



Perchè i brevetti del software  
sono un male per l'Europa  
(e non solo)

Massimo Portolani  
(28-02-2005)



*Note sull'autore: Massimo Portolani è imprenditore nel settore elettronico dal 1981, svolge attività di programmazione dal 1978 ed è laureato in Science Internazionali e Diplomatiche all'Università di Bologna. Questo testo è stato composto con il solo fine di portare all'attenzione degli interessati un punto di vista che si ritiene pertinente. Quelle riportate in questo testo sono valutazioni personali dell'autore e non vincolano imprese o altre organizzazioni a lui collegate.*



## **Brevetti del software? no grazie!**

La proposta di introdurre in Europa la possibilità di brevettare il software, in modo analogo a quanto già reso possibile negli Stati Uniti, avrà conseguenze deleterie per lo sviluppo di software da parte di programmatori indipendenti e piccole e medie aziende, mentre contribuirà ad aumentare la dipendenza dalle grandi aziende statunitensi. Sarà un passo nella direzione opposta rispetto ai principi sull'innovazione in Europa stabiliti a Lisbona.

Esiste già la legge sul diritto d'autore e la normativa sui marchi che tutelano a sufficienza la copia servile di un programma per elaboratore o un gioco per computer o un'interfaccia grafica, con il vantaggio della certezza dell'applicabilità, al contrario di un brevetto relativo ad una funzione astratta di difficile comprensione e definizione.

L'unico motivo per cui un simile progetto nefasto rischia di avere successo è la tecnicità dell'argomento, che può indurre persone normalmente sagge a fare valutazioni errate, con conseguenze catastrofiche.

Poichè ho cominciato a scrivere software nel 1978 e continuo a farlo, spero di essere in grado di dare un punto di vista informato sui fatti, che possa aiutare ad aprire gli occhi sui risvolti del problema.

Questo è il motivo per cui, dopo essermi confrontato con colleghi e amici, ho deciso di scrivere questo testo, che affido alla rete perchè lo diffonda.



## Copyright (diritto d'autore) e Patent (brevetto)

La prima differenza che si incontra nello studiare la materia è capire la differenza fra diritto d'autore, brevetto e deposito di marchio. Mi limito a dare qualche breve cenno che suggerisco di approfondire.

**La legge sul diritto d'autore** (*copyright* in inglese) protegge le opere dell'ingegno di carattere creativo.

La legge italiana sul diritto d'autore attualmente protegge anche i programmi per elaboratori, in quanto opere letterarie, stabilendo peraltro, nell'ultima revisione con le modifiche del 22 maggio 2004, la impossibilità per l'autore di impedire lo studio del metodo di funzionamento del programma (art.64 ter)<sup>1</sup>.

In pratica la legge sul diritto d'autore applicata al software ne impedisce la copia ma non impedisce che qualcuno riscriva il programma in modo diverso, al fine di ottenere lo stesso risultato funzionale.

La protezione per il diritto di autore è stata nel tempo estesa, fino ad arrivare a 70 anni dalla morte dell'autore.

---

<sup>1</sup> [http://www.interlex.it/testi/41\\_633.htm](http://www.interlex.it/testi/41_633.htm)

Legge 22 aprile 1941 n. 633

Protezione del diritto d'autore e di altri diritti connessi al suo esercizio

(G.U. n.166 del 16 luglio 1941)

(Testo coordinato con le modifiche introdotte dalla legge 22 maggio 2004, n. 128

### Art. 1

Sono protette ai sensi di questa legge le opere dell'ingegno di carattere creativo che appartengono alla letteratura, alla musica, alle arti figurative, all'architettura, al teatro ed alla cinematografia, qualunque ne sia il modo o la forma di espressione.

Sono altresì protetti i programmi per elaboratore come opere letterarie ai sensi della convenzione di Berna sulla protezione delle opere letterarie ed artistiche ratificata e resa esecutiva con legge 20 giugno 1978, n. 399, nonché le banche di dati che per la scelta o la disposizione del materiale costituiscono una creazione intellettuale dell'autore.

.....

### Art. 64-ter

1. Salvo patto contrario, non sono soggette all'autorizzazione del titolare dei diritti le attività indicate nell'art. 64-bis, lettere a) e b), allorché tali attività sono necessarie per l'uso del programma per elaboratore conformemente alla sua destinazione da parte del legittimo acquirente, inclusa la correzione degli errori.

2. Non può essere impedito per contratto, a chi ha il diritto di usare una copia del programma per elaboratore di effettuare una copia di riserva del programma, qualora tale copia sia necessaria per l'uso.

3. Chi ha il diritto di usare una copia del programma per elaboratore può, senza l'autorizzazione del titolare dei diritti, osservare, studiare o sottoporre a prova il funzionamento del programma, allo scopo di determinare le idee ed i principi su cui è basato ogni elemento del programma stesso, qualora egli compia tali atti durante operazioni di caricamento, visualizzazione, esecuzione, trasmissione o memorizzazione del programma che egli ha il diritto di eseguire. Le clausole contrattuali pattuite in violazione del presente comma e del comma 2 sono nulle.



**Il concetto di brevetto**<sup>2</sup>, (*patent* in inglese) si applica ad un'invenzione che abbia la caratteristica di novità. E' di fatto un monopolio legale<sup>3</sup> che dura 20 anni e che è stata concesso per permettere all'inventore di recuperare le spese, promuovendo così l'innovazione, e per ottenere la pubblicazione dei dettagli dell'inventore.

In pratica l'inventore rende pubblico il funzionamento di una nuova scoperta, e viene ricompensato con un periodo di esclusiva per la produzione e la commercializzazione del prodotto oggetto dell'invenzione.

---

<sup>2</sup> Brevetto: <http://hal9000.cisi.unito.it/ricerca/brevetti2.htm>

Definizione di brevetto

Il BREVETTO è lo strumento giuridico che conferisce all'autore di un'invenzione il monopolio temporaneo di sfruttamento dell'invenzione stessa, ossia il diritto di escludere terzi dall'attuare l'invenzione e dal trarne profitto. Il brevetto rappresenta pertanto un monopolio legale, se pur limitato territorialmente e temporalmente. Tale monopolio legale si giustifica con il fatto che il sistema brevettuale è basato su una forma di scambio: il titolare del brevetto riceve protezione per la propria invenzione e in cambio è obbligato a svelare e a descrivere l'invenzione. Le domande di brevetto e i brevetti già concessi sono infatti pubblicati dagli uffici brevetti di tutto il mondo e ciò li rende una primaria fonte di informazione tecnico-scientifica. Offrendo protezione in cambio di divulgazione, il sistema brevettuale crea incentivi ad investire in ricerca e sviluppo e garantisce alla società l'acquisizione immediata delle idee innovative.

<sup>3</sup> <http://www.autoreonline.net/LEGISLAZIONE/9.htm>

(Pubblicato nella Gazzetta Ufficiale del 14 agosto 1939, n. 189 e aggiornato al 2003).

TITOLO I

Diritti di brevetto

Art. 1.

I diritti di brevetto per invenzione industriale consistono nella facoltà esclusiva di attuare l'invenzione e di trarne profitto nel territorio dello Stato, entro i limiti ed alle condizioni previsti da questo decreto. Tale facoltà esclusiva si estende anche al commercio del prodotto a cui l'invenzione si riferisce, ma si esaurisce una volta che il prodotto stesso sia stato messo in commercio dal titolare del brevetto o con il suo consenso nel territorio dello Stato. La facoltà esclusiva attribuita dal diritto di brevetto non si estende, quale che sia l'oggetto dell'invenzione: a) agli atti compiuti in ambito privato ed a fini non commerciali, ovvero in via sperimentale, b) alla preparazione estemporanea, e per unità di medicinali nelle farmacie su ricetta medica, e ai medicinali così preparati (1).

(1) Articolo così sostituito dall'art. 1, d.P.R. 22 giugno 1979, n. 338.

Art. 1-bis (1).

1. In particolare il brevetto conferisce al titolare i seguenti diritti esclusivi: a) se oggetto del brevetto è un prodotto, il diritto di vietare ai terzi, salvo suo consenso, di produrre, usare, mettere in commercio, vendere o importare a tali fini il prodotto in questione; b) se oggetto del brevetto è un procedimento, il diritto di vietare ai terzi, salvo suo consenso, di applicare il procedimento, nonché di usare, mettere in commercio, vendere o importare a tali fini il prodotto direttamente ottenuto con il procedimento in questione .

(1) Articolo aggiunto dall'art. 12, d.lg. 19 marzo 1996, n. 198.



### **Protezione del marchio, (*trade mark* in inglese).**

Oltre al diritto d'autore per le opere dell'ingegno di tipo creativo e al brevetto per le invenzioni, esiste la possibilità di depositare il marchio di impresa<sup>4</sup>.

Si tratta del simbolo o del nome che contraddistingue un'impresa o un ramo delle sue attività. La protezione garantita per un marchio originale è di 10 anni, rinnovabili. Una causa legale per contraffazione di marchio registrato è relativamente veloce e mette un potente strumento in mano alle imprese per rendere i propri prodotti immediatamente distinguibili da quelli della concorrenza.

Questo mi pare senz'altro giusto e di fatto garantisce, a chi investe in promozione, di poter raccogliere i frutti del suo investimento; inoltre evita confusione fra i consumatori.

Poichè i consumatori spesso si limitano ad un'analisi superficiale del prodotto e scelgono in base ai marchi ed alla pubblicità, mi pare che questa sia un'arma sufficientemente potente per garantire alle imprese una certa forma di monopolio.

---

<sup>4</sup> Legge italiana Marchi

Legge Marchi Italiana ai sensi del Regio Decreto No.929 del 21 giugno 1942 e successive modifiche.

ART. 1

1. I diritti del titolare del marchio d'impresa registrato consistono nella facoltà di far uso esclusivo del marchio. Il titolare ha il diritto di vietare ai terzi, salvo proprio consenso, di usare:

un segno identico al marchio per prodotti o servizi identici a quelli per cui esso è stato registrato;

un segno identico o simile al marchio registrato, per prodotti o servizi identici o affini, se a causa dell'identità o somiglianza fra i segni e dell'identità o affinità fra i prodotti o servizi possa determinarsi un rischio di confusione per il pubblico, che può consistere anche in un rischio di associazione fra i due segni;

un segno identico o simile al marchio registrato, per prodotti o servizi non affini, se il marchio registrato goda nello Stato di rinomanza e se l'uso del segno senza giusto motivo consente di trarre indebitamente vantaggio dal carattere distintivo o dalla rinomanza del marchio o reca pregiudizio agli stessi.

2. Nei casi menzionati al comma 1 il titolare del marchio può in particolare vietare ai terzi di apporre il segno sui prodotti o sulle loro confezioni; di offrire i prodotti, di immetterli in commercio o di detenerli a tali fini, oppure di offrire o fornire i servizi contraddistinti dal segno; di importare o esportare prodotti contraddistinti dal segno stesso; di utilizzare il segno nella corrispondenza commerciale e nella pubblicità.

(Fonte: <http://www.autoreonline.net/LEGISLAZIONE/9.htm>)



## Pregi e difetti di queste protezioni

Esiste un conflitto di interessi fra l'autore, l'inventore e la società. Samuel Clemens, alias Mark Twain, fu, in età matura, un convinto assertore del diritto di autore e si chiese persino perchè ci dovesse essere un limite al godimento della proprietà di tale diritto, limite che, ad esempio, non viene posto al godimento della proprietà privata.

Un ragionamento di questo genere parte dal presupposto che l'opera sia interamente di proprietà dell'autore, che fluisca dalla sua mano.

Onestamente non mi pare corretto, poichè l'autore è vissuto, si è nutrito, ha studiato e si è ispirato a opere esistenti, e quindi ha beneficiato dell'humus fornito dalla società.

Mi pare giusto garantire una protezione, ma anche porre un limite; trascorso questo periodo è bene che l'opera ritorni ad essere di tutti. Il limite attuale di settant'anni dopo la morte dell'autore mi pare francamente eccessivo, ma in ogni caso non crea tanti problemi quanto invece ne possono creare i vent'anni previsti per i brevetti, specialmente in industrie soggette a rapido mutamento.

Nel caso dei brevetti, la motivazione per la protezione è permettere a chi inventa di avere un beneficio che costituisce al contempo un incentivo. In cambio di questo monopolio legale, l'inventore rivela la sua invenzione e in questo modo contribuisce a diffondere la conoscenza.

Occorre sempre tenere a mente che garantendo protezione a qualcuno si congela lo sviluppo e la competizione in quel settore perchè si concede un monopolio. E fondamentale bilanciare danni e benefici e garantire protezione solo nel caso in cui gli investimenti in gioco siano rilevanti. Non si può lasciare mano libera a chi considera il brevetto come una sorta di vincita alla lotteria o l'autorizzazione a vivere di rendita.

Dal concetto di invenzione brevettabile sono esclusi, per ora, in Europa, gli algoritmi matematici e i programmi per elaboratore, che vengono considerati alla stregua degli algoritmi. Sono anche esclusi nonchè i metodi commerciali. Negli Stati Uniti, in una sentenza della corte suprema del 1981 relativa al caso *Diamond v. Diehr*, fu ritenuto per la prima volta possibile brevettare un programma per elaboratore, poichè facente parte di un processo produttivo. Si trattava di un programma di gestione delle temperature in un sistema per la fusione della gomma.



E' importante differenziare un programma che serve in un processo che avviene in una macchina specifica, al quale magari (pur con molta cautela) potrà essere concessa la protezione, se utilizzato insieme a quella macchina, dalla richiesta di protezione tramite brevetto per funzioni generiche utilizzabili in vari contesti.

In particolare ritengo molto negativo che si possano brevettare metodi di commercio elettronico (es. one-click di Amazon), oppure che si possano brevettare funzioni software come le *macro*, come ha fatto Microsoft nel 2001<sup>5</sup>. E' molto probabile (e auspicabile) che questi brevetti non reggano, però fanno perdere tempo e confondono il mercato, a vantaggio dei potenti.

---

<sup>5</sup> *United States Patent Application 20040216138*

*Assignee name: Microsoft Corporation*

*Description: Method and system for processing input from a command line interface*

*Abstract*

*A method and system for processing input from a command line is provided, in which the command line contains a macro that gets parsed by a command line processor. The processor replaces the macro with the appropriate command and executes the command line. The macro may contain a message that the processor displays to prompt the user to identify the command that is to replace the macro. Other possible functions that the macro may perform include, but are not limited to: executing a function to generate a string to replace the macro, executing a function to generate another macro with which the first macro may be replaced, and prompting the user to enter the name of a dynamic-linked library (DLL) from which the replacement string for the macro may be obtained.*

Mia traduzione:

"Metodo e sistema per processare dei dati ricevuti da una interfaccia a linea di comando

Riassunto

Un metodo e sistema per processare i dati ricevuti da una linea di comando, in cui la linea di comando contiene una macro (n.d.r. si intende macro-istruzione o macro-comando) che viene analizzata da un processore della linea di comando. La macro può contenere un messaggio che il processore mostra richiedendo allo stesso tempo all'utente di identificare il comando che deve rimpiazzare la macro. Altre possibili funzioni che la macro può realizzare includono, tra l'altro: eseguire una funzione per generare una stringa alfanumerica che rimpiazza la macro, eseguire una funzione che generi un'altra macro con la quale la prima macro possa essere rimpiazzata, e richiedere all'utente di digitare il nome di una libreria dinamicamente collegata (DLL) dalla quale si possa ottenere la stringa alfanumerica che rimpiazza quella macro."

Mia spiegazione: anche se ci si perde in questo tipo di linguaggio (e questo serve come monito) la realtà è banale. Da quando faccio il programmatore, cioè dal 1978, si è sempre usato scrivere, per comodità, delle routine che realizzavano certi comandi e chiamarle con un nome che le identificasse. Questa è una macro. Per esempio se avessi voluto aprire tutti i documenti in modo che già presentassero il mio nome e la data, avrei fatto una macro che avrei chiamato, ad esempio, APRIBENE e semplicemente digitando questo nome avrei messo in moto un processo che consisteva in aprire il documento e scriverci un testo predisposto un un file, nonché posizionare il cursore ove avrei cominciato a scrivere. E' evidente che alla Microsoft non possono credere seriamente che questa, nel 2001 (data del deposito), sia un'invenzione, però intanto cominciano a cercare di ottenere il brevetto e poi si vedrà. Ci vogliono soldi anche per far dichiarare nullo un brevetto e per una piccola ditta può essere una catastrofe dover fronteggiare in tribunale, di fronte ad avvocati esperti e ben pagati, anche una sciocchezza di questo genere.

Il testo completo è disponibile in linea: <http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnetacgi%2FPTO%2FSrchnum.html&r=1&f=G&l=50&s1=%2220040216138%22.PGNR.&OS=DN/20040216138&RS=DN/20040216138>.

A questo esempio sono arrivato a partire dal sito:

[http://lawgeek.typepad.com/lawgeek/silly\\_patents\\_trix\\_are\\_for\\_kids/](http://lawgeek.typepad.com/lawgeek/silly_patents_trix_are_for_kids/)





Per capire come questi concetti si applichino al *software* occorre innanzitutto dire cosa questo sia; questo è ciò che mi accingo a fare.

## Cos'è il software e chi lo scrive

Una definizione banale, da dizionario, ci dice che il software è il codice che permette ad un elaboratore di svolgere alcune funzioni. In realtà ci sono parecchie precisazioni da fare. Esiste quello che si definisce *firmware* per una scheda elettronica, o BIOS (Basic Input Output System) in un computer, che è il codice che controlla direttamente l'hardware, scrive sul disco, accende luci, genera suoni o controlla temperature.

Esiste poi, per i computer, quello che viene definito *sistema operativo*, che esegue le funzioni di base e presenta all'utente un ambiente più o meno semplice da usare e fornisce ai programmi che vengono fatti girare in questo ambiente alcune funzioni *primitive* per interagire con la macchina. Ad esempio, un programma di grafica che traccia una linea usa le primitive grafiche del sistema operativo che a sua volta utilizza le funzioni di basso livello di scrittura dei pixel sullo schermo fornite dal BIOS.

I sistemi operativi più diffusi sono Windows<sup>6</sup>, che detiene la maggior quota di mercato, Mac OS e Linux o altri sistemi Unix-like, come Unix stesso o Free BSD. In dettaglio: Windows detiene senz'altro una quota superiore al 70% per i PC, Mac OS detiene la maggioranza per i sistemi di editoria elettronica e grafica, Linux e Free BSD o simili sistemi derivati da Unix, detengono probabilmente il primato per quanto riguarda i server.

Quindi, un sistema operativo è quello che l'utente percepisce essere il computer, anche se il sistema operativo non serve a nulla senza i programmi applicativi, che vengono fatti *girare* in quell'ambiente.

Chi produce sistemi operativi, se produce anche programmi applicativi, ha un innegabile vantaggio competitivo, poichè sa come far *girare* al meglio i programmi nel suo ambiente (magari anche con scorciatoie che vanno direttamente alla funzione), può dare con ritardo informazioni su modifiche presenti in nuove versioni, in modo da creare l'impressione che l'applicativo di un'impresa concorrente non funzioni.

---

<sup>6</sup> Il marchio Windows come tutti gli altri marchi che citerò sono registrati dalle rispettive aziende proprietarie.



Una ditta che produce sistemi operativi può agevolmente inglobare nel sistema ogni nuova idea introdotta da concorrenti, come ad esempio i browser o programmi multimedia. Questi ragionamenti erano probabilmente alla base della richiesta del giudice Jackson, poi decaduta, di dividere la Microsoft in due aziende, una che producesse sistemi operativi e una che producesse software. Il commissario alla concorrenza CEE Mario Monti deve aver fatto considerazioni analoghe, per decidere di comminare una multa di quasi 500 milioni di Euro a Microsoft per aver incorporato MediaPlayer nel sistema operativo. Da notare tra l'altro, che il principale beneficiato da questa decisione è un concorrente americano di Microsoft, che produce Real Player.

Questo ci porta a ragionare su come sia strutturata l'industria Europea del software: di fatto i programmi più usati, su tutte le piattaforme, sono di produzione USA. In Europa, ci sono ben poche imprese che producono importanti applicativi di largo consumo. Ciò non significa però che non ci sia un'attività di produzione, anche parecchio innovativa. L'Europa è piena di programmatori capaci, che creano, personalizzano, contribuiscono a progetti open-source (i cui frutti sono poi sfruttati gratuitamente anche da Microsoft o Apple, fra gli altri), fanno ricerca all'interno delle Università.

Chi è che veramente crea innovazione nel software? E' la grande azienda che deve mantenere le quote di mercato, o il programmatore indipendente che lavora per passione e magari senza neppure essere retribuito?

Chi ha creato internet? Chi ha scritto i protocolli di comunicazione che tengono in piedi la rete? Chi ha realizzato il sistema operativo più affidabile?

In base alla teoria che sta dietro ai brevetti, certamente sarà stato qualche inventore che sta beneficiando dei suoi vent'anni di monopolio. Niente affatto. Innanzitutto sarebbe stata un'azienda e non un inventore, poichè è impensabile che un programmatore riesca a godere di benefici in un mondo popolato da poche grosse imprese. In ogni caso non è stata neppure una grossa impresa. Se andassimo ad analizzare come si sono evolute tutte queste innovazioni che sono alla base della società dell'informazione alla quale siamo abituati oggi, ci accorgeremmo che si è sempre trattato di regalie liberali da parte di Università, finanziate dallo stato, come nel caso di Free BSD derivato da Unix, sviluppato a Berkeley, oppure di regalie da parte di altri enti statali o internazionali, come nel caso di internet, frutto in parte dei progetti della difesa USA e in parte, per quanto riguarda il browser, dei progetti del CERN di Ginevra.



Si potrebbero riempire pagine con esempi di questo tipo e del resto anche i sistemi operativi *proprietary* prendono a piene mani da progetti finanziati dalla comunità, nel pieno della legalità, poichè questi progetti, proprio perchè vengono finanziati dalla comunità, vengono messi in circolazione con licenze più o meno *liberali*.

Se prendiamo, ad esempio, Microsoft - ed è un esempio che si impone, data la sua onnipresenza - non mi risulta ci siano importanti innovazioni che le si possano attribuire. Questo comportamento è perfettamente comprensibile, un'azienda del genere non ha interesse a grossi cambiamenti. Ai cambiamenti la obbligano gli altri, la reazione tipica è il contrasto, e solo se questo non funziona si arriva alla competizione, agevolata dal fatto di produrre anche il sistema operativo.

Quando Apple introdusse il sistema a icone (inventato al PARC di Palo Alto, (centro ricerche allestito da Xerox) insieme all'Ethernet e a tante altre innovazioni di cui tutti hanno beneficiato), Microsoft si ostinò inizialmente a rimanere legata a MS-DOS, implementando poi a sua volta un sistema a icone molto simile, con Windows95.

Quando Netscape iniziò ad aprire il mercato dei browser (in seguito all'originale invenzione del CERN, ceduta come regalia), Microsoft si vide costretta a correre ai ripari con Internet Explorer. Nel futuro potremo vedere come si evolverà la sfida fra Google e Microsoft nei motori di ricerca (che sono il vero potere in internet).

Ci sono esempi innumerevoli che si potrebbero fare, ma non è questo l'argomento di cui mi sto occupando e del resto non è Microsoft il problema. Qualsiasi azienda con tanto potere di mercato finirebbe, quasi senz'altro, per comportarsi in quel modo, o forse peggio. Quando iniziai a programmare il mercato era in mano all'IBM che produceva grossi computer e le innovazioni venivano da una miriade di piccole aziende: Commodore, Apple, North Star, General Processor, Cromemco per citarne alcune; solo dopo vari anni IBM decise di produrre il PC.

Quello che voglio enfatizzare è che le grandi aziende di questo settore, spesso, fanno i soldi con le innovazioni, a volte dopo averle ostacolate; però quasi mai innovano.



## Il programmatore, chi è costui?

Paul Graham è un programmatore americano che ha scritto un libro, *'Hackers and painters'* (in cui peraltro fa varie affermazioni che non condivido), nel quale paragona una parte dei programmatori, i più bravi, ai pittori, arrivando a mettere sullo stesso piano il gusto estetico con il quale si scrive un certo tipo di *codice* ben fatto, al gusto per i particolari che portava Leonardo a curare dettagli delle sue opere, più per la sua particolare soddisfazione che per la possibilità di comprensione dei suoi committenti. Io non credo che si possa nemmeno lontanamente arrivare a paragoni con artisti di tale portata, ma c'è un fondo di verità. Parlo per me stesso e per i miei amici programmatori con i quali mi sono trovato di fronte a una birra a notte inoltrata, dopo aver speso la giornata davanti ad uno schermo ed una tastiera.

Innanzitutto ci sono vari tipi di programmatori: quelli che sono una sorta di impiegato manutentore di un sistema che non capiscono, quelli che si occupano di una infinitesima funzione in un programma realizzato 'a moduli' come nel caso di progetti di grandi aziende, spesso realizzati da centinaia di persone che non si conoscono in giro per il mondo. Questo tipo di programmatori difficilmente inventerà qualcosa. Ci sono però anche quelli che 'smanettano', che in gergo si definiscono *hackers*<sup>7</sup>, che vogliono sapere come funziona una cosa e che *godono* nel vedere realizzato ciò su cui stanno lavorando, che vivono come una *loro* creazione, né più e né meno di un artista.

Questa è la gente che è alla base di quasi tutte le innovazioni nel software, non le aziende. Per gli economisti risulta difficile capire perchè una persona dedichi ore di vita a un progetto per il quale non viene retribuita. In realtà questo è un esempio del comportamento egoistico, che dovrebbe, secondo gli economisti, sottendere ad ogni azione umana. E' egoismo buono, nel senso di realizzazione e di soddisfazione personale. Il buon programmatore affronta un progetto in due modi: può essere un noioso fardello che "*ti tocca fare*" perchè ti pagano; questo è il lavoro si vive con un malessere fisico, a prescindere dai soldi che vengono pagati. Può essere invece un progetto che "*ti prende*", sul quale passi notti, sabati e domeniche, trascuri la moglie e gli amici ma che deve essere portato a termine finchè dura il fuoco creativo.

---

<sup>7</sup> Purtroppo una pessima stampa ha creato il mito degli hackers cattivi, attentatori alla sicurezza dei sistemi altrui. Io sto usando questo termine nel suo originario senso buono di programmatore 'smanettone' e 'capace'.



Chi non ha mai programmato non si rende conto di questo e non riesce a capire perchè ci sia chi scrive software eccezionale e lo regali, o quasi; il ragionamento è semplicemente questo: *"sono bravo e lo dimostro, a me stesso e ai programmatori miei pari, dei quali mi conquisto così la stima"*.

Per un programmatore di questo genere è molto più importante avere a disposizione il computer più veloce o poter lavorare su progetti divertenti che guadagnare una barca di soldi e annoiarsi su un progetto poco interessante.

E' evidente che chi scrive software si trova di fronte alla necessità di risolvere problemi. Ci sono molti modi per arrivare alla stessa soluzione, ma i modi più efficienti possono essere pensati agevolmente da un italiano come da un francese o un americano (quelli delle barzellette); questi tre programmatori scriveranno routines che eseguono la stessa funzione, senza copiarsi in alcun modo. Dal punto di vista della legge sul diritto di autore non ci sarà alcun problema, poichè quella legge protegge un programma dai plagi ma non entra nel merito delle funzioni. Se invece si permettesse di brevettare il software potrebbe benissimo accadere che una persona, che riscrive in altro modo la stessa funzione, violi un brevetto.

Per un programmatore indipendente o una piccola azienda questo significa passare il tempo a leggersi i brevetti depositati, per capire se facendo una particolare cosa si stia infrangendo il diritto al monopolio di qualcuno. E' evidentemente pura follia pensare che ciò sia possibile. Piuttosto il programmatore continuerà a fare i suoi giochini privatamente ma si guarderà bene dal contribuire a progetti open-source con tutto il danno che ne deriverà per la comunità.

Per la grande azienda invece il brevetto è un ulteriore modo per mantenere il potere. Con centinaia o addirittura decine di migliaia di persone a disposizione, sarà possibile analizzare sistematicamente tutto quello che viene depositato, non solo per evitare di infrangere brevetti, ma anche per copiare meglio, aggirandole, le idee innovative. Sarà possibile depositare di tutto, e poi usare la minaccia contro qualsiasi piccolo concorrente che esca con un'idea innovativa.

Infatti, ciò che si dimentica spesso, è che per depositare un brevetto non basta perdere tempo preparando carte, cosa già di per se odiosa per un programmatore; una volta ottenuto il brevetto occorre difenderlo in tribunale in caso di copia, e questo costa molti soldi e molto tempo. In un settore in cui tutto invecchia in mesi, i tempi della giustizia non sono adeguati.



C'è poi il problema della specificità dell'argomento, una manna per i periti dei tribunali. Purtroppo è assai difficile che un programmatore scaltro faccia il perito per il tribunale, con il risultato che a fornire ragguagli al giudice ci saranno spesso, loro malgrado, persone che capiscono ben poco di quello che devono valutare.

Per tutti questi motivi è un errore imperdonabile permettere che si possano brevettare i programmi per elaboratore, ma è normale che gli USA spingano in tal senso perchè una tale legislazione è totalmente a loro vantaggio. Le imprese americane hanno già migliaia di brevetti di software che aspettano solo di essere estesi all'Europa e al resto del mondo. Ai nostri giorni il potere di mercato si consolida con la proprietà intellettuale.

Per l'Europa è un'altra storia. In Europa c'è ancora tanta imprenditorialità in questo settore, ma ci sono tante piccole aziende, estremamente innovative, ci sono ricercatori universitari e programmatori indipendenti. Per questa gente è già dura dover compilare moduli su moduli per accedere a un contributo, tant'è che preferiscono farne a meno, e io sono uno di quelli.

E' incredibile e folle che si voglia porre un freno alla creatività di questa gente e che ci si spari nei piedi facendo un favore alle aziende americane, imboccando una strada esattamente opposta a quella che si voleva prendere quando a Lisbona si fissarono le priorità per lo sviluppo della tecnologia europea.

Purtroppo al pari del giudice che dovrà giudicare, il deputato che dovrà votare si trova davanti una materia complessa ed è possibile che possa cadere in questo tranullo.

La comunità europea finanzia progetti *esoterici* come *'The disappearing computer'*<sup>8</sup> che mi lasciano alquanto perplesso (meglio sarebbe finanziare un sistema operativo europeo, open source, anche solo riscrivere Unix). Non vorrei che con una legge sui brevetti del software ci trovassimo davanti "*the disappeared european computer programmers*".

---

<sup>8</sup> "Mission Statement: To see how information technology can be diffused into everyday objects and settings, and to see how this can lead to new ways of supporting and enhancing people's lives that go above and beyond what is possible with the computer today."  
<http://www.disappearing-computer.net/mission.html>



## **Un appello a chi può fare qualcosa!**

1- La legge sul diritto d'autore tutela già a sufficienza il software dalle copie servili e non c'è bisogno di ulteriore protezione. Le aziende sono già ben protette dal diritto di autore e dalla legge sui marchi.

2- Il brevetto è una buona cosa per quanto riguarda progetti complessi che richiedono tempo e denaro e che necessitano di anni per il recupero degli investimenti. Lo sviluppo di soluzioni software a problemi non ha questi requisiti. Risolvere un problema con un algoritmo richiede delle valutazioni astratte e capacità creativa, non investimenti. Più persone, senza copiarsi a vicenda e senza conoscersi, possono facilmente e probabilmente arrivare alla stessa soluzione lavorando su codice che deve girare su macchine simili. Quello che può sembrare una grande innovazione al responsabile dell'ufficio brevetti può essere una cosa assai banale, magari semplicemente elaborata da software precedentemente scritto da altri, che potrebbero trovarsi a non poter più usare liberamente quello che avevano scritto loro stessi.

3- Il tempo necessario per preparare le carte, il costo per un eventuale ricerca preventiva, il costo legale di difesa di un brevetto: tutto ciò non spaventa una grande azienda ma rende impossibile per una piccola azienda o per un programmatore indipendente beneficiare di una eventuale legislazione che permetta il brevetto del software.

Evidentemente esperti di brevetti e avvocati non aspettano altro che leggi di questo tipo, ma questo non significa che la società ne tragga dei benefici.

4- Un giudice o un perito di un tribunale potrebbero solo con grande difficoltà valutare in modo adeguato una causa relativa ad un brevetto di software. Il rischio di errore e i tempi di giudizio, in un settore soggetto a rapidi mutamenti, renderebbero molto difficile un processo di questo tipo, con grande vantaggio per le grandi aziende che avrebbero buon gioco a costringere a vantaggiosi (per loro) accordi extragiudiziali, le piccole aziende a rischio di fallimento.

Un sistema capitalistico ha bisogno di certezze e questa legge aggiungerebbe incertezze.



5- Le grandi aziende avrebbero buon gioco a studiare tutti i brevetti depositati e studiare il modo per aggirarli. Le piccole aziende non avendo tempo e denaro per fare queste ricerche si troverebbero sempre con la spada di damocle di un altrui brevetto pendente.

6- I migliori programmatori, che contribuiscono all'open-source per puro piacere personale e per guadagnarsi la stima dei loro pari, smetterebbero di farlo poichè si troverebbero soggetti a rischio di cause legali per avere inventato autonomamente, e regalato, qualcosa che qualcuno aveva depositato.

7- Accertato che un sistema di questo genere beneficia le grosse imprese, e già questo appare iniquo, rimane il fatto in Europa ci sono poche grandi case di Software che non siano distributori di colossi americani. Oltre tutto le imprese USA avrebbero buon gioco a richiedere l'estensione dei loro brevetti ai paesi europei. L'Europa del 2005 farebbe un grande favore alle grandi aziende USA e un grosso dispetto a se stessa se approvasse una legge sulla brevettabilità del software.

8- Non intendo qui contrapporre gli interessi Europei a quelli USA, anzi sono convinto che la normativa vigente negli Stati Uniti costituisca un limite per le capacità di innovazione di quel paese, al cui contributo tecnologico tutti i programmatori guardano con rispetto. L'Europa, non adottando normative deleterie, può essere di esempio e dare un aiuto alle tante persone che anche dall'altra parte dell'Atlantico si battono contro i brevetti del software.

Forlì 28-02-05  
Massimo Portolani