

Inauguro così la sezione **Libri** (o categoria? mi confondo sempre... maremma giummla!).

Contenitore virtuale dove saranno raccolti giudizi ed impressioni su alcuni libri (ma no?! e vi dirò di più: libri che ho addirittura letto! uau!). Questo qua accanto l'ho visto l'anno scorso in una libreria a Ferrara e mi incuriosì, anche se non lo comprai. Me lo sono ritrovato davanti pochi giorni fa e stavolta l'ho acquistato. E' un libro filosofico, nel quale però non mancano gli esempi di situazioni concrete. Direi **metafisico** (e metà pratico).

Titolo: **Perché il software fa schifo** (Why software sucks).

Autore: **David Platt**.

Editore: **Mondadori informatica**.

Prezzo: **18,00 €** (mamma mia! ma sarà possibile che un libro costi così?! lasciamo stare...).

Codice ISBN: **978-88-6114-044-8**

Dato che mi piace rovinare le sorprese e che il titolo lascia intendere poco :) ecco la sintesi del contenuto: p**erché il software fa schifo?** Colpa dei **programmatori**. Incredibile vero?

Secondo l'autore tutto è dovuto al fatto che la "cerchia" dei programmatori, persone di solito con intelligenza sopra la media (li chiama **smanettoni** o **secchioni** e ci si mette anche lui. Credo che in inglese sia **giic**, scritto ***ge ek***, che significa geco), coscienti di questo fatto ed orgogliosi di esserlo [1], raramente si calano nei panni dei futuri utenti, partendo dal presupposto (sbagliato) che **essendo anche loro utenti di programmi** (se non altro del sistema operativo su cui girerà il programma e dei vari kit di sviluppo, che sono software a loro volta) **sanno già cosa voglia un utente da un programma**.

A questo si aggiunge la tendenza al '**miglioramento continuo**' di quello che programmano, cosa in sé positiva ma travisata: spesso utilizzano male il tempo a disposizione per aggiungere al programma funzionalità non richieste, trascurando di fargli fare **cose semplici** in **modo semplice**, intuitivo e in modo facilmente comprensibile dall'utente [2].

Poi ci si mettono la disposizione poco intuitiva dei menu, l'inutilità degli assistenti stupidi (tipo Clippy di Microsoft Word), le 'piccolezze' che per chi parte da zero nel mondo del computer possono creare malumore e disorientamento: ad esempio **domande inutili** e **fuorvianti**, conferme non necessarie e così via.

Altra pecca: il programmatore progetta la struttura ed il funzionamento del proprio software. Un utente non dovrebbe mai vedere come queste due cose sono fatte. L'importante è che il programma funzioni e il non dover essere costretto a pensare con la testa del programmatore.

Quando si guida un'auto non si pensa da meccanico. Quando si prende un'aspirina non si pensa da medico. Quando si usa un programma non si dovrebbe pensare da programmatore.

L'autore poi parla del web, di quello che offrono i siti oggi, di cosa si salvi, cosa proprio non va e come si potrebbe migliorare...

Ci sono situazioni effettivamente difficili da risolvere, come **coniugare sicurezza e riservatezza** dei dati personali (o **privasi**, privacy) in cui bisogna capire innanzitutto quali siano i dati sensibili e poi quanto lo siano. Infine valutare le situazioni e trovare il giusto compromesso per non superare il ***livello delle seccature*** che un utente è disposto a sopportare in cambio di qualche sicurezza in più.

L'ultima causa, ma non meno importante, **dei programmi scritti male sono** (siamo) **gli utenti**, che non si fanno sentire abbastanza: spesso è difficile fare reclami, sia telefonici che per posta elettronica, dato che fra segreterie automatiche e contatti nascosti bene fra le pagine dei siti, far arrivare le proprie rimostranze a qualche azienda per un servizio offerto scadente è un vero atto di volontà. Però anche quando non è così sono pochi gli individui che oltre a gridare al software che fa schifo poi scrivono un rapportino sul perché faccia schifo, cosa manca, cosa vorrebbero...

Il libro è simpatico, scritto in un linguaggio comprensibile da chiunque. Non tratta di cose supreflue, inutili o esagerate: non va a cercare il pelo nell'uovo, ma tratta gli argomenti in modo esauriente. Ci sono molti rimandi a siti reltivi agli argomenti trattati e a quello del libro in questione. L'autore spesso è divertente, come nelle battute, nelle situazioni buffe descritte o in alcune barzellette. Ad esempio la definizione di privacy:

"*La visione che viene data dagli articoli si concentra sempre sull'idea che la privacy (qualsiasi cosa significhi) è una Cosa Buona, e quindi rivelare qualsiasi informazione a qualcun altro è una Cosa Cattiva, perché Loro (chiunque siano) stanno guardando proprio te, e muoiono dalla voglia di rivelare i tuoi segreti (ad esempio quando usi il filo interdentale mentre guardi Mr. Bean) in prima pagina su USA Today* "

--*--

1. tanto che dimostrare che il loro prodotto è "stupido" (e quindi anche chi lo ha programmato) sembra sia la cosa che li fa infuriare di più.

2. un po' la filosofia di unix e del KISS (Keep It Simple, Stupid) ossi di mantenere il programma semplice e stupido ma che funzioni bene.

3. Gli inglesi dicono prìvasi e non pràivasi... did iù nó it (lo sapevi)?