

LUISS
Libera Università degli Studi Sociali
Guido Carli

Facoltà di Economia
Cattedra di Economia e Gestione delle Imprese

Il nuovo mondo delle imprese: l'e-Business e le Open Source

RELATORE

Prof. Paolo Boccardelli

CANDIDATO

Fabio Franza

A Paolo, Linda, Francesca e Laura

INDICE

Introduzione	5
Capitolo 1: Internet e l'impresa	6
1.1 Cos'è l'e-business?	7
1.2 La storia di Internet	8
1.3 La new economy	10
1.4 Fattori critici di successo e insuccesso	12
1.5 Effetti dell'e-business all'interno di un impresa	14
1.6 Effetti dell'e-business sui mercati	15
Capitolo 2: Il fenomeno Open Source	19
2.1 Definizione di open source	19
2.2 Storia delle open source	22
2.3 Open source e software libero	26
2.4 Open source e software proprietario	29
2.5 Legislazione del software open source	32

Capitolo 3: Un nuovo paradigma economico	42
3.1 La risorsa open source	44
3.2 Fattori critici di successo nel mondo open source	48
3.3 Relationship tra imprese e comunità OSS	49
3.4 Strumenti di gestione per il controllo delle relazioni con la comunità	57
3.5 I modelli di business open source	59
Conclusioni e previsioni	67
Bibliografia	71
Sitografia	73

Introduzione

La scelta dell'argomento del presente lavoro è frutto della voglia di conoscere le funzionalità di un mercato in continuo sviluppo e dell'istinto imprenditoriale nel ricercare nuovi mercati profittevoli.

In questa sede si vuole dare un punto di osservazione e monitoraggio del mercato italiano dell'e-business, e in particolare delle Open Source, al fine di fornire una dettagliata fotografia della situazione attuale, analizzandone le traiettorie evolutive e i fattori che ne sosterranno lo sviluppo.

Rilevante importanza ha la crescente espansione di Internet nella vita quotidiana dei consumatori, soprattutto nelle fasce più giovani e quelle comprese tra i 25 e 40 anni; tale fenomeno ha avuto un impatto molto forte su abitudini di consumo e stili di vita delle persone e conseguentemente nell'economia.

Le prospettive di crescita dell'e-business sono di rilevante importanza, infatti si stanno interessando, sempre di più, le aziende in cerca di visibilità, sia per quanto riguarda gli investimenti pubblicitari che per quello che concerne tutti gli altri servizi messi a disposizione da Internet, come il rapporto diretto tra impresa e cliente tramite il sito web.

Il lavoro è strutturato in 3 parti differenti. Nella prima parte si presenterà l'e-business, dove si concentrano tante attività che le imprese possono intraprendere grazie alla tecnologia Internet; in seguito, nella seconda parte, si presenterà il fenomeno Open Source e, nella terza parte, si approfondirà il discorso con temi importanti e centrali del lavoro.

Capitolo 1

Internet e l'impresa

La vera forza di internet è rappresentata dalle varie funzionalità che essa possiede e dalle differenti funzioni d'uso di cui un'impresa può usufruire nell'investire in questo mercato. Questa tecnologia di comunicazione permette di sviluppare processi aziendali in un modo più efficace ed efficiente e di creare nuove opportunità di business. Inoltre, il bacino di utenti internet è in costante crescita e di notevole importanza, fattore molto attrattivo per gli investitori.

E' attualmente riconosciuto che le nuove tecnologie, in particolare l'accesso ad Internet, tendono a modificare la comunicazione tra i diversi attori professionali mondiali, soprattutto :

- ✚ le relazioni tra l'azienda e i suoi clienti,
- ✚ il funzionamento interno dell'azienda, comprese le relazioni azienda-impiegati,
- ✚ la relazione dell'azienda con i suoi differenti partner e fornitori.

Viene quindi detto “*e-business*” l'integrazione all'interno dell'azienda di strumenti basati sulle tecnologie dell'informazione e della comunicazione (si parla generalmente di prosoftware) per migliorarne il funzionamento per creare del valore per l'azienda stessa, per i suoi clienti e per i suoi partner.

L'e-business non si applica quindi solo alle aziende virtuali (dette *click and mortar*) che fondano essenzialmente la loro attività sul Net ma anche alle aziende tradizionali (dette *brick and mortar*).¹

¹ Per un approfondimento si veda il sito <http://www.Kioskea.net>.

1.1 Cos'è l'e-business?

Il business elettronico è senz'altro uno dei fenomeni maggiormente dirompenti degli ultimi decenni, con ripercussioni notevoli non solo in termini di organizzazione dei processi aziendali, sempre più velocizzati e digitalizzati, ma anche in termini di approccio al cliente. Quest'ultimo è sempre più visto come punto di inizio e punto di arrivo dei processi, considerando che la vera ricchezza degli scenari competitivi non risiede più solo nei prodotti ma nella capacità di gestione dei propri clienti.

L'impressionante velocità di crescita della “*Rete delle reti*”, con i suoi nuovi strumenti di comunicazione ed interazione, impone a tutti, ed in particolare alle aziende, un'attenta analisi di come nuove applicazioni e nuovi servizi possano essere utili allo sviluppo delle proprie attività.

La nozione di e-Business è comunque molto flessibile e ingloba tutte le utilizzazioni possibili delle tecnologie dell'informazione e della comunicazione (ICT) per qualsiasi o una parte delle attività seguenti :

- ✚ Rendere più efficaci le relazioni aziendali con i suoi clienti e con i differenti partner (fornitori, amministratori, ecc.)
- ✚ Sviluppare delle nuove opportunità d'affari
- ✚ Fluidificare la circolazione dell'informazione interna
- ✚ Mettere sotto controllo i diversi processi aziendali (produzione, stock, acquisti, vendita, risorse umane, ecc.)

Si tratta quindi di nuovi strumenti che permettono alle imprese di gestire in maniera più efficiente ed efficace i processi aziendali, le relazioni con i fornitori, le relazioni con i clienti (e-marketing e CRM), le scorte di magazzino, la gestione della contabilità e in generale la propria attività.

1.2 La storia di internet

Nel 1958 il Governo degli Stati Uniti decise di creare un istituto di ricerca orientato al settore “Comando e Controllo della Difesa”. L’istituto venne denominato *ARPA* (acronimo di *Advanced Research Projects Agency*) e il suo compito era ambizioso: cercare una soluzione alle problematiche legate alla sicurezza nella rete di telecomunicazioni militari.²

Alla fine del 1969, quattro computer “ospiti” vengono connessi tra di loro in quella che è l’inizio di *ARPANET*, l’appalto per la costruzione di questa rete fu concesso a una società chiamata Bolt, Beranak and Newman (BBN) che collegò quattro università diverse: Stanford University, UCLA (University of California at Los Angeles), UCSB (University of California at Santa Barbara) e la University of Utah, usando linee telefoniche e installando in ciascuna di queste un *IMP* (*Information Message Processor*), vale a dire un particolare computer che gestiva il traffico in rete. Quest’ultimo fungeva da intermediario tra linee di connessione e mainframe, i grandi elaboratori centralizzati su cui all’epoca risidevano tutte le informazioni e tutti i programmi. Tutto il traffico che viaggiava su queste connessioni non era confidenziale e serviva prevalentemente a titolo di ricerca e sperimentazione. All’epoca i sistemi per trasmettere messaggi da una località all’altra non erano molto efficaci e il primo obiettivo dei ricercatori fu proprio quello di trovare soluzioni funzionali per convogliare i pacchetti su *ARPANET*. Decisero di adottare un modello il cui sviluppo era già iniziato in Europa (per opera del National Physics Lab inglese e della Società Internationale de Télécommunications Aeronatiques francese) e che sarebbe poi diventato comune in molti altri sistemi di comunicazione: la *commutazione di pacchetto*. Mediante questa tecnica, i messaggi e le informazioni sono suddivisi in pacchetti di lunghezza fissa e ogni singolo pacchetto diventa un’entità a sé stante, capace di viaggiare sulla rete in modo completamente autonomo perché dotata al proprio interno dell’indirizzo sia di provenienza che di destinazione. Non è importante che tutti i pacchetti che compongono un determinato messaggio rimangano uniti durante il percorso e non è nemmeno indispensabile che arrivino nella sequenza

² Tratto da dispense del 29/02/2008 del Prof. Balestreri docente dell’Università di Bologna.

giusta. Le informazioni che essi convogliano al proprio interno sono sufficienti per ricostruire, una volta arrivati a destinazione, l'esatto messaggio originale, indipendentemente dal percorso seguito da ciascuno dei suoi frammenti.

Grazie a questo sistema si ottengono due benefici immediati: qualunque sia lo stato della rete, il pacchetto può sempre trovare una via alternativa per giungere alla propria destinazione (requisito utile per gli obiettivi militari e per chiunque desideri avere un impianto il più possibile resistente ai guasti, anche a quelli accidentali). Inoltre i vari pacchetti provenienti da fonti diverse possono essere convogliati tutti assieme su una singola linea ad alta velocità anziché dover ricorrere a tante linee separate, usate solo parzialmente. Si riesce in questo modo a condensare il traffico su una linea collegata in permanenza che ripartisce dinamicamente la propria capienza tra i vari computer collegati.

Nel dicembre del 1970 il Network Working Group (NWG) completa il protocollo iniziale Host-to-Host di ARPANET, chiamato *Network Control Protocol* (NCP). Dopo che i siti ARPANET sono stati completamente implementati con NCP nel periodo 1971-1972, gli utilizzatori della rete possono finalmente cominciare a sviluppare le applicazioni.³

Per tutti gli anni settanta ARPANET continuò a svilupparsi in ambito universitario e governativo, ma dal 1974, con l'avvento dello standard di trasmissione TCP/IP, il progetto della rete prese ad essere denominato Internet.

È negli anni ottanta, grazie all'avvento dei personal computer, che un primo grande impulso alla diffusione della rete al di fuori degli ambiti più istituzionali e accademici ebbe il suo successo dichiarato, rendendo di fatto potenzialmente collegabili centinaia di migliaia di utenti. Nacque allora l'idea di rendere pubblico l'esercizio, dopo gli anni ottanta, i membri della comunità scientifica iniziarono a scambiarsi informazioni, dati e anche messaggi estemporanei.

Un'importante spinta per la diffusione di questa tecnologia si ha dopo l'avvento dell'interfaccia grafica, che facilitandone l'uso e la navigazione ha contribuito a far diventare Internet uno strumento di massa, aperto alla divulgazione di notizie e alla vendita di prodotti e servizi.

³ <http://www.windoweb.it>

Inoltre, possiamo affermare che la diffusione di Internet ha determinato un progressivo mutamento nei paradigmi di creazione del valore e nei modelli organizzativi: facendo nascere così la cosiddetta “*new economy*”, che si distingue dalla “*old economy*” per il primato dei beni intangibili sui beni materiali, e per un tasso di crescita esponenziale favorito dal ricorso alla tecnologia.

1.3 La new economy

Negli ultimi dieci anni il mondo degli affari e delle imprese è cambiato profondamente. I cambiamenti radicali, che stanno mutando non solo il mondo economico, ma anche la società e le sue strutture possono essere affrontati secondo un triplice aspetto: possono essere ignorati, accettati e subito passivamente, oppure si può cercare di comprenderli. Il cercare di comprenderli è, ovviamente, più complesso rispetto agli altri due atteggiamenti alternativi. E' necessario cercare di far luce in particolar modo sulla situazione attuale e sugli scenari futuri verso cui potremmo andare incontro per quel che concerne le mutazioni riguardanti il mondo delle imprese.

Tuttora l'Italia vive un periodo di transazione in cui molte potenzialità dell'e-business non sono sfruttate o, ancora peggio, non sono ben percepite dagli imprenditori. Un indagine condotta nell'ultimo trimestre 2008 dall'Istituto di Ricerca Sinaptica⁴ ha evidenziato come le aziende italiane hanno una conoscenza molto frammentata e poco approfondita degli strumenti di e-business e, in particolar modo, dei benefici che possono generare sotto il profilo della conquista di nuovi mercati, di nuove modalità di dialogo con i propri consumatori o dei metodi per ottimizzare la gestione ed i costi connessi allo svolgimento delle attività.

Per il 98% del campione, rappresentato da 120 tra top manager ed imprenditori di primarie aziende italiane, fare e-business significa principalmente

⁴ La ricerca è stata svolta con metodo CAWI (Computer Assisted Web Interviewing).

avere un sito aziendale. A gran parte del campione sfugge l'ampio spettro di strumenti offerto dall'e-business: dall'intranet all'extranet, dall'e-commerce all'e-procurement, ai nuovi media digitali. Tra le motivazioni che frenano gli investimenti in e-business nelle imprese, e che sono alla base della scarsa propensione del management aziendale all'utilizzo degli strumenti di e-business, c'è al primo posto la mancanza di una cultura e di competenze interne specifiche, ma anche di interlocutori esterni adeguati. Dall'indagine risulta inoltre che l'investimento delle aziende italiane in strumenti di e-business è oggi ancora a livelli estremamente bassi, pari a meno dell'1% del fatturato aziendale. Sono le grandi aziende italiane e le filiali in Italia di multinazionali le più sensibili a tali investimenti perché ne conoscono le potenzialità grazie anche ad esperienze "importate" dall'estero. In particolare: *"L'investimento annuo si aggira in media sui 38 mila euro per le aziende con un fatturato di 50 milioni di euro, mentre l'investimento sale a 500 mila euro per le aziende medie, dai 50 ai 300 milioni di euro, e supera i 2 milioni di euro per le grandi aziende con fatturato oltre i 300 milioni di euro"*.

Un'ulteriore fenomeno rilevato dall'Osservatorio è l'insoddisfazione per i servizi offerti dai fornitori di soluzioni di e-business: solo il 37% delle aziende intervistate si dichiara "molto o abbastanza soddisfatto" dei servizi prestati dalle web agencies e il 26% per le società di consulenza. Molto bassa la soddisfazione per i servizi di e-business offerti dalle agenzie di pubblicità: solo il 15%.

I motivi di questa insoddisfazione sono da ricercarsi in una mancanza di *"contributo alla sviluppo della strategia aziendale, scarsa propositività, insufficiente assistenza dopo il lancio"* ma anche nella *"creatività"* delle soluzioni sviluppate. Ed è proprio questa mancanza di una valida collaborazione esterna unita alla mancanza di competenze aziendali l'ostacolo principale all'incremento dell'utilizzo degli strumenti di e-business da parte delle aziende italiane.

Gli scenari futuri sono dipendenti, quindi, dalla diffusione della conoscenza sugli strumenti che l'e-business mette a disposizione delle imprese. Informazioni confortanti derivano dall'indagine almeno su questo argomento, infatti, la maggioranza del campione afferma che nei prossimi anni dovrà aumentare la voce

di spesa relativa all'e-business e che ciò comporterà un impatto positivo sui risultati dell'azienda.⁵

Questo sta a dimostrare come la situazione attuale italiana è alle fasi iniziali di questo nuovo mondo chiamato e-business.

1.4 Fattori critici di successo e insuccesso

Mostriamo un caso pratico per comprendere più a fondo come si possono sfruttare meglio determinate potenzialità: una recente ricerca commissionata da Google e condotta da Synovate in Gran Bretagna, Stati Uniti e Germania, con interviste su un campione di 1.589 adulti tra i 16 e i 64 anni, ha prodotto risultati interessanti che sono stati presentati e analizzati durante la manifestazione "*Web in Tourism*" a Milano. Il dato più evidente emerso dallo studio individua in internet la principale fonte d'informazione nel campo del turismo ed evidenzia l'importante ruolo che i motori di ricerca giocano nell'organizzazione di una vacanza e nella scelta della destinazione. I dati dimostrano che l'Italia, pur essendo al secondo posto in termini di considerazione/visite, non sfrutta al massimo il suo potenziale.

Il primo fattore da analizzare come causa è la scarsa comunicazione on-line dell'offerta turistica. L'Italia è percepita all'estero con il cliché della città d'arte o dell'enogastronomia, sottostimando il potenziale turistico dell'offerta mare. Infatti, in una ricerca generica condotta in internet come "*mare*" otterremo molti risultati di luoghi balneari spagnoli a discapito di quelli italiani.

Un altro elemento importante da tenere in considerazione è legato alla tempistica. Quasi la metà del campione naviga in rete alla ricerca d'informazioni almeno tre mesi prima della partenza, l'offerta italiana è debole proprio nella

⁵ <http://www.ebitinnovation.com> – Primo osservatorio italiano sull'e-business.

prima fase della ricerca, dove l'utente vuole informazioni a livello generale. Se la proposta turistica non è fruibile nei tempi giusti, ne deriva un forte tasso di perdita di efficienza italiana, a vantaggio di Spagna e Francia.⁶

Per spiegare quali sono i fattori critici di successo dobbiamo prima capire che l'obiettivo di un progetto e-business è di creare del valore e questo scopo si può raggiungere attraverso diversi modi:



Grazie ad un aumento dei margini, cioè un abbassamento dei costi di produzione o un aumento dei benefici. L'e-business permette di arrivarci attraverso i modi seguenti :

- posizionamento su nuovi mercati
- miglioramento della qualità dei suoi prodotti o servizi
- prospezione di nuovi clienti
- fidelizzazione della clientela
- razionalizzazione del suo funzionamento interno.



Grazie alla motivazione del personale. Il passaggio da un'attività tradizionale ad un'attività di e-business permette idealmente di motivare i collaboratori nella misura in cui:

- la strategia globale è maggiormente visibile degli impiegati e favorisce una cultura comune
- il modo di funzionamento implica una responsabilizzazione degli attori
- il lavoro in comune è propizio al miglioramento delle competenze.



Grazie alla soddisfazione dei clienti. L'e-business favorisce in effetti :

- un abbassamento dei prezzi legato all'aumento della produttività

⁶ Fabbricini, D. (2008). Il Web 2.0 il vero motore del turismo online. DailyNet

- un migliore ascolto del cliente
- prodotti e servizi adatti ai bisogni dei clienti
- un modo di funzionamento trasparente per l'utente.

✚ *Grazie alle relazioni privilegiate con i partner.* La messa in opera di canali di comunicazione con i fornitori permette :

- una migliore conoscenza reciproca
- un'accresciuta reattività
- delle migliori capacità di anticipazione
- una condivisione di risorse vantaggiosa per entrambe le parti.

Un progetto e-business non può che funzionare dal momento in cui apporta un valore aggiunto all'azienda, ma anche al suo personale, ai suoi clienti e ai suoi partner.⁷

1.5 Effetti dell'e-business all'interno di un impresa

L'economia va incontro a un continuo processo di “*dematerializzazione*” contraddistinto dal primato del know-how sul capitale, che porta con sé due principali conseguenze: una riduzione dei costi, che si traduce in benessere per la società nel suo complesso, e la globalizzazione del mercato, favorita dalla facilità con cui le informazioni, i prodotti e i servizi riescono a circolare rispetto ai beni fisici.

Il calo dei costi e la globalizzazione del mercato implica una spinta concorrenziale senza precedenti e una graduale diminuzione di importanza dei confini geografici. In un mercato dominato dalle caratteristiche dell'ipercompetizione, l'azienda si afferma grazie all'originalità e alle capacità

⁷ <http://www.Kioskea.net>.

innovative: la differenziazione della propria offerta, la diversificazione tra i prodotti, la presenza in aree geografiche nuove e distanti.

L'impresa si accorge di un fatto fondamentale: le moderne tecnologie la mettono in rapida e diretta competizione con un numero di concorrenti estremamente più elevato che in passato. Sul fronte della domanda, la conseguenza diretta è che i consumatori sviluppano una conoscenza più approfondita e dettagliata dell'offerta, la sfruttano a proprio vantaggio e manifestano esigenze sempre più particolareggiate. Il consumatore così acquisisce maggiore potere e controllo. Quindi, un'impresa che opera oggi in queste condizioni deve fare maggiore attenzione alla *customer satisfaction* e deve cercare di attuare una strategia di *differenziazione* per rendere il prodotto unico agli occhi dei consumatori, oppure deve avere i prezzi più bassi del mercato attuando delle strategie che tendono alla *leadership di costo*.

1.6 Effetti dell'e-business sui mercati

Ogni impresa oramai deve confrontarsi con nuovi mercati e le variabili che hanno portato a questi cambiamenti sono vari:

-  globalizzazione
-  virtualizzazione
-  tecnologia
-  customizzazione
-  informazione
-  comunicazione
-  velocizzazione dei processi aziendali.

La *globalizzazione* è un fenomeno che investe tutte le imprese. Essa si traduce nella possibilità di identificare segmenti transnazionali i cui bisogni divengono il riferimento per la definizione dell'offerta e delle politiche di

innovazione; significa quindi, in ultima analisi, opportunità di ampliare la "*rete di relazioni*" sul fronte della domanda; ma significa anche possibilità di approvvigionarsi secondo logiche di spinta verso l'efficienza ed anche la minaccia di dover sostenere il confronto con un numero maggiore di concorrenti, spesso difficili da identificare ed analizzare. Affrontare la globalizzazione richiede molte risorse e genera, ad esempio, una crescita consistente del bisogno di informazione proveniente dal mercato, di rapporti a livello distributivo, di sistemi logistici, di capacità di customizzazione, di nuove competenze.

La *virtualizzazione* dei mercati costituisce oggi la spinta più forte verso il completamento del processo di globalizzazione. La rilevante accelerazione dei processi competitivi, indotta dalla virtualizzazione, conduce al continuo miglioramento dell'efficienza ed efficacia dei processi aziendali. Le caratteristiche di funzionamento dei mercati virtuali impongono un ripensamento alle logiche di gestione del rapporto impresa/domanda, ad esempio verso lo sviluppo di strategie di "marketing individuale", di personalizzazione dell'offerta, di nuove modalità di analisi e monitoraggio del cliente, il quale sta recuperando potere nei confronti del venditore.

La *tecnologia* informatica ed elettronica ha consentito alle aziende di ampliare notevolmente la propria gamma di offerta superando il trade-off costo/differenziazione; inoltre ha permesso alle imprese più dinamiche e flessibili, di sfruttare l'evoluzione della tecnologia (soprattutto quella informatica) per abbattere significativamente i costi di produzione. L'altro vantaggio è che ha praticamente eliminato i problemi di comunicazione attraverso lo sviluppo di Internet e delle reti interne; ha ridotto inoltre i tempi di quasi tutti i processi aziendali ed ha favorito incrementi notevoli nella produttività delle risorse umane. La tecnologia è forse la variabile che ha modificato in modo più incisivo la realtà di tutti i mercati e che oggi, con il business on-line, si ripropone come veicolo primario di cambiamento. La tecnologia impegna le imprese in uno sforzo di adeguamento dei processi aziendali, implica investimenti in potenziamento e riprogettazione della struttura, attività di riqualificazione delle risorse umane,

nella consapevolezza che tale "sforzo" dovrà essere continuo, per quanto incrementale.

La *customizzazione* è una strategia quasi obbligata per creare il vantaggio competitivo nei mercati globali. I consumatori chiedono prodotti ad hoc e le aziende devono saper generare offerte adattate in condizioni di alta efficienza, basandosi sui concetti del "*micro-marketing*", del "*marketing-individuale*", o anche della "*one-to-one relationship*", strategia fortemente potenziata dallo sviluppo di Internet. Anche se la tecnologia ne ha ridotto notevolmente i costi, la customizzazione richiede flessibilità organizzativa e nelle risorse umane, forti capacità di relazione con la domanda, abilità di progettazione, programmazione e riorganizzazione dei processi aziendali. Per molte aziende la customizzazione si associa ad un incremento della complessità e ad una riduzione dei tempi dei cicli di progettazione-produzione-consegna, che diviene un importante fattore critico di successo nella considerazione che può essere molto difficile prevedere quello che il mercato "potrà chiedere".

Globalizzazione, customizzazione e tecnologia rendono l'*informazione* la risorsa nodale intorno alla quale si gioca la competitività. In effetti, la produttività di un'impresa cresce nella misura in cui la stessa dispone di qualificate informazioni di mercato. Il bisogno di informazione delle aziende non aumenta solo in termini quantitativi, ma anche, ad esempio, in termini di frequenza: l'informazione, infatti, diviene rapidamente obsoleta in condizioni di forte dinamicità del mercato. Le aziende hanno così bisogno di rinnovare in tempi ridotti parte del proprio patrimonio informativo, soprattutto quello relativo all'ambiente: si pone dunque il problema, ad esempio, di individuare un sistema di strumenti per il monitoraggio di informazioni dal mercato che non potranno essere individuati negli strumenti tradizionali, come ad esempio le ricerche di mercato, ma utilizzando strumenti più evoluti come ad esempio la creazione di una architettura che sia in grado di rilevare le tendenze del mercato. L'informazione è una risorsa e nello stesso tempo assorbe risorse per essere disponibile in modi e tempi utili. Oramai il mezzo di comunicazione che si sta diffondendo, cioè

Internet, viene considerato come strumento di marketing intelligence: occorre ricordare, tuttavia, che difficilmente Internet può essere fonte delle informazioni "critiche", che fanno la differenza nell'impostazione generale delle politiche di mercato, o, meglio, Internet consente la raccolta dei dati: affinché questi siano trasformati in informazione, in strategia, in visione dal management occorre comunque la visione umana, basata sull'esperienza, la capacità di interpretazione, lo spirito imprenditoriale. Ciò non toglie che, nella misura in cui è strutturato e strutturabile, il Web può essere una validissima piattaforma di partenza per lo sviluppo di un rinnovato sistema di analisi di mercato. I condizionamenti che la complessità e la variabilità del mercato esercita sui processi di raccolta e analisi delle informazioni sono quindi diversi: dal problema dell'individuazione di valide fonti, alle modalità della loro continua rilevazione, alla selezione di informazioni rilevanti.

Se la *comunicazione* rappresenta il principale meccanismo di coordinamento dei processi aziendali e interaziendali ed è alla base del concetto stesso di interazione, gran parte dei problemi operativi di comunicazione sono oggi risolti dalla strumentazione multimediale o ipermediale. La linea evolutiva dei sistemi di comunicazione è chiaramente segnata in direzione della ipermedialità. Per le imprese, la comunicazione ipermediale è una risorsa notevole nella misura in cui facilita l'acquisizione e la circolazione delle informazioni, moltiplica le opportunità di contatto e agevola la ricerca di nuovi clienti potenziali. La realtà dei mercati è che sono influenzati e per certi aspetti condizionati dai sistemi di comunicazione, che fissano gli standard di interazione più efficaci e diventano, come nel caso di Internet, veri e propri strumenti di marketing.

La *velocizzazione dei processi aziendali*, sia interni, sia relativi al rapporto con il mercato, è apparsa in tutta la sua portata competitiva quando la combinazione di tecnologia elettronica e abilità di riorganizzazione ha permesso ad alcune aziende di ridurre il consumo del tempo in alcuni processi nelle percentuali oscillanti fra il 30% ed il 70%; sempre nelle stesse aziende, attraverso

la riduzione del consumo di tempo, si è riuscito ad ottenere forti incrementi di produttività. Il tempo si è così dimostrato essere una sorta di "comune denominatore" per il recupero di efficienza, e quindi uno strumento competitivo attraverso il quale le organizzazioni più capaci possono riuscire ad ottenere una serie di benefici nella soddisfazione del cliente direttamente funzionali all'aumento della quota di mercato e della redditività.

Capitolo 2

Il fenomeno Open Source

Probabilmente tutti noi usiamo nei nostri computer alcuni programmi open source, anche senza saperlo, quali Firefox, Open Office, VLC; la particolarità delle licenze open source è che gli autori invece di vietare, permettono non solo di usare e copiare, ma anche di modificare, ampliare, elaborare, vendere e tutto questo senza imporre obblighi a ricompensare economicamente gli autori. Se pensiamo che il mercato è tradizionalmente basato sullo scambio tra un produttore ed un consumatore che paga un prezzo per ottenere un bene o un servizio, si capisce come l'open source sia un fenomeno molto interessante da studiare sia da un punto di vista economico che sociale, visto che la produzione del bene o del servizio avviene grazie al contributo di più persone.

Lo scopo primario delle licenze open source non è la gratuità del software, ma la sua sopravvivenza ovvero la certezza che vi sia la possibilità per chiunque e in qualunque momento, anche futuro, di apportare miglioramenti o comunque modifiche al programma, e di installarlo senza alcuna limitazione.

Analizziamo cosa vuol dire "essere open source"? E in particolare quali sono le caratteristiche che deve avere un progetto per potersi definire open source?

2.1 Definizione di open source

Open source non significa semplicemente libero accesso al codice sorgente di un programma. La licenza di un software, per potersi considerare open source, deve soddisfare i seguenti criteri:

1. *Libera redistribuzione*

La licenza non deve limitare nessuno per la vendita o la cessione del software come componente di una distribuzione aggregata di software contenente programmi provenienti da fonti diverse. La licenza non deve richiedere “royalties” o altri pagamenti per tali vendite.

2. *Codice sorgente*

Il programma deve includere il codice sorgente e ne deve essere permessa la distribuzione sia come codice sorgente che in forma compilata. Laddove alcune forme di un prodotto non siano distribuite con il relativo codice sorgente, deve essere chiaramente indicato il modo per ottenerlo, ad un costo non superiore ad una ragionevole spesa di distribuzione, preferibilmente scaricandolo gratuitamente da Internet. Il codice sorgente deve essere la forma preferita che un programmatore sceglie per modificare il programma. Inoltre, il codice sorgente deliberatamente reso illeggibile e le forme intermedie come l'output di un preprocessore o compilatore non rispondono ai requisiti.

3. *Prodotti derivati*

La licenza deve permettere modifiche e prodotti derivati, e deve consentire la distribuzione sotto le stesse condizioni della licenza del software originale.

4. *Integrità del codice sorgente originale*

La licenza può impedire la distribuzione del codice sorgente in forma modificata, a patto che venga consentita la distribuzione dell'originale accompagnato da "patch", ovvero file che permettono di applicare modifiche automatiche al codice sorgente in fase di compilazione. La licenza deve esplicitamente permettere la distribuzione del software costruito da codice sorgente modificato e può richiedere che i prodotti derivati portino un nome o una versione diversa dal software originale.

5. *Nessuna discriminazione contro persone o gruppi*

La licenza non deve discriminare alcuna persona o gruppo di persone.

6. *Nessuna discriminazione verso campi di applicazione*

La licenza non deve impedire di far uso del programma in un ambito specifico. Ad esempio non si può impedire l'uso del programma in ambito commerciale o nell'ambito della ricerca genetica.

7. *Distribuzione della licenza*

I diritti allegati a un programma devono essere applicabili a tutti coloro a cui il programma è ridistribuito, senza che sia necessaria l'emissione di ulteriori licenze.

8. *La licenza non deve essere specifica per un prodotto*

I diritti allegati al programma non devono dipendere dalle parti di un programma di una particolare distribuzione di software. Se il programma è estratto da quella distribuzione e usato o ridistribuito secondo i termini della licenza del programma, tutti coloro che ricevano il programma dovranno avere gli stessi diritti che sono garantiti nel caso della distribuzione originale.

9. *La licenza non deve limitare altri software*

La licenza non deve porre restrizioni su altro software che viene distribuito insieme al software concesso in licenza. Per esempio, la licenza non deve richiedere che tutti gli altri programmi distribuiti sugli stessi supporti devono essere software open source.

10. Conformità della licenza e della certificazione

Qualsiasi programma che faccia uso di licenze certificate come conformi alla Open Source Definition può utilizzare il marchio registrato Open Source e, inoltre, il codice sorgente può essere dichiarato esplicitamente di pubblico dominio. Nessun altro programma o licenza è certificato per fare uso del marchio registrato Open Source.⁸

2.2 Storia delle open source

Possiamo dividere la storia di questo fenomeno dalle origini negli anni 60 ai giorni d'oggi in tre fasi: la prima fase si ebbe tra gli anni sessanta e ottanta in cui il software open source veniva sviluppato attorno a centri universitari, come Berkeley o il MIT, e presso i centri di sviluppo di grandi multinazionali, come Xerox o Bell Labs, i quali si occupavano del coordinamento e diffusione sia interna che esterna.

Successivamente, in una seconda fase compresa tra gli anni ottanta e novanta, questo tipo di software esce dai laboratori dei grandi produttori e delle università con l'avvento della Free Software Foundation e della General Public License (GPL), licenza che sarà la più usata dagli sviluppatori open source. Questa licenza introduce un meccanismo virale per cui tutti i software derivati dovranno essere distribuiti con la stessa licenza. A questo punto è chiaro che

⁸ La definizione è stata tratta dalla pagina web <http://www.opensource.org/docs/definition.php> e corrisponde alla Open Source Definition (OSD).

alcune tipologie di software non potevano mai “chiudersi” diventando software commerciale.

Infine, il periodo che intercorre dagli anni novanta ad oggi rappresenta la terza fase del software open source, dove un impulso molto forte venne dato da Linux. Fin dallo sviluppo di GNU mancava una parte centrale del sistema operativo che fosse in grado di coordinare il lavoro delle altre parti che erano già state sviluppate e si occupava di singole funzioni del sistema operativo, Linux fece proprio questo e così nacque GNU-Linux, oggi chiamato semplicemente Linux. Lo sviluppo di un solido sistema operativo open source ha costituito la base per lo sviluppo di altri singoli successivi applicativi sempre open source (Firefox) e dunque ha rafforzato questo movimento. Possiamo, quindi, affermare che Linux è stato la pietra miliare del movimento open source, infatti in quegli anni è aumentato il numero di progetti e anche il numero di contributori per progetto. Una statistica interessante è quella del numero di sviluppatori per progetto; la tabella 2.1 seguente mostra che il 68% dei progetti hanno un solo sviluppatore, quelli tra 2 e 10 sviluppatori sono il 30% e meno del 2% quelli con più di 50 sviluppatori.⁹ Ciò significa che tanti progetti non riescono ad attirare l'interesse delle comunità open source e rimangono con un solo sviluppatore che di solito corrisponde con il fondatore del progetto.

Tabella 2.1 Il numero di sviluppatori per progetto

Numero sviluppatori per progetto	Numero di progetti	% su numero totale di progetti
1	77571	67,60%
2	17576	15,32%
3-4	11362	9,90%
5-9	6136	5,35%
10-19	1638	1,43%
20-49	412	0,36%
≥50	56	0,05%

⁹ Barbarito, L. (2008). *Economia dell'open source*. Torino: Pearson Paravia Bruno Mondadori.

Totale progetti	114751	100%
------------------------	---------------	-------------

Fonte: Chaim Fershtman e Neil Gandal, *Microstructure of Collaboration: The Network of Open Source Software*, Working Paper #08-01, Aprile 2008, Net Institute, www.netintest.org

La Tabella 2.2 ci mostra invece che in genere gli sviluppatori riescono a partecipare a pochi progetti alla volta. Il 77% riesce a partecipare a un solo progetto e il 14% a due progetti, ciò significa che solo il restante 9% sono gli sviluppatori che riescono a partecipare a tre o più progetti.

Tabella 2.2 La partecipazione degli sviluppatori nei progetti

Numero progetti seguiti	Numero di sviluppatori	% su numero totale degli sviluppatori
1	123562	77,18%
2	22690	14,17%
3-4	10347	6,46%
5-9	3161	1,97%
10-19	317	0,20%
20-49	26	0,02%
≥50	1	0,00%
Totale progetti	114751	100%

Fonte: Chaim Fershtman e Neil Gandal, *Microstructure of Collaboration: The Network of Open Source Software*, Working Paper #08-01, Aprile 2008, Net Institute, www.netintest.org

Un dato economico importante del mercato dei software open source è quello riguardante “l’ecosistema Linux” che vale oggi intorno ai 30 miliardi di dollari secondo l’agenzia di consulenza Idc.¹⁰

La diffusione di Internet e dei PC è un altro fattore importante che ha accelerato il meccanismo di scambio e di ampliamento dei progetti.

Se all’inizio della diffusione dei software, i programmatori erano soliti scambiarsi i programmi non immaginando nemmeno un possibile sfruttamento commerciale degli stessi e dando invece molta più importanza all’hardware, dagli anni novanta in avanti il ruolo del software è progressivamente cresciuto proprio

¹⁰ Anderson, C. (2009). *Gratis*. Milano: Rizzoli.

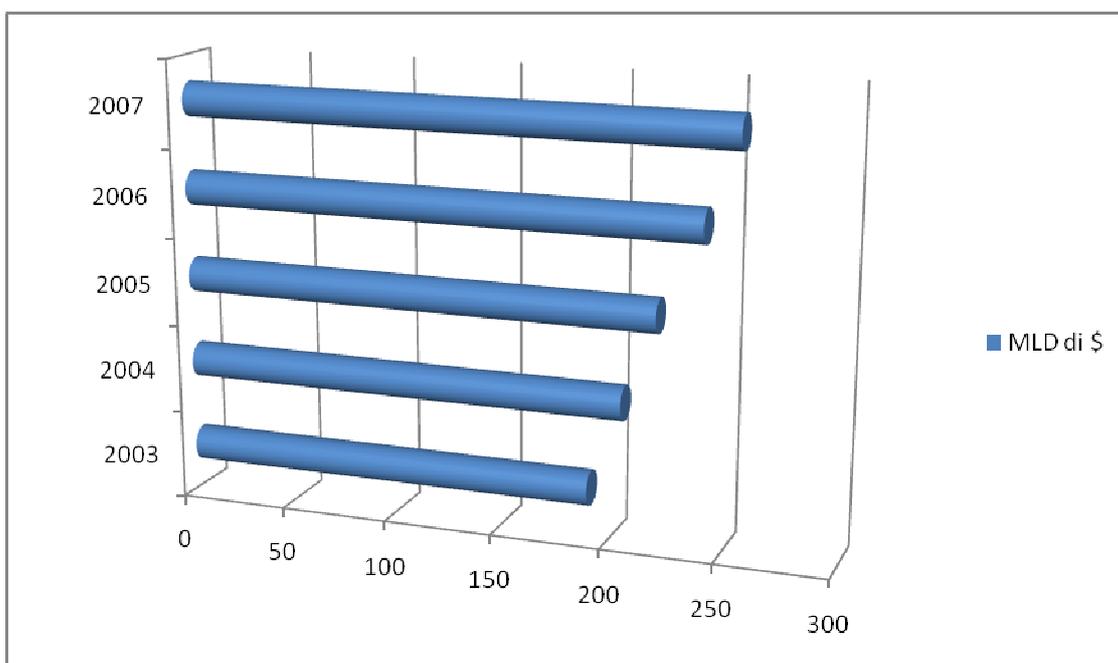
quando i personal computer cominciavano a penetrare nel mercato di massa. Il software e soprattutto il sistema operativo è diventato più importante rispetto all'hardware e ha condizionato lo sviluppo di tutto il più generale mercato dell'Information e Communication Technology (ICT). Si deve riconoscere che la penetrazione nel mercato di massa dei personal computer è dovuta certamente a una riduzione del prezzo dell'hardware e a un contemporaneo miglioramento qualitativo dello stesso.

Se all'inizio della storia dell'information technology (IT), negli anni sessanta, i costruttori di computer erano soliti auto-produrre i codici per il funzionamento delle proprie macchine, a partire dalla metà dello stesso decennio si è sviluppata una vera e propria industria del software, autonoma rispetto a quella dell'hardware. Alcuni programmi potevano girare su più macchine e le competenze degli sviluppatori erano completamente diverse da quelle dei produttori di hardware, per cui molti produttori cominciarono a creare divisioni interne che si occupassero del software oppure a esternalizzare questa funzione. Oggi il mercato del software, a livello mondiale, vale 251 miliardi di dollari (Tabella 2.3) e ha mostrato negli ultimi cinque anni un tasso di crescita medio annuo del 7,2% abbastanza regolare.

Il mercato dei software vede come principali attori tre continenti che sono: America, Europa e Asia; quest'ultimi si dividono quasi la totalità del mercato mondiale, circa un terzo ciascuno. Scomponendo invece il mercato europeo del software per paesi, la quota italiana è inferiore di circa il 30% rispetto a Francia, Germania e Regno Unito, paesi non dissimili rispetto al nostro per dimensioni. Certamente, una ragione, oltre al peggior quadro macroeconomico in cui si trova il nostro paese, è riconducibile alla dimensione tipica delle imprese italiane, solitamente molto piccole rispetto alla media europea. La piccola dimensione non aiuta l'affermazione dei nostri professionisti in un mercato che è guidato da aziende multinazionali, mentre è assolutamente compatibile con un maggior sviluppo del settore dei servizi, dove questo tipo di attività può benissimo essere svolta da piccole imprese.¹¹

¹¹ Barbarito, L. (2008). *Economia dell'open source*. Torino: Pearson Paravia Bruno Mondadori.

Tabella 2.3 Il mercato mondiale del software



Fonte: Datamonitor

2.3 Open source e software libero

*“Quasi tutto il software open source è software libero; i due termini descrivono all'incirca la stessa categoria di software, ma si basano su valori fondamentalmente diversi. L'open source è una metodologia di sviluppo; il software libero è un movimento sociale”.*¹²

¹² Richard Stallman, pubblicata sul sito <http://www.gnu.org>.

Il termine "*open source*" venne coniato agli inizi del 1998 su iniziativa di Bruce Perens, Eric S. Raymond, Hall, Tim O'Reilly, Linus Torvalds e altri importanti sviluppatori della "*Free Software Community*", come allora veniva chiamata. L'obiettivo principale era quello di rendere l'idea del software libero più accettabile all'ambiente commerciale, evitando le posizioni intransigenti di Stallman e contemporaneamente evitare l'equivoco generato dalla parola "*free*" in inglese (che significa sia *gratuito* che *libero*). La parola "*source*" stava a sottolineare il fatto che un software non è tanto il programma eseguibile, quanto il suo punto di partenza, il sorgente appunto. L'obiettivo era proprio quello di rendere il prodotto accettabile nelle aziende evitando l'uso della restrittiva licenza GPL.¹³

Il più noto fra i primi protagonisti della storia del software libero è certamente Richard Marshall Stallman, "l'ultimo degli hacker" come amava definirsi. La sua biografia intellettuale e personale è strettamente connessa a quella degli hacker puri dei primi laboratori informatici, per la stessa passione del calcolatore e per la stessa insofferenza verso gli intoppi burocratici, la segretezza, l'assenza di cooperazione e di scambio intellettuale. Stallman sosteneva che la sua inflessibilità morale rendeva difficile accettare le regole di scambio opportunistiche delle relazioni umane. Questa sua ricerca di informalità e di libertà lo aveva indotto a trasferirsi da Harvard alla comunità baluardo degli hacker, il laboratorio di Intelligenza Artificiale del M.I.T. (Massachusetts Institute of Technology), dove gli era stato offerto un posto di programmatore sistemista. E qui, dal 1971 al 1983, aveva ingaggiato un'infaticabile battaglia per un "*sistema aperto*" a tutti gli utenti, contro l'utilizzo obbligatorio di codici di accesso e contro i segreti dei sistemi di sicurezza. La sua convinzione sulla non utilità e, anzi, sulla dannosità di non diffondere il codice di controllo della macchina, basata su premesse insieme etiche e funzionali, trovava una continua conferma nei molti problemi quotidiani connessi all'utilizzo dei computer e di altra strumentazione elettronica. L'esempio più noto, che dimostrò come i principi della proprietà intellettuale costituissero un vincolo all'efficienza impedendo di risolvere un fastidiosissimo inconveniente, riguardò il caso della stampante laser Xerox, che

¹³ <http://www.wikipedia.org>.

messa generosamente a disposizione del Laboratorio di Intelligenza Artificiale dalla stessa azienda, si fermava in continuazione. Per ovviare ai frequenti guasti, Stallman aveva pensato di modificare il programma, per attuare, in un modo più veloce, un pronto intervento cooperativo senza aspettare l'addetto della Xerox. Condizione essenziale per l'attuazione della nuova procedura era la conoscenza del codice sorgente della macchina, ma la Xerox, diversamente dal passato, negò l'accesso a quel codice, in quanto protetto dal copyright. Il programma proprietario non poteva essere più conosciuto e trasformato. Il mondo esterno con le sue regole stava ponendo serie barriere al lavoro della comunità degli hacker e progressivamente ne incrinava anche i valori ideali, rendendo i suoi membri più sensibili a comportamenti strumentali.

Così, quando anche il laboratorio di Intelligenza Artificiale, con il rischio dei tagli dei finanziamenti, dovette adeguarsi a quelle richieste di sicurezza che imponevano di limitare e controllare il libero collegamento ai calcolatori del laboratorio, e quando con l'acquisto di un nuovo Digital PDP-10 si interruppe la consuetudine della condivisione libera delle risorse della macchina, resa possibile dal sistema operativo ITS (Incompatible Timesharing System), costruito dagli stessi programmatori e basato su un'architettura aperta, Stallman lasciò il M.I.T. e si pose come primo obiettivo lo sviluppo di un sistema operativo compatibile con lo Unix di AT&T (American Telephone and Telegraph, Inc.), il sistema operativo allora più diffuso nel mondo dei minicalcolatori. Nel giorno del "Thanksgiving" del 1983 attraverso Arpanet comunicò alla comunità hacker e agli interessati allo sviluppo del software libero la decisione di lasciare il M.I.T. per impegnarsi nella realizzazione del nuovo sistema operativo Unix compatibile. Il diritto di proprietà, il *copyright*, stava, infatti, contaminando e indebolendo la comunità degli hacker e, insieme ad essa, l'opportunità di un libero scambio sia materiale che intellettuale del software e dei suoi contenuti.

Stallman battezzò il nuovo sistema operativo con l'acronimo GNU, come "*Gnu is Not Unix*", una definizione formulata secondo un'antica consuetudine della comunità hacker. In altri termini: "*GNU non è lo Unix*" di AT&T, non è quindi proprietario, ma ha le stesse funzionalità ed è compatibile con quello. Stallman impose per GNU un requisito fondamentale, destinato a giocare un ruolo

centrale nel mondo del software libero: essere “*open source*”. Il software sorgente, prima di essere utilizzato, deve essere “*compilato*”, ossia tradotto nel codice eseguibile o codice di macchina, un'innumerabile sequenza di uno e di zero, che la macchina è in grado di interpretare, ma l'uomo generalmente no, a meno di affrontare anni di duro lavoro anche su programmi relativamente brevi.¹⁴

Nel prosieguo della trattazione utilizzeremo i termini “Open Source” e “Software Libero” come sinonimi, se non è specificata la differenza concettuale sopra descritta. In effetti essi sono utilizzati nel linguaggio comune come sinonimi essendo il loro significato coincidente in certi contesti.

2.4 Open source e software proprietario

Dall'inizio degli anni duemila le cose sono molto cambiate e stiamo assistendo a una rivitalizzazione del software open source rispetto a quello commerciale. Lo scenario che si potrebbe immaginare è una sorta di sfida tra il “*software dello scambio sociale e quello della produzione d'impresa*”, in effetti fino agli anni novanta i software proprietari erano in competizione con i software open source, ma oggi gli scenari sono cambiati.

Il macro fenomeno a cui appartiene l'open source è senza dubbio la produzione e la condivisione sociale, con progetti come il peer to peer (P2P), Wikipedia, Skype che stanno certamente variando gli equilibri di mercati tradizionali basati sullo scambio monetario, si pensi come il P2P ha condizionato in maniera profondissima l'industria discografica, ma la competizione tra questi due mondi si sta trasformando, in molti casi, nella cooperazione di grandi aziende produttrici di software proprietari con progetti open source.

¹⁴ Angelo Raffaele Meo, Politecnico di Torino – “Servizio di osservatorio tecnologico per la scuola, Ministero della Pubblica Istruzione”

Quindi, si è passati da una situazione di estrema competizione, dove venivano sminuite le potenzialità dei progetti open source da parte delle grandi imprese (tentando di screditarne la qualità, l'affidabilità e la sicurezza a favore dei propri prodotti la cui "chiusura" era un prezzo da pagare per tutte quelle caratteristiche), ad una situazione di cooperazione ed addirittura di sponsorizzazione di alcuni progetti open source e questo vale non solo per le aziende produttrici di software ma anche per quelle produttrici di hardware.

Perché e come i grandi produttori hanno cambiato atteggiamento? Sicuramente la forza con il quale i progetti open source si sono affermati, nonostante le strategie di screditamento messe in atto dalle imprese, è uno dei motivi. Ma la cosa più interessante da notare è data dalle strategie utilizzate dalle imprese per creare un business sul fenomeno open source.

La questione può sembrare quella di uno sfruttamento commerciale di un iniziativa di scambio sociale, come oggi sostiene Richard Stallman relegando le Open Source a "*un fenomeno che ha mancato l'obiettivo di software libero*", ma su questo tema si deve sottolineare quanto detto nella trattazione "open source e software libero"; in particolare ricordiamo che la scelta del termine "*open source*" venne fatta da importanti sviluppatori di quella che era la "*Free Software Community*", del calibro di Linus Torvalds, proprio per rendere l'idea del software libero più accettabile all'ambiente commerciale e non utilizzare la parola "*free*" che in inglese poteva generare qualche equivoco, significando sia *libero* che *gratuito*. Quindi, questa scelta venne fatta proprio per cercare una cooperazione con le imprese che possono aiutare lo sviluppo di progetti open source rendendoli più accessibili ai meno esperti di informatica.

Le strategie utilizzate dalle imprese sono sostanzialmente due ed è facile comprenderne anche le ragioni. La prima si basa sulla *promozione di un software open source allo scopo di aumentare le vendite di un proprio prodotto complementare a pagamento*. Per esempio, fornire il software open source (Linux) preinstallato su componenti hardware (PC) venduti a pagamento, oppure, distribuire software open source per poi vendere personalizzazioni e adattamenti secondo le esigenze del cliente. Il prodotto complementare (a pagamento) in questi casi è prima un bene e poi un servizio, si spiega così la convenienza da

parte delle aziende nella promozione di un progetto open source, anche finanziandolo con investimenti monetari. Alcuni esempi di questa strategia sono la IBM e la Red Hat: la prima ha finanziato nel 2001 lo sviluppo di Linux per 210 milioni di dollari e ha continuato successivamente ad investire in consulenze e training per lo sviluppo di questo sistema operativo, essa si concentra sull'hardware utilizzando Linux come sistema operativo, contando sul fatto che i miglioramenti di quest'ultimo facciano vendere più hardware e servizi personalizzati che lei stessa fornisce. Il secondo esempio è quello della Red Hat che distribuisce Linux con servizi accessori in versioni a pagamento con manuali e in un certo senso "certificati" e finanzia anch'essa lo sviluppo del sistema operativo.

Quindi, il motivo per il quale le imprese hanno cambiato il loro atteggiamento verso il fenomeno open source che deduciamo da questa prima strategia "*di cooperazione*" è dato dalle varie opportunità di business che il mondo open source mette a disposizione delle imprese, soprattutto per quelle operanti nei settori hardware e della distribuzione software.

La seconda strategia è attuabile dalle società di produzione di software commerciali e consiste nel *rilasciare in codice aperto parte del proprio codice proprietario o addirittura tutto*, così da consentire agli sviluppatori terzi di integrare meglio il programma con altri software o per rivitalizzare programmi "*chiusi*" che non hanno avuto successo. Una società che ci farà capire meglio questa strategia è Sun, che tramite tre suoi progetti ha attuato un processo di apertura verso il mondo open source. Il primo progetto è il rilascio del linguaggio di programmazione Java con licenza GPL, anche se la società continua a distribuire le proprie licenze commerciali per venire incontro a quelle aziende che desiderano integrare Java in software proprietari. Il secondo progetto è quello relativo al rilascio del codice sorgente di Solaris, un sistema operativo che non ha avuto fortuna come software proprietario e si affida al mondo open source per essere sviluppato e rivitalizzato da un lento declino. Il terzo progetto di Sun è l'acquisto di MySQL per integrare questo database nei propri sistemi. Un'altra azienda che segue questa politica è Oracle che ha aperto parte dei propri codici sorgente affinché i propri database siano integrabili, consultabili e comunque

possano interagire anche con altri database oltre che con Linux, quest'ultimo inoltre riceve una sponsorizzazione dalla stessa Oracle.

In generale, dall'esito delle strategie adottate dagli operatori esistenti, si può affermare che la strategia di aprire il codice sorgente o parte di questo costituisce sia l'ammissione di rilevanza del mondo open source che la volontà di rendere appetibile il proprio prodotto commerciale in quei segmenti dove si prospetta ancora una crescita a tassi rilevanti, ad esempio il segmento dei database. Tuttavia, l'apertura di parti di un codice sorgente può non risultare efficace se il software che si vuole aprire ha quote di mercato molto piccole ed è in fase di declino, come nel caso già visto della società Sun per il progetto Solaris che non ha trovato l'interesse del mondo open source per lo sviluppo di un sistema operativo il cui destino sembrava già segnato.

In un confronto tra software commerciale e software open source, si può affermare che già dagli anni ottanta il mercato del software cosiddetto "*proprietario*" cioè a pagamento si era definitivamente affermato, relegando il software open source a un ruolo minoritario e solo per esperti. Era la fascia più avanzata dei programmatori che, spesso per risolvere problemi difficili e circoscritti, usavano scambiarsi liberamente il software e contribuire congiuntamente allo sviluppo¹⁵.

Il software proprietario viene generalmente venduto in formato eseguibile, per rendere praticamente impossibile la sua interpretazione e la sua modifica in funzione delle esigenze del suo utilizzatore. Viceversa, i programmi "*open source*", proprio perché disponibili in forma simbolica, che un programmatore riesce facilmente ad interpretare, sono veramente "*open*", nel senso che possono essere letti, corretti e trasformati in funzione di specifiche esigenze.¹⁶

2.5 Legislazione del software open source

¹⁵ Barbarito, L. (2008). *Economia dell'open source*. Torino: Pearson Paravia Bruno Mondadori.

¹⁶ Angelo Raffaele Meo, Politecnico di Torino – “Servizio di osservatorio tecnologico per la scuola, Ministero della Pubblica Istruzione”

Il software è l'indiscusso protagonista di quella rivoluzione informatica che in poco più di un trentennio ha radicalmente mutato abitudini, usi, costumi e culture del mondo intero mettendo a dura prova la tenuta di regole e principi dell'economia e del diritto.

Non c'è dunque da meravigliarsi se proprio il mondo del software, nell'ultimo periodo, sia attraversato da idee e filosofie diverse e contrapposte e sia teatro di confronti e tensioni le cui conseguenze ed effetti sono destinati a propagarsi e ripercuotersi ben al di là dei confini dell'industria informatica ed a investire le dinamiche del progresso scientifico, tecnico e culturale del mondo intero.

In questo contesto si inserisce il confronto tra due modi diversi di pensare il software: quello dell'open source o come anche si dice, superando le distinzioni storiche e concettuali (che sono state esposte in precedenza), del software libero e quello del software proprietario.

L'approfondimento sotto il profilo giuridico delle questioni sottese alla crescente diffusione del modello OS deve essere fatto superando certi corollari su cui appare, invero, fondata gran parte della filosofia open source (ad esempio l'altruismo, la condivisione e la gratuità) e cercare di individuare regole certe ed univoche che garantiscono l'equo temperamento tra diritti e obblighi della collettività, libertà e ragioni del progresso. Tale compromesso costituisce, d'altra parte, il fondamento del sistema normativo, che è da adattare costantemente alle mutate esigenze dei tempi.

Il mercato del software è nato sul finire degli anni sessanta a seguito del provvedimento con il quale l'Autorità Antitrust americana impose all'IBM di interrompere la vendita congiunta di hardware e software consentendo così la circolazione autonoma dei due beni e trasformando i "*programmi per elaboratore*" da bene accessorio ad autonomo bene giuridico.¹⁷

¹⁷ Scorza, G. (2002). Associazione italiana per l'informatica ed il calcolo automatico. ICT globalizzazione e localismi. Bitonto (BA) : AICA, 2002. p. 701-720 ()1365453

Gli elevati costi connessi alla progettazione dei programmi per elaboratore e la relativa semplicità ed economicità con cui possono essere copiati, hanno messo in luce fin dall'ingresso sul mercato di questi prodotti tecnologici l'esigenza di una tutela giuridica in grado di remunerare gli elevati investimenti, sia in termini di risorse umane che finanziarie, necessari per la loro realizzazione. Essendo il software un'opera creativa si presentava al legislatore la scelta tra la legge a tutela delle invenzioni industriali o quella a tutela del diritto d'autore.

Il *brevetto* conferisce un monopolio temporaneo di sfruttamento dell'invenzione in un territorio e per un periodo ben determinato, al fine di impedire ad altri di produrre, vendere o utilizzare la propria invenzione senza autorizzazione. Il *diritto d'autore* garantisce al titolare dell'opera di godere dei benefici del proprio lavoro e degli investimenti connessi.

In tale contesto non è mancato chi, pur condividendo l'idea di collocare il *software* all'interno del sistema delle creazioni intellettuali, aveva sottolineato l'opportunità di considerarlo come un *tertium genus* alla stregua di quanto sarebbe poi accaduto per le topografie dei prodotti semiconduttori; l'esigenza di certezza ed uniformità del diritto scongiurarono, tuttavia, di seguire questa strada e spinsero il mondo giuridico ed economico del tempo a concentrare la propria attenzione sul diritto d'autore e sulla disciplina brevettuale.

La scelta dell'Europa, alla fine, cadde sul *diritto d'autore*.

Con l'approvazione della Direttiva comunitaria 91/250/CEE l'Europa sceglie la tutela offerta dal diritto d'autore che non conferisce monopoli che ostacolerebbero lo sviluppo della ricerca tecnica e salvaguarda contemporaneamente un'area di libertà sufficiente a permettere ad altri autori di creare programmi simili, purché non si tratti di mera duplicazione.

La Direttiva in esame accosta il software alle opere letterarie. Tale analogia viene ravvisata nell'uso di una forma espressiva, linguaggio o altra modalità espressiva percepibile ai sensi umani, che esteriorizza un'idea, frutto dell'abilità e dell'impegno creativo dell'individuo. Viene, quindi, sancita la tutelabilità giuridica dei programmi, in quanto assimilati alle opere letterarie, in base alle leggi sul diritto d'autore, in conformità alla disposizioni della Convenzione di Berna, che all'art. 2.1 recita: "*L'espressione - opere letterarie e artistiche - comprende tutte le*

produzioni in campo letterario, scientifico e artistico, qualunque ne sia il modo o la forma di espressione”.

Tuttavia, la qualifica di opera letteraria deve essere intesa come una finzione giuridica, volta ad accordare ai programmi la tutela d'autore. La artificiosa comparazione del software a un'opera letteraria è stata ampiamente dibattuta e spesso criticata. Infatti, nell'opera letteraria è l'aspetto artistico ad avere il sopravvento, e quindi la protezione dell'autore e la sua personale relazione con l'opera; al contrario, nei programmi per elaboratore, è prevalente l'aspetto utilitaristico e gli aspetti prettamente economici.

In Italia, nonostante la giurisprudenza, sin dai primi anni ottanta, riconosceva già ai programmi per elaboratore la tutela d'autore, il legislatore ha affrontato il problema solo nei primi anni novanta dando attuazione, con il Decreto Legislativo 518 del 29 Dicembre 1992, alla Direttiva Europea n.250 del 14 Maggio 1991; direttiva fortemente condizionata più dalla preoccupazione di individuare una disciplina uniforme del software che non del reale convincimento che tale nuovo bene giuridico fosse effettivamente assimilabile alle opere dell'ingegno.¹⁸

Inoltre nel primo articolo della Direttiva comunitaria si afferma che gli stati membri devono attribuire ai programmi per elaboratore la tutela riconosciuta dalla legge sul diritto d'autore ai sensi della Convenzione di Berna. Nonostante questa conclusione, è interessante evidenziare che, nell'operare la scelta del diritto d'autore come forma di tutela per il software, il legislatore dà per implicito che questa protezione non sia esaustiva, e che, nel futuro, andrà completata in modo da tutelare anche il contenuto e le idee sviluppate nel software, che in base al secondo punto dell'articolo 2 della Direttiva vengono espressamente esclusi dalla protezione; il completamento della protezione giuridica venne, presumibilmente, ipotizzata attraverso una tutela di tipo brevettuale la cui applicazione nell'articolo 9 non venne appositamente esclusa.

¹⁸ Scorza, G. (2002). Associazione italiana per l'informatica ed il calcolo automatico. ICT globalizzazione e localismi. Bitonto (BA) : AICA, 2002. p. 701-720 ()1365453

Per dissipare ogni dubbio su cosa venga tutelato dalla nuova normativa si precisa “*che solo l'espressione di un programma per elaboratore è oggetto di tutela e che le idee e i principi alla base di qualsiasi elemento di un programma, compresi quelli alla base delle sue interfacce, non sono tutelati dal diritto d'autore a norma della presente direttiva*” e si rimarca che non sono tutelati “*le idee e i principi che sono alla base della logica degli algoritmi e dei linguaggi di programmazione*”.

La dottrina, salvo isolati spunti, ha sempre considerato la disciplina del diritto d'autore e quella brevettuale come esclusive e antitetiche, poiché ogni creazione intellettuale non può che rientrare alternativamente tra le invenzioni industriali e modelli di utilità o tra le opere dell'ingegno, e quindi lo stesso principio si deve applicare anche ai programmi per elaboratore.

I limiti del diritto d'autore, che tutela la forma espressiva e non le idee stesse, rendono però tale soluzione non del tutto adeguata alla completa tutela del prodotto “*programma per elaboratore*”, poiché risultano protetti i programmi in quanto tali, mentre restano esclusi dalla tutela *la logica o gli algoritmi* sulla quale essi si basano. Si impone, quindi, la ricerca di una soluzione non sostitutiva ma complementare al diritto d'autore, che si spinga sin dove esso non è in grado d'arrivare. Il brevetto per le invenzioni industriali applicato al software colmerebbe il vuoto di tutela, dando così soddisfazione all'esigenza di protezione giuridica nell'aspetto ideativo - contenutistico.

La Commissione Europea ha presentato il 20 Febbraio 2002 una proposta di Direttiva del Parlamento Europeo e del Consiglio relativa alla “*brevettabilità delle invenzioni attuate per mezzo di elaboratori elettronici*” accompagnata da una relazione che fornisce sinteticamente l'ambito in cui si è mossa la Commissione, e una chiara spiegazione alle scelte da essa effettuate.

Il 6 Luglio 2005 il Parlamento ha respinto a grandissima maggioranza la proposta.

Un acceso dibattito è stato aperto prima e dopo la proposta in questione. Da una parte, i sostenitori dei sistemi OS i quali pongono l'accento sul fatto che il principio della brevettabilità del software sarebbe incompatibile con l'*open source*, in cui tutti concorrono all'evoluzione dei programmi proprio in virtù della

possibilità di impiegare parti di software sviluppate da altri (circostanza che ha favorito gli alti tassi di sviluppo dell'informatica); inoltre essi ritengono che qualora passasse una riforma di questo tipo si avrebbe comunque un aumento dei costi di sviluppo, che porterebbe ad un aumento dei prezzi per gli utenti finali, rallentando l'innovazione. Dall'altra, i sostenitori della brevettabilità pongono, invece, l'accento sulla necessità di garantire, tramite brevetti, idonei ritorni economici rispetto agli investimenti per la creazione di nuovi software come accade in Giappone e Stati Uniti.

In questa sede non si vuole dare un giudizio sulla brevettabilità, ma si vuole portare a conoscenza un dibattito importante che ha diviso e forse allontanato i due “mondi del software”.

Dopo aver esposto la regolamentazione dei software in generale, possiamo andare ad analizzare le Licenze Open Source.

Rientrano in quest'ambito tutte le licenze che corrispondono ai principi che abbiamo esposto nella definizione di software open source (Paragrafo 2.1); inoltre la Open Source Initiative certifica la conformità dei software alla sua definizione, pubblicando la lista aggiornata sul suo sito Internet¹⁹ delle licenze “ufficialmente” riconosciute come OS.

L'attuale lista racchiude le seguenti licenze:

- Academic Free License 3.0 (AFL 3.0)
- Affero GNU Public License
- Adaptive Public License
- Apache License, 2.0
- Apple Public Source License
- Artistic license 2.0
- Attribution Assurance Licenses
- New and Simplified BSD licenses
- Boost Software License (BSL1.0)

¹⁹ <http://www.opensource.org>.

- Computer Associates Trusted Open Source License 1.1
- Common Development and Distribution License
- Common Public Attribution License 1.0 (CPAL)
- CUA Office Public License Version 1.0
- EU DataGrid Software License
- Eclipse Public License
- Educational Community License, Version 2.0
- Eiffel Forum License V2.0
- Entessa Public License
- European Union Public License
- Fair License
- Frameworkx License
- GNU General Public License (GPL)
- GNU General Public License version 3.0 (GPLv3)
- GNU Library or "Lesser" General Public License (LGPL)
- GNU Library or "Lesser" General Public License version 3.0 (LGPLv3)
- Historical Permission Notice and Disclaimer
- IBM Public License
- IPA Font License
- ISC License
- Lucent Public License Version 1.02
- MirOS Licence
- Microsoft Public License (Ms-PL)
- Microsoft Reciprocal License (Ms-RL)
- MIT license
- Motosoto License
- Mozilla Public License 1.1 (MPL)
- Multics License
- NASA Open Source Agreement 1.3
- NTP License
- Naumen Public License
- Nethack General Public License

- Nokia Open Source License
- Non-Profit Open Software License 3.0 (Non-Profit OSL 3.0)
- OCLC Research Public License 2.0
- Open Font License 1.1 (OFL 1.1)
- Open Group Test Suite License
- Open Software License 3.0 (OSL 3.0)
- PHP License
- Python license (CNRI Python License)
- Python Software Foundation License
- Qt Public License (QPL)
- RealNetworks Public Source License V1.0
- Reciprocal Public License 1.5 (RPL1.5)
- Ricoh Source Code Public License
- Simple Public License 2.0
- Sleepycat License
- Sun Public License
- Sybase Open Watcom Public License 1.0
- University of Illinois/NCSA Open Source License
- Vovida Software License v. 1.0
- W3C License
- wxWindows Library License
- X.Net License
- Zope Public License
- zlib/libpng license

Dato un così cospicuo numero di licenze esistenti, in costante e significativo aumento negli ultimi anni, il passo successivo è capire quali siano le più diffuse.

I dati evidenziano l'inutilità o comunque la scarsa diffusione di tante licenze, infatti risulta che la maggioranza dei software OS utilizzano nella quasi totalità dei casi una sola licenza, che approfondiremo, la GNU GPL (General Public License).

Quest'ultima è seguita a notevolissima distanza da altre due: la GNU LGPL (Lesser General Public License) e la BSD, quasi tutte le altre sono poco utilizzate.

Il sistema OS non avrebbe mai potuto divenire un modello di sviluppo e distribuzione tanto affermato nel mondo se la filosofia che ne sta alla base si fosse limitata a prevedere la disponibilità del codice sorgente senza preoccuparsi di garantire tale elemento nel tempo. Come rileva lo stesso Richard Stallman: *“Se un programma è software libero quando esce dalle mani del suo autore, non significa necessariamente che sarà software libero per chiunque ne abbia una copia. Per esempio, il software di pubblico dominio (software senza copyright) è software libero, ma chiunque può farne una versione modificata proprietaria”*.

Per superare tale problema è stato creato un apposito schema contrattuale definito *GPL* allo scopo di consentire la circolazione del software Open Source e dei diritti ad esso relativi.

Stando alla descrizione proposta nel preambolo del *contratto di tipo GPL* pubblicato sul sito della Free Software Foundation²⁰, la peculiarità della fattispecie risiederebbe nella circostanza che *“le licenze della maggior parte dei programmi hanno lo scopo di togliere all'utente la libertà di condividere e modificare il programma stesso. Viceversa, la Licenza Pubblica Generica GNU è intesa a garantire la libertà di condividere e modificare il software libero, al fine di assicurare che i programmi siano liberi per tutti i loro utenti”*.

Lo stesso preambolo della GPL illustra altresì lo spirito e la ratio ispiratrice della licenza. In esso si riferisce, infatti: *“Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurarsi che ciascuno abbia la libertà di distribuire copie del software (e farsi pagare, se vuole) che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi e che ciascuno sappia di poter fare queste cose. Per proteggere i diritti dell'utente, abbiamo bisogno di creare delle restrizioni che vietino a chiunque di negare questi diritti o di chiedere di rinunciarvi. Queste restrizioni si traducono in certe responsabilità per chi distribuisce copie del software e per chi lo modifica. Per esempio, chi distribuisce copie di un software GPL, sia gratis sia in cambio di un compenso, deve concedere*

²⁰ <http://www.fsf.org>.

ai destinatari tutti i diritti che ha ricevuto. Deve anche assicurarsi che i destinatari ricevano o possano ottenere il codice sorgente. Inoltre, per proteggere ogni autore e noi stessi, vogliamo assicurarci che ognuno capisca che non ci sono garanzie per i programmi coperti da GPL. Se il programma viene modificato da qualcun altro e ridistribuito, vogliamo che gli acquirenti sappiano che ciò che hanno non è l'originale, in modo che ogni problema introdotto da altri non si rifletta sulla reputazione degli autori originari. Infine, ogni programma libero è costantemente minacciato dai brevetti sui programmi. Vogliamo evitare il pericolo che chi ridistribuisce un programma libero ottenga la proprietà di brevetti, rendendo in pratica il programma di sua proprietà. Per prevenire questo, abbiamo chiarito che ogni brevetto debba essere concesso in licenza d'uso a chiunque, o non avere alcuna restrizione di licenza d'uso”.

Capitolo 3

Un nuovo paradigma economico

L'*economia "tradizionale"* è caratterizzata da alcune proprietà che costituiscono i capisaldi della gestione aziendale; alcune volte queste *regole generali* vengono messe in discussione da fenomeni che sovvertono il normale *modus operandi* delle imprese. Il caso dell'open source è un'eccezione al concetto di "*risorsa nella prospettiva strategica*". Descriviamo, quindi, questo concetto per capire come la risorsa open source cambia i normali paradigmi: la risorsa è l'insieme dei fattori tangibili e intangibili che l'impresa controlla e utilizza nei suoi processi produttivi e costituisce una delle componenti fondamentali del sistema aziendale. Per avere un valore dal punto di vista strategico, le risorse devono avere tre proprietà: la scarsità; la rilevanza; l'appropriabilità.

Con *scarsità* si intende che una risorsa deve essere poco diffusa nell'ambito delle imprese concorrenti; la *rilevanza* va intesa rispetto ai fattori critici di successo nel mercato e, quindi, rispetto alle dinamiche competitive (per un'azienda bancaria, per esempio, l'eventuale sviluppo di massa dei servizi tramite la rete internet renderà progressivamente meno "rilevante" la risorsa fisica costituita dalla rete di sportelli nel territorio); infine, l'*appropriabilità* si basa sui diritti di proprietà che un soggetto può acquisire e mantenere sulla risorsa in

questione, da cui deriva un vantaggio competitivo rispetto ai concorrenti che non possono disporre di tale risorsa.²¹

Quindi, il vantaggio competitivo di un'impresa, secondo un approccio *resource based view*, dipende dal possesso di risorse che presentano le caratteristiche sopra citate. Inoltre l'inefficienza del mercato fa sì che la circolazione di queste risorse sia difficoltosa, determinando una diversa dotazione di risorse tra le imprese. Se il mercato delle risorse fosse efficiente, probabilmente tale disomogeneità nella dotazione di risorse e competenze non vi sarebbe, e quindi le imprese avrebbero tutte la stessa possibilità di performare allo stesso modo.

In questo contesto è necessario capire come funziona il mercato, quali sono le linee guida dettate dalla teoria economica per le imprese che vogliono acquisire un vantaggio competitivo e quali fattori, strutturali di un'impresa, determinano la sua unicità e capacità concorrenziale. Proprio per questo motivo è importante parlare, in questa sede, *dell'analisi dell'eterogeneità delle performance nella prospettiva strategica*, che si inserisce in un filone di studi finalizzati a determinare le fonti che spiegano le differenze nelle performance delle imprese. Gli studi di strategia, in proposito, hanno evidenziato differenti motivazioni, che assegnano di volta in volta un ruolo rilevante ai *fattori specifici dell'impresa*.

La dimensione economica del vantaggio sui concorrenti fa riferimento a due fonti differenti: da una parte, la presenza dei meccanismi o condizioni strutturali sul lato del mercato e del settore e, dall'altra, la presenza di eterogeneità nella distribuzione delle risorse tra gli operatori del settore stesso. La diversità dei due approcci deriva innanzitutto da una visione differente dell'impresa.²² Tuttavia, a prescindere dall'approccio sposato, i due filoni di studi analizzano fonti differenti di eterogeneità delle performance: il primo si riferisce all'appartenenza a un

²¹ Fontana, F., & Caroli, M. G. (2006). *Economia e gestione delle imprese*. Milano: The McGraw-Hill Companies.

²² Wernerfelt (1984).

determinato settore, mentre il secondo cerca di analizzare le differenze legate a fattori *firm-specific*.²³

Dall'analisi dell'ipotesi tradizionale *resource-based*, tuttavia, emerge un'interpretazione di matrice economica che fa riferimento, quasi esclusivo, al presidio di risorse scarse come fonte di vantaggio competitivo.²⁴ In tale ottica l'eterogeneità rispetto ai competitori deriva quasi esclusivamente dalla sezione di risorse distintive e dalla capacità di isolarle dal meccanismo concorrenziale di imitazione. In realtà uno dei contributi fondanti dell'analisi basata sulle risorse, quello di Penrose (1959), osserva che l'impresa è interpretabile come una collezione di risorse da cui estrarre servizi per le attività produttive. L'abilità dell'impresa, dunque, non è tanto nella specificità delle risorse, quanto nell'estrarre dalle risorse un valore superiore rispetto al costo d'acquisizione delle stesse.²⁵ Sebbene, dunque, la prospettiva sia ancora prevalentemente economica, la visione originale penrosiana fa riferimento a risorse non necessariamente scarse, né tanto meno a risorse forzatamente distribuite in modo eterogeneo tra gli operatori, ma piuttosto fa riferimento a impieghi eterogenei delle risorse. Dobbiamo sottolineare che, anche in situazione di omogeneità di risorse, le performance delle aziende possono differire a causa della specificità dei processi organizzativi posti in essere per combinare le risorse stesse e a causa della diversità nella condotta definita e implementata dal management. Questi fattori evidenziano un chiaro riferimento alla presenza di una visione *behaviourista* dei processi economici, che sposta l'attenzione sui fattori organizzativi. Tale considerazione, tuttavia, secondo l'opinione di alcuni studiosi,²⁶ non esclude la visione di matrice economica, ma rende la prospettiva *resource-based* un filone di studi in grado di sintetizzare interpretazioni economiche e organizzative del concetto di eterogeneità delle imprese attraverso l'analisi delle competenze aziendali.²⁷

²³ Boccardelli, P. (2008). *Strategie e modelli di business nell'industria del video entertainment: capitale umano, capitale relazionale e performance nel comparto cinematografico*. Il Mulino. Ricerca. Il Mulino.

²⁴ Boschetti (1999).

²⁵ Barney (1986).

²⁶ Boschetti e Lipparini (1998).

²⁷ Boccardelli, P. (2008). *Strategie e modelli di business nell'industria del video entertainment: capitale umano, capitale relazionale e performance nel comparto cinematografico*. Il Mulino. Ricerca. Il Mulino.

3.1 La risorsa open source

Dopo aver trattato le basi della teoria economica, dobbiamo capire come l'open source si inserisce in tale panorama.

Possiamo affermare che l'open source non è una risorsa con le “tradizionali” caratteristiche per generare un vantaggio competitivo. In particolare, si può capire come la *scarsità* e l'*inimitabilità* siano due fattori importanti, che mancano alla risorsa open source. Discorso diverso si deve fare per *la rilevanza* e l'*appropriabilità*, che non possono essere escluse a priori; analizziamoli, quindi, separatamente.

La rilevanza non può essere esclusa perché dipende dal mercato in cui opera l'impresa, quindi la stessa risorsa può essere rilevante per un'impresa e non rilevante per un'altra.

L'appropriabilità, come abbiamo detto in precedenza, riguarda la possibilità di acquisire un controllo proprietario sulla risorsa, così da escludere i concorrenti dalla disponibilità di essa. Il modo più semplice per fare ciò è disporre di un brevetto per tale risorsa. Per quanto riguarda la brevettabilità delle licenze open source sappiamo, per quello che si è detto precedentemente sui software, che è un tema molto rilevante, dove al momento vi è una disparità tra Europa e Stati Uniti. Infatti, nella prima, non viene permesso il brevetto sui software ma sono tutelati dal diritto d'autore, invece, in America è permesso anche il brevetto oltre al diritto d'autore. Si capisce che le conclusioni rispetto all'appropriabilità della risorsa open source dipendono dagli strumenti legali di protezione delle innovazioni che un'impresa può adottare.

Però, il brevetto non è l'unico modo per acquisire un vantaggio competitivo in quest'ambito, infatti, come abbiamo già detto, le “Relationship con le comunità

OSS” sono per un’impresa un’altra leva su cui poter fare affidamento e che possono permettere di “acquistare un controllo della risorsa”. Anche se non c’è un controllo proprietario dell’impresa sul progetto open source, essa può essere comunque molto influente sullo sviluppo dello stesso, così da rendere le relazioni con la comunità uno strumento alternativo di appropriabilità della risorsa OS, che potrà portare ad un vantaggio competitivo.

Quindi, diciamo che l’appropriabilità di una risorsa open source, per i concetti che abbiamo discusso, dipende principalmente da due fattori: il primo, e il più importante, consiste nel tipo di controllo proprietario che si adotta, infatti le imprese possono in alcuni paesi (Stati Uniti e Giappone) rendere la risorsa “propria” tramite il brevetto, oppure, possono usufruire di alcune licenze OS che permettono di utilizzare risorse open source per prodotti commerciali delle imprese (questo avviene anche in Europa). Il secondo fattore consiste nello sviluppare delle relationship con le comunità open source, tali da poter permettere all’impresa un certo grado di controllo sullo sviluppo del progetto OS. Inoltre, riguardo a questo fattore assume particolare rilevanza la strategia “*first mover*”, che consiste nel *muoversi per primi* per far percepire un nuovo progetto open source come un’integrazione indissolubile del prodotto offerto in cui il software OS sarà inglobato. Un esempio ci è fornito da Nokia, che ha aperto le porte al mondo open source. La società finlandese ha inserito nel suo ultimo smartphone un sistema operativo Open Source denominato *Maemo 5*, dove il software è stato creato da Linux con Licenza GNU GPL. Da notare che il sistema operativo Maemo è stato preferito da Nokia al loro sistema operativo proprietario storico Symbian. Nokia precisa comunque che lo sviluppo di Maemo non significa l’abbandono di Symbian, il sistema operativo dei suoi smartphone già esistenti. In questo caso potrebbe trattarsi di una fase di transizione, in attesa di vedere se e quanto il nuovo sistema operativo open source avrà successo e, in questo caso, se potrebbe diventare uno standard per gli smartphone Nokia, risparmiando alla società tanti costi di Ricerca e Sviluppo dei software.

Nel caso Nokia possiamo notare ciò che abbiamo detto prima, ovvero, la società finlandese muovendosi per prima nell’adottare un nuovo software OS per

gli smartphone sta riuscendo a fare percepire Maemo come il “sistema operativo di Nokia”.²⁸

Quindi, sarà difficile per i concorrenti mettere nei loro prodotti un sistema operativo che è percepito dalla clientela come se fosse di “proprietà” Nokia, perché nel momento in cui tale tecnologia fosse adottata in qualche prodotto concorrente, i clienti prenderebbero tale prodotto come un’imitazione di uno più conosciuto.

La strategia di Nokia sarà seguita da molti altri operatori che stanno sviluppando insieme alle comunità OSS altri sistemi operativi open source per smartphone (Android e LiMo). Infatti, secondo una stima di Juniper Research, un'azienda che si occupa di analisi e previsioni del settore hi-tech e delle telecomunicazioni, il numero di smartphone con sistemi operativi open source venduti in un anno aumenterà dall'attuale di 106 milioni di unità a 223 milioni entro il 2014.²⁹

Come abbiamo visto, le grandi imprese sia produttrici di hardware (per esempio IBM e Dell) che di software (Microsoft) hanno smesso di avere un atteggiamento di chiusura nei confronti del movimento open source e ne hanno compreso la rilevanza, tanto che oggi alcune imprese sponsorizzano o finanziano dei progetti open source al fine di controllarne lo sviluppo. Ma se questa risorsa non ha le “tradizionali” caratteristiche per generare un vantaggio competitivo perché le imprese investono sui progetti open source?

In primo luogo, perché costa poco, permette di risparmiare investimenti in R&S e produce molta pubblicità, differenziando la propria offerta dagli altri operatori. In secondo luogo, perché abbinare l’offerta di OS a quella di un bene principale significa: proporre al consumatore un offerta che soddisfi la domanda di più beni o servizi, consentire la fidelizzazione del cliente, favorire la coesione della filiera competitiva, alimentare una reputazione positiva nei confronti della ricerca e del sostegno alla ricerca. Ormai, le applicazioni open source sono diffuse nel mercato delle ICT come evoluzione di server web, sistemi operativi, browser web e vengono adottati con successo tanto da imprese private quanto da pubbliche amministrazioni. Il valore dell’investimento europeo nello sviluppo di OS, inteso

²⁸ Fonte: <http://www.techup.it>.

²⁹ Fonte: <http://www.juniperresearch.com>.

come costo di ricerca e sviluppo sostenuto da imprese, pubbliche amministrazioni e operatori censiti dall'osservatorio olandese UNU-MERIT, si aggira sui 22 miliardi di euro, rappresentando il 20,5% dell'investimento totale in software.³⁰

Quindi, *l'Open Source può essere considerata un'eccezione alla regola generale sulle risorse*; infatti essa ha un peso economico importante nel mercato dei software. Allora, quali sono le caratteristiche che determinano un vantaggio competitivo in questo tipo di risorsa? Andiamo ad analizzare questo quesito qui di seguito.

3.2 Fattori critici di successo nel mondo open source

Le differenze nelle performance delle imprese che investono nell'Open Source Software (OSS) possono dipendere da una serie di fattori. Alcune imprese sfruttano meglio le risorse che hanno a disposizione, con una superiore capacità di marketing, vendite e distribuzione; altre possono semplicemente generare dei prodotti di qualità superiore, rispetto ai concorrenti.

Un fattore molto importante, che può spiegare perché alcune imprese OSS sono migliori di altre, è *l'attività di "relationship" con le comunità OSS* coinvolte nella creazione di parti del software utilizzato in prodotti e servizi commerciali. Quest'ultimo tema, congiuntamente alla strategia del *"muoversi per primi"*, merita un approfondimento, perché di rilevante importanza in questo contesto.

Prima di affrontare nello specifico le relazioni tra comunità e imprese, appare opportuno chiarire una diffusa convinzione riguardante la *contrarietà delle comunità open source nel collaborare con le imprese* che sviluppano e finanziano i loro progetti con lo scopo di guadagnare anche grazie ad essi. Infatti, lo

³⁰ Barbarito, L. (2008). *Economia dell'open source*. Torino: Pearson Paravia Bruno Mondadori.

sfruttamento del valore finanziario dei software sviluppati congiuntamente va contro i valori di base dell'intero movimento OSS.³¹

È vero che l'idea di base delle comunità OSS non è quella del profitto, ma non per questo esse sono ostili al mercato. Infatti, Raymond (1999) sintetizza alcuni punti attraverso cui la cultura hacker (che ha gettato le basi tanto del movimento *free software* quanto di quello *open source*) interpreta la cooperazione con il mercato. In primo luogo, non esistono in genere obiezioni al fatto che qualcuno possa trarre profitto dal software open source purché questa possibilità sia data a tutti in egual misura, anche se una minoranza della comunità hacker si oppone effettivamente a una logica di profitto (specialmente quelli facenti parte della movimento *free software*). Un secondo tema sensibile nella cultura hacker concerne le conseguenze indesiderate derivanti da licenze troppo restrittive: più si diffondono restrizioni per lo sviluppo del software, maggiore sarà la difficoltà che questo raggiunga una qualità e una distribuzione adeguata.³²

Le comunità rilasciano “*codici sorgente*” in modo da rendere il loro prodotto largamente adottato, dato che una larga approvazione aumenta la probabilità di attrarre degli sviluppatori esperti e quindi di raggiungere un alto livello di sviluppo tecnologico. Inoltre, prodotti largamente diffusi possono ottenere diversi vantaggi se le imprese implementano delle strategie di *first-mover*: le start-up, che per prime si lanciano in un determinato settore di mercato, godono di un decisivo vantaggio sulla concorrenza, fissando così standard tecnologici o raggiungendo una percentuale di mercato notevole. Non si deve dimenticare, comunque, che il termine “*Open Source*” fu coniato per indirizzare il fenomeno verso una strada più attenta alle relazioni con il mercato rispetto a quella percorsa fino a quel momento dal *free software*, come abbiamo ampiamente descritto nel paragrafo “Open source e Software libero”.³³

³¹ Dahlander, L., & Magnusson M.G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34, 481–493.

³² Barbarito, L. (2008). *Economia dell'open source*. Torino: Pearson Paravia Bruno Mondadori.

³³ Dahlander, L., & Magnusson M.G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34, 481–493.

Le considerazioni fin qui fatte acquisiscono particolare rilevanza se si tiene conto del fatto che fonti sia di tipo accademico che di carattere divulgativo sono concordi nel constatare, al giorno d'oggi, una compenetrazione tra il mondo delle comunità e quello del mercato. Fitzgerald (2006) sintetizza con il termine “OSS 2.0” la nuova realtà del mondo Open Source costituita da una sovrapposizione e cooperazione tra le comunità e le imprese.

3.3 Relationship tra imprese e comunità OSS

Per illustrare le differenti logiche che esistono tra le aziende e le comunità, faremo riferimento alla tassonomia usata da Feller e Fitzgerald, distinguendo tra fattori motivazionali di tipo economico, sociale e tecnologico.

La seguente tabella riassume le principali differenze motivazionali tra imprese e comunità.

Tabella 1

Fattori motivazionali	Imprese	Comunità
Economico	Sviluppo progressivo e maggiore competitività (West, 2003) Riduzione dei costi	Ricompense monetarie (Hertel et al., 2003) Costi opportunità bassi (Lakhani e Von Hippel, 2003)
Sociale	Condivisione dei codici con la comunità (Lerner e Tirole, 2002) Percezione che il software sia libero (Feller e Fitzgerald, 2002)	Appartenenza alla comunità (Raymond, 1999) Sfide intellettuali (Himanen, 2001; Raymond, 1999) Altruismo o reciprocità generale (Raymond, 1999)

Tecnologico	Utilizzo del feedback (Lerner e Tirole, 2005)	Apprendimento (Lakhani e Von Hippel, 2003)
	Promozione di standard	Feedback e assistenza (Raymond, 1999) Lavorare con nuove tecnologie

Nota: Questa tavola è stata fortemente influenzata da Bonaccorsi e Rossi (2003).

Una caratteristica particolare dei software open source consiste nel fatto che le risorse importanti non sono direttamente controllate dalle aziende, ma in parte risiedono nelle comunità che coesistono con le aziende. Malgrado ciò, le aziende provano esplicitamente a utilizzare le risorse in comune al fine di creare valore e appropriarsene.

Nelle aziende, la relazione tra uno sviluppatore e il loro datore di lavoro è regolato da un contratto. Di conseguenza, questi ricercatori impiegati nelle imprese ricevono uno stipendio e altri tipi di compensi finanziari. Nei progetti open source le relazioni molto informali, ognuno è libero di diventare membro e per il lavoro prestato non si riceve, normalmente, un corrispettivo economico. Aiutare gli altri può aumentare la propria autostima, permette di dimostrare la propria competenza tecnica e fa guadagnare rispetto e prestigio. Questo implica che i partecipanti dei progetti open source diventano beneficiari del pubblico in quanto si preoccupano del sistema. In aggiunta, Lerner e Tirole suggeriscono che i contributori sono spinti da una motivazione relativa alla loro futura carriera, in quanto la partecipazione a una comunità di sviluppo può essere utile ad un soggetto nella ricerca di occupazione. Malgrado questo, sembra plausibile suggerire che è più probabile che i membri di una comunità che sviluppano software open source siano motivati da fattori sociali.

Bonaccorsi e Rossi, mediante un'analisi di 146 aziende che producono prodotti e servizi OSS, hanno scoperto che le aziende erano principalmente motivate da fattori tecnologici ed economici, piuttosto che da fattori sociali.

La creazione di un software solido all'interno di una azienda è più difficoltosa rispetto a quello creato da una comunità OSS; infatti, gli sviluppatori di un'azienda che lavorano ad un progetto commerciale sono pochi rispetto a quelli delle comunità che lavorano in un progetto open source solido e conosciuto. Per converso, il coordinamento del lavoro cooperativo all'interno di una comunità può presentare delle difficoltà e delle inefficienze di tempo e di sforzi da parte dei suoi sviluppatori. Quindi, l'impresa che riesce a indurre o generare nella comunità una “*gestione aziendale*” ha trovato un modo per mantenere l'equilibrio giusto; dove, l'influenza esercitata dall'impresa sulla comunità che le permette di avere un certo controllo del progetto stesso e di modificare in modo strategico lo sviluppo del software rappresenta un fattore importante nella *gestione delle relazioni con la comunità OSS*.³⁴

È lecito chiedersi se sia possibile generare l'interesse e la creatività della comunità nel caso ci sia troppo controllo da parte dell'impresa. Ma è anche vero che con troppo pochi controlli e direttive gli effetti degli investimenti possono essere piccoli, o persino contro-produttivi, nel caso gli obiettivi della comunità lavorino contro l'organizzazione. Quindi, assume particolare rilevanza il *grado di controllo* esercitato dall'impresa, che non deve essere mai troppo elevato da far disinteressare gli sviluppatori delle comunità e non deve essere troppo lieve da non riuscire ad indirizzare lo sviluppo del software.

Esistono diversi *metodi di gestione delle relazioni* tra azienda e comunità OSS. I tre approcci più importanti che vengono utilizzati nella letteratura economica e dalle aziende sono: *symbiotic, commensalistic, parasitic*.

L'*approccio “symbiotic”* implica che l'azienda è focalizzata nel realizzare dei benefici comuni sia per l'azienda stessa che per la sua comunità open source. Mentre l'ambizione delle aziende nell'influenzare le comunità è, in questo caso, significativamente bassa; i valori e le norme delle OSS sono rispettati e presi in considerazione. In parte una influenza minore nella direzione dello sviluppo deriva da una partecipazione attiva nei vari progetti, ma non c'è un co-allineamento strategico tra le imprese e le comunità. Le aziende che adottano

³⁴ Dahlander, L., & Magnusson M.G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34, 481–493.

questo approccio non si limitano a collaborare con il mondo open source per il progetto principale svolto nelle comunità con cui sono in relazione, ma cercano di ricambiare questi benefici sviluppando altri prodotti. Un modo per fare ciò è che le imprese decidono di fornire i codici che hanno sviluppato alle comunità, questo può avvenire per prodotti che sono in fase di declino o per progetti molto complessi. Un altro modo per far percepire l'impresa come un'entità utile alla comunità open source è fornire agli sviluppatori OSS una infrastruttura ben funzionante, che facilita l'esecuzione dei differenti compiti di sviluppo e tiene conto di un'interazione stimolante. Inoltre, la presenza di persone di spicco nelle attività OSS è stato storicamente un fattore importante per aumentare la reputazione delle aziende all'interno delle comunità, anche se le relazioni tra queste imprese e le loro comunità in una certa misura si sono deteriorati nel tempo in quanto le imprese sono sempre più orientate al commercio.

L'approccio "*parasitic*" implica che l'azienda si focalizza solo sui propri vantaggi, senza prendere in considerazione il fatto che le sue azioni potrebbero danneggiare la comunità.

Una via intermedia per mettersi in relazione con la comunità è quello di utilizzare un approccio "*commensalistic*", vale a dire di beneficiare della coesistenza con un altro soggetto, pur lasciandolo senza danni. L'idea di base in questo contesto specifico è quello di prosperare sulle risorse comuni che vengono continuamente elaborate, pur mantenendo minimo il coinvolgimento diretto nello sviluppo di tali risorse.

Un rischio collegato all'approccio *commensalistic* è che, col passare del tempo, si trasforma in una relazione parassitaria, dove l'impresa può essere percepita negativamente dalla comunità sia in termini di violazioni delle norme di base, principi e valori che per un atteggiamento "*free rider*", cioè beneficia di un'attività collettiva senza parteciparvi attivamente.

È chiaro che nessuna impresa di OSS sceglierebbe deliberatamente un metodo parassitario, in quanto causare danno alla comunità che la ditta sostiene non sembra essere un modello aziendale sostenibile. Tuttavia, definite le differenze fondamentali fra i fondamenti logici delle diverse parti, *la linea che separa l'approccio commensalistic da quello parasitic è sottile e non è sempre*

chiara. Riassumiamo i vari approcci in una tabella esplicativa nella seguente pagina (tabella 2).³⁵

Possiamo distinguere sette principali punti su cui si basa la *gestione della relationship con la comunità*:

1. *Rispettare le norme e i principi delle comunità OSS:*

Insieme ai meccanismi legali (licenze in primo luogo), il lavoro comune è protetto da norme e valori sociali che sono diffusi tra gli utenti e gli sviluppatori. Malgrado le difficoltà che ci sono nell'influenzare le norme e i valori delle comunità, alcuni tentativi in tal senso vi sono stati da parte di imprese che affidavano il compito a persone di spicco per avere una maggiore

tabella 2

	Symbiotic	Commensalistic	Parasitic
	<i>Vantaggioso per imprese e comunità</i>	Vantaggioso per imprese e indifferente per comunità	Vantaggioso per imprese e svantaggioso per comunità
Natura delle relazioni	Dare e ricevere vantaggi da un rapporto stabile tra impresa e comunità	Ricerca di un contributo utile dalla comunità	Ricerca di un contributo utile senza obbedire a norme, valori e regole
Possibilità di influenzare la comunità	Alta	Bassa	Nessuna
Sfide manageriali	Rispettare norme, valori e licenze Cercare di utilizzare le risorse sviluppate dalla comunità	Rispettare norme, valori e licenze Ottenere l'accettazione della comunità per utilizzare le sue risorse in prodotti commerciali	Evitare conflitti diretti

³⁵ Dahlander, L., & Magnusson M.G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34, 481–493.

Strumenti strategici di controllo	Attrarre sviluppatori Allineamento dei diversi interessi Risoluzione della ambiguità relativa al controllo - proprietà Destinare personale allo sviluppo del lavoro nella comunità Creare e mantenere una reputazione Fringe benefit Strumenti di interazione Vendita dei prodotti sviluppati	Destinare personale allo sviluppo del lavoro nella comunità
--	--	---

possibilità di successo. Questi individui sembrano, almeno in certi casi, aver la capacità di influenzare la comunità, poiché sono conosciuti e rispettati da essa. Le aziende che hanno fondato una comunità sono state anche attive nel creare eventi sociali, come fiere e workshop per utenti e sviluppatori. In linea con questo argomento, O'Mahony e Ferraro hanno trovato che le interazione faccia a faccia sono cruciali nel gestire i limiti dei progetti aperti. A quanto pare, gli eventi sociali sono un altro mezzo di plasmare norme e valori sociali, e per far accettare l'uso commerciale della conoscenza creata dalla comunità.

2. Usare le licenze in maniera fruttuosa:

Gestire le diverse licenze che impongono il modo in cui i software derivanti dai progetti open source possono essere usati è un tema centrale per le aziende.³⁶ Infatti, la licenza, oltre a decidere il modo in cui un software dovrebbe essere usato, ha anche un notevole valore simbolico.³⁷ In questo contesto le imprese devono affrontare un problema rilevante per poter usare i software già esistenti sviluppati dalle comunità OSS. Infatti, quando quest'ultime sviluppano i loro

³⁶ Raymond (1999).

³⁷ Lerner e Tirole (2005).

software, proteggono le licenze da variazioni che le aziende potrebbero effettuare sul copyright. Ciò limita a sua volta la possibilità per le imprese di usare tali software insieme a codici sorgente sviluppati dall'azienda stessa.

3. Attrarre sviluppatori e utenti:

Una sfida ovvia per le aziende OSS è di attrarre non solo i clienti, ma anche sviluppatori che possono contribuire allo sviluppo di nuovi software. Anche se numerosi studi hanno sottolineato i benefici delle comunità nel domare la complessità, soddisfare le esigenze degli utenti eterogenei, consentire la possibilità di “bug reporting” e lo sviluppo di nuovi codici; la creazione di un *Open Source Software* non implica che gli utenti e gli sviluppatori sono automaticamente interessati allo sviluppo del progetto, questo perché un gran numero di progetti competono per attirare l'attenzione e gli interessi di sviluppatori e utenti. Infatti, poiché gli sviluppatori sono motivati da fattori sociali, le aziende devono fornire progetti divertenti e stimolanti sfide per attrarre gli utenti e gli sviluppatori.

4. Gestire le risorse impiegate nello sviluppo della comunità:

Al fine di creare e mantenere i rapporti con le comunità, le imprese investono notevoli risorse (tempo e denaro). Ad esempio, investono del denaro per la costruzione di infrastrutture utili allo sviluppo del progetto, oppure, possono organizzare eventi sociali per le persone che lavorano nelle comunità open source.

5. Allineare i diversi interessi in base alla natura del progetto:

Lavorare strettamente con la comunità implica che l'impresa deve essere in grado di allineare i diversi interessi con la natura del lavoro. Siccome l'obiettivo primario di sviluppare il prodotto insieme alla comunità cambia nella vendita dello stesso ai propri clienti, probabilmente l'azienda risconterà una crescente difficoltà a lavorare a stretto contatto con la comunità.

6. *Risolvere l'ambiguità sul controllo e la proprietà del progetto:*

Le aziende che sono attive nella creazione di nuovi progetti open source hanno bisogno di risolvere l'ambiguità che riguarda la proprietà e il controllo dello stesso. Vari studi dimostrano che la proprietà dei progetti è un fattore critico per suscitare interesse negli sviluppatori e per non entrare in conflitto con la comunità OSS.³⁸ Di conseguenza, il coinvolgimento dell'impresa, in una certa misura, ostacola la possibilità per la comunità di avere il desiderato controllo sul progetto.

7. *Essere accettati per usare i software sviluppati dalla comunità in prodotti commerciali ed evitare conflitti diretti:*

Le comunità dipendono anche dalle innovazioni in via di sviluppo e condivise da altri, le imprese che usano i software di una comunità open source devono essere accettati per poter usare tali software in applicazioni commerciali e per evitare conflitti diretti. Quindi, le aziende devono cercare di farsi accettare dalla comunità per disporre liberamente delle risorse in comune. Un modo per fare ciò consiste nell'essere chiari sui pezzi di software, open source e proprietari, che l'azienda utilizzerà per il prodotto finale, e su come il "codice aperto" è usato congiuntamente ai "codici chiusi" sviluppati internamente dall'impresa.³⁹

3.4 Strumenti di gestione per il controllo delle relazioni con la comunità.

³⁸ Raymond (1999); O'Mahony (2002); O'Mahony (2003); West e O'Mahony (2004)

³⁹ Dahlander, L., & Magnusson M.G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34, 481–493.

Le imprese non si basano sul controllo diretto di sviluppatori e utenti di un movimento OSS, in quanto non vi è alcun rapporto formale con loro. Tuttavia, vengono utilizzati mezzi strategici di controllo che mirano a influenzare la comunità in una certa direzione. Si possono distinguere cinque meccanismi attraverso i quali ciò può essere realizzato:

1. *Dedicare personale dell'impresa allo sviluppo del lavoro o impiegare personale direttamente all'interno della comunità OSS;*
2. *Creare e mantenere una certa reputazione;*
3. *Fornire "fringe benefits";*
4. *Utilizzare strumenti di interazione;*
5. *Rendere interessanti i progetti da sviluppare.*

In primo luogo, dedicare personale dell'impresa allo sviluppo del lavoro o impiegare personale direttamente all'interno della comunità OSS è stato considerato un mezzo di controllo strategico; infatti, per i rappresentanti dell'azienda, lavorare come colleghi nei progetti open source costituisce una possibilità di conoscere anticipatamente quali saranno i progressi di tale software e talvolta può portare ad influenzare le decisioni. Il personale qualificato può accaparrarsi una buona reputazione agli occhi dei colleghi della comunità OSS, migliorando di conseguenza l'immagine dell'azienda.

In secondo luogo, *creare e mantenere una certa reputazione* è un importante meccanismo di controllo. I rappresentanti dell'azienda all'interno delle OSS, conosciuti e rispettati dalle comunità, hanno alte probabilità di influenzare le attività di sviluppo svolte all'interno della comunità rispetto a quelli meno conosciuti. La reputazione è importante anche da un altro punto di vista, vale a dire in termini di riconoscimento di persone qualificate, costituendo un fattore motivante per le persone a partecipare. Le imprese sembrano essere consapevoli del fatto che dare credito a persone che contribuiscono al "*bug-hunting*", con nuovi pezzi di codice e con le traduzioni è un fattore di vitale importanza. Di

conseguenza, tutte le aziende osservate con le loro comunità fanno uso di questo specifico meccanismo di controllo.

In terzo luogo, i diversi tipi di *benefici accessori* sono usati per favorire un certo tipo di comportamento. Questi vengono utilizzati solo quando le imprese e i loro prodotti sono correlati in una relazione stabile con una comunità OSS. A volte le aziende mettono a disposizione CD o attrezzature informatiche per un determinato compito. Un altro tipo di fringe benefits è quello di ottenere l'accesso alle nuove versioni prima di altri. Tuttavia, le aziende hanno difficoltà a creare un forte interesse degli sviluppatori usando tali *fringe benefits*.

Quarto, gli “*strumenti di interazione*” sono servizi che gli sviluppatori e gli utenti possono utilizzare (come forum on-line, mailing-list, etc.); in parte, questo concetto è in relazione con il quello di innovazione dei mezzi, che ha a che fare con la gestione delle infrastrutture nella relazione tra l'azienda e la comunità. Se i mezzi di interazione sono ben definiti, essi possono costituire una funzione sociale, in quanto tengono conto dell'interazione tra i differenti sviluppatori o tra le persone e gli sviluppatori. L'interazione non è necessariamente limitata allo spazio virtuale, ma può anche manifestarsi mediante meeting nel mondo reale. Anche in queste riunioni, le aziende OSS hanno l'opportunità di influenzare più o meno indirettamente la direzione dello sviluppo del progetto stesso.

Quinto, *rendere interessanti i progetti da sviluppare* è un possibile modo di influenzare una comunità, infatti i progetti che propongono le imprese devono attirare l'interesse di sviluppatori e utenti, in quanto alcuni sviluppatori che lavorano nelle comunità OSS sono motivati da sfide intellettuali.⁴⁰

3.4 I modelli di Business dell'Open Source

⁴⁰ Dahlander, L., & Magnusson M.G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34, 481–493.

Dopo aver descritto i fattori che possono costituire il vantaggio competitivo per l'impresa, che riesce a sfruttare la risorsa Open Source nel modo migliore, analizziamo come dal passaggio dal software commerciale al software open source è solo il valore di vendita ad essere variato, teoricamente azzerato, mentre invece il valore d'uso viene completamente preservato. Se quindi lo sviluppo con il *modello Open Source* è veramente più efficace e permette di ottenere un prodotto di qualità più elevata rispetto a quello commerciale, allora diventa plausibile che il solo valore d'uso sia sufficiente per finanziare lo sviluppo dell'open source in modo sostenibile.

In effetti non è difficile individuare almeno due casi di studio che hanno dimostrato la capacità del software OS di sostenersi dal punto di vista economico. La storia del web server *Apache* generalizza un modello in cui gli utenti di software trovano vantaggioso sostenere lo sviluppo dell'open source perché, così facendo, ottengono un prodotto migliore a un costo inferiore. L'open source serve non tanto ad abbassare i costi, quanto a ripartire il rischio. Tutte le parti in gioco si accorgono che la disponibilità del codice sorgente, così come la presenza di una comunità che agisce in collaborazione con gli altri soggetti, finanziata da più fonti di reddito indipendenti, fornisce una garanzia di sopravvivenza che ha un valore economico intrinseco o, comunque, un valore sufficiente per finanziarla.

L'open source rende piuttosto difficile trarre un valore di vendita diretto dai software; la difficoltà sta nella natura del contratto sociale che si trova alla base dello sviluppo "*Open*". Per tre motivi che si rinforzano a vicenda, la principale licenza Open Source (la GPL) proibisce la maggior parte delle forme di restrizione d'uso, redistribuzione e modifica che faciliterebbero la riscossione di profitti derivanti dal valore di vendita. Per comprendere questi motivi, occorre esaminare il contesto sociale nel quale si sono evolute le licenze: *la cultura hacker di Internet*. Come già detto, se è vero che una minoranza di hacker resta ancora ostile alla logica del profitto, la volontà generalizzata da parte della comunità di cooperare con aziende, che per esempio basano i loro prodotti su Linux come nel caso della Red Hat, dimostra che la maggioranza degli hacker sarebbe felice di lavorare a fianco del mondo dell'impresa, quando tutto ciò è nei

loro interessi. Le ragioni reali per cui gli “hacker” guardano con scetticismo alle licenze che consentono guadagni diretti, sono assai più sottili e interessanti.

La prima ragione riguarda la *simmetria*, infatti mentre la maggior parte degli sviluppatori open source non oppone obiezioni sostanziali al fatto che altri traggano profitto dai loro lavori, i più chiedono che nessuno (ad eccezione, eventualmente, del creatore di una parte del codice) si trovi in una posizione privilegiata per fare profitti.

La seconda ragione riguarda le *conseguenze indesiderate*. Gli hacker hanno osservato che le licenze che comportano la restrizione e il pagamento di contributi per uso "commerciale" (il tentativo più comune e, a prima vista, nemmeno tanto assurdo, di recuperare il valore di vendita diretto) hanno effetti davvero inquietanti. Le restrizioni sull'uso, la vendita, le modifiche, la distribuzione (e altre clausole presenti sulle licenze) comportano spese aggiuntive per la conformità della distribuzione e (con l'aumento dei pacchetti in circolazione e in utilizzo) una combinazione esplosiva di incertezza soggettiva e potenziali rischi legali. Poiché questi risultati si considerano dannosi, si registra una forte pressione sociale per mantenere le licenze semplici ed esenti da restrizioni.

L'ultima e la più importante ragione riguarda la *dinamica di revisione e di cultura del “dono”* descritta nel libro di Raymond. Le restrizioni sulle licenze, designate per proteggere la proprietà intellettuale o per percepire un valore di vendita, hanno spesso l'effetto di rendere legalmente impossibile la *biforcazione dei progetti*, ovvero quando un progetto si divide in due sotto-progetti per varie motivazioni. Spesso gli hacker motivano l'antipatia per la scissione dei progetti lamentando l'inutile duplicazione di lavoro che ciò porterebbe, con i progetti-figli che finirebbero per seguire un'evoluzione futura più o meno parallela. Si potrebbe inoltre osservare che la biforcazione tende a dividere la comunità dei co-sviluppatori, lasciando entrambi i progetti-figli con un minor numero di cervelli da utilizzare rispetto al progetto-madre. Se la biforcazione incontra molto scetticismo ed è considerata come l'ultima risorsa, la presenza di quest'ultima risorsa si considera di capitale importanza in caso di incompetenza o defezione del gestore (a causa, per esempio, di una licenza troppo restrittiva).

La comunità hacker mostra, invece, una certa elasticità in materia di simmetria: per questo, tollera licenze, come NPL di Netscape, che concedono alcuni privilegi in materia di profitti agli autori originali del codice (nel caso specifico di NPL, il diritto esclusivo di utilizzare il codice Open Source di Mozilla in prodotti derivati, anche commerciali). Ma ha meno elasticità nei confronti delle conseguenze indesiderate e nessuna sul mantenimento della possibilità di biforcazione (ed è per questo che gli schemi elaborati dalla Sun per la "Community License" di Java e Jini sono stati in gran parte rifiutati dalla comunità).

Queste ragioni spiegano le clausole della Open Source Definition, scritta per esprimere il parere congiunto della comunità hacker sui requisiti essenziali delle licenze standard (GPL, licenza BSD, licenza MIT e Licenza Artistica). Queste clausole hanno l'effetto (ma non l'intenzione) di rendere molto difficile l'ottenimento di un valore di vendita. Nonostante ciò, esistono modi per costruire mercati nei servizi legati al software che comportino un valore di vendita indiretto.

Sono *cinque i modelli conosciuti*, come presentati dal libro di Raymond (ma se ne potranno sviluppare altri in futuro):

1. Vendita di servizi

Si tratta del modello più semplice, i venditori di servizi software sono aziende che vendono sia il software che i servizi di supporto al cliente: configurazione ed installazione, mantenimento evolutivo, supporto post-vendita e garanzia. Nel commercio tradizionale del software "chiuso", il reddito che deriva da vendite del software è pari al 60% del totale, dal supporto arriva il rimanente 40%. Un modello Open Source sposta radicalmente il ricavo dalle vendite di licenze software al 0% mentre il supporto sale al 100%. Poiché il software è open source, l'azienda trarrà beneficio dalle esternalità positive che derivano dal "*bug fixing*" (processo di determinazione degli errori) e dalle nuove funzionalità implementate dagli utenti. Quindi in questo modello, si rende disponibile il codice sorgente di un software per creare una nicchia di mercato non nel settore del software commerciale, bensì nei servizi. Come esempio possiamo citare *Red Hat*

e altri distributori di Linux. Ciò che vendono, in realtà, non sono i software, ma il valore aggiunto nell'assemblaggio e nelle prove di un sistema operativo funzionante e con garanzie (anche solo implicite) di commerciabilità e compatibilità con altri sistemi operativi dello stesso *brand*. Altri elementi del valore da essi offerti comprendono l'assistenza gratuita per l'installazione e l'offerta di opzioni per contratti di assistenza continuativi.

Come ulteriore esempio riportiamo il caso di *Zope*, un programma per la creazione di siti web, sviluppato dalla Digital Creations. Quest'ultima, autrice di questo programma molto valido tecnicamente, contattò un esperto di venture capital per raccogliere fondi e poter crescere. L'esperto dopo una attenta analisi consigliò di pubblicare come Open Source tutta la suite. Stando agli standard tradizionali dell'industria del software, questa sembrerebbe una mossa completamente folle. La saggezza convenzionale, acquisita dalla scuola economica, ci insegna che una proprietà intellettuale vitale come questa rappresenta la punta di diamante di un'impresa ed essendo una "risorsa competitiva" non deve essere venduta in alcun caso o, ancora peggio, non deve essere regalata ai concorrenti tramite l'apertura del codice. Ma l'esperto aveva due argomenti a sostegno della sua strategia: il primo, che il vero patrimonio di *Zope*, in realtà, erano i cervelli e le abilità delle persone in esso impegnate; il secondo, che probabilmente, *Zope* avrebbe prodotto più valore nel costruire un mercato che come strumento segreto di un'azienda. Per capire tutto questo, paragonate le due situazioni. In quella convenzionale, *Zope* rimane l'arma segreta della Digital Creations. Ammettiamo pure che sia molto efficace. Il risultato è che l'impresa sarà in grado di ottenere una qualità superiore in tempi brevi, ma nessuno lo saprà. Sarà facile soddisfare i clienti, ma più difficile sarà iniziare ad avere una clientela di base. L'esperto capì che uno *Zope* Open Source avrebbe rappresentato una pubblicità importantissima per il vero patrimonio della Digital Creations: le persone. Ipotizzò che i clienti, valutando le prestazioni di *Zope*, avrebbero considerato più efficace contattare gli esperti, piuttosto che sviluppare una gestione interna di *Zope*. Da allora, uno dei rappresentanti di *Zope* conferma pubblicamente che la *strategia open source ha "aperto molte porte che non si sarebbero trovate altrimenti"*. Infatti, i potenziali clienti hanno risposto con forte

interesse alla logica sottesa a questa strategia e come previsto dall'esperto la Digital Creations è in auge.

2. *Articoli civetta*

In questo modello, si pubblica gratuitamente una versione software open source per creare pubblicità o difendere una posizione sul mercato, mentre invece i profitti vengono ottenuti attraverso una successiva versione proprietaria del software. Nella variante più comune, la versione open source ha capacità o modi d'uso ridotti rispetto alla versione commerciale. Oppure la versione OS realizza solo la funzione utente del software, lasciando alla versione commerciale le ben più lucrose attività di server centrale. Come esempio, in cui un software client "libero" agevola le vendite di un software server, possiamo citare il caso di *Netscape/Mozilla*; in questo caso avremo un software open source di comunicazione di Voce su Internet (VoIP), realizzato dall'azienda *XTEN*, che deve essere installato sul PC per essere utilizzato in modo limitato, infatti la versione OS non dà la possibilità di instaurare videoconferenze o fruire di un servizio di Instant Messaging, servizi disponibili solamente nella versione commerciale del software. *XTEN* commercializza inoltre centralini VoIP e sistemi di videoconferenza, ovviamente venuti attraverso licenze commerciali. *Il successo della versione Open Source ha permesso ad XTEN significativi vantaggi commerciali, azzerando di fatto le spese di marketing per la pubblicità.* A volte il profitto consiste solamente da abbonamenti e pubblicità in relazione al portale Web da cui si scarica il software OS.

3. *Perdita della leadership*

La decisione di una azienda di rilasciare il codice sorgente in modalità Open Source può dipendere anche da scelte strategiche di marketing. Come esempio citiamo la Netscape Communications che all'inizio del 1998, rese disponibile il codice sorgente del browser Mozilla.

La decisione fu dovuta dal calo del giro d'affari legato al browser che ammontava al 13% dei guadagni, in seguito al lancio del browser Internet Explorer da parte della Microsoft, che lo vendeva (e tuttora lo vende) soprattutto in maniera congiunta con il sistema operativo. Netscape temeva che la Microsoft intendesse monopolizzare il mercato dei browser, per poi sfruttare il proprio controllo sull'HTML, per eliminare Netscape dal mercato server. Rendendo disponibile il codice sorgente del browser Netscape, ancora assai diffuso, riuscirono a negare alla Microsoft la possibilità di istituire un monopolio sul mercato dei browser, accelerando lo sviluppo del browser Netscape attraverso lo spirito di collaborazione con il "mondo" Open Source. Tramite questa strategia Netscape ha inoltre costretto la Microsoft a tenere il passo e non le ha permesso la definizione dell'HTML in modo esclusivo per il browser Internet Explorer.

4. *"Widget frosting"*

Questo modello riguarda i produttori di hardware. Le pressioni del mercato hanno costretto le società che producono hardware a scrivere e gestire software (dai device driver agli strumenti di configurazione, fino a interi sistemi operativi), ma il software stesso non è fonte di profitti, bensì una spesa aggiuntiva, spesso consistente. In questa situazione, la disponibilità del codice sorgente e quindi il modello Open Source non dà molto da pensare all'azienda produttrice. Non ci sono guadagni da perdere, né aspetti negativi, ciò che il produttore "acquista" dall'Open Source è una risorsa di sviluppo di gran lunga maggiore, una risposta più rapida e flessibile alle esigenze dei clienti e di maggiore affidabilità, grazie alla possibilità di revisione reciproca. Inoltre, produce pubblicità gratuita, "apre nuove porte che non si sarebbero trovate altrimenti" e, probabilmente, accresce anche la fedeltà dei clienti, poiché il personale tecnico dedica più tempo al codice per esaudire le richieste di personalizzazione della clientela.

I produttori, solitamente, sollevano alcune obiezioni specifiche nei confronti della disponibilità del codice sorgente per i driver hardware. Il margine di sicurezza dell'open source è particolarmente evidente nel campo del *debugging di elementi d'interfaccia ("widget frosting")*. I prodotti hardware hanno produzione e tempi di assistenza limitati, scaduti i quali, i clienti si ritrovano da soli; ma se

hanno accesso al codice di altri driver, con la possibilità di utilizzarli per produrre patches secondo il bisogno, è più probabile che rimangano clienti della stessa società. Un esempio assai illuminante dell'adozione del "widget frosting" ci è fornito dalla decisione da parte di Apple Computer, a metà marzo del 1999, di rendere disponibile il codice sorgente di "*Darwin*", il fulcro del sistema operativo *MacOSX server*.

5. *Vendere accessori*

In questo modello, si vendono accessori per il software open source. Dalle tazze da tè e magliette, magari con l'effigie del pinguino di Linux, alla documentazione redatta e prodotta a livello professionale. La *ÒReilly Associates*, editrice di molti manuali eccellenti sul software open source, è un ottimo esempio di impresa fornitrice di accessori. Per la precisione, *ÒReilly* impiega famosi hacker operanti nell'open source (tra cui Larry Wall e Brian Behlendorf), come metodo per costruirsi una reputazione nel mercato di riferimento. Come altro esempio possiamo nuovamente citare la *RedHat* che vende, per una cifra tra i 50 e gli 80 dollari, un CD contenente la sua installazione ufficiale del software Linux; ed include un manuale e il supporto tecnico degli esperti di RedHat per 90 giorni. Tuttavia, l'intero sistema, compreso il manuale, può anche essere scaricato dal loro Website gratuitamente, o persino essere comprato assieme ad una rivista per pochi dollari. Redhat realmente non vende i programmi: vende "*il buon fattore di tatto*" di software già imballato con tanto di manualistica e pacchetto di supporto post-vendita. RedHat sta andando bene finora ed è entrata in affari con le grandi industrie come l'IBM e Dell.⁴¹

⁴¹ <http://www.puntocontatto.it>

Conclusioni e prospettive

In questo lavoro abbiamo presentato il mondo dell'e-business per poi trattare principalmente il fenomeno open source. Quest'ultimo è nato e si è sviluppato grazie a Internet diventando, al giorno d'oggi, sempre più importante.

Il fenomeno open source rientra in quel corollario di strumenti che l'e-business mette a disposizione delle imprese. Infatti, come abbiamo già detto, "e-business" non significa creare solamente un sito web per fare e-commerce o per gestire le relazioni con i clienti, ma riguarda anche un'insieme di strumenti e funzionalità che un'impresa può implementare grazie alla tecnologia Internet.

Inoltre, dopo la lettura di questa tesi, dovrebbe essere chiaro il processo che un'impresa deve adottare per utilizzare con successo una risorsa open source, ovvero, instaurare delle relationship forti con le comunità open source (che vivono e si sviluppano tramite Internet) dedicando personale allo sviluppo del progetto e

facendo investimenti in macchinari e strumenti che aiutino direttamente le comunità open source. Se poi a questo accoppiamo una strategia “first mover” avremo maggiori probabilità che quel determinato software OS non venga adottato da altre imprese.

Le prospettive del fenomeno open source sono promettenti; infatti per i prossimi anni si prevedono dei forti margini di crescita, soprattutto nel settore dei sistemi operativi (si pensi a Linux, Maemo e Android). In particolare lo sviluppo del mercato degli smartphone e dei Netbook sta aprendo le porte al sistema operativo open source, essendo quest’ultimo molto adatto a dispositivi non tanto potenti.

Fino ad ora si è parlato sempre e solo di software open source, ma un aspetto molto interessante che può costituire una nuova frontiera del mondo OS riguarda l’*hardware open source*.

Per parlare di ciò presentiamo il caso Arduino: un’impresa italiana ha creato il primo hardware open source che consiste in un processore denominato Arduino. Quest’ultimo è stato creato per artisti, designer e chiunque sia interessato a creare oggetti o ambienti interattivi utilizzando un processore, così Arduino si colloca principalmente nel settore dei costruttori di gadget fai-da-te.

Nel 2005 Massimo Banzi, il creatore del processore Arduino, decide di mettere on line gli schemi, i file di progetto e il software per la scheda che servono per produrre e far funzionare il processore, normalmente brevettati per impedire ad altri di poterli utilizzare. In questo modo si rende “Open Source” il processore Arduino, così chiunque può scaricarsi i segreti commerciali e può produrre, e anche vendere, Arduino senza pagare royalty a Banzi. Questo perché la scheda Arduino è un pezzo di hardware open source, messo gratuitamente a disposizione di chiunque lo voglia utilizzare, modificare o vendere senza che sia coperta da brevetto. Banzi e il suo gruppo in questa nuova frontiera del mondo open source non sono da soli. Con un’iniziativa quasi priva di coordinamento, decine di inventori di hardware di tutto il mondo hanno cominciato a pubblicare le loro specifiche tecniche. Si trovano sintetizzatori, lettori mp3, amplificatori per chitarra e addirittura router per telefoni voice-over-ip, tutti open source. Banzi ammette che il concetto sembra una pazzia: “Se pubblichi tutti i tuoi file, in un

certo senso è come invitare la concorrenza a venire a ucciderti”. Anche Linux sembrava un’idea folle. Nessuno credeva che un manipolo di volontari part-time avrebbe potuto creare qualcosa di complesso come un sistema operativo, o che sarebbe stato più stabile di Windows. Eppure, 17 anni dopo, il movimento per il software open source è stato cruciale per l’esplosione dell’economia in rete.

Per quanto riguarda l’aspetto legislativo, poiché la legge sul copyright che regola il software open source non è applicabile all’hardware, Banzi ha deciso di utilizzare una *licenza Creative Commons* chiamata *Attribution Share Alike* che permette a chiunque di produrre, riprogettare e vendere copie della scheda. Tuttavia, se il progetto di riferimento viene ripubblicato, occorre dare il riconoscimento al gruppo Arduino originale e se la scheda viene modificata o cambiata, il progetto deve utilizzare la stessa licenza Creative Commons o una simile, per fare in modo che le nuove versioni della scheda Arduino siano altrettanto libere e aperte (in pratica gli stessi concetti espressi per i software open source). L’unico elemento di proprietà intellettuale che il gruppo si è riservato è il nome, che è diventato il suo marchio di fabbrica. Se qualcuno vuole vendere delle schede utilizzando questo nome, deve pagare una modesta commissione ad Arduino, in modo che il marchio non venga danneggiato da copie di scarsa qualità.

L’idea di base per cui questo hardware è stato sviluppato come “open source” è che così facendo potesse ispirare maggiore interesse e ricevere più pubblicità gratuita di quanto ne avrebbe potuta ricevere altrimenti. Questa scelta è stata premiata dalle comunità open source, che hanno permesso di sviluppare e perfezionare il linguaggio di programmazione. Inoltre, un distributore si è offerto di mettere in commercio le schede: Arduino ha venduto nel 2006 cinque mila unità e trenta mila nel 2007 e ad oggi si parla di margini di crescita molto alti.

L’hardware “open source” non è proprio simile al software open source. Riprodurre software non costa quasi nulla, mentre il gruppo Arduino deve sostenere i costi delle materie prime per produrre le schede prima di poterle vendere. In base alla tradizionale logica economica, nessuno rischia del denaro per permettere a tutti di riprodurre il prodotto su cui investe; ma allora com’è possibile guadagnare nel mondo dell’hardware “open source”?

Esistono principalmente due modelli economici: il primo è quello di non preoccuparsi di vendere molto, ma piuttosto di cedere la propria esperienza di inventore. Se chiunque può fabbricare un apparecchio allora il produttore più efficiente lo farà al miglior prezzo. Questo è esattamente il modo in cui lavora il gruppo Arduino; guadagna poco, ma i redditi importanti provengono dai clienti che vogliono costruire dispositivi basati sul loro processore e che assumono gli *esperti Arduino come consulenti*.

Il secondo modello è vendere quel che si è creato, ma cercando di *mantenersi all'avanguardia rispetto ai concorrenti*. La concorrenza può essere qualitativamente più scarsa ed avere minor mercato rispetto a quelli che sono gli inventori di quel prodotto. Infatti, Banzi ha notato che venivano messe in commercio tramite Internet imitazioni della sua scheda realizzate in Cina e Taiwan. Eppure i distributori di Arduino continuavano ad aumentare lo smercio delle schede; questo perché molte copie asiatiche erano piene di difetti di saldatura e i collegamenti dei “pin” erano deboli. Quindi, il gruppo Arduino ha un vantaggio competitivo nell’essere il creatore di questo hardware e ciò fa sì che le copie possono essere un elemento positivo per la stessa società.

Il caso Arduino è solo un esempio di come lo sviluppo del mondo open source possa espandersi in nuovi settori; l’applicazione di questo concetto si potrebbe avere in vari ambiti e per varie categorie di imprese, infatti, oltre all’ultimo caso Arduino, abbiamo menzionato nel resto dell’elaborato solo grandi imprese. Inoltre, si noti che l’ultimo caso è l’unico esempio di impresa italiana che ha adottato una “strategia open source”. In Italia soprattutto le piccole e medie imprese (PMI) dovrebbero guardare l’open source con maggiore interesse, perché potrebbero ricevere maggiore potenzialità e potere concorrenziale nell’adottare delle tecnologie sviluppate insieme alle comunità open source di quanto non avrebbero potuto produrre internamente.

Concludendo, lo sviluppo del mondo open source ha determinato degli effetti positivi per il mercato dei software in termini di maggiore efficienza, innovazione, concorrenza, varietà e qualità dei prodotti offerti; quindi non

possiamo che augurarci che lo stesso avvenga in altri settori, come quello dell'hardware.

Bibliografia

Anderson, C. (2009). *Gratis*. Milano: Rizzoli.

Balestreri, A. (2008). *Dispense cattedra E-business e l'azienda*. Università di Bologna.

Barbarito, L. (2008). *Economia dell'open source*. Torino: Pearson Paravia Bruno Mondadori.

Barney, J. B. (1986). *Strategic factor markets: Expectations, luck and business strategy*. *Management Science (MSc)*, 31, 1231 - 1241.

Boccardelli, P. (2008). *Strategie e modelli di business nell'industria del video entertainment: capitale umano, capitale relazionale e performance nel comparto cinematografico*. Il Mulino. Ricerca. Il Mulino.

Boccardelli, P., Fontana, F., & Manzocchi, S. (2007). *La diffusione dell'ICT nelle piccole e medie imprese*. Roma: Luiss University Press.

Bonaccorsi, A., & Rossi, C. (2003). *Comparing motivations of individual programmers and firms to take part in the open source movement*. From community to business. Working paper. Available at:
<http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf>

Boschetti C., Lipparini A. (1998). *Il vantaggio competitivo nella prospettiva resourcebased tra controllo e combinazione di risorse*. Carocci, Roma, 1998

Daft, R. L. (2007). *Organizzazione Aziendale*. Milano: Apogeo.

Dahlander, L., & Magnusson M.G. (2005). *Relationships between open source software companies and communities: Observations from Nordic firms*. Research Policy 34, 481–493.

Fabbricini, D. (2008). *Il Web 2.0 il vero motore del turismo online*. DailyNet

Fario, M., & Storti, V. (2000). *Commercio elettronico*. Napoli: Cuen.

Feller, J., Fitzgerald, B., (2002). *Understanding Open Source Software Development*. Addison Wesley, Boston

Fontana, F., & Caroli, M. G. (2006). *Economia e gestione delle imprese*. Milano: The McGraw-Hill Companies.

Lerner, J., & Tirole, J. (2002). *Some simple economics of open source*. Journal of Industrial Economics 52, 197–234.

Lerner, J., & Tirole, J., forthcoming (2005). *The scope of open source licensing*. Journal of Law, Economics and Organization, 21.

Muffatto, M., & Faldani, M. (2004). *Open Source*. Bologna: il Mulino.

O'Mahony, S., (2002). *The emergence of a new commercial actor: community managed software projects*. Unpublished Ph.D. dissertation, Stanford University.

O'Mahony, S., (2003). *Guarding the commons: how community managed software projects protect their work*. Research Policy 32, 1179–1198.

O'Mahony, S., Ferraro, F., forthcoming (2004). *Managing the boundary of an 'Open' project*, in: Padget, J., Powell, W. (Eds.), Market Emergence and Transformation.

Raymond, E., (1999a). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, Sebastopol, CA.

Raymond, E., (1999b). *The magic cauldron*. Available at: <http://www.catb.org/~esr/writings/magic-cauldron/>.

Scorza, G. (2002). *Associazione italiana per l'informatica ed il calcolo automatico. ICT globalizzazione e localismi*. Bitonto (BA) : AICA, 2002. p. 701-720 ()1365453

Teti, A., & Cipriano, E. (2006). *EUCIP - Il Manuale per l'informatico professionista*. Milano: Hoepli.

West, J., & O'Mahony, S.,(2004). *Contrasting community building in sponsored and community founded open source projects*. Working Paper. Available at: <http://opensource.mit.edu/papers/westmahony.pdf>.

Sitografia

<http://www.ebitinnovation.com>

<http://www.fsf.org>

<http://www.gnu.org>

<http://www.juniperresearch.com>

<http://www.Kioskea.net>

<http://www.opensource.org>

<http://www.puntocontatto.it>

<http://www.techup.it>

<http://www.wikipedia.org>

<http://www.windoweb.it>