

Software libero e diritti digitali

Uno dei tratti tipici della MED è la stretta dipendenza tra gli elementi tecnici e quelli più propriamente educativi. Alcuni degli approcci più caratteristici della MED, il learning by doing, l'apprendimento informale e tra pari, la dinamica progettuale volta alla produzione multimediale, rendono molto difficile, se non impossibile, discriminare tra gli aspetti strettamente tecnico strumentali (software, strumenti, prodotti) e quelli etico-pedagogici (acquisizione di competenze, empowerment, consapevolezza).

Questo è particolarmente vero nel caso particolare che prendiamo in esame, ossia la relazione tra la speciale "famiglia" di strumenti digitali che rientra sotto l'etichetta di "software libero" e uno degli obiettivi della Media education: l'affermazione della cittadinanza digitale.



Il rapporto tra Software libero e cittadinanza digitale è infatti profondo e strutturale, tanto che solo a fini di chiarezza esplicativa cercheremo di separare l'aspetto tecnico e quello etico-teorico, tenendo sempre ben presente però che il software libero è molto più che semplice software così come la cittadinanza digitale è un principio tutt'altro che astratto e teorico, non potendo prescindere da importanti presupposti tecnici.

I termini ci aiutano ad individuare un primo importante indizio di questa inscindibilità. Parlare di "software libero" significa affermare che, anche nella scelta degli strumenti, "la libertà è importante". Così come non parliamo solo di diritti ma di come tradurre nel mondo digitale il diritto di cittadinanza, ovvero le competenze e la consapevolezza necessarie per agire pienamente nella dimensione del cittadino.

Chiarezza dei termini

La terminologia tecnico-informatica risulta ancora imprecisa e, soprattutto nell'uso comune, il termine software libero viene spesso usato impropriamente, come sinonimo di altre famiglie di programmi (software gratuito o open-source).

Per una maggiore chiarezza utilizzeremo la formula anglosassone, che identifica il software libero con l'acronimo F.L.O.S.S.: Free Libre and Open Source Software.

Vediamo uno per uno questi termini.

SOFTWARE: evidentemente stiamo parlando di programmi informatici, ovvero di una sequenza di informazioni ed istruzioni, scritte in un particolare linguaggio di programmazione. Per comodità identificheremo 3 grandi sottogruppi di programmi: i sistemi operativi (che coordinano le parti di un dispositivo informatico e permettono l'interazione con l'utente); gli applicativi (che permettono la fruizione-creazione-modifica di prodotti digitali: testi, immagini, suoni) e i servizi web (che permettono la comunicazione tra macchine diverse, cioè la "rete", permettendo di coordinare la richiesta-offerta-scambio di informazioni e prodotti in formato digitale).

Vale la pena sottolineare due aspetti su cui solitamente non ci si sofferma:

- il software è "digitale", ovvero "numerico". Ogni prodotto digitale, sia esso software o opera compita, è una traduzione numerica di una "entità reale". Possiamo dire che ogni

sviluppatori, perchè ne facciano l'uso che ritengono.

Specifiche e vantaggi del FLOSS

Combinando assieme gli elementi caratteristici del FLOSS possiamo individuarne alcune importanti specifiche ed elementi di "preferibilità" rispetto alle alternative (software gratuito, software open source o software proprietario).

Un primo elemento fondamentale è la "community", ovvero il bacino di individui che utilizza-sviluppa-distribuisce il software. La community è al tempo stesso il motore del software, perchè è dalle esigenze di un determinato pubblico che nascono gli impulsi allo sviluppo del software ed alle sue successive modifiche, la sua "catena di trasmissione", perchè attraverso la community il software viene distribuito, commentato, spiegato e portato a piena funzionalità; ed infine è anche "l'officina di diagnostica": la community modifica, controlla e "monitora" il software, nelle sue varie versioni.

E' bene sottolineare che la community è composta da individui con competenze ed esigenze molto differenti: utenti di vario livello e sviluppatori più o meno esperti. Questa differenza porterebbe ad una forte "polarizzazione" tra utenti attivi e passivi se il software non fosse gratuito, libero e open source. Solo in questo modo la grande varietà di esigenze e competenze si traduce in condivisione

Un secondo aspetto degno di riflessione è la "durata", del software libero. Essendo gratuito il software libero è sempre potenzialmente aggiornato: non c'è ragione perchè un utente non installi l'ultima versione di un sistema operativo o di un programma (ad eccezione della sua affidabilità, che, ancora una volta, viene testata dalla community). D'altra parte, essendo modificabile e restituibile, non esiste mai una reale "versione definitiva" di un software libero. Ipotizzando che un gruppo di sviluppatori metta a disposizione la versione 2.0 di un programma, in breve tempo alcuni utenti particolarmente competenti potrebbero modificarlo (per esempio tradurre l'interfaccia, adattarlo a pc obsoleti, aggiungerne o toglierne alcune funzionalità) e rendere disponibile una versione 2.1, o una "derivata non ufficiale". Ci muoviamo nell'orizzonte del "beta continuo": ogni prodotto digitale è passibile di trasformazioni ed ogni punto fermo è in realtà una tappa in un percorso di modifiche potenzialmente ininterrotto.

Ma la caratteristica forse più interessante del FLOSS, così come lo abbiamo descritto, è che questi elementi nel loro complesso non identificano tanto uno "strumento", quanto un "metodo" di produzione. Un metodo di produzione, tra l'altro, con un intrinseco valore educativo e portatore di alcuni valori che (eccoci!) sono fondamentali nella definizione e nell'esercizio del diritto di cittadinanza.

Trasparenza, condivisione, competenza (intesa come "saper fare") e partecipazione sono componenti imprescindibili del software libero, senza le quali, semplicemente, non esisterebbe.

Oltre a queste considerazioni di carattere generale, il software libero ha anche alcune peculiarità tecniche che, soprattutto in determinati contesti (in particolare nel contesto educativo) lo rendono uno strumento particolarmente efficace.

Gratuità: ovvio vantaggio, ma significa anche poter provare un programma prima di adottarlo, poterlo condividere in più contesti e con più soggetti senza alcun vincolo legale e finanziario;

Personalizzazione: posso modificare (o chiedere/cercare modifiche) il programma secondo le mie particolari esigenze: ad esempio la localizzazione;

Portabilità: posso usare il programma in contesti e su dispositivi differenti;

sicurezza: la trasparenza del codice normalmente garantisce una maggiore stabilità e sicurezza dei programmi.

Esempi e risorse

Sistemi operativi:

<http://www.debian.org/>

<http://www.ubuntu-it.org/>

<http://www.dynebolic.org/>

<http://distrowatch.com/>

Applicazioni:

<http://www.openoffice.org/it/>

<http://www.gimp.org/>

<http://www.blender.org/>

<http://audacity.sourceforge.net/?lang=it>

<http://inkscape.org/it/>

<http://mixxx.org/>

<http://sourceforge.net/directory/home-education/>

Servizi web:

<http://www.apache.org/>

<http://www.wampserver.com/en/>

CMS:

<https://drupal.org/node/1611502>

<https://moodle.org/?lang=it>

<http://www.opensourcecms.com/>