

Il progetto Debian

Capitolo 1

1. Il progetto Debian	pag. 1
1. Cosa è debian?	pag. 2
1. Un sistema operativo multi piattaforma	pag. 2
2. La Qualità del Free Software	pag. 4
3. L'ente giuridico: organizzazione non-profit	pag. 4
2. I Documenti Fondanti	pag. 5
1. L'impegno nei confronti degli utenti	pag. 5
2. Le Linee Guida di Debian Free Software	pag. 7
3. Gli incarichi interni del Progetto Debian	pag. 9
1. Gli sviluppatori di Debian	pag. 9
2. Il ruolo attivo degli utenti	pag. 14
1. Come effettuare il report dei bugs	pag. 14
2. Traduzioni e documentazione	pag. 15
3. Come condividere una fix [riparazione]	pag. 15
4. Altri modi per contribuire	pag. 17
3. Teams e Sotto-Progetti	pag. 18
1. Gli attuali Sotto-Progetti Debian	pag. 18
2. Teams di Amministrazione	pag. 19
3. Teams di Sviluppo, Schieramenti Trasversali dei Teams	pag. 21
4. Come seguire gli aggiornamenti di Debian	pag. 21
5. Il ruolo delle distribuzioni	pag. 23
1. Il programma che installa Debian: debian-installer	pag. 23
2. La libreria Software	pag. 23
6. Il ciclo di vita di un rilascio	pag. 24
1. Lo Status di Experimental (condizione)	pag. 24
2. Lo Status di Unstable (condizione)	pag. 24
3. Il trasferimento (migrazione) a Testing	pag. 26
4. L'avanzamento (conversione) da Testing a Stable	pag. 27
5. Gli Status (condizioni) di Oldstable e di Oldoldstable	pag. 31

<< Prima di tuffarvi immediatamente nel mondo della tecnologia, consentiteci un breve sguardo su cosa è il progetto Debian, sui suoi obiettivi, sui suoi mezzi ed infine sulle sue attività. >>

1.1 Cosa è Debian?

CULTURA Origini del suo nome	Non vi scervellate. Debian non è un acronimo. Questo nome è infatti la contrazione di due nomi di persona, ossia del compianto fondatore del Progetto Debian Ian Murdock e della sua ragazza dell'epoca, Debra. Debra + Ian = Debian
--	---

Debian è una distribuzione GNU/Linux. Tratteremo cos'è una distribuzione in maniera più dettagliata nel paragrafo 1.5 “Il ruolo delle distribuzioni” a pagina 23, ma per ora preferiamo limitarci a dire che è un sistema operativo completo, che comprende software ed applicazioni per l'installazione e la gestione, interamente basati su kernel linux e su free software (specialmente quelli provenienti dal Progetto GNU).

Quando è stato creato Debian, nel 1993, sotto la supervisione di FSF, Ian Murdock aveva chiari i suoi obiettivi, al punto da fissarli nel Manifesto di Debian. Il sistema operativo libero che sognava avrebbe dovuto avere due caratteristiche principali. Innanzitutto, la qualità: Debian sarebbe stato sviluppato con la migliore diligenza, degna del kernel linux. Sarebbe stato anche una distribuzione non commerciale, sufficientemente credibile per competere con le migliori distribuzioni commerciali. Questa duplice ambizione, ai suoi occhi, poteva essere soddisfatta soltanto rendendo accessibile il modello di sviluppo di Debian come quello di Linux e del progetto GNU. In quanto, una revisione paritaria avrebbe ininterrottamente migliorato il risultato. [La peer review o revisione paritaria, a grandi linee, è una revisione di un prodotto di una o più persone da parte di terzi con competenze simili a quelle dei “produttori” (generalmente i terzi sono dei colleghi, infatti “peer” in inglese significa “collega”). La revisione paritaria viene solitamente utilizzata per mantenere alti standard di qualità, per migliorare le prestazioni e per ottenere credibilità. Il development process o modello di sviluppo, molto genericamente, è un modello teorico, realizzato durante la fase ex ante dello sviluppo di un software, attraverso il quale sono suddivise le varie fasi di progettazione del software al fine di ottimizzarle, nonché fissati gli obiettivi da perseguire. È noto anche come “software development life cycle (SDLC)”).]

CULTURA Gnu, il progetto di FSF	Il progetto GNU è una linea di free software, sviluppata o sponsorizzata dalla Free Software Foundation (FSF), fondata dal suo rappresentativo leader Dr. Richard M. Stallman. GNU è l'acronimo permanentemente ricorsivo di “GNU is Not Unix” (GNU non è Unix).
---	--

CULTURA Richard Stallman	Fondatore di FSF e autore della licenza GPL, Richard M. Stallman (spesso menzionato solo attraverso le sue iniziali RMS) è un carismatico leader del movimento Free Software. A causa delle sue posizioni intransigenti, non è unanimemente ammirato, ma i suoi contributi non tecnici al Free Software (in particolare sotto la sfera legale e filosofica) sono rispettati da tutti
------------------------------------	--

1.1.1. Un sistema operativo multi piattaforma

COMUNITÀ L'avventura di Ian Murdock	Ian Murdock fu il fondatore ed il primo leader del Progetto Debian, dal 1993 al 1996. Dopo aver lasciato il comando a Bruce Perens, Ian prese un incarico di minor visibilità. Ritornò a lavorare dal dietro le quinte della comunità free software, creando la compagnia Progeny, con l'intenzione di realizzare una distribuzione commerciale derivata da Debian. Questa sua nuova scommessa fu, sfortunatamente, un fiasco commerciale e lo sviluppo fu abbandonato. La compagnia, dopo diversi anni di insolvenza, nelle vesti di fornitore di servizi (service provider), chiuse per fallimento nell'Aprile del 2007. Dei vari progetti cominciati da Progeny, è rimasto in piedi solo discover. Uno strumento che rileva automaticamente le specifiche hardware (automatic hardware detection tool). Ian Murdock è deceduto il 28 Dicembre 2015 a San Francisco dopo aver pubblicato diversi tweet inquietanti che denunciavano abusi e violenze subiti dalla polizia. Il caso della morte di Ian Murdock è stato archiviato nel 2016 (luglio) come suicidio.
---	---

Debian, rimanendo salda ai suoi originari principi, ha avuto così tanto successo che ha raggiunto una dimensione enorme. Debian supporta ufficialmente ben 10 architetture hardware ed anche altri kernels, tra cui FreeBSD (sebbene i ports relativi a FreeBSD non fanno ancora parte della raccolta delle architetture ufficialmente supportate). Inoltre, con più di 28,000 pacchetti sorgente, il software distribuito soddisfa qualsiasi esigenza, a casa o in un'impresa.

[Il termine inglese “Ports” in italiano significa letteralmente “porti”. Un aneddoto per una maggiore comprensione potrebbe essere immaginare un “porto di rilascio”, che altri non è che un porto dove una nave fa scalo solo per permettere il riposo dell’equipaggio o per rifornirsi di viveri, combustibile, ecc.. Leggendo i capitoli successivi vi sarà chiaro il suddetto riferimento.]

Rilasciare l’intera distribuzione in un unico formato sarebbe stato inopportuno: sarebbe stato davvero eccessivo distribuire 16 DVD-ROMs per installare una versione completa su un PC standard ... Per questo Debian è stata sempre più congegnata come una “meta-distribuzione”, dalla quale si possono estrarre distribuzioni mirate, realizzate per una categoria d’utenza definita: Debian Science per scopi scientifici, Debian Edu per un uso educazionale e pedagogico nell’ambiente scolastico-accademico-universitario, Debian Med per servizi medico-sanitari-ospedalieri, Debian Jr per i bambini, ecc. Potrete trovare una lista più completa dei sotto-progetti nel paragrafo dedicato a questo argomento, 1.3.3.1 Gli attuali Sotto-Progetti Debian, pagina 18.

I suddetti progetti dedicati sono organizzati in una struttura ben definita, così da garantire senza problemi la compatibilità tra le varie “sotto-distribuzioni”. Tutte loro seguono difatti la pianificazione generale per il rilascio delle nuove versioni. E dal momento che hanno un’unica origine, possono essere facilmente ampliate, integrate e personalizzate con le applicazioni disponibili nei repositories Debian.

Tutti gli strumenti di Debian contribuiscono a tale fine: `debian-cd` ha da tempo consentito la creazione di una collana di CD-ROMs, che racchiude solo una pre-selezionata serie di pacchetti; `debian-installer` è uno strumento di installazione modulare, senza dubbio adatto a peculiari esigenze. APT installa pacchetti di diversa origine e nel contempo garantisce la totale solidità del sistema.

STRUMENTI TOOLS

La creazione di
un CD-ROM
Debian

`debian-cd` crea immagini ISO di installazione per dispositivi media (CD, DVD, Blu-Ray, ecc.) pronti per l’uso. Qualsiasi argomento che prende in considerazione questo software è trattato (in inglese) nella mailing list `debian-cd@lists.debian.org`. Il team è diretto da Steve McIntyre che si occupa delle builds ISO ufficiali di Debian.
[Build in inglese significa letteralmente struttura.]

BASILARE

Ad ogni computer
la sua
architettura

Il termine “architettura” denota un modello di computer (i modelli di computer più noti sono Mac o PC). Ogni architettura si differenzia fondamentalmente per il suo processore, comunemente incompatibile con altri. Queste differenze di hardware implicano differenti metodi di funzionamento, per cui è necessario che il software sia compilato specificamente per ogni architettura.

La maggior parte del software disponibile in Debian è scritto in portable programming languages [trad. non lett. un linguaggio di programmazione portable – ovvero un linguaggio di programmazione per lo sviluppo software compatibile con diversi sistemi informatici. Ad esempio, C ++ è un linguaggio portable]: lo stesso codice sorgente può essere compilato per diverse architetture. In realtà un codice binario eseguibile, compilato spesso per una specifica architettura, generalmente non funzionerà su altre architetture.

Ricordatevi che ciascun programma è stato creato scrivendo un codice sorgente; il codice sorgente è un file di testo composto di istruzioni in un dato linguaggio di programmazione. Prima che voi possiate usare il software, è necessario compilare il codice sorgente, il che significa che occorre trasformare il codice in codice binario (una serie di istruzioni macchina eseguibili dal processore). Ciascun linguaggio di programmazione ha uno specifico compilatore per eseguire queste operazioni (per esempio `gcc` per il linguaggio di programmazione C).

STRUMENTI TOOLS Il programma di installazione: Installer	debian-installer è il nome del programma di installazione. La sua concezione modulare ne consente l'impiego nella vasta serie di scenari di installazione. L'attività di sviluppo è coordinata su debian-boot@lists.debian.org , mailing list sotto la direzione di Cyril Brulebois.
---	--

1.1.2. La Qualità del Free Software

Debian segue tutti i principi del Free Software e le sue nuove versioni non sono rilasciate fino a quando non sono pronte. Gli sviluppatori non sono costretti da qualche schema inflessibile a rispettare precipitosamente una data di scadenza prefissata. Le persone si lamentano frequentemente del lungo tempo trascorso fra i rilasci di Debian Stable, ma questa precauzione garantisce anche la sua leggendaria sicurezza: lunghi mesi di test sono ritenuti necessari affinché l'intera distribuzione riceva il soprannome di “stable”.

Debian non transigerà mai sulla sua qualità: tutti i bugs critici noti devono essere risolti in ogni nuova versione, anche qualora questo richieda rimandare la data prefissata di rilascio. I pacchetti opzionali che non soddisfano i requisiti di qualità verranno semplicemente rimossi dalla release stable.

1.1.3. L'ente giuridico: organizzazione non-profit

Legalmente parlando, Debian è un progetto gestito da un'associazione di volontari no-profit statunitense. Il progetto annovera circa mille sviluppatori Debian, ma contribuisce nel contempo all'incontro di un numero di gran lunga superiore di contributori (traduttori, segnalatori di bugs, artisti, sviluppatori occasionali, ecc.).

Per portare a termine la sua missione, Debian ha una vasta infrastruttura con molti servers connessi attraverso internet, offerti ed ospitati da molti sponsors.

COMUNITÀ Alle spalle di Debian l'associazione SPI e le piccole organizzazioni.	Debian non ha intestato alcun server a proprio nome, essendo questi tra l'altro un progetto interno dell'associazione Software in the Public Interest (SPI), che gestisce l'hardware e gli aspetti finanziari (donazioni, l'acquisto di hardware, ecc.). Questa associazione, sebbene inizialmente fu creata esclusivamente per Debian, ora gestisce altri progetti free software, tra cui in particolar modo PostgreSQL (database), freedesktop.org (progetto per la standardizzazione di vari contenuti del moderno ambiente grafico desktop, come gli stessi GNOME e KDE) e la collezione di programmi di video-scrittura per ufficio denominata Libre Office. ♦ https://www.spi-inc.org/ Oltre all'associazione SPI, altre diverse piccole associazioni collaborano a stretto contatto con Debian allo scopo di raccogliere fondi, pur non centralizzando ogni cosa negli Stati Uniti: sono conosciute nel gergo di Debian come “Trusted Organizations” (organizzazioni certificate). Questa manovra evita costi proibitivi di trasferimento internazionale di denaro (o dazi doganali) e ben si confa' alla natura decentralizzata del progetto. Non esitate ad iscrivervi alle associazioni più vicine ed a sostenere il progetto! ♦ La lista (https://wiki.debian.org/Teams/Auditor/Organizations/) ♦ Debian France (https://france.debian.net/) ♦ debian.ch (https://debian.ch)
--	--

1.2 I Documenti Fondanti

A pochi anni dal suo lancio, Debian formalizzò i principi che avrebbe dovuto seguire nelle vesti di progetto free software. Questa ponderata decisione gestionale (militante) ha consentito una crescita ordinata e pacifica, garantendo allo stesso tempo che tutti i membri procedano nella stessa direzione. Per diventare uno sviluppatore Debian ogni candidato deve accettare e dimostrare il proprio sostegno ed adesione ai principi stabiliti nei Documenti Fondanti del progetto.

Il modello di sviluppo è costantemente oggetto di discussione, ma i Documenti Fondanti sono largamente e consensualmente sostenuti, per questo raramente ci sono dei cambiamenti. La costituzione di Debian offre anche altre garanzie per la loro stabilità: è richiesta una maggioranza qualificata di tre-quarti per approvare ogni emendamento.

1.2.1 L'impegno nei confronti degli utenti

Il progetto ha anche un “contratto sociale”. Che importanza può avere un documento simile in un progetto destinato solo allo sviluppo di un sistema operativo? Ciò è piuttosto semplice: Debian lavora per i suoi utenti e, di conseguenza, per la società. Questo contratto riassume gli impegni che il progetto ha assunto. Consentiteci il loro esame per filo e per segno:

1. Debian rimarrà 100% free [libero].

Questo è il primo articolo. Debian è e resterà composto interamente ed esclusivamente da free software. Per di più, tutto il software sviluppato all'interno del progetto Debian sarà, medesimamente, libero.

IN PROSPETTIVA

Alle spalle del
software

La prima versione del Contratto Sociale di Debian riportava “Debian rimarrà 100% Free Software”. La cancellazione di quest'ultimo termine (avvenuta con la ratificazione della versione 1.1 del contratto nell'Aprile 2004) denota la volontà di conseguire la libertà non solo nel software, ma anche nella documentazione ed in ogni altro documento che Debian desidera fornire con il suo sistema operativo. Questo cambiamento, che era destinato solo all'editoria, ha avuto, in effetti, numerose conseguenze, in particolare la rimozione di qualche dubbia documentazione. Inoltre, il crescente uso di firmware nei driver reca problematiche: molti sono non-free e sono ancora indispensabili per il corretto funzionamento del relativo hardware.

2. Restituiremo alla comunità free software.

[da intendersi elasticamente: qualsiasi contributo della comunità sarà restituito migliorato (o, semplicemente, ricompensato)]

Ogni miglioria realizzata dal progetto Debian ad un'opera integrata nella distruzione è restituita all'autore dell'opera stessa (definito upstream - a monte). Generalmente Debian preferisce cooperare con la comunità piuttosto che produrre da sola.

COMUNITÀ Autore Upstream o sviluppatore di Debian	<p>Il termine “autore upstream” (autore a monte) è assegnato agli autori e/o sviluppatori che hanno scritto e prodotto un’opera. D’altro canto, uno “sviluppatore di Debian” (il termine “debian maintainer” - manutentore debian - è più appropriato) utilizza un’opera esistente per poi riprodurla in un pacchetto Debian. La differenza fra i due ruoli non è sempre nettamente definita. Il debian maintainer può scrivere una patch, della quale potranno beneficiare tutti gli utenti dell’opera. [In inglese patch significa letteralmente toppa.] Solitamente Debian incoraggia gli sviluppatori al comando di un pacchetto coinvolgendoli anche nello sviluppo upstream (divenendo loro stessi contributori senza essere relegati nel ruolo di semplici utenti di un programma).</p>
---	--

3. Non nasconderemo i problemi

Debian non è perfetto e troveremo nuovi problemi da risolvere ogni giorno. Debian manterrà il bug report database sempre disponibile alla consultazione degli utenti. Le segnalazioni che gli utenti registrano online diverranno rapidamente visibili alle altre persone.

4. Le nostre priorità sono i nostri utenti ed il free software.

Questo impegno è più difficile da definire. Debian impone, quando deve essere presa una decisione, un orientamento che escluda una decisione facile per gli sviluppatori, ma che potrebbe inficiare l’user experience (l’esperienza utente), a favore di una soluzione più raffinata, anche se è più difficile da implementare. Questo significa fare propri, come priorità, gli interessi degli utenti e del free software.

[“L’esperienza utente” è una espressione entrata anche nel gergo comune italiano anche se in questo caso per esperienza non si intende una mera conoscenza bensì si riferisce alla praticità di Debian per gli utenti]

5. Le opere che non corrispondono ai nostri standards free software.

Debian accetta e comprende che gli utenti potrebbero voler utilizzare qualche programma non-free. Per questo il progetto consente che alcune componenti della sua infrastruttura siano dedicate alla distribuzione dei pacchetti Debian di software non-free che possono essere poi redistribuiti in sicurezza.

COMUNITÀ A favore o contro la sezione non- free?	<p>L’impegno a mantenere una divisione per l’adattamento del software non- free (la sezione non-free è meglio chiarita nella casella di testo “Gli archivi Main [principale], Contrib e Non-free” a pag. 109) è di frequente oggetto di dibattito nella comunità Debian. I detrattori argomentano che allontana l’utenza dai programmi free software equivalenti [nelle funzionalità a quelli proprietario] e contraddice la massima di servire solo la causa free software. I sostenitori con flemma asseriscono che più di un pacchetto non-free è “quasi-free” se non per una o due seccanti restrizioni (la più comune è il divieto dell’impiego del software per scopi commerciali). Inserendo queste opere nella sezione non-free, indirettamente, facciamo capire ai loro autori che se fossero accluse nella sezione Main (principale) sarebbero più conosciute ed ampiamente impiegate. Così facendo gli autori sono gentilmente invitati a modificare la loro licenza pur di ottenere i suddetti benefici.</p> <p>Dopo un primo infruttuoso tentativo nel 2004, la completa rimozione della sezione non-free è improbabile possa essere messa nuovamente in atto, giacché contiene molti documenti utili, che sono stati spostati al suo interno in quanto non in linea con i nuovi requisiti della sezione Main. Questo è specialmente il caso di certi files di documentazione software rilasciati dal progetto GNU (in particolare di Emacs e Make).</p> <p>La costante presenza della sezione non-free è fonte di saltuari attriti con la Fondazione Free Software e la ragione principale per cui quest’ultima rifiuta di raccomandare ufficialmente Debian come sistema operativo.</p>
--	--

1.2.2. Le Linee Guida di Debian Free Software

Questo documento basilare chiarisce quale software si può considerare “free enough” (sufficientemente libero) per essere incluso in Debian. Se la licenza di un programma segue le seguenti clausole, il programma in questione può essere incluso nella sezione main; in caso contrario, ed ammesso e concesso che la distribuzione libera sia permessa [dalla sua licenza], lo si potrà trovare nella sezione non-free. La sezione non-free non fa parte ufficialmente di Debian; è un servizio aggiuntivo fornito agli utenti.

Questo testo è molto più di un criterio di selezione ed è diventato un’autorità in fatto di free software, prestandosi come fonte per l’“Open Source Definition” (Definizione Open Source). Storicamente è dunque una delle prime definizioni formali del concetto di “free software”.

La GNU General Public License, la BSD License e la Artistic License sono esempi delle tradizionali licenze libere che seguono i 9 punti riportati da questo testo. Sotto potrete trovare il testo pubblicato sul sito ufficiale di Debian.

♦ https://www.debian.org/social_contract#guidelines

1. Redistribuzione libera. La licenza di un componente Debian non potrà rattenere nessuna parte interessata, a partire da quella che vende o distribuisce il software [per software si intende l’ordine di appartenenza del summenzionato componente Debian, senza specificarne la famiglia o il genere] in quanto mero componente di una distribuzione software aggregata che contiene programmi di parecchie sorgenti diverse. La licenza non potrà richiedere royalty [concessione dei diritti a pagamento] o altri diritti simili alla vendita.

2. Il codice sorgente. Il programma deve comprendere il codice sorgente e deve consentire la distribuzione sotto forma di codice sorgente e di compiled form [linguaggio macchina].

3. Opere derivate. La licenza deve consentire modifiche ed opere derivate, inoltre deve consentire a quest’ultime di essere distribuite alle stesse condizioni della licenza del software originale.

4. Integrità del codice sorgente originale La licenza può rattenere il codice sorgente dall’essere distribuito in diversa forma solo se suddetta licenza consente la distribuzione di “patch files” assieme al codice sorgente al fine di modificare il programma nel “build time” [“al momento della sua compilazione”, dall’inglese “build” che letteralmente significa “costruzione”— in questo caso per “build” si intende il processo di conversione del codice sorgente in artefatti software]. La licenza deve espressamente autorizzare la distribuzione di software build basate su codice sorgente modificato. La licenza può richiedere alle opere derivate di portare un differente nome o numero di versione rispetto al software originale. (Questo è un compromesso. Il gruppo di Debian incoraggia tutti gli autori a non rattenere nessun file, sorgente o binario, dall’essere modificato).

5. Nessuna discriminazione contro persone o gruppi. La licenza non deve discriminare nessuna persona o gruppo di persone.

6. Nessuna discriminazione contro campi d’azione. La licenza non può rattenere nessuno dall’utilizzo del programma in un specifico “campo d’azione” [“field of endeavor”, dove endeavor in inglese significa letteralmente “tentativo”. Tale lemma a sua volta viene definito da “Il vocabolario italiano 2008” edito HOEPLI come <<l’azione mediante la quale si cerca di riuscire in qualcosa>>] Per esempio, la licenza non può rattenere il programma dall’essere usato in un affare o dall’essere usato per la ricerca genetica.

7. Distribuzione [efficacia] della licenza. I diritti ed i doveri annessi al programma devono applicarsi a tutti coloro i quali il programma è redistribuito senza la necessità per la sua efficacia di una licenza aggiuntiva per quelle parti interessate.

8. La licenza non deve essere specifica per Debian. I diritti ed i doveri annessi al programma non devono essere subordinati al fatto che il programma sia parte integrante di un sistema Debian. Se il

programma è stato estratto da Debian e usato o distribuito senza Debian, quantomeno entro le condizioni della licenza del programma, tutte le parti interessate, alle quali il programma è redistribuito, dovranno avere quegli stessi diritti e quegli stessi doveri che sono garantiti ed annessi al sistema Debian.

9. La licenza non deve influenzare altro software. La licenza non deve porre limiti su un altro software che è distribuito insieme al software registrato [sotto la suddetta licenza]. Per esempio, la licenza non deve sostenere che tutti i programmi distribuiti attraverso lo stesso software debbano essere free software.

BASILARE

Copyleft

Copyleft è un diritto che consiste nell'uso dei copyright per garantire la libertà di un'opera e dei suoi derivati, piuttosto che rattenere i diritti degli utenti come avviene per i software con diritti di proprietà intellettuale. È anche un gioco di parole basato sul termine "copyright". Richard Stallman trovò l'idea quando un suo amico, amante delle battute, scrisse un bigliettino destinato a lui: "copyleft: all rights reversed".

[“copyleft: tutti i diritti ed i doveri revocati” – “left” in inglese significa letteralmente “andato via”]. Copyleft impone la conservazione di tutte le libertà basilari sia sulla distribuzione di una versione originale, sia sulla distribuzione di una versione modificata della stessa opera (solitamente un programma). In questo modo è impossibile distribuire un programma come software con diritto di proprietà intellettuale se il codice con cui è scritto deriva da un programma registrato sotto la licenza copyleft.

La più diffusa classe di licenze copyleft è composta, certamente, dalla GNU GPL (GNU General Public License) e dalle sue derivate: la GNU LGPL (GNU Lesser General Public License) e la GNU GFDL (GNU Free Documentation License).

Sfortunatamente le licenze copyleft sono generalmente incompatibili tra loro. Di conseguenza un'ottima scelta è impiegarne al bisogno solo una.

10. Licenze da esempio. GPL, BSD e ARTISTIC sono degli esempi di licenze che possono essere considerate “libere”.

BASILARE

Licenze libere

La GNU GPL, la BSD license e l'Artistic License ottemperano tutte alle Linee Guida del Debian Free Software, sebbene siano molto differenti fra loro.

La GNU GPL, si serve ed è promossa dall' FSF (Free Software Foundation), è la più diffusa. La sua caratteristica principale è che si può anche applicare ad ogni opera derivata che è redistribuita: un programma che integra o che utilizza codice registrato sotto licenza GPL può essere distribuito solo in base alle condizioni della licenza GPL. È proibito, allo stesso tempo, qualsiasi riutilizzo in un'applicazione con diritti di proprietà intellettuale. Ciò comporta seri problemi per il riutilizzo di codice registrato sotto licenza GPL in free software incompatibili con la licenza GPL. Ad esempio, qualche volta è impossibile il linking (il collegamento) di un programma pubblicato sotto un'altra licenza free software ad una libreria distribuita sotto GPL. D'altro canto, questa licenza ha molta efficacia nel sistema giuridico americano: gli avvocati dell'FSF hanno partecipato alla sua stesura e hanno spesso costretto i suoi trasgressori a raggiungere un accordo extragiudiziale con l'FSF piuttosto che giungere ad una corte.

♦ <https://www.gnu.org/copyleft/gpl.html>

La licenza BSD è la meno restrittiva: ogni cosa è permessa, incluso l'uso di codice modificato, in origine registrato sotto licenza BSD, in una applicazione con diritti di proprietà intellettuale.

♦ <https://www.opensource.org/licenses/bsd-license.php>

E per ultima l'Artistic License raggiunge un compromesso fra le altre due: è consentito integrare codice registrato sotto l'Artistic Licence con applicazioni con diritti di proprietà intellettuale, ma qualsiasi modifica deve essere dichiarata pubblicamente.

♦ <https://www.opensource.org/licenses/artistic-license-2.0.php>

Il testo completo di queste licenze è disponibile in /usr/share/common-licenses/ su qualsiasi sistema Debian (per BSD la recente 3-Clause License).

COMUNITÀ
Bruce Perens, un leader controverso

Bruce Perens fu il secondo leader in ordine cronologico del progetto Debian, in carica dopo le dimissioni di Ian Murdock. Fu molto discutibile nei suoi metodi energici ed autoritari. Ciò nondimeno rimane un importante contributore, a cui Debian deve molto per il suo intervento nella realizzazione del famoso documento “Debian Free Software Guidelines (DFSG)”, frutto dell’ispirazione di Ean Schessler. Successivamente Bruce, dal summenzionato documento volle trarne la famosa “Open Source Definition” (“definizione Open Source”), rimuovendo tutti i riferimenti a Debian.

♦ <https://www.opensource.org/>

Il suo congedo fu alquanto commovente, ma Bruce è rimasto vivamente affezionato a Debian, tanto da continuare a promuovere questa distribuzione dal punto di vista politico ed economico. Sporadicamente appare nelle e-mail list per fornire assistenza e presentare le sue ultime iniziative a favore di Debian.

Inoltre, come ultimo aneddoto, fu Bruce l’artefice dell’ispirazione dei differenti nome in codice per le versioni di Debian (1.1 – Rex, 1.2 – Buzz, 1.3 – Bo, 2.0 – Hamm, 2.1 – Slink, 2.2 – Potato, 3.0 – Woody, 3.1 – Sarge, 4.0 – Etch, 5.0 – Lenny, 6.0 – Squeeze, 7 – Wheezy, 8 – Jessie, 9 – Stretch, 10 – Buster, 11 – Bullseye (al momento della stesura di questo manuale non ancora rilasciato), 12 – Bookworm, Unstable - Sid (al momento della stesura di questo manuale non ancora rilasciato), ecc.). I nomi in codice difatti sono tratti dai nomi dei personaggi del film Toy Story. Il film animato, interamente realizzato in computer graphics [computer grafica], fu prodotto dalla Pixar Studios, di cui era alla dipendenza Bruce durante la sua dirigenza nel progetto Debian. Il soprannome “Sid” invece si riferisce ad uno specifico status [condizione] ed è stato [dall’applicazione dell’artefizio dei nomi in codice], incessantemente associato al ramo Unstable. Nel film, questo personaggio era un giovanissimo vicino, che distruggeva sempre i giocattoli – pertanto fate attenzione quando vi avvicinate troppo alla Unstable. Sid è, per di più, l’acronimo di “Still In Development” [Ancora in fase di sviluppo].

1.3 Gli incarichi interni del Progetto Debian

I molteplici risultati raggiunti dal progetto Debian provengono simultaneamente: dal lavoro sull’infrastruttura eseguito da sviluppatori Debian esperti; dal lavoro individuale o collettivo dei sviluppatori sui pacchetti Debian e dalle segnalazioni degli utenti.

1.3.1 Gli sviluppatori Debian

Gli sviluppatori Debian hanno diverse responsabilità ed, in quanto membri ufficiali del progetto, hanno una grande incidenza sull’orientamento intrapreso dal progetto. Gli sviluppatori Debian sono generalmente responsabili almeno di un pacchetto, ma a seconda del loro tempo disponibile e dei loro propositi, sono liberi di farsi coinvolgere in numerosi teams, assumendosi così, molteplici responsabilità all’interno del progetto.

- ♦ <https://www.debian.org-devel/people/>
- ♦ <https://www.debian.org/intro/organization/>
- ♦ <https://wiki.debian.org/Teams>

STRUMENTI TOOLS

Developer's database. La banca dati degli sviluppatori.

Debian ha un database che comprende tutti gli sviluppatori iscritti al progetto ed i loro dati personali (indirizzo, telefono, coordinate geografiche – latitudine e longitudine, ecc.). Alcune di queste informazioni (nome e cognome, paese, nome utente nel progetto, account, chiave GnuPG, ecc.) sono pubbliche e disponibili sul web.

♦ <https://db.debian.org/>

Le coordinate geografiche consentono di tracciare la posizione di tutti gli sviluppatori nel mondo in una carta geografica (planisfero). Debian è davvero un progetto internazionale: i suoi sviluppatori si trovano ovunque, sebbene la maggior parte siano provenienti dall'Occidente.



Figura 1.1 Rappresentazione grafica della presenza degli sviluppatori in tutto il mondo

La manutenzione di un pacchetto è un'attività disciplinata, verbalizzata, nonché regolamentata. Infatti, deve assecondare tutte le condizioni stabilite dalla *Debian Policy* (Linee guida di condotta). Fortunatamente ci sono molti strumenti per il lavoro dei manutentori. Lo sviluppatore può, così, concentrarsi sulle caratteristiche del proprio pacchetto e sulle tasks (attività) complesse, come ad esempio lo squashing bugs (rimozione di bugs – il termine inglese “bug” significa letteralmente baco ed identifica in informatica un errore nella scrittura del codice sorgente di un programma software).

♦ <https://www.debian.org/doc/debian-policy/>

BASILARE

La manutenzione di un pacchetto

La manutenzione di un pacchetto consiste innanzitutto nel “packaging” (creazione e gestione di un pacchetto) di un programma. Nello specifico, questo significa definire i metodi di installazione così che, una volta installato, questo programma funzionerà ed ottempererà alle condizioni che Debian stabilisce. Il risultato di questa operazione è salvato in un file .deb. L'effettiva installazione del programma non richiederà niente di più dell'estrazione di questo archivio compresso e l'esecuzione di qualche scripts pre o post installazione contenuto al suo interno.

Dopo questa fase iniziale, il ciclo di manutenzione inizia davvero: preparazione degli aggiornamenti per conseguire l'ultima versione della *Debian Policy*; correzione dei bugs segnalati dagli utenti, inclusa una nuova versione upstream (a monte) che spontaneamente continua in parallelo la sua fase di sviluppo. Per esempio, nella fase iniziale del packaging, il programma è giunto alla sua versione 1.2.3. Dopo pochi mesi, gli autori originali del programma rilasciano una sua nuova versione 1.4.0. A questo punto il manutentore Debian deve aggiornare il pacchetto, in modo che gli utenti possano usufruire dell'ultima versione stabile del programma.

Le Policy (le linee guida di condotta), un elemento essenziale del progetto Debian, stabiliscono le regole che garantiscono non solo la qualità dei pacchetti, ma anche la perfetta interoperabilità della distribuzione. Grazie a queste Policy, Debian resta coerente nonostante la sua enorme dimensione. Queste Policy non sono “scolpite fisse sulla pietra”, bensì migliorano ininterrottamente grazie ai suggerimenti proposti attraverso la mailing list debian-policy@lists.debian.org. Gli Emendamenti [le correzioni], che sono concordati da tutte le parti interessate, sono accolti ed applicati al testo da un piccolo gruppo di manutentori, che non hanno alcuna responsabilità revisionale (si limitano ad applicare al testo le modifiche concordate dagli sviluppatori Debian, membri della summenzionata mailing list). Potete leggere le attuali proposte di correzione sul bug tracking system:

- ♦ <https://bugs.debian.org/debian-policy/>

COMUNITÀ

Le fasi della
revisione delle
Policy - Gli
Emendamenti

Chiunque può proporre una correzione alle Debian Policy inviando soltanto una segnalazione bug con un livello di severità wishlist (lista dei desideri) nei confronti del pacchetto `debian-policy`. Successivamente ha luogo un procedimento, illustrato nelle sue fasi in <https://www.debian.org/doc/debian-policy/approval-process.html>: qualora venisse accertata la sussistenza del problema segnalato questi deve essere risolto attraverso la creazione di una nuova regola nella Debian Policy; a tal proposito inizia una discussione sulla mailing list `debian-policy@lists.debian.org` che si conclude al raggiungimento del consenso richiesto ovvero quando emerge una proposta. Qualcuno poi prepara una bozza dell'emendamento e lo invia per la ratifica (sotto forma di patch da revisionare). Al più presto possibile altri due sviluppatori accertano che il contenuto dell'emendamento rifletta l'accordo raggiunto nella precedente discussione (come “cautela”) e così facendo la proposta potrà essere inserita nel documento ufficiale da un manutentore del `debian-policy` package. Se una delle fasi della procedura fallisce, i manutentori chiudono il bug, dichiarando la proposta respinta.

DEBIAN POLICY

La
documentazione

La documentazione di ciascun pacchetto è conservata nel `/usr/share/doc/package/`. Questa directory spesso contiene un file `README.Debian` che descrive dettagliatamente le modifiche Debian apportate dal manutentore al pacchetto. È pertanto consigliabile leggere questo file prima di effettuare una configurazione al fine di trarre beneficio dalla loro esperienza. Troviamo anche un file `changelog.Debian.gz` che descrive le modifiche apportate dal manutentore Debian da una versione precedente alla successiva. Non si deve confondere questo file con il file `changelog.gz` (o equivalente), che descrive i cambiamenti apportati dagli sviluppatori upstream. Il file `copyright` contiene informazioni circa gli autori e le licenze annesse al software. Alla fine, potremo anche trovare un file `NEWS.Debian.gz` che consente allo sviluppatore Debian di comunicare importanti informazioni riguardo agli aggiornamenti; se `apt-listchanges` è installato, questi messaggi saranno automaticamente visualizzati. Tutti gli altri file sono specifici al relativo software. In particolare ci piacerebbe far notare le subdirectories `examples`, che frequentemente contengono esempi di files di configurazione.

Le Policy forniscono un rilevante supporto agli aspetti tecnici del packaging. Anche la dimensione del progetto comporta problemi organizzativi; questi sono affrontati dalla Costituzione Debian, che stabilisce la struttura e gli strumenti per prendere le decisioni. In altre parole, delineando un sistema di governance formale.

La Costituzione chiarisce un definito numero di ruoli, di orientamenti e le responsabilità ed i poteri derivanti. È particolarmente degno di nota che gli sviluppatori Debian abbiano sempre l'ultima parola e che il loro potere decisionale derivi da una deliberazione collettiva, nella quale è richiesta una maggioranza qualificata di tre quarti (75 %) per eseguire modifiche rilevanti (in quanto queste

comportano conseguenze per i Foundation Documents – Documenti Fondanti). Nonostante ciò gli sviluppatori eleggono annualmente un “leader”, per rappresentarli negli incontri e garantire il coordinamento interno fra i vari teams. Questa elezione implica un periodo di intenso dibattito. Il ruolo del leader non è formalmente delineato da nessun documento: i candidati per questa posizione di solito prefiggono la definizione dell’incarico. Allo stato pratico il ruolo del leader comprende un incarico come rappresentante nelle pubbliche relazioni, di coordinamento per i teams interni e soprattutto un’offerta di orientamento al progetto, in cui gli sviluppatori possono rispecchiarsi: le opinioni del DPL (Debian Project Leader) sono implicitamente condivise dalla maggior parte dei membri del progetto.

Il leader ha un potere particolarmente efficace; la sua posizione è decisiva al fine del superamento del vincolo dei voti [o pareggio durante le votazioni]; al bisogno può emettere una decisione al posto di chi ne abbia incarico qualora non fosse ancora pronta e delegare parte dei suoi poteri.

Sin dalla creazione del progetto Debian, fino alla stesura di quest’opera l’incarico di leader è stato consecutivamente svolto da Ian Murdock, Bruce Perens, Ian Jackson, Wichert Akkerman, Ben Collins, Bdale Garbee, Martin Michlmayr, Branden Robison, Anthony Towns, Sam Hocevar, Steve McIntyre, Stefano Zacchiroli, Lucas Nussbaum, Mehdi Dogguy, Chris Lamb e Sam Hartman.

La costituzione definisce anche una “technical committee” (commissione tecnica). L’incarico fondamentale di questa commissione è di prendere decisioni su questioni tecniche quando gli sviluppatori sono arenati e non raggiungono un accordo fra loro. In altre parole, questa commissione fornisce un servizio di consulenza per ciascun sviluppatore che non riesce a prendere una decisione a lui spettante. È importante chiarire che la commissione interviene solo su richiesta delle parti interessate. Infine, la costituzione delinea la posizione del “project secretary” (il segretario del progetto) che accerta lo svolgimento delle votazioni durante le varie elezioni e le general resolution (o decisioni finali).

La procedura delle “general resolution” (decisioni finali) è espressa dettagliatamente nella costituzione, dalla discussione iniziale sino al conteggio dei voti. L’aspetto più interessante è che durante le votazioni gli sviluppatori possono esprimere anche una graduatoria di preferenze dei candidati ed il vincitore viene poi scelto in base al metodo Condorcet (https://en.wikipedia.org/wiki/Condorcet_method - più specificamente il metodo Schulze) Per maggiori informazioni visitate:

♦ <https://www.debian.org-devel/constitution>

CULTURA

Flamewar, la discussione che prende fuoco

Una flamewar (discussione infuocata) è un’appassionata discussione, ma eccessiva, che frequentemente scaturisce in un rissa verbale in cui tutte le parti interessate esauriscono le argomentazioni ragionevoli. Certi temi sono più frequentemente soggetti a polemiche rispetto ad altri (la scelta di un text editor, <> voi preferite vi o emacs?>, è ad esempio un pretesto noto). Le “querelle” spesso provocano scambi di email molto repentini anche a causa del numero delle persone coinvolte che hanno tutte opinioni diverse sull’argomento e la prendono sul personale.

Niente di buono scaturisce da queste discussioni; la raccomandazione comune è di starne fuori e probabilmente di leggere rapidamente i loro contenuti onde evitare di perdere troppo tempo con il formato intero.

Anche se la costituzione delinea le sembianze di una democrazia, la realtà quotidiana è completamente differente: Debian naturalmente persegue le condizioni del free software, ma basandosi su una do-ocrazy [dal verbo inglese to do che significa letteralmente fare: solo chi realizza le cose decide come farle funzionare]. Difatti si può perdere troppo tempo discutendo riguardo alle varie soluzioni per affrontare un problema ed ai rispettivi benefici; la soluzione prescelta dovrà essere quella che è allo stesso tempo funzionale e convincente... quella che scaturirà anche in base al tempo che una persona competente ha deciso di impiegare su di essa.

CULTURA

Meritocrazia, il regno della conoscenza.

La meritocrazia è una forma di governo nella quale l'autorità è esercitata da coloro i quali hanno i migliori meriti. Per Debian, il merito è il metro di misura per la competenza, fissato attraverso lo studio delle azioni passate di uno o più membri all'interno del progetto. (Stefano Zacchiroli, un precedente project leader, parla di "do-ocrazy", utilizzando palesemente un gioco di parole per esternare la massima "il potere a coloro i quali realizzano cose"). La loro semplice esistenza testimonia un definito livello di competenza; le loro opere essendo solitamente in free software, con disponibilità di codice sorgente, possono essere revisionate da membri con pari competenze per accertarne la qualità.

Questo è l'unico sistema per guadagnarsi i galloni del comando: realizzare qualcosa di utile e dimostrare di aver lavorato bene. Molti teams "amministrativi" selezionano i candidati attraverso la cooptazione, preferendo i volontari che hanno prontamente ed efficacemente contribuito e provato le loro competenze.

La natura pubblica del lavoro di questi teams consente ai nuovi contributori di esaminarlo e di iniziare a dare una mano senza la necessità di particolari privilegi. Questo è il motivo per cui Debian è spesso descritta come una "meritocrazia".

COMUNITÀ

L'integrazione dei nuovi manutentori

Il team preposto per l'ammissione degli sviluppatori è il più criticato. Prima di giudicare è bene sapere che nel corso degli anni il progetto Debian ha ricevuto numerose richieste di collaborazione da parte di sviluppatori, decisamente in più rispetto a quelle che potrà mai accettare. Anche se qualcuno intravedrà in ciò ingiustizia, dobbiamo confessare che le piccole sfide intraprese all'inizio della fondazione del progetto sono diventate alquanto gravose in una comunità con più di 1000 persone, soprattutto per assicurare con continuità la qualità e l'integrità del servizio di tutto quello che Debian realizza per i suoi utenti.

Per giunta, la procedura di selezione si conclude con il riesame del candidato da parte di un esiguo team, i Debian Account Managers. Questi Managers sono, in questo modo, particolarmente esposti a critiche, per il semplice fatto che hanno l'ultima parola sull'ammissione o l'esclusione di un volontario all'interno della comunità Debian nelle vesti di sviluppatore. In sostanza, qualche volta sono costretti a differire l'ammissione di un candidato sino a quando questi non ha imparato le procedure del progetto. In ogni caso si può contribuire prima della promozione a sviluppatore ufficiale, se si è sostenuti da sviluppatori in carica.

Questo efficiente metodo amministrativo garantisce la qualità dei contributori nei team "chiave" di Debian. Il suddetto metodo non deve essere perfetto ed occasionalmente qualcuno non accetta questo modo di fare. La selezione degli sviluppatori ammessi nei teams può apparire alquanto discrezionale, o persino ingiusta. Ma se dobbiamo essere sinceri, tutte le parti interessate non hanno lo stesso identico metro di giudizio riguardo al servizio svolto da questi teams. Difatti per qualcuno è inaccettabile attendere otto giorni per l'inclusione di un nuovo pacchetto Debian, mentre altri attenderebbero volentieri anche tre settimane senza problemi. A causa di tale contesto, ci sono diverse lamentele dovute al malcontento riguardo alla "qualità del servizio" svolto da qualche team.

1.3.2. Il ruolo attivo degli utenti

C'è da chiedersi se sia doveroso menzionare gli utenti fra coloro i quali lavorano nel progetto Debian, ma la risposta è palesemente affermativa: gli utenti svolgono un ruolo cruciale. Piuttosto che beneficiare passivamente del progetto, qualche utente concorre allo sviluppo della versione di Debian e regolarmente segnala bugs nei files per evidenziare le falte del sistema. Altri preferiscono spingersi oltre inviando idee per i miglioramenti, attraverso la presentazione di un bug report con livello di severità wishlist (lista dei desideri), oppure correzioni al codice sorgente, definite "patches" (per maggiori informazioni vedere il paragrafo 1.3.2.3 "Come condividere una fix [riparazione]" a pag. 15).

1.3.2.1 Come effettuare il report dei bugs

Lo strumento fondamentale per effettuare il report dei bugs è il Debian Bug Tracking System (Debian BTS), usato dalla maggioranza delle parti interessate al progetto. La schermata comune (l'interfaccia web) del sistema consente agli utenti di vedere tutti i bug segnalati, con la possibilità di visualizzare un'assortita lista di bug ordinata secondo vari criteri, come ad esempio: affected package (il pacchetto colpito dal bug), severity (la severità della problematica), status (la condizione), address of the reporter (l'indirizzo di chi segnala il bug), address of the maintainer (l'indirizzo del manutentore preposto), tag (parole chiave), ecc. È anche possibile curiosare nella cronologia completa che comprende tutte le discussioni inerenti ciascun bug.

Sostanzialmente il Debian BTS è basato sulle email: tutta l'informazione che immagazzina proviene dai messaggi delle diverse persone interessate. Ciascuna email inviata a 12345@bugs.debian.org sarà, pertanto, contrassegnata nella cronologia come "bug numero 12345". Solo le persone autorizzate possono chiudere un bug scrivendo un messaggio che delucida le ragioni di tale azione a 12345-done@bugs.debian.org (solitamente un bug è chiuso quando il problema è risolto o non attualmente rilevante). Un nuovo bug è segnalato attraverso l'invio di un'email a submit@bugs.debian.org in un formato specifico, che identifica il pacchetto in questione.

L'indirizzo control@bug.debian.org consente la correzione di tutte "meta-informationi" inerenti al bug.

Il Debian BTS ha anche altre configurazioni utili, come quelle già espresse, per contrassegnare i bugs tra cui l'uso di tags. Per maggiori informazioni vedere:

♦ <https://www.debian.org/Bugs/>

DIZIONARIO

La severità di un bug

La severità di un bug assegna formalmente un grado di serietà con cui etichettare la problematica. Infatti non tutti hanno la stessa rilevanza; per esempio un errore tipografico nella pagina di un manuale non è paragonabile ad una vulnerabilità di sicurezza in un server software.

Debian utilizza un'ampia scala di livelli per descrivere la severità di un bug. Ciascun livello è espresso esplicitamente in modo da facilitarne la scelta.

♦ <http://www.debian.org/Bugs/Developer#severities/>

Gli utenti possono anche usufruire della riga di comando per inviare i bugs reports riguardo ad un pacchetto Debian attraverso lo strumento reportbug. Lo strumento reportbug (report in inglese significa letteralmente segnalazione) facilita la segnalazione di bugs di un pacchetto Debian. Questi accerta che il bug in questione non sia già stato segnalato da terzi, scongiurando così un inutile accumulo di reports nel sistema. Il suddetto tool suggerisce, inoltre, all'utente la nomenclatura dei livelli di severità in modo che la segnalazione possa essere la più accurata possibile (in ogni caso lo sviluppatore può sempre, successivamente, modificare al bisogno tali parametri, raffinandoli). Il reportbug facilita quindi una comunicazione esaustiva del bug senza che l'utente debba conoscere la sintassi, scrivendo di fatto al suo posto il report e consentendogli di modificarlo. La segnalazione

sarà poi trasmessa ad un server email (locale, di default, ma può essere usato anche un server remoto).

Lo scopo principale di questo strumento è il miglioramento delle versioni in sviluppo di Debian, dove in pratica saranno corretti i bugs. Difatti, i cambiamenti nella versione stabile di Debian, con poche eccezioni per gli aggiornamenti di sicurezza ed altri importanti aggiornamenti (ad esempio quando un pacchetto non funziona del tutto), non sono graditi. Pertanto la correzione di un bug con un livello di severità non rilevante deve attendere la successiva versione stabile.

1.3.2.2 Traduzioni e documentazione

Inoltre, numerosi utenti soddisfatti del servizio di Debian contribuiscono direttamente al progetto. Anche coloro i quali non hanno competenze di programmazione, possono optare di assistere con la traduzione e la revisione della documentazione. Se siete interessati ci sono mailing list specifiche per le traduzioni volte a coordinare questo incarico.

- ♦ <https://list.debian.org/i18n.html>
- ♦ <https://www.debian.org/international/>

BASILARE

Cosa sono i18n e
l10n

“i18n” e “l10n” sono le abbreviazioni dei lemmi “internationalization” e “localization”, ovvero, rispettivamente, la prima e l’ultima lettera di ciascuna parola, con in mezzo il numero delle lettere che la compongono.

Internazionalizzare un programma significa prepararlo attraverso una modifica alla sua traduzione (localizzazione). Ciò comporta la parziale revisione di un programma, inizialmente scritto per funzionare in una sola lingua, al fine di renderlo “aperto” a tutte le lingue.

Localizzare un programma significa tradurre i messaggi originali (di solito in inglese) in un’altra lingua. Per questo, occorre che sia già stato preparato con l’internazionalizzazione.

In sintesi, l’internazionalizzazione prepara il software per la traduzione, che è poi messa in atto con la localizzazione.

1.3.2.3 Come condividere una fix [riparazione]

Gli utenti più avanzati qualora lo desiderassero e fossero in grado potranno inviare le loro correzioni come patch.

[In informatica, un “reference” (riferimento) è un valore che consente a un programma di accedere indirettamente a un dato particolare, come il valore di una variabile o un record, nella memoria del computer o in qualche altro dispositivo di archiviazione. Invece “l’indirezione” (detta anche “riferimento indiretto”) è la tecnica che consente di indicare un oggetto o un valore mediante un suo riferimento invece che direttamente. Il termine viene utilizzato frequentemente nei linguaggi di programmazione per indicare l’uso dei puntatori. I puntatori sono tipi di dati che rappresentano la posizione (usando indirizzi di memoria) di elementi del programma come variabili, oggetti, strutture dati, sottoprogrammi.] Una patch (in inglese significa letteralmente “tappa”) è un file che descrive le modifiche da eseguire sui uno o più reference files. Nello specifico, conterrà una lista di righe da rimuovere o aggiungere al codice, righe estratte (occasionalmente) dal reference text, rimpiazzando le correzioni nel context (in questo modo l’identificazione della collocazione delle modifiche da apportare avverrà comunque qualora il numero delle righe fosse cambiato nel file da modificare). Lo strumento usato per applicare le correzioni contenute in un dato file è chiamato semplicemente allo stesso modo del file ovvero patch. Lo strumento che crea il file patch è chiamato diff ed è impiegato come segue:

```
$ diff -u file.old file.new >file.patch
```

Il file patch contiene le istruzioni per cambiare il contenuto del file.old in file.new. Possiamo inviare il file patch a qualcuno, così che possa generare il file.new dagli altri due (il file.old ed il file.patch) in questo modo:

```
$ patch -p0 file.old <file.patch
```

Il summenzionato comando patch cambierà il file.old, rendendolo identico al file.new, attraverso il file.patch.

Solitamente il software è manutenuto in repositories Git ed è pertanto più probabile che i contributori utilizzino Git per recuperare il codice sorgente e proporre poi delle modifiche. Inoltre i comandi git diff e git apply funzionano rispettivamente allo stesso modo dei comandi diff -u e patch.

CULTURA	
Git	<p>Git è uno strumento per il lavoro collaborativo su molteplici files, in grado di conservare una cronologia storica delle modifiche. I files in questione sono generalmente files text, come quelli del codice sorgente di un programma. Se diverse persone lavorano insieme sullo stesso file, git può incorporare solo le modifiche realizzate in porzioni differenti del file. In caso contrario, questi "conflitti" devono essere risolti manualmente. Git è un "distributed system" in cui ciascun utente ha un repository con l'intera cronologia delle modifiche. I "central repositories" sono usati per scaricare il progetto (git clone) e condividere il lavoro realizzato con gli altri (git push). Il repository può contenere molteplici versioni dei files, ma solo una versione, alla volta, può esserne rimaneggiata: questa è definita working copy (può essere convertita allo stato di avanzamento di un'altra versione con git checkout). Git può mostrarvi le modifiche effettuate su una working copy (git diff), può visualizzare in anteprima le modifiche da includere (git add) può depositare le modifiche nel repository allo scopo di creare una nuova voce (new entry) nella cronologia delle versioni (git commit). Inoltre Git è in grado di aggiornare la working copy per incorporarvi le modifiche effettuate in parallelo da altri utenti (git pull) e può registrare una configurazione ad hoc nella cronologia così da portela estrarre con facilità in seguito (git tag). Git rende facile gestire molteplici versioni convergenti di un unico progetto, senza che queste possano interferire fra loro. Queste versioni sono chiamate branches ["branches" in inglese significa letteralmente "rami"]. La metafora di un albero è perfettamente calzante, dal momento che il programma è inizialmente sviluppato da un common trunk [un unico tronco, infatti "trunk" in inglese significa letteralmente "tronco"]. Quando viene raggiunto un traguardo non indifferente (come ad esempio la prima versione di un programma, denominata solitamente "versione 1.0"), lo sviluppo continua su due branches differenti: il development branch si occuperà della release successiva, mentre il maintenance branch gestirà gli aggiornamenti e le "fixes" ["fixes" in inglese significa letteralmente "correzioni"] della "versione 1.0". Git è, oggi, la più popolare versione di control system, pur non essendo l'unica [a grandi linee l'espressione "control system", indica genericamente un sistema che gestisce, comanda, dirige o regola il funzionamento di altri dispositivi o sistemi, utilizzando un control loops, che, a sua volta, consiste nelle componenti fisiche e nelle funzioni di controllo necessarie per regolare automaticamente il valore delle variabili di un processo sequenziale e cadenzato, al fine di raggiungere il valore di una configurazione desiderata]. Storicamente, CVS (Concurrent Versions Systems) fu il primo strumento ad essere ampiamente usato, ma le sue numerose limitazioni hanno contribuito a fare emergere maggiormente le alternative free moderne. Fra queste, in particolare, troviamo subversion (svn), git, bazaar (bzr) e mercurial (hg).</p> <ul style="list-style-type: none">◆ https://www.nongnu.org/cvs/◆ https://subversions.apache.org/◆ https://git-scm.com/◆ https://bazaar.canonical.com/◆ https://mercurial.selenic.com/ <p>In questa sede non è possibile dare ulteriori delucidazioni su Git, dato che non è lo scopo di questo libro, ma potrete comunque fare riferimento al manuale Pro Git :</p> <ul style="list-style-type: none">◆ https://git-scm.com/book

Per quanto l'output di git diff sia sufficiente per condividere un file fra sviluppatori, esistono dei metodi più efficienti per inviare le modifiche. Ad esempio se gli sviluppatori desiderano ricevere le

patch via email, solitamente pretendono siano generate attraverso `git format-patch` in modo da poterle poi integrare nel repository con `git am`. Ciò consente di preservare le meta-information delle commit e la loro condivisione multipla.

Ma nonostante la popolarità acquisita il workflow via email tende ad essere rimpiazzato dall'impiego di `merge requests` (o `pull requests`) specialmente se il software è ospitato in una piattaforma come Github o GitLab - e Debian usufruisce di GitLab sul suo server `salsa.debian.org`. Nei suddetti sistemi occorre semplicemente creare un account, generare un fork del repository (ossia creare una copia del repository di vostro interesse nel vostro stesso account), dopodiché potrete clonare il repository ed immettere le vostre modifiche. Giunti a questo punto l'interfaccia web vi suggerirà di inviare una `merge request`, in modo che gli sviluppatori siano informati della disponibilità delle vostre modifiche, possano revisionarle facilmente e decidere se accettarle con un click.

1.3.2.4 Altri modi per contribuire

Tutte queste procedure di sostegno allo sviluppo del software sono rese in concreto funzionali dal comportamento degli utenti. Questi difatti, piuttosto che rimanere in disparte, sono una vera e propria comunità, nella quale hanno luogo numerosi scambi di informazione e supporto reciproco. Ci riferiamo in particolar modo all'impressionante attività di collaborazione fra gli utenti nella mailing list `debianuser@lists.debian.org` (il capitolo 7 “Come risolvere le problematiche e trovare le informazioni adeguate” a pagina 148 tratta questo argomento con maggiori dettagli).

♦ <https://lists.debian.org/users.html>

Gli utenti, discutendo fra loro, non aiutano solo loro stessi (ed altri) nelle questioni tecniche di cui sono direttamente interessati, bensì confrontano le loro opinioni per contribuire nel miglior modo possibile al progetto stesso e spingerlo oltre – discussioni che di conseguenza si tramutano spesso in vere e proprie proposte di migliorie.

STRUMENTI TOOL `how-can-i-help`

Il programma `how-can-i-help` elenca le opportunità per contribuire ai pacchetti Debian installati localmente. Dopo ogni invocazione di APT il summenzionato programma propone dei modi per aiutare segnalandovi i bugs taggati come “newcomer” [in ital. trad. lett “nuovo arrivato”] (che rappresentano per i nuovi contributori un buon mezzo per fare strada) o gli orphaned packages (i pacchetti orfani) che necessitano di un nuovo maintainer. Il suddetto programma può essere eseguito anche direttamente.

Dal momento che Debian non impiega denaro per campagne di promozione commerciale, i suoi utenti ricoprono un ruolo fondamentale nella sua diffusione, garantendone la notorietà attraverso il passaparola.

Questo sistema funziona piuttosto bene, visto che i sostenitori di Debian operano a tutti i livelli della comunità free software: dagli install parties (workshops dove gli utenti stagionati aiutano i neofiti all’installazione del sistema) organizzati dalle LUGs territoriali o “Linux Users Groups” (Gruppi Utenti Linux), alle bancarelle delle associazioni sino alle vaste conventions tech che supportano Linux, ecc..

I volontari realizzano per il progetto posters, brochures, stickers ed altri materiali promozionali, resi da questi “liberi” e rilasciati da Debian gratuitamente sul suo sito:

♦ <https://www.debian.org/events/material/>

1.3.3. Teams e Sotto-Progetti

Il progetto Debian è stato imperniato, sin dalla sua creazione, intorno alla figura dei pacchetti sorgente, ciascuno con il loro manutentore o gruppo di manutentori. Molti teams operativi sono emersi nel corso del tempo, assicurando anche l'amministrazione dell'organizzazione, la gestione di attività generiche non legate ad un particolare pacchetto (la garanzia di qualità, Debian Policy, l'installer, ecc.), ma i teams più recenti si stanno concentrando sui sotto-progetti.

1.3.3.1. Gli attuali Sotto-Progetti Debian

A ciascuno il proprio Debian! Un sotto-progetto è gestito da un gruppo di volontari il cui obiettivo è adeguare Debian ad esigenze particolari. I sotto-progetti non sono coinvolti solo nella scelta di una sotto-categoria di programmi finalizzati ad uno specifico settore (didattico, per servizi medico-sanitari-ospedalieri, creazione multimediale, ecc.), ma anche nella manutenzione dei pacchetti esistenti, nel packaging del missing software (software per ragioni varie non più disponibile), nell'adeguamento dell'installer, nella creazione di documentazione ad-hoc e molto altro ancora.

<p style="text-align: center;">DIZIONARIO Il Sotto-Progetto e le distribuzioni derivate</p>	<p>Il modello di sviluppo di una distribuzione derivata consiste nell'apportare un numero N di modifiche ad una specifica versione di Debian. L'organizzazione impegnata a questo scopo è totalmente estranea al Progetto Debian. Nelle derivate non occorre applicare la Debian Policy per le migliorie apportate. Questa differenza spiega perché una derivata potrebbe "divergere" rispetto alla versione Debian da cui è stata realizzata e la sua necessità di una costante "sincronizzazione" con quest'ultima, pur di beneficiare delle migliorie realizzate upstream (al monte). Diversamente dalle distribuzioni derivate, un sotto-progetto non può "divergere", in quanto tutto l'operato su di esso consiste nel perfezionare una versione di Debian allo scopo di indirizzarla ad un preciso obiettivo. La più famosa distribuzione derivata è sicuramente Ubuntu, ma ce ne sono altre. Per saperne di più al riguardo, in particolare sulle loro specifiche e sul loro rapporto di parentela con Debian, andate a vedere l'Appendice A, "Distribuzioni Derivate" a pagina 469.</p>
--	--

Riportiamo qui un'esegua selezione di attuali sotto-progetti:

- Debian Jr, di Ben Armstrong, propone un sistema Debian adeguato ai bambini, accattivante e facile da utilizzare;
- Debian Edu, di Peter Reinholdstein, è un sotto-progetto finalizzato alla creazione di una distribuzione specifica per il mondo accademico;
- Debian Med, di Andreas Tille, sotto-progetto dedicato al campo della medicina;
- Debian Multimedia, sotto-progetto finalizzato alla produzione di opere audio e video;
- Debian GIS, sotto-progetto finalizzato alle applicazioni tipiche di un geographic information system (sistema informativo geografico o anche sistema informativo territoriale), generalmente abbreviato in GIS, ed ovviamente ai suoi utenti;
- Debian Accessibility, sotto-progetto che di fatto perfeziona Debian adeguandolo alle esigenze delle persone con disabilità;
- Debian Science, dedicata ai ricercatori ed agli scienziati in modo da migliorare la loro l'interazione con Debian;

- DebianChem, destinata alla Chimica, include suites e programmi per la chimica.

L'elenco dei progetti diverrà sempre più esteso in quanto i sotto-progetti Debian continueranno a crescere nel corso del tempo, avvalorando così i loro vantaggi.

Pienamente sostenuti dall'attuale infrastruttura Debian, questi sotto-progetti possono difatti concentrarsi sul valore aggiunto del loro operato, senza doversi preoccupare di rimanere sincronizzati con una particolare versione di Debian, dal momento che loro stessi sono sviluppati all'interno del progetto.

1.3.3.2. Teams di Amministrazione

La maggior parte dei teams di amministrazione sono relativamente distaccati dal resto della comunità ed i loro membri sono reclutati solo attraverso la cooptazione. Il miglior modo per entrare a farne parte è assistere diligentemente i membri in carica, dimostrando di averne compreso gli obiettivi e le procedure.

Gli ftpmasters sono responsabili dell'archivio ufficiale dei pacchetti Debian. Loro manutenzionano il programma, che riceve i pacchetti inviati dagli sviluppatori ed automaticamente li immagazzina dopo qualche verifica sul server preposto (ftp-master.debian.org).

Gli ftpmasters devono anche validare le licenze dei nuovi pacchetti, accertandosi che Debian li possa distribuire, prima di immetterli nella raccolta dei pacchetti già esistenti. Quando uno sviluppatore desidera rimuovere un pacchetto, contatta il team attraverso il bug tracking system e lo "pseudo-package" ftp.debian.org (solitamente destinato alle segnalazioni dei disservizi inerenti ai servers FTP).

[In breve, riguardo allo "pseudo-package", è possibile comunque aprire una segnalazione bug BTS via email (con un formato specifico) se il problema non è legato ad un solo pacchetto, ma ad un generico servizio di Debian elencato in una lista e definito "pseudo-package" su:

<https://www.debian.org/Bugs/pseudo-packages>]

DIZIONARIO

Lo pseudo-package, uno strumento di monitoraggio

Il bug tracking system, uno strumento di tracciamento, fu inizialmente designato per correlare le segnalazioni bug ad un pacchetto Debian, ma si è dimostrato valente nel sbrigare altre faccende: difatti elenca i problemi da risolvere o le attività da gestire, senza la necessità di un collegamento ad uno specifico pacchetto Debian. I "pseudo-packages" consentono pertanto a determinati teams di usare il bug tracking system, senza necessità di alcuna correlazione fra un reale pacchetto ed il loro team. In questo modo chiunque può evidenziare i generici disservizi che necessitano un approfondimento, senza doverli correlare ad un pacchetto qualora ne sconoscesse l'origine. Di conseguenza il BTS può avere ad esempio una nuova segnalazione ftp.debian.org, che è usata per verbalizzare e delineare le problematiche di un pacchetto dell'archivio ufficiale o semplicemente per sollecitarne la rimozione. Mentre lo pseudo-package su www.debian.org fa risalire alla causa degli errori del sito ufficiale di Debian e... lo pseudo-package su lists.debian.org raccoglie tutte anomalie relative alle mailing lists e... così via.

STRUMENTI TOOLS

GitLab, l'hosting per i repositories Git ed altro ancora

Un'istanza GitLab, denominata salsa.debian.org, viene utilizzata da Debian per ospitare i repositories che gestiscono i pacchetti Git anche se in realtà questo software offre maggiori funzionalità rispetto al mero hosting, onde per cui gli sviluppatori Debian ne hanno approfittato per sfruttare le funzioni di integrazione continua (esecuzione di tests, compilazione dei pacchetti ad ogni push). Inoltre i contributori Debian finalmente possono avvantaggiarsi di un workflow pulito grazie al processo merge request (simile alle pull requests di GitHub).
GitLab ha sostituito FusionForge (in esecuzione attraverso il servizio denominato alioth.debian.org) per la manutenzione collaborativa dei pacchetti. GitLab è amministrato da Alexander Wirt, Bastian Blank e Jörg Jaspers.

- ◆ <https://salsa.debian.org/>
- ◆ <https://wiki.debian.org/Salsa/Doc>

Il team Debian System Administrators – DSA (il team degli Amministratori di Sistema Debian), debian-admin@lists.debian.org, come chiunque potrebbe aspettarsi, è responsabile dell'amministrazione del sistema di molti servers usati dal Progetto Debian. I suddetti amministratori assicurano il funzionamento ottimale di tutti i servizi di base (DNS, Web, e-mail, Shell, ecc.), installano il

software preteso dagli sviluppatori e prendono tutte le precauzioni necessarie in merito alla sicurezza.

- ♦ <https://dsa.debian.org>

STRUMENTI TOOLS

Debian Package Tracker – interfaccia web [“tracker” in inglese significa letteralmente “perlustratore”]

Questa è una delle creazioni di Raphaël. L’idea di base è di raccogliere la maggior parte delle informazioni possibili, relative ad un pacchetto, in un’unica pagina. Così che chiunque possa rapidamente verificare lo status di un programma, conoscere le tasks per perfezionarlo ed offrire la propria assistenza. Tutto ciò è concretamente possibile in quanto questa pagina contiene tutte le statistiche bugs, le versioni disponibili per ciascuna distribuzione, lo stato di avanzamento di un pacchetto nella distribuzione Testing, lo status delle traduzioni delle descrizioni e dei debconf templates, l’eventuale disponibilità di una nuova versione upstream, avvisi relativi la sua non ottemperanza alle condizioni dell’ultima versione della Debian Policy, informazioni sul manutentore, e qualsiasi altra informazione che il suddetto manutentore desideri accludervi.

- ♦ <https://tracker.debian.org>

Un servizio di sottoscrizione email completa questa interfaccia web. Tale servizio invia automaticamente informazioni mirate agli iscritti della lista: bugs e relative discussioni, la disponibilità di una nuova versione sui servers Debian, la disponibilità di nuove traduzioni per i revisori, ecc.

Gli utenti avanzati potranno, così, seguire tutte queste informazioni accuratamente e contribuire eventualmente al progetto, quando avranno una sufficiente padronanza delle sue procedure.

Un’altra interfaccia web, conosciuta come Debian Developer’s Package Overview (DDPO), fornisce la sinossi dello status dei pacchetti Debian in base al loro manutentore ossia l’utente potrà ottenere tutte le informazioni di un pacchetto indicandone il manutentore incaricato nell’apposito format.

- ♦ <https://qa.debian.org/developer.php>

Questi due siti web sono strumenti sviluppati e gestiti dal gruppo responsabile della garanzia di qualità di Debian (conosciuti come Debian QA)

I listmasters amministrano l’email server che gestisce le mailing lists. Creano nuove liste, gestiscono le risposte automatiche delle mail inviate (come gli avvisi di mancato recapito – delivery failure notices) e verificano i filtri spam (per le email di massa indesiderate).

CULTURA

Il traffico sulle mailing lists: qualche cifra

Le mailing lists sono, senza alcun dubbio, la miglior forma di comunicazione delle attività del progetto, dal momento che divulgano tutto ciò che ruota intorno ad esse. Un po’ di numeri estratti dai resoconti (a partire da Maggio 2019) sulle nostre mailing lists chiarisce quanto appena espresso: Debian ospita più di 315 liste, con in totale circa 303,000 sottoscrizioni individuali. Sono 227,000 le comunicazioni emails trasmesse ogni giorno.

Ciascun servizio specializzato ha un suo apposito team, generalmente composto dai volontari che lo hanno installato (e, spesso, programmato anche i relativi tools). Questo è il caso di: Bug Tracking System (BTS), Package Tracking System (BTS), salsa.debian.org (server con installato GitLab, per maggiori chiarimenti leggere la casella di testo “GitLab, l’hosting per i repositories Git ed altro ancora” a pagina 19) e dei servizi disponibili su qa.debian.org, lintian.debian.org, buildd.debian.org, cdimage.debian.org, ecc..

1.3.3.3. Teams di Sviluppo, Schieramenti Trasversali dei Teams.

Diversamente dai teams amministrativi, i teams di sviluppo sono decisamente piuttosto aperti, anche ai contributori esterni. Pur non avendo Debian un orientamento prefissato per sviluppare il software, il progetto necessita comunque di programmi mirati per raggiungere i suoi obiettivi. Questi strumenti, naturalmente, sviluppati sotto una licenza free software, sfruttano i sistemi testati altrove nel mondo free software.

Debian ha sviluppato solo una parte del suo stesso software, ma diversi programmi hanno rivestito un ruolo cardine e la loro reputazione si è protesa ben oltre la portata del progetto. Buoni esempi sono dpkg, il programma per la gestione dei pacchetti Debian (è infatti l'abbreviazione di Debian PackAge, e si pronuncia generalmente come se si dovesse leggere in inglese il lemma “dee-package”), e apt, uno strumento per installare automaticamente qualsiasi pacchetto Debian, e le sue dipendenze, garantendo la sicurezza del sistema dopo un aggiornamento (il suo nome è l'acronimo di Advanced Package Tool). I loro teams, ciononostante, sono molto esigui, dal momento che è richiesto un ragguardevole livello di abilità di programmazione per poter poi acquisire una comprensione su ogni aspetto delle operazioni di questo genere di programmi.

Il più importante team è probabilmente quello che si occupa del programma di installazione di Debian, debian-installer, che ha svolto un incarico di proporzioni epiche sin dalla sua creazione nel 2001. Sono stati necessari numerosi contributori, poiché scrivere un programma in grado di installare Debian su una dozzina di architetture differenti non è facile. Ciascuna architettura ha un suo meccanismo di avvio e di conseguenza necessita di un apposito programma di lancio (bootloader). Questo incarico è interamente coordinato sulla mailing list debian-boot@lists.debian.org, sotto la guida di Cyril Brulebois.

- ♦ <https://www.debian.org/devel/debian-installer/>
- ♦ https://joeyh.name/blog/entry/d-i_retrospective/

Il team (straordinariamente ristretto) del programma debian-cd giustamente ha un obiettivo ancora più ridimensionato. I suoi pochissimi contributori sono incaricati della loro stessa architettura, dal momento che lo stesso sviluppatore al comando potrebbe non conoscere tutte le sottigliezze o addirittura il modo corretto per lanciare l'installer dal CD-ROM.

Molti teams sono costretti a collaborare con altri nell'attività del packaging: debian-qa@lists.debian.org si sforza, per esempio, di assicurare la qualità sotto ogni punto di vista del progetto Debian. Il debian-policy@lists.debian.org si occupa dello “sviluppo” della Debian Policy assecondando i suggerimenti giunti da ogni dove e sopra ogni cosa. Il team incaricato di ciascuna architettura (debian-architecture@lists.debian.org) compila tutti i pacchetti, adattandoli, se necessario, alle loro rispettive architetture.

Altri teams gestiscono i più importanti pacchetti in modo da assicurarne la manutenzione ed allo stesso tempo non sovraccaricare di responsabilità nessuno; questo è il caso della libreria di C seguita da debianlibc@lists.debian.org, del compiler (il compilatore) di C approfondito sulla mailing list debian-gcc@lists.debian.org, o di Xorg trattato su debian-x@lists.debian.org (questo gruppo è anche conosciuto come X Strike Force).

1.4 Come seguire gli aggiornamenti di Debian

Come abbiamo appena accennato, il progetto Debian evolve perseguiendo un sentiero molto ordinato, molto strutturato. Di conseguenza, potrebbe essere difficile restare informati sulle ultime novità all'interno del progetto, senza poi essere sopraffatti da una valanga di notifiche senza fine.

Se volete conoscere soltanto le notizie più importanti su Debian, dovreste probabilmente iscrivervi alla mailing list debian-announce@lists.debian.org. Questa è una mailing list con un indice di traffico decisamente basso (circa una dozzina di messaggi all'anno), che trasmette soltanto le comunicazioni più importanti come ad esempio: la disponibilità di un nuova release stable, l'elezione di un Project Leader oppure la Conferenza Debian annuale.

♦ <https://lists.debian.org/debian-announce/>

Notizie più generiche (e costanti) su Debian sono trasmesse tramite le mailing list debian-news@lists.debian.org. Il traffico su questa mailing list è altrettanto ragionevole (solitamente una manciata di messaggi al mese) ed include, con una frequenza di aggiornamento semi-regolare, la “Debian Project News” (DPN), una raccolta di informazioni di pochi bits sulle novità del progetto.

♦ <https://lists.debian.org/debian-news/>

COMUNITÀ

Il team pubblicitario

I canali delle comunicazioni ufficiali di Debian sono gestiti dai volontari del team pubblicitario. I membri ricevono il loro mandato dal Debian Project Leader e gestiscono news ed annunci. Molti volontari collaborano con il team pubblicitario ad esempio sotto forma di articoli per il “Debian Project News” ed il “Debian’s official blog” (bits.debian.org - <https://bits.debian.org>), oppure sotto forma di aggiornamenti per il servizio di microblog (micronews.debian.org - <https://micronews.debian.org>) che supporta siti social networks con contenuti tipici del microblogging [brevi messaggi di testo, immagini, video, citazioni, appunti, ecc].

♦ <https://wiki.debian.org/Teams/Publicity>

Per ricevere più informazioni sull’evoluzione di Debian e per conoscere le attività dei vari teams di un dato periodo, è possibile iscriversi alla mailing list debian-devel-announce@lists.debian.org. Come il nome della suddetta mailing list suggerisce, le comunicazioni in essa contenute probabilmente attireranno di più l’interesse degli sviluppatori, ma debian-devel-announce@lists.debian.org consente anche alle altre parti interessate di dare un’occhiata alle attività in termini più concreti, piuttosto che limitarsi di informarle del rilascio di una versione stabile. Difatti la mailing list debian-announce@lists.debian.org trasmette le notizie riguardo ai traguardi raggiunti di cui gli stessi utenti possono facilmente accertarsi, mentre la mailing list debian-devel-announce@lists.debian.org comunica le informazioni riguardo alle modalità con cui i summenzionati risultati sono stati realizzati. Si fa presente, di conseguenza, che la debian-devel-announce@lists.debian.org o “d-d-a”, come solitamente ci si riferisce, è l’unica mailing a cui gli sviluppatori dovrebbero iscriversi.

♦ <https://lists.debian.org/debian-devel-announce/>

Anche il blog ufficiale di Debian (bits.debian.org - <https://bits.debian.org>) è una buona fonte di informazioni. Difatti condivide le notizie pubblicate attraverso le diverse mailing list oppure le news dei membri della comunità. Ovviamente, dato che tutti gli sviluppatori possono contribuire, ciò avviene solo se il caso lo richiede, e nel frattempo il blog di Debian si occupa di darne visibilità senza perdere di vista lo scopo del progetto in sé.

Una fonte informale di aggiornamenti è sicuramente Planet Debian, che condivide i post pubblicati dai contributori Debian sui loro rispettivi blogs. Anche se i suoi contenuti non sono dedicati esclusivamente allo sviluppo di Debian, forniscono in ogni caso una prospettiva su cosa avviene all’interno della comunità e cosa i suoi membri sono in grado di mettere in piedi.

♦ <http://planet.debian.org/>

Il progetto è promosso anche sui social networks. Debian, di per sé, ha delle pagine ufficiali sulle piattaforme realizzate con free software (come ad esempio sulla piattaforma microblogging Identि.ca, promossa da pump.io), ma allo stesso tempo ci sono molti accounts che condividono i contenuti degli RSS feeds di microneWS.debian.org e diversi contributori Debian che condividono a loro volta su accounts non ufficiali [per “accounts” il testo si riferisce a siti social network].

- ◆ <https://identi.ca/debian>
- ◆ <https://fosstodon.org/@debian>
- ◆ <https://twitter.com/debian>
- ◆ <https://www.facebook.com/debian>
- ◆ <https://www.flickr.com/groups/debian>
- ◆ <https://www.linkedin.com/company/debian>

1.5 Il Ruolo delle Distribuzioni

Una distribuzione GNU/Linux ha due principali obiettivi: installare un sistema operativo free su un computer (con o senza un sistema operativo nativo) e fornire una collezione di software in grado di soddisfare tutte le necessità dei suoi utenti.

1.5.1 Il programma che installa Debian: debian-installer

Il debian-installer, è stato progettato per essere estremamente modulare al fine di renderlo generico ed accessibile, ricalcando l’obiettivo cardine del progetto, ossia gli utenti. Il suddetto programma copre una vasta serie di contesti d’installazione ed in generale, facilita parecchio la creazione di un installer per le derivate, al fine di soddisfare qualsiasi esigenza specifica.

La struttura modulare di questo strumento, che lo rende anche molto complesso, potrebbe scoraggiare gli sviluppatori neofiti; ciononostante garantisce la stessa user experience (esperienza utente) sia se usato in modalità grafica (graphical mode), sia se usato in modalità testuale (text mode). [“L’esperienza utente” è una espressione entrata anche nel gergo comune italiano anche se in questo caso per esperienza non si intende una mera conoscenza bensì si riferisce alla praticità dell’installer di Debian per gli utenti]. Grandi sforzi sono stati compiuti pur di ridurre le richieste avanzate dal sistema durante l’installazione, in particolare grazie all’introduzione di un software che rileva automaticamente l’hardware.

È importante far presente che le derivate differiscono molto da Debian per quest’ultimo aspetto, in quanto l’installer da queste fornito è meno compatibile (spesso addirittura limitato alle architetture i386 o amd64), ma più user-friendly (facile – traduzione non letterale) per i profani. D’altra parte abitualmente le derivate evitano di allontanarsi troppo dal contenuto del pacchetto (di debian-installer) così da poter, quanto più possibile, beneficiare della sua vasta collezione di software offerti, senza incappare nei suddetti problemi di compatibilità.

1.5.2. La libreria Software

Quantitativamente, Debian è il leader indiscusso sotto questo punto di vista, con oltre 28,000 pacchetti sorgente. Qualitativamente, la Debian Policy e il lungo periodo di test prima del rilascio della nuova versione stabile confermano la reputazione sulla sua stabilità e sulla sua sicurezza. In qualsiasi luogo, seppur remoto ma raggiungibile da internet, ogni cosa è disponibile online grazie ai numerosi worldwide mirror, con aggiornamenti emessi ogni sei ore.

Molti vendori al dettaglio offrono DVD-ROMs su internet a bassissimo prezzo (spesso al costo di produzione) delle “immagini” che sono scaricabili gratis. L’unico inconveniente è soltanto la bassa frequenza dei rilasci delle nuove versioni stabili (il loro sviluppo qualche volta comporta anche più di due anni), che ritarda l’inclusione del nuovo software.

La maggior parte dei nuovi programmi free software sono generalmente compatibili e trovano il loro perché con la versione di sviluppo, che ne consente un’installazione rapida. I programmi possono essere anche ricompilati per la versione stabile di Debian, se l’installazione nella testing dovesse richiedere di per sé troppi aggiornamenti per soddisfarne le dipendenze (leggite il capitolo 15, “Come creare un Pacchetto Debian” a pagina 448 per maggiori informazioni su questo argomento).

1.6. Il ciclo di vita di un rilascio

Il progetto avrà simultaneamente da tre a sei differenti versioni di ciascun programma, denominate Experimental, Unstable, Testing, Stable, Oldstable, e anche Oldoldstable. Ciascuna corrisponde ad una differente fase di sviluppo.

Per una migliore comprensione, consentiteci di approfondire le fasi di progettazione, dall’iniziale packaging sino all’inclusione nella versione stabile di Debian.

DIZIONARIO

Release

Il termine “release” (che letteralmente in inglese significa “rilascio”), nel progetto Debian, può indicare sia una particolare versione di una distribuzione (ad esempio “unstable release”, che in questo caso significa non letteralmente “versione instabile”), sia l’annuncio ufficiale del lancio di una qualsiasi nuova versione stabile di Debian (stable).

1.6.1. Lo Status di Experimental (condizione)

Innanzitutto consentiteci di soffermarci su tutto ciò che ruota nello specifico intorno alla distribuzione Experimental: il suo omonimo gruppo di Pacchetti Debian, corrisponde al software correntemente in fase di sviluppo, e non necessariamente completo, da cui chiaramente deriva il suo nome. Non è obbligatorio attraversare questa fase; qualche sviluppatore include i pacchetti nella suddetta distribuzione per ricevere commenti da utenti più esperti (o più eccelsi).

È superfluo aggiungere, che questa distribuzione immagazzina le rilevanti modifiche effettuate sui pacchetti base, dei quali l’integrazione nella distribuzione Unstable (già con bugs gravi) comporterebbe ripercussioni critiche. Ciò la rende una distribuzione radicalmente a se stante ed i suoi pacchetti non migreranno mai ad un’altra versione (tranne nel caso di diretto ed espresso intervento del maintainer o degli ftpmasters). Inoltre non è indipendente: nell’Experimental è presente solo un sottoinsieme dei pacchetti correnti, senza includere generalmente il sistema base. Quindi questa distribuzione è per lo più utile in combinazione con un’altra distribuzione indipendente, come la stessa Unstable.

1.6.2. Lo Status di Unstable (condizione)

Permetteteci di riprendere un argomento precedentemente trattato ovvero l’evoluzione di un pacchetto standard in Debian. Il maintainer crea un pacchetto (iniziale), che viene compilato per la versione Unstable e collocato sul server ftp-master.debian.org. Questo primo intervento richiede il controllo e la validazione degli ftpmasters. Il software è poi distribuito nella distribuzione Unstable,

che è l’“edizione ridotta” delle distribuzioni Debian, scelta da quegli utenti che sono più ansiosi di conoscere i nuovi pacchetti, piuttosto di essere preoccupati dei bugs gravi. Gli utenti, così facendo, scoprono il programma e lo testano.

Se gli utenti incontrano dei bugs, li segnalano al maintainer del pacchetto. Il maintainer prepara, con regolarità, le versioni revisionate, da caricare sul server.

Ogni pacchetto aggiornato di recente è caricato su tutti i mirrors Debian sparsi nel mondo entro sei ore. Gli utenti poi testano le correzioni e verificano la presenza di altre problematiche derivanti dalle nuove modifiche. Di conseguenza diversi aggiornamenti possono susseguirsi rapidamente. Durante queste fasi, i robots autobuilder entrano in azione. Molto spesso, il maintainer ha un unico pc tradizionale e, pertanto, compila il pacchetto solo per un tipo di architettura (le più diffuse sono amd64 e i386) oppure si limita a caricare la sorgente, senza alcun pacchetto precompilato; gli autobuilders estraggono il pacchetto ed automaticamente lo compilano per le altre architetture. La compilazione degli autobuilders potrebbe fallire; qualora ciò accadesse il maintainer riceverà un bug report sul disservizio, che sarà perciò ripristinato nella prossima versione. Qualora il bug venisse scoperto da un esperto dell’architettura afflitta dal problema, questi potrà sempre allegare una patch, già pronta all’uso, al bug report.

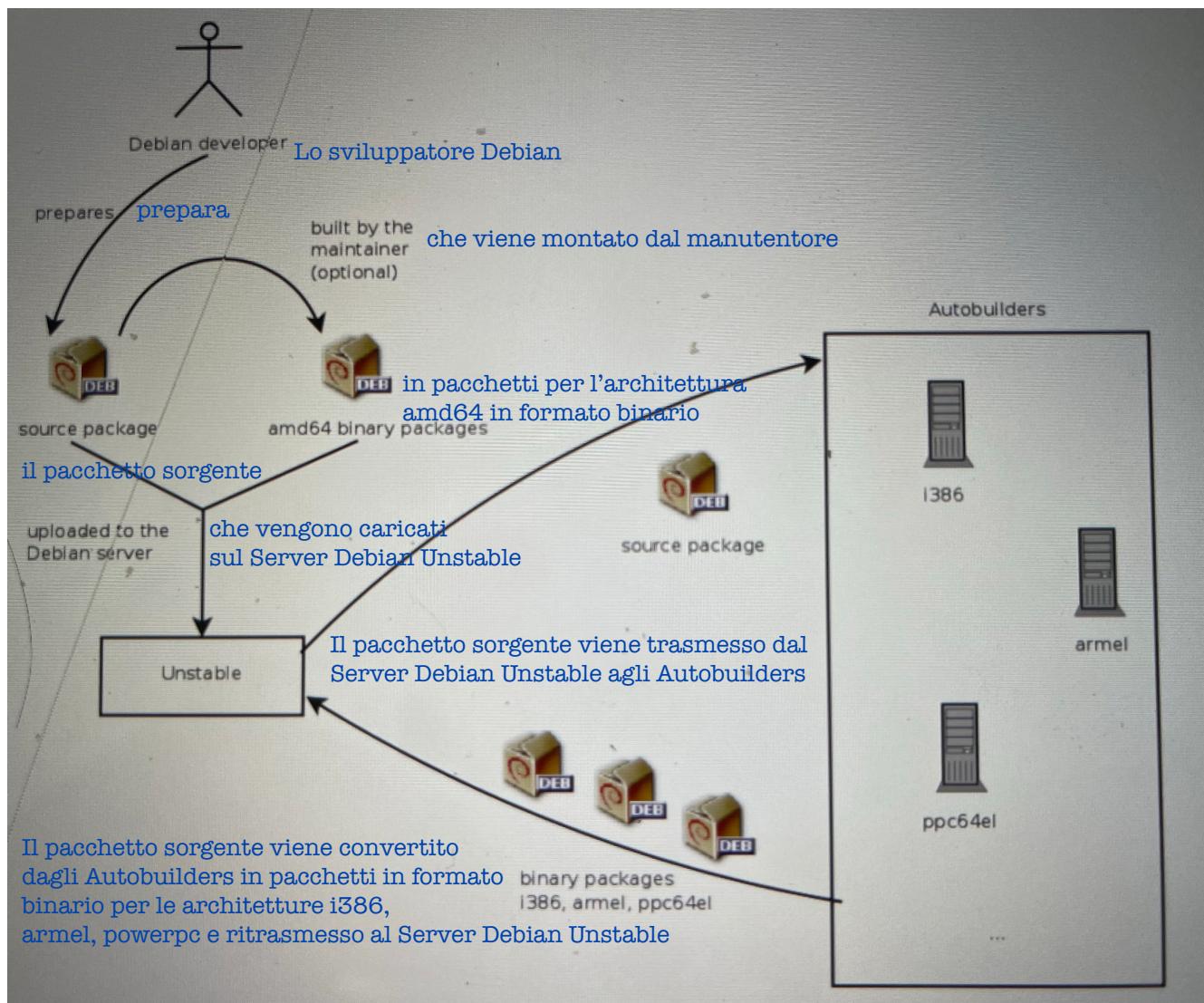


Figura 1.2 Compilazione di un pacchetto da parte degli autobuilders

BREVE ACCENNO buildd, il pacchetto Debian recompiler

buildd è l'abbreviazione di "build daemon". Questo programma ricompila automaticamente le nuove versioni dei pacchetti Debian per le architetture sulle quali è installato (la compilazione incrociata – la compilazione di un programma attraverso un computer di architettura simile, ma diversa rispetto a quella in cui il programma in questione è installato – è evitata per quanto possibile). Pertanto, per produrre i pacchetti in formato binario per l'architettura arm64, il progetto arm64 ha delle macchine appropriate a disposizione. Il programma buildd rimane avviato nelle suddette macchine continuamente e crea pacchetti in formato binario per l'architettura arm64 dai pacchetti sorgente inviati dagli sviluppatori Debian. Questo software è usato su tutti i computer impiegati nel progetto Debian come autobuilders. Per estensione, il termine buildd è adoperato frequentemente quando ci si riferisce a queste macchine, che sono solitamente riservate esclusivamente a questo scopo.

1.6.3. Il trasferimento (migrazione) a Testing

Poco dopo la summenzionata fase, il pacchetto sarà ulteriormente perfezionato; sarà compilato fino a che non ci sarà più necessità di sottoporlo ad ulteriori revisioni. Sarà proposto per l'inclusione nella distribuzione Testing – un gruppo di pacchetti della distribuzione Unstable selezionati in base a dei criteri computazionali. In questo modo ogni giorno un programma sarà automaticamente selezionato per essere incluso nella distribuzione Testing, a patto che questi non trascuri quei fattori che garantiscono un indubbio livello di qualità:

1. il pacchetto deve essere privo di bugs critici o perlomeno devono essere inferiori rispetto alla versione corrente inclusa nella Testing;
2. il pacchetto deve aver trascorso almeno 5 giorni nella distribuzione Unstable, periodo da ritenersi sufficiente per scovare e segnalare qualsiasi anomalia grave (il superamento della suite di test, se disponibile, dello stesso pacchetto riduce i tempi prescritti);
3. il pacchetto deve essere compilato con esito positivo per tutte le architetture supportate ufficialmente;
4. le dipendenze del pacchetto devono poter trovare supporto ed essere soddisfatte nella distribuzione Testing o quanto meno poter essere trasferibili nella distribuzione Testing insieme al pacchetto in questione durante la sua migrazione;
5. i tests automatici di qualità del pacchetto (`autopkgtest`) - se definiti - non devono rilevare alcuna regressione.

Il suddetto sistema non è chiaramente infallibile; i bugs critici sono scovati regolarmente nei pacchetti inclusi nella distribuzione Testing. Ciò nonostante è ancora in vigore, e la distribuzione Testing pone di gran lunga meno problemi dell'Unstable, divenendo per molti, un buon compromesso fra stabilità e novità.

COMUNITÀ

Il Release
Manager

Il Release Manager è un incarico di rilievo, a cui sono affidate gravose responsabilità. Chi si assume tale incarico deve, difatti, coordinare il rilascio di una nuova versione stabile di Debian e fissare il processo di sviluppo della distribuzione Testing in modo che raggiunga i criteri di qualità della summenzionata versione Stable. I Release Managers fissano anche una tabella di marcia provvisoria (non sempre seguita). Esistono anche i Stable Release Managers, spesso menzionati con l'abbreviazione SRM, che coordinano e selezionano gli aggiornamenti per la versione stabile di Debian in corso. Questi sistematicamente includono le patches di sicurezza e valutano, caso per caso, tutte le altre inclusioni suggerite, inviate dagli sviluppatori Debian desiderosi di aggiornare i loro pacchetti nella versione stabile.

NOTA I limiti della distribuzione Testing	<p>Per quanto possa essere un progetto interessante in linea di principio, la distribuzione Testing presenta alcune problematiche prettamente pratiche: la matassa delle dipendenze incrociate fra i pacchetti è tale che raramente un pacchetto può esservi trasferito in condizioni ottimali. A causa dell'interdipendenza dei pacchetti fra loro, qualche volta sarebbe necessario trasferire un vasto numero di pacchetti simultaneamente, ma ciò è impossibile dato che parecchi sviluppatori caricano aggiornamenti costantemente. D'altra parte, lo script, che riconosce le famiglie [in questo caso per famiglia si intende una grande classificazione di pacchetti, superiore al genere ed inferiore all'ordine] dei correlati pacchetti, lavora sodo per richiamarle (questo è un sedicente problema NP-completo, per il quale fortunatamente, conosciamo molto bene le euristiche).</p> <p>[Nella teoria della complessità computazionale, un problema è “NP-completo” quando può essere risolto solo attraverso una limitata classe di algoritmi “brute force search”, denominati anche “ricerca esaustiva” o “metodo forza bruta”, e può essere utilizzato per simulare qualsiasi altro problema con un algoritmo simile.</p> <p>“Il metodo forza bruta” è un algoritmo di risoluzione di un dato problema e consiste nel verificare tutte le soluzioni teoricamente possibili fino quando non si trova quella effettivamente corretta.</p> <p>Per “euristica” si intende quel complesso di tecniche e di metodi della ricerca scientifica o filosofica per giungere ad una soluzione].</p> <p>Affermiamo ciò in quanto possiamo interagire e indirizzare lo script manualmente suggerendogli i gruppi dei pacchetti, o obbligandolo all'inclusione di certi pacchetti nel gruppo, anche se ciò rompe nel contempo parecchie dipendenze. Questa procedura è concessa ai Release Managers ed ai loro assistenti. Si ricorda che un problema NP-completo è di una complessità algoritmica esponenziale a seconda della mole di dati, ovvero nel nostro caso in base alla lunghezza del codice (in termini di cifre numeriche) ed agli elementi da coinvolgere. Nella maggior parte dei casi l'unico metodo per risolverlo è di valutare tutte le configurazioni potenziali, e ciò potrebbe o non potrebbe esigere ingenti mezzi. Pertanto l'euristica è una soluzione approssimativa, ma soddisfacente.</p>
--	---

1.6.4. L'avanzamento (conversione) da Testing a Stable

Poniamo che il nostro pacchetto, dopo i suddetti passaggi, sia ora incluso nella Testing. Qualora ce ne fosse ancora la possibilità, il suo manutentore deve continuare a migliorarlo, ricominciando il procedimento a partire dall'Unstable (sebbene le sue successive inclusioni nella Testing siano più repentine: difatti a meno che non cambi significativamente, tutte le sue dipendenze sono solitamente pronte per essere soddisfatte). Quando il pacchetto raggiunge la sua perfezione, si può ritenere ultimato l'incarico del manutentore. La fase successiva prevede l'inclusione nella distribuzione Stable, che è, di fatto, la semplice copia di una versione della distribuzione Testing scelta appositamente dal Release Manager. Teoricamente questa decisione viene presa quando l'installer è pronto, e quando nessun programma nella Testing presenta bugs critici noti.

In realtà in questa fase niente giunge davvero esente da problemi, di conseguenza, Debian deve transigere: rimuovere i pacchetti dei quali la correzione dei bugs non è stata ultimata dal relativo manutentore in tempo, oppure accettare di rilasciare una distribuzione con parecchi bugs in migliaia di programmi. Il Release Manager, pertanto, annuncerà una procedura denominata “freeze period” [“freeze”, in inglese, in senso figurato significa “rimanere immobile per il gelo”], durante la quale qualsiasi aggiornamento della Testing dovrà essere approvato. L'obiettivo di tale procedura è di impedire qualsiasi nuova versione (con conseguenti nuovi bugs) e di ratificare solo gli aggiornamenti in cui sono stati corretti i bugs.

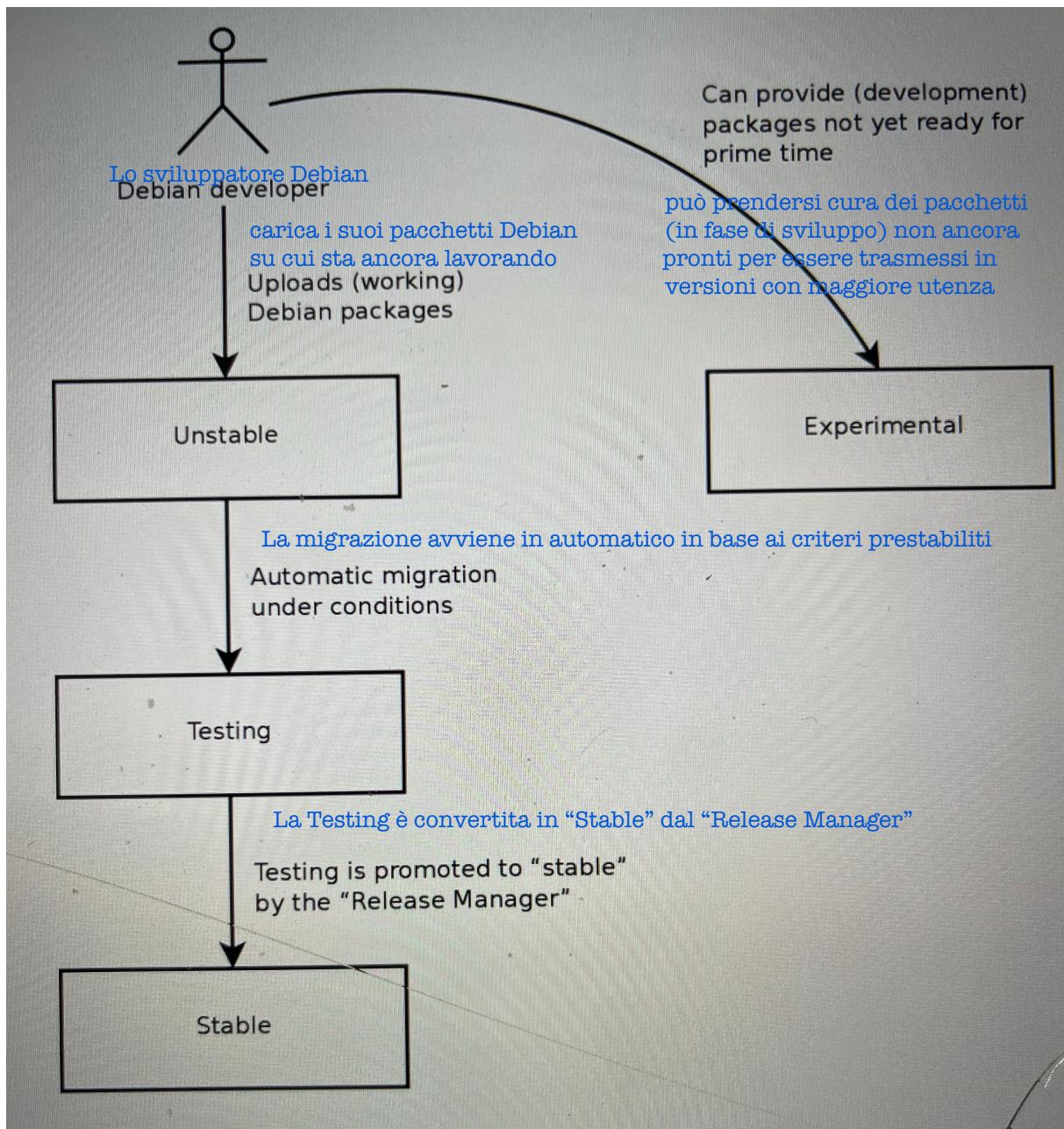


Figura 1.3 Il percorso procedurale e la conseguente evoluzione di un pacchetto attraverso le diverse versioni Debian

DIZIONARIO

Freeze: la fase che precede il vero traguardo, ossia la Stable

Durante il "freeze period", lo sviluppo della distribuzione Testing è bloccato; gli aggiornamenti automatici non sono consentiti. Solo i Release Managers sono pertanto autorizzati a modificare i pacchetti, secondo i loro stessi criteri. Lo scopo di ciò è di scongiurare la comparsa di nuovi bugs derivanti da nuove versioni della distribuzione; sono ammessi solo gli aggiornamenti valutati accuratamente ossia quelli in cui sono stati corretti bugs rilevanti.

Dopo il rilascio di una nuova versione stabile, il Stable Release Manager coordina il susseguente progetto di sviluppo (definito "revision", in italiano revisione – ad esempio le revisioni 7.1, 7.2, 7.3

della versione ?). Questi aggiornamenti includono sistematicamente tutte le patches di sicurezza. Inoltre includeranno anche le più importanti correzioni (i manutentori dei pacchetti devono però dimostrare la gravità della problematica che intendono correggere per poter fare includere i propri aggiornamenti).

Conclusa la suddetta procedura, il nostro ipotetico pacchetto sarà finalmente incluso nella distribuzione stabile.

Inoltre la summenzionata procedura, non esente da difficoltà, chiarisce l'indispensabile lasso di tempo ed i conseguenti significativi ritardi fra un rilascio e l'altro della distribuzione Stable.

Ciò contribuisce, soprattutto, alla sua reputazione di qualità. Per di più, la maggior parte degli utenti si accontenta di utilizzare soltanto una delle tre distribuzioni contemporaneamente disponibili.

Difatti gli amministratori di sistema, particolarmente preoccupati della stabilità dei loro servers, non avranno necessità della più recente ed apprezzata versione di GNOME; e per poter soddisfare le loro esigenze avranno la possibilità di scegliere Debian Stable.

Diversamente gli end-users, più interessati all'ultima versione di GNOME o KDE Plasma piuttosto che ad una stabilità "solida come una roccia", troveranno in Debian Testing un giusto compromesso fra esigui disagi e software relativamente aggiornato.

[Per end-user o utilizzatore finale, o anche utente finale, si intende un soggetto, non necessariamente una persona fisica, che, nel processo di progettazione e produzione di un determinato prodotto o servizio, si trova al termine della catena di produzione-commercio-utilizzo, ovvero che è il destinatario finale dell'oggetto in questione e che ne beneficia dell'uso.]

Infine gli sviluppatori e gli utenti più esperti potranno aprire la strada agli altri, testando tutti gli ultimi aggiornamenti di Debian Unstable appena sfornati, al prezzo di grattacapi e bugs tipici di qualsiasi nuova versione di un programma. A ciascuno il proprio Debian!

CULTURA	GNOME (GNU Network Object Model Environment) e Plasma di KDE (precedentemente nota come K Desktop Environment) sono i due più popolari ambienti grafici desktop nel mondo free software, di cui troverete maggiori informazioni nel paragrafo 13.3 Graphical Desktops a pag. 385. Un ambiente desktop è un complesso di programmi messi assieme per trasmettere una padronanza intuitiva delle più ordinarie funzioni per mezzo di un ambiente grafico. Generalmente queste racchiudono un file manager, una suite office, un web browser, un programma di gestione email, strumenti multimediali, ecc. La differenza più evidente fra gli ambienti grafici risiede nella scelta della libreria grafica da questi usata: GNOME ha scelto GTK+ (licenza free software sotto LGPL) e la comunità KDE ha preferito Qt (un progetto supportato da una società, disponibile oggigiorno sotto licenza che è allo stesso tempo sia GPL, che commerciale). ♦ https://www.gnome.org/ ♦ https://www.kde.org/
GNOME e KDE Plasma, gli ambienti grafici desktop	

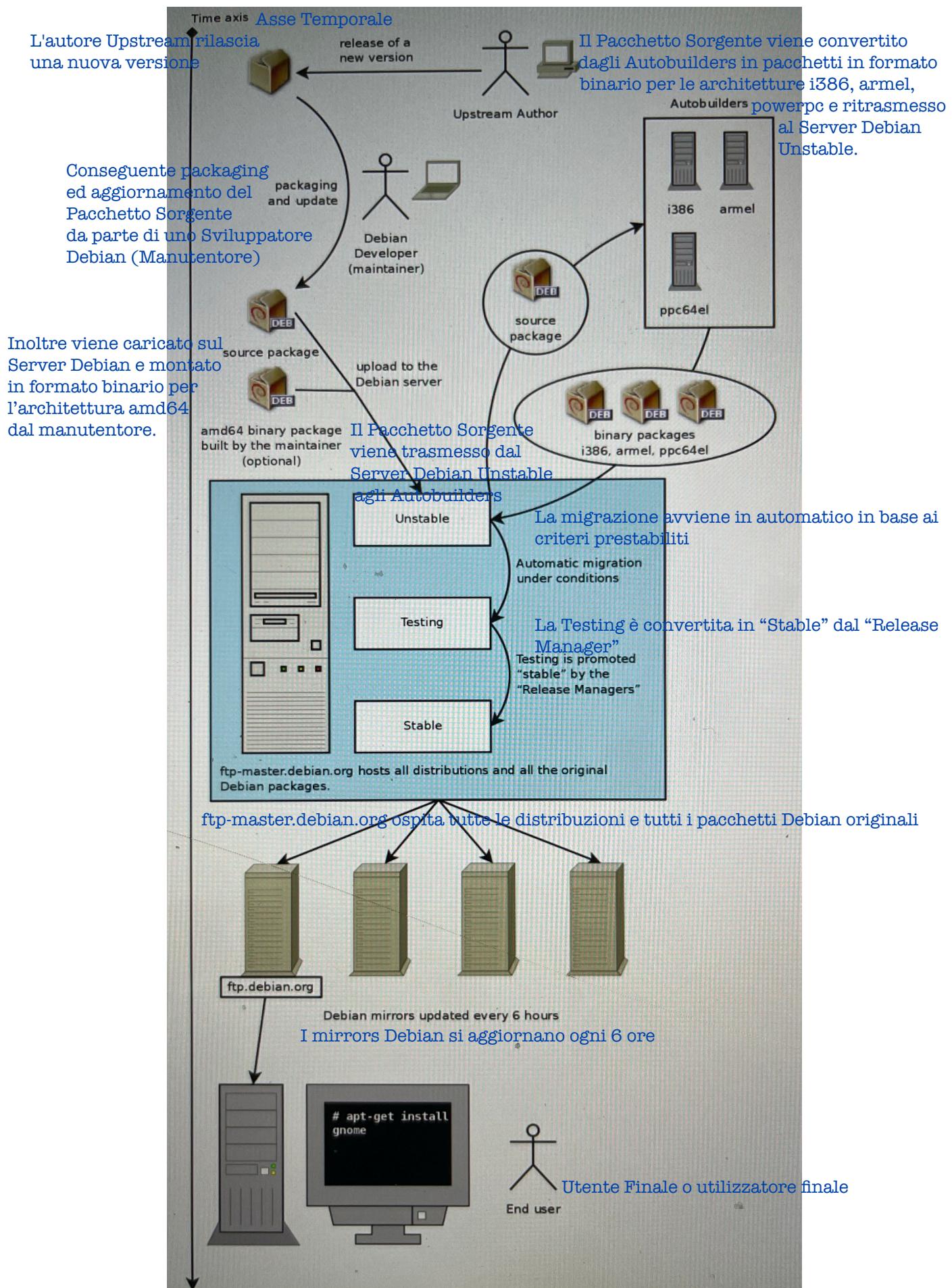


Figura 1.4 Packaging Debian – Il percorso procedurale e cronologico di un pacchetto Debian

1.6.5 Gli status (condizioni) di Oldstable e di Oldoldstable

Ogni distribuzione Stable ha un'aspettativa di vita di circa 5 anni e dato che i rilasci tendono a succedersi ogni 2 anni, possono essere in attività sino a 3 rilasci nel medesimo arco temporale. Quando un nuovo rilascio Stable subentra, le precedenti versioni Stable ed Oldstable diventano rispettivamente Oldstable ed Oldoldstable.

Il supporto a lungo termine (Long Term Support – LTS) è un'iniziativa recente: i singoli contributori e le società hanno unito le loro forze per dare vita al team Debian LTS. I più vecchi rilasci ancora supportati, ma non per molto, dal team Debian security ricadono sotto la responsabilità del suddetto nuovo team.

Il team Debian security dirige il supporto alla sicurezza della versione Stable in corso ed anche della rispettiva versione Oldstable (ma solo per 1 anno, perché è indispensabile garantire il supporto durante l'anno di sovrapposizione con la corrente versione Stable). Ciò significa, in poche parole, tre anni di supporto ad ogni rilascio [a partire dal suo status di Stable]. Il Debian LTS team dirige gli ultimi (due) anni di supporto alla sicurezza, così che ciascuna versione usufruisca di fatto, dal suo rilascio con lo status di Stable sino alla conclusione del suo ciclo di vita, di 5 anni di supporto e gli utenti, di conseguenza, possano effettuare l'aggiornamento dalla versione N alla versione N+2, ad esempio da Debian 8 (Jessie) a Debian 10 (Buster).

♦ <https://wiki.debian.org/LTS>

[In sintesi ed approssimativamente – La Stable usufruisce di 2 anni di supporto del team Debian security. La Oldstable usufruisce di 1 anno di supporto del team Debian security e di 1 anno di supporto del team Debian LTS. La Oldoldstable usufruisce solo di 1 anno di supporto del team Debian LTS terminato il quale conclude il suo ciclo di vita. In totale sono 5 anni di vita per ciascun rilascio a partire dal suo status di Stable di cui: 2 anni con lo status di Stable; 2 anni con lo status di Oldstable; 1 anno con lo status di Oldoldstable.]

COMUNITÀ

Le società che sostengono l'impresa del team LTS

Il Supporto a Lungo Termine è un difficile impegno da mantenere nel progetto Debian in quanto i suoi volontari tendono ad evitare gli incarichi noiosi. Per molti contributori, difatti, prendersi cura per 5 anni del supporto alla sicurezza del software non recente è meno divertente del packaging delle nuove versioni upstream o dello sviluppo di nuove funzionalità.

Per portare alla luce questo team, il progetto Debian ha contato sul fatto che il supporto a lungo termine fosse particolarmente rilevante per le società e che queste a loro volta sarebbero state interessate a concorrere ai costi del supporto alla sicurezza.

Il progetto per la formazione di questo team e la realizzazione del relativo servizio è iniziato nel giugno 2014: parecchie società hanno permesso ai loro impiegati di lavorarci part-time mentre altre hanno preferito sostenere il progetto economicamente in modo che i contributori Debian, non desiderosi di occuparsene spontaneamente e gratuitamente, fossero invogliati da una retribuzione. La maggior parte dei contributori desiderosi di essere pagati per il loro operato nel servizio LTS si sono uniti per realizzare una offerta di supporto dettagliata gestita dalla Freexian (società di Raphaël Hertzog):

♦ <https://www.freexian.com/services/debian-lts.html>

Nel team Debian LTS, i volontari si occupano solo dei pacchetti di cui hanno interesse, mentre per i contributori retribuiti hanno precedenza assoluta i pacchetti utilizzati dalle società che li sostengono economicamente. Il progetto è sempre alla ricerca di nuovi sponsors: di cosa si occupa la vostra società? Potete consentire ad un vostro dipendente di occuparsi part-time del supporto a lungo termine? Potere destinare una piccola quota del vostro budget per il supporto alla sicurezza?

♦ <https://wiki.debian.org/LTS/Funding>

Parole chiave
Falcot Corp
SMB
Crescita Sostenuta
Master Plan
Migrazione
Riduzione dei Costi

