

Apertura verso il network e presidio della posizione

Le strategie per creare imprese di rete e il caso dell'open source software

Francesco Rullani

Copenhagen Business School¹



¹ Francesco Rullani è Assistant Professor in *Management and Economics of Innovation* presso il Dipartimento di Innovation and Organizational Economics della Copenhagen Business School, Kilevej 14A, office 3.81, 2000 Frederiksberg, Denmark; tel: +45 3815 2992; fax: +45 3815 2540, email: fr.ino@cbs.dk; web: <http://www.francescorullani.com>

Introduzione

In questo breve elaborato espongo un approccio alla costruzione di modelli di business aperti da utilizzarsi in contesti di rete, cioè in contesti dove le funzioni produttive e innovative sono distribuite tra diversi attori specializzati, connessi da un network di relazioni.

Lo scopo è duplice.

Da un lato, cerco di delineare una teoria in grado di spiegare come fare “impresa di rete” in un network con altri soggetti, bilanciando *apertura* (cioè il fatto che per rendere efficace il lavoro in rete l’impresa deve rendere trasparenti i suoi processi produttivi e innovativi agli altri attori del network) e *presidio* (cioè il controllo esclusivo della posizione che l’azienda viene ad avere nel network).

Dall’altro, cerco di connettere le linee guida di questa teoria all’osservazione dei business model applicati dagli imprenditori che ogni giorno si confrontano sul mercato *dell’open source software*, un mercato in cui l’apertura del prodotto è estrema. Per realizzare quest’ultimo passaggio mi rifaccio a diversi studi empirici sull’open source software (ad esempio, Picerno e De Rossi, 2009; Corsino, Giuri e Torrisi, 2005) e a due investigazioni principali: un’analisi su larga scala condotta da Daffara (2009) e quella di due casi studio condotta da Dahlander (2005).

Il fine ultimo è quindi sistematizzare i caratteri concreti e attuali del fare impresa in una rete aperta in una teoria che possa essere utile agli imprenditori che hanno intenzione di avventurarsi nella rete.

Indice

1. Strategie per competere nella rete.....	2
1.1. La rete	2
1.2. Apertura e presidio.....	3
1.3. Realizzare l’apertura senza farsi male: le ragioni del presidio	3
2. Un esempio estremo di business model aperto e presidabile: l’open source software	5
2.1. Caratteri dell’open source software	5
2.2. Dimensione del fenomeno: la domanda.....	5
2.3. Dimensione del fenomeno: l’offerta	7
2.4. Presidiare anche laddove l’apertura è estrema: un esempio	8
3. Analisi empirica su larga scala degli open-source based business model.....	10
3.1. Product Specialist.....	10
3.2. Open Core	11
3.3. Indirect Revenues.....	11
3.4. Dual Licensing	11
3.5. R&D Cost Sharing	12
3.6. Aggregate Support Provider.....	12
3.7. Legal Certification and Consulting	13
3.8. Training and Documentation	13
3.9. Platform Provider.....	13
3.10. Selection/Consulting Services.....	13
4. Due emblematici casi studio di open-source based business model	16
4.1. MySQL.....	16
4.2. Cendio	17
5. Conclusione.....	19
6. Bibliografia	20

1. Strategie per competere nella rete

1.1. La rete

Uno dei concetti chiave sviluppati nella letteratura economica e manageriale negli anni recenti è quello di *rete*, un concetto tanto più importante oggi di fronte ad una crisi dalle dimensioni globali, che rende impotenti anche attori di grandi dimensioni e sottolinea la necessità di costruire percorsi produttivi che sappiano sfruttare economie di collaborazione.

L'idea di fondo è che, invece di distribuire competenze e funzioni produttive ai centri di attività all'interno dell'azienda, l'impresa assegni a fornitori esterni dell'azienda le attività designate come *non-core*, usando le competenze e le capacità produttive di altri soggetti: altre imprese specializzate nella catena del valore, centri di ricerca e fornitori di servizi specializzati. Ma anche organizzazioni meno "ortodosse", come le comunità di consumatori o di professionisti (Micelli, 2000).

L'organizzazione "a rete", nelle sue diverse varianti, consente alle singole imprese della rete di avere vantaggi importanti in termini di flessibilità e di risposte creative a situazioni e opportunità non previste. Infatti, quando si lavora in rete, si ha la possibilità di organizzare:

- a) una efficace *divisione della lavoro manifatturiero*, determinando incrementi di produttività nelle singole aziende specializzate in specifiche fasi della produzione;
- b) una efficace *divisione del lavoro cognitivo*, per cui la conoscenza viene scambiata e condivisa entro le maglie della rete, consentendo importanti economie nella creazione di innovazione e nell'allargamento del bacino delle applicazioni;
- c) una efficace *divisione dei costi di investimento e dei rischi*, che vengono distribuiti su più soggetti, rendendoli compatibili con le disponibilità finanziarie delle singole imprese, specialmente se di piccole dimensioni.

Lo sviluppo delle tecnologie ICT ha ulteriormente favorito la collaborazione attiva nel settore dell'*Internet economy*, permettendo la nascita di processi avanzati di collaborazione in rete come Wikipedia, o *social networks* come Facebook o Lazio-Side (Benkler, 2006; Tapscott e Williams 2006).

Oggi, l'idea che la conoscenza sia distribuita tra diversi attori e debba essere scambiata, messa a sistema, per incrementare il valore prodotto, l'idea che il prodotto possa "cambiare di mano" molte volte prima di arrivare al consumatore, e l'idea che le fasi di preparazione ed erogazione di un servizio siano gestite da una molteplicità di agenti, sono dati assodati. E questo non solo per le piccole e medie imprese italiane, che con i distretti industriali (si veda ad esempio, Becattini, 2001) hanno aperto la strada a questo dibattito in Italia e nel mondo, ma anche per molte grandi aziende statunitensi che si sono già mosse in questa direzione (Berger, 2005).

1.2. Apertura e presidio

La partecipazione in rete si compone di due fondamentali caratteristiche.

Da un lato è evidente che è possibile agganciarsi alla rete solamente *aprendo* i propri processi produttivi, e soprattutto la propria capacità innovativa, verso l'esterno (Chesbrough, 2003; Chesbrough, et al., 2006), lasciando le proprie idee libere di trovare l'applicazione in grado di farle fruttare al meglio. Allo stesso tempo, la connessione con la rete dà accesso ad un ambiente esterno ricco di opportunità e di idee -generate dalla capacità innovativa dei consumatori, dei fornitori o addirittura dei concorrenti- che possono essere fatte fruttare all'interno dell'impresa, o che viceversa permettono di bypassare quelle parti del processo produttivo che non si reputano interessanti.

Dall'altro lato, è ovvio che questo tipo di apertura è destinata a diventare insostenibile per l'impresa se ad essa non si affianca la seconda caratteristica fondamentale della partecipazione alla rete: il *presidio* della propria *network position*. Far parte di una rete di soggetti che producono valore è fondamentale per accrescere il valore prodotto dal network nel suo complesso, ma diventa altrettanto imperativo costruire meccanismi che permettano alla singola azienda di aumentare il suo grado di insostituibilità, per i clienti e per la filiera di appartenenza. Il grado di insostituibilità di ogni specialista della rete costituisce infatti l'elemento fondamentale per definire quale sarà la sua capacità di appropriarsi, agendo sui prezzi interni alla filiera, di una parte del valore prodotto.

Apertura e presidio della network position sono quindi i fattori chiave che permettono all'azienda di partecipare profittevolmente ad una rete di diversi soggetti, sapendo che se da un lato mantenere il bilanciamento tra apertura e appropriabilità, e quindi assicurarsi una congrua fetta della torta, sarà una continua sfida, dall'altro la partecipazione nella rete permetterà la moltiplicazione del valore prodotto dall'intero network, accrescendo le dimensioni della torta da dividere.

1.3. Realizzare l'apertura senza farsi male: le ragioni del presidio

Il bilanciamento delle strategie di apertura e di appropriabilità non è semplice. L'idea chiave per capire come realizzare un modello di *network innovation* del tipo descritto precedentemente è quella di creare una *competenza distintiva* presidiata dall'impresa e difficilmente imitabile dai concorrenti. Tuttavia non è semplice immaginare quali nodi della rete possano essere efficacemente presidiati in un network in cui, per tutti gli altri aspetti, la conoscenza dei partecipanti viene condivisa, a monte, a valle e anche orizzontalmente.

Per mettere a fuoco il problema partiamo dalla difficoltà prima che deve affrontare un'impresa appartenente ad una rete in cui viene condivisa la conoscenza. Quando la conoscenza è astratta e

decontestualizzata, trasformabile in un testo, in un algoritmo, in una formula, in un software, è solitamente anche copiabile e distribuibile a costo zero. Senza un supporto esterno, o senza protezioni di tipo legislativo (i diritti di proprietà intellettuale come il copyright o i brevetti) sarebbe impossibile per chi produce quella conoscenza estrarre valore dalla sua commercializzazione (Arrow, 1962). L'impatto e la diffusione della pirateria di prodotti digitali testimoniano l'importanza di questa proprietà della conoscenza codificata, proprietà che, rendendone impossibile la commercializzazione *sic et simpliciter*, consuma il terreno su cui potrebbe poggiare un eventuale modello di business incentrato su di essa.

Dando un secondo sguardo al problema, tuttavia, ci si accorge che quanto detto non tiene conto di un fatto fondamentale: che per produrre valore la conoscenza codificata deve poi essere applicata, specificata e contestualizzata. Di conseguenza, non solo le imprese che la producono, ma anche quelle che la gestiscono e fanno utilizzarla diventano attori fondamentali per l'intero network: senza di esse quella conoscenza non troverebbe applicazione. In altre parole, la grande quantità di informazione, comunicazione e conoscenza codificata che viene prodotta nel network deve essere continuamente ritrasformata per essere adattata alle esigenze delle singole applicazioni, dei singoli luoghi, dei singoli clienti. Di conseguenza le imprese che sono in grado di presidiare gli snodi in cui la conoscenza del network si cala "nella realtà", nel contesto del consumo, diventano essenziali per il network stesso e quindi in grado di appropriarsi di una congrua parte del valore prodotto.

Questo esempio mostra uno dei meccanismi chiave attorno ai quali è possibile costruire un modello di business capace di sfruttare il bilanciamento tra apertura (che accresce la torta) e presidio (che permette all'impresa di catturarne una congrua fetta).

Nelle sezioni che seguono cercherò di dare sostanza a questa impostazione teorica rifacendomi al caso dell'open source software, un settore in crescita proprio grazie a meccanismi simili a quelli descritti.

2. Un esempio estremo di business model aperto e presidiabile: l'open source software

2.1. Caratteri dell'open source software

Consideriamo il caso di un settore in grande espansione, nel mondo così come in Italia, in cui apertura e libera circolazione della conoscenza sono portate all'estremo: *l'open source software*. Con il termine open source software si intende un particolare software, che può essere un sistema operativo o anche un semplice applicativo, che viene distribuito dal suo produttore tramite una licenza approvata dalla *Open Source Initiative* (<http://www.opensource.org/>), una *public benefit corporation* californiana creata nel 1998 per sostenere e promuovere questo tipo di software. I criteri per rientrare nel novero delle licenze open source si basano sull'idea di *copyleft*, un termine coniato per indicare l'utilizzo del copyright non al fine di restringere i diritti degli utilizzatori ma al contrario di accrescerli. Quando si entra in possesso di software open source, tipicamente scaricandolo da internet, non solo si può avvalersi della versione "binaria" del codice, quella che contiene i comandi per la macchina, ma se ne riceve anche una copia in un linguaggio di più alto livello, comprensibile a chiunque possieda le necessarie conoscenze. In questo modo, chi ne entra in possesso è in grado di leggere il codice sorgente, individuarne possibili bachi, aggiustarli, o addirittura costruire nuovo codice volto a modificare o migliorare il programma. Inoltre, il ricevente – in accordo con la licenza - può ridistribuire il software originale ed anche quello modificato, con un'unica restrizione: che anche il programma derivato sia distribuito utilizzando la stessa licenza². La diretta conseguenza dell'uso di queste licenze è che il software diviene "non appropriabile", nel senso che chiunque decida di sviluppare un programma a partire da un codice open source iniziale non potrà poi apporre alcun copyright classico sul nuovo codice, da lui ottenuto in derivazione dal primo. E questo anche nel caso in cui i miglioramenti e gli ampliamenti prodotti *in-house* siano la maggioranza. Di fatto, il software open source e la conoscenza prodotta dai suoi sviluppatori sono totalmente aperti e, solitamente, facilmente reperibili a prezzo nullo o simbolico.

2.2. Dimensione del fenomeno: la domanda

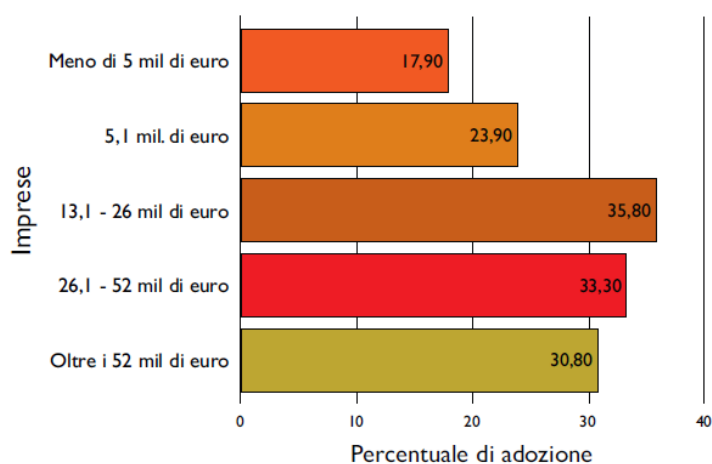
Alcune recenti ricerche hanno tentato di dare una dimensione al fenomeno open source software sia a livello mondiale che a livello italiano. Sebbene la fetta di mercato occupata da sistemi operativi open source per desktop computer corrisponda a meno dell'1% del mercato mondiale, in altri mercati il software open source è in grado di raggiungere la quota maggioritaria. Ad esempio il 73%

² Il grado in cui questa clausola deve essere rispettata varia da licenza a licenza. Si veda Lerner e Tirole (2005) per una panoramica delle diverse licenze.

dei web server mondiali si basa su Apache, che è software open source. Tra le applicazioni, Firefox, un browser open source, è riuscito a conquistare in relativamente pochi anni un quinto del mercato, così come ha fatto MySQL assicurandosi circa il 10% del mercato dei database.

Anche in Italia l'open source software ha raggiunto una dimensione ragguardevole. Tra i server che gestiscono domini italiani (cioè con suffisso “.it”) il 59% utilizza Apache, mentre i prodotti Microsoft occupano il 28% del mercato³. Concentrandosi sulle imprese, si può notare che alcune stime accertano che il 12,2% delle imprese italiane, e in particolare il 39,7% delle aziende con più di 250 dipendenti, adotta sistemi operativi liberi o open source. Altre fonti conducono a numeri comparabili (si veda la Fig. 1)⁴.

Fig. 1 - Percentuale di adozione di OSS per fatturato



Fonte: Picerno e De Rossi, 2009, p. 7

Le motivazioni alla base della domanda di open source software sono molteplici. Da un lato le imprese possono fare affidamento sul fatto che il prezzo da pagare per la licenza degli applicativi e dei sistemi open source sarà mantenuto relativamente molto basso grazie al fatto che il prodotto stesso è per definizione copiabile e redistribuibile (Picerno e De Rossi, 2009). Ovviamente i costi relativi all'adozione di un certo software non si limitano al prezzo della licenza, ma comprendono anche il suo adattamento, la sua manutenzione, il training per gli utilizzatori, e così via⁵. Ma anche considerando tutto questo, la contrazione dei costi è certamente una delle motivazioni principali alla base dell'adozione di open source software. La possibilità di customizzare i programmi, anch'essa

³ Dati relativi al 2007 e al 2008 ricavati da Barbarito (2008).

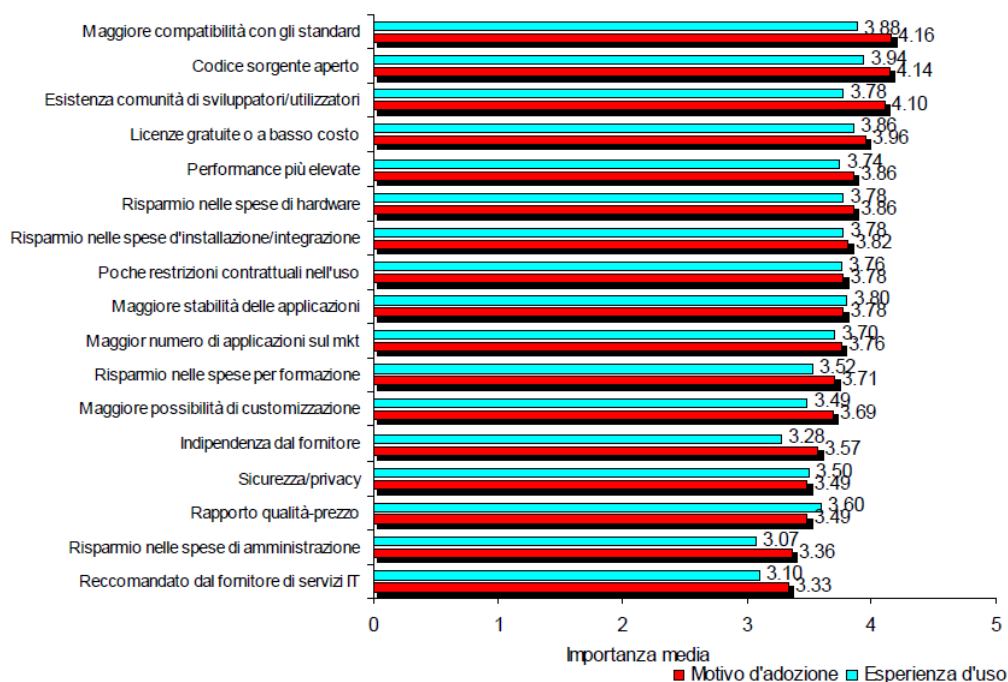
⁴ Dati relativi al 2008 ricavati da Picerno e De Rossi (2009). La Fig 2 riporta dati relativi al rapporto dell'Osservatorio TeDIS del 2004 ma, come sottolineano gli autori "Il dato è stato riconfermato, con variazioni minime, nel corso delle indagini degli anni successivi (il rapporto segnalava risultati invariati dal 2004 al 2007, sulle medesime domande)." P. 7

⁵ E' il cosiddetto *total cost of ownership*, si veda Corsino, Giuri e Torrisi, 2005; per un approfondimento del caso italiano

direttamente collegabile alla caratteristica apertura di questa tipologia di applicazioni e sistemi, è un altro importante driver dell'adozione.

Come mostra la Fig. 2 elaborata da Corsino, Giuri e Torrisi (2005) a partire dall'analisi di 93 imprese italiane, accanto a questi due principali driver vi sono anche la riconosciuta qualità del software open source, ed un ulteriore fattore, anch'esso derivante "dell'apertura": la sua maggiore compatibilità con gli standard.

Fig. 2 - Criteri rilevanti nell'adozione e nell'esperienza d'uso dei sistemi operativi OSS



Fonte: Corsino, M., Giuri, P. e Torrisi, S. (2005), p. 73.

Valori medi su una scala da 1 a 5, campione di 93 imprese italiane.

2.3. Dimensione del fenomeno: l'offerta

Come si può facilmente dedurre dalla discussione precedente, il beneficio che le imprese possono avere dal distribuire apertamente il proprio software o dallo sponsorizzare o addirittura fondare progetti open source risiede principalmente nell'*effetto moltiplicatore* generato dalla sua intrinseca e totale apertura. Questo effetto moltiplicatore si estende lungo due dimensioni.

Innanzitutto, il software distribuito come open source è solitamente reperibile facilmente (ad esempio, su internet) gratuitamente o ad un prezzo molto conveniente, diventando così accessibile ad un'ampissima base di consumatori. La facilità di accesso ne promuove la diffusione, moltiplicando quindi l'impatto di qualsiasi attività ad esso associata.

In secondo luogo, la gran parte del software open source è sviluppato da una comunità di sviluppatori che contribuiscono alla creazione e manutenzione dei programmi per le motivazioni più

varie, molte delle quali non hanno nulla a che vedere con il profitto. Si va dal bisogno di una particolare *feature* di un prodotto che non è reperibile sul mercato (von Hippel, 2002), alla volontà di imparare nuove *skill* tramite l'interazione su problemi concreti (e quindi su prodotti software) con altri sviluppatori (von Hippel e von Krogh, 2003), alla spinta politica e ideologica contraria alla "chiusura" del codice (Stewart e Gosain, 2006), alla volontà di essere notati per le proprie capacità (Dalle et al., 2004). Ovviamente in quest'arena si trovano anche molte imprese e dipendenti di imprese pagati per sviluppare software open source (Roberts et al., 2006) o che cercano di sfruttare la propria eventuale notorietà come sviluppatori per trovare impieghi migliori (Lerner e Tirole, 2002). Attorno ad uno stesso progetto possono quindi lavorare soggetti di diversa natura i cui scopi e interessi si incrociano e co-evolvono. Il risultato è che un'impresa che partecipi o diriga un progetto open source avrà a disposizione come fattore moltiplicativo la comunità dei tanti soggetti che riuscirà a raccogliere attorno al proprio progetto. Se la comunità sarà motivata e partecipe, oltre che numericamente importante, l'impresa potrà contare sul fatto che molti sviluppatori e bug tester apporteranno modifiche importanti al codice, migliorandolo ed estendendolo, o addirittura sviluppandone la maggior parte.

Queste caratteristiche rendono particolarmente attraente per le piccole imprese l'utilizzo di un modello di business aperto basato sull'open source (Benussi e Rossi, 2007). I limiti posti dalle dimensioni ridotte dell'impresa, infatti, possono essere superati grazie a questo effetto moltiplicatore che supporta e amplifica non solo la distribuzione del prodotto, ma anche lo sviluppo del software stesso. Ciò che diventa fondamentale in questo caso *non* è quindi la dimensione dell'impresa, ma la sua capacità di avere un ruolo attivo e non sostituibile, presidiando uno dei nodi attraverso cui l'interno network di imprese, pubbliche amministrazioni, comunità, sviluppatori dipendenti e semplici volontari sviluppa e mantiene software di qualità.

2.4. Presidiare anche laddove l'apertura è estrema: un esempio

Dato il carattere di apertura estrema del modello, la ricerca manageriale ha cercato di capire come sia possibile coniugare questa caratteristica con modelli di business che assicurino all'impresa una duratura profittabilità. Uno dei principali modelli di business creati dagli imprenditori nel settore ha preso vita a partire dalla duplice considerazione che la conoscenza codificata, espressa in forma di codice sorgente e documentazioni relative, viene prodotta da una comunità di sviluppatori e non può essere appropriata, ma anche che le imprese possono porsi a presidio degli snodi in cui il software deve calarsi nel contesto di una sua applicazione.

Analizziamo, per esempio, le opportunità di business che si sono aperte nel mercato dei programmi utilizzati per gestire la contabilità aziendale. In particolare, immaginiamo di avere come obiettivo la

valutazione delle opportunità di business in questo mercato per un'applicazione basata su software open source. Il software, in questo caso, non è appropriabile, ma proprio il suo prezzo basso o nullo rende conveniente investire risorse e fornire servizi che *rendano possibile applicarlo in contesti di uso specifici*. Vediamo come.

La gratuità del codice di base crea una domanda latente, legata alla possibilità di risparmiare sul costo delle licenze dei software proprietari precedenti, che – in molti casi – hanno pesato in modo rilevante sui bilanci delle aziende. Il software open source, che può essere acquisito gratuitamente, può infatti rappresentare una valida strategia di riduzione dei costi sostenuti. Ma se il software usato è sviluppato da una comunità che agisce sull'onda di motivazioni che, come abbiamo visto, sono spesso non legate al profitto e al mercato, è molto probabile che il software gratuito non sia direttamente utilizzabile per risolvere i problemi specifici che interessano la singola azienda. Inoltre questo software sarà, in genere, distribuito senza garanzie e non vi sarà alcun soggetto in grado di prendersi la responsabilità di assicurare, nel corso del tempo, la manutenzione, *l'improvement* e la gestione del programma.

Vi è dunque un chiaro *gap* tra ciò che le imprese vorrebbero usare, cioè un software a basso prezzo che gestisca la loro contabilità aziendale, e le specifiche caratteristiche contestuali che quel software deve avere per funzionare in quell'ambiente. E' nelle attività richieste per colmare questo gap che è possibile costruire un business. L'impresa che si farà carico dell'adattamento del software gestionale alla struttura organizzativa e finanziaria di ogni singola azienda che desidera utilizzarlo acquisterà (e presidierà) una posizione che le darà il diritto di ottenere un adeguato compenso per il proprio operato anche se il software è stato prodotto da una comunità più ampia ed è -e resterà sempre- aperto e distribuibile. Inoltre, la realizzazione dell'adattamento (anche attraverso la scrittura di nuovo software che andrà quindi ad accrescere il pool di codice condiviso e liberamente accessibile a tutti) permetterà all'impresa di acquisire una posizione di vantaggio competitivo rispetto alle altre aziende che offrono servizi simili, dato che conoscerà meglio di ogni altro le specifiche del software adattato. Di più, ogni impresa che sarà riconosciuta come quella che ha per la maggior parte contribuito alla costruzione del un software stesso –che, ripetiamo, è e sarà sempre aperto- avrà il vantaggio di essere percepita come quella che ha le più approfondite conoscenze in merito, e quindi diverrà la più richiesta sul mercato per consulenze, *training* e servizi collegati. L'impresa può costruire i propri profitti su questi servizi, persino spendendo risorse nella scrittura di codice che poi dovrà inevitabilmente lasciar confluire nel programma distribuito gratis su internet, semplicemente perché ciò che l'impresa stessa presidia non è più il prodotto, ma la conoscenza specifica e contestuale che ha acquisito nello studiarlo, accrescerlo o persino realizzarlo.

3. Analisi empirica su larga scala degli open-source based business model

Quanto descritto in precedenza mostra come in un ambiente totalmente aperto sia possibile immaginare un modello di business profittevole semplicemente spostando il “presidio” dell’azienda dal prodotto alla conoscenza del prodotto e dei suoi usi. Ma ovviamente quello che abbiamo presentato sopra è un caso estremo, sebbene sia, a mio modo di vedere, il modello di business basato sull’open source che nella realtà si dimostra più avanzato.

In effetti gli anni recenti hanno visto l’esplosione di una miriade di modelli di business che realizzano in maniera molto eterogenea, e a volte contraddittoria e con diversi gradi di successo, il bilanciamento tra apertura e presidio. In particolare, si può osservare come ciò che muta da un business model all’altro sia soprattutto il *locus* del presidio delle imprese, mentre altri caratteri, come la tipologia di apertura, dettata dalla licenza utilizzata, risultano più omogenei tra i diversi modelli.

La letteratura manageriale ha recentemente cominciato a studiare le strategie che diverse imprese hanno realizzato per costruire business nel mondo open source (per esempio, Picerno e De Rossi, 2009; Bonaccorsi, Giannangeli e Rossi, 2006). Il contributo su cui vorrei soffermarmi in questa sezione dà una visione d’insieme dei modelli di business ricavando dall’osservazione di 275 imprese 10 archetipi cui la maggior parte delle strategie imprenditoriali in atto può essere ricondotta (Daffara, 2009).

3.1. *Product Specialist*

La prima e più diffusa tipologia di business model basato sul software open source si incentra sull’idea di spostare il presidio dell’impresa dal prodotto, il software sviluppato dall’impresa ed esposto al più alto grado di apertura tramite la licenza open source, alla conoscenza del prodotto e dei suoi usi. Tale conoscenza può essere facilmente presidiata dall’impresa fondatrice o comunque promotrice del progetto open source grazie alla maggiore *expertise* sviluppata nella costruzione stessa del codice e nel suo mantenimento. Allo stesso tempo, tale conoscenza può generare valore perché il software nella maggior parte dei casi non potrà essere utilizzato direttamente ma dovrà essere adattato alle specifiche caratteristiche contestuali del suo utilizzo. Di conseguenza l’impresa potrà fornire servizi di adattamento, training, mantenimento e così via. E’ facile notare come questa descrizione richiami proprio l’esempio offerto nelle sezioni precedenti.

Un esempio di impresa che utilizza questo business model è *Alfresco* (<http://www.alfresco.com>)

3.2. Open Core

Le imprese che utilizzano questa tipologia di business model sviluppano un prodotto open source che viene diviso in due versioni. La prima versione è rilasciata con licenza open source e contiene le *basic feature* del programma. La seconda versione è costruita sulla prima includendo nella parte open source alcuni *improvement* che sono tuttavia proprietari e quindi vengono distribuiti con una licenza che richiede il pagamento di un corrispettivo al licenziatario. In questo caso l'apertura ha il vantaggio di diffondere il software di base e contribuire non solo ad aumentare il numero di potenziali utilizzatori della versione *improved*, ma anche di far sì che la comunità di sviluppatori che dovesse venirsi a creare attorno al progetto possa contribuire a migliorare il prodotto. Ciò che l'impresa presidia in questo caso è dunque esclusivamente la parte più avanzata del prodotto e ovviamente la conoscenza sottostante le *advanced feature*.

In questo caso Daffara (2009) propone di guardare a Zimbra (<http://www.zimbra.com>) per avere un esempio di impresa che utilizza prevalentemente questo modello.

3.3. Indirect Revenues

Anche questo business model sfrutta l'apertura per diffondere il software il più possibile, ed ottenere così i vantaggi visti precedentemente, ma in questo caso il prodotto collegato al software open source non è una sua versione arricchita con moduli proprietari ma più genericamente un prodotto complementare al software stesso. Il tipico esempio di questo tipo di bene complementare è una *device hardware*, come una stampante, il cui sistema operativo è un precedente programma open source adattato, oppure un nuovo software creato dall'impresa e poi rilasciato in forma proprietaria.

Il lettore può riferirsi al caso rappresentato da Cendio (<http://www.cendio.com/>), studiata in maniera approfondita da Dahlander (2005) e che vedremo nella sezione successiva.

3.4. Dual Licensing

In questo caso l'impresa produce un software originale che viene rilasciato in due modalità diverse. Con la prima licenza, di tipo open source, il programma è scaricabile, modificabile e redistribuibile da chiunque. Poiché questa licenza non permette a chi entra in possesso del software di rivendere il programma derivato in forma proprietaria, l'impresa che produce il software rilascia -a pagamento- anche una versione commerciale dello stesso software. I clienti che decidono di comprare questa

licenza sono quindi in grado di utilizzare il software dell'impresa come fosse un prodotto proprietario, ad esempio di includendolo nei loro pacchetti o programmi derivati e distribuendo il tutto con licenze commerciali. Il cliente è disposto a pagare il differenziale di prezzo tra la licenza open source e quella commerciale perché in questo modo ottiene il diritto di “chiudere” il software acquistato all'interno di un prodotto distribuito con una licenza non più open source. In questo modo, l'impresa realizza un profitto presidiando l'utilizzo commerciale a valle del proprio software. L'impresa che meglio rappresenta questo tipo di strategia è *MySQL* (<http://www.mysql.com/>), che sarà oggetto di più ravvicinata analisi nella prossima sezione.

3.5. R&D Cost Sharing

Le imprese che adottano questo business model sono imprese che offrono servizi di consulenza e adattamento simili a quelli descritti per il caso di *Product Specialist*, anche se nella presente fattispecie non presidiano la conoscenza relativa ad un preciso prodotto, ma quella relativa ad un processo: quello di sviluppo di un certo tipo di programmi. Il caso tipico è quello in cui il cliente necessita di un software particolare e si rivolge all'impresa perché questa sviluppi un software adatto ai propri scopi e alle proprie necessità. In questo caso l'impresa presidia la conoscenza relativa allo sviluppo del software in questione, e utilizza software open source già esistente come punto di partenza allo scopo di tagliare i costi di sviluppo.

Redomino (<http://redomino.com/it> , si veda Benussi, 2007) può rappresentare un esempio di questo tipo di business model.

3.6. Aggregate Support Provider

Questo business model è un'estensione o una generalizzazione del precedente, in quanto si incentra sulla capacità dell'impresa di presidiare processi di sviluppo generici, adattabili a molti contesti, tramite una conoscenza di ampio respiro. L'idea di fondo è che i clienti possano rivolgersi all'impresa quando necessitano di un intervento specifico su di un problema che non necessariamente si lega ad un dato software.

Un'impresa che può essere presa in esame per approfondire questo business model è *OpenLogic* (<http://www.openlogic.com/>)

3.7. Legal Certification and Consulting

L'azienda che decida di adottare questo business model offre servizi di consulenza legale ai soggetti che sviluppano open source. Come nei casi precedenti, l'apertura del codice aiuta la diffusione del prodotto, mentre ciò che presidia l'impresa è un servizio/bene complementare che in questo caso si concretizza nella certificazione o nella fornitura di servizi legati alla dimensione legale dello sviluppo e dell'utilizzo del codice.

Daffara (2009) suggerisce di guardare a *Palamida* (<http://www.palamida.com/>) per avere un esempio di questo modello.

3.8. Training and Documentation

In maniera simile al modello precedente, in questo caso l'impresa presidia un *bundle* di servizi complementari alla produzione, utilizzo e gestione del software open source, servizi che in questo caso si basano non solo sulla conoscenza relativa al software open source, ma anche al trasferimento al cliente di questa conoscenza. Inoltre, le imprese impegnate in questo tipo di business possono presidiare anche la produzione di un bene complementare al software e allo stesso servizio offerto: la documentazione. Presidiare il trasferimento di conoscenza, sia esso basato su servizi o su prodotti, è dunque ciò che permette all'impresa di appropriare parte del valore prodotto dall'apertura del software.

Un esempio in merito è offerto da *Gbdirect* (<http://www.gbdirect.co.uk/>)

3.9. Platform Provider

In questo modello il presidio dell'impresa si sposta dal trasferimento di conoscenza al supporto della fase di sviluppo, integrazione e adattamento di diversi software. Ad esempio, fanno parte di questo gruppo imprese che combinano diversi programmi per realizzare prodotti il più possibile *plug and play* da distribuire su larga scala, presidiando quindi la fase di supporto all'implementazione successiva.

Le classiche *Linux distribution* come *RedHat* (<http://www.it.redhat.com/>) sono la migliore esemplificazione del modello.

3.10. Selection/Consulting Services

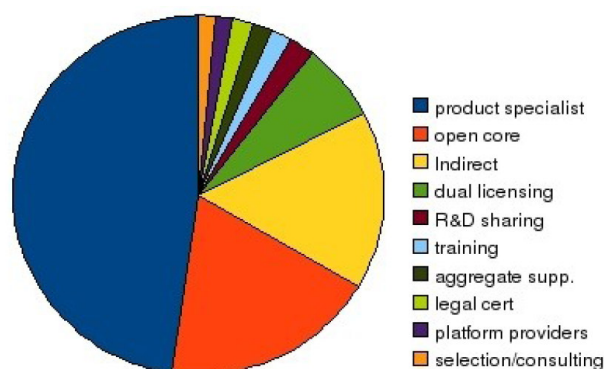
L'ultimo modello individuato si basa ancora sulla fornitura di servizi, ma di livello ancor più generico di quanto riscontrato per il caso di R&D Cost Sharing. In questo caso ciò che l'impresa

presidia è la conoscenza del settore che le permette di indirizzare il cliente verso il progetto o il software open source più adatto alle proprie esigenze.

Il caso richiamato da Daffara (2009) in questo caso è *Navica* (<http://www.navicasoft.com/>)

Per avere un'idea della diffusione dei modelli e della importanza relativa di ognuno di essi si consideri la Fig. 2, che dà una rappresentazione grafica del numero di imprese del campione che applicano *in prevalenza* uno specifico modello. Si veda anche la Tab. 1, che oltre a riportare i dati su cui è basata la Fig. 2 riporta anche un riassunto dei punti di forza e di debolezza di ogni business model così come rilevati da Daffara (2009).

Fig. 2 – Numero di imprese del campione di Daffara (2009) stratificate per business model



Fonte: Daffara (2009), p. 45.

E' utile sottolineare che nella realtà quasi tutte le imprese applicano delle versioni ibride di questi dieci archetipi, i quali sono stati definiti precedentemente nella loro versione più "pura" per fornire al lettore un'adeguata sistematizzazione delle principali caratteristiche di ogni modello.

Tab. 1 – Imprese del campione utilizzato da Daffara (2009) stratificate per business model

Business Model	Vantaggi	Svantaggi	Caso	Numero Imprese	Perc.
<i>Product Specialist</i>	Reduction of R&D, reduced maintenance costs, visibility, increased dissemination, external ecosystem of add-ons	Low barrier of entry for third-parties	<i>Alfresco</i>	131	47.6
<i>Open Core</i>	Reduction of R&D, reduced maintenance costs, visibility, increased dissemination, external ecosystem of add-ons, self-segmentation of the market for the proprietary add-ons	Difficult to estimate the right balance between open and closed parts, external groups may create substitutes for the proprietary parts	<i>Zimbra</i>	52	18.9
<i>Indirect Revenues</i>	Source availability reduces engineering costs and increase visibility on multiple platforms	Requires a large external market for incentives, may be dependent on a single (or small number) of actors increasing risk	<i>Cendio**</i>	44	16
<i>Dual Licensing</i>	Dissemination for the product with reduced costs, creation of external ecosystem of add-ons (outside the source), visibility, self-segmentation of the market	Low external participation (limited code contributions)	<i>MySQL</i>	19	6.9
<i>R&D Cost Sharing</i>	Reduction of R&D, reduced maintenance costs	Establishing the management and contribution structures may be complex and costly, requires constant effort	<i>Redomino*</i>	6	2.2
<i>Aggregate Support Provider</i>	Cost of support can be partially shared across customers, economies of scale	Limited market, may be perceived as in partial competition with existing specialists	<i>OpenLogic</i>	5	1.8
<i>Legal Certification and Consulting</i>	Cost of legal certification and secondary-level insurance can be shared across the most used OSS projects	Limited market, difficult to estimate risk probabilities, need to cover separate legal frameworks across the world with different rules	<i>Palamida</i>	5	1.8
<i>Training and Documentation</i>	A significant portion of training development costs can be shared across customers, economies of scale, reuse of community-developed material	May be perceived as in partial competition with existing specialists, human intensive, most of it cannot be replicated at low cost	<i>Gbdirect</i>	5	1.8
<i>Platform Provider</i>	Reduction of R&D, reduced maintenance costs, visibility, increased dissemination, external ecosystem of software and additions	Platform engineering requires large R&D efforts even with shared resources	<i>RedHat</i>	4	1.5
<i>Selection Consulting Services</i>	Cost of software certification and selection can be partially shared across customers, as most adopters have a large share of common needs	Limited market, difficulty in following rapid evolution of the products covered (evaluation costs)	<i>Navica</i>	4	1.5
Totale				275	100

Fonte: Elaborazione da Daffara (2009).

*Le informazioni relative a questo caso sono state ricavate da Benussi (2007).

**Le informazioni relative a questo caso sono state ricavate da Dahlander (2005).

4. Due emblematici casi studio di open-source based business model

In questa sezione vorrei dare corpo ai business model descritti nella precedente parte del testo riportando due casi aziendali analizzati da Dahlander (2005), che son a mio avviso emblematici: *MySQL* e *Cendio*. In questo modo vorrei da un lato far capire come i principi esposti nel testo sono stati messi in pratica e allo steso tempo dare ragione della eterogeneità dei modelli possibili, tipicamente lontani dagli archetipi puri riportati in tabella 1 e invece caratterizzati da un misto di strategie che possono rafforzarsi a vicenda oppure creare una tensione che l'impresa si trova a dover gestire con un più alto grado di attenzione.

4.1. *MySQL*

MySQL (<http://www.mysql.it/>), recentemente acquistata da Sun per un miliardo di dollari, viene fondata in Svezia durante gli anni '90. L'azienda sviluppa un software che gestisce database relazionali e inizialmente distribuisce il programma utilizzando una licenza creata *ad hoc* che regola l'utilizzo del software. L'azienda bilancia apertura e presidio rilasciando apertamente il codice del proprio prodotto ma presidiando specifici usi del software da parte di alcune specifiche tipologie di clienti, cui chiede un corrispettivo (Dahlander, 2005). Questa strategia si rivela tuttavia difficile da gestire, poiché il bilanciamento tra presidio e apertura è definito in maniera *imprecisa, dinamica e mutevole*. Come spiega un manager dell'impresa:

“Prima era difficile dire se [un cliente] dovesse pagare per il nostro prodotto o no. Percepivamo che qualcuno stava usando in modo improprio la nostra licenza e che la utilizzava in maniera contraria alle nostre intenzioni. Chiudevano il codice o creavano un pacchetto mettendo insieme il database e altro software, così abbiamo dovuto far qualcosa per cambiare questa situazione” (Dahlander, 2005; p. 273, nostra traduzione)

All'inizio degli anni 2000 il management ha quindi optato per una nuova strategia adottando il modello del Dual Licensing. Ha dunque distribuito apertamente il software utilizzando la GNU General Public License (GPL), una delle licenze più restrittive in tema di fusione con software distribuito con licenze dissimili, e ha poi proposto ai propri clienti lo stesso prodotto ma con una licenza creata *ad hoc* che permetteva l'utilizzo del software in forma “chiusa”, cioè senza dover necessariamente rivelare il codice. In questo modo la compagnia ha spostato il proprio presidio da una posizione di difficile gestione (“alcuni usi da parte di alcuni clienti”) ad una posizione di più

facile attuazione (“l’utilizzo commerciale di parte di chi acquista il prodotto tramite il secondo tipo di licenza”), rendendo sostenibile il bilanciamento tra apertura e presidio.

Come giustamente nota Dahlander (2005), tuttavia, MySQL non si limita a distribuire il proprio prodotto tramite una strategia di Dual Licensing. Un’ampia fetta degli introiti si deve alla proposta di servizi collegati al prodotto. Il business model Dual Licensing viene dunque mescolato con quello denominato Product Specialist, in cui l’impresa fornisce servizi collegati ad uno specifico prodotto. In questo caso è interessante notare che non solo l’azienda utilizza un ibrido, mescolando due modelli di business diversi, ma soprattutto che questi due modelli si rafforzino a vicenda. Il ruolo di Product Specialist rafforza la strategia di Dual Licensing poiché assicura ai clienti che il supporto di cui necessiteranno per gestire un prodotto che non hanno acquistato in versione open source ma con licenza “chiusa” verrà fornito proprio dall’impresa che presidia la conoscenza del prodotto. Al tempo stesso, il fatto di poter utilizzare il prodotto in modi difforni da quelli previsti dalla GNU GPL grazie al Dual Licensing permette di immaginare una nuova serie di servizi che ampliano l’offerta di servizi forniti da MySQL e quindi la sua base di offerta come Product Specialist.

4.2. Cendio

Cendio (<http://www.cendio.com/>), azienda svedese fondata all’inizio degli anni ’90, ha come caratteristica l’aver provato diversi business model lungo il suo arco di vita, mescolando la fornitura di servizi con la produzione di codice (Dahlander, 2005). All’inizio degli anni 2000 la compagnia decide di cambiare strategia, e si focalizza sul presidio di un prodotto “chiuso” e commercializzato in maniera “ortodossa”, ma costituito a partire dalla composizione di più software open source. La costruzione di un *bundle* di prodotti open source all’interno di un *framework* proprietario ha avuto due effetti principali. Da un lato è stato possibile tagliare drasticamente i costi di sviluppo, poiché il software incluso nel *bundle* veniva sviluppato esternamente dalla comunità di sviluppatori raccolti attorno ad i progetti open source. Dall’altro, l’impresa ha dovuto acquisire una non facile *expertise* nella gestione delle relazioni con questa comunità, soprattutto a livello legale e di gestione delle licenze.

Il business model applicato in questo caso è una generalizzazione del R&D Cost Sharing ed una sua ibridazione con il modello denominato Indirect Revenues, in cui l’impresa utilizza l’apertura del software per tagliare i costi di sviluppo, e presidia un prodotto chiuso e complementare al software open source su cui questo si fonda.

In questo caso i due modelli che vengono fusi non realizzano lo stesso circolo virtuoso descritto per MySQL, ma al contrario generano una tensione che costringe l'impresa ad un più alto livello di attenzione. Come nel caso del modello basato su R&D Cost Sharing l'impresa non ha una stretta connessione con la comunità di sviluppatori dei progetti di cui usa il software. Ciò determina l'impossibilità di ampliare oltre un certo limite l'apertura, non realizzando quel *feed-back* positivo sul mercato del prodotto complementare tipico del modello basato invece sulle Indirect Revenues. Allo stesso tempo l'impresa deve gestire con attenzione l'interfaccia tra il proprio prodotto proprietario le licenze dei prodotti open source che utilizza. Alle già complesse competenze necessarie per presidiare la conoscenza del prodotto in un modello di R&D Cost Sharing, si viene quindi ad aggiungere anche l'incertezza del presidio di un prodotto basato su una compatibilità tra componenti *open* e *closed* (come per Indirect Revenues) di natura mutevole e dinamica.

5. Conclusione

La discussione sin qui svolta ha avuto lo scopo di mostrare come sia possibile realizzare business anche in un contesto “di rete”, in cui *l’apertura* verso l’esterno dei processi produttivi e innovativi interni all’azienda sia la via necessaria per ancorarsi al network. Si è voluto mostrare come questa apertura, a prima vista percepibile come un fattore di debolezza, in grado di minare qualsiasi modello di business che volesse appoggiarsi ad una rete, si può invece rivelare “una leva”, un fulcro su cui fare perno per moltiplicare la domanda e la capacità produttiva oltre i limiti tipici delle piccole imprese.

Lo strumento che permette questa “rivoluzione” è il *presidio della posizione* nel network da parte dell’impresa che decide di agganciarsi alla rete. Infatti, se da un lato l’impresa in questione produce valore per tutto il network aprendo i propri processi innovativi e produttivi agli altri partecipanti, dall’altro se detiene il controllo di una risorsa fondamentale e difficilmente imitabile (cioè se presidia la sua posizione nel network) può appropriarsi di un congrua fetta del valore prodotto.

Per rendere questa discussione il più possibile concreta, nel testo sono stati analizzati i business model applicati dalle imprese in uno dei settori che hanno fatto dell’apertura estrema una loro caratteristica strutturare: l’open source software. Poiché la maggior parte del software distribuito con licenze di tipo open source è ampliabile, copiabile, modificabile e redistribuibile ma solo se il regime applicato ai prodotti derivati rimane invariato (cioè totalmente aperto), risulta evidente che questo prodotto gode della massima apertura possibile, e quindi è un caso perfetto per analizzare la nostra ipotesi. I vari business model costruiti in questo settore e analizzati nelle precedenti sezioni sia su larga scala che con casi studio specifici hanno mostrato come la chiave per costruire impresa in un contesto di rete e di apertura totale sia proprio il bilanciamento tra apertura del software e il presidio da parte delle singole imprese di diverse settori della conoscenza necessaria per il suo ampliamento o utilizzo.

La conclusione è che in una rete partecipata da diversi soggetti è in effetti possibile costruire tutta una serie di modelli di business “aperti” semplicemente facendo leva sulla capacità delle imprese di presidiare diversi tipi di conoscenza necessari alla rete per produrre valore, bilanciando in modo creativo apertura e presidio.

6. Bibliografia

Arrow K., (1962) Economic Welfare and the Allocation of Resources for Innovation, 1962, in Nelson, editor, *The Rate and Direction of Inventive Activity*

Barbarito L. (2008) (a cura di), *Economia dell'open source*, Pearson Paravia.

Becattini G. (2001). *The Caterpillar and the Butterfly. An Exemplary Case of Development in the Italy of the Industrial Districts*, Le Monnier, Firenze

Benkler Y. (2006), *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale University Press, New Haven, traduzione italiana *La ricchezza della Rete. La produzione sociale trasforma il mercato e aumenta le libertà*, Egea Università Bocconi Editore, Milano, 2007

Benussi L., Rossi C. (2007), *L'Open Source: il caso della provincia di Torino*. in Antonelli c (ed.) *Analisi Economica dell'Innovazione nei Servizi di Rete nel Caso Piemontese*. Report Finale del progetto di collaborazione tra il Dipartimento di Economia "S. Cogne di Martini" dell'Università degli Studi di Torino e il CSI Piemonte.

Berger Suzanne (2005) *How We Compete: What Companies Around the World Are Doing to Make It in the Global Economy*. New York: Doubleday.

Bonaccorsi A., Giannangeli S., Rossi C (2006). *Entry strategies under competing standards. Hybrid business models in the Open Source software industry*. Management Science, 52(7), 1085-1098.

Bonaccorsi, Andrea; Rossi, Cristina, e Scateni, Alessandro (2003), *An analysis of Open Source production in Italy*, Final report of a survey on Italian firms that do business with Open Source software. Online: <http://hdl.handle.net/2038/960>

Chesbrough, Henry (2003) *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Boston: Harvard Business School Press.

Chesbrough, Henry, Wim Vanhaverbeke, and Joel West, (2006), *Open Innovation: Researching a New Paradigm*. Oxford: Oxford University Press.

Corsino, M., Giuri. P. e Torrisi, S. (2005) *L'adozione di software proprietario e open source nelle imprese italiane. Indagine su un campione di imprese manifatturiere italiane*, Rapporto di ricerca, Laboratorio di Economia e Management, Scuola Superiore Sant'Anna, Pisa, febbraio.

Daffara Carlo, (2009), *Free/Libre Open Source Software: a guide for SMEs*, Report edition: 4, July 2009, created in the context of the FLOSSMETRICS EU project, IST Programme 2.S.5, contract FP6-033982, www.flossmetrics.eu.

Dahlander, L. (2005). Appropriation and Appropriability in Open Source Software. *International Journal of Innovation Management*, 9(3): 259-285

Dalle, J.-M., David, P.A., Ghosh, R.A., Wolak, F.A. (2004). *Free & open source software developers and 'the economy of regard': Participation and code-signing in the modules of the Linux kernel*. Presented at The Oxford Workshop on 'Libre Source,' Oxford Internet Institute, 25–26 June.

Lerner J., Tirole J. (2005) *The Scope of Open Source Licensing*, The Journal of Law, Economics, and Organization, vol. 21, pg. 20-56

Lerner, J., Tirole, J., (2002). *Some simple economics of open source*. Journal of Industrial Economics 50 (2), 197–234.

Picerno A. e De Rossi A. (2009), *L'offerta Open Source in Italia: analisi di un settore in evoluzione*, VIU Papers 01.09

Roberts, J.A., Hann, I.-H., Slaughter, S.A., (2006). *Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects*. Management Science 52 (7), 984–999.

Stewart, K. J. & Gosain, Sanjay (2006). *The Impact of Ideology on Effectiveness in Open. Source Software Development Teams*. MIS Quarterly (30:2), 291-314

Tapscott D., Williams A.D. (2006), *Wikinomics. How Mass Collaboration Changes Everything*, Portfolio, Penguin Group (USA), traduzione italiana *Wikinomics. La collaborazione di massa che sta cambiando il mondo*, Rizzoli, Milano, 2007

von Hippel, E., (2002). *Open source projects as horizontal innovation networks - by and for users*, Working paper 4366-02, MIT Sloan School of Management, Cambridge, MA. Available at SSRN: <http://ssrn.com/abstract=561744>.

von Hippel, E., von Krogh, G., (2003). *Open source software and the 'private-collective' innovation model: Issues for Organization Science*. *Organization Science* 14 (2), 209–223.