando, che questo nuovo paradigma non solo continuasse a svilupparsi, ma stava addirittura sorpassando colossi del

settore tipo Microsoft, con strategie economiche alla cui base vi erano variabili di gratuità, collettività e dominio pubblico.

Ho raccolto informazioni attraverso libri, articoli di quotidiani, siti internet; ho partecipato a diversi forum attraverso i quali la comunità interagisce, risolve problemi, e crea;

ho fatto domande e proposto argomenti di discussione alla comunità, attraverso i quali ho avuto anch'io la possibilità di interagire con la comunità stessa.

Inoltre ho cercato di dimostrare come il software open non solo può convivere con quello proprietario, ma può anche originare una serie di nuove forme di organizzazione economica nelle quali elementi open e proprietari si combineranno variamente.

A mio avviso l'Open Source rappresenta una promessa per vari motivi, e cercherò di arrivare alla conclusione che l'Open Source non è un assurdo economico.

# Capitolo 1

## **RADICI STORICHE DEL MOVIMENTO OPEN SOURCE**

### **Introduzione**

Nel primo capitolo cercheremo di analizzare, attraverso i processi storici di cambiamento sociale e tecnologico e attraverso l'analisi delle esigenze di modificabilità dei prodotti software, intese come coinvolgimento personale e diretto alla programmazione, quali sono state le reali motivazioni che hanno spinto alcuni programmatori a lavorare allo sviluppo del movimento Open Source.

Ma oltre ad analizzarne le radici storiche, osserveremo il significato del movimento Open, nel senso di apertura e condivisione, gli strumenti giuridici da cui esso è tutelato ed il passaggio dal free software all'Open Source, attraverso la creazione del più grande prodotto Open, ovvero Linux, e la creazione della Free software Foundation.

# 1.La storia dell'Open Source

Agli albori dell'informatica, quando ancora gli elaboratori si trovavano soltanto nei grandi laboratori di ricerca dell'università, il software era libero. Ognuno poteva mettergli mano, fare le proprie modifiche e renderle disponibili agli altri: in un certo senso era naturale che fosse così.

Il software non aveva ancora valenza commerciale in senso stretto e poche erano le persone in grado di programmare.

Nei primi anni '80 con la diffusione industriale dell'informatica le cose iniziarono a cambiare: la tecnologia informatica iniziò a diffondersi rapidamente, vennero distribuiti molti strumenti informatici, ovvero i Personal Computer, ed il software assumeva sempre più valenza commerciale.

Gli utenti dei PC in generale, non possedevano competenze di programmazione e quindi non essendo autonomi rispetto allo strumento erano disposti a dare un valore commerciale al software stesso.

La conseguenza più rilevante della nascita del mercato software fu la migrazione dei migliori programmatori dalle strutture di ricerca accademiche verso i laboratori industriali, evidenziando così la crescita di un mercato del software caratterizzato dalla protezione delle proprietà intellettuali.

Il settore software andava verso la crescita, e lo sviluppo di prodotti sempre più chiusi, brevettati e non modificabili, raccoglieva sempre più i disagi dei

più esperti che lamentavano la non possibilità di adattare i prodotti software alle loro specifiche esigenze, in cambio di una sensazione sempre più evidente di schiacciante sottomissione e schiavitù dettata dai brevetti.

Il Massachusetts Institute of Technology (MIT) fu uno dei primi istituti a subirne le conseguenze: i programmatori iniziavano a lamentarsi di non poter nemmeno mettere le mani nelle stampanti Xerox per implementare una comoda funzione di avviso per la carta inceppata.

Fu così che un gruppo di programmatori, ribellandosi alla nuova politica di licenze proprietarie, decise di creare un sistema operativo alternativo per le macchine della Digital Equipment Corporation (DEC) che allora si utilizzavano al MIT.

Queste erano fornite di un sistema operativo proprietario che il MIT non aveva nessuna intenzione di rendere libero, e crearono così un nuovo sistema operativo che fu chiamato Incompatible Time Sharing (ITS) che era volutamente incompatibile con quello della DEC, rimanendo per 15 anni il sistema operativo preferito degli hacker<sup>1</sup>.

Successivamente, nel 1983, il primo ad avvertire repulsioni per quel nuovo modello proprietario di informatica che si andava sviluppando fu proprio il padre dell'Open Source : Richard Stallman.

Riferendosi a quel sistema Stallman nel suo saggio dice: “Una comunità cooperante era vietata; la regola creata dai titolari di software proprietario, era: se condividi il software con il tuo vicino, sei un pirata; se vuoi modifiche pregaci di farle.”

---

<sup>1</sup> Hacker: sono esperti di informatica che lavorano ai codici, con l'unico scopo di migliorarli diffonderli e aiutare chi ha problemi ad utilizzarlo

Stallman, fu il primo a muoversi concretamente e visibilmente nella direzione del risveglio delle coscienze di coloro che l'informatica l'avevano simbolicamente partorita.

All'epoca poteva vantare una decina di anni di esperienza come programmatore, dato che nel 1971 era stato assunto nel laboratorio di Artificial Intelligence del Massachusetts Institute of Technology (MIT), e in quegli anni aveva avviato la prassi di distribuire liberamente e gratuitamente i suoi programmi, incoraggiando chiunque a modificarli e migliorarli.

Quando poi al MIT decisero per la prima volta di implementare le password ai terminali, Stallman si oppose tenacemente e riuscì a trovare il modo di scardinare il sistema venendo a sapere le password di tutti gli utenti.

Decise quindi di inserire un messaggio che veniva visualizzato all'autenticazione, che incitava la modificazione di tutte le password con una generica uguale per tutti: Pluto.

Un quinto degli utenti seguì il suo consiglio.

Fu questa anche la sorte dell'Emacs, il diffusissimo programma editor da lui stesso creato, che portò alla genesi della "Comune di Emacs", una specie di catena di utenti che avevano come unico obbligo quello di diffondere a loro volta liberamente le modifiche apportate.

Sempre nel 1983 Stallman decise dunque di abbandonare il MIT per dedicarsi a progetti personali che tenessero in vita lo spirito hacker a cui era tanto affezionato.



Il primo fra tutti fu la realizzazione di un sistema operativo<sup>2</sup> free, liberamente accessibile, di tipo Unix<sup>3</sup> che fosse svincolato dall'impostazione esclusiva del copyright tradizionale e quindi distribuirlo liberamente: cioè un sistema completo di applicazioni e strumenti di sviluppo.

Nasce così il Progetto GNU, che sta per GNU's not Unix.

---

<sup>2</sup> Sistema operativo: è un sistema responsabile del diretto controllo della gestione dell'hardware che costituisce un computer, e delle operazioni base; si occupa anche di controllare gli accessi degli utenti e dei processi che vengono eseguiti.

<sup>3</sup> Unix: è un sistema operativo portabile per computer, inizialmente sviluppato da ricercatori di AT&T

## 2. Il progetto GNU e la Free Software Fondation

Con un'espressa vena di antagonismo e di sfida nel 1984 nasce il progetto GNU , che sta per GNU's not Unix, al fine di creare un sistema operativo completo di tutte le sue parti e completamente libero, che avrebbe dovuto coinvolgere migliaia di esperti di informatica e condensare le conoscenze e gli sforzi condivisi.

Stallman voleva proprio sottolineare- con scopi altamente propagandistici- la doppia implicazione di libertà e gratuità.

Stallman dice “ Molta gente crede che lo spirito del progetto GNU sia che non si debba far pagare per distribuire copie del software o che si debba far pagare il meno possibile: solo il minimo per coprire le spese. In realtà noi incoraggiamo chi vuole ridistribuire il software libero a far pagare quanto vuole o quanto può”.

Costruire interamente un sistema sarebbe stato davvero complesso, così Stallman decise di utilizzare ed adattare pezzi già esistenti di free software tutte le volte che sarebbe stato possibile.

Ad esempio per la fase iniziale decise di utilizzare TeX come principale formattatore di testo e successivamente optò per X Wind System, piuttosto che scrivere interamente un altro Wind system per GNU.

Alla fine del 1985 GNU era pienamente utilizzabile, e Stallman si iniziò ad interrogare su come diffonderlo, data la grande richiesta degli utilizzatori.

Escluse la rete come canale distributivo, perché allora internet non era ancora diffuso come oggi per cui non tutti potevano accedervi facilmente; allora decise di inviarlo a chi lo avrebbe richiesto, dietro pagamento di una forma di iscrizione di \$150: iniziò così il business della distribuzione del free software.

Con lo sviluppo della comunità di GNU, Stallman, ebbe un'idea per la quale tutti i potenziali adepti, avrebbero dovuto avere un riferimento unico che si facesse portavoce dell'impresa: sempre per opera di Stallman e sotto la sua super visione, nacque la Free Software Foundation, un'associazione senza scopo di lucro, con l'obiettivo di diffondere il software libero, coordinare i progetti e sensibilizzare il popolo dell'informatica fornendo protezione legale e uno spazio libero di comunità.

In seno alla FSF, Stallman, fu creatore della Gnu Public Licence (GPL) che successivamente diventerà General Public Licence, uno strumento che permise una diffusione sicura e legale del software libero influenzando notevolmente la comunità di sviluppo e offrendogli regole certe per i progetti sviluppati dalla Free Software Foundation.

Le cose iniziarono a cambiare radicalmente, e con l'avvento della rete telematica, quando Internet, verso la fine degli anni '80, iniziò a collegare stabilmente un numero cospicuo di utenti, il messaggio della Free Software Foundation poté raggiungere gli hacker sparsi per il mondo, rimasti isolati e disorientati per il cambiamento, e riuscì a coinvolgerli nel nuovo modello di produzione dei prodotti informatici, un modello cioè all'insegna della cooperazione, della libertà e della condivisione.

Il seme della rivoluzione, poteva trovare solo nelle coscienze dei singoli appassionati, il terreno fertile per germogliare: anche se GNU non avesse avuto alcun vantaggio tecnico su Unix, avrebbe avuto sia un vantaggio sociale, permettendo agli utenti di cooperare, sia un vantaggio etico, rispettando la loro libertà.

### **3. La tutela giuridica del software**

Proprio in seno al progetto GNU Stallman con la General Public Licence diede il suo più grande contributo al movimento Open Source: essa infatti è un documento dal linguaggio giuridico che permette una diffusione sicura e perfettamente legale del Free Software dando così un punto di svolta alla comunità.

La licenza varata nel 1998, porta con se il concetto sarcastico di copyleft in contrapposizione al concetto di copyright: il copyleft applica i diritti dei copyright, e li sovverte per servire allo scopo totalmente opposto. L'espressione copyleft indica un tipo di licenza libera per la quale pur garantendo le libertà previste dalla definizione, vengono imposte delle restrizioni sul rilascio di opere derivate, in modo tale da far sì che queste si mantengano sempre libere, generalmente sotto la stessa licenza dell'opera originale: in altre parole il copyright non è più sulla proprietà ma sull'apertura.

E' un metodo generale per sviluppare un software libero, e l'idea centrale è quella di concedere il permesso di eseguire il programma, copiarlo modificarlo e ridistribuire le versioni modificate a certe condizioni.

Una per esempio è la non possibilità di rendere proprietario qualsiasi modifica o aggiunta effettuata, in modo che, chiunque riceva il software può avere la libertà di dividerlo e di cooperare con altra gente, fino a

diventare un diritto imprescindibile per collaborare con altre persone e formare una comunità.

Il copyleft, quindi, protegge la libertà di utilizzo , di modifica e distribuzione del codice sorgente, ovvero protegge la così detta ‘apertura’; in assenza di tale protezione, il software Open, potrebbe ricadere nel dominio del software proprietario, sviluppando comportamenti opportunistici tesi a massimizzare i benefici personali, piuttosto che valorizzare il software come bene collettivo.

L’espressione copyleft, in un senso non legale, può essere vista come il contrario di copyright, perché priva l’autore di alcuni suoi diritti e li cede all’utente.

Naturalmente è l’autore stesso che decide di usare questo tipo di licenza e quindi è lui stesso che decide di perdere alcuni suoi diritti.

La pratica comune per raggiungere lo scopo di imporre la libertà di copia e distribuzione di una creazione o di un lavoro e dei suoi derivati, è quella di distribuirlo con una licenza; questa licenza, per far sì che il lavoro sia sotto copyleft, occorre che assicuri che il possessore della copia derivata la possa distribuire solo con la stessa licenza.

In altre parole la differenza tra il software Open Source e quello proprietario, sta nel diverso utilizzo del copyright:

da un lato le licenze proprietarie concedono agli utenti il diritto di utilizzare il software, in cambio del pagamento di un determinato prezzo, ma non permettono un accesso ai codici sorgente con possibilità di modifica che comporterebbe la visibilità del codice stesso;

dall'altro lato invece, le licenze Open Source concedono l'accesso ai codici anche per modificarli, controllandone le modalità di modifica e diffusione, stabilendo che le suddette modalità siano eseguite in modo 'aperto' cioè "libero".

Qualcuno si è anche avventurato in una forzata traduzione in "permesso di copia" o "permesso d'autore" ma il concetto di copyleft è stato poi evidenziato dai tenaci della Free Software Foundation (FSF) : alcuni di loro amavano apporre ironicamente sui loro lavori una nota sul copyright il cui testo era letteralmente

Copyleft – all rights reversed 

Con una 'C' rovesciata, invece del canonico copyrights© - all rights reserved.

Dopo tale annotazione venivano poi elencati tutti i diritti e le libertà di cui l'utente aveva ufficialmente diritto, in base alla licenza d'uso, e poi veniva rimarcato l'obbligo di tenerle intatte in futuro e renderle disponibili agli altri utenti.

Analizzando la struttura della GPL, seguendo il testo della licenza nella sua più recente versione (giugno 1991) tradotta in italiano<sup>4</sup>, si nota come il documento procede con un preambolo che si presenta come un sunto dei principi etici e giuridici di cui la FSF si fa massima portavoce.

In seguito sette paragrafi spiegano l'intento programmatico della licenza, la sua funzione tecnico-giuridica, il suo corretto utilizzo ed i rapporti di responsabilità e garanzia nel sistema GNU di distribuzione del software.

---

<sup>4</sup> Vedi: [www.softwarelibero.it/gnudoc/gpl.it.txt](http://www.softwarelibero.it/gnudoc/gpl.it.txt)

Tutti aspetti che si ripercorrono nella lettura delle tredici sezioni della licenza vera e propria.

Alla fine dei “termini e delle condizioni” è apposto un'appendice esemplificativa su come servirsi in modo corretto della General Public Licence, contenente un fac-simile delle annotazioni dei copyleft che si necessitano, affinché una nuova opera software sia coperta a tutti gli effetti da tale licenza.

Sulla scia dei segnali di successo della licenza di Stallman, raccolti nell'ambito della comunità di hacker, formata non solo da ricercatori e studenti, ma anche da esperti ed interessati di informatica, cominciarono a diffondersi altre licenze di software che, pur con funzioni e criteri diversi, si ispiravano al modello individuato dalla FSF. Ne vediamo alcune con maggior dettagli

#### La BSD Licence

BSD si riferisce alla Barckley Standard Distribution, ovvero la versione di Unix sviluppata negli anni '70 nell'Università della California, sul modello della condivisione tipico dell'epoca.

Si tratta di una licenza senza componenti ideologiche e programmatici i cui termini risultano sintetici ed essenziali, e la cui applicazione si risolve nell'inserimento di una breve nota standard da inserire nei file che si intendono tutelare.

La problematica di questa licenza consisteva nel dare la possibilità di usare il codice da lei tutelato, per sviluppare software proprietari: aspetto che la pone al di fuori del paradigma 'copyleft' poiché non è garantita all'infinito,



la diffusione dei diritti da essa concessa, i quali si interrompono nel momento in cui il codice diventa proprietario.

Una frase di un attento osservatore dell'Open Source, Brian Behlendorf riassume le dispute più accese riguardanti la BSD circa la così detta "clausola pubblicitaria: "Ecco il codice, fateci quello che volete, non c'interessa; se lo provate e lo vendete datecene credito".

La licenza obbligava l'utente a dare esplicito riconoscimento all'Università nelle eventuali inserzioni pubblicitarie, raggiungendo così l'obiettivo di un programma di sovvenzioni amministrative da parte del governo; ma col tempo alcuni sviluppatori aggiunsero alla clausola i nomi di altre organizzazioni che avevano sostenuto lo sviluppo del software adducendo il fatto che dovessero essere citati sempre più soggetti nelle inserzioni. Questo portò nel 1999 l'università della California a decidere di eliminare dalla versione ufficiale della licenza quella clausola facendo così cessare anche l'astio dimostrato da Stallman nei confronti della BSD license, la quale nonostante tutto vanta una discreta diffusione.

Più o meno identica è la MIT license, che prende il nome dal famoso centro di ricerca del Massachusetts precedentemente citato.

### La MOZILLA Public License

Quando la Netscape Navigator decise di diffondere il suo prodotto di punta, fu costretta a dover fare i conti con una situazione diversa da cui si trovava il progetto GNU: infatti questo aveva degli intenti propagandistici e no-profit, che non combaciavano assolutamente con la realtà di una grande azienda come la Netscape.

Bisognava quindi liberare il sorgente senza però rendere il software un effettivo software libero nel senso voluto dalla FSF: era giunto il momento di sviluppare una licenza nuova che avrebbe sposato i criteri giuridico-ideologici del software libero accompagnando lo sviluppo a livello aziendale di software a sorgente aperto.

Nacque così la Mozilla Public License: la licenza prevede che chiunque partecipi alla stesura del progetto con codici, rinunci ad ogni tipo di diritto di brevetto a cui il codice potrebbe dare adito.

## 4. Il Free Software diventa Open Source

Il termine Free Software nacque agli inizi degli anni Ottanta dall'idea di Stallman di creare un software libero, ovvero un prodotto liberamente distribuibile e liberamente eseguibile per qualsiasi scopo, con riferimento alla libertà di usare e migliorare il software.

Free in inglese assume due significati, gratuito e libero, e gli interlocutori non esperti a cui veniva proposto il software “free” tendevano a considerarlo “gratuito” e quindi di scarsa qualità.

Il termine Free Software era sinonimo di quattro libertà:

- di eseguire i programmi;
- di modificarli secondo i propri bisogni
- di distribuire copie del programma gratuitamente o dietro compenso;
- di distribuire versioni modificate del programma, così che la comunità possa fruire dei miglioramenti apportati.

Nel 1998 tra le varie polemiche, il software libero, venne ribattezzato Open Source: il termine Open Source venne coniato da Bruce Perence come un modo per mascherare l'ambiguità del termine Free Software.

Open Source è un marchio registrato della Open Source Iniziative (OSI) fondata nel 1987 da Bruce Perence e Eric Raymond allo scopo di proteggere l'idea della filosofia Open Source.

L'OSI si occupa di definire quali licenze possono essere considerate Open Source, attraverso un documento che ne specifica gli standard, chiamato Open Source Definition, secondo cui chiunque deve avere la possibilità di utilizzare, studiare, modificare ed integrare il così detto codice sorgente dei programmi, ridistribuendo liberamente il risultato della sua attività.

L'obiettivo della OSI, a dire il vero, è molto più ambizioso: cerca di trasformarsi in una specie di consorzio di vigilanza che controlla la corretta applicazione e la corretta interpretazione della Open Source Definition, e che attribuisce una certificazione ai prodotti che ne rispettano i principi apponendo sopra di essi un marchio di garanzia.

D'altra parte L'Open Source, si occupa di definire, in una prospettiva totalmente interna alle logiche di mercato, quali siano le modalità migliori per diffondere un prodotto Open, secondo criteri Open, cioè aperti.

Il Free Software ha un senso che va ben oltre il mercato, pur non escludendolo a priori; l'Open Source invece esiste per adattare un modello preesistente, cioè quello del free software, al mercato.

Quando poi tra gli anni Ottanta e Novanta, il software diventava sempre più importante, come prodotto tecnologico, l'attenzione delle aziende nei confronti del Free Software, delle comunità hacker e dei suoi metodi, diventava sempre più grande.

Infatti, mentre all'inizio il mercato di riferimento, vedeva crescere solo il colosso Microsoft ed i suoi prodotti close, che grazie ad alcune manovre di mercato lo aveva condotto ad una fase sostanzialmente monopolistica, la radicalità della politica del Free Software, questa insistenza sulla libertà, era piuttosto scomoda e controproducente: non solo il termine free non piaceva

alle aziende che cercavano quote di mercato, ma anche la licenza GPL non era soddisfacente, perché col suo sistema di diffusione, rischiava di lasciar fuori aziende che vedevano nelle comunità hacker, uno sbocco naturale delle proprie strategie economiche.

Si trattava ora solo di dimostrare come le logiche organizzative e produttive delle comunità da una parte e il profitto dall'altra potessero andare d'accordo. Diventava così molto semplice ampliare il bacino delle aziende impegnate ad entrare nel fantastico mondo Open Source.

## 5. La svolta di Linux

Per un risultato completo e perfetto – pensavano gli stessi volontari del progetto GNU - c'era bisogno di un certo grado di controllo e centralizzazione del lavoro: nel 1991 Linus Torvalds, uno studente universitario finlandese dell'Università di Helsinki, scrisse a velocità record un kernel<sup>5</sup> Unix-compatibile, stabile e funzionante e lo battezzò, un po' per gioco, Linux.

Decise di licenziarlo sotto la GPL e quindi di renderne disponibile il codice sorgente.

Il kernel era proprio l'ultimo tassello mancante per avere un sistema operativo completamente libero, e gli sviluppatori di free software di tutto il mondo non ci misero molto a capirlo. Si raccolse tutto il software libero GNU esistente, si aggiunse il kernel di Linus Torvalds, e nacquero così le prime distribuzioni di software libero, denominate convenzionalmente “distribuzioni Linux”.

GNU/Linux fu il primo software che permise agli utenti di utilizzare il proprio computer in modo assolutamente libero, in quanto non apparteneva a nessuna software house, e Linux divenne così la prima concreta e chiara dimostrazione pubblica che la comunità hacker ed il movimento per il software libero non erano soltanto un gruppo disorganizzato di idealisti e che le cose potevano realmente cambiare.

---

<sup>5</sup> Il kernel è il cuore del sistema operativo

Affinché Linux il movimento Open diventasse credibile, le aziende di software indipendente dovevano impegnarsi a portare le loro applicazioni su queste piattaforme.

Il momento in cui si diede avvio alla campagna Open era anche il momento in cui, alcune aziende, tipo la Microsoft, avrebbero potuto assumere atteggiamenti ostili: un' operazione di marketing ben condotta infatti avrebbe potuto far affondare Linux ed il movimento Open, ma quando i distributori di data-base iniziarono a dimostrare il loro entusiasmo nei confronti del pinguino, si aprirono le strade per altri, ed iniziò così l'effetto valanga.

Dal 1998 allora, sempre più gente iniziava ad usare Linux, ed il così detto 'parco clienti' si ingrandiva sempre più, fino a coinvolgere aziende tipo Cisco System che stava nascendo in quel periodo convincendo anche gli investitori più scettici ad iniziare a investire in Linux: qualcosa stava cambiando.

## **6. Il software Open Source e la situazione attuale**

Nel 1984 la Free Software Foundation e la creazione della General Public License, furono la fonte di un'idea innovativa di Stallman, oggi meglio conosciuta come movimento Open Source.

Ribadita nel 1997 nella Open Source Definition, secondo cui chiunque deve avere la possibilità di utilizzare, studiare, modificare ed integrare il codice dei programmi, ridistribuendo liberamente il risultato della sua attività, oggi l'Open Source, sembra essere un vero e proprio concorrente temibile per le imprese che dominano il mercato del software.

Le imprese di fatto non si trovano a dover fronteggiare una minaccia di un nuovo concorrente che fa le stesse cose, magari più velocemente o offrendo altri tipi di servizi, ma si trovano d'innanzi ad un nuovo paradigma completamente contrapposto a quello tradizionale.

Nel mondo proprietario, gli sviluppatori di software, lavorano alla scrittura di codici che sono protetti da una licenza commerciale che ne vieta la distribuzione, creando così forti limiti nella diffusione e nell'utilizzo, e impedendone la duplicazione e qualsiasi tipo di modifica.

Il movimento Open Source invece fornisce ai programmatori la possibilità di lavorare liberamente sul codice sorgente, facilitando così la correzione degli errori e liberandoli da qualunque tipo di pressioni imposte dalle software house creando quindi un software qualitativamente migliore.

Programmatori di tutto il mondo, che si identificano in esperti di informatica o studenti alla ricerca di informazioni nel campo software, spendono molto



del loro tempo libero lavorando ai progetti Open Source, acquisendo o rilasciando grandi esperienze nel settore della programmazione, a favore della comunità.

La forte gratificazione intellettuale che i programmatori acquisiscono nello scrivere nuovi codici sorgenti, è del tutto simile a quella di chi fa nuove scoperte scientifiche, e così come il processo di ricerca scientifica prevede la condivisione dei risultati, anche i codici Open prevedono la condivisione dei codici sorgente.

Condividendo i codici sorgente i programmatori hanno la possibilità di ricevere un feed-back dai propri colleghi, attraverso i quali possono correggere e migliorare il proprio lavoro, ricevere riconoscimento e prestigio ed essere notati da aziende software.

Inoltre molti sviluppatori sono studenti che utilizzano la programmazione per lo sviluppo della loro tesi acquisendo un'esperienza che potranno poi sfruttare all'interno del mondo del lavoro.

Ma qual è il giudizio dei responsabili sull'OS?

La fiducia nell'Open Source sembra ormai più che consolidata.

Il 66% degli intervistati si dichiara favorevole ad adottare software libero nei propri sistemi aziendali.

Le più grandi compagnie di hardware e software non mettono più nulla sul mercato che non sia almeno compatibile con Linux e alcune ne incoraggiano addirittura l'utilizzo, in previsione di un miglioramento dei supporti tecnici e degli aggiornamenti; spiegando non solo i vantaggi del software aperto,

ma anche il fatto che pur avendo definizioni diverse, in realtà si tratta della stessa cosa.

A tal proposito è interessante notare come il software libero regga soprattutto alla prova dei fatti: nel 75% dei casi chi ha utilizzato soluzioni OS è poi maggiormente predisposto ad adottarne altre.

Il mensile statunitense Business Week dedica un articolo al software Open Source alle aziende che operano in questo campo dichiarando che il 2005 è stato un anno estremamente positivo e che gli analisti prevedono un 2006 in crescita sia dal punto di vista numerico che di popolarità del software aperto.

In particolare, dopo una prima fase nella quale i progetti Open Source venivano intrapresi a livello aziendale su iniziativa isolata e personale di tecnici desiderosi di sperimentare soluzioni innovative, ora è ragionevole affermare che i progetti Open Source vengano lanciati ai livelli più alti della gerarchia aziendale e sostenuti in maniera compatta da tutto il management.

Per raggiungere questa situazione ci sono voluti ovviamente anni di duro lavoro e in particolare il supporto dei venture capitalist che negli ultimi due anni hanno iniziato a puntare seriamente sul finanziamento di start-up legate a progetti Open Source.

Secondo gli analisti, imprenditori ed investitori consultati da Business Week, cinque sono gli eventi che hanno lasciato il segno nel 2005 e che lasciano ben sperare per il 2006:

-Red Hat ha finalmente dimostrato la sostenibilità del modello economico sottostante al software Open Source. Ricordiamo come Red Hat abbia appena annunciato risultati record per il terzo trimestre del 2005 e sia valutata dagli analisti borsistici come una delle aziende meglio posizionate per sfruttare la crescita del mercato Open Source.

Il caso di Red Hat, lo analizzeremo meglio nel terzo capitolo.

-SUN Microsystems ha deciso lo scorso novembre di rendere disponibile buona parte del proprio software in modalità aperta, dando un notevole ritorno di immagine e credibilità all'intero movimento Open Source;

-Linux guadagna terreno anche nei dispositivi mobili. Molti produttori stanno attualmente sperimentando versioni embedded di Linux mentre altre aziende (ad esempio Motorola) hanno già annunciato che Linux gestirà tutti i propri cellulari di nuova generazione;

-Firefox esce dalla nicchia e diventa un prodotto di massa. Da poco festeggiato il 100 milionesimo download, il popolare browser Open Source dimostra che qualsiasi prodotto a codice aperto può competere degnamente nel mercato quando sia pensato e costruito su basi solide (ad esempio, un'interfaccia grafica finalmente al livello dei prodotti commerciali);

-i venture capitalist rompono gli indugi e decidono di investire seriamente in aziende che promuovono codice aperto. Oltre 400 milioni di dollari sono stati investiti nel 2005 ed è ragionevole pensare che tale cifra sia destinata ad aumentare.

## 7. Una barriera all'Open Source: Il caso Palladium

Sulla scia della sempre più radicata affermazione del paradigma Open, molto importante è anche la campagna informativa che sta svolgendo la comunità informatica "pesante" sui rischi e sulle distorsioni che potrebbe creare l'introduzione nel campo informatico della tecnologia TCPA (Trusted Computing Platform Alliance), introdotta nell' hardware e nel software non solo dei PC, ma anche dei dispositivi tecnologici e degli elettrodomestici che usiamo quotidianamente.

TCPA ha lo scopo di realizzare una nuova piattaforma informatica, in grado di fornire un aumento della sicurezza nei Personal Computer: così cercano di vendercela, ma in realtà TCPA avrà solo il compito di blindare totalmente i nostri computer e di privarci delle libertà che ancora ci restano.

"Palladium" si chiama il software che Microsoft ha intenzione di incorporare nelle future versioni di Windows Vista, e nelle prossime versioni Mac e l'hardware su cui verrà costruito è proprio TCPA.

Si sta parlando dell'introduzione sul mercato dei primi sistemi TC-compatibili, vale a dire i primi sistemi conformi alle specifiche dettate dal consorzio TCG (Trusted Computing Group).

La funzione principale di questo sistema è quella di fornire una piattaforma informatica che non permetta di modificare i programmi utilizzati dall'utente; ma non solo. Il cuore di questo sistema è un chip chiamato **Trusted Platform Module (TPM)** meglio conosciuto come Fritz che

custodisce al proprio interno una chiave crittografica tale da criptare tutte le informazioni che passeranno attraverso i nostri PC.

In più ogni volta che accenderemo il computer, saremo identificati da una password, o meglio ancora da un controllo biometrico e Fritz farà un po' da vigile dei nostri PC, con il potere di impedirci di utilizzare file, fare copie, collegarci ad un sito web: insomma ogni singolo atto verrà controllato e sottoposto all'approvazione di Fritz.

Da questo ne deriva che l'obiettivo più immediato non solo è quello di gestire il copyright sui beni digitali, aprendo così un'infinità di nuove vie per il marketing su beni e servizi, ma TCPA/Palladium renderà molto difficile l'utilizzo del software senza licenza.

Le aziende di software potranno rendere molto più difficile il passaggio ad un prodotto concorrente permettendo la lettura dei file esclusivamente con il proprio software: è a discrezione di una applicazione definire le politiche di sicurezza per i propri file, e non dovremo meravigliarci se Microsoft cerchi di ottenere qualsiasi tipo di accordo con i fornitori di contenuti.

Insomma non sarà più possibile prestare musica ad un amico, o farci prestare un DVD da nostra zia, a meno che non si utilizzi lo stesso software.

La pressione per lo sviluppo di software TCPA-compatibili, apporterà un vantaggio competitivo sempre più grande e maggior profitti per aziende che sono sempre più in grado di gestire anche i mercati dei prodotti complementari. Ad esempio la vendita di stampanti che riconoscono l'originalità delle cartucce di inchiostro, impediscono all'utente di usare prodotti di concorrenza; tutto questo a spese dell'innovazione e della crescita tecnologica.

Queste politiche renderanno la vita sempre più complessa, sarà sempre più difficile sfuggire al sistema e sarà un modo sempre più attraente per fare affari.

Molte aziende che si occupano di sicurezza verranno messe da parte, e chiunque gestisca il chip Fritz, come unico strumento di controllo, avrà sempre più un potere monopolista, promuovendo la censura e negando la libera circolazione dei file di proprietà mettendo a rischio l'enorme eredità che Gutenberg ci ha lasciato.

Inoltre il TCPA potrebbe uccidere il software libero e le piccole aziende di programmazione, perché, di fatto potrebbe porre ostacoli tecnici ed economici al processo di certificazione; inoltre i sistemi operativi privi di supporto per il Trusted Computing, non possono accedere ai dati che appartengono ad applicazioni Trusted Computing, e ciò significa che se il TC diventasse la norma, Linux non sarebbe in grado di accedere a nessun documento elaborato da un computer TC, come file di testo, pagina web, o filmati.

L'unica speranza è che le tecnologie TCPA siano sempre disabilitabili dall'utente, e soprattutto sperare in una forte reazione dei consumatori che potrebbero anche decidere di non comprare hardware TCPA.

## 8. Conclusioni

Dal punto di vista economico, il fenomeno Open Source costituisce un'importante innovazione nel processo di produzione del software in contrapposizione al tradizionale modello proprietario del mondo commerciale; questo è il risultato di un percorso evolutivo tecnologico economico e sociale che ha coinvolto più di una generazione di ricercatori programmatori ed utenti.

Sebbene all'inizio interessasse solo una ristretta comunità di esperti è diventato nel tempo oggetto di attenzione per imprese istituzioni e ricercatori.

Come per altri fenomeni con forti caratterizzazioni tecnologiche economiche e sociali, è difficile dire quali saranno gli sviluppi.

I segnali di diffusione del software Open e di estensione del concetto di apertura fanno pensare che siamo ancora in una fase giovane del ciclo di vita e che molti interessanti sviluppi sono all'orizzonte: lo sviluppo futuro dipende molto dalla solidità del modello di sviluppo e dalla sua riproducibilità.

Per una buona valutazione del modello è necessario esaminare le caratteristiche organizzative e i processi attraverso il quale si organizza il software Open Source;

allo stesso tempo si dimostrerà come il software aperto può convivere con quello proprietario, ma può anche originare una serie di nuove forme di

organizzazioni economiche nelle quali elementi open e proprietari si combineranno variamente.



## **CAPITOLO 2**

# **L'OPEN SOURCE COME NUOVO MODELLO DI BUSINESS**

### **Introduzione**

In questo capitolo, analizzeremo l'Open Source come nuovo modello di business, sia dal punto di vista di architettura, che dal punto di vista economico.

Analizzeremo inoltre la struttura della comunità Open, e gli attori che la compongono, e osserveremo come da questa comunità composta da soggetti eterogenei, che può essere vista anche come un calderone ribollente di idee, continuamente alimentato da soggetti volontari, chiunque può attingere per loro necessità, senza preoccuparsi di dover contribuire con denaro, se non con altri prodotti della conoscenza.

# 1. La struttura organizzativa della comunità Open Source

I prodotti Open sono realizzati all'interno di comunità composte da soggetti eterogenei che collaborano tra loro, spinti da interessi e motivazioni diverse.

Raymond nel suo contributo al mondo Open con il testo "La cattedrale e il Bazar", analizza cosa aveva reso possibile il successo dell'Open Source.

Allora si parlava ancora di free software e non di Open Source, e nel libro Raymond spiega cosa aveva reso possibile lo sviluppo del free software e perché erano in grado di produrre free software pur violando le leggi standard dell'ingegneria informatica.

Raymond fa il confronto tra due stili di sviluppo diversi ed opposti:

-il primo chiuso e tradizionale che chiamerà Cattedrale, caratterizzato dalle specificazioni degli obiettivi e da piccoli gruppi di progetto gestiti in modo autoritario e gerarchico, in cui tra una release e l'altra trascorrono lunghi intervalli di tempo.

E' il metodo tradizionale di sviluppo, in cui i gruppi dedicati a questa attività sono di piccole dimensioni. Il progetto e la funzionalità sono chiari prima di iniziare lo sviluppo, i piani sono completi, dall'architettura alla progettazione, dallo sviluppo alle integrazioni alle verifiche, sono ben chiari a tutti i documentati, e i feed-back, sono ricercati solo durante i cicli di verifica;

- l'altra era una struttura decentralizzata, basata su rapporti paritari: uno stile tipo Bazar dove gli intervalli tra una release e l'altra sono brevi, e

vi è una costante sollecitazione da parte di persone formalmente al di fuori del progetto.

Vi è quindi un intenso processo paritario di continua revisione, il feedback viene ricercato da subito il più frequentemente possibile, “e la cosa incredibile” dice Raymond “è che più si analizza questa situazione e più si nota che i tradizionali vantaggi della revisione paritaria, assolutamente indipendente, sembrava essere la carta vincente e dare ottimi risultati”.

Apparentemente questa struttura potrebbe sembrare disorganizzata e disordinata, e pensare che un simile sistema possa continuare a mantenersi sembra inconcepibile, e ancora più improbabile sembra che la sua apparente disorganizzazione possa produrre risultati concreti.

I prodotti sviluppati negli ultimi anni hanno però permesso di dimostrare l'esistenza anche di alcuni criteri organizzativi all'interno di questa comunità, criteri che la caratterizzano e danno una vera e propria struttura al nuovo paradigma Open Source.

Dall'osservazione di alcuni progetti Open è stato possibile individuare la presenza di alcuni soggetti o categorie:

-**UTENZA:** questa categoria è composta da tutti gli utenti che utilizzano i prodotti Open senza partecipare direttamente allo sviluppo del prodotto stesso, perché non capaci o perché non possiedono il tempo o gli strumenti adatti per farlo .

Questa componente è fondamentale, perché consente di sottoporre il prodotto, e quindi il codice, ad un tempestivo ed esauriente processo

d'osservazione e controllo del funzionamento del prodotto, che costituisce un feedback per gli sviluppatori.

-PROSUMER: sono i programmatori che partecipano direttamente allo sviluppo dei prodotti, non solo per soddisfazioni proprie, ma anche per aumentare il loro 'capitale-professionale' e per dare validi contributi alla comunità. Il gruppo dei prosumer costituisce il nucleo di sviluppo dei codici, ed è composto da persone con background culturali e professionali molto diverse tra loro.

-TEAM DEI LEADER: è formato sostanzialmente da una stretta cerchia di prosumer che hanno partecipato sin dall'inizio alla definizione e alla produzione dei prodotti Open Source. I leader hanno l'importante compito di coordinare e la comunità di sviluppo, diffondendo i contributi del progetto e i feedback provenienti dalla comunità, diventando un punto di riferimento per tutti i soggetti, aziende e istituzioni comprese interessati più o meno direttamente al software Open Source.

-IMPRESA: contribuiscono con la comunità attraverso l'utilizzo il finanziamento ed eventualmente lo sviluppo congiunto del software.

In tale modo possono influenzare i percorsi di sviluppo e di evoluzione dei prodotti oppure possono osservare la comunità per migliorare la propria struttura organizzativa ed il proprio business.

-ISTITUZIONI: costituite da organizzazioni senza fini di lucro, governi e pubbliche amministrazioni interessate all'utilizzo e alla diffusione di prodotti Open Source.

-TEAM VIRTUALE: la comunità è formata da un team virtuale che si incontra in rete attraverso forum o mailing list, in cui periodicamente entrano ed escono soggetti appartenenti alle diverse categorie sopra citate attuando così la diffusione di idee concetti e strategie per tutta la comunità, evitando dispersioni e rallentamenti al flusso delle informazioni e delle decisioni.

Le caratteristiche della struttura precedentemente descritta, possono essere rilette e approfondite tenendo in considerazione altre variabile fondamentale della filosofia Open, che sono il coordinamento e la reale distribuzione dei ruoli.

Variabili che a loro volta condizionano la forma della struttura della comunità, sottolineando il grado di capacità di ogni categoria di agente di assumere i diversi ruoli in misura diversa.

Di questo ne parleremo nel terzo capitolo in modo più dettagliato.

## **2. La comunità Open Source: le motivazioni**

A questo punto nasce spontaneo domandarsi il perché ci sono persone che vogliono leggere scrivere e fare revisione di tutti quei codici, senza nessun tipo di ricompensa.

Le motivazioni di chi collabora al mondo Open, man mano che il fenomeno si concretizzava sempre più, fino a diventare un vero e proprio sistema alternativo di business, hanno destato molta curiosità da parte delle teorie economiche.

La necessità di comprendere quali siano gli incentivi di chi lavora a progetti Open ci riconduce al classico calcolo costi-benefici qualora se ne evidenziano chiaramente gli elementi costitutivi.

Dal lato dei costi è facile capire che a livello di infrastrutture i costi sono per lo più a zero, di fatto tutti i programmatori possiedono già un computer per ragioni di studio o di lavoro, o addirittura scrivono codici durante le ore di lavoro per cui anche la connessione ad internet, non costituisce per lo più un problema.

La partecipazione alla comunità Open Source, comporta per questi soggetti, dei costi-opportunità: un programmatore che sceglie di lavorare ad un progetto Open deve avere delle motivazioni economiche poco contingenti che giustificano la partecipazione stessa, la rinuncia almeno parziale al tempo libero e la rinuncia alle ricompense economiche.

Queste rinunce però saranno colmate dalle opportunità di apprendimento e di miglioramento della propria reputazione professionale e quindi aumenteranno il proprio “capitale professionale”.

I principali investimenti hanno per oggetto il tempo e le proprietà intellettuali: un'indagine eseguita su un campione di 1700 partecipanti ad un newsgroup per la risoluzione dei problemi legati al server Apache Lakani e Von Hippel ci racconta che oltre l'80% di chi risponde a domande di ausilio tecnico, ci impiega non più di 5 minuti.

Questo perché la maggior parte dei programmatori conosce già le risposte ed il funzionamento dei principali strumenti di posta elettronica, per cui riescono a farle girare rapidamente in rete zippando i file e riducendone la grandezza.

A fronte dei bassi costi, molti sono i benefici che spingono la comunità ad arricchire il movimento Open:

- in primo luogo la gratificazione intrinseca della programmazione come esperienza estetica o anche intesa come fun to program;
- il concetto della “gift economy”, ossia altruismo, priorità, reputazione, e consumo che è un modo per trovare e mantenere relazioni sociali racchiudendo l'obbligo alla reciprocità di scambio di informazioni; ciò si realizza anche quando lo scambio avviene tra persone sconosciute;
- la reputazione e l'autoproduzione sono da sempre i motori di sviluppo della comunità scientifica: la possibilità di guadagnarsi una reputazione nel creare codici sorgenti e competere con altri programmatori per lo sviluppo di una nuova soluzione informatica e quindi sottolineare le proprie capacità, è chiaramente interpretabile alla luce della teoria economica.

La visibilità che un giovane programmatore può avere nel collaborare alla programmazione Open può essere maggiore di quella che si potrebbe acquisire in una grande azienda di software, poiché in rete vi accedono programmatori di tutto il mondo, e così molti esperti hanno fatto leva sulla reputazione acquisita nelle comunità Open per fondare nuove imprese di successo; molti dei progetti Open inoltre nascono come autoproduzioni, per soddisfare una domanda in assenza di offerta.

La presenza di una comunità Open produce effetti anche sulla catena del valore, soprattutto dal punto di vista della distribuzione, e dal punto di vista della produzione: infatti la disponibilità dei codici sorgente favorisce la nascita di diversi distributori in competizione sia sulla qualità dei servizi offerti, sia sui relativi prezzi. Inoltre le imprese possono apportare autonomamente modifiche e quindi produrre in base alle proprie esigenze, altrimenti impossibile in presenza di prodotti proprietari coperti da copyright.

Considerazioni diverse vanno fatte nel caso in cui un'azienda decide di partecipare direttamente alla creazione di un software in collaborazione con le comunità Open Source, ovvero senza limitarsi ad osservare la comunità ed utilizzare i suoi prodotti.

Il coinvolgimento dello sviluppo richiede di mettere a disposizione risorse tecnologiche ed umane, e tale impegno e i relativi costi devono trovare riscontri economici che non possono ovviamente derivare dai ricavi di vendita.

Una prima motivazione che potrebbe spingere un'impresa a prendere parte ad un progetto Open, è la limitazione dei costi di ricerca e sviluppo: infatti



l'impegno per queste attività, può essere condiviso con l'intera comunità; da qui ne deriva che i risultati ottenuti dalla cooperazione non possono essere rivendicati dalla singola impresa, ma si convertiranno in vantaggi in termini di qualità dei prodotti e dei processi interni.

In aggiunta l'Open Source può risultare un'utile, se non unica, arma efficace per contrastare i monopoli: attraverso l'Open le imprese possono contribuire allo sviluppo di standard imparziali e di ottima qualità anche in presenza di altri prodotti dominanti. La promozione degli stessi standard indipendenti dal business delle singole imprese è sicuramente un incentivo alla competizione con conseguenti vantaggi in termini di qualità, affidabilità stabilità e convenienza dei prodotti.

### 3. Il problema del free riding

L'idea che il proprio contributo sia fondamentale per la riuscita del progetto ed il grande valore attribuito all'obiettivo del progetto stesso, ci riporta, attraverso la struttura degli incentivi ed una semplice analisi dei costi benefici, e al problema legato al free riding: cioè la possibilità di avere un prodotto o un servizio ricavandone un beneficio senza che sia richiesto qualcosa in cambio.

L'esistenza dei free riders nella comunità Open Source, che deriva dalla disponibilità dei codici sorgente e dalla libertà concessa ai soggetti di modificarlo, ha dato al bene software le caratteristiche di un vero e proprio bene collettivo, e in quanto tale ne possono beneficiare tutti, anche chi non ha assolutamente contribuito alla produzione: quindi chiunque potrebbe utilizzare il bene senza sostenere nessun costo.

Per ciascun individuo quindi la risposta ottimale sarebbe quella di non contribuire per niente sfruttando poi la contribuzione altrui per avere benefici, senza dover affrontare nessun tipo di costo .

Ma se facessero tutti così, il prodotto non esisterebbe ed il problema del free riding, cioè ciò che da benefici individuali porterebbe a risultati collettivamente subottimali.

Una realtà di free riding totale, ossia l'ipotesi che tutti i soggetti scelgano una contribuzione nulla, si osserva raramente sia nel mondo reale, sia negli esperimenti di laboratorio, ma soprattutto nella realtà Open, dove si cerca di

sviluppare un software attraverso l'azione collettiva, che vada al di fuori dei canali commerciali, sfatando così la tesi di Olson secondo la quale vi è una relazione inversa tra ampiezza del gruppo e la probabilità che si raggiungano con successo gli obiettivi di un'azione collettiva.

Secondo Olson, all'aumento della dimensione del gruppo diminuisce la probabilità che un comportamento di defezione, cioè di tradimento, possa essere individuato con corrispondente aumento dell'incentivo a godere del bene senza riportare nessun tipo di contributo.

L'analisi di Olson però non tiene in considerazione alcune variabili il cui effetto è determinante per la realtà Open: prima di tutto l'eterogeneità, dei soggetti partecipanti alle azioni collettive, ma ancora di più l'eterogeneità di risorse ed interessi. Un piccolo insieme di soggetti molto motivati e dotati delle risorse necessarie è in grado di fornire il bene senza la cooperazione altrui.

Da qui ne deriva il concetto di sottogruppo efficace di cui ne parleremo in seguito.

## **4. Il problema della diffusione: il software Open e le esternalità di rete**

I meccanismi alla base della diffusione dell'Open Source, non sono ancora chiari persino ai suoi padri fondatori. Toarvalds afferma: " ho avvertito la transizione dall'essere l'unico utente di Linux all'essercene un centinaio, ma quella da cento ad un milione di utenti mi è sfuggita".

Data la vasta letteratura circa la diffusione delle tecnologie è importante circoscrivere l'ambito di analisi facendo riferimento alle caratteristiche del bene oggetto di diffusione e dei soggetti chiamati a domandarlo.

Partendo dal presupposto che secondo Shapiro<sup>6</sup> (1999) è informazione tutto ciò che può essere digitalizzato', il software è informazione.

Dal punto di vista economico, la produzione di informazione si caratterizza per avvenire con alti costi fissi e costi marginali trascurabili: tutta la massa dei costi è concentrata nella prima copia i cui costi sono estremamente ridotti. Dal lato della produzione il software presenta forti economie di scala: quante più copie si producono e si vendono i costi della prima verranno assorbiti dal bassissimo costo delle successive

Dal lato della domanda si nota invece il fenomeno delle esternalità<sup>7</sup> di network: Katz e Shapiro<sup>8</sup> ne hanno dato la più esauriente affermazione: l'utilità di un bene cresce con il numero di utenti che utilizzano il bene stesso.

---

<sup>6</sup>

<sup>7</sup> esternalità: situazione in cui il comportamento di un operatore causa vantaggi o danni ad altri operatori, non interessati dalla transazione, senza che per ciò si paghi un prezzo.

<sup>8</sup>4/5 Vedi testo "Network Externalities, Competition and Compatibility"(1985)

Accanto a questa definizione, gli autori forniscono una accurata classificazione delle fonti di esternalità: così l'esternalità può essere diretta o indiretta a seconda che derivi dalla appartenenza fisica dei soggetti ad una rete o all'appartenenza a market-mediated effects nel caso dei beni complementari.

Il caso più frequentemente citato come esempio di esternalità diretta è quello del telefono o del fax, la cui utilità per i potenziali utilizzatori del bene, dipende dal numero di altri soggetti che fanno parte della rete corrispondente; chiaramente vi è l'esternalità di network diretta anche quando i consumatori appartengono ad una rete che non è fisica, ma virtuale, come nel caso degli utenti di uno stesso sistema operativo.

L'esternalità indiretta, invece caratterizza il così detto paradigma hardware-software in cui due o più beni complementari sono uniti a formare un sistema che per operare ha necessità dell'apporto di tutte le componenti. L'esternalità positiva deriva dal fatto che un bene complementare diventa più economico e disponibile in maniera più veloce al crescere del mercato del bene ad esso compatibile.

L'esternalità dei network hanno rilevanti conseguenze sulla struttura del processo di diffusione, che nel caso particolare corrisponde al problema dell'affermazione di uno standard: un piccolo vantaggio iniziale a favore di uno standard è sufficiente perché esso si imponga, conquistando tutto il mercato che in seguito resta bloccato su di esso per la presenza di elevati costi di switching (cioè costi di cambiamento). Se un software riesce a guadagnare una fetta consistente di mercato, si innesca un circolo vizioso per cui i consumatori avranno ancora più incentivo ad adottarlo, crescerà il

mercato dei prodotti complementari e il software dominerà il mercato divenendo lo standard.

Ma analizziamo meglio quali sono le conseguenze delle esternalità di rete.

## 4.a).I rendimenti crescenti ed il fenomeno del lock-in

I concetti di lock-in (cioè quel fenomeno che avviene quando i costi di cambiamento sono talmente elevati da scoraggiare il passaggio da un prodotto ad un altro) e path-dependence (cioè la dipendenza da una via di accesso) sono stati recentemente oggetto di un acceso dibattito rispetto al quale, proprio la natura delle tecnologie Open Source consente di formulare alcune precisazioni rilevanti: Liebowitz e Margolis<sup>9</sup>, sottolineano come l'analisi degli standard da parte della teoria economica, enfatizzi il rischio di incorrere nella 'trappola' esemplificata dalle conclusioni di David (1985).

Nell'articolo scritto nella rivista *American Economic Review*, circa la battaglia Qwerty-Dvorak David commenta: 'se ci sono solo tastiere QWERTY, i dattilografi studieranno solo quelle tastiere; ma se i dattilografi studiano solo tastiere QWERTY ci saranno solo tastiere QWERTY'.

Per spiegare come questo si verifichi, David fa riferimento allo scritto di Arthur sulla rivista *The Economic Journal*, circa i rendimenti crescenti nel processo di diffusione di due tecnologie in competizione tra loro. Per cui, più una tecnologia è adottata, più alta è la probabilità che lo sarà in futuro.

I rendimenti crescenti sono immediata conseguenza delle esternalità di network: in una rete si creano facilmente esternalità di rete in quanto il valore di un prodotto per l'utilizzatore, dipende dal numero di altri utilizzatori.

---

<sup>9</sup> vedi testo: *Market Process and Selection of Standard* (1996)

Data allora la presenza nel processo di diffusione del software delle esternalità di network dirette, il meccanismo del lock-in sembra essere un esito inevitabile: se un software riesce a guadagnare una fetta consistente di mercato, si innesca un circolo vizioso per cui i consumatori avranno ancora più incentivo ad adottarlo, crescerà il mercato dei prodotti complementari e il software dominerà il mercato divenendo lo standard.

Nel caso Open Source invece, si contraddice questa previsione, in quanto questo guadagna terreno sullo standard dominante rappresentato da Microsoft facendo pensare che i due paradigmi, quello gratuito e quello non gratuito possano arrivare ad un equilibrio per cui i due riusciranno a vivere fianco a fianco; e non solo. L'Open Source non sfugge alle logiche di lock-in, ma in realtà ne prospetta una forma ancora più esasperata almeno nelle zone di mercato in cui si affaccia, tale da lasciar intravedere una previsione di 'lock-in' non solo di prodotto, ma anche di 'sistema' rendendo così ancora più difficile se non impossibile, il ritorno a standard chiusi, una volta passati a quelli aperti.

Questo conferma ancora una volta, come il sistema Open Source, basato sulla collaborazione e cooperazione, potrebbe essere più efficace di un sistema chiuso.

Bisogna quindi cercare di spiegare come un nuovo standard possa affermarsi in presenza di un altro già consolidato e come i due possano convivere.

Ancora una volta ci rifacciamo alle ipotesi di Arthur, secondo cui esistono piccoli eventi casuali che possono dare il necessario vantaggio ad una delle due tecnologie



Se fosse, così non esisterebbe nessun tipo di difesa per la possibile affermazione di tecnologie inefficienti a scapito di un'altra che se adottata sarebbe sicuramente superiore: è proprio quello che spiega David circa le macchine da scrivere QWERTY, di fatto inferiori per prestazioni alle DSK di Dworak, ma presenti da prima sul mercato.

Arthur continua osservando come la dimostrazione matematica del lock-in dipenda dalle assunzioni iniziali, mutando le quali l'autore riesce a dimostrare come la nuova introduzione di una tecnologia potrebbe disseminare successo in un mercato, a discapito delle esternalità positive, già esistenti.

Le ipotesi di Arthur, sono difficilmente sostenibili nel caso del software Open: con la "virgin market competition" (1997) egli assume che le due tecnologie in competizione arrivino sul mercato nello stesso momento e che le scelte sulle adozioni tecnologiche non sono rivedibili.

Questo potrebbe non essere sostenibile nel caso del software dove il sistema operativo Linux si trova a competere con il predominio decennale dei sistemi già esistenti, ed inoltre non è mai vero che il mercato potenziale di una tecnologia è infinito e che le scelte della sua adozione non sono rivedibili.

Questo vale in particolar modo nel caso di programmi per elaboratore che hanno un mercato in crescita ma necessariamente limitato (il loro utilizzo non è indispensabile per ogni attività umana) e non prevedono investimenti in capitale fisso tali da determinare proibitivi costi di revisione delle scelte. Difficilmente sostenibile è poi l'indipendenza delle scelte ipotizzata da

Arthur, poiché l'influenza degli altri soggetti sulle scelte del sistema da adottare è ampiamente esaminata dalla teoria economica.

La diffusione avviene per contagio nei confronti dei non adottanti da parte degli adottanti, fino ai recentissimi modelli di interazione locale in cui le decisioni di ognuno sono influenzate da quelle di un sottoinsieme di soggetti.

Per spiegare la diffusione del sistema Linux in sostituzione di WindowNT, secondo Dalle e Julien<sup>10</sup>, per un utente di un sistema operativo non è tanto rilevante il numero totale dei soggetti che lo utilizzano quanto il numero di coloro che lo fanno all'interno del gruppo con cui normalmente l'individuo interagisca.

Da qui ne emerge una diversa fonte di "eterogeneità sociale" che sottolinea la diversità tra gli agenti, che non riguarda soltanto le loro caratteristiche intrinseche, ma anche la rete delle relazioni sociali che sono in grado di attivare al fine di ottenere informazioni che li aiutino nel loro processo di scelta.

Le caratteristiche di tale rete sono diverse a seconda che i nodi siano rappresentati da utenti di software Open Source oppure da chi impiega software commerciale.

In altri termini, per l'affermazione degli standard Open non è stata fondamentale la qualità della soluzione, ma al contrario è stato fondamentale e determinante il raggiungimento anticipato di una massa critica rispetto alle soluzioni alternative.

---

<sup>10</sup> Vedi: Trattato Linux or Some Explanation into Economics of Free Software (1999)

E' inoltre necessario analizzare il fatto che normalmente l'affermazione di uno standard tende a produrre situazioni di monopolio, da cui normalmente ne deriva un aumento dei prezzi e il fatto che l'innovazione tende a dipendere dai monopolisti stessi: in tal caso l'innovazione è prodotta in un sistema essenzialmente chiuso, e quindi la sua diffusione andrà sicuramente a rilento.

Nel caso dell'Open Source, invece, l'innovazione si sviluppa in contesti in cui le proprietà intellettuali sono rese di dominio pubblico, garantendo un'elevata qualità che si aggiunge al contenuto innovativo dei prodotti stessi, spostando così il fulcro della competizione, unicamente sulla qualità ed in misura minima o irrilevante, sul prezzo.

Ma analizziamo meglio il concetto di massa critica, e osserviamo quanto ne sia importante il più rapido raggiungimento.

## 5. Il concetto di Massa Critica

Trasportando il concetto di Massa Critica dal campo fisico a quello economico, si parla di effetto Massa Critica, quando determinate variabili che caratterizzano un processo superano una data soglia per cui il fenomeno esplose facendo sì che il sistema abbandoni l'equilibrio stabile su cui si trovava posizionato, per passare ad un equilibrio altrettanto stabile.

Questo potrebbe aiutarci ad analizzare il passaggio da tecnologie standard a quelle Open: nei modelli di diffusione di tecnologie la suddetta variabile è rappresentata dal numero di adottanti e Dalle e Julien<sup>11</sup> analizzano il fenomeno della Massa Critica attraverso delle simulazioni.

Osservano che l'emergere di suddetta variabile dipende dal valore assunto da un determinato parametro alfa, che è una misura della preferenza degli utenti per l'utilizzo standardizzato della tecnologia presa in esame all'interno della loro rete di riferimento: più grande è alfa più gli attuali utenti sono propensi ad adottare il sistema attuale, cioè WindowNT.

Tale parametro è anche una misura della forza del 'proselitismo' degli utenti di Open Source: minore è alfa, maggiore è la loro forza all'interno di una rete locale. Le simulazioni evidenziano la presenza di Massa Critica quando il valore di alfa è sufficientemente basso; in più analizzando la diffusione di due tecnologie in competizione tra loro, le cui performance crescono al crescere del numero di utenti che le utilizza, e si conclude che il 'tempo di

---

<sup>11</sup> Linux or Some Explanation into Economics of Free Software (1999)

attesa' prima dell'insorgenza della Massa Critica decresce quanto più sono ottimistiche le aspettative dei soggetti.

Sempre dalle simulazioni inoltre si evince che:

- il passaggio da una tecnologia all'altra è più veloce quanto più i soggetti sono eterogenei
- - che in presenza di un gruppo piccolo di soggetti, dotati di propensione doppia ad innovare, in base alla media del gruppo, la Massa Critica emerge quasi subito;
- ancora una volta risulta evidenziato il ruolo degli hacker e della loro cultura del movimento Open Source.

## **6.IL Coordinamento: Modularità del software e il ruolo della rete Internet**

Un altro elemento importante da analizzare è il coordinamento della comunità Open: alcuni protagonisti del movimento sono giunti a riprendere il concetto di “ordine spontaneo”, (Perkins 1999) paragonando il meccanismo di coordinamento spontaneo e decentrato a quello che sta alla base dell’ incontro tra domanda ed offerta nei mercati ed al funzionamento della mano invisibile.

Non costituendo un’entità monolitica i programmi possono essere divisi in moduli da ricomporre ed utilizzare in vari modi, in modo tale che i programmatori possano aggiungere moduli senza che lavorino simultaneamente sugli stessi punti.

Ma una domanda che nasce spontanea è: come avviene la comunicazione tra i programmatori?

L’esplosione del movimento Open, ha coinciso di fatto con la sorprendente diffusione dello strumento Internet, che ha permesso la fornitura di connettività al grande pubblico a prezzi sempre più bassi.

La comunità infatti si è talmente appoggiata a questa infrastruttura di comunicazione che moltissimi progetti sono stati sviluppati da programmatori che non si sono mai incontrati fisicamente.

In più grazie anche al livello dell’architettura che è resa modulare dall’evoluzione dei linguaggi di programmazione, è facile intuire che il

coordinamento gerarchico non è strettamente necessario, al contrario, rischierebbe di reprimere le motivazioni intellettuali, estetiche e di divertimento, che sembrano essere intrinseche alla comunità dei programmatori.

In cerca di una possibile spiegazione, alcuni protagonisti del movimento hanno ripreso il concetto di ordine spontaneo, paragonando il meccanismo di coordinamento spontaneo e decentrato a quello che sta alla base dell'incontro tra domande e offerta nei mercati e al funzionamento della mano invisibile.

E' importante fare un'accurata analisi tenendo in considerazione sia il lato dello sviluppo tecnologico di produzione del software, sia i nuovi strumenti di comunicazione resi possibili dalla rete Internet.

Per quanto riguarda il primo, non costituendo una realtà monolitica i programmi possono essere divisi in moduli da ricomporre ed utilizzare in vario modo: ogni singola parte è progettata per svolgere ottimamente e nel modo più semplice possibile una singola funzione.

La modularità, infatti è indispensabile per lo sviluppo parallelo e decentralizzato dei progetti, poiché facilita il controllo e l'introduzione di nuove parti di codice nel progetto: persino Microsoft sta cercando di introdurre un certo grado di modularità nello sviluppo dei propri prodotti, e l'introduzione dei linguaggi di programmazione ad oggetti ne è la dimostrazione.

Tuttavia Microsoft preferisce ancora oggi sviluppare i propri prodotti in modo integrato: l'approccio integrale al prodotto consiste infatti nell'unione diretta (linking) dei diversi codici sorgente nelle fasi finali dei processi di

sviluppo, senza attivare una preventiva progettazione modulare nelle fasi iniziali. La ragione che spinge Microsoft all'utilizzo di questa strategia è prettamente commerciale: l'approccio integrale permette una facile distribuzione e quindi la vendita dei diversi prodotti uniti in un unico pacchetto.

L'integrazione dei prodotti per l'ottenimento di un vantaggio commerciale è usata da Microsoft nei suoi principali prodotti. Ad esempio l'integrazione del sistema operativo Windows con il codice del browser Explorer ha permesso la veloce diffusione di quest'ultimo sfruttando la diffusione del primo (esternalità di rete).

L'integrazione del codice tramite linking, oltre a costringere indirettamente la clientela all'utilizzo di Explorer, ha reso difficile la rimozione del browser dal sistema operativo.

Di conseguenza anche la parte di clientela che avrebbe preferito utilizzare il prodotto della concorrenza, ovvero Netscape, si è trovata avanti alla scelta di decidere se utilizzare l'intero prodotto Microsoft, oppure usare un altro sistema operativo. La maggior parte degli utenti abituati ad usare il sistema operativo Microsoft, scelse la prima strada.

Tuttavia ai vantaggi commerciali, si contrappongono altri svantaggi di natura gestionale: attraverso l'integrazione, risulta difficile organizzare un reale sviluppo parallelo e decentralizzato dei prodotti.

Un'altra importante differenza, riguarda la valutazione del concetto di parallelismo nei due modelli.



Per Microsoft il parallelismo è sinonimo di ridondanza e scarsa ottimizzazione degli sforzi: in effetti il fatto di avere due o più persone impegnate nella stessa attività è considerato uno spreco di risorse.

Viceversa nella comunità di Linux il parallelismo è considerato un vantaggio: la sovrapposizione degli sforzi è sostenibile in quanto nessun programmatore viene ricompensato economicamente e in modo diretto dalla comunità.

Il parallelismo, come spiegato, permette di esplorare uno stesso problema da diversi punti di vista, e non solo: nemmeno in Linux il parallelismo equivale ad avere una molteplicità di programmatori impegnati nello sviluppo delle stesse linee di codice, ma essi preferiscono, nella generalità dei casi, partire da questi risultati per arrivare a qualcosa di nuovo.

La quasi totalità di sviluppatori del kernel di Linux ( il cuore del sistema) lavora in modo ordinato ai subsistemi del kernel, in cui i file hanno lo scopo di far dialogare il computer con le periferiche o di permettergli la lettura dei file video o audio.

I concetti di Modularità e Riutilizzabilità sono alla base dell'utilizzo del software e sono resi ancora più effettivi dalla condivisione di un protocollo comune(linguaggio di programmazione) dove gli errori sono individuati e corretti attraverso il meccanismo della compilazione.

Secondo Linus Torvald, nella modularità, sta la chiave del successo dell'Open Source; egli afferma: “ quello che ci serve è un sistema il più possibile modulare, per non avere gente che lavori simultaneamente sugli stessi punti” e la fonte della modularità di Linux è il suo kernel monolitico.

Il coordinamento è reso possibile quindi da una oculata scelta tecnica che in parole semplici è quella di tenere il kernel piccolo e limitare al minimo il numero di interfacce e di altri vincoli al suo sviluppo in modo che i programmatori possono aggiungere moduli senza pestarsi i piedi l'uno con gli altri.

Siamo quindi di fronte ad un fenomeno di coordinamento che sfrutta profonde innovazioni intervenute:

- a livello di attività tecniche che sono rese più interfacciabili dalla evoluzione della tecnologia software;
- a livello dell'architettura, che è resa modulare in generale dall'evoluzione dei linguaggi di programmazione ed in particolare dalle scelte di strategie di linux sul kernel;
- a livello di costi di comunicazione, che sono stati abbattuti o resi trascurabili dall'avvento di Internet.

Sotto queste condizioni il coordinamento gerarchico non è strettamente necessario: al contrario esso viene assicurato da forme di leadership basata sulle competenze di programmazione, che somigliano più alla legittimazione della comunità scientifica che alla catena di comando di un'impresa.

## **7. Il sottogruppo efficace**

A questo punto interviene Hardin con la sua teoria che sottolinea l'importanza dell'eterogeneità delle risorse e dell'interesse: un piccolo gruppo di persone motivate e dotate delle risorse necessarie, riesce a produrre il bene anche senza la cooperazione altrui formando il così detto **sottogruppo efficace**.

Il sottogruppo efficace è fondamentale per il mondo Open, ed è facile pensare che questo sia costituito dagli hacker che hanno come risorse necessarie il know how nel campo dell'informatica e che hanno come motivazione principale la soddisfazione intellettuale nel creare nuovi codici sorgente, sia per la ricerca accademica ma anche per dimostrare la loro capacità ed il loro talento.

L'azione di fornitura è inoltre resa più efficace dal fatto che il bene codice sorgente è non rivale nel consumo in una realtà come quella Open, in cui il consumatore è anche produttore e i benefici da esso apportati sono invariati rispetto al numero dei suoi utilizzatori: chiunque può scaricare il codice dalla rete, compilarlo, modificarlo ed utilizzarlo senza che questo riduca la possibilità per altri soggetti di farlo.

Limitiamoci ad immaginare il prodotto Open sviluppato solo dagli hacker più interessati, senza tener conto della collaborazione di altri soggetti che contribuiscono alla realizzazione di una collaborazione che si sta facendo sempre più estesa.

A questo punto si può fare riferimento alla teoria della Massa Critica, secondo la quale la presenza dei soggetti molto interessati fa sì che si superi la fase iniziale del processo di fornitura del bene collettivo, quella in cui per la maggior parte dei soggetti il costo della contribuzione eccede il beneficio. Superata tale fase sempre più soggetti trovano vantaggioso contribuire, innescando così un meccanismo per cui il fenomeno si autosostiene e tutti i soggetti del collettivo tendono verso la stessa scelta che è quella della cooperazione.

In quest'ottica il ruolo del sottogruppo efficace, composto dai soli hacker interessati ha il compito non di creare il prodotto, ma di creare le condizioni necessarie affinché la produzione diventi collaborativa, impegnandosi per primi a cooperare favorendo così il superamento della fase di start up dell'Azione Collettiva.

Questo schema spiega bene la dinamica dei progetti Open Source nati da un piccolo gruppo che lavora per raggiungere un particolare obiettivo: per cui se riesce a trovare soluzioni valide nella fase di start up pubblica i propri risultati in internet, pubblicizzando nelle mailing list o newsgroup l'esistenza del progetto; dopodiché questo può o attrarre un numero crescente di cooperatori che iniziano a collaborare o non destare alcun tipo di interesse.

Un'altra variabile importante è anche la "eterogeneità sociale": non è tanto rilevante il numero totale dei soggetti che utilizzano il software, quanto il numero di coloro che lo fanno all'interno del gruppo con cui normalmente l'individuo interagisce.



## 8. Conclusioni

In questo capitolo ho descritto gli attori e la struttura che compongono la comunità Open Source e i loro relativi ruoli;

ho analizzato le motivazioni che spingono i diversi soggetti a prendere parte alla comunità e quali siano i vantaggi percepiti da imprese e istituzioni.

Molto rilevante è anche l'analisi dell'organizzazione della comunità, delle regole che autonomamente si è data e come , nonostante le apparenti contraddizioni, non si venga a creare il caos, ma al massimo il 'caos organizzato' .

Ma, affinché l'Open Source non si limiti ad essere considerato solo un fenomeno sociologico, è necessario verificare anche la bontà del processo di sviluppo del software e domandarsi : 'L'Open Source potrebbe diventare un nuovo e superiore processo di sviluppo del software rispetto a quello proprietario già esistente ' ?

Molti elementi fanno ritenere che il paradigma proprietario debba necessariamente convergere verso un'accettazione della necessità di rendere trasparente almeno parte dei codici sorgente entrando di fatto dentro un modello di business così detto ibrido.

Occorre innanzitutto analizzare le modalità con cui si sviluppa il nuovo software e i relativi processi di sviluppo e pervenire a d un confronto tra essi.

Questo confronto è l'oggetto del prossimo capitolo.

## **CAPITOLO 3**

# **NUOVI MODELLI AZIENDALI, COMPETIZIONI E STRATEGIE NEL MONDO OPEN SOURCE**

### **Introduzione**

La maggior parte degli utenti che si avvicinano al mondo Open, come primo approccio pensano al software Open come un software libero nel senso di gratuito.

Non riescono a spiegarsi come sia possibile l'affermazione ed il continuo sviluppo del prodotto Open ( in senso gratuito) all'interno di un settore, come quello software, in cui esistono standard già affermati da tempo.

In questo capitolo analizzeremo, come attorno al nuovo paradigma Open Source, si sia sviluppata una vera e propria Open Source Economy, fatta anch'essa di corsa ai profitti, competizioni e strategie.

Analizzeremo la nascita di nuove tipologie di aziende nate in risposta al nuovo modello di business e osserveremo come il mondo Open abbia assorbito e fatto sue alcune caratteristiche concorrenziali che il mercato ha imposto.



# 1. Open Source Economy

Nel panorama globale del settore software fin'ora descritto, in cui il nuovo paradigma Open Source non solo si afferma sempre più, ma sempre più costituisce una vera e propria minaccia per gli standard dominanti, appare opportuno fare un'analisi della così detta Open Source Economy .

Di fatto attorno al nuovo paradigma Open si è sviluppata una nuova struttura economica meglio conosciuta come Open Source Economy, che se la volessimo analizzare rispetto alla Close Economy dei produttori di software proprietari, noteremmo che tra le due metaforicamente parlando, vi è la stessa differenza che c'è tra una multinazionale e la bottega di un artigiano.

A parte le differenze di scala, le due economie differiscono a livello di obiettivi, pratiche, ed a livello etico, conservando però un tratto in comune, cioè quello di insistere entrambe in una situazione di mercato di tipo capitalistico, e di posizionarsi entrambe nel mercato rispondendo alle sue leggi.

I caratteri salienti della Open Source Economy infatti, sono la metodologia, il permesso d'autore, il codice libero e la cooperazione tra gli individui della comunità, ma con questo non si vuole dire che lo spirito della Free Software Foundation (FSF) sia contro il mercato, ossia essa lo contempla fin dalle origini come una delle molteplici possibilità, ma certo non la più importante.

Per creare un software libero, la FSF sostiene che non necessita solo denaro, bensì tempo libero messo a disposizione da parte dei programmatori e volontà di cooperare ad un progetto libero.

Oltre la metodologia, il permesso d'autore, il codice libero e la cooperazione tra gli individui, un altro carattere saliente dell'Open Source Economy, è la nascita di nuovi modelli di business, sviluppati attorno al paradigma Open Source: i software Open sono infatti liberi, e chiunque volesse concentrarsi sull'assistenza, risoluzione dei problemi e verifiche ad hoc, potrebbe farlo generando profitto.

In questo caso non si vende il prodotto ma bensì l'assistenza, cioè si passa da un business di prodotto ad un business di servizio, ed è legittimo chiedere un riconoscimento monetario per la propria creatività. Il modo in cui viene gestita questa legittima richiesta da parte della Close Economy, secondo la filosofia Open ed il manifesto GNU, non è corretto: spremere denaro agli utenti di un programma, imponendo restrizioni sull'uso, è distruttivo perché riduce i modi di utilizzo del programma, diminuendo così anche la quantità di ricchezza complessiva del programma, limitandone inoltre la possibilità di sviluppo e di ampliamento.

Alla base di questo approccio vi è una scelta personale tanto da parte del programmatore, che sceglie la modalità di distribuzione del suo lavoro, quanto da parte dell'utente che sceglie il prodotto libero, promuovendo i processi di autoformazione che il software libero comporta, a differenza dei prodotti close proprietari già preconfezionati.

Questo non significa negare la possibilità di profitto per i programmatori, anche se è chiaro che rispetto ai sistemi di programmazione close, i

guadagni saranno sicuramente inferiori, ma d'altra parte è altrettanto evidente che nel lungo periodo l'approccio etico del modello Open, oltre a garantire la possibilità di vivere degnamente, consente anche un incremento complessivo delle libertà di diffusione delle tecnologie.

Rendere i programmi liberi significa soprattutto uscire dalle logiche di "bisogno", liberando i programmatori dal "dover programmare per forza", attraverso uno spostamento delle fonti di reddito sulla formazione e sui servizi, diminuendo così il peso della burocrazia e delle attività non produttive, in primis di marketing: la minor concorrenza infatti, diminuirebbe la mole di lavoro per i programmatori, eliminando tempistiche ferree da rispettare o fette di mercato da conquistare, con rilasci di software nuovi e con performances superiori, diminuendo anche l'importanza di quelle attività di marketing che in un circolo virtuoso per il mercato e vizioso per i consumatori, "creano" bisogni e nuove dipendenze.

Nella Open Source Economy, quindi, non è la concorrenza o la segretezza del codice a creare denaro, ma bensì nuovi business sviluppati attorno al modello Open, come per esempio il servizio di supporto al prodotto, ed il lavoro di cooperazione della comunità, che lasciano ai programmatori la libertà di gestirsi il loro tempo e le loro idee.

Data questa centralità del prodotto/servizio, il pericolo è che si affermi la tendenza a sviluppare prodotti che necessitano di molto servizio di assistenza e di lunghi tempi di formazione per essere utilizzati: da qui nasce l'esigenza di sviluppare le dinamiche di condivisione dei progetti ad ogni livello, a partire dalla mappatura dei bisogni reali degli utenti proseguendo

poi per tutte le fasi del processo di sviluppo del software, affinché non si creino prodotti senza alcuna attenzione per la funzionalità e per l'utilità.

Meno concorrenza inoltre significherebbe l'apertura di nuovi mercati sia per i consulenti autonomi, sia per le piccole società di assistenza e centri di formazione, lasciandogli così la possibilità di avere un proprio mercato.

Microsoft, ad esempio non riesce assolutamente a coprire da solo la massa di richieste di assistenza tecnica dei propri utenti/clienti in quanto essendo i codici chiusi, solo i programmatori Microsoft vi possono accedere, e essendo questi in netta minoranza rispetto agli utilizzatori dei software, Microsoft non riesce a garantire un efficiente servizio di supporto tecnico.

Di fatto esistono già da tempo diversi orientamenti all'Open Source, da parte di imprese che pur non contribuendo direttamente allo sviluppo dello standard, lo supportano attraverso altri modelli di business.

Affrontiamo meglio questo argomento nei prossimi paragrafi.

## **2 L'emergenza dei nuovi modelli di business Pure Player e Ibridi**

Il crescente sviluppo e diffusione dei prodotti Open Source, hanno spinto un numero sempre più elevato di imprese, a parteciparvi e ad adottare politiche di parziale accoglimento di questo nuovo modello di produzione del software.

Nel mercato europeo infatti si osserva una buona diffusione del software Open, soprattutto per quanto riguarda il lato del server delle reti: molte sono infatti le imprese che da tempo si stanno specializzando nelle soluzioni di ambiente Open come per esempio IBM, che ormai produce macchine completamente Linux compatibili.

La decisione di collaborare con la comunità Open è quindi motivata da un insieme di ragioni tecniche economiche e strategiche:

- l'esigenza da parte delle imprese di dare soluzioni alle crescenti richieste di migliore qualità, rendendo i propri prodotti compatibili con le soluzioni Open Source considerata l'ultima frontiera della qualità;
- la riduzione notevole dei costi di sviluppo, data la possibilità di trasferirli alla comunità volontaria;
- la possibilità di contrastare i concorrenti nel senso di favorire gli standard aperti contro gli standard proprietari della concorrenza.

A prima vista, sembra che esistano solo due estremi di approccio strategico al paradigma Open Source, rappresentati da un lato dalla strategia proprietaria e dall'altro da una strategia interamente aperta.

E' importante invece sottolineare che tra i due estremi troviamo diverse categorie di imprese che hanno sviluppato il loro business esclusivamente intorno alla produzione e alla distribuzione dei prodotti Open.

La struttura Open Source infatti, non è fatta solo di programmazioni tecnologiche, ma vi sono anche aziende che costruiscono il loro business sullo sviluppo e sulla distribuzione dei prodotti Open così dette Pure Player, ed altre aziende di programmazione così dette "Ibride" fatte di interfacce grafiche, redazioni di manuali tecnici e attività di supporto nei newsgroup, che focalizzano il loro business sui prodotti complementari.

Analizziamole meglio.

## 2a) I Pure Player

Le aziende Pure Player, sono imprese nate e cresciute in parallelo con la comunità e i prodotti Open, e si sono impegnate da sempre nello sviluppo e nella distribuzione del più importante prodotto Open Source, ovvero il sistema operativo Linux.

Sono aziende che sviluppano prodotti software in un contesto di copyleft e propongono un business, completamente focalizzato sulla distribuzione di prodotti open e dei servizi di contorno richiesti.

Esse sviluppano attività di alti valori creativi e di produzione artistica, e nonostante abbiano un basso contenuto innovativo, sono viceversa fondamentali ai fini dell'adozione del software da parte degli utenti; si pensi ad esempio alle due grandi interfacce utilizzate per Linux, KDE e GNOME: ideate nel 1998, hanno messo a disposizione degli utenti Linux l'ambiente desktop mouse driver che da un po' di tempo è diventato uno standard nel mondo commerciale, favorendo sia l'adozione del sistema operativo Torvalds da parte dei soggetti meno esperti, sia il passaggio ad esso da parte degli utenti Window e Mac.

Le imprese Pure Player hanno saputo modellare il proprio modello di business sul mercato dei prodotti Open, sapendo sostenere le proprie attività di ricerca e di sviluppo senza utilizzare procedure di protezione dei diritti di proprietà intellettuale sui prodotti software di loro interesse.

Ma a questo punto è ovvio domandarsi: “come si possono ottenere profitti dalla distribuzione di un prodotto gratuito o distribuibile a prezzi contenuti attraverso la rete Internet”?

Tutto ciò è possibile grazie alla creazione di un nuovo business sviluppato attorno al valore dei prodotti Open.

Infatti normalmente la commercializzazione di un prodotto software dovrebbe richiedere la protezione dei diritti d'autore sui prodotti distribuiti, ma il modello Open Source non permette invece la protezione di tali diritti.

Ci si può chiedere allora come possono le imprese Pure Player sostenere un modello di business basato quasi esclusivamente sul modello Open Source?

La risposta è rappresentata da imprese come Red Hat : il valore aggiunto che Red Hat commercializza è quello di confezionare il prodotto Linux venderlo come sicuro e affidabile grazie ad accurate verifiche di compatibilità e ai controlli dei codici, dimostrando così di saper generare profitti elevati anche solo basandosi sulla distribuzione, spostandosi così da un business di prodotto ad un business di servizio.

Inoltre il confezionamento e la distribuzione del prodotto Linux, il controllo dei codici e della loro compatibilità, aggiungono valore al prodotto e lo rendono più sicuro efficiente e completo, garantendo alle imprese Pure PLayer la sopravvivenza e la possibilità di continuare a sostenere lo sviluppo dei prodotti Open e del proprio business.

Solo una parte dei profitti proviene dalla vendita diretta dei pacchetti software, e il valore di questi pacchetti non è costituito da singoli prodotti distribuiti, ma dal contributo in termini di integrazione e confezionamento.



Il prezzo dei prodotti distribuiti da Red Hat risulta di conseguenza molto inferiore rispetto al prezzo di altri pacchetti software costruiti intorno a prodotti proprietari: Red Hat infatti ha la possibilità di condividere i costi di produzione del software con la comunità di sviluppo.

D'altro canto chiunque può scaricare copiare e distribuire gratuitamente i prodotti Red Hat coperti da licenze copyleft, per cui è chiaro che i profitti dell'azienda derivano da tutti i servizi aggiunti che l'impresa è in grado di offrire; inoltre la strategia dei Pure Player, è da sempre impegnate nella diffusione Open nelle vesti di un prodotto commerciale anche presso le pubbliche amministrazioni.

Questo dimostra come attraverso la commercializzazioni di questi valori aggiunti le aziende Pure Player non solo possono generare profitti garantendo la loro sopravvivenza, ma allo stesso tempo possono sostenere con continuità lo sviluppo e la diffusione di Linux.

## 2b) I modelli di business Ibridi

Questi nuovi modelli aziendali vengono denominati ibridi, perché si sviluppano attraverso una dettagliata regolazione, non priva di sviluppi giuridici innovativi, del rapporto tra prestazioni gratuite e prestazioni a pagamento: infatti mentre i Pure Player basano il loro business sulla commercializzazione di valori aggiunti, creati da loro stessi su un prodotto già esistente, i nuovi modelli aziendali ibridi, produttori di sistemi<sup>12</sup>, li distribuiscono gratuitamente e ricavano profitti facendo pagare costi di consulenza manutenzione e aggiornamento.

Questa strategia dei modelli di business ‘ibridi’ è perseguita da imprese che non possiedono uno standard proprietario che domina il mercato, quindi imprese che non hanno il problema della “cannibalizzazione” dei propri prodotti, anzi cercano di utilizzare l’Open per vendere prodotti complementari (hardware e software).

Un esempio di modello di business ibrido è il caso di IBM, che ormai da tempo supporta con convinzione il processo di diffusione del sistema operativo Linux, presentandosi come uno dei principali competitori dell’alleanza Wintel.

IBM, decisamente coinvolta nel settore e-business, ha intuito la possibilità di ricavare profitti, non più vendendo sistemi equipaggiati da software proprietari, ma offrendo servizi di consulenza assistenza e attività di

---

<sup>12</sup> Con il termine produttori di sistemi ci si riferisce a tutte le componenti principali di un’architettura, vale a dire hardware, software e servizi complementari

formazione richieste da aziende sempre più interessate a prodotti affidabili e a basso costo per la gestione delle infrastrutture informatiche.

E' ancora una volta il caso di Linux, che essendo una soluzione qualitativamente valida, viene adottata da imprese che cercano sempre più soluzioni migliori in termini di qualità ed affidabilità: in risposta ad una crescente richiesta di servizi e assistenza da parte delle aziende stesse, IBM ha sviluppato corsi di supporto tecnico e di formazione per Linux a pagamento, dimostrando così di poter generare profitti pur garantendo la diffusione gratuita del prodotto Linux.

IBM ha sviluppato una sensibilità verso i modelli di business più flessibili in grado di adattarsi alle esigenze del mercato, definendo così una strategia denominata 'cooperazione competitiva' da cui ne deriva che la competizione sui servizi è favorita dalla cooperazione sulle tecnologie e sulle infrastrutture riconoscendo il valore e le opportunità offerte dal modello Open Source per tale collaborazione competitiva.

L'iniziativa Open Shared Standard propone lo sviluppo di standard aperti accessibili a chiunque, giustificando così la necessità di sviluppare prodotti standard in grado di assicurare un mercato libero e competitivo per le future applicazioni di IBM.

Uno standard aperto aggiunge inoltre maggiori garanzie di flessibilità e qualità dei prodotti, e attraverso il modello Open Source diventa per le imprese un utile meccanismo per diffondere rapidamente tutte le nuove tecnologie, attivare collaborazioni e contributi dalla comunità, al fine di raggiungere più elevati livelli qualitativi di prodotto, per estendere i confini del proprio modello di business verso il mercato dei servizi e per

promuovere l'innovazione l'integrazione e lo sviluppo collaborativi degli standard stessi a vantaggio dell'intero settore.

Un altro esempio di azienda ibrida di produttori software, è Netscape il cui modello di business è principalmente concentrato nella produzione e diffusione dei software, rilasciando alcuni codici sorgenti di alcuni prodotti di loro proprietà.

Netscape si definisce ibrida in quanto solo parte della sua politica è influenzata dalla filosofia Open.

E' importante stabilizzare ed incrementare la sua quota di mercato per la collaborazione allo sviluppo di alcuni prodotti Open perchè l'aumento della quota di mercato, implica direttamente dei vantaggi di vendita riferiti ad altri prodotti e servizi le cui imprese mantengono il pieno controllo: anche Netscape sembra si sia avvicinata al modello di sviluppo Open per ragioni prettamente economiche.

Net infatti decise di rilasciare il proprio web browser alla comunità Open Source ed avviare il progetto Mozilla con l'obiettivo di contrastare la supremazia del prodotto Explorer di proprietà del concorrente Microsoft: intaccare il monopolio dell'avversario significava, per Netscape, garantire la salvezza delle proprie applicazioni e del proprio futuro.

Oltre il ruolo giocato nella diffusione, l'esistenza di modelli ibridi è importante per due motivi:

- da un lato tranquillizzano i potenziali adottanti del sistema, circa l'ampia disponibilità dei servizi complementari che sono recepiti come decisivi per l'affidabilità e l'effettiva usabilità dei programmi software;

- dall'altro forniscono incentivi monetari per lo sviluppo di modelli "ibridi", per le quali le motivazioni Open non sono sufficienti; senza contare che l'esistenza di utilizzatori di software Open è una garanzia della futura sopravvivenza del software stesso. Un fattore chiave per la decisione di adozione è la sicurezza che il software continuerà ad essere prodotto e sviluppato evitando così agli utenti cambiamenti di applicazioni e piattaforme che comportano alti costi di cambiamento in termini di infrastrutture e formazione del capitale umano.

Tra i vari argomenti di dibattito dei diversi forum di siti Open , oggi compare sempre più spesso quello sui soggetti 'ibridi'. Una parte della comunità non è d'accordo sull'esistenza di questi soggetti, perché sembrano andare contro quella che è il fulcro del movimento Open, ossia la filosofia intesa come etica Open;

altri, la maggior parte, considerano la presenza degli 'ibridi' fondamentale per l'esistenza futura del modello Open, perché pensano che solo appoggiandosi agli ibridi, si può raggiungere uno sviluppo futuro del software Open Source.

E' quindi normale pensare che dietro un qualsiasi modello di business, esistano diversi modelli di aziende; ed è altrettanto normale pensare che accanto ai diversi modelli di aziende, si sviluppino diverse tipologie strategie.

### **3. Altre strategie applicate al modello Open Source**

L'approccio delle imprese all' Open Source potrebbe essere molto diversificato: in questo panorama appena descritto, fatto di imprese di diversa natura e di diversi sviluppi di modelli di business è facile dedurre che esistano anche diverse tipologie di strategie.

Si passa infatti da strategie di difesa e attacco di Microsoft, che considera una minaccia tutti i prodotti in grado di porsi come valida alternativa ai propri, a strategie di supporto applicate per esempio da Sap, definita anch'essa un'azienda produttrice di software ibridi.

Le minacce considerate da Microsoft, possono essere :

- di profitto, se il prodotto concorrente è in grado di attirare le preferenze dei clienti incidendo così direttamente sul profitto dell'impresa;
- di piattaforma, quando esiste un' alternativa che è in grado di intaccare il monopolio di mercato di un'intera gamma di prodotti, ed è quella più rilevante perché comporta effetti a lungo termine sul business;
- di sviluppo, rappresentata dai prodotti in grado di attirare l'attenzione della comunità di sviluppo nel senso di sostenere prodotti alternativi provocando una serie di debolezze nei confronti dei prodotti concorrenti (in questo caso dei prodotti Open) che beneficino di più contributi e innovazioni.

La strategia applicata da SAP, invece è definita di "supporto":

SAP, è il leader mondiale per le produzioni per il business, con un' offerta completa di software e servizi in grado di soddisfare ogni necessità.

Definita anch' essa un'azienda produttrice di software ibridi, SAP offre alcuni suoi servizi in modalità 'Open 'decidendo di lasciare i codici sorgenti di alcuni prodotti per limitare i costi di sviluppo e quindi diminuire i prezzi dei propri servizi sottolineando il fatto che per molte imprese l'Open Source è una delle poche possibilità di contrastare i monopoli o uscire dai pericoli del mercato.

Un altro esempio di "strategia di supporto", potrebbe essere il caso di Compaq e Hewlett Packard , la cui politica è quella di integrare nei propri prodotti, sia software proprietario che software Open Source .

Queste due aziende sebbene simpatizzanti per il modello Open, hanno deciso di non schierarsi apertamente a favore di Linux o Windows, optando per la volontà di mantenere una leadership nella fornitura di Personal Computer indipendentemente dal sistema operativo prevalente.

L'attenzione verso prodotti Open Source, riflette la convinzione che le tecnologie 'open' siano in grado di favorire l'innovazione e la competizione.

In particolare Hewlett Packard dopo aver intuito che alcuni principi di ingegneria del software proposti dal modello di sviluppo Open possono risultare vincenti, e che il modello assume importanti aspetti soprattutto in ambito organizzativo, ha predisposto alcuni piani strutturati per una progressiva introduzione dei principi dell'Open Source nei modelli di sviluppo e nell'organizzazione interna all'azienda.

Si può parlare di strategie di supporto poiché le politiche di Compaq e Hewlett Packard vanno oltre la semplice osservazione e sfruttamento dei prodotti Open, ma bensì ne applicano alcuni principi, e ne supportano il sistema.

Altri tipi di strategie sono:

-Strategia di osservazione:

questa strategia è associata al processo di osservazione avviato da Intel , azienda produttrice di soluzioni hardware, nei confronti del modello Open Source.

A causa della propria alleanza con Microsoft, Intel è ancora legata ai prodotti proprietari, ma nonostante ciò ha dichiarato apertamente che il modello Open sembra molto più affidabile riguardo aspetti quale l'affidabilità e la sicurezza, ed è quindi in grado di migliorare e velocizzare la diffusione dei prodotti relativi alle attività di commercio elettronico.

Ottenere software con prestazioni migliori, significa riuscire a soddisfare meglio le esigenze della clientela, sia in termini economici, che in termini legati all'espansione del mercato aperto da Internet.

Gli interessi per i quali Intel strizza l'occhio al modello di sviluppo dei software Open Source, sono quindi di tipo economico/strategico, e non vanno assolutamente trascurati.

Per un miglioramento della qualità e della diffusione delle proprie soluzioni Intel ha avviato alcuni progetti riguardanti l'integrazione tra i prodotti Open Source (tra cui in particolare Linux) e le proprie soluzioni di hardware offerte.

Decidendo di non vincolare le proprie strategie alla sola alleanza Wintel, Intel si è potuto allontanare dall'immagine negativa che persegue da tempo i prodotti Microsoft



Le strategie di Intel Compaq e Hewelett Packard possono essere definite anche strategie di attesa.

Infatti pur interessandosi ai prodotti Open, queste imprese non hanno ancora apertamente dichiarato la propria posizione, rimanendo appunto in attesa dei futuri sviluppi di mercato, a differenza di alcune imprese attive prevalentemente sui mercati dei sistemi, come ad esempio Sun, o attive sui mercati del software come ad esempio Netscape o Sap che sembrano essere maggiormente convinte.

## 4. La concorrenza nel mondo Open Source

Se inizialmente quando si pensava al software libero, si pensava ad un business prettamente etico e lontano dalla rincorsa ai profitti e dalla concorrenza, gli ultimi episodi dell'Open Source sembrerebbero dimostrare l'esatto contrario: anche le imprese Pure Player non stanno certo a guardare. Nell'ottobre del 2004 Red Hat ha acquistato da American On Line alcuni asset di Netscape Security Solution.

Il valore dell'operazione non è stato svelato, ma Red Hat ha detto di aver speso almeno 25 milioni di dollari.

La cosa interessante è che Red Hat oltre ad aver acquistato un prodotto ha anche acquistato un metodo: infatti alle critiche di Sun, sul fatto di aver acquistato un prodotto considerato obsoleto, Red Hat risponde di cercare di migliorarlo ed aggiornarlo, grazie all'aiuto della comunità di Linux di un piccolo team di sviluppatori ereditato da Netscape.

Con questo acquisto Red Hat intende ampliare la propria piattaforma enterprise e competere più da vicino con le soluzioni di directory e di gestione proposte da Novell e Sun, su piattaforma Unix e Linux.

L'idea di aggiornare il prodotto attraverso la comunità sottolinea come l'acquisto di Red Hat, non sia stato solo un acquisto di prodotto, ma bensì un acquisto di metodo: infatti, grazie alla metodologia Open fatta di cooperazione e collaborazione, Red Hat cercherà di rendere il prodotto il più

aggiornato possibile e in grado di competere con le altre imprese tipo Novell e Sun.<sup>13</sup>

Non mancano poi atteggiamenti curiosi, di chi da poco ha scoperto l'esistenza del nuovo paradigma Open Source: ormai se ne sono accorti tutti, "Linux è entrato nel salotto buono".

Così ha titolato un noto quotidiano italiano per sottolineare la rilevanza economico/strategica che sta assumendo il nuovo paradigma nel mondo degli affari: questo infatti si rivolge al mondo Open per rigenerarsi, infatti molti imprenditori, offrono finanziamenti per scoprire imperfezioni dei software Open Source con lo scopo di crearne nuovi e rimpiazzarli attraverso modalità proprietarie.

Il primo a farlo è stato Mark Shuttleworth, imprenditore americano, che ha messo a disposizione i primi 5000 dollari per chi scovava difetti di Mozilla, in modo da poter analizzare rimpiazzare e concorrere col prodotto Mozilla, creandone uno simile, e senza difetti.

Inoltre i sistemi di infiltrazione e finanziamento più o meno nascosti, effettuati dagli investitori in un settore sempre più regolato da licenze, sono sempre più chiari; basta osservare la gestione gerarchica delle comunità, la concorrenza: il panorama dell'Open Source si affaccia sempre più al mondo dei "grandi", scopre il marketing, o meglio è scoperto dal marketing, e si appiattisce, obbedendo a quelle che sono le regole del mercato.

---

<sup>13</sup> vedi [//punto-informatico.it/p.asp?i=49824](http://punto-informatico.it/p.asp?i=49824)

## 5. Le Gare a progetto e la portabilità dei software

Fra le novità più curiose che hanno caratterizzato il mondo Open troviamo le così dette “gare” a progetto. Si tratta di vere e proprie sfide di scrittura di intere parti di codice con premi economici per i vincitori: chi prima consegna parti di codice o chi prima trova e corregge i difetti dei software, riceve un premio che va tra i 200 e i 1200 dollari.

L'esempio più eclatante di applicazione di questo tipo di strategia è stato sviluppato dalle comunità MONO e DotGNU: esse sviluppano la medesima struttura su piattaforme GNU/Linux con l'idea di far girare applicazioni Microsoft e in genere per Windows anche su sistemi operativi liberi.

Avere una solida base digitale per programmare con un linguaggio intermedio che si posiziona tra il linguaggio macchina e il linguaggio scelto dal programmatore, rende ogni programma teoricamente “portabile” su qualsiasi sistema operativo dotato di tale struttura (o framework): in altre parole il programmatore potrà decidere il suo linguaggio, il compilatore lo trasforma in linguaggio intermedio ed il framework lo farà girare su ogni architettura.

La “portabilità”, che quindi consiste nella capacità di combinazione con numerosi sistemi di hardware e software, è una caratteristica rilevante per le imprese, che consente di aggiornare o sostituire le infrastrutture hardware mantenendo inalterati i software; inoltre la disponibilità dei codici e la

libertà di modificarli, permette alla comunità di migliorare continuamente la portabilità del prodotto.

Progettare questi nuovi compilatori, significa modificare notevolmente progetti già da tempo affermati e per il mondo Open Source, significa la scrittura da zero di nuovi compilatori per linguaggi che sono di proprietà di Windows.

Entrambi i gruppi si sono confrontati con la difficoltà dei lavori intrapresi ed hanno optato per una scelta simile: pagare chi consegnava per primo una delle parti più noiose da sviluppare.

Le gare indette anche da DotGNU, dimostrano come si cerca di andare sempre più verso un non equilibrio fatto di prestazioni seriali e di competenze a fronte della ricerca di una vera e propria identità. Dilatare il tempo dell'emergenza, continuando ad insistere sulla necessità di risultati, subito adesso e sempre, significa mutare una metodologia propria della produttività capitalista; ma dilatare il tempo ' dell'urgenza perenne ', ossia centrare gli obiettivi ad ogni costo, corrisponde ad una realtà prettamente capitalista in cui il desiderio non ha più spazio.

Il potere tecnico dell'individuo, inteso come semplice atomo e non parte di una comunità, viene ceduto in cambio di un riconoscimento troppo ristretto, ridotto e non condiviso senza adesione ad un piacere più ampio: il desiderio deve allora essere sostituito con il denaro; questa è la nuova identità scoperta. Scambio di know how tecnico per alimentare una dipendenza che consiste nello stare in corsa a qualsiasi prezzo: sopravvivenza immediata, non più hacking creativo.

A questo punto il ritorno nella realtà è completamente privo di stile; i piani vengono schiacciati banalizzati e ridotti ad un puro semplice scambio di denaro e potere, aprendo così uno scenario in cui si apre un confronto ed una lotta sui rapporti di forza nel lavoro telematico e sui diritti digitali provenienti dalla vittoria di fatto della GPL.

E' importante ora prendere atto del fatto che l'etica di un software non sta solo nella licenza, ma soprattutto nell'interazione tra le modalità di distribuzione e la metodologia di sviluppo, poiché il sistema è diventato più complesso: da una parte i programmatori dovranno fare i conti con il dilagante vortice di precarietà in cui potrebbero cadere se il loro lavoro fosse isolato da un confronto critico; dall'altra i futuri consumatori/utenti dei prodotti Open Source dovranno affinare le proprie capacità 'autoformative' al di là delle vesti user-friendly.

Più in generale è necessario rendere pubblici i modelli di organizzativi della propria equipe di lavoro in un'ottica di apertura e di miglioramento collettivo definendo questa pratica di capitale importanza per la comprensione delle dinamiche in atto.

## 6. L' Autoformazione

Un' altra caratteristica concorrenziale del paradigma Open Source è la così detta "Autoformazione".

La formazione attuale cerca di istruire senza però mettere in discussione origini e metodologie di apprendimento; è molto concentrata sull'esito, poco sui metodi e processi.

Per quanto riguarda il modello Open Source invece il processo di formazione è un po' diverso da quello dettato dall'elite tecnoburocratica; i momenti sono la riappropriazione, rispecchiamento, inteso come modifica e personalizzazione, e rappresentazione, ed è un processo che crea un equilibrio dinamico fra i singoli individui e le comunità a cui appartengono.

La differenza fondamentale tra le due tipologie di formazione sta nella scelta: per spiegare meglio restringiamo momentaneamente la prospettiva di analisi al coder puro, cioè colui che scrive i codici.

Il coder decide di prendere parte ad un progetto o di iniziarne uno nuovo, perché spinto dal proprio gusto personale, da un desiderio di creazione nel quale investe il proprio tempo e la propria intelligenza. La prima fase consiste nel riuscir a "metterci le mani sopra" (detta alla hacker) , sviluppando un processo di riappropriazione smontaggio e comprensione dell'oggetto.

L'individuo in questo caso non è una tabula rasa da formare dal principio, ma è un individuo già formato in parte e che si autoformerà ancora di più

sperimentando ed ampliando in continuazione la ‘sua cassetta degli attrezzi’ negando così l’ipotesi che il coder non venga profondamente influenzato dallo stile di altri coder nel processo di ‘autoformazione’; nella seconda fase lo stile personale che il coder sviluppa e che rispecchia la sua personalità, trova posto in un contesto di condivisione, influenzando a sua volta lo stile di altri coder. I codici rappresentano la personalità di chi li crea, e informano di se, tutti i soggetti delle comunità che semplicemente lo utilizzano; da questo si deduce l’importanza di far parte di una comunità nel senso di sentirsi rappresentato da quella comunità, e non certo perché si ha diritto di veto o potere di voto, ma perché si è in grado di influenzare la comunità stessa attraverso le proprie creazioni e anche lasciarsi influenzare; si cambia e si inducono cambiamenti;

Autoformazione significa farsi individui nel farsi comunità.

Il processo di formazione deve necessariamente coinvolgere l’individuo e la comunità, avere il coraggio di trovare nuove vie di fuga, crescendo e modificando, che non siano totalmente flessibili all’inflessibilità del mercato, ma processi di riappropriazione, rispecchiamento e rappresentazione di se in un contesto comunitario.

Nel momento in cui la comunità, per una serie di ragioni, non riesce più a riflettere la propria individualità, le possibilità sono molte: creare una nuova comunità (forking)<sup>14</sup> o l’introduzione di un nuovo software che ci rappresenti di più. Si tratta comunque di un equilibrio dinamico e di una rinegoziazione continua della propria individualità e della conformazione della comunità

---

<sup>14</sup> Il ‘forking’ si verifica quando una parte di comunità impegnata in un progetto già avviato si distacca dal nucleo dei programmatori per creare una versione diversa del software



nel tempo: non si possono infatti immaginare programmatori di una comunità di sviluppo completamente statici che aderiscono totalmente ad un manifesto o ad una dichiarazione di intenti, se per comunità di sviluppo si intende una realtà in continuo sviluppo.

## 7.Lo stile

Un'altra variabile competitiva che caratterizza il modello di sviluppo Open è "lo stile": immaginare realtà di autoformazione significa dare atto all'esistenza di scarti rispetto ad una presunta norma.

In ambito software il pensiero dell'autoformazione, va all'organizzazione di corsi a diversi livelli, alla scrittura di manuali, alle creazioni comunitarie, a possibili laboratori ed esperimenti per risolvere problemi pratici o anche solo per giocare con una nuova tecnologia.

L'idea che attraverso i metodi di condivisione di informazioni si possa aumentare il livello complessivo di competenze, e soprattutto di piacere che determinate attività creano, si basa sulla convinzione che le conoscenze debbano essere sempre più diffuse e contaminate.

Se una conoscenza non è comunicabile, diventa solo un bel gioco solitario, o nella peggiore delle ipotesi, diventa uno strumento di potere potenzialmente pericoloso e oppressivo: la condivisione è la miglior garanzia che abbiamo.

Dato che non tutto si può comunicare con un codice, ma esistono differenti codici in base al target a cui si vuole comunicare, è importante il "come" si comunica almeno quanto il "cosa" si comunica.

Questo "come" cioè il metodo con cui si comunica è "lo stile", ed è una discriminante essenziale nell'utilizzo di un codice: dal punto di vista dell'utente, che situato al di fuori dell'ambito tecnico informatico diventa il

ricevente del messaggio, lo stile è il metodo con cui il messaggio gli viene passato.

Si deve a Linus Torvalds la dimostrazione che lo stile “bazar” funzionava, e funzionava anche bene.

Nella realtà capita spesso che l’utente risponde al messaggio (cioè al codice): si tratta di un noto meccanismo per cui l’utente (consumer), specie se evoluto, avanza richieste di migliorie, diventando così anche produttore cioè prosumer (ovvero colui che inizia un nuovo progetto); è interessante notare che più lo stile cerca di facilitare l’utente, rendendo facile un feedback, più si creano collaborazioni volte a migliorare il codice stesso.

Inoltre possiamo analizzare lo stile secondo due aspetti: il primo definisce i canoni di una interazione utente-software, rappresentato per esempio da un’interfaccia grafica o con differenti approcci, in base al tipo di target da raggiungere;

il secondo riguarda il rapporto tra coder e software: il suo stile si manifesterà attraverso la struttura della routine, nella qualità dei commenti e nella pulizia complessiva del codice.

Si può concludere, quindi, che lo stile risponda al complesso dei propri immaginari ed al metodo di realizzazione concreta del desiderio di creazione di un oggetto.

## 8. Conclusioni

Nel settore informatico sono sempre più ricercati caratteristiche quali l'autonomia la flessibilità e l'indipendenza; si producono così effetti interessanti e di cambiamento nei rapporti di lavoro tra singoli e imprese ed anche sulle organizzazioni di lavoro: si abbandona la struttura di lavoro rigida fin'ora utilizzata per passare ad una più elastica e flessibile, ma anche più insicura.

Per limitare questa insicurezza occorre pensare a nuove forme di appartenenza e identità, dando più importanza a nuove forme di collaborazione date da associazioni, comunità professionali, network di relazioni e altre forme di aggregazioni in grado di esercitare un ruolo di interazione ed offrire supporto in senso di sicurezza ed anche identità sociale.

Ed è proprio qui che si inserisce a pieno titolo il modello Open Source, sottolineando come il nuovo modello aperto e quello chiuso siano destinati a convivere.

# CAPITOLO 4

## L'Open Source nella pubblica amministrazione

### 1. Coinvolgimento della Pubblica Amministrazione

Negli ultimi anni il nuovo paradigma Open Source si sta sviluppando diffondendo e affermando sempre più richiamando l'attenzione strategica ed economica di molte grandi imprese che ne riconoscono e ne evidenziano la potenzialità.

Il software Open Source è diventato un tema importante nelle discussioni accademiche e, in misura sempre più crescente, nel dibattito politico: da fenomeno limitato ad una stretta cerchia di appassionati, oggi rappresenta una leva strategica per la gestione di sistemi informativi di imprese, fino ad entrare nelle politiche commerciali di colossi della tecnologia informatica come Sun e Ibm.

Sono tre i principali fattori che in questi ultimi tempi stanno valorizzando il software Open Source:

- il successo nel mercato delle infrastrutture informatiche basate su questo modello (Linux ed Apache ne sono un esempio),
- una diffusa insofferenza nei confronti del monopolio statunitense;
- la sempre più crescente presa di coscienza che soluzioni chiuse non rappresentano le migliori soluzioni atte a soddisfare la complessa domanda di applicativi di cui si necessita.

L' affermazione del nuovo modello ha prodotto anche un allargamento delle tipologie dei soggetti coinvolti: di fatto dopo le imprese anche i governi e le pubbliche amministrazioni si stanno avvicinando ai prodotti ed al modello Open Source.

Sono ormai numerosi i casi nazionali ed internazionali di enti e Pubbliche Amministrazioni, che hanno adottato software Open Source; cerchiamo ora di capirne meglio le motivazioni, le problematiche e soprattutto i vantaggi.

La propensione e la politica delle varie amministrazioni pubbliche, in particolar modo quelle europee, sull'adozione del OSS (Open Source Software), sono molto diverse: Francia e Germania hanno sposato in toto la filosofia, la gran Bretagna invece, più pragmatica, si avvicina a piccoli passi; anche l'Italia si sta muovendo in questa direzione, ne sono esempio alcune realtà provinciali come Pisa e Udine che hanno avviato progetti di notevole interesse e lo è anche la direttiva del ministro Stanca che tratta lo sviluppo dei programmi informatici delle pubbliche amministrazione e da particolare importanza all'utilizzo di software Open Source.

Ma analizziamo meglio le motivazioni che spingono la PA all'adozione di soluzioni software Open Source.

## **2.Motivazioni all'adozione dell' Open Source nella Pubblica Amministrazione**

La pubblica amministrazione è un importante attore nel mercato del software, e per capire realmente la portata del fenomeno è interessante individuare l'importanza del soggetto pubblico nel settore software e conoscere l'ammontare della spesa sostenuta dalla Pubblica Amministrazione stessa.

Il soggetto pubblico infatti può influenzare il mercato da due punti di vista:

-da un lato la domanda di software provenienti dalla PA è molto importante per l'impatto sulla spesa pubblica, in quanto le Pubbliche Amministrazioni sono costrette ad acquistare parecchio software per gestire la burocrazia e la maggior sicurezza;

-inoltre la PA sembra favorire soluzioni personalizzate e configurate ad hoc.

Da entrambi i punti di vista l'Open Source può essere considerata una vera e propria soluzione attraente:da un lato infatti senza dubbio l'utilizzo di OS porta ad un risparmio iniziale in termini di costi per le licenze, e dall'altro il software Open Source è più adatto ad essere personalizzato rispetto ad uno proprietario.

Le motivazioni ed i benefici che spingono le Pubbliche Amministrazioni ad un possibile trasferimento tecnologico verso le soluzioni Open Source, sono del tutto simili a quelle che hanno influenzato le imprese private, ma vi sono

due elementi in più che caratterizzano le scelte delle PA circa il passaggio al nuovo paradigma:

- **l'indipendenza dai fornitori** è un elemento che valorizza la scelta di utilizzare software Open Source, ed è una delle ragioni che ha spinto molte Pubbliche Amministrazioni a migrare dal software proprietario al free software, permettendo alla Pubblica Amministrazione stessa di svincolarsi dalle scelte tecnologiche delle singole imprese, che spesso sono dettate più da esigenze competitive che dalla ricerca di una reale efficienza e qualità dei prodotti, dando inoltre la possibilità al soggetto pubblico di risparmiare sui servizi di supporto, assistenza e manutenzione, poiché si verrà a creare una maggior concorrenza tra diversi operatori in grado di offrire servizi sulle piattaforme e sui prodotti standard;

**affidabilità e sicurezza:** la PA ha il compito di garantire la riservatezza dei dati raccolti e organizzati nei propri database, e avere a disposizione i codici sorgente dei programmi utilizzati all'interno della propria organizzazione, da la possibilità alla PA di dare un grado di maggiore sicurezza ai cittadini: il software Open Source, infatti, rispetto a quello proprietario, consente ad un numero maggiore di specialisti del sistema informativo delle pubbliche amministrazioni, di ricercare errori ed imperfezioni potenzialmente dannose o illecite che permettono l'accesso non consentito a informazioni di particolare importanza, garantendo così più sicurezza e affidabilità alla comunità che in questo caso coincide con la cittadinanza pubblica.

Se da una parte l'adozione dell' OS offre alle pubbliche amministrazioni libertà, personalizzazione, sicurezza e permette di svilupparne



l'informatizzazione, dall'altra ci sono alcuni fattori critici che possono frenarne la diffusione: difficoltà nella migrazione da un software proprietario conosciuto ed utilizzato dalla gran parte del personale, investimenti effettuati in passato su software proprietario e non ancora ammortizzati, scarsa diffusione di competenze specifiche che sfruttino al meglio le possibilità e le potenzialità di software Open Source: tutte criticità di non semplice ed immediata soluzione.

Nonostante le difficoltà, l'adozione di software OS da parte della pubblica amministrazione è una realtà, e numerosi casi che verranno trattati in seguito, lo dimostreranno.

Le pubbliche amministrazioni che vogliono fare questo salto devono prendere una decisione esente da pregiudizi e che non si limiti a considerare questa scelta solo come una banale riduzione dei costi, ma bensì un progetto rivoluzionario capace di creare valore economico e sociale. La possibilità di accedere al codice sorgente e la presenza di una comunità di sviluppatori che mettono a disposizione le proprie competenze, può rappresentare, soprattutto per la PA dei paesi in via di sviluppo, un importante strumento per il progresso e la formazione di competenze informatiche.

Ma nel caso del soggetto pubblico, l'aspetto economico non è l'unico criterio di valutazione per l'adozione del software Open Source: infatti il ruolo della Pubblica Amministrazione, l'importanza delle attività e delle informazioni da essa gestite, i problemi relativi all'accessibilità ed alla sicurezza delle informazioni e alla garanzia della comunità di gestione, inducono a considerare altri criteri di valutazione: uno dei ruoli fondamentali della Pubblica Amministrazione è quello di gestire e

diffondere le informazioni riguardanti i cittadini, lasciando ad essi il diritto di accedere facilmente ai propri dati e garantirgli inoltre l'accessibilità, l'integrità e la riservatezza di queste informazioni, senza però costringerli ad utilizzare uno specifico software. Utilizzando quindi standard aperti e ampliando le possibilità di scelta la PA riesce a garantire più facilmente la continuità di gestione ed il mantenimento delle informazioni, indipendentemente dalle soluzioni scelte.

In altre parole la logica Open Source deve essere interpretata come un'opportunità per promuovere e diffondere l'innovazione e per produrre un reale cambiamento anche nel settore pubblico, diventando così uno stimolo per incentivare investimenti ed iniziative: in quest'ottica il soggetto pubblico si potrebbe inserire, sostenendo parte di questi investimenti.

### **3.Criteri di valutazione del ruolo della Pubblica Amministrazione**

Come descritto in precedenza l'adozione di prodotti software Open da parte della PA potrebbe avere effetti rilevanti nel mercato del software stesso: la PA potrebbe infatti favorire una soluzione invece di un'altra (nel caso specifico la soluzione Open rispetto a quella proprietaria), e influenzarne lo sviluppo.

A questo proposito bisogna osservare che le due alternative potrebbero essere oggetto di selezione da parte del mercato, se non esistono significativi "fallimenti di mercato".

I "fallimenti di mercato", si verificano quando si affermano prodotti che non hanno un valore soddisfacente per gli utenti; cioè il fallimento è da intendersi come un mal funzionamento del mercato tale da non poter garantire l'affermazione della miglior soluzione, e in tal caso l'errata selezione del mercato potrebbe giustificare un intervento da parte del soggetto pubblico che influenzerà la diffusione e la crescita della miglior soluzione.

Ci si deve ora chiedere se tale intervento pubblico sia giustificato nel settore software; a tal proposito esistono diversi punti di vista tra cui alcune considerazioni di alcuni studiosi : James Bessen, un attento studioso del nuovo paradigma, nell'opera "What Good is Free Software?" non considera l'Open Source come un'alternativa al software tradizionale, ma lo vede

come un completamento della qualità e delle caratteristiche ottenibili con il software proprietario: non rileva alcun “fallimento di mercato”, ma considera l’Open Source, come un’opportunità per superare alcune imperfezioni che accompagnano il software proprietario: cioè egli considera opportuna una convivenza tra i due modelli senza dover necessariamente scegliere a priori l’uno o l’altro, e cerca di spingere il soggetto pubblico verso un’attenta comparazione basata sui costi e benefici.

In sostanza Bessen non ritiene necessario nessun tipo di supporto economico allo sviluppo dell’Open Source da parte della PA, in quanto la sua realizzazione e diffusione si sono sviluppate fino ad oggi senza intervento pubblico; considera invece un intervento opportuno e necessario circa la possibilità di limitare gli effetti negativi delle attuali leggi in materia di brevetti e di diritti di proprietà.

Anche Bradford Smith nell’opera “The Future of Software” non intravede fallimenti di mercato e fa notare come le iniziative di alcune grandi aziende evidenziano una fruttuosa integrazione tra le due tipologie di software.

In altre parole, sono d’accordo sulla non opportunità di un intervento pubblico a sostegno del software Open Source, in quanto il software proprietario e quello Open sono ritenuti ugualmente importanti nel settore software e la presenza di entrambe le soluzioni è indispensabile e complementare.

Solo Lawrence Lessing, docente presso la facoltà giuridica dell’Università di Stanford (California), nell’opera “Open Source Baselines: Compared to

What?”, si discosta da questa visione: infatti ritiene che la PA debba assumere una posizione più decisa soprattutto per quanto riguarda lo sviluppo e la diffusione di standard aperti: Lessing infatti riconosce ed identifica l’esistenza di un “fallimento di mercato” relativo all’Open Source, in quanto egli sottolinea che possa esistere la possibilità che i programmatori siano disincentivati allo sviluppo del software Open, dall’impossibilità di ottenere un compenso economico del loro lavoro di sviluppo, e ciò potrebbe causare una diminuzione di interesse per i prodotti Open e una riduzione delle performance dei prodotti stessi, spostando così l’attenzione e quindi le preferenze verso soluzioni proprietarie.

Per quanto riguarda invece l’atteggiamento che dovrebbe mantenere la PA nei confronti dell’Open Source, Bessen e Smith sostengono che la scelta dovrebbe basarsi su valutazioni tipicamente economiche secondo gli stessi criteri utilizzati dalle aziende.

Soltanto Lessing si discosta ancora una volta da questa visione e sottolinea che un altro carattere importante da considerare, oltre a quello economico è il carattere sociale: lo sviluppo del software Open Source infatti ha rilevanza sociale, ed è proprio questo che ha permesso lo sviluppo del nuovo modello grazie al lavoro di cooperazione di una comunità volontaria, fatta di soggetti eterogenei, volontari, disposti a collaborare allo sviluppo di un prodotto che soddisfi le esigenze di tutti.

Si parla di rilevanza sociale perché questo nuovo modo di produrre, più libero e slegato dalle rigide strutture aziendali, influisce molto sullo spirito e sulle motivazioni che spingono i programmatori ad aderire a questo nuovo modello di produzione, molto più coinvolgente dal punto di vista sociale e

quindi tecnico, e proprio per questo il suo utilizzo dovrebbe essere guidato anche da considerazioni di tale natura.



## **4.L'Open Source fa breccia nella Pubblica Amministrazione europea**

Il crescente interesse da parte delle Pubbliche Amministrazioni verso l'Open Source, è segnato dall'avvio di alcuni progetti e dalla formulazione di alcune proposte di introduzione di software liberi, da parte di alcuni paesi europei: per questo la commissione europea ha formulato un piano di azioni per la creazione di una società per l'informazione per tutti.

È interessante lo studio fatto dalla commissione europea a riguardo, attraverso lo sviluppo di diverse proposte :una di queste è l'IDA (Interchange of Data between Administrations) project, che rappresenta in linea generale l'utilizzo di Software Open Source nella PA.

La guida dell'IDA, redatta grazie alla consulenza di esperti del settore pubblico europei, fornisce tutta una serie di suggerimenti, volti ad aiutare gli amministratori a decidere non solo se e in che ambiti passare al software Open Source, ma anche quali prodotti scegliere e quali passi seguire durante la migrazione: essa per esempio mette in evidenza come risulti vincente l'accoppiata Apache/Linux per quanto riguarda il lato server; un esempio

dell'utilizzo di IDA che analizzeremo meglio nel corso del paragrafo è fornito dal Ministero delle Finanze danese;

o ancora un'altra delle proposte di questo piano di azione da parte della comunità europea, si chiama e-Europe, e riguarda l'intenzione di sostenere attività di standard aperti e la possibilità di adottare software libero per la PA.

Per valutare il grado di accoglimento dell'Open da parte del soggetto pubblico, bisogna analizzare due criteri:

il primo consiste nell'impegno da parte di un paese nell'emanare norme direttive o raccomandazioni in grado di regolare il nuovo paradigma;

il secondo tiene conto del numero effettivo dei progetti e delle iniziative già realizzate, dei progetti in corso di esecuzione e dell'utilizzo effettivo dei prodotti Open Source.

Dall'analisi di questi due criteri, evince il fatto che la situazione a livello europeo è abbastanza differente: da una parte infatti troviamo Francia e Germania che da molto tempo sono impegnati nell'utilizzo dei prodotti Open Source nella Pubblica Amministrazione;

dall'altra parte invece troviamo paesi, come per esempio l'Italia in cui il fenomeno Open Source è ancora oggetto di studio.

Ma analizziamo meglio alcuni eventi che hanno segnato il percorso di avvicinamento all'Open Source in vari paesi:

nel Regno Unito, il governo britannico considera il software Open, come un'opportunità per evitare il problema del lock-in delle soluzioni proprietarie.



In prospettiva dell'utilizzo di software OS nella PA il Governo ha avviato uno studio per analizzarne l'introduzione, attraverso le seguenti linee guida:

- effettuare un obiettivo confronto tra le soluzioni Open e le soluzioni proprietarie, nella prospettiva di value for money;
- favorire l'interoperabilità dei prodotti e delle soluzioni utilizzando formati Open Standard;
- valuta l'applicabilità del modello OS per la distribuzione dei risultati dei programmi pubblici di ricerca e sviluppo.

Interessante inoltre è il caso della Francia: il Governo utilizza soluzioni Open Source nei settori di interesse nazionale, quale la sanità l'ambiente i trasporti, tecnologie per i disabili e formazione on-line a livello di struttura informatica, con lo scopo di rafforzare migliorare e garantire l'interoperabilità tecnologica delle funzioni Pubbliche.

La Francia inoltre ha compiuto il suo primo concreto passo quando il Ministero dei Trasporti francese, guidato da Gilles de Robien, ha rimpiazzato su 1.500 server - localizzati all'interno di vari uffici pubblici e ministeriali - il sistema operativo Windows NT con la Mandrakelinux Corporate, la versione server della famosa distribuzione Linux della società parigina.

Oltre al sistema operativo, Mandrakesoft fornirà al Governo anche tutti i relativi servizi di installazione, formazione e supporto, in questo aiutata dall'acquisizione recente della società di servizi per Linux Edge-IT.

Inoltre il Ministero dei Trasporti francese, che si occupa anche della gestione del territorio e del turismo, amministra un'infrastruttura informatica composta da oltre 60.000 programmatori e circa 2.000 server con Windows NT: per il momento non c'è nessun piano circa la migrazione di questa infrastruttura verso Linux, tuttavia il ministero conta di rimpiazzare tutti gli attuali server Windows entro la fine del prossimo anno.

Questo progetto su larga scala è in accordo con l'intento del Governo francese di promuovere le tecnologie basate su standard pubblici, di aprire il mercato pubblico ad una maggiore competizione e di ridurre i costi della Information Technology: questo si legge in un comunicato diffuso da Mandrakesoft.

La piattaforma delineata dal Governo francese è stata confermata di recente dal ministro per la funzione pubblica, Renaud Dutreil, il quale ha svelato che la Francia intende portare il software open source su almeno un milione di computer della pubblica amministrazione.

Per attuare questo piano, il governo di Parigi ha finanziato la creazione di una nuova licenza aperta, chiamata CeCILL, compatibile con la GNU GPL: la licenza, redatta da tre istituti di ricerca francesi, ha l'obiettivo di adattare i concetti della GPL alla realtà legislativa locale, soprattutto in campi - come quello del copyright e delle garanzie sui prodotti - dove le leggi americane differiscono sostanzialmente da quelle francesi ed europee. La licenza, prevede espressamente la possibilità che il codice possa essere integrato in software GPL e ripubblicato sotto tale licenza.

.Un'altra notizia interessante è quella della recente collaborazione avviata tra Monaco e la vicina Vienna. Dato che anche la capitale austriaca ha avviato il

salto (quantomeno parziale) verso il pinguino, le due città hanno deciso di cooperare su varie questioni inerenti il software open source. In particolare, hanno spiegato i rispettivi responsabili dell'ambito info-tech, che si tratterà di sviluppare insieme un "public authority desktop" ovvero un group-ware, cioè un software di supporto al lavoro di gruppo, per facilitare la comunicazione e il coordinamento tra diversi gruppi di utenti. Si punterà cioè a sfruttare quelle aree in cui il lavoro cooperativo rende meglio al fine di semplificare lo scambio delle reciproche esperienze sul campo. Va ricordato che Monaco prevede di completare il passaggio all'Open Source di tutte le 14.000 macchine desktop entro il 2009, mentre per Vienna si tratta di una migrazione più ridotta. Entro il 2005 si prevede di offrire il salto da Windows a Linux per circa 7.500 utenti sui 16.000 totali, con annesso studio valutativo a un anno di distanza per eventuali ampliamenti del programma. Intanto anche in Danimarca il Ministro delle Finanze ha virato verso l'Open Source per semplificare lo scambio dati tra sistemi differenti. Anziché usare soluzioni Microsoft la scelta è caduta sull'applicazione JBoss che gira su Red Hat Linux. Come spiega un rapporto della Computer Science Corporation, in tal modo si riusciranno a trasmettere 1,5 megabit di dati al secondo tra circa 400 istituzioni pubbliche e Ministeri, e il motivo principale della scelta, ha spiegato Peter Henningsen, manager del progetto, è stato il costo: il budget stanziato era di soli due milioni di corone danesi. Importanti però anche le questioni tecniche: Henningsen ha aggiunto che JBoss è una piattaforma assai stabile, punto cruciale per l'operatività complessiva, e in tal modo prevede di superare i problemi avuti con le precedenti impostazioni: le prime fasi dell'implementazione di JBoss non

hanno creato alcun problema; elemento non da poco se si considera che l'affidabilità dello scambio dati è essenziale visto che il sistema viene usato da numerosi dicasteri danesi, dalla Difesa alla Giustizia agli Affari Pubblici, per inviare ogni genere di dati a quello delle Finanze. Il sistema serve anche per organizzare gli stipendi, il bilancio e la gestione finanziaria interna. Dopo due anni di preparativi, il progetto è partito il mese scorso: i dati provenienti da varie fonti vengono convalidati usando gli strumenti di Apache Xerces, poi il server JBoss-Red Hat si occupa dello scambio e delle transazioni relative necessarie :quest' ultima nota ricorda come l'operazione si pone perfettamente in linea con la strategia generale del governo danese di usare, per quanto e fin dove possibile, gli standard aperti.



## **5. Il caso più eclatante: Monaco e il progetto LiMux**

Monaco è forse il caso più eclatante di Pubblica Amministrazione votata all'Open Source. Non solo i server, ma anche i Desktop verranno migrati verso tecnologie aperte. I numeri che accompagnano questa transizione sono da capogiro: 14000 computer Desktop e 16000 portatili verso Linux, un piano pluriennale e uno stanziamento di ben 35,7 milioni di euro.

Nel maggio 2003 il comune di Monaco ha deliberato la completa transizione verso tecnologie Open Source di sistemi server e sistemi desktop. Il progetto, chiamato LiMux (un gioco di parole tra Munich e Linux), era iniziato in realtà due anni prima con la richiesta del consiglio della città di risolvere il noto problema della dipendenza della P.A. da un solo fornitore. Nel 2002 e nel 2003 sono stati quindi svolti studi di fattibilità sulla transizione verso tecnologie Open Source con la collaborazione di IBM e SuSE.

Quando sembrava ormai che il progetto fosse ben avviato, il comune ha dovuto sospendere il tutto per via della contestatissima direttiva sulla brevettabilità del software; direttiva che dovrebbe portare in Europa la possibilità di brevettare parti di codice. Uno studio dell'Open Source Risk Management ha evidenziato infatti come Linux violi potenzialmente 283 tecnologie proprietarie. Il comune di Monaco ha quindi affidato al noto giurista Bernhard Frohwitter, uno dei massimi esperti europei in materia di

proprietà intellettuale, uno studio che valutasse le implicazioni derivanti dall'uso di Linux in relazione alla nuova direttiva europea. Lo studio ha evidenziato come il rischio che Linux infranga brevetti sia molto modesto, e in ogni caso non superiore rispetto a quello di qualsiasi altro software.

Finalmente, giunti a ottobre 2004, il comune ha deliberato la continuazione del progetto, che ha così potuto andare avanti senza intoppi.

La migrazione verso Linux e altre tecnologie Open Source rappresenta un vero e proprio salto. I costi quindi non comprendono solamente le voci di consulenza, installazione, manutenzione e supporto; si devono includere anche i cosiddetti "switching costs", i costi derivanti dal cambio di fornitore. Questi ultimi comprendono ad esempio il passaggio a formati aperti dei documenti office, l'addestramento del personale all'uso di nuove tecnologie, resistenze di tipo "ideologico".

I costi totali sono stati stimati in 35,7 milioni di euro nell'arco dell'intero svolgimento del progetto.

La decisione del comune di Monaco ha destato grande preoccupazione in casa Microsoft, soprattutto per le possibili ripercussioni verso altre Pubbliche Amministrazioni. Steve Ballmer stesso, programmatore di Microsoft, è volato in Germania per convincere la giunta di Monaco a cambiare idea. La contro-offerta è stata quantomeno allettante: 27,3 milioni di euro per l'aggiornamento di tutti i sistemi all'ultima tecnologia Microsoft, offerta inferiore di 8,4 milioni di euro rispetto ai costi totali del progetto LiMux. Il comune di Monaco ha rifiutato l'offerta perchè la transizione "ha carattere di tipo strategico"; i "switching costs" tendono infatti ad aumentare esponenzialmente con il passare del tempo.

La reazione di Microsoft si è fatta sentire attraverso un suo portavoce di Monaco, il quale ha commentato: "non tutte le applicazioni girano su Linux".

Il perchè della decisione del comune di Monaco di migrare verso l'Open Source vanno ricercati in tre ordini di motivi:

-il parco software in circolazione era piuttosto vecchio. Sulla maggior parte delle macchine della P.A. girava Windows NT 4.0, il cui supporto da parte di Microsoft è stato interrotto a giugno 2004. La transizione era quindi del tutto necessaria;

-spezzare la dipendenza della Pubblica Amministrazione verso un unico fornitore;

-creare maggiore competitività all'interno del mercato del software.

La transizione va quindi vista come un obiettivo di tipo strategico di lungo periodo. Nel breve termine, conviene infatti aggiornare la tecnologia esistente. Il ragionamento sottostante è che la Pubblica Amministrazione non può e non deve dipendere da un solo fornitore, specie quando questo "chiude" i formati delle sue applicazioni. Ad esempio un documento scritto con l'editor di testi Microsoft Word può essere letto solamente con un'altra copia dello stesso programma; non c'è alcuna garanzia che altre applicazioni, quali OpenOffice, riescano a visualizzarlo con la stessa efficacia. Inoltre se la P.A. utilizza tecnologie non aperte, spesso anche i suoi clienti (i cittadini, le imprese) devono adeguarsi a tale tecnologia, appunto perchè non esiste il modo di creare dei connettori validi verso altre soluzioni.





## **6.Un caso reale: il CNIPA lancia l'Osservatorio Open Source**

In seguito alla Direttiva del Ministro per l'Innovazione del 19 dicembre 2003, il CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione) ha deciso di istituire l'Osservatorio Open Source ed il suo relativo Centro di Competenza, al fine di promuovere ed analizzare l'uso e la diffusione di applicativo a codice sorgente aperto.

Il Centro di Competenza "Open Source" - nato nell'ambito dell'Osservatorio Open Source - svolge un'importante funzione di catalizzatore delle "best practice" e della conoscenza in materia di open source.

I modelli e le problematiche introdotte dall'adozione di software open source necessitano un'adeguata e corretta comprensione e diffusione della materia per favorirne ritorni positivi. Le infrastrutture e l'organizzazione del Centro favoriscono l'accentramento di conoscenze ed esperienze e la diffusione del know-how nella pubblica amministrazione, grazie anche alla promozione di processi di valutazione e comparazione di software.

Gli obiettivi perseguiti dall'Osservatorio, e le linee di azione attuate dal Centro di Competenza, non solo sono allineati a quelli dei principali paesi europei, (esistono già nell'Unione europea dei centri di competenza o delle organizzazioni assimilabili all'Osservatorio del CNIPA), ma rientrano tra le

iniziative auspicate dal programma europeo IDA (Interchange of Data between Administrations).

In sintesi le principali attività del Centro di Competenza:

- raccordare e diffondere il patrimonio di esperienze svolte presso le Università, la ricerca pubblica, le piccole e medie imprese del settore, attivando anche collaborazioni, partnership;
- offrire link verso altri centri di competenza (italiani ed europei), e scambiare informazioni con gli stessi;
- mettere a disposizione forum di discussione in materia di OS aperti agli utenti, pubblicare i risultati delle rilevazioni sull'uso dell'OS, nonché studi specifici in materia;
- studiare tipologie di licenze d'uso e di contratti per prodotti e servizi OS adeguate per l'impiego nella PA (es: capitolati di gara);
- fornire supporto alle PA ove richiesto;
- svolgere azioni di verifica, di mantenimento e di miglioramento del software OS prodotto ex novo per la PA o frutto di riuso;
- allestire e gestire una *vetrina* di prodotti OS per la pubblica amministrazione, accessibile attraverso il portale del CNIPA, al fine di diffondere la conoscenza del software OS e di facilitare l'incontro tra la domanda di soluzioni tecnologiche e l'offerta di prodotti e servizi in ambito OS; tale vetrina, tra l'altro, elencherà i software di maggiore affidabilità e diffusione e pubblicherà le best practice segnalate dalle PA e le informazioni relative ai progetti di ricerca attinenti all'OS segnalati dalle Università;
- allestire e gestire una *piattaforma di sviluppo cooperativo* (CDE – Collaborative Development Environment) che metterà a disposizione le

risorse tecniche necessarie alla creazione di un ambiente, in ambito prettamente nazionale, per incentivare e facilitare lo sviluppo di software OS per la pubblica amministrazione, facendo in modo che il Centro di Competenza del CNIPA sia protagonista dell'iniziativa, non solo come fornitore e gestore del CDE, ma anche come ispiratore delle iniziative progettuali.

Il Centro di competenza si pone come struttura essenziale per il conseguimento degli obiettivi, configurandosi non solo come distributore, ma soprattutto come punto di riferimento e collettore di contributi della comunità alla stregua di quanto già realizzato in numerosi paesi membri dell'UE.

L'Osservatorio Open Source, di cui fa parte il Centro di Competenza, si pone tra i suoi obiettivi quello di avviare iniziative volte a promuovere il patrimonio d'esperienza maturata da università e centri di ricerca nazionale, verificare la compatibilità tra le licensing dei prodotti Open Source e le politiche in materie messe in atto dalla PA, a favorire l'incontro tra domanda e offerta di prodotti Open Source per gli istituti pubblici predisponendo strumenti online ad hoc.

Inoltre, si predisporranno attività di supporto all'adozione di software Open Source e verranno avviate indagini conoscitive su specifiche tematiche quali ad esempio best practice, sviluppo di singole applicazioni, aspetti gestionali, ecc, ed infine verrà incentivato lo scambio con analoghi centri di competenza europei.

Il progetto, secondo quanto riportato dal comunicato del CNIPA, si svilupperà in due tempi, in una prima fase verranno avviate iniziative ad incentivare lo scambio di informazioni sull'attuale presenza di software Open Source presso le singole unità operative; verranno pertanto richieste informazioni relative a:

- esigenze risolte parzialmente o completamente con utilizzo di software OS;
- prodotti OS utilizzati;
- aziende che hanno realizzato il progetto e/o fornito i servizi;

successivamente verranno coinvolte le università, alle quali verrà proposta la segnalazione dei progetti di ricerca attinenti alla tematica OS e a tal fine sarà approntato un protocollo d'intesa da concordarsi tra il CNIPA e ciascuna università.



## **7.Open Source e Pubblica Amministrazione: il comune di Udine insegna**

Il Comune di Udine ha deciso di utilizzare in modo pervasivo tecnologie Open Source in numerosi ambiti applicativi. A partire dal portale istituzionale e dell'Intranet interna, realizzate utilizzando OpenCMS (Content Management Systems interamente open source), per finire col comprendere parte delle infrastrutture (server Linux e database MySQL) e delle applicazioni di help-desk interne, l'open source viene utilizzato quotidianamente con successo. Antonio Scaramuzzi, responsabile dei Sistemi Informativi del Comune di Udine, spiega che attualmente i progetti a codice libero in via di sviluppo all'interno dell'amministrazione friulana sono essenzialmente tre:

i primi due riguardano sia il sito istituzionale del comune che l'Intranet dell'amministrazione. Entrambi sono basati su un software open source chiamato OpenCMS: ovvero una piattaforma su cui abbiamo sviluppato i servizi informativi che l'amministrazione offre ai cittadini tramite la Rete. Successivamente su questa base, quindi sempre utilizzando lo stesso applicativo, abbiamo realizzato anche l'Intranet aziendale per il supporto alla comunicazione interna e al fine di migliorare l'efficienza dei dipendenti comunali.

Il terzo progetto, più recente è un progetto per lo sviluppo di un applicativo di help-desk rivolto agli utenti dei sistemi informativi, ovvero ai dipendenti dell'amministrazione.

Scaramuzzi spiega che in realtà il processo di avvicinamento al mondo open source era in atto già da tempo sul fronte dell'infrastruttura Web: avevano già iniziato una progressiva migrazione verso ambienti non proprietari, in particolare per quanto riguarda il progetto del sito istituzionale la decisione è caduta in un momento di cambiamento dell'intera infrastruttura Web.

Già da tempo il sito da un punto di vista sia tecnologico che contenutistico richiedeva un rinnovamento sostanziale. Da parte dei responsabili dell'Ufficio Relazioni al Pubblico era sentita fortemente l'esigenza di poter aggiornare in modo più rapido e razionale le oltre 1000 pagine del sito. Si sono trovati dunque di fronte la necessità di adottare un sistema di CMS (Content Management Systems) per la gestione di tutta la mole di materiale prodotto e potendo scegliere il nuovo CMS abbiamo preso in considerazione anche software Open Source.

Sicuramente sin da subito una forte spinta è stata dettata dai costi inferiori: la possibilità di abbattere le spese delle licenze in favore di una maggior personalizzazione del software è stata determinante; inoltre, una serie di opportunità contingenti hanno ulteriormente avvallato questa scelta: infatti, sia internamente che sul territorio il comune friulano ha trovato ottimi partner commerciali come Tecnoteca srl, una software house friulana specializzata in soluzioni open source e che ha implementato il sistema OpenCMS, che aveva da tempo sviluppato competenze significative in ambito Open Source e comunque erano in grado di dare un notevole grado di sicurezza.

Scaramuzzi sottolinea che è bene chiarire sin da subito che l'Open Source non è gratuito, free in questo caso potrebbe creare un equivoco. Sebbene non

vi siano costi di licenza, argomento non indifferente sia chiaro, vi sono significativi costi iniziali legati alla start-up del progetto . Il set-up iniziale richiede non solo la presenza di abilità interne, ma anche la necessità di appoggiarsi ad una azienda esterna con esperienza e know-how nel settore. Molto dipende quindi dal partner a cui si fa riferimento: per la loro esperienza il comune di Udine ha potuto constatare che ci sono aziende sul mercato valide ed in grado di realizzare ottimi prodotti, nel caso specifico del portale istituzionale: Tecnoteca , infatti ha dimostrato ottime conoscenze nell'ambito dell'applicativo OpenCMS.

L'unico punto di attenzione per chi volesse intraprendere questa strada e di fare attenzione nella scelta delle società partner: molto spesso si tratta di giovani software house che non sempre hanno la dovuta accortezza nell'implementazione di infrastrutture delicate ed in ambito aziendale.

Tirando le somme il bilancio è senz'altro positivo, dice Scaramazza; a parità di costo abbiamo ottenuto risultati in termini di infrastrutture e prodotti di pari o migliore qualità ed affidabilità rispetto a soluzioni proprietarie; a giudizio degli utenti finali i risultati ottenuti sono stati piuttosto buoni ed i processi aziendali ne hanno guadagnato in snellezza e semplificazione.

Il risultato finale è stato un prodotto su misura, fatto a posta sulle esigenze della PA friulana; e sicuramente ottenere tali risultati tramite personalizzazioni con software proprietario si sarebbero dovuti affrontare sicuramente costi complessivi superiori

La scelta di applicativi Open Source è una strada che il comune di Udine ha intrapreso con soddisfazione da più di tre anni e che vorrebbe continuare a percorrere anche in futuro, magari non solamente confinato all'utilizzo sulle

infrastrutture e sui sistemi gestionali centrali ma anche per un utilizzo di programmi applicativi da installare direttamente sulle stazioni di lavoro degli utenti finali.

Uno dei principali parametri da prendere in considerazione per la valutazione di un Content Management Systems è la soddisfazione degli utenti finali che lo utilizzano, ovvero coloro che realizzano e gestiscono quotidianamente i contenuti del portale.

Marina Galluzzo, responsabile dell'Ufficio Comunicazione del Comune di Udine, ci spiega che in qualità di responsabili della comunicazione del comune sentivano forte l'esigenza di poter aggiornare facilmente e soprattutto in modo tempestivo i contenuti del sito Internet, sempre più importante per una corretta comunicazione con il cittadino/utente.

Inizialmente il sito era costruito in modo statico e questo non permetteva di gestire in modo efficiente le oltre 1000 pagine già allora esistenti. Abbiamo quindi riorganizzato l'intera struttura per agevolare una realizzazione dei contenuti decentralizzata, delegando alla sede centrale solo il coordinamento e la valutazione dei contenuti; inoltre, grazie ad OpenCMS è stato possibile raggiungere un buon livello di automazione nella gestione dei processi, cosa che ha reso il lavoro sicuramente più veloce e più efficiente: se per un ente pubblico la comunicazione con l'esterno è molto importante, la comunicazione interna non è da meno.

La realizzazione di un portale interno per i dipendenti del comune permette una maggior efficienza e una condivisione di informazioni molto diffusa. Ogni area ha una persona responsabile per l'inserimento dei contenuti nel proprio ambito di competenza, mentre il coordinamento avviene a livello



centrale dove in seguito un addetto alla comunicazione vaglia ciò che può essere condivisione su Internet, oppure solo internamente nell'Intranet.

Dopo tre anni dall'avvio del progetto, sicuramente il comune di Udine ritiene di poter dire che il bilancio è positivo, anche se i margini di miglioramento sono ampi. Oltre all'aspetto comunicativo sono stati implementati numerosi servizi al cittadino come il pagamento dell'ICI on line, la visualizzazione dei pagamenti, la possibilità di compilare le pratiche legate all'edilizia, e un supporto per l'ufficio oggetti smarriti.

Inoltre, nel rispetto delle direttive della legge, il portale istituzionale rispetta tutti i parametri di usabilità ed accessibilità (attualmente ha raggiunto il livello di raccomandazione 2A). A mesi inoltre partirà anche un forum per le attività giovanili, al momento non ancora attivo.

## 8. Conclusioni

L'Open Source sta coinvolgendo sempre più una molteplicità di protagonisti, che vanno dalle imprese private alle Pubbliche Amministrazioni, fino ai governi di alcuni paesi europei e non.

La Pubblica Amministrazione copre un importante ruolo nel settore software: la domanda che essa sviluppa è molto rilevante, e quindi le sue scelte possono influenzare molto il mercato e le strategie delle imprese: infatti una valutazione positiva del modello Open Source, potrebbe dare un nuovo impulso e diffusione del mercato.

Gli studi e le indagini effettuate hanno dimostrato che i principali motivi che spingono le PA ad adottare il software Open, sono sicuramente i bassi costi ma anche l'esigenza di offrire ai cittadini un buon servizio, di garantire l'indipendenza dalle strutture informatiche, dalle scelte dei singoli fornitori, di proteggere la sicurezza delle applicazioni, e di sfruttare al meglio le risorse e le conoscenze già presenti.

Tuttavia è importante fare scelte di adozione, sulla base di un confronto tra software Open e software proprietario; non è quindi opportuno per la PA un sostegno acritico all'Open Source: questo infatti offre rilevanti potenzialità di carattere sociale economico e culturale.

Il rapporto della commissione istituita dal Ministero per l'Innovazione e la Tecnologia dice: il fenomeno Open Source si ricollega a tematiche sociali, quali il tema della circolazione del sapere, della libertà di divulgazione

scientifici dei risultati di ricerca e al dibattito sulle questioni della tutela del diritto d' autore.

E' inoltre importante la diffusione dell'informazione presso i cittadini, relativa alla circolazione di documenti o dati, perché essa sviluppa implicazioni diffuse: in particolare sul tema dei formati aperti, essa è destinata ad avere un impatto sul rapporto tra PA e cittadini, stimolando così la cultura della condivisione.

## **Capitolo 5**

# **APPLICAZIONI OPEN SOURCE NEI CAMPI NO SOFTWARE**

## **Introduzione**

Fino ad ora abbiamo analizzato il nuovo paradigma Open Source, dal punto di vista tecnico-economico: siamo partiti dalle radici storiche del movimento, siamo passati attraverso l'analisi dell'Open Source come nuovo modello di Business, fino ad arrivare a spiegare le strategie e le competizioni che si sviluppano attorno al nuovo paradigma per la corsa ai profitti. Abbiamo visto inoltre l'importanza dello sviluppo del modello Open in ambiti tipo la Pubblica Amministrazione, e quanto il soggetto pubblico sia di rilevante importanza per l'introduzione all'utilizzo del software Open Source.

Cerchiamo ora di chiudere il cerchio della nostra indagine analizzando le modalità con cui negli ultimi anni è stata applicata la filosofia Open ed il sistema di licenze copyleft, in molti altri settori e rami, anche della creatività, che non sono strettamente legati ai campi software: ci addentriamo cioè nel mondo Open access.

## **1.Un nuovo concetto di produzione**

Accanto al modello Open Source si è sviluppato un nuovo concetto di produzione, meglio conosciuto come “produzione alla pari” (peer production), ovvero produzione “peer-to-peer” (P2P).

Gli effetti del fenomeno Open Source, infatti, non si limitano al contesto delle applicazioni software.

Il concetto di apertura che è alla base del modello Open, si sta diffondendo sempre più in vari campi di applicazione che riguardano altri beni della conoscenza: le iniziative che ricadono in questa analisi realizzano produzione, raccolta, organizzazione, valutazione e distribuzione dei beni della conoscenza.

Ma prima di analizzare alcuni casi in diversi campi di applicazione, è interessante soffermarsi sul legame tra il concetto di produzione ed il concetto di collaborazione, nel senso di apertura.

Come già osservato, il software è un prodotto della conoscenza, per cui le attività di sviluppo sono più rilevanti delle attività di produzione: la produzione infatti, nel caso software, assume i caratteri di una semplice riproduzione realizzata a costi irrilevanti.

Ciò consente, in questo caso, di utilizzare il termine produzione con riferimento al processo di sviluppo, creando così una larga sovrapposizione dei concetti di sviluppo e di produzione che non si verifica per i prodotti costituiti da una reale fisicità.

Nell'ambito dello sviluppo della conoscenza, così come nella ricerca scientifica, sono di fondamentale importanza le relazioni orizzontali, o alla pari.

Si tratta di interrelazioni di tipo peer-to-peer che hanno lo scopo di individuare i percorsi di sviluppo e valutare i risultati conseguiti: il peer-to-peer è una modalità di interazione bidirezionale, in cui ogni soggetto facente parte di un network di comunicazione, ha le stesse possibilità di ricevere, inviare e gestire informazioni.

Sono state costruite vere e proprie comunità, dove i soggetti possono condividere informazioni, opinioni e risorse, e quindi collaborare allo sviluppo e all'evoluzione di diversi progetti, evidenziando così un nuovo modello di scambio e di sviluppo di nuove iniziative.

Acquistano così più importanza, nuovi concetti di distribuzione (peer-distribution), di valutazione congiunta (peer-evaluation), di attribuzione di valore e di importanza (peer-relevance).

In questo scenario si inserisce anche il modello dell'Open Source, poiché si basa sulla collaborazione di numerosi soggetti che lavorano allo sviluppo di un software, confermando così l'esistenza di una peer-production, riferita a gruppi di persone auto-organizzate, cioè non subordinate a contratti di lavoro dipendenti.<sup>15</sup>

La peer-production, consente ad un gruppo di gente di auto-assegnarsi obiettivi e attività, e questo rende il modello particolarmente adatto per l'organizzazione di iniziative nelle quali la risorsa umana è di importanza superiore a qualsiasi altro tipo di risorsa.

---

<sup>15</sup> Benkler 2001-“Coase's Penguin, or, Linux and the Nature of the Firm”

La peer-production rappresenta un nuovo modello di elaborazione delle informazioni basato sullo sfruttamento combinato delle risorse e del capitale umano presente in una comunità di pari (peer) dove ognuno è considerato alla pari dell'altro, con gli stessi diritti, doveri e responsabilità.

In un modello 'chiuso' i contratti e la proprietà servono per assicurarsi la presenza e l'accesso limitato a pochi, delle risorse di produzione; viceversa un modello di produzione 'aperto', di tipo peer-to-peer, non limita e non vincola l'insieme di risorse disponibili, perché chiunque vi ci può accedere e aggiungersi autonomamente al progetto. Il vantaggio rilevante consiste nell'utilizzo e nella raccolta del capitale umano disponibile.

Il concetto Open si integra quindi con quello più generale di collaborazione (peer): tale effetto si nota quando si accede alla rete Internet.

Il web infatti è l'esempio più eclatante di produzione collaborativa gratuita fatta di milioni di utenti, e il suo valore non risiede nella gratuità, ma bensì nella sua apertura, consentendo così di sfruttare una ricchezza illimitata di capitale umano.

Analizziamo ora come questo nuovo modello di produzione Open Source sia stato applicato anche in altri campi, apportando rilevanti sostegni anche ai paesi meno sviluppati.

## 2. Nuovi campi di applicazione

Il concetto di apertura dell'Open Source può essere esteso ad altri campi di applicazioni. La logica Open Source, estesa nel più ampio concetto di peer-production, insieme alla logica del copyleft, potrebbe coinvolgere lo sviluppo e la diffusione non solo del software, ma anche di altri prodotti della conoscenza.

Infatti, oltre ai vari progetti legati allo sviluppo e alla distribuzione di software Open Source, il prefisso Open è stato spesso usato per indicare la vocazione dei progetti verso la filosofia della condivisione e della libertà.

Pensiamo ad esempio a OpenPress che viene comunemente e generalmente utilizzato per indicare ogni tipo di pubblicazione gratuita disponibile in Internet: ad esempio alcuni siti Internet di alcune case editrici italiane, riportano delle sezioni dedicate al fenomeno dell'OpenPress in cui sono scaricabili gratuitamente articoli, documenti, saggi, libri in versione parziale o anche intera.

A proposito della libera letteratura, in Italia esiste da anni il sito [www.liberliber.it](http://www.liberliber.it), che raccoglie diverse opere letterarie i cui diritti di autore sono scaduti, e prova a creare un'immensa biblioteca il cui accesso sia libero e gratuito.

Interessante è anche il progetto sviluppato nel 2001 in Germania, meglio conosciuto come OpenMusic<sup>16</sup>, che garantisce tre libertà:

-libertà di ascoltare musica quando si vuole;

---

<sup>16</sup> vedi: <http://openmusic.linuxtag.org>



-libertà di distribuire musica sia a livello privato che commerciale (nel secondo caso bisogna fare in modo che l'autore possa beneficiare dei profitti)

-la libertà di modificare la musica.

Un altro ambito decisamente interessante su cui è stato applicato il criterio di copyleft è quello delle opere di compilazione, come ad esempio enciclopedie, dizionari, banche dati, ecc...

Esistono già molti esempi di opere di questo tipo con permesso di copia e soprattutto di modifica: infatti il successo di questo paradigma di tutela dipende dalla continua necessità di aggiornamento che tali opere presentano e che viene pienamente soddisfatta dalla filosofia Open Source.

Mettere a disposizione un' enciclopedia o una banca dati liberamente aggiornabile da chiunque fa sì che il prodotto sia sempre completo, aggiornato e funzionale alle attività di ricerca.

L'esempio più eclatante di questa applicazione è l'enciclopedia libera Wikipedia<sup>17</sup>.

---

<sup>17</sup> vedi: [www.wikipedya.org](http://www.wikipedya.org)

### 3. Il progetto Wikipedia

Dall'esigenza delle comunità di cercare uno spazio on-line, che permetta il confronto, il dibattito e la discussione, e che vada al di là dei forum, è nato il progetto Wikipedia.

Wikipedia, la nuova enciclopedia on-line, è la dimostrazione di come il dibattito e la discussione può essere costruito proprio dagli stessi membri della comunità.

Il progetto è nato nel 2001, e da allora fino ad oggi conta più di 500 mila articoli.

Tutti gli articoli saggi e le semplici definizioni, contenuti nell'enciclopedia, e disponibili in tutte le lingue (si consiglia di esplorare la sezione nel linguaggio Esperanto), sono frutto del contributo degli utenti, sono catalogati per argomento o in ordine cronologico e comunque ritrovabili grazie ad un apposito motore di ricerca.

Esistono inoltre anche sezioni monografiche dedicate per esempio alle biografie, all'attualità e agli anniversari storici.

Inoltre, all'interno del sito, è spiegata la modalità per contribuire alla realizzazione dell'opera con una dettagliata procedura di *upload*<sup>18</sup>, e come segnalare alcuni articoli diventati obsoleti, oppure redatti con un'ottica poco obiettiva.

---

<sup>18</sup> vedi: [it.wikipedia.org/wiki.cgi?Wikipedia-Guida Essenziale](http://it.wikipedia.org/wiki.cgi?Wikipedia-Guida_Essenziale)

Questa è una delle caratteristiche più interessanti di Wikipedia: infatti ogni utente può diventare l'autore dell'enciclopedia, e se un utente/autore non è d'accordo con una definizione o un articolo, può liberamente modificarlo.

Esiste nel sito, infatti, il Neutral Point of View <sup>19</sup>, cioè il punto di vista neutrale nella stesura degli articoli, che ha il compito di controllare che le informazioni siano diffuse in modo neutro, standard, senza prese di posizione.

Questo vuol dire che gli articoli pubblicati su Wikipedia passano sotto il severo controllo dell'intera comunità, e se sopravvivono, vuol dire che sono di affidabile credibilità.

Il progetto Wikipedia, prende spunto da un precedente progetto di enciclopedia on-line, meglio conosciuto sotto il nome di Nupedia, che però in poco tempo si è rivelato obsoleto.

L'approccio di Nupedia era quello convenzionale: gli articoli erano scritti da un esperto e pubblicati solo dopo un lungo processo di controllo e di accurata analisi dei fatti. Al contrario Wikipedia, iniziò a far leva su uno stile informale di interazione, che ben presto mise in luce un grande potenziale di crescita.

I fondatori di Wikipedia attuarono una strategia diversa e originale rispetto a quella di Nupedia: infatti gli articoli pubblicati potevano essere controllati e modificati in ogni momento da parte di chiunque.

All'inizio questa applicazione Open Source, fu considerata una metodologia disordinata ed impropria, come strumento di divulgazione e di consultazione

---

<sup>19</sup> vedi: [it.wikipedia.org/wiki.cgi?NPOV](http://it.wikipedia.org/wiki.cgi?NPOV)

delle informazioni, ma il tempo ha dato conferma che il suo ampio grado di apertura ha comunque goduto di un notevole successo.

Un altro elemento che ha permesso il successo del prodotto on-line, è la facilità, dal punto di vista tecnico, non solo di potervi accedere, ma anche la facilità di modifica degli articoli.

All'interno di Wikipedia, esiste infatti un archivio che assicura che una cattiva pubblicazione non può causare nessun danno permanente.

Le versioni archiviate sono infatti contenute nella "page history": essa contiene un link di accesso alla versione salvata, un link per visionare le differenze tra la versione salvata e quella precedente, mostrando le modifiche effettuate e indicando inoltre la data e l'ora dell'avvenuta modifica, l'autore della modifica, ed i commenti che ha lasciato il soggetto che ha apportato la modifica stessa.

Inoltre Wikipedia dà l'opportunità ai suoi utenti di tenere una "watch list" delle pagine che si desidera monitorare: quando una pagina di questa lista viene modificata, l'utilizzatore viene avvisato via e-mail.

Questo è uno strumento che consente un continuo monitoraggio per gli esperti wikipediani, in quanto consente di esaminare da subito le modifiche apportate agli articoli e correre subito ai ripari.

Lo scopo di Wikipedia è quello di alimentare e incoraggiare la crescita della conoscenza e dare la possibilità a chiunque di ingrandire il proprio bagaglio culturale.

Per gestire l'enciclopedia multilingue, parallelamente al progetto wikipedia, se n'è sviluppato un dizionario libero e multilingue, meglio conosciuto come

Wiktionary<sup>20</sup>, che può vantare un minor numero di versioni linguistiche rispetto a Wikipedia, solo perché è nato più tardi.

Nessuno degli argomenti trattati nel progetto Wikipedia è sottoposto alla “super” visione di specialisti, ma ancora una volta la volontà e la capacità di cooperazione di un’intera comunità, dimostra come il lavoro collettivo, può garantire la crescita individuale e comunitaria.

---

<sup>20</sup> vedi: [www.wiktionary.org](http://www.wiktionary.org)

## **4. Il movimento Openaccess**

Un campo di applicazione della filosofia Open Source che risulta molto interessante da analizzare è quello della ricerca scientifica.

Dall'esigenza di liberare la diffusione dei contenuti di valore scientifico accademico in ambito telematico, nasce l'iniziativa Openaccess.

Con il termine Openaccess non si identifica un progetto specifico, ma bensì un movimento, nato specificatamente in ambito accademico.

L'Openaccess rappresenta un modello concettuale e una filosofia che intende diffondere i principi Open Source a varie tipologie di argomenti e informazioni, liberamente contenuti in rete.

Il termine Openaccess, descrive la possibilità, da parte di chiunque, di pubblicare in rete prodotti della conoscenza, come riviste scientifiche, libri, saggi, ecc., lasciando la possibilità a chiunque di leggere, copiare, usare, o distribuirle.

L'Open access è diventato nel tempo un fenomeno sociale di rilevante importanza, in grado di alimentare l'innovazione tecnologica e soprattutto l'evoluzione culturale.

Il movimento per l'accesso aperto, promuove forme alternative di comunicazione culturale, che traggano il massimo profitto dalle tecnologie di rete. L'obiettivo unificante di tutte le diverse anime del movimento è l'accesso gratuito all'informazione, almeno nella versione elettronica.

Anche sulla circolazione dell'informazione scientifica affidata alle pubblicazioni editoriali, vigono le regole di diritto di autore o copyright, le stesse che governano il campo dell'informatica, e nonostante ciò le pubblicazioni continuano ad essere un elemento fondamentale per la comunità scientifica.

Attraverso la pubblicazione su riviste specializzate, l'informazione circola e permette il confronto, la competizione e la cooperazione tra i vari ricercatori, a costi zero, ed inoltre permette la visibilità in ambito accademico di molti dei ricercatori.

## **5.L'illusione del pubblico dominio**

In realtà, la metodologia Open access, in ambito scientifico-tecnologico, è ancora tutta da elaborare.

La novità di questo approccio provoca resistenze anche da parte di chi si oppone alla privatizzazione della ricerca.

Alcuni esempi che andrò a spiegare, sottolineeranno come “l'illusione del pubblico dominio”, nel campo scientifico, in realtà potrebbe rivelarsi solo una strategia economica, per la corsa ai profitti, sfruttando però le conoscenze della comunità.

E' il caso per esempio di alcuni ricercatori del Ranger Institute di Cambridge, in Inghilterra, impegnati nell'attività di sequenza del DNA umano, nello Human Genome Project: questi infatti intuirono che la brevettabilità delle sequenze avrebbe privatizzato il loro lavoro.

In un primo momento infatti provarono ad utilizzare le proprietà intellettuale per garantire l'accesso alle ricerche, cercando l'ispirazione proprio nella cultura informatica del copyleft; poi per rispettare lo spirito di condivisione, tipico della comunità scientifica, hanno scelto semplicemente di rendere le scoperte di pubblico dominio, a disposizione di tutti, certo, ma senza garanzia che prima o poi quei risultati non sarebbero stati utilizzati in altri brevetti, o più in generale che quei risultati non sarebbero stati privatizzati.



La strategia di brevettare un'invenzione per liberarla, è stata applicata anche nel caso dei geni BRCA1 e BRCA2: questi geni sono responsabili al 5-10% dei casi di tumore al seno.

La Myriad Genetics, deteneva brevetti su due geni, sul loro gene soppressore P15 e su anticorpi e proteine associati alla soppressione dei due geni.

Questi brevetti davano il monopolio alla Myriad sui test genici, sui kit diagnostici e sui prodotti terapeutici, negli Stati Uniti, in Europa e in Giappone.

Va sottolineato che la maggior parte della ricerca che condusse alla scoperta dei due geni, fu la maggior parte il frutto della collaborazione internazionale fra vari gruppi di ricerca, e coinvolse anche donne portatrici del tumore che si adoperarono per favorire lo studio del fenomeno.

Solo quando la ricerca stava per avvicinarsi alla fase conclusiva, la Myriad decise di impossessarsi della ricerca, richiedendone i brevetti, approfittando anche dei risultati delle ricerche altrui.

La maggior parte delle ricerche, furono effettuate nel regno Unito, al Ranger Center, e all' Institute of Cancer Research.

Come conseguenza alla richiesta dei brevetti da parte di Myriad, tutt'oggi negli Stati Uniti le donne possono accedere a test generici sul tumore al seno solo pagando profumatamente.

Diverso invece è la realtà dell'Europa: infatti dal 2000 al 2003, la Myriad cominciò a fare pressioni sulle organizzazioni della sanità inglese a causa dei bassi costi che proponeva sui test generici sul tumore.

Myriad rivendicava i suoi brevetti accusando la sanità inglese di implementare questi test in maniera "illegale", in quanto non rispettavano le

proprietà intellettuali, e richiedeva a gran voce di porre fine a questi test a basso costo.

Nella prima metà del 2004, i brevetti furono revocati: la corte europea infatti riconobbe l'eccessiva estensione dei brevetti e quindi il rischio che essi rappresentavano per la ricerca futura.

Inoltre secondo l'Institute Pasteur francese, la richiesta del brevetto sul gene BRCA1 era basata su una sequenza genetica errata.

A conclusione dei fatti, la Myriad perse il suo dominio, e ora il test sul gene è in mano al Cancer Research Uk, che si è impegnato a garantire la fruibilità per i laboratori pubblici.

Anche in questo caso il Cancer Research Uk, ha promesso di rendere di "pubblico dominio" i risultati delle ricerche, ma non ha garantito la non privatizzazione in nessun tipo di circostanza; ma il forte movimento di protesta contro i brevetti su quelle sequenze, espose il Cancer Research Uk a un'opinione pubblica preparata, e solo questo potrebbe scongiurare possibili deviazioni in senso privatistico.

Nonostante però questa "illusione del pubblico dominio", che lascia sempre più perplessa la comunità scientifica, si stanno sviluppando metodologie di ricerca, di diffusione di programmi e nuovi strumenti per apportare nuove forme di terapie del tutto Open Source e del tutto credibili.

Sono sicuramente più affidabili, in quanto sono sviluppate sotto tutela giuridica di licenze Creative Commons.

Inoltre sono strumenti ancora in via di sviluppo e di sperimentazione, anche perché a differenza del settore software, quello biotecnologico è prima di tutto più chiuso a livello di proprietà intellettuale, e inoltre a differenza del

settore software, in cui chiunque può identificarsi in un hacker (un hacker di fatto può essere uno studente, una casalinga esperta di informatica, un ingegnere informatico, un impiegato o un programmatore), nel caso della comunità scientifica, possono alimentare le ricerche e apportare sviluppi, solo chi detiene una professionalità nel settore.

Nonostante il lento sviluppo, però, questi nuovi strumenti potrebbero essere di reale supporto per la comunità.

Analizziamole meglio.

## 6. Il personal biotech

L'informatica degli ultimi anni, ha dimostrato come l'interesse di molti programmatori si sta spostando sempre più dal PC al DNA: molti di loro, infatti, convertiti di recente allo studio del DNA, non hanno voluto abbandonare lo spirito di condivisione in cui erano cresciuti nel mondo dell'informatica free.

Il risultato di questo desiderio si chiama Bioinformatics.org: si tratta di un'organizzazione no-profit, attiva su Internet, fondata nel 1998 da Jeff Bizarro, ex dottorato di ricerca dell'Università del Massachusetts di Lowell, e dal suo direttore di ricerca, Keneth A. Marx.

Attualmente i programmi distribuiti e sviluppati attraverso Bioinformatics.org, sono più di novantamila, e seguono la stessa filosofia Open Source.

Ma non si tratta soltanto di programmi utili in laboratorio: Bioinformatics, infatti mette a disposizione anche banche dati di libero accesso, contenenti soprattutto sequenze genetiche.

Il sito Bioinformatics, si sviluppa sotto licenza Creative Commons: il progetto nasce nel 2001 ed è diretto da Lawrence Lessino professore della facoltà di legge di Stanford.

L'intento di Creative Commons è quello di estendere il copyleft a campi diversi dal software, mettendo a punto licenze adatte ad altre forme di creatività come siti web, musica immagini, letteratura, testi, ed altro ancora;

ma la cosa più interessante è che le licenze sono personalizzabili attraverso il sito [www.creativecommons.org](http://www.creativecommons.org).

Dal 2005 nasce Science Commons: esso fornirà licenze comprensibili e giuridicamente valide per consentire un accesso più libero alle pubblicazioni scientifiche, agli strumenti di formazione e alle banche dati utilizzate dai ricercatori.

Ma l'urgenza di un brevetto copyleft è spinte dall'integrazione di medicina, biologia, farmacologia e ingegneria genetica, a cui stiamo assistendo.

Grazie all'approfondimento della conoscenza dei meccanismi genetici, si sta sviluppando sempre più l'immagine di un organismo di tipo modulare, composta da geni "accesi" o "spenti", nei singoli individui, in varie fasi della vita e in risposta a stimoli differenti paragonabili a sequenze di 0 e 1 comprensibili da un calcolatore: l'attenzione si sta spostando sempre più dalla biologia della specie umana a quella del singolo individuo.

In risposta a questo mutamento culturale, si aggiunge la possibilità tecnica di attribuire ad ognuno una "carta di identità genetica", cioè l'elenco completo della nostra sequenza di basi, registrata in una banca dati e disponibile su supporto digitale a costi contenuti.

Molti sistemi sanitari nazionali si stanno muovendo verso la creazione di banche dati, in modo che ogni singolo cittadino ha a disposizione una scheda sanitaria collegabile ad un banca dati centrale, con lo scopo di monitorare al massimo dettaglio cure, costi e lo stato di salute.

Le schede di salute verranno rese disponibili appena i loro costi saranno abbordabili.

La personalizzazione in atto sarà meglio conosciuta come “farmacogenomica”<sup>21</sup>, che consisterà nello sviluppo di farmaci personalizzati in funzione del particolare profilo genetico: servirà soprattutto per i pazienti in cui i farmaci attuali provocano effetti indesiderati, che secondo alcune stime provocano centinaia di vittime ogni anno.<sup>22</sup>

La farmacogenomica è ancora un sistema in via di sviluppo.

La caratteristica più importante di questa medicina personalizzata, consiste nella compensazione delle economie di scala delle case farmaceutiche: in primo luogo la disponibilità dei farmaci potrebbe essere più rapida in quanto meno mirata<sup>23</sup>; inoltre i farmaci su misura potrebbero essere somministrati preventivamente sullo sviluppo di patologie per pazienti geneticamente predisposti, e intervenire direttamente sulla base genetica di molte funzioni, come la funzione di proteine o l’attività di alcuni enzimi, e correggere alla radice anomalie patologiche..

Se così fosse la medicina “farmacogenomica” agirebbe preventivamente sui pazienti, e non per il tempo limitato della medicina tradizionale.

Per l’industria farmaceutica, la transazione può rivelarsi un affare.

I costi necessari a produrre molecole biologiche attraverso le biotecnologie vanno verso la riduzione: in poco tempo, la produzione di alimenti, farmaci e sostanze chimiche con strumenti biotecnologici da personalizzare in funzione dei bisogni di ciascuno, potrebbero essere alla portata di tutti, ed inoltre la convergenza tra biologia e informatica, potrebbe rivoluzionare il

---

<sup>21</sup> vedi: rivista “Science” n.286 pp. 487-491

<sup>22</sup> vedi: “Journal of Human Genetics”, vol.47, n.11, pp.561-566, 2002

<sup>23</sup> Le case farmaceutiche spendono troppo tempo e troppo denaro per la sperimentazione di un farmaco: costi aggiuntivi che verranno poi compensati dal paziente nell’atto di acquisto del farmaco  
vedi: “Modelli di business per Linux e Open Source” di Martin Fink, pp.160

mercato, creando una piattaforma su cui far convivere dispositivi biologici ed informatici.

## 7. Biologia Open Source: Bios

Il concetto di Openaccess, e la filosofia Open Source è stata applicata anche nel campo biologico.

L'iniziativa nasce in Australia si chiama Bios, cioè Biological Innovation for an Open Society, e tra gli obiettivi del progetto vi è la possibilità di esplorare strategie innovative per le invenzioni biologiche.

Il progetto è stato lanciato da un centro di ricerca australiano di nome CAMBIA<sup>24</sup> (Centro per l'applicazione della biologia molecolare all'agricoltura internazionale), e diretto dal biologo Richard Jefferson.

Il centro CAMBIA è dedicato alla diffusione delle tecniche biologiche locali, ed è già stato applicato in molti paesi in via di sviluppo attraverso i suoi programmi di cooperazione.

L'obiettivo principale è quello di fare in modo che anche l'agricoltura non industriale, possa usufruire dei vantaggi biotecnologici, senza farseli necessariamente imporre dall'industria del Nord del mondo.

Grazie all'enorme numero di brevetti depositati, le grandi industrie agroalimentari, possono costringere agricoltori di tutto il mondo ad utilizzare semi brevettati sterili e prodotti chimici sviluppati appositamente.

Così facendo le grandi imprese interrompono la diffusione dei saperi, delle risorse e delle tecniche tipica della società contadina, provocando un

---

<sup>24</sup> vedi: [www.cambia.org](http://www.cambia.org)



violento cambiamento circa l'impatto della biotecnologia sulle economie rurali.

Secondo i promotori di Bios, è importante importare nella comunità scientifica della biologia molecolare gli strumenti e le modalità dell'informatica Open Source.

Le esigenze degli agricoltori, sono molto simili a quelle degli utenti informatici davanti ad un software proprietario: poter manipolare liberamente la tecnologia per adattarla ai propri bisogni.

Anche i piccoli agricoltori, infatti, potrebbero trarre profitto dalle biotecnologie, se avessero libero accesso ad esse e se avessero la libertà di adattarle alle loro esigenze locali.

I contadini Hacker non hanno intenzione di copiare i brevetti sui semi, ma di aggirarli e di creare nuove e, se possibile, migliori varietà ecocompatibile.

A questo punto ci possiamo arrivare solo condividendo le esperienze, i risultati delle sperimentazioni e i semi selezionati.

Interessante inoltre è l'iniziativa del Prof. Raoul Robinson che sul sito [www.sharebooks.ca](http://www.sharebooks.ca), mette a disposizione gratuita i suoi libri su come selezionare le piante ristabilendo la loro naturale resistenza alle malattie, e anche libri di buona qualità come 'Breed your own variety' di Carol Deppe dove si insegna a creare nuove varietà e a trarre linee stabili dai semi ibridi.

Per superare le barriere dei copyright, il progetto Bios lavora su due piani: da un lato cerca di facilitare l'accesso alle informazioni circa gli innumerevoli brevetti esistenti, non sempre di facile consultazione, sviluppando strumenti informatici che permettono l'accesso gratuito al database già sviluppato da Cambia e mettendo a disposizione di tutti le informazioni contenute nei testi dei brevetti.

Dall'altra, Bios vuole facilitare la condivisione degli strumenti, creando una vera e propria comunità telematica, meglio conosciuta come BioForge<sup>25</sup> dedicata a chi fa ricerca nel campo biotecnologico.

In rete sarà quindi possibile ricercare informazioni tecniche rilevanti: Cambia di fatto ha già messo a disposizione le invenzioni del proprio centro di ricerca.

Inoltre per assicurare la libera diffusione delle informazioni, Bios creerà delle licenze in uso Open Source, che diano libero accesso agli strumenti di ricerca per tutti gli utenti a condizione che siano rispettate le clausole di modifica, di miglioramento e di condivisione, e considerate come commons,<sup>26</sup> cioè pubbliche.

Queste licenze sono la base di un movimento crescente, che permetterà ai ricercatori, anche nei paesi meno sviluppati, di scegliere e sviluppare tecnologie di miglioramento agrario che soddisfano le priorità dei popoli e non quelle delle multinazionali.

Bios inoltre è stato anche citato dalla rivista “Plos Medicine<sup>27</sup>” da parte di due membri del gruppo di Medici senza frontiere, per la richiesta di un trattato internazionale per la ricerca e lo sviluppo in campo farmaceutico, che dia priorità alla salute globale, piuttosto che al business delle case farmaceutiche.

Tra i punti salienti della proposta vi è una legislazione che protegga l'accesso libero ai risultati della ricerca e che incentivi alla cooperazione internazionale.

---

<sup>25</sup> vedi: [www.bioforge.org](http://www.bioforge.org)

<sup>26</sup> vedi: [www.bios.net/uploads/images/16/BIOS\\_Initiative\\_CC.pdf](http://www.bios.net/uploads/images/16/BIOS_Initiative_CC.pdf)

<sup>27</sup> rivista Openaccess

La più importante scoperta rilasciata da CAMBIA è inerente all'interazione naturale tra batteri e piante per conseguire la trasformazione delle piante stesse.

Fino ad oggi, la chiave del trasferimento di DNA modificato sulle piante è stato un organismo meglio conosciuto come *Agrobacterium tumefaciens* (At).

Questo è uno dei più efficaci sistemi di manipolazione genetica delle piante, capace di trasferire i geni estranei alle piante per modificarle, e chi vuole usare l'At deve pagare costosi diritti d'autore.

Inoltre è stato scoperto che l'At è un batterio della terra che apporta difetti alle piante, e che molti raccolti di importanti specie di colture mondiali sono difficili da trasformare, attraverso questa metodologia, perché secondo gli autori, l'At trasforma le piante in patogene<sup>28</sup>

In un articolo di una rivista scientifica Openaccess, di nome Nature<sup>29</sup> CAMBIA riporta la scoperta dell'esistenza di altri batteri, capaci di trasferire i geni nelle piante.

Rhizobium, Sinorhizobium e Mesorhizobium, rappresentano solo due delle differenti famiglie di batteri, e gli autori credono che l'abilità di trasferire geni, può essere estesa anche ad altri tipologie di batteri.

Il potenziale è alto per lo sviluppo di nuovi metodi di trasferimento di geni accessibili alle ricerche mondiali, e tali metodi possono anche essere più vicini alle necessità di sviluppo di ciascun paese del mondo.<sup>30</sup>

---

<sup>28</sup> cioè portatori di malattie

<sup>29</sup> vedi: [www.nature.org](http://www.nature.org)

<sup>30</sup> vedi: [www.wrenmedia.co.uk](http://www.wrenmedia.co.uk)

## **8. Open Source, Free Software e Open Formats nei processi di Ricerca Archeologica**

I processi di ricerca archeologici hanno visto, negli ultimi dieci anni, un notevole sviluppo grazie all'introduzione del calcolatore. Questa crescita ha messo in risalto due temi centrali per il futuro dell'informatica applicata all'archeologia: analisi e diffusione dei dati.

Parallelamente, nello stesso decennio, si è assistito al crescente sviluppo di applicativi e sistemi Open Source e Free Software (OS/FS). L'esempio più famoso è costituito dal sistema operativo GNU/Linux che, nelle varianti offerte dalle diverse distribuzioni, viene utilizzato da milioni di utenti. Ma sono molti gli applicativi OS/FS che si contraddistinguono per qualità ed affidabilità, caratteristiche favorite dall'esistenza di comunità interessate, intraprendenti e spesso numerose.

Il grado di popolarità del software OS/FS è dimostrato anche dalla sua crescente diffusione anche in ambito archeologico. Di fatto, essi garantiscono all'archeologo l'utilizzo di applicativi avanzati, dal costo contenuto, con ritmi di crescita e di aggiornamento competitivi rispetto a quelli del software commerciale proprietario, e con la possibilità concreta di intervenire, direttamente o indirettamente, nel processo di sviluppo, personalizzazione o, addirittura, rielaborazione del software. La crescente sensibilità su questi temi è testimoniata dalla nascita in Italia di gruppi di archeologi dedicati alla promozione del software OS/FS. Altro tema di grande importanza in questo ambito è quello dei formati aperti. Per

imprimere una svolta ancor più decisiva nei campi dell'analisi e della pubblicazione dei dati archeologici, non basta infatti la disponibilità (ovvero la libera circolazione) del software: occorre anche muoversi affinché l'enorme volume di dati prodotto dalla ricerca archeologica sia codificato digitalmente in formati aperti (cioè accessibili e utilizzabili da chiunque senza alcuna restrizione).

Il presente invito è indirizzato in particolare a tutti quei ricercatori che desiderino presentare e discutere lavori, progetti e problematiche relative all'impiego di software e formati liberi in campo archeologico. Particolarmente graditi saranno i contributi che, andando oltre la semplice comunicazione dell'uso di un pacchetto software, forniscano elementi di riflessione, proposte o soluzioni concretamente utilizzabili (quali confronti tra alternative libere e proprietarie, discussioni critiche delle funzionalità presenti o desiderabili, spunti per la personalizzazione, analisi delle possibilità di integrare vari componenti per costituire ambienti funzionalmente più completi, suggerimenti per la rappresentazione digitale di dati archeologici, eccetera). Tenendo conto delle sottomissioni, il workshop potrà essere suddiviso in più sessioni tematiche quali, ad esempio: Web-based Applications, Database Applications, GIS Quantitative Methods, Open Formats.

## Conclusioni

Da quando ho deciso di trattare come argomento di tesi l'Open Source, le ricerche di informazioni specifiche all'argomento, si andavano sempre più intensificando.

Mi incuriosiva molto l'idea di un business sviluppato all'insegna di comunità, cooperazione, pubblico dominio e gratuità, e grazie a testi di letteratura economica e a testi che in modo più dettagliato analizzavano il nuovo paradigma Open Source (talaltro disponibili e scaricabili gratuitamente in Internet), sono riuscita a farmi un'idea di questa nuova forma di produzione, di come si sviluppasse, ed ho avuto la possibilità di paragonarla a sistemi di produzione chiusi e di capirne le differenze.

Di fatto il fenomeno Open Source nasce da un'esigenza non solo tecnologica, ma prima di tutto sociale, in risposta ai malcontenti nei confronti delle proprietà intellettuali.

L'informatica di questi anni, ha dimostrato come ci possa essere innovazione in un mercato in cui le informazioni sono alla portata di tutti, e ancora quanto sia importante che ognuno possa usarle liberamente e arricchirle.

L'Open Source si inserisce così a pieno titolo in questa prospettiva di libera espressione comunitaria, attraverso la quale programmatori di tutto il mondo, condividono senza dazi, le loro invenzioni.

I suoi più accesi sostenitori ne hanno fatto quasi un'ideologia, legata al concetto di software libero: un'entità sempre più pervasiva nelle attività economiche e di relazioni sociali.

L'Open Source è legato ad un'idea di libertà e ad un concetto di comunità di sviluppo ed è basato sullo sfruttamento di alcune diversità presenti nella comunità: l'eterogeneità di soggetti, l'indipendenza e la disponibilità volontaria sono caratteristiche che vengono opportunamente valorizzate invece di essere contrastate, come avviene in un ambiente di sviluppo tradizionale.

A mio avviso, sono esattamente queste diversità che hanno destato paura e timore alle grandi software house, tipo Microsoft.

Di fatto queste non si sono trovate a dover competere con minacce di concorrenti che introducevano nel mercato prodotti innovativi con produzioni più veloci o che offrivano servizi differenti, ma bensì si trovavano dinanzi ad un paradigma che proponeva le stesse tipologie di prodotti, ma con una modalità di produzione completamente contrapposta a quella tradizionale, coinvolgendo diverse generazioni eterogenee di programmatori.

Ho cercato di capire, attraverso un'analisi dei costi-benefici, quali fossero le motivazioni che avrebbero potuto spingere un qualsiasi esperto di informatica a dedicare parte del suo tempo allo sviluppo di programmi software in modo del tutto gratuito.

Infatti dal lato dei costi è facile dedurre che a livello di infrastrutture sono pari a zero, di fatto tutti i soggetti che collaborano alla comunità Open possiedono già un computer per ragioni di studio o di lavoro, o addirittura

scrivono parti di codice durante le ore lavorative, per cui anche la connessione ad internet costituisce un costo.

Dal lato dei benefici, invece, la loro rinuncia al tempo libero ed alle ricompense economiche, saranno colmate dalle opportunità di apprendimento e di miglioramento della propria reputazione professionale da cui ne deriva un aumento del “capitale professionale”.

In più la visibilità che un giovane programmatore può avere nel collaborare alla programmazione Open, e la possibilità di guadagnarsi una reputazione nel creare codici sorgenti e competere con altri programmatori per lo sviluppo di una nuova soluzione informatica, sottolineando così le proprie capacità, può essere maggiore di quella che si può acquisire all'interno di una grande azienda informatica.

Di fatto molti esperti hanno fatto leva sull'esperienza acquisita nelle comunità Open, per fondare nuove imprese di successo.

La capacità e la bravura dei programmatori Open acquisita nelle comunità, hanno dimostrato di aver saputo sviluppare prodotti Open che hanno conquistato la preferenza degli utilizzatori, anche in presenza di prodotti proprietari già pienamente affermati.

L'affermazione dei prodotti proprietari e la loro diffusione è avvenuta grazie ai loro ritorni crescenti.

A questo punto bisogna capire da dove arrivano i rendimenti crescenti nel caso dei prodotti Open e come sia possibile che questi prodotti si siano diffusi in presenza di standard già pienamente affermati.

Per spiegare questo concetto, dobbiamo innanzitutto analizzare rapidamente il prodotto software: il software è un prodotto della conoscenza e in quanto



tale può essere scambiata e usata senza che si impoverisca; inoltre il software è un prodotto digitale, quindi infinitamente riproducibile e trasmissibile a costi irrilevanti, attraverso la rete Internet.

Il costo di realizzazione di un sistema operativo è molto elevato, ma le copie successive sono a costo quasi zero, per cui si hanno alti costi fissi per la prima copia e costi marginali irrilevanti per la riproduzione.

Inoltre una rete, in particolare quella Internet, è uno strumento molto potente per la diffusione di informazioni, e l'economia di rete premia molto di più i sistemi aperti rispetto a quelli chiusi.

Infatti in una rete si creano facilmente *esternalità di rete*, in quanto il valore di un prodotto per un utilizzatore, dipende molto dal numero di altri utilizzatori: cioè più persone si collegano alla rete e più aumenta il valore della rete stessa, non in modo lineare bensì in modo esponenziale.

Così si va producendo un continuo feed-back positivo, che rende sempre più grande una rete ed il suo valore, nel senso di ricchezza di informazioni, che aumenta all'aumentare del numero dei suoi utenti, in modo che diventi sempre più interessante e vantaggioso appartenere ad una rete grande e diffusa.

I ritorni crescenti in una rete, quindi, sono creati da una molteplicità di soggetti, e di conseguenza i vantaggi sono condivisi da tutti i partecipanti, e anche in caso di disparità nell'appropriazione del valore, il valore stesso risiede nella rete di relazioni tra gli agenti che interagiscono per crearlo.

Anche il teorema di Coase sottolinea la positività delle esternalità di rete che caratterizzano il modello Open Source: infatti il teorema di Coase, sostiene

che quando le parti interessate da esternalità possono negoziare tra loro senza costi, si perviene sempre ad un risultato socialmente efficiente.

In aggiunta ai ritorni crescenti e all'affermazione di nuovi standard, si manifesta il cosiddetto fenomeno del lock-in.

Si produce un effetto di lock-in quando l'utilizzatore di un certo prodotto o tecnologia, sostiene costi elevati di cambiamento tali da scoraggiare il passaggio ad un prodotto della concorrenza.

Di solito le scelte di adozione di uno standard da parte di un'azienda o di un consumatore dipendono molto dal raggiungimento dell'affermazione di uno standard, rispetto a quelli della concorrenza: anche se l'affermazione di uno standard è quasi sempre sinonimo di monopolio.

Tale contesto provoca delle barriere all'entrata per nuovi standard, non solo nel senso di guerra di prezzi la cui immediata conseguenza è un aumento di prezzi, tipico di situazioni monopolistiche, ma provocano grandi limiti soprattutto in termini di innovazioni, in quanto queste tendono a dipendere sempre dagli stessi monopolisti.

Infatti in questo caso l'innovazione si produce in un sistema prettamente chiuso, che difende standard e proprietà intellettuali relative, favorendo uno sviluppo molto rallentato dell'innovazione stessa.

Nel caso di un sistema aperto l'innovazione è favorita dalla possibilità di potervi accedere, da parte di chiunque, per arricchirla correggerla e permettere quindi un rapido sviluppo dell'innovazione stessa e una rapida diffusione.

Il modello di sviluppo Open Source, oltre a garantire prodotti di più alta qualità, grazie al suo sistema aperto, permette un arricchimento continuo dei

prodotti stessi, cioè una continua innovazione di prodotto, ma soprattutto una rapida diffusione dell'innovazione stessa.

D'altro canto, se un utilizzatore deve valutare i costi associati all'utilizzo di un prodotto, se da una parte ci sono costi di cambiamento legati ai costi di apprendimento dei nuovi prodotti, dall'altra i costi di possesso e di aggiornamento, sono molto bassi per effetto della libera diffusione del prodotto.

Questo è il caso del modello Open Source che si riesce a diffondere in presenza di standard già esistenti, favorendo soluzioni di maggior valore del prodotto e spostando la competitività unicamente sulla qualità e ,in misura minima, sul prezzo.

Il lock-in visto in precedenza, in questo caso, è basato sul valore attribuito al prodotto, e poiché nel caso Open Source il prodotto è realizzato con un modello prettamente diverso da quello standard, si potrebbe avanzare l'ipotesi di un lock-in ancora più forte: si potrebbe parlare non più di lock-in di prodotto, ma bensì di lock-in di sistema.

Tale effetto di lock-in potrebbe rendere estremamente difficile il ritorno a standard chiusi in quanto non si parla più semplicemente di prodotto, ma bensì di sistema di produzione.

A questo punto ho cercato di analizzare meglio come si sviluppa il sistema produttivo comunitario Open Source e quindi quali cambiamenti ha apportato anche dal punto di vista di sistema produttivo.

Come già accennato precedentemente , il sistema produttivo Open Source, è basato sulla cooperazione volontaria di soggetti eterogenei e indipendenti: da questo si deduce l'importanza che da la comunità Open alle risorse

umane, ovvero alle conoscenze ed alle competenze che ogni soggetto della comunità possiede.

Eric Rayon, uno dei pionieri del movimento Open, nella sua opera, “La Cattedrale e il Bazar”, fa un paragone tra i due sistemi di produzione, quello close e quello open, e associa il primo ad una cattedrale, cioè un metodo tradizionale di sviluppo, in cui i gruppi di lavoro sono di piccole dimensioni, sanno dall’inizio come sviluppare il prodotto e i feed-back, sono ricercati solo alla fine del lavoro.

Il secondo, lo paragona ad un Bazar, cioè ad un metodo di produzione in cui chiunque ha una buona idea può contribuire a rimediare alle imperfezioni o aggiungere nuove funzionalità interessanti, e in cui i feed-back, sono ricercati continuamente.

Questo sistema produttivo tipo Bazar, potrebbe apparire disordinato e caotico: ma in realtà non è così.

Alcuni protagonisti hanno ripreso il concetto di ordine sparso, che va contro la classica struttura gerarchica aziendale, e che sfocia poi in una variabile che caratterizza il sistema Open: la modularità.

Infatti i progetti Open Source, vengono divisi in sottoprogetti, e le varie funzionalità si sviluppano in modo incrementale.

La comunità Open Source è capace di sfruttare l’esecuzione parallela delle attività di sviluppo, allo scopo di velocizzare il processo di convergenza verso la soluzione migliore, e la modularità è indispensabile per lo sviluppo parallelo e decentralizzato dei progetti perché facilita il controllo e l’introduzione di nuove parti di codice nel progetto.

Anche questa modalità di progettazione è possibile solo in caso di sistemi aperti.

Ma a questo punto non riesco a spiegarmi come si potesse reggere questo nuovo modello di business sulla base del volontariato e della cooperazione.

Non riesco a spiegarmi, come fosse possibile che questo nuovo modello, in quanto business, oltretutto all'interno di un settore dominato da forte concorrenza e dalla rincorsa ai massimi profitti, non avesse anch'esso bisogno di appoggiarsi a quelle che sono le regole di mercato e di marketing, fatte di concorrenza competizione e di ricerca di utile.

Continuando le ricerche ho scoperto che attorno a questo nuovo modello di business, si è costruita una vera e propria Open Source Economy in contrapposizione alla Close Economy dei produttori di software proprietari.

Ho paragonato queste due tipologie di economie e ho dedotto che, a parte le differenze di scala, le due economie differiscono anche a livello di obiettivi, pratiche ed eticità, conservando però un tratto in comune che è quello di svilupparsi entrambe in un mercato di tipo capitalistico.

I caratteri salienti della Open Source Economy, sono sia la metodologia, il permesso d'autore il codice aperto e la cooperazione: di fatto per sviluppare un prodotto software libero, non occorre denaro, bensì tempo libero messo a disposizione da parte dei programmatori e volontà di cooperare.

Ma la cosa più interessante è stata la nascita di nuovi modelli aziendali, nati attorno al nuovo paradigma Open Source.

Si parla infatti di soggetti aziendali Pure Player o ibridi, che sono aziende che sviluppano il loro business attorno al modello Open Source, offrendo interfacce tecniche assistenza e risoluzione dei problemi, dietro pagamento.

Un esempio di azienda Pure Player è Red Hat : il valore aggiunto che Red Hat commercializza è quello di confezionare il prodotto Linux venderlo come sicuro e affidabile grazie ad accurate verifiche di compatibilità e ai controlli dei codici, dimostrando così di saper generare profitti elevati anche solo basandosi sulla distribuzione, spostandosi così da un business di prodotto ad un business di servizio.

Inoltre il confezionamento e la distribuzione del prodotto Linux, il controllo dei codici e della loro compatibilità, aggiungono valore al prodotto e lo rendono più sicuro efficiente e completo, garantendo alle imprese Pure Player la sopravvivenza e la possibilità di continuare a sostenere lo sviluppo dei prodotti Open e del proprio business.

Solo una parte dei profitti proviene dalla vendita diretta dei pacchetti software e il valore di questi pacchetti non è costituito da singoli prodotti distribuiti, ma dal contributo in termini di integrazione e confezionamento.

Il prezzo dei prodotti distribuiti da Red Hat risulta di conseguenza molto inferiore rispetto al prezzo di altri pacchetti software costruiti intorno a prodotti proprietari: Red Hat infatti ha la possibilità di condividere i costi di produzione del software con la comunità di sviluppo.

D'altro canto chiunque può scaricare copiare e distribuire gratuitamente i prodotti Red Hat coperti da licenze copyleft, per cui è chiaro che i profitti dell'azienda derivano da tutti i servizi aggiunti che l'impresa è in grado di offrire.

E' quindi la commercializzazioni di questi valori aggiunti che permette alle aziende Pure Player di sopravvivere e sostenere con continuità lo sviluppo di Linux.

Inoltre mentre i Pure Player basano il loro business sulla commercializzazione di valori aggiunti, creati da loro stessi su un prodotto già esistente, i modelli di business ibridi, produttori di sistemi lo distribuiscono gratuitamente e ricavano profitti facendo pagare costi di consulenza manutenzione e aggiornamento.

Questa strategia dei modelli di business 'ibridi' è perseguita da imprese che non possiedono uno standard proprietario che domina il mercato, quindi imprese che non hanno il problema della "cannibalizzazione" dei propri prodotti, anzi cercano di utilizzare l'Open per vendere prodotti complementari (hardware e software).

Un esempio di modello di business ibrido è il caso di IBM, che ormai da tempo supporta con convinzione il processo di diffusione del sistema operativo Linux, producendo per la maggior parte, macchine Linux compatibili.

IBM, decisamente coinvolta nel settore e-business, ha intuito la possibilità di ricavare profitti, non più vendendo sistemi equipaggiati da software proprietari, ma offrendo servizi di consulenza assistenza e attività di formazione richieste da aziende sempre più interessate a prodotti affidabili e a basso costo per la gestione delle infrastrutture informatiche.

E' ancora una volta il caso di Linux che essendo una soluzione qualitativamente valida viene adottato ormai da moltissime imprese e ciò

comporta una crescente richiesta di servizi e assistenza, da parte delle aziende stesse.

Queste situazioni mettono in evidenza come il software libero non solo stia mettendo in discussione quello proprietario, ma addirittura si ipotizza la possibilità di un nuovo equilibrio di coesistenza competitiva.

Inoltre alcuni elementi fanno ritenere che il paradigma proprietario debba progressivamente convergere verso una accettazione della necessità di rendere libere almeno parti di codice sorgente, entrando di fatto dentro un modello di business ibrido.

Interessanti inoltre sono i diversi metodi di coinvolgimento da parte della comunità: un esempio potrebbe essere per esempio le Gare a progetto.

La comunità mette a disposizione un premio in denaro, per chi tra i partecipanti della comunità alla gara, risolve prima problemi o difetti tecnici, o per chi sviluppa prima parti di codice un po' noiose da creare.

Da qui deriva anche il concetto di autoformazione: di fatto la possibilità di prendere parte ad una comunità di sviluppo ed alle sue gare, permette a chiunque vi partecipi di acquisire una formazione a 360 gradi e soprattutto gratuita, senza dover sottostare alle aziende che spesso formano, non solo sotto pagamento, ma soprattutto secondo le loro esigenze, lasciando così al personale una formazione settoriale, limitata e non generica.

Un altro argomento a mio avviso interessante da analizzare nel panorama Open è l'applicazione del nuovo modello nella Pubblica Amministrazione. Di fatto questa copre un ruolo importante non solo dal punto di vista economico, in quanto la PA utilizza molti software per la gestione di informazioni, e quindi adottare software prettamente Open non solo



diminuirebbe la spesa pubblica, ma allo stesso tempo influenzerebbe le scelte di adozione di software libero per i cittadini e per le aziende, offrendo inoltre ai cittadini stessi un servizio di informazione indipendente dalle strutture informatiche, e dalle scelte dei fornitori.

Inoltre l'adozione di prodotti Open nelle PA si ricollega a tematiche sociali, quali il tema della circolazione del sapere, della libertà di divulgazione scientifica, dei risultati della ricerca e al dibattito sulle questioni della tutela del diritto d'autore.

E' inoltre importante la diffusione dell'informazione presso i cittadini, relativa alla circolazione di documenti o dati, perché essa sviluppa implicazioni diffuse: in particolare sul tema dei formati aperti, essa è destinata ad avere un impatto sul rapporto tra PA e cittadini, stimolando così la cultura della condivisione.

Ritornando poi al tema circa l'importanza dell' applicazione delle leggi economiche per un benessere collettivo, ho trovato interessante analizzare le diverse applicazioni della filosofia Open nei campi no software.

Infatti grazie al concetto di cooperazione e condivisione, sono nati prodotti del tipo Wikipedia, che hanno come unico scopo quello di diffondere informazioni e conoscenze, grazie all'impegno di chiunque di alimentare questa enciclopedia telematica, che ormai supporta più di 500 mila articoli, in tutte le lingue, fino a comprendere persino l'Esperanto.

Con questa stessa formula sono nati siti di Open music, in cui si mette a disposizione di tutti musica da scaricare e copiare, siti Open press: di fatto molte case editrici italiane di libri informatici danno la possibilità a chiunque di poter scaricare interamente dalla rete libri di informatica.

Ma i campi di applicazioni della filosofia Open nei settori no software più interessanti da analizzare sono il settore biotecnologico e quello farmaceutico.

Grazie all'idea di apertura Open, sono stati creati modalità di sviluppo di farmaci specifici e biotecnologie, che si basano sulla collaborazione di persone che hanno messo a disposizione i loro casi per analizzarli meglio, per trovare cure specifiche, che andassero contro le economie di scala delle produzioni di farmaci generici delle multinazionali farmaceutiche, che come dimostrato sono state causa di diversi casi di morte.

Lo sviluppo di questi nuovi strumenti nel campo scientifico, è ancora in via di sviluppo, perché il settore biotecnologico e farmaceutico, oltre ad essere un settore chiuso da proprietà intellettuali, è un settore che richiede competenze molto più specifiche, per cui chi vi può apportare contributi, sono solo persone professionalmente preparate.

Nonostante il lento sviluppo, questi nuovi strumenti promettono e rappresentano una promessa: aspetteremo gli sviluppi per analizzarli meglio.

In conclusione, dopo le varie analisi dal punto di vista economico, sociale, ed anche scientifico, posso affermare che la filosofia della cooperazione, della collettività, e del dominio pubblico, è uno strumento che oltre ad avere la capacità di apportare profitti nel modo migliore, in senso stretto, apporta anche progresso e sviluppo dal punto di vista sociale e umano.

Per cui affermo che l'Open Source, non è un assurdo economico.

## **Bibliografia**

**\*Aliprandi Simone (2005) “Copyleft & opencontent”:**

**l’altra faccia del copyright.**

**\*Arthur W. B. (1994) “Competitive Technologies,**

**Increasing Returns, and Lock-in by Historical Events” .**

**\*Apogeo (2001) “Codice Ribelle”**

**\* Beccaria, Marchetti, Stasi (2004) “Noscopyright”.**

**\*Dalle, Julien (1999) “Linux or some explanation into economics of free software.**

**\*Di Bona, Ockman, Stone (1999) “Open Source”.**

**\*David P. (1985) “CLIO and the economics Qwerty”**

**\*Hardin R. (1982) “Collective Action” University press of Baltimora**

**\*Ippolita (2005) “Open non è Free”**

## **Comunità digitali tra etica hacker e mercato globale**

**\*Katz M. , Shapiro (1985) “Esternalità di network, competizioni e compatibilità”.**

**\*Martin Fink (2003) “Modelli di business per Linux e Open Source” il nuovo vantaggio competitivo:scenari strategici e operativi.**

**\*Muffatto Moreno e Caldani Matteo (2004) “OPEN SOURCE”**

**Strategie, Organizzazioni, Prospettive.**

**\*Peter Wayner (2000) “FREE FOR ALL”.**

**\*POSSE (2004) Politica Filosofia Moltitudini.**

**\*Università di Pavia (2004) “HIGHT-TECH Marketing: lock-in”: strumenti operative, strategie e culture manageriali.**

**\*Rivista “Science” .**

**\*Rivista “Journal of Human Genetics”.**

**\*Salvatore Romagnolo, Alberto Mari (2001) “Revolution Open Source”**

**\*Salvatore Romagnolo (2000) “Red Hat sarà il prossimo gigante del software?”**

**\*Sam Williams (2000) “Codice LIbero”**

## **SITI INTERNET**

**punto-informatico.org**

**pinospazio.altervista.org/forum**

**[www.lavoce.itnfo](http://www.lavoce.itnfo)**

**[www.javaopenbusiness.it](http://www.javaopenbusiness.it)**

**[www.zeusnews.it](http://www.zeusnews.it)**

**[www.wikipedia.it](http://www.wikipedia.it)**

**[www.lem.sssup.it](http://www.lem.sssup.it)**

**[www.apogeonline.it](http://www.apogeonline.it)**

[www.cambia.org](http://www.cambia.org)

[www.creativecommons.org](http://www.creativecommons.org)

[www.bioforge.org](http://www.bioforge.org)

[www.bios.ne](http://www.bios.ne)

[www.nature.org](http://www.nature.org)

[www.wrenmedia.co.uk](http://www.wrenmedia.co.uk)

[www.wikipedia.org](http://www.wikipedia.org)

## **Ringraziamenti**

Un ringraziamento particolare ai miei genitori che mi hanno dato la possibilità di vivere questa esperienza di studio, alla prof. Vincenza Odorici, che mi ha seguito nella stesura della tesi, mi ha incoraggiato e soprattutto ha cercato di conoscermi, prima di tutto dal punto di vista umano.

Un ringraziamento anche a Afrika, Michele, Elena e Teddy, al mio grande amico Luca, Angelo&Laura, Bambina, Francesca, Super Andy, Marco, e a tutti coloro con cui sono cresciuta, condividendo questi cinque anni di vita.

Ahul!!!!