

# Cosa significa Open Source? Cos'è il software libero? Applicazioni ai GIS

“OPEN SOURCE” È UN TERMINE ORMAI DI MODA, ANCHE IN AMBITO GEOGRAFICO. I VANTAGGI DEL SOFTWARE SVILUPPATO CON QUESTO APPROCCIO SONO INNEGABILI, E MOLTI SONO GLI ESEMPI DEL SUO SUCCESSO. È PERÒ FACILE, SENZA UNA CORRETTA DEFINIZIONE DEI TERMINI, FARE CONFUSIONE. CERCHIAMO DI CHIARIRE ALCUNI PUNTI.

Tecnicamente, Open Source significa “a codice sorgente aperto”. La maggior parte dei programmi sono infatti scritti in linguaggi (più o meno) leggibili dagli umani, quali il C, C++, C#, ecc.; questo è detto codice sorgente. Per essere eseguibili da un computer, queste liste di istruzioni devono essere convertite in codice (detto codice eseguibile) illeggibile per noi, ma facilmente comprensibile dal processore. Il processo di conversione da sorgente ad eseguibile è detto compilazione, e viene effettuato da programmi detti compilatori. Il processo è in generale ad una sola direzione, nel senso che è estremamente difficile risalire dal codice eseguibile a quello sorgente. Quando acquistate la licenza di uso di un programma proprietario vi viene fornito il codice eseguibile, il che di fatto rende il programma non modificabile dall'utente. Al contrario, i programmi Open Source sono forniti sia con il codice sorgente che, generalmente, con l'eseguibile (in ogni caso, l'utente può procedere autonomamente alla compilazione). Di conseguenza, è sempre possibile per l'utente modificare (o far modificare da un programmatore) ogni aspetto di un programma Open Source. È quindi possibile per ciascuno, a qualunque livello di preparazione informatica, partecipare allo sviluppo del programma, segnalando bugs, migliorando la documentazione, traducendo le parti che gli sono più utili, ecc. Questo rende inoltre possibile verificare in dettaglio il funzionamento del programma, accertandosi che non ci sia (come avviene) codice nascosto (ad esempio trojan) che esegue, all'insaputa dell'utente, operazioni potenzialmente lesive della pri-

vacy o della sicurezza. Molti programmi proprietari, ad esempio, si collegano con siti non noti, inviando dati relativi alla macchina su cui il programma stesso sta girando. Per questo, alcune agenzie di sicurezza hanno chiesto alle ditte fornitrici di software di poter ispezionare il codice sorgente.

È importante sottolineare come la sola possibilità di leggere il codice sorgente (magari, come talvolta avviene, impegnandosi a non divulgarne o riusarne alcun aspetto) non fornisca grandi garanzie dal punto di vista della sicurezza e dell'assenza di funzionalità malevole o nascoste. Se infatti dal codice sorgente non è possibile ottenere un eseguibile (ma si può solo consultare quello che ci è stato detto essere il sorgente), non si potrà verificare la corrispondenza dell'eseguibile con il sorgente fornito. Inoltre se tutto quello che si può fare è leggere il codice si perde ogni possibilità di effettuare correzioni o adattamenti per le proprie esigenze, che è una delle caratteristiche più importanti del vero Open Source.

La disponibilità del codice sorgente infatti non garantisce l'effettiva libertà di utilizzo del programma; per questo, è necessario che la licenza preveda alcuni elementi essenziali. In generale, perché un programma sia libero deve essere disponibile il suo sorgente, ma non è necessariamente vero il contrario (ad esempio, una software house può consentire la lettura del codice sorgente, senza concederne la modifica o il riuso). Per questo è nata l'iniziativa “Open Source” (Open Source Initiative: [www.opensource.org](http://www.opensource.org)), che elenca e certifica le condizioni minime perché un software si possa considerare Open Source. Occorre inol-

tre distinguere chiaramente fra programmi gratuiti, ma non liberi, e programmi liberi (che generalmente, ma non necessariamente, sono gratuiti). Ad esempio, il lettore di animazioni flash è gratuito e liberamente scaricabile, ma non modificabile; il suo codice sorgente non è infatti disponibile. Molti programmi scaricabili da Internet, o che si trovano preinstallati all'acquisto del computer (spesso genericamente chiamati freeware) sono di questo tipo. È importante capire che nessuno di questi dà i vantaggi fondamentali dei programmi liberi.

#### LE QUATTRO LIBERTÀ

Dunque essere in possesso del sorgente, pur essendo un vantaggio, non è sufficiente a rendere un programma libero, se ad esempio la licenza con cui viene fornito impedisce di cambiarlo, migliorarlo o distribuirlo ad altri. Perciò, quello che conta non è tanto l'aver a disposizione il sorgente, quanto una serie di libertà fondamentali perché il software si possa dire libero:

- la libertà di usare il programma per qualunque scopo; questa è una libertà meno ovvia di come potrebbe sembrare, se considerate che molte licenze di software proprietario impongono restrizioni, talvolta significative. Ad esempio, un noto database di livello enterprise vieta di pubblicare le valutazioni comparative in cui si confronta quel database con altri analoghi, impedendo così confronti razionali;
- la libertà di studiare come è fatto il programma, il che rende implicitamente necessario avere a disposizione il codice sorgente;
- la libertà di copiare e distribuire il programma a chiunque; questo consente di distribuire gratuitamente il software senza dovere pagare licenze;
- libertà di modificare il programma e di distribuire le proprie modifiche; questo è particolarmente importante in campo geografico, dove gli utenti hanno spesso esigenze molto specifiche, che possono essere meglio soddisfatte da programmi opportunamente modificati (spesso semplificati, talvolta integrati con soluzioni ad hoc). Inoltre, le modifiche realizzate, se utili ad altri, possono essere incluse nel programma.

Sono queste libertà che in ultima analisi definiscono quello che è il software libero ed Open Source. Si noti che da nessuna parte c'è scritto che il software deve essere ceduto o sviluppato gratuitamente, anzi è più che legittimo farsi pagare per il proprio lavoro e richiedere un compenso per creare e distribuire software libero. Perché si tratti di software libero queste quattro libertà devono però essere sempre garantite.

Perché allora praticamente tutto il software libero ed Open Source è scaricabile gratuitamente da Internet? Semplicemente perché il costo di duplicazione e distribuzione è praticamente nullo, per cui

chiunque acquisti un software può a sua volta redistribuirlo senza spendere; in breve tempo prevale quindi la tendenza alla distribuzione gratuita.

Ci si può allora chiedere quale sia il vantaggio di creare e sviluppare questo software: le risposte sono molte, il primo vantaggio è che se non c'è da pagare una licenza, c'è comunque da pagare per assistenza, sviluppo e formazione, e molte imprese e professionisti basano le loro attività sul software libero, e contribuiscono al suo sviluppo in modo da poter continuare a fornire questi servizi.

Per un'impresa che opera in questo campo, infatti, usare software libero significa in sostanza essere proprietari (seppure non in via esclusiva) del principale strumento che si utilizza nella propria attività; questo permette di svincolarsi da tutte le logiche (e le strategie commerciali) di lock-in adottate comunemente dai produttori di software proprietario, come le politiche che costringono ad aggiornamenti forzati per mancanza di supporto o correzioni, o addirittura il ritiro dal commercio di software giudicati non più redditizi. Questo consente a chi usa software libero di guadagnare una preziosissima indipendenza strategica, non dovendo più dipendere dalle logiche commerciali e dalle decisioni prese da un fornitore esterno.

Un ulteriore grandissimo vantaggio è dato dal fatto che si possono eseguire personalizzazioni ed adattamenti a piacere, rivolgendosi direttamente al mercato per richiedere lo sviluppo di funzionalità mancanti, senza essere vincolati alle scelte di un produttore o ai tempi di rilascio di quest'ultimo. Inoltre essendo il software libero a disposizione di tutti, sarà possibile operare in un mercato intrinsecamente aperto, dato che chiunque abbia sufficienti competenze potrà operare su di esso. Questo consente di investire ad esempio nello sviluppo di funzionalità mancanti piuttosto che nell'acquisto di licenze.

Soprattutto per le strutture più grandi, un evidente vantaggio viene dal fatto che non si dovranno più mettere in atto complesse politiche di gestione delle licenze, non essendoci limiti né sul numero di installazioni, né sul numero di utilizzatori. L'utente sarà cioè in grado di espandere l'utilizzo senza dover affrontare quelli che in molti casi sono costi veramente onerosi (si pensi a tutti quei programmi che, superato un certo limite di utenti, devono essere acquistati in versioni soggette a licenze molto più care).

Infine, la possibilità di provare molti software senza impegno economico abbassa sensibilmente la soglia d'ingresso nel mercato, un fattore particolarmente importante per i giovani professionisti.

#### TIPICI DI LICENZE

Benché, per essere libero, il software richieda semplicemente il rispetto delle quattro libertà appena enunciate, questo all'atto pratico è garantito dall'uso di una licenza, proprio come per il software

## Libero e Open Source

proprietario. Infatti, uno degli errori più comuni è quello di pensare che il software libero sia senza licenza. Non è così se non in casi molto particolari (il cosiddetto public domain, o pubblico dominio, in cui l'autore rinuncia completamente al copyright; una opzione ormai piuttosto rara). Un software distribuito senza licenza infatti non potrebbe essere né copiato né modificato né ceduto ad altri perché, per la normativa del diritto d'autore, i diritti di distribuzione e modifica sono esercitabili solo dall'autore stesso. Potrete fare queste cose solo quando avete una licenza che ve lo consente esplicitamente. Per questo motivo la stragrande maggioranza del software libero è dotata di una sua licenza, che fornisce esplicitamente le libertà citate.

Le licenze possibili sono varie, ma una classificazione molto sommaria porta a dividerle in due grandi classi, le licenze cosiddette copyleft e non. Dicendo copyleft si fa in realtà un gioco di parole, traducibile approssimativamente in permesso d'autore che, invece di riservarsi tutti i diritti, concede il permesso di fare quello che usualmente il copyright vieta. Con copyleft in sostanza si vuole indicare una logica rovesciata rispetto al copyright, in cui appunto si danno dei diritti invece che imporre delle restrizioni. La differenza fra queste due classi è che, mentre nel caso delle licenze non copyleft è possibile usare il software libero per produrre anche versioni modificate non libere, con le licenze copyleft il software viene fornito liberamente, ma ogni modifica deve restare libera, mantenendo la licenza originaria.

La licenza copyleft più comune è la GPL (General Public Licence: una formulazione completa in italiano all'indirizzo [www.softwarelibero.it/gnudoc/gpl.it.txt](http://www.softwarelibero.it/gnudoc/gpl.it.txt)), che è anche la licenza più utilizzata nel software libero, ed è usata per il kernel Linux ed una grande quantità di altri programmi. Questa licenza consente a tutti di copiare, vendere, distribuire il software da lei coperto, ma pone due condizioni: la prima è che se si distribuisce solo codice eseguibile deve essere anche messo a disposizione (ad eventuale richiesta) il codice sorgente, la seconda è che se si modifica un programma GPL (usando quindi codice coperto da questa licenza), anche il codice derivato dovrà mantenere la stessa licenza. Questo è alla base del principio "ereditario" del software coperto da licenza GPL: è sempre possibile incorporare pezzi di programmi GPL all'interno del nostro software, ma i programmi risultanti dovranno essere coperti dalla stessa licenza, risultando così in una progressiva espansione del numero di programmi liberi disponibili.

Esempi di programmi GPL in campo geografico sono GRASS, QGIS e PostGIS. Alcuni programmi, ed in particolare le librerie (programmi non direttamente usati dall'utente, ma da altri programmi che se ne servono per effettuare operazioni di fondo) sono invece coperti dalla Lesser General Public License

(LGPL), che consente una stretta correlazione (linking) con programmi non liberi.

L'esempio più comune di licenza non copyleft è la BSD (Berkeley Software Distribution), utilizzata per l'omonimo sistema operativo creato dall'Università di Berkeley. In questo caso la licenza richiede semplicemente che si dia credito all'autore, e che se ne riporti la presenza; resta possibile creare versioni modificate di codice BSD rilasciate come software proprietario. La licenza BSD (o simili, quali MIT, X11, ecc.) è tutt'ora molto usata, in particolare negli USA per progetti finanziati da fondi pubblici, i cui risultati, essendo lo sviluppo finanziato dallo stato con le tasse dei cittadini, devono essere messi a disposizione di tutti. La si usa anche in tutte le implementazioni di riferimento di algoritmi e codici, che si vuole abbiano la massima diffusione anche in campo proprietario. In campo geografico, UMN Mapserver e GDAL/OGR sono coperti da licenze di tipo BSD.

Il problema della licenza BSD è che non protegge da forme di sfruttamento in cui si utilizza il codice messo a disposizione di tutti, ma non si restituisce niente alla comunità di utenti e sviluppatori. La GPL invece richiede che le modifiche vengano rilasciate con la stessa licenza, e probabilmente proprio per questo è la licenza più diffusa. Uno sviluppatore che usa la GPL sa infatti che il suo codice potrà anche essere copiato e rivenduto da altri, ma tutte le correzioni ed i miglioramenti dovranno essere rilasciati con la stessa licenza, per cui alla fine anche lui ne potrà trarre beneficio. Esistono inoltre molte altre licenze libere, più o meno riconducibili a quelle precedenti; perché siano effettivamente certificate come Open Source, devono però rispettare alcune condizioni minime ([www.opensource.org/docs/definition.php](http://www.opensource.org/docs/definition.php)), che sostanzialmente equivalgono alle quattro libertà citate.

Da un punto di vista pratico, quindi, i programmi Open Source possono consentire a chi li usa (sia il semplice utente finale, sia lo sviluppatore, sia l'azienda) attività ben diverse, a seconda della licenza da cui sono tutelati: un programma di pubblico dominio può essere usato, copiato, ed incorporato in qualunque altro tipo di software, sia libero che proprietario, senza alcuna restrizione; la licenza BSD (o simili) consente essenzialmente le stesse possibilità, mentre incorporando codice GPL si è tenuti a non rilasciare il programma risultante se non con licenza identica. È quindi possibile, ad esempio, "privatizzare" UMN Mapserver, senza rilasciarne il codice sorgente, mentre non si può commercializzare un GIS basato su GRASS senza rilasciarne anche il codice sorgente. Di conseguenza, combinando codice GPL e BSD, il risultato dovrà essere distribuito con licenza GPL, mentre combinare codice BSD e proprietario dà generalmente origine a software proprietario. È in generale lesivo della licenza GPL la combinazione di codice GPL con codice proprietario, fatto salvo

l'uso personale; è questo il motivo, ad esempio, per cui non è possibile redistribuire GRASS compilato con il supporto a DWG tramite le librerie (non libere, né aperte, a dispetto del nome) OpenDWG. Bisogna notare però come non si sia obbligati alla redistribuzione delle modifiche fatte al codice: è perfettamente legittimo non redistribuire quanto da noi modificato; la licenza GPL impone unicamente la redistribuzione del codice sorgente unitamente al codice eseguibile, qualora questo venga redistribuito.

#### COME FUNZIONA UN SOFTWARE LIBERO ED OPEN SOURCE: QUANTUMGIS

Un programma libero sviluppato in modo cooperativo si appoggia su una serie di infrastrutture, quasi sempre basate su Internet. Un buon esempio è QuantumGIS, che mette a disposizione di sviluppatori ed utenti:

- un portale, con informazioni sul programma e le sue componenti, link per il download del programma e di dati di esempio, forum, mappa degli utenti (a cui ci si può aggiungere autonomamente online), ecc.: <http://qgis.org>;
- un sito collaborativo (wiki, basato sulla tecnologia della famosa wikipedia) per la condivisione delle idee e il coordinamento del lavoro: <http://wiki.qgis.org/qgiswiki>;
- un sito per la verifica e segnalazione di bugs e richieste di miglioramento: <https://svn.qgis.org/trac/report>; in modo inusuale per chi è abituato al software proprietario, in cui si cerca di nascondere i malfunzionamenti, qui è possibile verificare cosa non funziona, segnalare ulteriori bugs, e verificarne la soluzione;
- un sito per lo sviluppo collettivo (subversion, svn): <http://svn.qgis.org/WebSVN/>; qui tutto lo sviluppo avviene "alla luce del Web": ogni riga di codice che gli sviluppatori (abilitati dal responsabile del progetto) inseriscono nel codice è analizzabile da tutti; da qui ognuno (con le competenze necessarie) può anche scaricare, in tempo reale, il software nell'attuale fase di sviluppo, il che è utile per avere l'ultima versione una volta che un bug, per noi particolarmente fastidioso, sia stato sistemato;
- una mailing list per gli utenti, in cui si può richiedere ed offrire aiuto, segnalare eventi e problemi, ed in generale influenzare lo sviluppo e la "vita" del programma;
- una mailing list per gli sviluppatori, dove si discute degli aspetti tecnici dello sviluppo;
- un canale di chat, in cui le interazioni fra utenti e sviluppatori sono più rapide ed immediate.

In questo modo, tutti possono verificare non solo il prodotto finito, ma anche il funzionamento interno del programma, e fare scelte più consapevoli ed avvedute fra i vari software disponibili.

#### LUOGHI COMUNI (MA FALSI)

##### I GIS liberi sono difficili, solo per esperti

Questo è vero solo per alcuni programmi, dove le maggiori possibilità di personalizzazione possono spaventare un neofita. Un desktop GIS come QuantumGIS non è sostanzialmente diverso dagli equivalenti proprietari. I programmi lato server sono a volte anche molto più facili degli equivalenti proprietari.

##### I GIS liberi sono inaffidabili, non adatti al lavoro vero

Grandi aziende ed amministrazioni pubbliche basano servizi mission-critical sul software libero. Non esistono dimostrazioni sistematiche di una minore affidabilità dei GIS liberi rispetto a quelli proprietari, e in molti casi è dimostrabile il contrario.

##### I GIS liberi non hanno assistenza, formazione e documentazione

Esistono un buon numero di ditte e professionisti, sia a livello internazionale che nazionale, che offrono supporto professionale (inclusa la formazione e la produzione ed adattamento della documentazione). Attualmente il loro numero è inferiore a quello delle ditte che danno supporto ai software proprietari, ma questo deriva dal fatto che questi ultimi sono per il momento più diffusi.

##### I GIS liberi non offrono garanzie

Tutte le licenze (soprattutto quelle proprietarie) escludono le responsabilità per danni derivati dall'uso del software. Inoltre non vi è alcuna garanzia di corretto funzionamento. In questo aspetto il software proprietario è assolutamente equivalente al software libero. Dato però i risparmi sui costi di licenza, è di solito possibile garantirsi un'assistenza e una formazione più efficaci, ed adeguate garanzie di continuità del servizio.

##### La migrazione da un GIS proprietario a uno libero è difficile e costosa

Grazie ad un buon numero di standard, sia fra i formati dati (ad esempio shapefiles, GeoTIFF, ecc.) sia fra i servizi (si vedano gli standard elaborati da Open Geospatial Consortium, condivisi da software liberi e proprietari), adeguati piani di migrazione possono essere elaborati e realizzati, in continuità di servizio, con un risparmio netto di risorse.

##### La manutenzione è molto costosa

Sebbene il progresso sia incessante, e i cicli di rilascio del software libero siano molto rapidi, non è indispensabile, per avere un sistema in perfetta efficienza, procedere a continui aggiornamenti. Di conseguenza, la manutenzione può essere pianificata e realizzata in modo non sostanzialmente differente da quanto accade con il software proprietario, e a costi di solito comparabili.

## Libero e Open Source

### *I malfunzionamenti (bug) non vengono sistemati, o se sì dopo tanto tempo*

Ci sono molte evidenze del contrario. In molti casi un bug può essere riparato dalla comunità, in modo totalmente gratuito, in poche ore. In ogni caso ci si può garantire la pronta risoluzione, e la distribuzione di una versione aggiornata, tramite un opportuno supporto commerciale.

### *Manca la mia funzione preferita, quella più importante per il mio lavoro*

In generale i GIS liberi sono molto ricchi di funzionalità (ma anche fra quelli proprietari mancano funzioni ampiamente disponibili fra quelli liberi). Nel caso ne manchi una davvero importante, c'è da tener presente che i costi per lo sviluppo di una nuova funzione sono di solito sorprendentemente bassi, grazie al riutilizzo di componenti già disponibili e testati (principio di base del software Open Source).

### *Il costo del software è irrilevante per i GIS, perché la maggior parte viene speso per i dati*

I dati sono una componente fondamentale dei GIS, ed il loro costo è sicuramente molto elevato. Ma il software ha un peso non trascurabile, perché:

- nel costo dei dati è incluso implicitamente anche l'utilizzo di software spesso molto costoso;
- i dati GIS vengono utilizzati anche da molti altri soggetti (ad esempio professionisti indipendenti, consulenti, università ed enti di ricerca, studenti, associazioni, ecc.); quando si include anche il costo del software per tutti questi soggetti, le percentuali cambiano in modo sostanziale.

### *Il software libero è gratuito*

Benché spesso il software libero sia scaricabile in modo gratuito da Internet, il software libero non impone la gratuità, libertà e gratuità sono due aspetti ortogonali. Ci sono molti programmatori e aziende che vendono software libero, ed in particolare i servizi associati:

- installazione;
- configurazione;
- formazione;
- assistenza;
- Quality Assurance.

### *Il software libero è privo di copyright*

È falso: la grande maggioranza di software libero è protetto da una licenza.

### *Il contrario di libero è commerciale*

Sbagliato, il contrario di software libero è proprietario. La commercializzazione del software non dipende dalla licenza utilizzata: ci sono molte aziende che vendono software libero o offrono servizi basati su software libero.

### **ALCUNI LINKS UTILI**

- Free Software Foundation:  
<http://fsfeurope.org/index.it.html>
- AsSoLi (Associazione per il Software Libero):  
[www.softwarelibero.it](http://www.softwarelibero.it)
- Open Source Initiative:  
[www.opensource.org](http://www.opensource.org)
- Domande frequenti sulla licenza GPL:  
[www.gnu.org/licenses/gpl-faq.it.html](http://www.gnu.org/licenses/gpl-faq.it.html);

E, naturalmente, wikipedia: <http://it.wikipedia.org>.

### [ autori ]

**Paolo Cavallini**

Faunalia

Piazza Garibaldi, 5 - Pontedera (PI)

[www.faunalia.it](http://www.faunalia.it)

**Simone Piccardi**

Truelite

Via Monferrato, 6 - 50142 Firenze

[www.truelite.it](http://www.truelite.it)

### *Il software gratuito è libero*

Niente di più falso; il software gratuito quando rilasciato nella sola forma binaria e con una licenza proprietaria, non garantisce nessuna delle libertà che contraddistinguono invece il software libero.