

NICOLA FURINI

**PUBBLICARE CONTENUTI IN
INTERNET CON LA TECNOLOGIA
“*OPEN SOURCE*”**

Versione 1.0
E - T E A M S

Prefazione di Stefano Petrone

Al bene più prezioso che mi sia mai stato donato: i miei figli.

“Linux si trova oggi ad avere milioni di utenti, migliaia di sviluppatori e un mercato in espansione. È presente in sistemi integrati, è usato per il controllo di dispositivi robotizzati e ha volato a bordo dello Shuttle. Sarebbe bello poter dire che avevo immaginato quello che sarebbe successo, che era tutto parte di un mio piano per la conquista del mondo: ma, onestamente, sono stato colto di sorpresa.”
(Linus Torvalds)

ATTENZIONE!

Questo libro è rilasciato sotto “GNU *free documentation license*” (<http://www.gnu.org/licenses/licenses.html>), chiunque può modificarlo e migliorarlo rispettandone la sola proprietà intellettuale dell'autore e citando in modo visibile nella prima, seconda, penultima e ultima pagina:

"Versione originale di Nicola Furini (nick@graces.it), www.e-teams.info, questo libro, in tutte le sue versioni per volontà dell'autore, può essere riprodotto anche integralmente senza violare nessuna legge in quanto questo libro è rilasciato sotto GNU free Documentation License (<http://www.gnu.org/licenses/licenses.html>)"

In particolare, questo libro potrà essere:

- ➔ modificato parzialmente o integralmente creando manuali per aziende, enti o persone che si occupano di formazione;
- ➔ distribuito sia in forma originale che in forma modificata, in formato elettronico come in formato cartaceo;
- ➔ i sorgenti modificabili del libro sono disponibili in formato “sxi” (OpenOffice) sul sito www.e-teams.info. Openoffice è una suite gratuita scaricabile da www.openoffice.org

Perché questo libro.....	2
Prefazione	3
1 – Da dove cominciamo?	4
2 - Un mondo “Open Source”	5
2.1 – Il regno degli hacker	5
2.2 – Software “libero”	6
2.2.1 - Il “manifesto”	7
2.2.2 – La definizione di <i>Open Source</i>	7
2.2.3 – La licenza “GPL” (<i>General Public License</i>).....	9
2.3 - I punti di forza	9
2.4 - I punti deboli.....	10
2.5 – Un nuovo modello di <i>business</i>	11
2.5.1 – Le applicazioni aziendali	11
2.5.2 – Il supporto tecnico.....	12
2.5.3 - Il caso Zope	12
2.6 – Il mercato	13
2.6.1 – Analisi della domanda.....	13
2.6.2 – Analisi dell’offerta	15
3 – La gestione dei contenuti in Internet.....	18
3.1 – Caratteristiche	18
3.2 – <i>Content Management System</i>	19
3.2.1 - Tipologie.....	20
3.2.2 - Come funzionano.....	21
3.2.3 – Vantaggi.....	23
3.2.4 – Gli application Server	23
4 - Rassegna dei principali CMS <i>Open Source</i>	24
4.1 - APC ActionApps	25
4.2 – Ariadne	25
4.3 – E107.....	26
4.4 – Envolution.....	27
4.5 – E-Xoops	28
4.6 - eZpublish	29
4.7 – Fullxml.....	30
4.8 – Maxwebportal	31
4.9 – Metadot	32
4.10 – MySource.....	33
4.11 – Nucleus	34
4.12 - PHPNuke	34
4.13 - Plone.....	36
4.14 – PMachine	36
4.15 – SmartPublish.....	37
4.16 – Typo3.....	38
4.17 – Xaraya.....	39
4.18 – WebGui.....	40
4.19 - Zope.....	41
5 – CMS “quasi” <i>Open Source</i>	42
5.1 – Constructioner.....	42
5.2 – SiteSupra.....	43
5.3 – Siteworkspro	44
6 – Un osservatorio permanente	45
7 – Come scegliere il sistema giusto	46

Comparazione delle caratteristiche di alcuni CMS Open Source

Webliografia

Bibliografia

Perché questo libro

Se qualcuno dovesse mai chiedersi perché mi è venuto in mente di scrivere questo libro potrei rispondergli che, essenzialmente, l'ho fatto per offrire un contributo per quanto modesto, alla diffusione della conoscenza di una tecnologia che può cambiare il mondo rendendolo migliore.

Promuovere l'utilizzo delle tecnologie *Open Source* vuol dire infatti praticare una forma (per quanto raffinata) di consumo critico e responsabile; significa affermare il rifiuto dei monopoli e della sudditanza nei confronti delle grandi multinazionali che ci considerano degli utenti-merce. Ma vuol anche dire promuovere l'alfabetizzazione informatica, per una più equa distribuzione delle risorse (in termini di sapere, informazione, competenze e strumenti), contro un accesso riservato alla tecnologia, non alla portata di tutti.

In una scuola dove si fatica a pagare lo stipendio agli insegnanti, come purtroppo accade in tanti paesi in via di sviluppo, come si può attrezzare un'aula informatica con dei personal computer dotati di sistema operativo e di programmi applicativi commerciali, le cui licenze per singola postazione rappresentano magari una somma pari decine di volte lo stipendio medio di un insegnante? Vi sembra giusto? E vi sembra giusto, oltreché sensato, che la pubblica amministrazione di uno stato (uno qualunque, ma anche il mio, l'Italia) spenda milioni di Euro prelevati dalle tasche dei propri cittadini contribuenti per dotarsi di un sistema per la gestione degli uffici (c.d. *Office Automation*) pagando licenze commerciali per dei software di cui esistono dei perfetti sostituti utilizzabili liberamente senza dover pagare un solo Euro? E ancora, non vi sembra poco prudente per qualunque azienda legarsi ad un solo fornitore per la gestione del proprio sistema informativo?

Insomma, questo libro vuole fornire degli utili, si spera, elementi per valutare la possibilità di abbandonare la strada del *software* commerciale per una nuova via, lastricata da uno spirito di condivisione e di solidarietà in grado di portare dei concreti vantaggi alla "comunità globale".

N.F.

Prefazione

Questo *e-book* è un approfondimento del mondo dei *Content Management Systems*, collegato al primo *e-Book* di nostra produzione “E-Learning”, sul quale trovate una introduzione a tutto il mondo dell’*E-Learning* ed un approfondimento sugli *e-book* e gli strumenti di comunicazione digitale.

Il nostro obiettivo è chiarire il mondo della formazione ed editoria online, ancora troppo sottovalutata rispetto a quello dell’editoria stampata, soprattutto dal punto di vista delle “potenzialità di creazione individuali” che l’editoria online offre, rispetto all’editoria cartacea.

Troverete gli aggiornamenti di questo e gli altri *e-book* complementari, sul sito del nostro gruppo, <http://www.e-teams.info>. Sono graditissimi, da parte dei nostri lettori, commenti e contributi per il miglioramento della nostra collana di *e-book*.

Stefano Petrone

1 – Da dove cominciamo?

La maggiorparte degli utilizzatori di programmi per pc è abituata a considerare i *software* come delle “scatole nere”. Il software è infatti spesso un oggetto “proprietario”, chiuso, di cui difficilmente si può conoscere il codice sorgente (ovvero come è fatto dentro, il suo “*DNA*”). Se al supermercato trovassimo sugli scaffali dei prodotti di cui sull’etichetta si evidenziano pregi e qualità senza rivelare però con quali ingredienti sono fatti, sicuramente ci verrebbe almeno qualche dubbio. Se poi questi prodotti dovessero anche costare caro, a maggior ragione prima di decidere l’acquisto ci interesseremmo circa le modalità di produzione e confezionamento. Nel caso dei software commerciali, ci comportiamo invece in modo poco razionale perché acquistiamo prodotti di cui non vengono resi noti gli “ingredienti” e qualunque modifica alla versione originale (attività considerata peraltro illecita essendo questi prodotti protetti da leggi sul copyright o da brevetti industriali) viene resa impossibile.

Agli albori dell’informatica la situazione era profondamente diversa: i codici sorgenti dei programmi erano aperti (quindi disponibili a tutti) affinché ogni utente potesse intervenire migliorandone le prestazioni e personalizzare le funzioni dei singoli prodotti. I software nascevano da comunità di programmatori che ispiravano la propria attività secondo un modello cooperativo, in grado di coinvolgere e portare il meglio della professionalità, dell’ingegno e della creatività di ciascuno.

Per fortuna, oggi qualcosa sta cambiando. Prodotti come il sistema operativo *Linux*, il *web server Apache*, linguaggi di programmazione come *Perl* e *Python* sono la dimostrazione che è possibile produrre software di qualità estremamente raffinata, affidabile e robusta grazie al contributo di centinaia (a volte migliaia) di programmatori sparsi in tutto il mondo. Così come all’interno della comunità scientifica le scoperte di ogni singolo scienziato sono messe a disposizione di tutti, allo stesso modo nelle comunità *Open Source* il codice scritto è libero, aperto, nessun programmatore possiede le singole righe di codice e neppure ne rivendica la paternità cercando di trarne un profitto. Tutti invece contribuiscono a migliorarne la qualità..

Queste esperienze rappresentano un modello completamente alternativo di produrre, gestire e distribuire il software. E alternativi sono anche i modelli di business che da questi hanno preso piede; sono numerose le aziende nate al fine di distribuire e garantire la manutenzione di soluzioni software *Open Source*.

2 - Un mondo “Open Source”

Scambiarsi codice, correggerlo, migliorarlo, sono stati dei semplici e scontati strumenti per consentire il rapido sviluppo dell’*Open Source* nei laboratori dei pionieri dell’informatica. E molto probabilmente i protagonisti di questa rivoluzione non ne hanno nemmeno avuta la ben che minima consapevolezza. Quando il software uscì però dai laboratori per essere utilizzato per fini commerciali, molti ricercatori vennero assorbiti dalla nascente industria e si perse rapidamente il concetto di software come “bene scientifico”. Ma non per tutti fu così. Negli anni ’80, un ricercatore del laboratorio di intelligenza artificiale del MIT, Richard Stallman, decise per deliberata scelta personale di non cedere alle lusinghe economiche del software commerciale e di iniziare una “crociata” a difesa del diritto al software liberamente disponibile. Altri programmatori aderirono al progetto di Stallman e cominciarono presto a nascere i primi prodotti. I risultati furono subito incoraggianti e ancora molti altri programmatori si aggregarono, grazie anche all’avvento di Internet. Nascono in questo periodo le prime comunità *hacker* (vedi oltre).

Nei primi anni ’90 un eclettico studente finlandese, Linus Torvalds, inizia a lavorare sulla costruzione di un sistema operativo derivato da UNIX che prenderà il nome di “Linux”. Il progetto cresce velocemente oltre ogni aspettativa, grazie soprattutto ai contributi di scrittura del codice e di testing offerti dalle comunità *hacker*.

Il modello del software libero attira investitori che potrebbero fornire mezzi per affrontare altri progetti; la fermezza di Stallman nel difendere lo spirito “puro” dell’idea scoraggia però ogni approccio di tipo imprenditoriale.

Arriviamo così nel 1997 per compiere un nuovo passo in avanti. Nasce la “Open Source Initiative”, con l’obiettivo dichiarato di avvicinare la cultura del software proprietario (sviluppato da società grandi e piccole che hanno fatto del software la propria fonte di reddito) a quella del software libero.

2.1 – Il regno degli hacker

Il termine *hacker* viene coniato alla fine degli anni '50, nel tempio della tecnologia che è il “Mit” (Massachusetts Institute of Technology), quando i pc erano ancora un sogno in là a venire e le intuizioni dei giovani ricercatori costituivano un patrimonio da mettere in comune. *Hacker* era considerato colui che sapeva trovare una soluzione brillante a un problema informatico; nel tempo si è fatto poi

riferimento a colui che mette in pratica una serie di valori quali la cooperazione, la libertà di rielaborare e migliorare i prodotti intellettuali altrui. È grazie a questo modo di lavorare che sono nati il primo personal computer, il web, Linux (il sistema operativo alternativo a Windows). Gli *hacker* rivendicano con vigore la libertà di diffondere le conoscenze a tutti i livelli, a favore di chiunque e per il beneficio di tutti, in primo luogo del sapere informatico.

Un giovane finlandese, Pekka Himanen, prendendo ispirazione dal famoso saggio “L'etica protestante e lo spirito del capitalismo” del sociologo Max Weber, ha scritto un libro sugli *hacker*. Anzi, si potrebbe quasi dire sull'*hacker* che c'è in ognuno di noi. L'etica protestante è quella di chi mette il lavoro come dovere al centro della vita, principio su cui è modellata la società capitalistica e consumistica. In funzione del lavoro e del denaro sono regolati i rapporti sociali e il tempo delle nostre esistenze. All'etica protestante fa da contrappunto, secondo Himanen, l'etica *hacker*, che mette al centro invece la passione per il lavoro, l'intrattenimento e il divertimento, con il gusto di sapersi sfidare e di condividere il sapere con gli altri. Chiunque può quindi essere un *hacker* nella vita, anche senza occuparsi di informatica: tutto sta nell'atteggiamento che si assume di fronte alle cose che si fanno.

È importante distinguere gli *hacker* dai *cracker*. Si tratta di due comunità ben distinte, anche se i giornalisti, purtroppo, utilizzano spesso i due termini come sinonimi. La differenza tra i due gruppi è abissale: gli *hacker* creano, i *cracker* distruggono. Anche se entrambi sono in grado di individuare i punti deboli dei sistemi informatici, per gli *hacker* è fondamentale non danneggiare il sistema visitato, i *cracker* fanno del danno la loro principale attività. Per gli *hacker* la conoscenza è un patrimonio da condividere. I *cracker* si comportano diversamente: si scambiano i codici dei programmi “killer” ma sono gelosi riguardo le conoscenze alla base del programma. Questo spiega anche perché la comunità *cracker* segna il passo rispetto alla comunità *hacker*.

2.2 – Software “libero”

Un programma è software libero (detto anche *freeware*) se l'utente ha libertà di:

- ⇒ eseguire il programma per qualsiasi scopo;
- ⇒ modificare il programma secondo i propri bisogni (perché questa libertà abbia effetto in pratica è necessario garantire l'accesso al codice sorgente del programma);
- ⇒ distribuire copie del programma, gratuitamente o dietro compenso;

⇒ di distribuire versioni modificate del programma, così che la comunità possa fruire dei miglioramenti apportati.

Il termine “*free*” del software può essere male interpretato perché “*free*” vuol dire anche gratuito, oltre che libero, ma non ha niente a che vedere con il prezzo del software. Si tratta di libertà, e quindi è perfettamente lecito utilizzare questo software anche per motivi commerciali. Una parte della comunità “Free Software” introdusse nel 1998 il termine “Open Source” con un duplice obiettivo:

- escludere il termine “free” e l’ambiguità fra “libero” e “gratuito”
- promuovere il movimento rinnovandone il “marchio” e costruendo una reputazione che attirasse gli interessi delle aziende

Nel 1998 venne fondata la “Open Source Initiative” (OSI) ed Eric Raymond ne divenne il rappresentante più conosciuto. “Software libero” (*free software*) e “sorgente aperto” (Open Source) descrivono in linea di massima la stessa categoria di software, pur dicendo cose differenti sui valori e sulle libertà associate. Diverse filosofie hanno anche portato a diversi modelli di licenze (GPL, LGPL, MPL, BSD, etc), ma le differenze non incidono sui punti di forza del modello.

2.2.1 - Il “manifesto”

Un documento scritto e pubblicato da Eric S.Raymond nel 1998, con il titolo “La cattedrale ed il bazaar”, viene unanimemente considerato il “manifesto” del software *Open Source*. In questa opera, l’autore analizza un progetto open source di successo, “fetchmail”, utilizzandolo appositamente come esperienza per la verifica di alcune affascinanti teorie sullo sviluppo del software suggerite dalla storia di Linux. Le tesi argomentate mettono a confronto due diversi stili di sviluppo, il modello “cattedrale” (specifico nel mondo commerciale) opposto al modello “bazaar” (caratterizzante invece il mondo *Open Source*).

2.2.2 – La definizione di *Open Source*

Open Source non significa solamente accesso al codice sorgente. I termini di distribuzione di un programma *Open Source* devono rispettare i criteri seguenti:

1. Libera redistribuzione: la licenza non può limitare nessuna delle parti nella vendita o nella fornitura di software come componente di una distribuzione di software aggregati, contenente programmi provenienti da fonti diverse. La

licenza non può richiedere il pagamento di una royalty o di diritti per tale rivendita.

2. Codice sorgente: il programma deve includere il codice sorgente, e deve consentire la distribuzione sia sotto forma di codice sorgente sia in forma compilata. Nei casi in cui un prodotto non venga distribuito con il codice sorgente, deve esserci la possibilità, ben pubblicata, di scaricare il codice sorgente via Internet senza costi aggiuntivi. Il codice sorgente deve essere la forma privilegiata in cui il programmatore modificherà il programma. Codice sorgente deliberatamente nascosto non è ammesso. Forme mediate, come l'output di un preprocessore non sono ammesse.
3. Prodotti derivati: la licenza deve consentire l'attuazione di modifiche e di prodotti derivati, consentendo inoltre la loro distribuzione sotto gli stessi termini di licenza del software originale.
4. Integrità del codice sorgente dell'autore: la licenza può imporre limitazioni sulla distribuzione del codice sorgente in forma modificata solamente se la licenza consente la distribuzione di file "patch" insieme al codice sorgente con lo scopo di modificare il programma durante l'esecuzione del build. La licenza deve consentire esplicitamente la distribuzione di software realizzato a partire dal codice sorgente modificato. La licenza può richiedere che i prodotti derivati portino un nome o un numero di versione diverso dal software originale.
5. Nessuna discriminazione verso singoli o gruppi: la licenza non deve porre discriminazioni verso qualsiasi persona o gruppo di persone.
6. Nessuna discriminazione verso campi di applicazione: la licenza non deve porre limitazioni sull'uso del programma in un particolare campo di applicazione. Per esempio, non può impedire l'uso del programma in una azienda o per la ricerca genetica.
7. Distribuzione della licenza: i diritti allegati al programma devono applicarsi a tutti coloro a cui viene ridistribuito il programma, senza la necessità di applicare una licenza supplementare per queste parti.
8. La licenza non deve essere specifica per un prodotto: i diritti allegati al programma non devono dipendere dal fatto che il programma faccia parte di una distribuzione particolare. Se il programma viene estratto da tale distribuzione e usato o distribuito nei termini della licenza del programma, tutte le parti a cui il programma viene ridistribuito devono avere gli stessi diritti garantiti in occasione della distribuzione originale del software.

9. La licenza non deve contaminare gli altri programmi: la licenza non deve porre limitazioni su altro software che venga distribuito insieme con il software in licenza. Per esempio, la licenza non deve asserire che tutti gli altri programmi distribuiti sullo stesso supporto devono essere software *open source*.
10. Conformità della licenza e della certificazione: qualsiasi programma che faccia uso di licenze certificate come conformi alla *Open Source Definition* può utilizzare il marchio registrato *Open Source*, e il codice sorgente può essere dichiarato esplicitamente di pubblico dominio. Nessun altro programma o licenza è certificato per fare uso del marchio registrato *Open Source*.

La “GNU GPL”, la “LGPL”, la licenza “BSD”, la licenza “X Consortium”, la “Artistic”, la “MPL” e la “QPL” sono esempi di licenze da considerare conformi alla *Open Source Definition*. Per sottoporre a certificazione una licenza, occorre scrivere a certification@opensource.org, mentre eventuali abusi del marchio *Open Source* possono essere segnalati a mark-misuse@opensource.org.

2.2.3 – La licenza “GPL” (General Public License)

La “GPL” (*General Public License*) è la più diffusa licenza di distribuzione del software *Open Source*. In estrema sintesi specifica che il software può essere copiato, modificato e ridistribuito liberamente, ma sempre assieme al codice sorgente. Il grande vantaggio dei software distribuiti secondo queste condizioni è che se qualcuno desidera migliorarli, può farlo, rendendo poi disponibile alla comunità la nuova versione. Questa pratica conduce ad avere programmi eccellenti scritti da tanti sviluppatori differenti.

Una panoramica completa, per la gran parte tradotta in lingua italiana, sulle diverse tipologie di licenze *Open Source* è disponibile sul sito “GNU”, agli indirizzi:

- ⇒ <http://www.gnu.org/licenses/license-list.it.html>
- ⇒ <http://www.gnu.org/philosophy/categories.it.html>

2.3 - I punti di forza

Viene da chiedersi per quale strano motivo le aziende dovrebbero legare le sorti dei propri sistemi informativi a delle soluzioni software che sono sviluppate con il contributo prevalente di programmatori che agiscono spontaneamente,

normalmente su base puramente volontaria.

Generalmente, il costo di un progetto software si compone di tre parti: acquisizione della licenza, installazione e personalizzazione dei programmi, costi interni di adeguamenti organizzativi. Il costo di un progetto basato su software *Open Source*, libero da licenze commerciali a pagamento, è quindi complessivamente pari a due terzi di un progetto basato su software commerciale. Quindi i vantaggi per un'azienda sono evidentemente prima di tutto economici. Ma non solo. Ricordiamo i punti di forza di ogni progetto *Open Source*:

- ⇒ sensibile riduzione dei tempi di sviluppo grazie al coinvolgimento di una vasta comunità di sviluppatori (spesso anche diverse centinaia di esperti competenti e motivati che partecipano al progetto, sviluppo, test e documentazione dei progetti). Le aziende che producono software commerciale devono invece basarsi su risorse molto più limitate;
- ⇒ riutilizzo più facile del codice sorgente e condivisione delle esperienze;
- ⇒ libertà dell'utente dalla dipendenza nei confronti del fornitore;
- ⇒ possibilità di verificare con la massima trasparenza che nel software non siano nascosti sistemi di spionaggio informatico che violano la privacy degli utilizzatori;
- ⇒ garanzia di possibilità di personalizzazione massima del software;
- ⇒ portabilità del software, ricompilabile su sistemi operativi diversi e con la traduzione nelle più svariate lingue;
- ⇒ continuità (nel caso di interruzione del progetto da parte degli sviluppatori originali, altri possono subentrare);
- ⇒ manutenzione e l'assistenza al software ottenibile velocemente, via internet, contattando le varie comunità virtuali interessate (diversamente dal software commerciale la cui vita è invece legata all'andamento e alle scelte commerciali del produttore).

2.4 - I punti deboli

Non sarebbe forse onesto parlare solo dei punti di forza del software *Open Source*, trascurando di accennare anche ai possibili punti deboli. Aldilà degli attacchi operati dai tanti detrattori di un nuovo modello di intendere l'organizzazione civile ed economica di una società, in effetti, anche il software *Open Source* può soffrire di punti deboli. Non intendiamo dilungarci in tediose disquisizioni al riguardo, basti solo tenere presente che un software proprietario può fornire, in talune circostanze, dei concreti vantaggi rispetto al software libero. Stanno proprio in questo i punti deboli (se così proprio dobbiamo chiamarli) dell'*Open Source*.

2.5 – Un nuovo modello di *business*

Ma chi investe oggi nei progetti *Open Source*? Il software libero è generalmente reperibile su internet, ma spesso necessita di conoscenze tecniche per la sua installazione, manutenzione e gestione. L'utente non esperto è disposto a pagare una cifra ragionevole per disporre di un aiuto che ne semplifichi l'utilizzo. Le fonti di guadagno per le società informatiche quindi non mancano di certo e sono principalmente le seguenti:

- ⇒ supporto tecnico, installazione, configurazione e soluzione di problemi;
- ⇒ personalizzazione, adattamento del software alla specifiche necessità del cliente;
- ⇒ formazione: istruzione di personale tecnico per il supporto, istruzione degli utenti all'utilizzo;
- ⇒ consulenza: per la scelta del prodotto, per l'integrazione con altri sistemi già presenti;
- ⇒ documentazione: manuali, libri, riviste, corsi pubblici, etc.

Alcune aziende forniscono software *Open Source* "pacchettizzato", cioè completo di manualistica e di strumenti per la sua installazione e configurazione guidata. Altre società producono e vendono materiale di supporto al software, quali manuali, libri, riviste. L'attività può essere svolta direttamente dall'azienda che ha sviluppato il progetto oppure da altre aziende che possiedono una buona conoscenza del prodotto. Altre aziende produttrici di prodotti hardware distribuiscono in formato libero il software di base necessario al loro funzionamento.

2.5.1 – Le applicazioni aziendali

La maturità raggiunta da molti prodotti *Open Source* ha convinto anche le aziende più blasonate a considerare la possibilità del loro utilizzo. Molti dei più importanti prodotti commerciali nel settore gestionale (vedi SAP) e dei database (Oracle e DB2 in testa) sono disponibili su Linux e si integrano con strumenti ed applicazioni *Open Source*. Molti dati a confronto sull'utilizzo dei prodotti *Open Source* sono disponibili al sito: http://www.dwheeler.com/oss_fs_why.html

L'ampiezza della diffusione del software *Open Source* è difficile da quantificare perché normalmente lo si può scaricare senza troppe formalità direttamente da Internet. *Source Forge*, il servizio più utilizzato per lo sviluppo e distribuzione di software *Open Source*, contiene oltre 36.000 progetti (marzo 2002) di ogni genere e categoria.

2.5.2 – Il supporto tecnico

Bene, vinte le perplessità iniziali viene da chiederci: ma è affidabile questo software? E il supporto tecnico? Come la mettiamo con la sicurezza?

Anche in questo caso, possiamo sgombrare il campo da ogni ulteriore dubbio:

- ⇒ Linux ha vinto già qualche anno fa il premio come prodotto dotato del miglior supporto tecnico. Quando gli utenti di un software commerciale si trovano in difficoltà possono trovare supporto solo presso i centri di assistenza legati alla ditta produttrice e sovente pagando salati canoni di manutenzione. Chi utilizza software *Open Source* ha a disposizione (in Internet) una quantità enorme di risorse libere e gratuite (manuali, database, documenti multimediali) e di comunità di utenti cui sottoporre il proprio problema e da cui farsi consigliare. Uno dei principali benefici e vantaggi del software *Open Source* è proprio di non essere più vincolati ad un singolo fornitore.
- ⇒ Il software *Open Source* è in genere più affidabile del software proprietario (è controllato e verificato da ampie comunità di sviluppatori, il codice sorgente è pubblico e quindi esposto a severe verifiche, non dipende da politiche commerciali che concentrano gli investimenti sui prodotti più redditizi a scapito di quelli meno diffusi).
- ⇒ I *crackers* (da non confondere con dagli *hackers*) attaccano indifferentemente i sistemi commerciali come quelli *Open Source*. A fronte del rilevamento di un problema il numero di esperti in grado di mobilitarsi per correre ai ripari è enormemente superiore nel caso del software *Open Source*. Offrire il codice sorgente assieme al programma eseguibile significa anche portare a conoscenza l'utilizzatore di cosa fa quel software e di come lo fa. Molte statistiche dimostrano comunque la maggiore sicurezza di Linux e Apache (server per Internet) rispetto i prodotti commerciali concorrenti.

2.5.3 - Il caso Zope

Nel 1996 l'azienda statunitense Digital Creation realizza uno strumento di sviluppo per il web chiamato Bobo, che distribuisce come prodotto *Open Source*. L'anno seguente realizza Principia, una potente e stabile piattaforma di sviluppo per applicazioni web, basata su Bobo, che però distribuisce come prodotto commerciale proprietario. Nonostante l'indiscutibile qualità, Principia non riesce

a trovare idonea collocazione sul mercato già saturo degli *Application Server*. Nel 1998 la Digital Creation decide la svolta strategica: riunire le funzionalità di Bobo e Principia in un unico prodotto da distribuire con licenza *Open Source*. Era nato Zope.

A pochi anni di distanza il grande successo del prodotto e le migliaia di sviluppatori in tutto il mondo dimostrano come la scelta dalla Digital Creation sia stata vincente. I vantaggi ottenuti dalla Digital Creation si possono così riassumere:

- ⇒ numero forte e crescente di utilizzatori del software;
- ⇒ nascita di una grande comunità di sviluppatori (divenuta nel tempo un potente strumento di “marketing virale”), con conseguente garanzia di continuità di sviluppo, mantenimento e correzione degli errori del software a costo praticamente nullo;
- ⇒ aumento del *know how* per l’azienda;
- ⇒ concentrazione degli investimenti in attività di consulenza, personalizzazione e formazione;
- ⇒ commercializzazione di applicazioni evolute che consentono un elevato ritorno economico.

2.6 – Il mercato

Dopo aver esaminato un pò di storia sulla quale poggiano le origini del software *Open Source*, risulta ora sicuramente interessante verificare l’impatto sul mercato che è scaturito da queste esperienze. La situazione è infatti in grande fermento, e a dimostrazione che le prospettive di mercato e la validità dell’offerta sono ormai rilevanti, sono sempre più frequenti le notizie di nuove aziende fornitrici, di nuovi prodotti e soluzioni, di nuovi importanti clienti conquistati dalla proposta *Open Source*.

2.6.1 – Analisi della domanda

Gli utilizzatori di software *Open Source* spaziano ormai veramente in tutti i settori e interessano realtà di dimensioni molto differenti tra loro, dal professionista alla multinazionale. A titolo puramente esemplificativo, si riportano alcune esperienze particolarmente significative raggruppate per settore produttivo.

⇒ **Grandi aziende informatiche:**

- ➔ IBM: fra le grandi aziende informatiche, IBM si qualifica oggi come uno dei maggiori sostenitori del software *Open Source*, cui ha destinato e

continua a destinare notevoli investimenti. IBM è fra i promotori del progetto “Eclipse”, un ambiente di sviluppo *Open Source* per la realizzazione di applicazioni web che si porrà in netta concorrenza con “VisualStudio.Net” della Microsoft. Sono certamente note, anche al grande pubblico, le attività di marketing di IBM a favore di Linux (in particolare con i simpatici spot televisivi sulla sicurezza e stabilità delle reti aziendali).

- ➔ SAP: è leader di mercato nel settore delle soluzioni software ERP (Enterprise Resource Planning), sistemi per la gestione integrata dell’azienda. SAP partecipa da tempo alla comunità *Open Source*: già nel marzo 1999 SAP è divenuto il primo produttore di software gestionale interaziendale a realizzare soluzioni e-business su Linux. Nel 2000 ha inoltre reso disponibile il proprio sistema di gestione database “SAP DB” alla comunità *Open Source* nell’ambito della “GNU General Public License”.
 - ➔ Sun: contribuisce attivamente a numerosi e importanti progetti *Open Source*. Ricordiamo OpenOffice (la versione *Open Source* di StarOffice), GNOME (interfaccia grafica per Linux), Mozilla (browser web), Apache (web server). Sun offre supporto al sistema operativo Linux, utilizzabile con molte delle proprie macchine. E’ in preparazione un nuovo “*PC client*” con software *Open Source* preinstallato: Linux, Mozilla, Open Office, Evolution, Gnome.
 - ➔ Oracle: molto attiva sul fronte *Open Source* tanto che già nel 1998 annunciò il rilascio della release 8 del proprio database su Linux. Più di recente Oracle ha presentato un’iniziativa per l’estensione della compatibilità con Linux all’intera linea di prodotti Oracle 9i.
 - ➔ HP: promuove progetti per la standardizzazione delle distribuzioni di Linux.
 - ➔ Compaq: partecipa a numerosi progetti *Open Source* per software funzionante sul proprio hardware.
 - ➔ Computer Associates: sponsor dell’Open Source Development Lab, una iniziativa per rendere Linux il principale sistema operativo per lo sviluppo di applicazioni di e-business.
- ⇒ **Grandi industrie:**
- ➔ Siemens: gestisce il proprio dipartimento “Risorse Umane” con applicazioni basate su MySAP e Linux.
 - ➔ Altri imprese tedesche che impiegano Linux: Deutsche Bahn (ferrovie nazionali), Deutsche Post (poste), Daymler Chrysler, Bayer, Mercedes-Benz.
 - ➔ In Svezia hanno adottato Linux l’Ikea e la SAAB.
 - ➔ Kodak, Boeing, Xerox, CISCO, Amazon utilizzano Linux.

- ➔ Credite Suisse First Boston: la banca di investimenti internazionali ha reso noto di aver scelto Linux per trattare con riservatezza e affidabilità le operazioni di compravendita titoli.
 - ➔ Molti operatori finanziari di Wall Street utilizzano Linux per gran parte dei propri sistemi informatici.
- ⇒ **Governi nazionali:**
- ➔ Italia: nel giugno 2002 il ministro per l’Innovazione e le tecnologie Lucio Stanca ha predisposto il documento “Linee guida del Governo per lo sviluppo della società dell’informazione nella legislatura”. Di particolare rilevanza il riconoscimento e il supporto fornito al software *Open Source*.
 - ➔ Francia: il governo francese si è attivato per diffondere Linux nell’apparato della Pubblica amministrazione. E’ stato riconosciuto alla MandrakeSoft, distributrice di Linux Mandrake, un contratto per la fornitura di Linux a ministeri ed agenzie dello stato.
 - ➔ Germania: il governo tedesco ha concluso un accordo di partnership con IBM e SuSE per la diffusione nell’amministrazione pubblica dei sistemi *Open Source*, allo scopo di ridurre la dipendenza dalle tecnologie proprietarie.
 - ➔ Spagna: è stata realizzata la connessione in rete di 200 server e di oltre 3000 postazioni su PC impiegando Linux. È prevista la progressiva connessione di punti nevralgici dell’amministrazione dello stato (per esempio le stazioni di polizia).
 - ➔ Repubblica Popolare Cinese: il governo di Pechino promuove lo sviluppo di Linux per limitare la dipendenza del paese da sistemi operativi chiusi, sui quali non può avere alcun controllo.

2.6.2 – Analisi dell’offerta

Sono ormai disponibili prodotti *Open Source* nei più svariati settori del software, con ambizioni differenti e pensati per funzionare sulle più svariate piattaforme:

⇒ **sistemi operativi:**

- ➔ Verso la fine degli anni 70 ebbe inizio lo sviluppo del sistema operativo UNIX nei laboratori Bell della AT&T. Il nuovo sistema, creato a fini di ricerca, venne poi ceduto alle università unitamente al suo codice sorgente. In seguito, la licenza d’uso di UNIX subì delle restrizioni che stimolarono la nascita di versioni “aperte”. E fu solo nel 1991 che Linus Torvalds, all’epoca studente all’università di Helsinki, creò la primissima versione di un nuovo sistema che avrebbe preso il nome di “**Linux**”. Nei successivi anni di vita, grazie al lavoro di centinaia di sviluppatori in tutto il mondo, Linux è cresciuto enormemente sotto il profilo della qualità e della quota

di mercato conquistata. Linux può essere utilizzato su svariate piattaforme hardware (INTEL, Sun, Mac, PowerPC, etc) ed è ormai supportato da molti dei principali produttori di software. È dotato di caratteristiche tecniche di notevole qualità e fornisce prestazioni, a parità di hardware, solitamente superiori ai sistemi commerciali direttamente concorrenti. Linux è disponibile in varie distribuzioni, tra cui le più note sono Red Hat, SuSe, Mandrake, Caldera.

⇒ **servizi web e di posta elettronica:**

➔ **Apache:** *web server* stabile, affidabile, veloce e facile da utilizzare e configurare. È disponibile per piattaforme Linux, ma non solo, e secondo le statistiche dei web server più utilizzati su computer collegati a Internet (Netcraft Web Server Survey), Apache detiene di gran lunga la leadership di mercato.

➔ **Sendmail:** sistema di spedizione della posta elettronica più utilizzato al mondo (oltre la metà delle e-mail inviate nel mondo).

➔ **Qmail:** secondo sistema di posta per diffusione, ma è forte in crescita grazie alle sue caratteristiche di sicurezza. Nel 1997 era stato offerto un premio di 1000 dollari a chi avesse trovato un bug in Qmail. Inutile dire che il premio non è mai stato vinto (mentre i mille dollari sono stati donati alla Free Software Foundation). Hotmail, USA.net e Yahoo! usano Qmail come server SMTP.

⇒ **Database:**

➔ **MySQL:** database relazionale nato nel 1996, veloce, potente ed affidabile, con prestazioni ottimizzate per sostenere un alto carico di richieste. Offre le principali funzionalità fornite dal linguaggio SQL ed è particolarmente indicato come supporto per siti web dinamici con basi di dati anche molto estese. Fra gli utilizzatori figura nientemeno che la NASA.

➔ **Postgres:** è il più evoluto database *Open Source* oggi disponibile. Offre prestazioni di poco inferiori a MySQL nell'interrogazione dei dati, ma è dotato di funzionalità più estese che lo avvicinano ai più potenti database relazionali presenti sul mercato.

⇒ **application server** ⁽¹⁾:

⁽¹⁾ Un *application server* è un ambiente di sviluppo integrato, omogeneo e già completo di importanti servizi di base, in grado di: semplificare le attività di sviluppo, agevolare il riutilizzo del codice, fornire una architettura facilmente espandibile, gestire le funzioni specifiche di sicurezza tramite un ambiente unico di configurazione degli utenti e delle aree protette, offrire meccanismi di gestione delle versioni, *editing* in remoto, esportazione e salvataggio dei dati, connessione a database esterni.

- ➔ **Zope:** *application server Open Source* più diffuso sul mercato. È multiplatforma, orientato agli oggetti, utilizzabile completamente tramite un comune browser web, fornito di centinaia di moduli applicativi già pronti, gode del supporto di una comunità di migliaia di sviluppatori in tutto il mondo.
 - ➔ **Jboss:** *application server* che riscuote un crescente successo tra gli addetti del settore (oltre 40000 download al mese), sia per il suo costo nullo che per le prestazioni e caratteristiche offerte. È scritto con linguaggio Java e funziona su piattaforme Linux/Unix oppure Windows NT /2000. La comunità di sviluppatori di JBoss è al momento composta da più di mille persone e la disponibilità di forum e mailing list offre un continuo supporto tecnico. Semplice da installare richiede poche risorse hardware e supporta i principali database del mercato.
- ⇒ sistemi di **groupware** ⁽²⁾:
- ➔ **PHProject:** progetto tedesco che offre ottime funzionalità ed una buona interfaccia grafica per l'uso tramite browser web.
- ⇒ **software gestionale:**
- ➔ **Compiere:** la più popolare applicazione *Open Source* per le aziende oggi disponibile. Offre tutte le funzionalità per interagire in tempo reale, tramite internet, con clienti, fornitori e partner. L'applicazione è particolarmente adatta per i settori della distribuzione e dei servizi, e può anche essere utilizzata come interfaccia fra i processi aziendali operanti sul web e il sistema informativo tradizionale.
 - ➔ **Mosaico:** applicazione per le aziende messa recentemente a disposizione con licenza *Open Source* dalla Computerinside Srl, una *software house* italiana. Mosaico è un programma applicativo per la gestione operativa della piccola e media impresa, si fa apprezzare per la sua semplicità d'uso per l'immediatezza nella gestione documenti (fatture, bolle, ordini, ecc.).
- ⇒ **office automation:**
- ➔ **OpenOffice** è un progetto nato in casa Sun con lo scopo primario di sviluppare una *suite* per l'ufficio multilingue e multiplatforma, basata sullo standard aperto XML di salvataggio dei documenti. Tra gli strumenti a disposizione in OpenOffice.org: Writer, Calc, Impress, Draw, HTML Writer, Math. Uno dei punti di forza del software è quella di essere compatibile con Windows Microsoft Office 98/2000/XP, sia per l'apertura che il salvataggio dei *file*, e la compatibilità con i sistemi operativi Linux, Solaris, Apple.
- ⇒ **prodotti per la grafica:** Gimp.

⁽²⁾ Si tratta di sistemi di supporto a gruppi di utenti. Consentono di comunicare a bassissimo costo con colleghi e agenti ovunque essi si trovino, condividendo piani di lavoro, progetti, documenti e informazioni in genere. Possono essere integrati con il sistema informativo aziendale.

- ⇒ **linguaggi di programmazione e compilatori:** Perl e Python.
- ⇒ **strumenti per la gestione dei servizi di rete:** Samba.

3 – La gestione dei contenuti in Internet

Con l'avvento di Internet, tutti abbiamo iniziato a beneficiare degli innumerevoli servizi forniti dalla rete. La sua diffusione è partita dal settore della ricerca nell'ambito della comunità scientifica, ma poi l'interesse si è rapidamente esteso alle aziende, alle organizzazioni, alle associazioni non profit, agli enti della Pubblica amministrazione, ai privati cittadini. Sempre di più, la rete costituisce un supporto, spesso insostituibile, al nostro lavoro, alle nostre ricerche, al nostro svago, al nostro desiderio di tenerci informati e di comunicare con altre persone.

Insomma, Internet è diventato uno strumento che ci consente meglio di realizzare la nostra libertà. Una libertà che si costruisce sulla possibilità di comunicare con tante persone, in qualunque parte del mondo, di manifestare liberamente le proprie opinioni e il proprio pensiero, e tutto a un costo accessibile per chiunque. Proviamo soltanto a pensare come potremmo, oggi, fare a meno delle email (nonostante la maggiorparte degli attuali utilizzatori fino a non più di 5-6 anni or sono non ne conoscesse forse nemmeno l'esistenza, almeno in Italia). Ma pensiamo anche all'utilità straordinaria degli strumenti a supporto delle comunità virtuali (*newsgroup, mailing list, newsletter, faq, ecc.*) oppure all'efficacia di un sito Internet, all'interno del quale possiamo inserire qualunque tipo di oggetto multimediale che ci riguardi (testi, immagini, animazioni, filmati, suoni, musica) oppure delle applicazioni software che aiutano a svolgere determinati compiti.

Ed è proprio dei contenuti pubblicabili nei siti internet, e delle tecniche utilizzabili per semplificarne la gestione, di cui ci occuperemo nei capitoli seguenti

3.1 – Caratteristiche

Le principali caratteristiche dei sistemi per la gestione dei contenuti comprendono:

- separazione tra contenuti, definizione della struttura e relativa visualizzazione: la separazione logica di queste tre dimensioni produce notevoli vantaggi lungo tutto l'arco temporale di vita di un sito internet (il sito può essere ridisegnato, ristrutturato in modo da stravolgerne l'aspetto e la navigazione senza per questo dover modificare nemmeno una virgola dei contenuti preesistenti);

- produzione di contenuti facilitata: disponendo di un'interfaccia utente visuale (simile allo standard dei più comuni software di *office automation*) il personale che crea i contenuti ha la possibilità di scrivere testo, inserire immagini o altri elementi multimediali, definire la schedulazione temporale dei contenuti stessi, senza per questo dover possedere particolari competenze tecniche;
- utilizzo di modelli (*template*) grafici per la presentazione dei contenuti;
- personalizzazione dei criteri grafici di presentazione dei contenuti (temi);
- gestione di ruoli distinti di utenti per specifiche funzionalità di *workflow*;
- *database* per l'archiviazione di tutti gli elementi di testo e multimediali;
- recupero e integrazione dei contenuti provenienti da altri sistemi informatici;
- gestione di funzioni a supporto di comunità virtuali (*mailing list, forum, chat, newsletter, etc*);
- funzionalità di interrogazione e ricerca.

3.2 – Content Management System

Con la denominazione “**Content Management System**” (CMS) si fa comunemente riferimento ai sistemi utilizzati in Internet (o reti private, Intranet) per la gestione di contenuti (pagine, articoli, rubriche, notizie, faq, forum, schede prodotto, *banner* promozionali, etc). Quanto più un sito è composto da un elevato numero di pagine, tanto maggiore è l'esigenza di disporre di sistemi informatici semplici e intuitivi, per la manutenzione e la gestione dei relativi contenuti. Un buon CMS consente dunque ai creatori di contenuti di concentrarsi sul loro lavoro senza essere distratti da problematiche tecnologiche connesse all'ambiente operativo nel quale poi troveranno collocazione i contenuti stessi.

L'implementazione di portali e siti informativi richiede lo sviluppo di pagine dinamiche, generate da una sorta di “motore informatico” che aggrega i contenuti prelevandoli da appositi archivi (*database*) solo al momento in cui ne viene fatta richiesta. L'inserimento dei contenuti negli archivi viene effettuato dal *web content manager* (colui che redige e organizza i contenuti editoriali di un sito) attraverso un'interfaccia di gestione che consente di operare semplicemente e con rapidità. Grazie all'uso di interfacce “amichevoli”, un sistema di *Content Management* permette agli utenti autorizzati, indipendentemente dalla loro localizzazione geografica e dalle competenze informatiche, e secondo i poteri e i livelli attribuiti, di contribuire alla definizione, impaginazione e amministrazione dei contenuti del sito. Per la pubblicazione vera e propria dei contenuti (affinché

vengano resi disponibili al pubblico), viene normalmente utilizzato un sistema di controlli ed approvazioni (*workflow*) che prevede l'intervento di figure diverse:

- l'amministratore, che definisce gli autori e gli editori, assegnando loro le aree su cui possono lavorare;
- l'autore, che inserisce i contenuti nelle aree di propria competenza;
- l'editore, che approva, modifica, rifiuta e pubblica le pagine approvate.

3.2.1 - Tipologie

Si possono individuare quattro categorie generali nei quali classificare i CMS:

1. **framework**: si tratta generalmente di veri e propri ambienti di sviluppo e programmazione. Un *framework* contempla funzioni di base molto evolute come la gestione dei processi autorizzativi (*workflow*) e la creazione di *template* per la rappresentazione visuale dei contenuti. Questi sistemi sono normalmente piuttosto complessi e presuppongono delle attività di implementazione tali da richiedere la disponibilità di personale molto qualificato. Per tali motivi, sistemi di tale natura vengono utilizzati solo da grandi imprese. Il mercato del software commerciale offre una gamma veramente vastissima di soluzioni (solo per citare i sistemi più noti, ricordiamo Vignette, Broadvision, Documentum). Nel mondo *Open Source* citiamo Zope, leader indiscusso e capace di competere senza complessi di inferiorità con i citati costosissimi e blasonati concorrenti commerciali.
2. **sistemi basati su "pagina"**: si tratta di sistemi modellati in modo tale da separare i contenuti (testi, immagini, audio, video, archivi di qualunque genere), che vengono immagazzinati in un database, dalla relativa rappresentazione video. Tali sistemi presuppongono interventi di personalizzazione spinta per aderire alle specifiche esigenze del sito web. Si tratta di una famiglia abbastanza numerosa (come esempi tra i sistemi *Open Source* si citano: EzPublish, Typo3, Mambo).
3. **sistemi basati su modulo**: si tratta di sistemi che si focalizzano su particolari funzioni o tipi di contenuto. Un sistema basato su modulo include notizie, eventi e forum di discussione. I benefici ottenibili da tali sistemi consistono nella possibilità di ottenere un portale standard funzionante in tempi rapidi e con costi molto contenuti. Ovviamente, male si prestano per la gestione di contenuti non standard (necessitano di interventi di scrittura di codice). Si tratta di una famiglia molto numerosa (come esempi tra i sistemi *Open Source* si citano: PHPNuke, Postnuke, Envolution, Xoops e molti altri).

4. **sistemi con contenuti a oggetti**: si tratta di sistemi centrati sull'idea che un contenuto (l'oggetto) rappresenti un pezzo di informazione utilizzabile in modi diversi all'interno di un sito. Con questo approccio, diventa facile il riutilizzo dei contenuti per sezioni diverse di un sito o per applicazioni diverse (notizie, eventi). A questi sistemi piace lavorare come un database tradizionale, con ogni articolo trattato come un record di database. Si tratta di una famiglia meno numerosa delle precedenti (come esempi tra i sistemi *Open Source* si citano: ActionApps e WebGui).

3.2.2 - Come funzionano

I tradizionali sistemi per il *web* permettono la creazione di pagine HTML da differenti oggetti, attraverso un continuo scambio di informazioni tra chi gestisce i contenuti e chi, tramite programmazione, procede alla realizzazione tecnica delle pagine HTML che verranno successivamente pubblicate. Per questi motivi la gestione di siti classici è notevolmente lunga e onerosa. I sistemi di *Content Management* permettono invece di ridurre sensibilmente tempi e costi relativi alla pubblicazione in Internet: la pagina (nella sua struttura) viene definita una sola volta tramite opportuni modelli e alimentata continuamente nei contenuti da utenti che non necessariamente devono avere conoscenze tecniche specifiche. Il sistema provvede a gestire la sequenza di tutti gli stati dei contenuti dalla fase iniziale di creazione fino all'approvazione e pubblicazione *online*.

Di seguito, viene rappresentato lo schema tipico di un processo di creazione-pubblicazione di contenuti in Internet, in comparazione con lo schema di un processo gestito tramite un CMS.

a) Schema tipo del sistema “tradizionale”:

- ➔ il richiedente:
 - ✓ avvia la richiesta di pubblicazione di pagine Internet al gestore del sito *web*;
 - ✓ procede alla creazione dei contenuti e produce un documento;
 - ✓ invia il documento al gestore con le indicazioni sulle modalità di fruizione/visualizzazione.
- ➔ il gestore del sito *web*:
 - ✓ acquisisce il documento e le indicazioni fornite;
 - ✓ procede alla realizzazione delle pagine con la collaborazione dell'esperto di grafica;
 - ✓ invia al richiedente il documento per le verifiche sulla pubblicazione.

- ➔ il richiedente:
 - ✓ approva il documento o segnala le modifiche al gestore.
- ➔ il gestore del sito *web*:
 - ✓ procede alla pubblicazione *online* del documento, se approvato, altrimenti procede alle modifiche e rinvia il documento in approvazione.

b) Schema tipo con soluzione di *Content Management*:

- ➔ il richiedente:
 - ✓ seleziona un modello predefinito per la creazione del documento nell'ambito del sistema di *Content Management*;
 - ✓ inserisce i contenuti nel modello di CM;
 - ✓ il sistema di *Content Management* provvede automaticamente a generare il codice HTML;
 - ✓ visualizza il nuovo documento creato;
 - ✓ procede alla autorizzazione e pubblicazione del documento;
 - ✓ il sistema di CM provvede direttamente ad inserire il nuovo documento nel sito Web.
- ➔ il gestore del sito *web*:
 - ✓ definisce i ruoli da assegnare ad ogni richiedente; gestisce le relative autorizzazioni (amministratore, autore, editore); provvede alla gestione sistemistica del CM.



3.2.3 – Vantaggi

I vantaggi derivanti dall'utilizzo di un sistema di CM per gestire contenuti in Internet sono numerosi:

- ➔ la manutenzione viene svolta tipicamente tramite un normale web browser ed è decentralizzata; l'accesso può avvenire dovunque e in qualunque orario (vengono eliminati i colli di bottiglia legati alla disponibilità dei programmatori HTML);
- ➔ gli accessi sono protetti e riservati; ogni utente interessato alla creazione di contenuti viene abilitato a un particolare ruolo con permessi di accesso tali da impedire di modificare contenuti invece non autorizzati;
- ➔ grazie alla separazione dei contenuti dalla relativa visualizzazione, viene mantenuta l'integrità della visualizzazione grafica; i contenuti preparati da ciascun autore vengono pubblicati con lo stesso impianto grafico comune definito dall'amministratore generale del sito;
- ➔ i menu vengono generati automaticamente dal database dei contenuti e quindi si evita il rischio di esporre dei *dead links* (link verso pagine non esistenti);
- ➔ l'archiviazione dei contenuti in un database centralizzato consente di formattare i contenuti stessi in base allo specifico canale utilizzato dall'utente (web browser, WAP, PDA, web TV, stampa, etc);
- ➔ i servizi di supporto alle comunità virtuali (forum, sondaggi, negozi, ricerche, news) vengono sviluppati come moduli riutilizzabili;
- ➔ ogni giorno i contenuti possono essere modificati senza dover fare ricorso a programmatori o altri specialisti di internet;
- ➔ viene facilitato il contributo da parte di molti autori che collaborano alla manutenzione del sito;
- ➔ la pubblicazione dei contenuti può essere programmata temporalmente (consentendo la visualizzazione solo entro certi intervalli temporali).

Nonostante tutti i vantaggi e benefici di un buon CMS, bisogna fare attenzione alle lacune. Un CMS non rende migliori i contenuti "poveri", anche se fa risparmiare tempo per potersi concentrare nella creazione dei contenuti stessi, invece di sprecare tempo e denaro in inutili preliminari tecnici.

3.2.4 – Gli application Server

Quando si parla di CMS, occorre fare riferimento anche agli *Application Server*. Un *Application Server* è una piattaforma integrata di sviluppo e installazione software per gli ambienti Internet/Intranet e consente di realizzazione applicazioni *web* dinamiche e siti interattivi risolvendo problematiche di sicurezza, prestazioni, usabilità. L'utilizzo di un *Application Server* consente di rendere disponibili le

funzionalità richieste da un sistema di *Content Management* in modo semplice e naturale.

Un *Application Server* fornisce delle funzionalità di supporto predisposte per:

- ✓ definire e riutilizzare i *template* grafici da associare a ogni tipologia di oggetto;
- ✓ gestire gli utenti ed i rispettivi diritti di accesso alle aree e funzioni riservate;
- ✓ gestire gli archivi di immagini e documenti da cui attingere in fase di preparazione e impaginazione dei contenuti;
- ✓ disporre di un motore di ricerca integrato, in grado di indicizzare automaticamente testi e documenti e di consentire poi ricerche avanzate per parole chiave e operatori logici;
- ✓ la definizione di sistemi di *workflow* con cui organizzare le fasi di lavoro ed in particolare i processi di approvazione e pubblicazione automatica.

È difficile dunque tracciare una linea precisa di separazione fra strumenti di tipo *Application Server* e di tipo *Content Management*. Quasi sempre la prima categoria di prodotti offre funzionalità tipiche di un CMS e tanti prodotti ormai famosi come CMS svolgono anche funzioni di *Application Server*.

4 - Rassegna dei principali CMS Open Source

In questo capitolo sono state raccolte le schede di alcuni dei prodotti CMS Open Source più utilizzati. L'antologia che ne deriva non ha ovviamente la pretesa di essere esaustiva e il suo scopo è quello di fornire, a tutti quanti sono interessati alla gestione di contenuti in Internet, degli elementi utili per orientarsi nella valutazione dei tanti strumenti disponibili. Va precisato che non si può affermare in linea di principio che un CMS sia in assoluto il migliore: può esistere però un CMS più adatto di altri a rispondere a determinate esigenze.

Nelle schede presentate di seguito si è tenuto conto delle informazioni fornite dal sito web dello sviluppatore (o della comunità di supporto del progetto) e degli esiti dei test eseguiti personalmente dall'autore o da altri utenti pilota che hanno a loro volta messo a disposizione il proprio lavoro. In appendice, viene infine proposta una tabella di comparazione sintetica dei vari strumenti esaminati.

4.1 - APC ActionApps

ActionApps è un sistema creato dalla “Association for Progressive Communications” per gestire portali della realtà del mondo non-profit. Risulta particolarmente indicato per fare degli upgrade da siti statici esistenti verso un nuovo portale dinamico. Altra caratteristica importante riguarda la facilità di condividere contenuti tra siti differenti (la maggiorparte degli altri prodotti utilizza soltanto tecniche di condivisione delle “*headlines*” utilizzando la tecnica RSS).

Il pacchetto di installazione base di ActionApps fornisce funzioni per la gestione di notizie e articoli. Se servono altri tipi di contenuti, questi possono essere facilmente creati utilizzando un’apposita funzione di “*field editor*”.

Tecnologia:

Linguaggio di *scripting*: PHP

Database: MySQL

Sistemi operativi: Linux

Web server: Apache

Pregi: consente una facile integrazione dei contenuti di siti preesistenti nel nuovo portale. Fornisce delle funzioni per la condivisione di contenuti tra siti diversi.

Difetti: la configurazione del portale richiede una certa dose di esperienza. La personalizzazione grafica è difficile e complessa. La comunità di supporto non è così folta come quella di altri prodotti. Occorre per forza attivare il portale presso uno dei provider affiliati al progetto (in Italia non ce n’è ancora uno) e vale solo per organizzazione non profit.

Riferimenti:

❖ www.apc.org/actionapps: sito del progetto originale.

4.2 – Ariadne

Ariadne è una piattaforma applicativa e allo stesso tempo un CMS utilizzabili per lo sviluppo di applicazioni non eccessivamente complesse. Si tratta di una applicazione orientata agli oggetti che consente di separare i contenuti dal layout grafico. Possono essere utilizzati template HTML, XML, RTF, etc. Ariadne prevede anche un linguaggio di *scripting* proprietario (PINP), derivato da PHP, con il quale possono essere estese le funzionalità di base.

Sviluppato dalla ditta olandese Muze, che lo ha reso disponibile sotto la licenza GPL, il prodotto fornisce una buona interfaccia di gestione, con l'utilizzo di *wizard* (processi guidati per la generazione di oggetti), menu a cascata e un editor visuale per l'impaginazione dei contenuti. Sono disponibili numerose tipologie di oggetti predefiniti (archivio indirizzi, articoli, calendario, cartelle, file, gruppi, notizie, foto, etc).

Attraverso l'interfaccia web è possibile:

- creare nuovi oggetti;
- esportare/importare un database;
- editare testi e grafica con un apposito editor visuale;
- creare e modificare template;
- attribuire permessi di accesso agli utenti sui diversi oggetti

Tecnologia:

Linguaggio di *scripting*: PHP

Sistemi operativi: Linux, Windows

Database: MySQL, Postgres

Web server: Apache 1.3.x

Requisiti server: Apache con modulo PHP abilitato, MySQL server

Pregi: è un sistema completo, non eccessivamente complesso, adatto anche per grandi aziende.

Difetti: necessita di un pò di tempo per acquisire dimistichezza con alcune funzioni di amministrazione. Anche l'installazione e la configurazione richiedono l'intervento di personale piuttosto esperto.

Riferimenti:

❖ <http://ariadne.muze.nl/>: sito del progetto originale.

4.3 – E107

e107 è un sistema basato su modulo del tipo "*portal in a box*" nato da un progetto di Steve Dunstan e corredato da numerosi moduli, utili per supportare le funzioni più disparate. Il progetto è stato concepito abbastanza recentemente ma esiste già una discreta comunità di sviluppatori, peraltro rapidamente in crescita, e quindi è possibile ricevere un buon supporto in caso di difficoltà con l'utilizzo di e107. Si tratta di un sistema di cui val la pena osservarne con attenzione gli sviluppi futuri.

Tecnologia:

Linguaggio di *scripting*: PHP.

Database: MySQL.

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: facile da installare e configurare, consente un settaggio molto rapido. È dotato di una buona interfaccia di amministrazione e sono disponibili diversi moduli aggiuntivi. Progetto in forte crescita.

Difetti: progetto ancora giovane (pochi sviluppatori coinvolti; mancano ancora alcuni tipi di moduli). Anche se sono disponibili temi differenti per personalizzare la grafica, le operazioni di personalizzazioni grafica possono risultare complesse. Manca gestione workflow.

Riferimenti:

❖ www.e107.org: sito del progetto originale.

4.4 – Envolution

Envolution è un sistema basato su modulo molto interessante, derivato da Postnuke, semplice ma efficace, ideale per un'utenza di dimensioni non eccessivamente grandi. Envolution può essere configurato per permettere ai visitatori del sito di inviare articoli, commentare gli articoli pubblicati, partecipare a sondaggi, amministrare il proprio account e molto altro ancora, tutto con strumenti semplici e pochi click del *mouse*. Il sistema gestisce la pubblicazione di articoli e fornisce numerose funzioni interattive tramite moduli aggiuntivi da installare come estensioni del motore principale (meteo, ricette di cucina, mercatino annunci, biglietterie, galleria fotografica, etc).

Una delle caratteristiche importanti di Envolution consiste nel "motore grafico", chiamato Encompass, che permette di eseguire facilmente delle personalizzazioni alla grafica del portale.

Dal punto di vista funzionale, Envolution prevede:

- ⇒ sistema di *template* e modelli per il *layout* del sito;
- ⇒ controllo completo sul posizionamento dei blocchi in base ai moduli;
- ⇒ invio di articoli multipagina usando HTML o testo pieno, con un editor visuale;
- ⇒ sistema sofisticato di autorizzazioni per controllare l'accesso al sito;
- ⇒ gruppi di utenti da integrare col sistema delle autorizzazioni (ogni utente viene assegnato a un gruppo);
- ⇒ sistema di messaggistica privata per scambiare messaggi on line.

Tecnologia:

Linguaggio di *scripting*: PHP (molti moduli hanno integrato altri *scripting* quali Javascript, Java, Flash.).

Database: qualunque tipo di database SQL (MySQL, mSQL, PostgreSQL, ODBC, ODBC_Adabas, Sybase or Interbase).

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: facile da installare e configurare, consente un settaggio molto rapido anche di portali complessi. Le funzioni di amministrazione sono numerose e complete. Esiste ampia documentazione e una folta comunità di sviluppatori (anche in Italia) disponibile a fornire supporto.

Difetti: manca la gestione di un vero *workflow*. Anche se sono disponibili molti temi differenti per personalizzare la grafica, e il motore Encompass agevola le modifiche al layout del portale, le operazioni di personalizzazioni grafica possono risultare comunque complesse.

Riferimenti:

- ❖ www.envolution.it: sito del progetto originale (italiano);
- ❖ www.envolution.com: sito del progetto originale (inglese);
- ❖ www.postnuke.com: sito del progetto Postnuke (da cui deriva Envolution);
- ❖ www.postnuke.it: sito della comunità italiana di supporto al progetto Postnuke.

4.5 – E-Xoops

E-Xoops è un sistema basato su modulo per la gestione dinamica dei contenuti di un sito web. Oltre a realizzare in maniera automatica il layout del sito, permette agli utenti autorizzati di modificarne i contenuti multimediali (ovvero testi, immagini, suoni, video) senza necessità da parte dell'utente di particolari conoscenze tecniche. Il prodotto, sviluppato con *scripting* PHP, è stato concepito secondo un'architettura orientata agli oggetti da cui deriva una certa modularità del sistema e una conseguente maggiore semplicità nello sviluppo dei moduli stessi. Uno degli obiettivi di E-Xoops è quello di ottenere un codice il più semplice e pulito possibile in modo da facilitarne la personalizzazione.

Una caratteristica interessante di E-Xoops è la compatibilità con w.bloggar, uno strumento che permette di pubblicare le news utilizzando un'interfaccia più amichevole di quella disponibile sul browser. La composizione dei contenuti avviene con un editor molto completo che comprende l'upload di file e immagini. Il risultato è un testo html + immagini "*uploadate*" (tramite ftp) del materiale che

si vuole allegare, del quale si può avere un'utilissima anteprima prima della pubblicazione (si può scaricare w.bloggar da wbloggar.com). Inoltre, è disponibile un modulo che crea altri moduli (*moduleCreator*), scaricabile dal sito <http://www.aporta.com/modules/start/> (questo modulo consente di esportare tutto il codice necessario per creare nuovi moduli tramite una comoda interfaccia online).

Tecnologia:

Linguaggio di *scripting*: PHP.

Database: MySQL.

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: facile da installare e configurare, consente un settaggio molto rapido anche di portali complessi. Le funzioni di amministrazione sono numerose e complete. Esiste ampia documentazione e una folta comunità di sviluppatori (anche in Italia) disponibile a fornire supporto.

Difetti: manca la gestione di un vero *workflow*. Anche se sono disponibili temi differenti per personalizzare la grafica, le operazioni di personalizzazioni grafica possono risultare complesse.

Riferimenti:

- ❖ www.e-xoops.com: sito del progetto originale;
- ❖ www.xoops.org: sito del progetto Xoops da cui deriva E-Xoops;
- ❖ www.e-xoops.it: sito della comunità italiana di supporto al progetto.

4.6 - **eZpublish**

eZpublish è un CMS sviluppato a partire dal 1999 dalla eZsystem, piccola società norvegese, e si rivolge a gestori di portali informativi, siti aziendali, sistemi intranet ed extranet commerciali. Si tratta di un sistema nato per sviluppare rapidamente dei portali complessi, ma diversamente dagli analoghi PHPNuke, PostNuke o Evolution si dimostra particolarmente flessibile nella gestione del layout e della grafica. Altra caratteristica importante è la possibilità di riutilizzare alcuni moduli in differenti sezioni del portale. Per questo motivo, eZPublish si può considerare un sistema a metà strada tra la tipologia con contenuti a oggetti e un vero *framework* come Zope (vedi infra).

EZpublish viene fornito con due licenze:

- ❖ GPL, per sviluppare soluzioni *Open Source* (e quindi “free”);

- ❖ Commerciale, per sviluppare soluzioni che possono diventare proprietarie e alle quali è possibile dare un proprio marchio (distinto da eZpublish).

Le principali caratteristiche di eZPublish si possono così sintetizzare:

- multiplatforma (anche se sviluppato con *scripting* PHP);
- multilingua (disponibile in oltre 20 lingue);
- indipendenza dal database;
- disponibilità di numerosi moduli predefiniti per la gestione degli oggetti di base (articoli, calendario, forum, link, messaggi, notizie, sondaggi, commercio elettronico, banner pubblicitari, amministrazione portale, etc);
- completa gestione dei template per garantire la separazione dei contenuti dal layout;
- statistiche di accesso;
- gestione di gruppi di utenti con permessi diversi.

Tecnologia:

Linguaggio di *scripting*: PHP

Database: MySQL, Postgres

Sistemi operativi: Linux, Windows, Mac Os X

Web server: Apache

Pregi: facile da installare, consente una configurazione abbastanza rapida anche di portali molto complessi e articolati. Le funzioni di amministrazione sono numerose e complete. La versione 3.0 ha fatto passi da gigante rispetto alla precedente 2.2 (specialmente per quanto riguarda la gestione della grafica, del *layout*, del *workflow*). Consente la creazione di contenuti con modalità “a oggetti”.

Difetti: le attività di configurazione e personalizzazione necessitano di personale esperto. La documentazione disponibile non è sufficiente e, quel poco che esiste, in certi casi è addirittura ambigua (davvero un peccato per un progetto così importante; sarebbe molto utile un *tutorial*).

Riferimenti:

- ❖ <http://developer.ez.no>: sito del progetto originale.

4.7 – Fullxml

FullXML è un sistema di tipo “*portal in a box*” basato sull’utilizzo della tecnologia XML. Non pretende di essere il più potente CMS disponibile in circolazione ma si candida a diventare un ottimo riferimento per le esigenze di

sviluppo di piccoli siti o portali con un minimo di funzioni disponibili senza dover impiegare troppo tempo per capirne il funzionamento.

Fullxml è un progetto nato verso la fine dell'estate del 2001, per intuizione di John Roland. Quasi per gioco, Roland e un amico si misero a scrivere (in un sol giorno) il codice di una piccola demo per saggiare le potenzialità di XML; il risultato fu talmente soddisfacente che i due si convinsero delle ottime prospettive fornite da un tale sistema. Qualche tempo dopo, Fullxml divenne ufficialmente un progetto *Open Source* con la concessione di uno spazio su "Sourceforge.net".

I concetti fondamentali su cui poggia il progetto Fullxml sono: archiviazione dei dati tramite file XML (nessun utilizzo di db esterni), impiego di tecnologia ASP/MSXML, funzioni di amministrazione semplici ed efficaci.

Tecnologia:

Linguaggio di *scripting*: ASP - XML

Database: i dati sono memorizzati in file XML

Sistemi operativi: qualunque sistema compatibile con lo scripting ASP e XML

Web server: qualunque webserver in grado di far girare ASP e XML

Pregi: sistema di facile installazione e configurazione, si presta a sviluppare con estrema rapidità ed efficacia dei portali completi e flessibili, adattabili alle esigenze più disparate. Interfaccia di amministrazione molto intuitiva.

Difetti: non sono disponibili molti moduli, tranne quelli "classici" dei più comuni analoghi sistemi. I parametri degli utenti (tipicamente, le password) sono archiviati in chiaro nei file XML senza essere criptati (questi file vanno quindi memorizzati in cartelle protette per impedirne la lettura). Manca un processo di workflow.

Riferimenti:

- ❖ www.fullxml.com: sito del progetto originale;
- ❖ <http://sourceforge.net/projects/fullxml>: sito del progetto su Sourceforge.

4.8 – Maxwebportal

Maxwebportal è un sistema basato su modulo, di tipo "*portal in a box*", sviluppato sull'impalcatura dello SnitzForum, altro importante progetto *Open Source* realizzato con scripting ASP (Active Server Pages). Maxwebportal è particolarmente adatto per le esigenze di piccoli siti o portali per i quali sono sufficienti le funzioni fornite, senza dover impiegare troppo tempo per capirne il funzionamento. Sono disponibili tutti i classici moduli per gestire un portale (news, forum, annunci, faq, etc).

Tecnologia:

Linguaggio di *scripting*: ASP

Database: MS Access, MS SQL

Sistemi operativi: qualunque sistema compatibile con lo scripting ASP

Web server: qualunque webserver in grado di far girare ASP

Pregi: sistema di facile installazione e configurazione, si presta a sviluppare con estrema rapidità ed efficacia dei portali completi e flessibili, adattabili alle esigenze più disparate. Interfaccia di amministrazione molto intuitiva.

Difetti: non sono disponibili molti moduli, tranne quelli “classici”. La personalizzazione grafica e del layout richiede l’intervento di personale esperto. Manca un processo di workflow.

Riferimenti:

- ❖ www.maxwebportal.com: sito del progetto originale;
- ❖ <http://www.maxwebportal.it>: sito della comunità italiana a supporto del progetto.

4.9 – **Metadot**

Metadot è un potente sistema per gestire contenuti, ideale per intranet, extranet, siti e portali (anche di grandi dimensioni). Agli utenti non sono richieste particolari conoscenze tecniche.

Una soluzione sviluppata con Metadot si compone di tre componenti principali:

- l’interfaccia standard ;
- l’interfaccia utente personalizzata, nella quale ogni utente può scegliere un modello grafico di proprio gradimento oppure la tipologia di contenuto che desidera visualizzare nelle pagine del sito;
- contenuti utente personalizzati, nella quale ogni utente può impostare un certo numero di pagine da egli stesso modificabili.

L’accesso degli utenti alle risorse del sito è regolato da un sofisticato sistema di sicurezza a livello di utente e di gruppi. Sono disponibili numerosi moduli standard (calendario, forum, news, FAQ, etc). e molti altri aggiuntivi (chiamati “gizmos”) sono scaricabili dalla rete.

Tecnologia:

Linguaggio di *scripting*: Perl.

Database: MySQL.

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: si tratta di un ottimo sistema, veramente completo, flessibile, si adatta alle esigenze più disparate. È dotato di una buona interfaccia di amministrazione (tramite la quale è possibile modificare il layout e la grafica).

Difetti: essendo scritto in perl, l'installazione e la configurazione richiedono l'intervento di personale piuttosto esperto (così come eventuali interventi di personalizzazione).

Riferimenti:

❖ <http://www.metadot.net/metadot/index.pl>: sito del progetto originale

4.10 – MySource

MySource è un sistema sviluppato dalla *software house* australiana Squiz, adatto per l'implementazione di siti, portali, intranet aziendali. È stato progettato con l'obiettivo di consentire, anche ai meno esperti, di sviluppare soluzioni sul *web* con la massima sicurezza, professionalità, risparmio. MySource gestisce gli accessi utente di tipo multilivello per consentire la collaborazione di redazione dei contenuti.

Altre caratteristiche del prodotto sono le seguenti:

- incorpora vari strumenti di pubblicazione del tipo “WYSIWYG” (*what you see is what you get*);
- compatibile con pagine web esistenti;
- supporto al caricamento di oggetti multimediali;
- gestione documenti con possibilità di inserire allegati;
- disponibile un modulo per sviluppare altri moduli.

Tecnologia:

Linguaggio di *scripting*: PHP.

Database: MySQL.

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: dotato di un'interfaccia di amministrazione semplice e completa, tramite la quale è possibile modificare il layout e la grafica. Disponibile una buona documentazione di supporto.

Difetti: l'installazione e la configurazione richiedono l'intervento di personale piuttosto esperto. Manca la gestione del workflow.

Riferimenti:

❖ <http://mysource.squiz.net>: sito del progetto originale.

4.11 – Nucleus

Nucleus è un sistema nato dal progetto di Wouter Demuynck, giovane informatico belga, per supportare lo sviluppo di BLOG o di giornali *online*. Le principali caratteristiche del prodotto sono:

- gestione di uno o più BLOG, anche sulla stessa pagina;
- disponibilità di molti *plugin* (moduli aggiuntivi) per aggiungere funzionalità extra;
- gestione articoli con possibilità di aggiungere commenti, archivio per categorie, data;
- gestione di *media* caricabili online (immagini, video, audio, etc);
- possibilità di personalizzare grafica e *template*;
- compatibile con i principali standard per l'acquisizione da remoto dei contenuti di altri siti (XML-RPC, RSS syndication).

Tecnologia:

Linguaggio di *scripting*: PHP.

Database: MySQL.

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: facile da installare e configurare, consente un settaggio molto rapido. È dotato di una buona interfaccia di amministrazione (tramite la quale è possibile modificare il layout e la grafica) e sono disponibili alcuni moduli aggiuntivi. Progetto in crescita.

Difetti: sistema adatto per esigenze limitate e particolari (weblog o giornale online). Non sono disponibili molti moduli aggiuntivi. Manca la gestione del workflow.

Riferimenti:

- ❖ <http://www.nucleuscms.org>: sito del progetto originale;
- ❖ <http://www.xiffy.nl/wakka/Nucleus>: sito dal quale è possibile scaricare moduli e temi aggiuntivi.

4.12 - PHPNuke

PHPNuke è un sistema basato su modulo del tipo “*portal in a box*” scritto dal venezuelano Francisco Burzi, il quale nell'estate del 2000 creò la prima versione di PHPNuke. Il prodotto catturò subito la simpatia e l'interesse di molti e attorno al progetto iniziale si è rapidamente affermata una delle più folte comunità di

sviluppatori del mondo *Open Source*. Il successo e la crescita del prodotto sono stati tali che di recente è stata rilasciata la versione 6.5.

PHPNuke è un sistema che integra una serie di strumenti utili a creare un sito/portale informativo. È dotato di un vasto numero di funzioni standard presenti già nel pacchetto di installazione (articoli, news, forum, faq, recensioni, sondaggi, banner, etc), ma sono reperibili centinaia di altri moduli, sviluppati da terzi e liberamente utilizzabili, adatti a tantissime esigenze anche molto particolari (meteo, ricette di cucina, mercatino annunci, biglietterie, galleria fotografica, etc). Ma sono reperibili anche centinaia di temi per personalizzare la grafica del portale a soli colpi di pochi click. PHPNuke è pertanto adatto alla gestione di Intranet aziendali come a sviluppare veri sistemi di e-commerce, portali per associazioni, enti pubblici e aziende, testate giornalistiche, sistemi di e-learning, etc.

Tecnologia:

Linguaggio di *scripting*: PHP (molti moduli hanno integrato altri *scripting* quali Javascript, Java, Flash.).

Database: qualunque database di tipo SQL (MySQL, mSQL, PostgreSQL, ODBC, ODBC_Adabas, Sybase or Interbase).

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: facile da installare e configurare, consente un settaggio molto rapido anche di portali molto complessi e articolati. Le funzioni di amministrazione sono numerose e complete. Esiste ampia documentazione e una fortissima comunità di sviluppatori (anche in Italia) disponibile a fornire supporto.

Difetti: anche se sono disponibili tantissimi temi differenti per personalizzare la grafica, il portale prevede una struttura di visualizzazione fissa e non facilmente modificabile (se non da personale piuttosto esperto).

Riferimenti:

- ❖ www.phpnuke.org: sito del progetto originale;
- ❖ www.nukeforums.com: supporto tecnico agli utenti;
- ❖ www.nukedownloads.com: *mirror* di file da scaricare;
- ❖ www.somara.com: raccolta di temi e grafica;
- ❖ www.nukethemes.com: raccolta di temi e grafica;
- ❖ www.ecomjunk.com: raccolta di moduli aggiuntivi;
- ❖ www.nukeaddon.com: raccolta di moduli aggiuntivi;
- ❖ www.spaghettribrain.com: comunità di utenti e sviluppatori in lingua italiana;
- ❖ www.phpnuke.it: comunità di utenti e sviluppatori in lingua italiana;
- ❖ www.splatt.it: comunità di utenti e sviluppatori in lingua italiana;
- ❖ www.nukeitalia.com: comunità di utenti e sviluppatori in lingua italiana.

4.13 - **Plone**

Plone è un sistema costruito sul *framework* di Zope e si presenta come soluzione ideale per applicazioni di tipo *intranet* e per la pubblicazione di documenti nell'ambito di realtà medio-grandi. Uno dei pregi di Plone consiste nel mettere in grado qualunque utilizzatore di attivare, con una certa facilità e piuttosto rapidamente, un ottimo CMS basato su ZOPE. Plone viene fornito con un motore di *workflow*, livelli di sicurezza e ruoli predefiniti, una gamma di diversi tipi di contenuto, supporto multi lingua. Esiste una comunità di sviluppatori molto folta in grado di fornire supporto. Chi si volesse spingere in attività di personalizzazione ai moduli o al *layout*, incontrerà però le stesse difficoltà che pone il sistema Zope (scripting Python).

Tecnologia:

Linguaggio di *scripting*: C e Python

Database: ZODB (Database nativo interno), database relazionali esterni (mysql, postgres, sybase etc...)

Sistemi operativi: Linux, Windows, Mac OSX, Unix, Solaris

Web server: ZServer (Web Server nativo interno), Apache

Requisiti server: Apache (opzionale), MySQL o altro database esterno prescelto (opzionale)

Pregi: consente di attivare rapidamente un portale sulla “grande” piattaforma Zope. Facile da installare.

Difetti: le attività di configurazione e personalizzazioni dei moduli o del *layout* necessitano di personale esperto (occorre una buona conoscenza di Zope e dello scripting Python).

Riferimenti:

- ❖ www.plone.org: sito del progetto originale;
- ❖ http://artix.dyndns.org/zope/AeM/AM/infosito/plonebookaem/backtalk_book_view: documentazione in italiano;
- ❖ <http://www.communitye.net/>: sito di “Community Enabler”, ottimo *framework* che si appoggia a Plone (lo si installa come componente aggiuntivo e si ottiene un ottimo sistema di tipo “portal in a box”).

4.14 – **PMachine**

PMachine è un sistema nato per supportare lo sviluppo di BLOG o di giornali *online*. Le principali caratteristiche del prodotto sono:

- gestione di uno o più BLOG;
- gestione articoli con possibilità di aggiungere commenti, archivio per categorie, data;

- gestione di *media* caricabili online (immagini, video, audio, etc);
- possibilità di personalizzare grafica e *template*;
- compatibile con i principali standard per l'acquisizione da remoto dei contenuti di altri siti (XML-RPC, RSS syndication).

Tecnologia:

Linguaggio di *scripting*: PHP.

Database: MySQL.

Sistemi operativi: Linux, Windows.

Web server: Apache.

Pregi: facile da installare e configurare, consente un settaggio molto rapido. È dotato di una buona interfaccia di amministrazione (tramite la quale è possibile modificare il layout e la grafica).

Difetti: sistema adatto per esigenze limitate e particolari (weblog o giornale online). Non sono disponibili molti moduli aggiuntivi. Manca la gestione del workflow.

Riferimenti:

- ❖ www.pmachine.com: sito del progetto originale.

4.15 – SmartPublish

SmartPublish è un sistema ideato per la gestione di contenuti all'interno di siti e portali internet dinamici, con particolare riferimento alla gestione di contenuti di tipo giornalistico. Il suo sviluppo è iniziato per opera di Paata Gugushvili, uno scienziato georgiano con un curriculum incredibile. La versione attualmente utilizzabile con licenza *Open Source* è la Light II. Paata sta lavorando allo sviluppo (ormai giunto a buon punto) della versione Pro, con molte funzioni aggiunte alla versione Light. Non si capisce ancora se la versione Pro verrà rilasciata ancora con licenza *Open Source*.

Le caratteristiche di base offerte da SmartPublish comprendono:

- contenuti organizzati con struttura del tipo *file system*;
- creazione automatica di menu, anche di tipo grafico;
- supporto alla gestione di file e immagini;
- ridimensionamento automatico delle immagini caricate;
- gestione delle protezioni sulle singole pagine, e permessi di accesso, per gruppi o singoli utenti;
- gestione del *frontend* tramite pannello di amministrazione.

Tecnologia:

Linguaggio di *scripting*: ASP

Database: MS Access

Sistemi operativi: Windows

Web server: IIS

Requisiti server: qualunque webserver in grado di far girare ASP.

Pregi: facile da installare e configurare, è un buon sistema, adatto soprattutto per gestire contenuti di tipo giornalistico. Disponibile buona documentazione (anche se per ora solo in inglese).

Difetti: il sistema è nato di recente e quindi non esiste ancora una comunità di supporto. Le personalizzazioni grafiche vanno fatte modificando il codice.

Riferimenti:

- ❖ <http://bpg.sytes.net/splight/journal/default.asp>: sito del progetto originale;
- ❖ <http://bpg.sytes.net/sp/journal/default.asp>: sito del progetto nella versione “PRO”.

4.16 – Typo3

Typo3 è un sistema molto evoluto per la gestione di contenuti *web* all’interno di siti e portali internet dinamici, anche di dimensioni rilevanti. Il suo sviluppo è iniziato nel 1999 dall’idea di un ragazzo venticinquenne danese, Kasper Skarhoj, che ha poi scelto di rendere disponibile il risultato del suo lavoro come software *Open Source* con licenza GPL.

Le caratteristiche di base offerte da Typo3 comprendono:

- sviluppo delle pagine basato su *template*;
- contenuti organizzati con struttura del tipo *file system*;
- creazione automatica di menu, anche di tipo grafico;
- disegno delle pagine personalizzabile in funzione del browser che le richiede (in base al numero di IP, al sistema operativo, alla nazionalità, ai parametri di configurazione scelti dall’utente);
- disponibilità di decine di componenti aggiuntivi a quelli di base (moduli funzione), già pronti per gli usi più disparati;
- supporto alla gestione di file e immagini;
- ridimensionamento automatico delle immagini caricate;
- gestione delle protezioni sulle singole pagine, e permessi di accesso, per gruppi o singoli utenti;
- gestione di processi di *workflow*.

Caratteristica importante e ben gestita in Typo3 riguarda l'ampio utilizzo dei *template*. Il prodotto è fornito di documentazione e manualistica sufficientemente estesa e chiara, scaricabile dal sito.

Tecnologia:

Linguaggio di *scripting*: PHP

Database: MySQL 3.22.x o superiori

Sistemi operativi: Linux, Windows

Web server: Apache 1.3.x

Requisiti server: Apache con modulo PHP abilitato e configurato con supporto PNG e GIF. MySQL server, libreria ImageMagick

Pregi: facile da installare e configurare, è un ottimo sistema, veramente completo, adatto anche per grandi aziende. Disponibile ampia ed esauriente documentazione (anche se per ora solo in inglese).

Difetti: il sistema, essendo piuttosto sofisticato, necessita di un pò di tempo per acquisire dimistichezza con alcune funzioni di amministrazione, specialmente quelle relative alla gestione dei *template*.

Riferimenti:

- ❖ www.typo3.com: sito del progetto originale;
- ❖ www.typo3.org: sito della comunità di sviluppatori.

4.17 – Xaraya

Xaraya è un sistema basato su modulo del tipo “*portal in a box*”, scritto in PHP, che utilizza un robusto sistema di permessi utente, gestione dei dati multilingue ed è supportato da una quarantina di sviluppatori. Xaraya è un sistema modulare, indipendente dal tipo di database e consente lo sviluppo di siti-portali completamente dinamici e multiplatforma. Sono presenti numerosi moduli standard (news, links, forum, FAQ, download, galleria fotografica, banner, etc).

Le principali caratteristiche del prodotto sono:

- completamente basato su *database* (PHP-ADODB) ;
- personalizzabile con numerosi moduli sviluppati da terze parti;
- particolare attenzione riservata alla sicurezza (è dotato di un modulo per una gestione utenti multilivello);
- possibilità di personalizzare grafica e *layout*;
- si installa facilmente su tutte le principali piattaforme;
- compatibile con i principali standard per l'acquisizione da remoto dei contenuti di altri siti (RSS).

Tecnologia:

Linguaggio di *scripting*: PHP.

Database: MySQL, PostgreSQL.

Sistemi operativi: Linux, Windows, MacOSX, Solaris, HP-UX, AIX..

Web server: Apache o IIS.

Pregi: facile da installare e configurare, consente un settaggio molto rapido. È dotato di una buona interfaccia di amministrazione (tramite la quale è possibile modificare il layout e la grafica). Sistema in crescita, da tenere sotto osservazione.

Difetti: sistema “giovane” e quindi non sono disponibili molti moduli aggiuntivi. Manca la gestione del workflow.

Riferimenti:

❖ www.xaraya.com: sito del progetto originale.

4.18 – WebGui

WebGUI è una piattaforma sviluppata dalla “Plain Black Software” con una particolare attenzione all’esigenza di creare un sistema per gestire contenuti modulare, scalabile, personalizzabile, installabile su diverse piattaforme e semplice da usare. Queste caratteristiche ne fanno un buon strumento anche per esigenze sofisticate, quali possono essere quelle di un portale oppure quelle di una Intranet o Extranet aziendale, fino ad applicazioni di *ecommerce* o di *E-learning*. Costruire un’applicazione in WebGui vuol dire costruire un oggetto riutilizzabile (*Wobject*). Per esempio, una volta creata un’applicazione per un forum, è possibile inserire forum in varie sezioni differenti del sito (nella maggior parte degli altri sistemi occorre invece reindirizzare gli utenti alla "sezione forum" del tuo sito).

Tecnologia:

Linguaggio di *scripting*: Perl.

Database: MySQL, PostgreSQL.

Sistemi operativi: Linux, Windows.

Web server: Apache, IIS.

Pregi: si tratta di un buon sistema, completo, flessibile, si adatta alle esigenze più disparate. È dotato di una buona interfaccia di amministrazione (tramite la quale è possibile modificare il layout e la grafica).

Difetti: essendo scritto in perl, l’installazione e la configurazione richiedono

l'intervento di personale piuttosto esperto (così come eventuali interventi di personalizzazione). Manca la gestione del *workflow*.

Riferimenti:

- ❖ <http://www.plainblack.com/webgui>: sito del progetto originale;
- ❖ <http://www.webgui.it>: sito italiano di supporto al progetto.

4.19 - Zope

Zope è il leader di mercato degli *Application Server Open Source*. Offre un'architettura di tipo "a oggetti" che permette di costruire portali Web potenti, modulari, dinamici e con ottime prestazioni di navigazione. Zope, che incorpora un server web e un *database object oriented*, rappresenta un vero e proprio strumento di sviluppo e di *publishing* dinamico, scritto nel linguaggio di scripting *Python*. Zope separa il livello dei dati (archiviati nel proprio database o in database relazionali esterni), dai livelli della logica, scritta con metodi Python, e della presentazione, realizzata con un HTML arricchito denominato DHTML (o con il linguaggio di template TAL).

Le principali caratteristiche di ZOPE comprendono:

- possibilità di utilizzo su piattaforme diverse: Windows, Linux, Solaris;
- linguaggi DHTML e TAL per la costruzione dinamica delle pagine Web;
- linguaggio Python per la implementazione delle classi di base dell'applicazione;
- organizzazione in gerarchie di oggetti;
- raffinata gestione interna delle problematiche di sicurezza;
- supporto di XML;
- database object oriented interno (ZODB);
- connettività con i principali database relazionali;
- *webserver* interno e supporto dei principali *webserver* presenti sul mercato;
- supporto nativo protocollo di SSL (Secure Sockets Layer).

Sul sito di ZOPE sono disponibili centinaia di moduli aggiuntivi prodotti da una nutrita comunità di sviluppatori sparsi in tutto il mondo.

Tecnologia:

Linguaggio di *scripting*: C e Python

Database: ZODB (Database nativo interno), database relazionali esterni (mysql, postgres, sybase, etc.)

Sistemi operativi: Linux, Windows, Mac Osx, Unix, Solaris

Web server: ZServer (Web Server nativo interno), Apache

Requisiti server: Apache (opzionale), MySQL o altro database esterno prescelto (opzionale)

Pregi: è considerato unanimemente il migliore prodotto della categoria.

Difetti: per personalizzare i moduli e il *layout* grafico è richiesta una buona conoscenza dello scripting C e Python, oltre allo ZODB *database* e lo ZServer *web server* (serve avere a disposizione personale esperto).

Riferimenti:

- ❖ www.zope.org: sito del progetto originale
- ❖ www.zope.it: comunità di utenti e sviluppatori in lingua italiana.

5 – CMS “quasi” *Open Source*

Nonostante l’obiettivo di questo libro sia soprattutto quello di fare apprezzare le potenzialità della tecnologia e dei software *Open Source*, è pur sempre nello spirito di favorire la riduzione del cosiddetto “divario digitale”, tra chi ha molti mezzi a disposizione e chi invece non ne ha molti (o non ne ha affatto), che prendiamo ora brevemente in esame anche alcuni sistemi che non afferiscono a progetti *Open Source* ma che vengono commercializzati con un “pudore” che merita molto rispetto. Si tratta infatti di prodotti di ottima qualità che vengono offerti con una licenza di valore veramente accessibile (nell’ordine di poche centinaia di Euro). Molto spesso, tali sistemi sono perfettamente paragonabili a molti altri prodotti commerciali per la cui licenza vengono però chieste centinaia di migliaia di Euro.

Con una certa dose di forzatura, possiamo definire queste soluzioni software “quasi” *Open Source*; ai rispettivi sviluppatori è forse mancato un pochino di coraggio per fare delle proprie licenze commerciali delle licenze di utilizzo libero.

5.1 – Constructioner

Ottimo sistema per configurare rapidamente e con estrema efficacia siti o portali di qualunque dimensione. Constructioner dispone di un sistema di *backoffice* molto evoluto, fornisce alcune delle principali funzioni di base di un portale e consente di sviluppare facilmente proprie funzioni.

Le caratteristiche peculiari del prodotto sono le seguenti:

- editor di testo del tipo “WYSIWYG” (*what you see is what you get*) con possibilità di caricare nuove immagini nella galleria;
- gestione evoluta degli utenti: ogni utente viene profilato definendo cosa può fare nel sito (quali sezioni, quali funzioni o contenuti, etc);
- l’aspetto grafico e la struttura del sito possono essere modificati online tramite semplice intervento sul *template*;
- possibilità di modificare con intuitiva interfaccia grafica le tabelle del database.

Tecnologia:

Linguaggio di *scripting*: PHP

Database: MySQL

Sistemi operativi: Linux, Windows

Web server: Apache, IIS

Requisiti *server*: MySQL Zend Optimizer.

Pregi: sistema di facile installazione e configurazione, si presta a sviluppare con estrema rapidità ed efficacia dei portali completi e flessibili, adattabili alle esigenze più disparate. Interfaccia di amministrazione molto intuitiva, tramite la quale è possibile realizzare facilmente dei moduli aggiuntivi (*plugin*). Costa 390 USD.

Difetti: non sono disponibili moduli aggiuntivi a quelli forniti con il pacchetto. Manca gestione del *workflow*.

Riferimenti:

- ❖ <http://www.constructioner.com>: sito della casa produttrice.

5.2 – SiteSupra

SiteSupra è un ottimo sistema per lo sviluppo di siti o portali di dimensione non eccessivamente grandi. Utilizza un sistema rivoluzionario per la gestione della grafica e del *layout* (con la tecnica “*drag&drop*”). SiteSupra mette a disposizione una certa quantità di moduli standard adattabili a molte esigenze e fornisce la gestione del *workflow* di pubblicazione delle pagine. Lo sviluppo di propri moduli personalizzati è facilitata dalla possibilità di utilizzare delle comode API (*Application program interface*).

Le caratteristiche peculiari del prodotto sono le seguenti:

- editor di testo del tipo “WYSIWYG” (*what you see is what you get*) con possibilità di caricare nuove immagini nella galleria;

- gestione evoluta degli utenti: ogni utente viene profilato definendo cosa può fare nel sito (quali sezioni, quali funzioni o contenuti, etc);
- gestione con modalità di tipo “*drag&drop*” dell’aspetto grafico e della struttura del sito;
- supportato il processo di *workflow* di pubblicazione delle pagine.

Tecnologia:

Linguaggio di *scripting*: PHP

Database: MySQL

Sistemi operativi: Linux, Windows

Web server: Apache, IIS

Requisiti *server*: MySQL Zend Optimizer.

Pregi: sistema di facile installazione e configurazione, si presta a sviluppare con estrema rapidità ed efficacia dei portali completi e flessibili, adattabili alle esigenze più disparate. Interfaccia di amministrazione molto intuitiva. Disponibile il supporto al *workflow*. Costa 139 USD.

Difetti: non sono disponibili molti moduli aggiuntivi a quelli forniti con il pacchetto.

Riferimenti:

- ❖ <http://www.sitesupra.com>: sito della casa produttrice.

5.3 – Siteworkspro

SiteWorksPro è un ottimo sistema per configurare rapidamente e con estrema efficacia piccoli siti o portali, per i quali sono sufficienti le funzioni standard fornite senza dover impiegare troppo tempo per capirne il funzionamento. Particolarmente indicato per chi deve gestire contenuti con modalità di collaborazione tra più redattori-editori (articoli, notizie, commenti, sondaggi, recensioni, etc).

Le caratteristiche peculiari del prodotto sono le seguenti:

- editor di testo del tipo “WYSIWYG” (*what you see is what you get*) con possibilità di caricare nuove immagini nella galleria;
- gestione evoluta degli utenti: ogni utente viene profilato definendo cosa può fare nel sito (quali sezioni, quali funzioni o contenuti, etc);
- l’aspetto grafico e la struttura del sito possono essere modificati online tramite semplice intervento sul *template*;
- possibilità di inserire articoli composti da più pagine, corredati da immagini o allegati scaricabili e altri articoli o notizie correlati. I lettori possono votare per ogni articolo e inserire commenti.

Tecnologia:

Linguaggio di *scripting*: PHP

Database: MySQL

Sistemi operativi: Linux, Windows

Web server: Apache, IIS

Pregi: sistema di facile installazione e configurazione, si presta a sviluppare con estrema rapidità ed efficacia dei portali completi e flessibili, adattabili alle esigenze più disparate. Interfaccia di amministrazione molto intuitiva. Costa 99 USD.

Difetti: non sono disponibili moduli aggiuntivi a quelli forniti con il pacchetto.

Riferimenti:

❖ <http://www.siteworkspro.com/index.php>: sito della casa produttrice.

6 – Un osservatorio permanente

La rassegna di alcuni dei principali sistemi *Open Source* per la gestione dei contenuti in Internet, così come sviluppata nel capitolo 4, è “viziata” da un grosso limite, quello temporale. Nel momento in cui il lettore sta scorrendo queste pagine, lo scenario di riferimento delle tecnologie esaminate potrebbe essere cambiato, e anche di molto. Per questo motivo, se nessun altro autore ha aggiornato questo libro, suggerisco di consultare con attenzione le seguenti risorse disponibili in Internet:

- www.cmsinfo.org: sito di una comunità virtuale di utenti e sviluppatori di sistemi di gestione contenuti in internet (CMS). Lo scopo del sito è quello di fornire notizie, informazioni e “dritte” sempre aggiornate sul panorama dei CMS, in particolare *Open Source*, ma anche di sistemi commerciali che non siano i soliti blasonati Broadvision, Vignette & Co.
- <http://www.cmsreview.com/cmslisting.html>: sito che propone un’interessante rassegna di schede prodotto CMS (abbastanza aggiornato).
- <http://www.intranetjournal.com/tools/cm/>: sito che propone una semplice raccolta di link verso pagine di approfondimento a vari CMS.
- <http://www.cms-list.org/>: sito di riferimento della una *mailing list* di una comunità virtuale che si pone come obiettivo il confronto continuo tra utenti e sviluppatori di CMS relativamente a strategie di prodotto, aspetti tecnici, nuove tecnologie, standards.
- www.openlabs.it: sito dell’omonima associazione culturale senza fini di lucro fondata con l’impegno di promuovere e sostenere progetti coerenti con la filosofia del software libero e *Open Source*. L’Associazione si propone di

promuovere e diffondere l'esercizio cosciente della libertà sia riguardo alle scienze informatiche e telematiche sia all'uso delle relative tecnologie nella società civile, nonché di sostenere le forme di licenza e di distribuzione che consentano l'uso libero della produzione della attività intellettuale. Organizza corsi, seminari, convegni e *workshop* molto utili.

- <http://opensource.dschola.it/>: sito dedicato al mondo della scuola, vuole fornire un punto di riferimento e di contatto a tutte le persone e alle organizzazioni interessate allo sviluppo del software *Open Source*. Si pone come obiettivo di poter contribuire allo sviluppo di un movimento che contribuisca al progresso dell'informatica non solo nel campo teorico, ma anche in quello economico e sociale, rendendo il mercato più aperto e più libero.
- http://www.plainblack.com/product_comparison: interessante tabella di comparazione di alcuni CMS, a cura della Plain Black Software.

7 – Come scegliere il sistema giusto

È stato già precisato come non si possa affermare che un CMS sia in assoluto il migliore; si può però individuare un sistema che sia più adatto di altri a rispondere alle proprie esigenze.

Per agevolare la scelta del sistema più adatto alle proprie esigenze, proponiamo una semplice *check-list* che speriamo possa essere di aiuto:

1. **tipologia** di sito sviluppare: occorre chiarirsi bene, fin da subito, quali sono le esigenze che il sito dovrà soddisfare. Potrebbe servire un semplice sito informativo con pagine statiche oppure un vero portale, un sito di *news* oppure un *blog*, un sistema di *e-learning* (formazione a distanza) oppure un sistema di *groupware* per una intranet-extranet. Ma potrebbe servire un sistema che risulti dalla combinazione di una o più delle tipologie indicate.
2. **caratteristiche** e funzioni: occorre capire se dovranno essere gestite informazioni statiche o anche dinamiche (calendario eventi, upload/download documenti, galleria immagini, sondaggi, etc.). Per gestire contenuti dinamici si può fare ricorso a moduli e *plug-in* extra che potrebbe essere necessario aggiungere alle funzioni base del sistema. Occorre fare attenzione perché ci possono essere differenze molto significative da un sistema all'altro e anche alla versione del sistema prescelto. Potrebbe essere utile fare un elenco delle funzionalità ritenute necessarie (tipo lista della spesa) e verificare quale sistema è dotato del maggior numero di funzioni desiderate. In ogni caso, sarà molto utile fare qualche test preliminare.
3. **grafica** e *layout*: occorre capire che esigenze di personalizzazione grafica e di layout delle pagine sono necessarie (tema del sito, detto anche *skin*). L'aspetto grafico (detto anche *look&feel*) può essere un biglietto da visita fondamentale. Molti sistemi utilizzano i CSS (*cascading style sheet*, fogli di stile) e i

template (modelli di *layout* e grafica preconfezionati) per consentire di cambiare rapidamente il *look&feel* del sito. Presentare lo stesso contenuto con differenti temi o *skin* può avere un notevole impatto sui visitatori del sito. Anche senza dover creare un proprio tema, si possono trovare in rete centinaia di temi compatibili con il proprio sistema; normalmente questi temi sono liberamente scaricabili e utilizzabili.

4. gestire i **contenuti**: il principale problema, a regime, dovrà essere la produzione dei contenuti da inserire nel sito. Senza contenuti interessanti, un sito, per quanto ben fatto, risulterà inutile ai visitatori e presto verrà abbandonato a se stesso. Un visitatore tornerà ancora sul nostro sito solo se avrà la sensazione che potrà trovare dei contenuti utili e interessanti per lui. Le tecniche per riempire il sito di contenuti interessanti sono molte: pubblicare articoli e storie, condividere oggetti multimediali (foto, immagini, video, file musicali), consentire anche ai visitatori di inserire propri contenuti, partecipare a discussioni o sondaggi, etc. Ovviamente molto dipende dal tipo di sistema utilizzato (blog, sito di news o di aggregazione contenuti da altri siti tramite RSS). Chiunque, visitando un sito, si aspetta di scoprire o imparare qualcosa di nuovo, indicendolo magari a tornare ancora su quello stesso sito.
5. **manutenzione** e sviluppo: ultima cosa da considerare (anche se non meno importante) è la necessità di pianificare futuri cambiamenti, specialmente nella prospettiva che il sito riscontri successo e debba quindi evolvere verso dimensioni maggiori e caratteristiche più raffinate. Sarà buona norma pensare anche ad un sistema di *disaster recover*. I vari sistemi disponibili gestiscono in modi anche molto differenti queste esigenze. Potrebbe essere utile scambiare qualche parola e chiedere l'opinione ad altri amministratori di sistemi analoghi al proprio per capire come risolvono le problematiche indicate.

Appendice

Comparazione delle caratteristiche di alcuni CMS Open Source

Nome prodotto	Tecnologia	Installazione	configurazione	Interfaccia amministrazione	Moduli/funzioni aggiuntive	Personalizzazione grafica/layout	Gestione ruoli utenti	Community
APC ActionApps	Linux, PHP, Apache, MySQL	**	*	**	*	*	**	*
Ariadne	Linux, PHP, Apache, MySQL	*	*	***	***	**	***	*
E107	Linux/Windows, PHP, Apache, MySQL	****	****	***	***	**	**	**
Envolution	Linux/Windows, PHP, Apache, MySQL	****	****	****	****	***	***	****
E-Xoops	Linux/Windows, PHP, Apache, MySQL	****	****	***	**	**	**	**
EZPublish	Linux/Windows, PHP, Apache, MySQL	*	*	**	*****	**	*****	*
FullXML	Windows, ASP, MS IIS, XML	****	****	****	**	***	**	**
Maxwebportal	Windows, ASP, MS IIS, MSAccess	****	**	**	**	**	**	*
Metadot	Linux/Windows, Perl, Apache, MySQL	*	*	****	****	**	**	*
MySource	Linux/Windows, PHP, Apache, MySQL	*	*	****	***	***	****	*
Nucleus	Linux/Windows, PHP, Apache, MySQL	****	****	***	**	**	**	**
PHPNuke	Linux/Windows, PHP, Apache, MySQL	****	****	***	****	***	**	*****
Plone	Linux/Windows, PHP, Apache, MySQL, Zope	**	*	***	***	**	****	***

Nome prodotto	Tecnologia	Installazione	configurazione	Interfaccia amministrazione	Moduli/funzioni aggiuntive	Personalizzazione grafica/layout	Gestione ruoli utenti	<i>Community</i>
Pmachine	Linux/Windows, PHP, Apache, MySQL	***	***	***	*	***	**	*
SmartPublish	Linux/Windows, ASP, IIS, MS Access	****	***	****	*	*	***	*
Typo3	Linux/Windows, PHP, Apache, MySQL	****	*	*****	****	***	***	**
Xaraya	Linux/Windows, PHP, Apache, MySQL	****	****	***	***	***	***	*
WebGui	Linux/Windows, Perl, Apache, MySQL	*	*	**	**	***	***	*
Zope	Linux/Windows, C-Python, ZODB, Zserver, Apache	*	*	***	*****	***	****	****

Webliografia

- ⇒ http://eu.conecta.it/paper/Economics_open_source.html: analisi della sostenibilità economica dei progetti *Open Source*.
- ⇒ <http://softwarelibero.it/altri/economia-sl.shtml>: un'economia del software libero: vantaggi e pericoli – articolo dell'Associazione Software Libero.
- ⇒ <http://sourceforge.net/>: il principale sito al mondo di raccolta degli innumerevoli progetti *Open Source* (il sito raccoglie codice e applicativi già compilati, e mette a disposizione molti servizi agli sviluppatori).
- ⇒ <http://www.apogeonline.com/openpress/libri/545/index.html>: Open Sources, voci dalla rivoluzione Open Source (raccolta di saggi scritti dai 'guru' del software *open source*).
- ⇒ <http://www.apogeonline.com/openpress/articoli/freeware2.txt>
- ⇒ www.fsf.org: il sito ufficiale della Free Software Foundation
- ⇒ www.hackmeeting.org: il sito dell'ultimo *meeting hacker*.
- ⇒ <http://linux.kuht.it/risorse/freelosofo/libro/>: “Codice libero, Richard Stallman e la crociata del software libero” (di Sam Williams).
- ⇒ <http://www.opensourcecms.com/index.php>: sito che offre la possibilità di testare online alcuni prodotti CMS *Open Source*.
- ⇒ http://www.openlabs.it/faq.php?myfaq=yes&id_cat=2&categories=Open+Source+%26+Free+Software: raccolta di FAQ sul software *Open Source*.
- ⇒ <http://www.opensource.org>: sito “istituzionale” del mondo *Open Source*.
- ⇒ <http://www.tuxedo.org/esr/writings/cathedral-bazaar>: documento-manifesto del mondo *Open Source*.
- ⇒ www.softwarelibero.it: sito dell'Associazione italiana free software.
- ⇒ www.linux.org: uno dei tanti siti dedicati a Linux.
- ⇒ www.s0ftpj.org: sito italiano di cultura hacker.
- ⇒ <http://www.annozero.org/nuovo/pages.php?page=etica+e+ragione>: il software libero come scelta etica e come scelta razionale.
- ⇒ http://www.steptwo.com.au/papers/kmc_evaluate/index.html: indicazioni utili su come valutare un CMS.
- ⇒ www.2600.com: sito di 2600, storica rivista americana di cultura hacker.

Bibliografia

- ⇒ “Internet come luogo di scelte”, Christian Albini – in “Aggiornamenti sociali” 01/2003.
- ⇒ “L’etica hacker e lo spirito dell’età dell’informazione”, Pekka Himanen – ed. Feltrinelli.
- ⇒ “L’era dell’accesso”, Jeremy Rifkin.
- ⇒ “Media activism”, Matteo Pasquinelli – ed. Derive Approdi – Bologna.
- ⇒ “Open Source, analisi di un movimento”, Nicola Bassi – ed. Apogeo.

Ringraziamenti

Grazie a l'e-teams per il supporto fornito nella “confezione” finale di questo lavoro e, soprattutto, grazie a mia moglie per la sua pazienza (e anche per aver desistito dall'inquietante proposito di prendere a martellate l'ignaro e incolpevole pc sul quale ho scritto questo libro).

**PUBBLICARE CONTENUTI IN INTERNET
CON LA TECNOLOGIA “*OPEN SOURCE*”**

Versione 1.0

E - T E A M S