

Open source in biblioteca: how to?

Giovanna Frigimelica^{*}, Andrea Marchitelli^{}**

^{*}Associazione Italiana Biblioteche, Roma;

^{**}CILEA, Roma

Abstract

Sempre maggiore interesse destano in tutti i settori, anche in Italia, le possibili applicazioni informatiche open source, in alternativa ai software proprietari. È importante, dunque, far chiarezza per evitare errori che possono seriamente compromettere le funzionalità del servizio e generare spese impreviste. Questo breve articolo si pone l'unico obiettivo di individuare alcuni elementi ai quali prestare attenzione nel momento in cui si voglia intraprendere un'analisi costi-benefici riguardo all'adozione di uno o più applicativi open source per le biblioteche e i servizi di editoria elettronica.

Open source applications become more and more widespread against proprietary systems in every field. It is important to avoid mistakes that can compromise services and increase costs. This short contribution points to some elements that need attention when a cost-benefit analysis is undertaken before the choice of open source applications for libraries or e-publishing services.

Keywords: ILS, Open source software, free software, biblioteche.

“La libertà non è star sopra un albero / non è neanche il volo di un moscone / la libertà non è uno spazio libero / libertà è partecipazione”

G. Gaber

Le libertà fondamentali

Il free software [1], movimento da cui poi è derivato quello di open source, discende dal concetto di libertà, non di prezzo. Un software, per essere definito “libero”, deve ammettere contemporaneamente quattro libertà fondamentali:

- Libertà di eseguire il programma per qualsiasi scopo (libertà 0).
- Libertà di studiare come funziona il programma e modificarlo perché faccia qualsiasi cosa voglia l'utente (libertà 1). L'accesso al codice è una precondizione necessaria.
- Libertà di copiare il programma in modo da aiutare il prossimo (libertà 2).
- Libertà di migliorare il programma e di distribuirne pubblicamente i miglioramenti (e qualsiasi versione modificata, in generale), in modo tale che tutta la comunità ne tragga beneficio (libertà 3).

A metà degli anni Novanta, una costola del movimento dà vita all'Open Source Iniziative, che modifica e precisa i termini della questione,

indicando le caratteristiche minime che deve avere una licenza tale da far definire “open source” il software che accompagna. Esse sono:

1. libera redistribuzione del software;
2. inclusione del codice sorgente;
3. deve permettere modifiche e prodotti derivati;
4. integrità del codice sorgente originale;
5. non deve discriminare persone o gruppi;
6. non deve impedire di far uso del programma in un ambito specifico;
7. i diritti relativi al programma devono applicarsi a tutte le redistribuzioni;
8. non deve essere specifica ad un prodotto;
9. non deve porre restrizioni su altro software distribuito insieme;
10. deve essere neutrale rispetto alle tecnologie.

A che gioco giochiamo?

Questa la prima domanda che occorre porsi. Qual è lo scopo del progetto? Una volta identificati con chiarezza l'obiettivo, possibilmente con uno studio sulle aspettative a breve e lungo termine, una volta individuati i vari attori del progetto, è necessario fare un serio *benchmarking*, un confronto, tra le caratteristiche dei software disponibili. Questa

analisi, oggi, non può escludere a priori i software OS: lo suggerisce il buon senso, oltre a indicarlo una direttiva specifica emanata nel 2003 dal Governo italiano.

Ma quanto mi costi (davvero)? Il *total cost of ownership*.

“Libertà di parola”, non “birra gratis” è il motto dello sviluppo open source. Un mito senz'altro da sfatare, infatti, vuole che scegliere open source significhi necessariamente risparmio. Se un discorso del genere, in linea di massima, può essere vero, in molti casi, nel medio-lungo periodo, quasi mai questa eventualità si verifica nelle fasi iniziali di un progetto OS.

Gli unici costi che effettivamente vengono azzerati sono quelli relativi alle licenze: quelle dei singoli applicativi, ma anche quelle dei software di base (es. Linux vs. Microsoft per il sistema operativo, Postgres vs. Oracle per il database ecc.). Altre spese invece rimangono; è importante, proprio per questo, capire quali siano queste voci di spesa e quanto incidano su un progetto. Imprescindibili le voci legate all'hosting del server, sia che lo si mantenga “in casa” o che ci si affidi a un provider di servizi, come i costi legati alla formazione e all'addestramento, al momento del passaggio ad un nuovo sistema. Inoltre, andrà considerata l'eventualità di installare nuove *release* del software, con conseguente impegno in ore di un informatico esperto, o di impiegare risorse per sviluppi e personalizzazioni.

Le personalizzazioni possono essere una voce di spesa importante: la libertà di fare ciò di cui si ha effettivo bisogno va rapportata al costo delle risorse necessario per apportare e mantenere le modifiche al codice.

In tutti questi casi, tuttavia, un *benchmarking* ben fatto dovrà considerare i cd. “costi interni”, cioè anche il complesso di ore, spesso ingente, spese da personale interno della struttura per mantenere il software, sia esso open source o proprietario: informatici per l'installazione e l'aggiornamento, bibliotecari per la configurazione e i test ecc. [2].

Sviluppo, supporto e assistenza

Quello dell'OS è uno sviluppo guidato dalla comunità. Ciò vuol dire che, per la terza libertà fondamentale, qualunque utente ha la possibilità di modificare il codice, rilasciando poi pubblicamente le modifiche introdotte. Diversi software OS hanno comunità di decine di utenti, con diversi livelli di formalizzazione, che vanno dal modello seguito da DSpace, il cui

sviluppo viene regolato da una fondazione e da un gruppo selezionato di sviluppatori che verificano le modifiche al codice proposte e come ricomprenderle nelle successive versioni ufficiali del software [3], a modelli più elastici e meno formalizzati, come quello della comunità di Koha (di cui si parla in questo stesso numero). Esistono naturalmente anche software OS che non sono seguiti da una vera e propria comunità, progetti fioriti e presto abbandonati, progetti mai arrivati neppure al rilascio di una vera e propria prima versione. In quest'ultimo caso, di certo, bisognerà diffidare, a meno che non si disponga di risorse tali da poter provvedere in proprio al recupero e al sostegno del progetto.

Supporto e assistenza agli utenti di questi prodotti, siano essi sviluppatori interessati ad apportare modifiche al codice o i veri e propri utenti finali, vengono infatti forniti dalle community. Così, è necessario prevedere la necessità che per avere assistenza si debba conoscere, almeno a livello base, la lingua inglese, per poter usufruire della documentazione (se non disponibile in italiano) e per poter interloquire, nelle varie mailing list generalmente disponibili allo scopo, con i membri della comunità. Un'alternativa possibile è quella di rivolgersi a un fornitore esterno (per esempio CILEA, o altre società che operano nell'ambito dell'*information technology*); in tal caso andrà calcolata una specifica voce di costo, che tuttavia verrà compensata dal minore impegno richiesto alle risorse interne. Anche in questo caso, andrà fatto un vero *benchmarking* delle diverse soluzioni OS e proprietarie, considerato che queste ultime possono avere costi di gestione e assistenza piuttosto gravosi.

Personalizzazione e indipendenza

La Direttiva 19 Dicembre 2003, “Sviluppo ed utilizzazione dei programmi informatici da parte delle pubbliche amministrazioni”, meglio nota come “Direttiva Stanca” prevede (art. 4): «Le pubbliche amministrazioni, nella predisposizione o nell'acquisizione dei programmi informatici, privilegiano le soluzioni che presentino le seguenti caratteristiche: [...] soluzioni informatiche che, in assenza di specifiche ragioni contrarie, rendano i sistemi informatici non dipendenti da un unico fornitore o da un'unica tecnologia proprietaria; la dipendenza è valutata tenendo conto dell'intera soluzione».

Questo è, in effetti, un elemento fondamentale e decisivo per scegliere un'applicazione OS

rispetto a una proprietaria (una volta appurato che entrambe rispondono alle necessità del progetto di automazione): la possibilità di non essere più dipendenti da un unico fornitore. Così, ad esempio, le personalizzazioni si possono ottenere, nei modi delineati sopra, anche se non rientrano nei piani di sviluppo del fornitore. Ciascun utente ha dunque la possibilità di trovare risposta a bisogni specifici: siamo tutti diversi, le soluzioni non possono essere le medesime per tutti.

Il valore dei formati aperti

La libertà di scelta, però, spesso è condizionata dal passato e dalla disponibilità dei dati creati con altre applicazioni: in quale formato sono disponibili? In quale è possibile ottenerle? Vengono recuperati facilmente tutti i dati o, per esempio, solo quelli bibliografici, mentre quelli gestionali no, o molto difficilmente? Scegliere un software open source può significare anche essere garantiti non solo sugli utilizzi presenti, ma anche sul futuro.

Open source, in conclusione, è una scelta di libertà, quella cantata da Gaber, fatta di persone che collaborano assieme, sviluppando software che diventano sempre migliori grazie alla partecipazione di un gran numero di tecnici e altri esperti. Nel momento in cui si opta per un software open source in un progetto di automazione, si compie dunque una scelta di libertà, partecipazione, scambio, comunità, ma anche autonomia: valori importanti, e senza vincoli.

Bibliografia

- [1] Per free software e open source
URL:
<http://www.fsf.org/licensing/essays/free-sw.html> <http://www.opensource.org/docs/osd>.
Per una breve introduzione al tema e specifici approfondimenti sull'OS in biblioteca, v. G. Frigimelica, "La diffusione di software Open Source per la gestione di biblioteche in Italia". *Biblioteche oggi*, 2009, vol. 27, n. 6, pp. 37-43.
URL: <http://eprints.rclis.org/17095/>
- [2] R. Lugg, R. Fischer, *The Real Cost of ILS Ownership*. White Paper Presented at IUG, April 2003.
URL:
http://www.ebookmap.net/pdfs/ILS_Ownership_final.pdf
- [3] Dspace Contribution guideline,
URL:
<http://wiki.dspace.org/index.php/ContributionGuidelines>.