

Tecnologie Open-source

Il progetto OpenBSD



Michele Marchetto

Indice generale

1.Introduzione.....	3
1.1.La Storia.....	3
1.2.Visione.....	3
1.3.Il mio contributo.....	4
2.Licensing.....	4
2.1.La licenza.....	4
2.2.Open documentation e BLOB.....	5
2.3.La disputa con Richard Stallman.....	5
3.Modello di business.....	6
3.1.Distribuzione del software.....	6
3.2.Donazioni e merchandise.....	6
3.3.La fondazione.....	7
3.4.Cancellazione dei fondi da parte del DARPA.....	8
4.Processi di sviluppo.....	8
4.1.Hackathon.....	10
5.Strumenti della comunità.....	11
5.1.Sito Web.....	11
5.2.Mailing List.....	11
5.3.ICB.....	12
5.4.CVS.....	12
5.5.diff.....	13
5.6.GNATS.....	13
5.7.Lint.....	13
6.Analisi della comunità.....	13
6.1.Leader.....	13
6.2.Comunicatività.....	14
6.3.Conferenze.....	14
7.Note.....	15

1. Introduzione

1.1. La Storia

OpenBSD vide la luce nel 1995 dopo che Theo de Raadt fu cacciato¹ dal progetto NetBSD di cui era uno dei membri fondatori. Questa decisione derivò dalla difficoltà che gli sviluppatori del core team avevano nel relazionarsi con de Raadt, famoso per essere arrogante e poco incline ad essere contraddetto.

Il primo commit, di fatto un import del tree di NetBSD nel repository di OpenBSD, è datato Venerdì 18 Ottobre 1995 alle ore 08:50:31 UTC.

La prima release, OpenBSD 1.2, uscì nel luglio 1996 seguita subito dopo dalla 2.0 rilasciata dell'ottobre dello stesso anno. Da allora il progetto segue un programma di rilasci ogni 6 mesi.

1.2. Visione

Gli obiettivi del progetto OpenBSD sono sempre stati estremamente chiari e dichiarati con forza. Fin dai primi tempi, ad imporre queste linee guida intransigenti è stato il leader Theo de Raadt e nel corso del tempo esse si sono rivelate vincenti ed hanno permesso a OpenBSD di ricavarci una fetta di mercato ben precisa e decisamente consistente.

Innanzitutto è necessario dire che, sebbene OpenBSD si presenti come un sistema operativo general purpose, e possa essere tranquillamente utilizzato come tale, è particolarmente indicato come soluzione per firewall e router. Infatti moltissimo impegno degli sviluppatori è dedicato ad affrontare e risolvere problematiche inerenti a queste realtà.

La visione e gli obiettivi del progetto, imposti da de Raadt possono essere riassunti in due punti fondamentali.

Sicurezza e qualità del codice

Il progetto fonda il proprio successo sulla sicurezza che è in grado di offrire ai propri utenti. A riprova di ciò lo slogan di OpenBSD è:

“Only two remote holes in the default install, in a heck of a long time!”

Ciò è dovuto allo stress che viene applicato dagli sviluppatori alla correttezza del codice ed alla qualità dello stesso. Infatti è visione comune all'interno del progetto che sicurezza ed alta qualità del codice siano concetti strettamente correlati. Inoltre la sicurezza è garantita da un serio impegno di audit continuativo del codice, e da tutta una serie di infrastrutture e meccanismi creati ad hoc per garantire anche laddove esistesse un baco di sicurezza di circoscrivere fortemente le possibilità di manovra dell'attaccante.

Openness

Questo punto può essere esaminato considerandolo da due prospettive diverse che verranno solo accennate qui e trattate in maniera estensiva nei capitoli appropriati.

Prima e più importante è quella filosofica. Si è sempre voluto creare un sistema operativo

completamente libero che non vincolasse l'utente in alcun modo. In quest'ottica anche la licenza GPL viene vista come troppo restrittiva. Questo punto è *fondamentale* nella visione del progetto.

Il secondo aspetto è strettamente legato all'obiettivo sicurezza. Si è sempre creduto (e non solo all'interno della comunità OpenBSD) che la sicurezza di un software sia accresciuta e comprovata dalla propria disponibilità ad essere esaminato *in tutte le proprie componenti* da chiunque desideri farlo.

Questo rende OpenBSD estremamente attento a tutto ciò che riguardi la licenza del software che incorpora.

1.3. Il mio contributo

Sono entrato a far parte (primo italiano) del core team nell'ottobre del 2006, occupandomi di protocolli di routing dinamico, e contribuendo alla scrittura di demoni quali OpenRIPD, OpenOSPF, OpenOSPFv6, OpenDVMRPD, OpenBGPD, OpenLDPD. Attualmente mi occupo dello stack di rete unicast e multicast, ed in particolar modo del supporto MPLS, di cui sono principale sviluppatore.

Inoltre faccio parte del gruppo utenti OpenBSD italiano² e sono organizzatore di OpenCON, l'unica manifestazione al mondo interamente dedicata a OpenBSD³.

2. Licensing

In questo capitolo si analizzeranno la politica adottata dal progetto in fatto di licenza e tutte le battaglie condotte negli anni per mantenere il prodotto più libero possibile.

2.1. La licenza

L'obiettivo, in fatto di licenza, che inizialmente si volle soddisfare e che viene perseguito tutt'oggi dal progetto è quello di mantenere pressoché invariato lo spirito del copyright dello Unix di Berkeley originario che permetteva una distribuzione completamente libera dei sorgenti. Per ottemperare a questi obiettivi è stato necessario introdurre una politica estremamente rigida per quanto riguarda l'inclusione di nuovo codice nel sistema base. Infatti sebbene esso presenti alcune piccole porzioni di software licenziato sotto GPLv2, è esplicitamente vietato dal progetto includere nuovo software che abbia una licenza troppo restrittiva come ad esempio la GPLv2. Tra le licenze accettate per le nuove inclusioni troviamo (ne cito solo alcune):

- ✓ BSD modificata in ogni sua forma (a 2 o 3 clausole)
- ✓ MIT
- ✓ ISC

Per quanto riguarda il nuovo software scritto direttamente dagli sviluppatori di OpenBSD, si predilige utilizzare la ISC che è funzionalmente equivalente ad una BSD modificata. A questo proposito esiste anche uno sforzo decisamente rilevante per rimpiazzare e spesso riscrivere qualsiasi software con licenza più restrittiva esista nel sistema base. Addirittura recentemente è stato importato un compilatore C licenziato sotto BSD, che possa (forse) prima o poi sostituire GCC (licenziato sotto GPL).

Questa attenzione particolare alla licenza è sempre stata un obiettivo fondamentale del progetto ma vide un acuirsi nel 2001. In quest'anno la licenza del packet filter compreso

allora in OpenBSD, IPFilter, fu lievemente cambiata. Questo scatenò una vera e propria campagna di ricerca e pulizia di tutto il codice presente all'interno di OpenBSD che non presentasse licenza appropriata. Questo avvenimento fu anche uno dei punti di svolta nella storia di OpenBSD in quanto costrinse gli sviluppatori a rimuovere IPFilter ed a progettare un rimpiazzo, PF, che a tutt'oggi è considerato uno dei maggiori punti di forza di OpenBSD.

Questa politica impedisce quindi di attingere dal vasto calderone di software GPL. Sarebbe impensabile però fare a meno di tutto il software GPL che esiste, poiché significherebbe di fatto rendere il sistema operativo impossibile da utilizzare con profitto. OpenBSD quindi mette a disposizione il “ports tree”, un repository di Makefile, che permettono in maniera automatizzata di scaricare, compilare e installare software di terze parti. In questo modo, pur non includendolo direttamente è possibile utilizzare software GPL su OpenBSD.

2.2. Open documentation e BLOB

Come abbiamo potuto vedere nel precedente capitolo è strettamente vietato includere software con licenza “restrittiva” nel sistema base. E' altresì vietato includervi software non libero e driver binari (blob), al contrario di quanto avviene su FreeBSD e GNU/Linux. Le motivazioni sono molte e molto semplici:

- ✓ Un vendor può abbandonare in qualsiasi momento il supporto per un driver binario;
- ✓ I driver binari non possono essere supportati dagli sviluppatori di OpenBSD;
- ✓ I driver binari non possono essere corretti dagli sviluppatori;
- ✓ I driver binari non possono essere migliorati dagli sviluppatori;
- ✓ I driver binari non possono essere garantiti sicuri;

In particolare l'ultimo punto è di importanza capitale. Sarebbe infatti impossibile garantire da parte della comunità OpenBSD l'assoluta sicurezza del proprio prodotto se questo includesse driver binari.

Inoltre non esiste alcun driver scritto da uno sviluppatore che abbia accettato un NDA. Questo per le stesse motivazioni di affidabilità e sicurezza.

A tal proposito il 2006 ha visto de Raadt impegnato in una massiccia campagna di “pressing” sui vendor affinché rilasciassero la documentazione relativa ai propri prodotti in modo che fosse possibile scrivere driver per essi. Alcuni si rifiutarono di farlo (e lo fanno tutt'oggi), costringendo gli sviluppatori di OpenBSD a scegliere tra driver binari e la rimozione completa del supporto. E' facile immaginare quale sia stata la scelta.

2.3. La disputa con Richard Stallman

Come detto OpenBSD include quello che viene chiamato il “ports tree” per l'installazione di software di terze parti. Questo repository comprende, in alcuni casi, anche la possibilità di installare software non libero. E' necessario precisare nuovamente però che questo software non è incluso nel repository, ma è possibile installarlo mediante una procedura automatizzata.

Nel 2007 Richard Stallman scrisse sulla mailing list di OpenBSD (misc@) che nel ports tree viene “suggerito” l'utilizzo di software non libero. Concluse quindi che non avrebbe consigliato l'utilizzo di OpenBSD ad alcuno. Ne nacque un thread⁴ estremamente ampio che vide l'apice nella risposta di Theo de Raadt:

“Richard, you are wrong. You said very clearly in your interview that the ports tree contains non-free software. It does not. It is just a scaffold of Makefiles containing URLs, and an occasional patch here or there. You are just plain wrong. And you are not enough of a man to admit that you are wrong. I may be unfriendly at times, but you are a power-misusing hypocritical liar who attacks projects that try harder than any others to only make free software available. Shame on you.”

Ad ogni release di OpenBSD viene pubblicata anche una canzone il cui testo riguarda i temi con cui gli sviluppatori hanno dovuto confrontarsi nello sviluppo di quella release. Nella release 4.3 la canzone associata tratta proprio di questo tema⁵. Vale sicuramente la pena di ascoltarla.

Per quanto mi riguarda, penso che le dichiarazioni di Stallman siano realmente ingiuste, in quanto, come abbiamo potuto estensivamente analizzare, OpenBSD è forse l'unico sistema operativo a contenere nel sistema base solo free software. Penso inoltre che Stallman dovrebbe dirigere le sue attenzioni su altri progetti che fanno un utilizzo *massivo* di software non libero e lo includono nelle loro release.

3. Modello di business

3.1. Distribuzione del software

E' possibile ottenere una release di OpenBSD principalmente in due modi. Il primo è di scaricare la ISO masterizzarla. Fino a poco tempo fa (un anno e mezzo) non veniva offerta una ISO scaricabile, ma solo tutte le componenti del sistema operativo. In questo modo un utente era costretto a costruirsi una ISO avviabile a mano. Questo (ovviabile attraverso UN comando) era una sorta di blando deterrente che avrebbe voluto incitare gli utenti a comprare i CD invece di scaricare la release dal sito. Dalla versione 4.2 è presente la ISO nel sito.

L'altra modalità di acquisizione è appunto, comprare il CD set.

3.2. Donazioni e merchandise

OpenBSD basa il proprio sostentamento sulla vendita dei CD set, del merchandise e sulle donazioni volontarie.

Il CD set (Figura 1) è il modo più semplice e diretto per finanziare il progetto. Il costo si aggira intorno ai 40 euro se ne viene acquistata una sola copia. Per gli utenti particolarmente affezionati ed entusiasti, o per chi vuole semplicemente dare un piccolo contributo in più esiste una ampia scelta di merchandise quali magliette e felpe tra cui scegliere.

Le donazioni monetarie volontarie tuttavia rappresentano il maggior introito del progetto, in quanto, oltre ad un ingente numero di “piccoli” donatori, anche aziende di grossissimo calibro quali Google Inc, Adobe, Mozilla e HP finanziano⁶ il progetto.

Le donazioni monetarie volontarie possono essere “generiche”, cioè al progetto nella sua interezza, oppure mirate a finanziare uno specifico aspetto verso cui si ha maggior interesse. In particolar modo accade spesso che in prossimità di un hackathon (di cui si parlerà in seguito) dedicato ad uno specifico argomento, giungano donazioni esplicitamente atte a finanziarlo.

Questa possibilità offerta dal progetto a mio modo di vedere è assolutamente vincente. Infatti oltre a far sentire gli utenti molto più partecipi all'interno della comunità, permette loro di agevolare lo sviluppo nei settori a cui sono più interessati, senza comunque che questo intacchi in alcun modo l'indipendenza del progetto.



Come ultima, ma estremamente importante modalità di donazione esiste quella in “hardware”. Accade

spesso che uno sviluppatore abbia la volontà di progettare qualche nuovo driver o supporto, ma che non possieda l'hardware necessario. Manifestando questa sua necessità attraverso il sito⁷, potrà ricevere donazioni in hardware dagli utenti interessati a quello specifico supporto.

Personalmente ho sempre considerato quest'ultima possibilità come assolutamente remota, nel senso che non confidavo assolutamente in questo genere di donazioni. Ho dovuto, tuttavia, ricredermi. Recentemente mi è capitato di aver bisogno di un componente hardware fuori dalle mie possibilità economiche, e sono quindi ricorso, con poca fiducia a questa possibilità. Ebbene nell'arco di 24 ore, mi sono giunte decine di offerte di hardware qualitativamente (e monetariamente) ben superiori a quello che avevo richiesto.

Per quanto riguarda gli introiti non espressamente indirizzati a qualcosa, essi vanno a coprire gli innumerevoli costi che il progetto deve sostenere. In primissimo luogo attraverso essi, Theo de Raadt riceve uno stipendio fisso. Secondariamente sono necessari per pagare la corrente elettrica, la connettività e altri costi relativi ai server di sviluppo ed ai vari mirror. In ultima istanza ma assolutamente fondamentale, per finanziare gli hackathon.

Il progetto OpenBSD rimane pur sempre una eterogenea comunità di volontari. La grandissima maggioranza degli sviluppatori così detti “core”, cioè quelli che dedicano parecchio tempo allo sviluppo, sono incitati dalle aziende per cui lavorano a farlo. Queste aziende vendono prodotti commerciali basati su OpenBSD ed in ultima analisi quindi, quando un loro dipendente sviluppa per OpenBSD sta di fatto lavorando per la propria azienda. Un esempio di questo modello è la tedesca Vantronix⁸.

3.3. La fondazione

Dal 2007 esiste la OpenBSD Foundation. Questa fondazione ha visto la luce per alcune necessità burocratiche nel sistema delle donazioni che si sono evidenziate nel corso degli anni. E' registrata come una “Canadian Federally Incorporated Non-Profit”, il cui obiettivo è finanziare tutti i progetti relativi a OpenBSD. Le donazioni un tempo avvenivano come “regali” a Theo de Raadt. Se per il singolo individuo questa modalità poteva anche andare bene per le aziende risultava difficile dal punto di vista legale motivare la spesa. Attraverso

la fondazione che è una entità legale a tutti gli effetti questi problemi sono stati ovviati.

3.4. Cancellazione dei fondi da parte del DARPA

Potrebbe apparire ovvio che alcuni donatori estremamente importanti possano “vincolare” la libertà della comunità OpenBSD minacciando il ritiro dei fondi. Questo è avvenuto già in passato e piuttosto che rinunciare alla propria indipendenza si è deciso di rinunciare ai fondi. Nel 2003 Theo de Raadt, in una intervista rilasciata ad un giornale di Toronto, espresse la propria contrarietà alla guerra in Iraq da parte degli Stati Uniti d'America. A seguito di questa esternazione il governo americano ritirò ingenti fondi al progetto POSSE, di cui faceva parte anche il progetto OpenBSD. Attraverso quei soldi era stato già stato organizzato un hackathon. Si decise comunque di non fare marcia indietro e grazie al lodevole contributo di utenti e simpatizzanti l'evento è stato rifinanziato e si è svolto tranquillamente.

4. Processi di sviluppo

“Twelve years ago OpenBSD developers started engineering a release process that has resulted in quality software being delivered on a consistent 6 month schedule – 25 times in a row, exactly on the date promised, and with no critical bugs. This on-time delivery process is very different from how corporations manage their product releases and much more in tune with how volunteer driven communities are supposed to function. Developer and testing lazyness is mostly circumvented and leader frustration is kept to a minimum.”

– Theo de Raadt

Potendo godere di una certa libertà nello sviluppo ed una quasi completa indipendenza di scelte progettuali e tempistiche correlate, la crescita di OpenBSD risulta svincolata da quelli che sono gli usuali processi di sviluppo. Nonostante questa libertà, si pone molta enfasi sulla puntualità delle release (una ogni sei mesi). Questo grazie ad una collaudata metodologia di sviluppo che si è dimostrata assolutamente vincente.

Il processo attraverso il quale è possibile fornire ogni sei mesi una release stabile del prodotto è abbastanza semplice. Esistono tre “versioni” di OpenBSD: la stable, la release e la current.

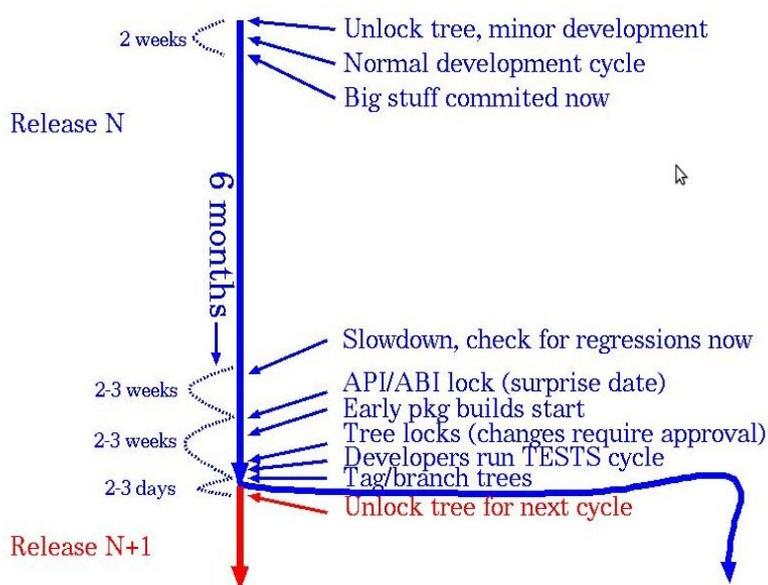


Figura 2: OpenBSD release calendar

La current è l'ultima versione disponibile, quella dove tutti gli sviluppatori inseriscono il proprio contributo e tutte le nuove funzionalità da loro ideate. Questa versione è in continuo mutamento. La release rappresenta una “immagine” della current in un preciso

istante di tempo ed è quella che andrà a costituire una certa release, appunto, di OpenBSD che possiamo trovare all'interno del CD. Infine esiste la stable, che altro non è che una release alla quale vengono applicate tutte le patch di sicurezza ritenute necessarie nel corso degli anni.

Naturalmente, per offrire un prodotto stabile ed affidabile, è necessario che il momento in cui viene presa una immagine della current per farne una release sia preceduto da un breve periodo preparatorio. Tre settimane prima viene annunciato da de Raadt quello che in gergo viene chiamato “soft lock”. Sta a significare che gli sviluppatori non potranno più applicare cambiamenti che modificano le API del sistema operativo. Lo sviluppo, comunque, continua

invariato. A questo punto vengono lanciati i build dei packages (software di terze parti) per tutte le architetture, e successivamente viene annunciato lock vero e proprio. In questo lasso di tempo che in genere dura una o al massimo due settimane non è possibile fare commit di alcun genere poiché questa fase viene dedicata solo al testing ed al bugfix. Non appena questo ciclo si conclude viene creato un nuovo branch nel CVS che rappresenterà la stable e avviene l'unlock dell'albero. Una nuova release di OpenBSD è nata. Nel periodo immediatamente successivo all'unlock avvengono i commit delle modifiche più importanti e sostanziose. Inoltre è in questo lasso di tempo che, generalmente hanno luogo gli hackathon. Questo perché avendo circa sei mesi di tempo prima della prossima release, le grosse modifiche e tutto il codice prodotto durante gli hackathon hanno la possibilità di essere testati e migliorati.



Figura 3: Sistemi di continuous integration

Eppure questo processo sarebbe del tutto insufficiente per creare software di qualità se fosse concluso qui, poiché due o tre settimane di test sebbene svolto da ogni singolo sviluppatore, sono decisamente poche.

La reale politica è quella di generare una release virtuale (chiamata snapshot) ogni giorno. Esistono infatti sistemi di continuous integration (Figura 3), che giornalmente compilano tutto il sistema per ogni architettura e ne producono una immagine. In questo modo è possibile stabilire nell'arco di poco tempo eventuali malfunzionamenti all'interno del prodotto. Inoltre questa immagine generata giornalmente viene fatta girare sui principali sistemi in gestione al progetto ivi compresi il server dove gira il CVS principale, e tutti i mirror sparsi nei cinque continenti.

In questa maniera è come se si producesse una release ogni giorno ed un gruppo rilevante di persone la testassero continuamente. Ne risulta quindi, che la release effettiva che andrà a costituire il contenuto dei CD, non è null'altro che una delle “tante” release prodotte, ma maggiormente testata.

Poiché la comunità dei developer è formata interamente da volontari, lo sviluppo (come d'altra parte quello di molti altri sistemi operativi liberi) non è strettamente coordinato né pianificato. Ogni individuo a seconda delle proprie esigenze lavorative o dei propri interessi si dedica a ciò che più lo attira. Questo approccio garantisce che gli sviluppatori abbiano notevole libertà di movimento e ne aumenta l'entusiasmo.

Ogni developer può fare commit sull'intero albero dei sorgenti, senza nessuna limitazione. E' necessario però (anche se si vuole modificare il proprio stesso codice), sottoporre la modifica ad altri sviluppatori e ricevere da questi l'assenso. Questo garantisce che tutte i cambiamenti siano visionati e possibilmente testati da più di una persona.

Non esiste alcuna infrastruttura per la visione/revisione delle modifiche proposte. Lo sviluppatore interessato crea delle patch mediante il comando diff e le invia per email agli sviluppatori che ritiene interessati, i quali le visionano ed eventualmente le approvano.

Ogni sviluppatore è strettamente responsabile del codice che scrive e ne detiene il copyright.

4.1. Hackathon

Gli "hackathon"⁹ sono dei raduni riservati agli sviluppatori e finanziati interamente dal progetto. Lo scopo esplicito di questi incontri è lo sviluppo cooperativo. Si svolgono in una località sempre diversa nei cinque continenti per permettere a tutti gli sviluppatori di partecipare. Il progetto OpenBSD pone grossissima importanza negli hackathon. Infatti essi velocizzano notevolmente il processo di sviluppo ordinario avendo a disposizione un numero ingente di sviluppatori a tempo pieno per una settimana. Inoltre le idee migliori si manifestano spesso durante questi raduni quando parecchie menti pensano allo stesso problema.

A parte il contenuto prettamente tecnico di questi incontri l'aspetto che non deve essere trascurato durante gli hackathon è fare amicizia. Potrà sembrare una cosa banale o di poco conto ma riveste al contrario, un ruolo primario. Innanzi tutto permette agli sviluppatori appena entrati nel progetto di conoscere i veterani ed entrare subito in confidenza con le dinamiche di sviluppo, rompendo gli indugi iniziali che potrebbero manifestarsi. In secondo luogo conoscere le persone assieme alle quali si sviluppa durante tutto l'anno crea una affinità che altrimenti sarebbe difficile da raggiungere e questo aumenta notevolmente la sinergia nello sviluppo.

In genere (ma esistono eccezioni!) gli hackathon si tengono appena dopo l'unlock del tree per la release. Questo perché essendo una settimana di programmazione massiva che vede coinvolti innumerevoli sviluppatori, dove spesso vengono introdotte numerose nuove feature, è importante permettere a queste ultime di essere testate dalla comunità prima che vadano a costituire la release.

Durante l'arco di un anno vengono organizzati due, massimo tre hackathon. Il così detto "common hackathon" si svolge tra maggio e giugno in Canada e non ha un focus particolare. Tutti gli sviluppatori sono invitati a parteciparvi. Gli altri hackathon organizzati sono focalizzati su un singolo argomento che può variare: stack di rete, filesystem, device drivers. Sono invitati solo gli sviluppatori competenti rispetto a quell'argomento.

Come ultimo aspetto, marginale ma non trascurabile, è necessario sottolineare che ad ogni hackathon viene stampata una maglietta⁹ e regalata agli sviluppatori partecipanti. Il gesto in sé non ha alcuna valenza ma accumulare un ingente numero di magliette e portarle orgogliosamente fa sentire gli sviluppatori parte del progetto, e contribuisce ad amalgamare la comunità.

5. Strumenti della comunità

La comunità degli sviluppatori che partecipa al progetto oscilla dai 100 ai 130 individui. La comunità degli utenti invece è difficile da quantificare, ma utilizzando come indicazione il numero di iscritti alle mailing list il numero si attesta nell'ordine delle decine di migliaia. Per poter coordinare lo sviluppo di un così ingente numero di persone, mantenere aggiornata la comunità degli utenti e permettere ad essa di interagire attivamente con il progetto, vengono messi a disposizione alcuni strumenti di information management e development management.

5.1. Sito Web

Sicuramente la più importante “vetrina” per un qualsiasi progetto, sia esso opensource o meno, è il sito web. Da esso si possono evincere moltissime informazioni sulla natura del progetto e sulla politica dello stesso. Non a caso infatti il sito di OpenBSD si presenta ben organizzato ma comunque estremamente semplice. Contiene tutte le informazioni necessarie, ma presentate senza fronzoli o grafiche particolarmente accattivanti. Il testo è il protagonista principale del sito. Questo è assolutamente in sintonia con quello che OpenBSD è. Non esistono aree wiki o qualsiasi strumento di interazione bidirezionale con l'utente, in quanto si prediligono altri mezzi (quali le mailing list) per la raccolta di informazioni. Ciò che OpenBSD cura con estrema attenzione sono le FAQ che coprono realmente ed estensivamente le problematiche più frequenti. Per qualsiasi altro problema è necessario fare riferimento alle mailing list e relativi archivi.

Tutti i sorgenti del sito web sono sotto controllo di versione CVS e ogni sviluppatore è tenuto ad aggiornarne le parti di propria competenza, ivi comprese le slide delle presentazioni che vengono tenute durante le conferenze. Nel server principale gira uno script che ciclicamente fa il checkout dei sorgenti del sito dal CVS e li immette nel webserver.

5.2. Mailing List

Come in qualsiasi progetto opensource le mailing list ricoprono un ruolo essenziale. Non prenderò in considerazione qui la comunità che le popola, poiché essendo molto particolare, preferisco analizzarla approfonditamente nel capitolo 6.

Il progetto prevede sei mailing list pubbliche. Analizziamole quindi brevemente dal punto di vista tecnico.

- ✓ `misc@openbsd.org`
la mailing list più frequentata e “chiassosa”. Non possiede un argomento dominante a cui attenersi, tratta di tutto ciò sia inerente a OpenBSD od alla sua comunità. Possiamo trovarvi discussioni blandamente tecniche, problemi di utilizzo, consigli da parte degli utenti, pubblicizzazione di eventi, etc. A questa mailing list è iscritto, in genere, un utilizzatore generico, oppure semplicemente un curioso di OpenBSD.

- ✓ `tech@openbsd.org`
questa mailing list prevede argomenti prettamente tecnici. Qui vengono inserite discussioni legate a problemi complessi, proposte patch di utenti e sviluppatori che necessitano un test intensivo. A questa mailing list è in genere iscritto l'utilizzatore esperto di OpenBSD.

- ✓ `ports@openbsd.org`
in questa mailing list vengono portate avanti discussioni che riguardano il repository dei ports. Qui è possibile richiedere chiarimenti e proporre nuovi ports da parte degli utenti.

- ✓ `source-changes@openbsd.org`
questa mailing list raccoglie tutti i log dei commit fatti dagli sviluppatori. Talvolta contiene anche brevissime discussioni o domande riguardanti una singola modifica. E' uno strumento di importanza inestimabile per gli sviluppatori che possono comodamente essere informati su tutto ciò che succede all'albero dei sorgenti, ed anche per gli utenti esperti che possono rimanere aggiornati sulla direzione che sta prendendo il progetto, e sui nuovi supporti introdotti.

- ✓ `ports-changes@openbsd.org`
esattamente come `source-changes`, soltanto che si applica all'albero dei ports.

- ✓ `bugs@openbsd.org`
questa mailing list contiene tutti i bug report sottoposti attraverso GNATS da parte degli utenti e relative comunicazioni da parte degli sviluppatori.

Esiste poi una mailing list privata, a cui hanno accesso solo gli sviluppatori in cui si dibatte di questioni interne al progetto. Qui le questioni tecniche la fanno da padrone e sono spesso molto lunghe e particolarmente accese, ma non mancano discussioni sull'organizzazione degli hackathon e problematiche di stampo “umano”.

5.3. ICB

ICB è un protocollo, antenato del più moderno IRC, che permette il dialogo in tempo reale tra i partecipanti. OpenBSD mette a disposizione degli sviluppatori un server ICB dove essi possano discutere dei problemi tecnici che stanno affrontando. E' un comodo strumento per ottenere responsi più veloci e partecipativi di quanto non possa avvenire via email, sulle questioni poste.

Il fatto che venga utilizzato il vetusto ICB invece di IRC è semplicemente per questioni storiche. Negli anni 90 il progetto ha adottato questo protocollo e fino adesso non si sono presentate sufficienti motivazioni per migrare, visto comunque il numero limitato di persone che possono accedervi.

5.4. CVS

CVS è il cuore dello sviluppo. Permette agli sviluppatori di mantenere aggiornati i propri sorgenti ed agli utenti di ottenere l'ultimissima current od una stable in maniera facile e veloce.

La politica di OpenBSD è di mantenere quanto più possibile lineare lo sviluppo e conseguentemente anche il tree CVS che (come visto in precedenza) contiene una sola linea principale di sviluppo (current) ed a intervalli fissi viene creato un nuovo branch (stable).

Al contrario di alcuni altri progetti opensource (come ad esempio il kernel di Linux), non esistono branch di sviluppo particolari. Tutti gli sviluppatori inseriscono le proprie modifiche su current senza alcuna limitazione.

Ogni qualvolta un developer esegue un commit viene generata una email contenente il log della modifica ed inviata a source-changes.

Il sito mette a disposizione la possibilità di navigare nei sorgenti del progetto attraverso CVSweb.

5.5. *diff*

Questo semplice strumento è alla base del processo di sviluppo di OpenBSD. Infatti diff permette la creazione di patch che possono essere consultate dagli sviluppatori e applicate (mediante patch(1)) ai propri sorgenti. Queste patch non sono altro che file di testo e forniscono una idea immediata delle modifiche proposte

5.6. *GNATS*

GNATS è un insieme di strumenti attraverso il quale la comunità degli utenti può segnalare eventuali bug riscontrati agli sviluppatori. Ogni bug report compilato da un utente arriva in copia alla mailing list bugs@openbsd.org, assieme ad ogni comunicazione da parte degli sviluppatori. Permette di gestire i bug assegnandoli allo sviluppatore competente e gestendone gli stati da aperto, analizzato, chiuso. GNATS permette anche di proporre modifiche alla comunità degli sviluppatori.

5.7. *Lint*

Gli strumenti di controllo automatico della qualità del codice non sono tenuti in grande considerazione all'interno della comunità OpenBSD, in quanto si crede fermamente che la qualità del codice sia ad esclusivo appannaggio dello sviluppatore. L'unico strumento che viene talvolta usato è Lint. Questo strumento controlla il codice sorgente in cerca degli errori più comuni. Si tratta comunque di un utilizzo del tutto marginale.

6. Analisi della comunità

L'analisi della comunità di OpenBSD è un aspetto estremamente controverso. Il modo di rapportarsi con gli utenti da parte degli sviluppatori è spesso stato aspramente criticato e non sempre a torto.

Considerata la completa indipendenza economica che OpenBSD mira a mantenere, sarebbe del tutto naturale affidarsi alla comunità degli utenti per incrementare la propria popolarità e quella del progetto. Vedremo che persistono alcuni ostacoli comunicativi tra OpenBSD ed i suoi utenti che rende difficile questo rapporto.

6.1. *Leader*

E' il leader che fornisce le linee guida che un progetto deve seguire e che mantiene il dialogo con gli utilizzatori e con le altre comunità opensource.

Per quanto riguarda OpenBSD questo è particolarmente vero. Gli sviluppatori sono *fortemente* influenzati dalle decisioni ed opinioni di Theo de Raadt e dal suo modo di fare. Egli dirige non soltanto lo sviluppo ma anche la maniera di confrontarsi con la comunità

degli utilizzatori e con le altre realtà opensource. E' necessario comprendere questo aspetto per capire il progetto nella propria interezza.

De Raadt è una persona estremamente intelligente e brillante. Risulta quindi notevolmente difficile non venirme influenzati. Tuttavia i suoi modi spesso bruschi ed arroganti lo portano ad avere violenti scontri con le altre comunità, ed in particolar modo con Richard Stallman. Lo stesso Linus Torvalds lo ha etichettato come una persona “difficile”.

Personalmente ritengo comunque che questa realtà “leader-centrica”, sia positiva. E' necessario per la buona salute di un progetto avere un leader estremamente preparato e deciso, che influenzi con la propria personalità lo sviluppo ed affascini le comunità.

6.2. Comunicatività

La comunità è improntata su una intransigenza abbastanza spinta, influenzata da de Raadt, e comunicare attraverso le mailing list o con gli sviluppatori è talvolta difficile. Spesso si ricevono risposte inutilmente sgarbate o saccenti, anche se il contenuto tecnico di queste è sempre molto alto. Bisogna sottolineare però che i contributi in termini di codice sono sempre tenuti in altissima considerazione.

Ne consegue che un individuo interessato a contribuire o semplicemente ad utilizzare OpenBSD deve essere spinto da una *fortissima* motivazione e non scoraggiarsi di fronte alle prime difficoltà od alle prime risposte sgarbate. Questo sicuramente influisce negativamente su moltissimi contributi che un comportamento un po' più affabile potrebbe portare, ma d'altra parte “filtra” gli stessi ed opera una selezione naturale che porta solo le persone tecnicamente e moralmente motivate allo status di sviluppatore. La comunità degli sviluppatori ufficiali è conseguentemente abbastanza ristretta rispetto a quella di altri sistemi operativi opensource, ma il tasso di motivazione e skill è mediamente più alto.

Questo particolare modo di comunicare assieme alle prese di posizione piuttosto spinte rispetto a questioni ideologiche penso siano alla base della minor diffusione di OpenBSD rispetto ad esempio a FreeBSD, nonostante la qualità del software sia notevolmente migliore.

6.3. Conferenze

Nonostante quello che si è detto in precedenza, gli sviluppatori rimangono pur sempre persone, e come tali, incontrate dal vivo si rivelano nettamente differenti da come possono apparire per via telematica. Lo stesso de Raadt tacciato come arrogante e prepotente, incontrato dal vivo è gentile ed affabile.

Questo aspetto collegato al fatto che gli sviluppatori di OpenBSD sono sparsi nei cinque continenti è di primaria importanza poiché si ha la possibilità di presenziare e pubblicizzare il prodotto ad ogni conferenza opensource si svolga nel pianeta. Per gli utenti incontrare gli sviluppatori (magari in maniera totalmente informale) è molto importante poiché amalgama la comunità e dipana quei problemi di comunicazione che abbiamo analizzato in precedenza. Tutti gli sviluppatori sono *fortemente* incoraggiati a partecipare al maggior numero di conferenze possibile.

Il progetto inoltre finanzia tutte quelle conferenze che danno particolare spazio a OpenBSD, come ad esempio OpenCON.

7. Note

1. <http://mail-index.netbsd.org/netbsd-users/1994/12/23/0000.html>
2. <http://www.openbeer.it>
3. <http://www.opencon.org>
4. <http://marc.info/?l=openbsd-misc&m=119730630513821&w=2>
5. <http://www.openbsd.org/lyrics.html#43>
6. <http://www.openbsd.org/donations.html>
7. <http://www.openbsd.org/want.html>
8. <http://www.vantronix.com>
9. <http://www.openbsd.org/hackathons.html>
10. <http://www.openbsd.org/cgi-bin/cvsweb/>