Patch rejected – 12 Gründe, warum dein Patch immer abgelehnt wird

FrOSCon 2018

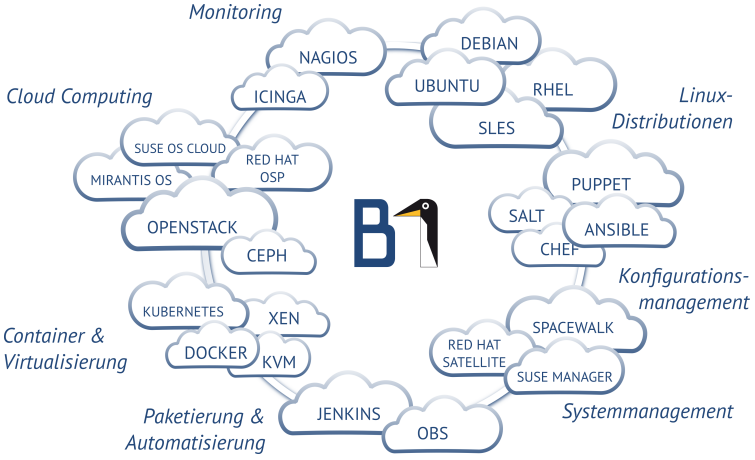
26. August 2018

Ralf Lang
Linux Consultant & Developer
B1 Systems GmbH
lang@b1-systems.de

Vorstellung B1 Systems

- gegründet 2004
- primär Linux/Open Source-Themen
- national & international tätig
- über 100 Mitarbeiter
- unabhängig von Soft- und Hardware-Herstellern
- Leistungsangebot:
  - Beratung & Consulting
  - Support
  - Entwicklung
  - Training
  - Betrieb
  - Lösungen
- Standorte in Rockolding, Köln, Berlin & Dresden

Schwerpunkte



Begriffserklärung

Begriffserklärung 1/2

Patch: Änderung einer Software
- um Fehler zu beseitigen oder
- Funktionen beizutragen

Upstream das offizielle Projekt-Repository, z. B. bei GitHub, GitLab

Downstream eine abgeleitete Quelle, z. B. ein RPM-Repository, das veränderte Software enthalten kann

Begriffserklärung 2/2

Fork: eine Kopie des Upstream-Codes, die Änderungen enthalten kann

Pull Request: Anfrage an Upstream, eine Änderung aufzunehmen

Core-Entwickler: Entwickler mit direktem Zugriff auf das Upstream-Repository

Typischer Ablauf

Typischer Ablauf 1/2

- Alice findet einen Fehler in Software oder Dokumentation oder will eine Funktion einbauen
- Alice erstellt eine Entwicklerkopie (Fork) des Upstream-Repositories
- Alice baut ihre Änderungen in die Entwicklerkopie ein
- Alice stellt ihre Änderungen entweder als Patchdatei per E-Mail oder als Pull-Request bereit

Typischer Ablauf 2/2

- Core-Entwickler Bob schaut sich den Patch an und hat Kritik
- Core-Entwickler Bob lehnt den Pull-Request ab oder bittet um Nachbesserung
- Alice ärgert sich
- Alice sendet irgendwann eine verbesserte Version
- Core-Entwickler Bob schaut, ob jetzt alles gut ist
- nach 1..nnnn Runden nimmt Core-Entwickler Bob den Patch an und bedankt sich bei Alice

Wer sendet Pull Requests?

- Core-Entwickler bauen kontinuierlich an der Software
- Distributions-Entwickler (z. B. openSUSE, Fedora) tragen zur Software bei
- Administratoren und Benutzer entdecken Bugs
- Firmen bauen Erweiterungen und geben diese an Upstream zurück, z. B. Linux-Kernel
- externe Consultants helfen Firmen, Software für ihre Zwecke zu verbessern

Warum verbessert der Core-Entwickler nicht einfach selber, was er nicht mag?

- keine Zeit: Bob will selbst entwickeln und bekommt viele Anfragen
- kein Interesse: die Änderung betrifft seine Art, die Software zu benutzen, nicht sehr
- Kritik soll Alice helfen, eine bessere Entwicklerin zu werden
- manchmal akzeptiert Bob kleinere Abweichungen oder korrigiert sie selbst

Mögliche Gründe für die Ablehnung

Grund 1: Coding Standards und Einrückungsregeln

Coding Standards und Einrückungsregeln

- Software-Projekte halten sich an selbst definierte oder allgemein gebräuchliche Standards
- Wie viele Leerzeichen oder Tabs zum Einrücken? Klammern auf gleiche oder nächste Zeile?
- Welche Operatoren und Idiome sind zu bevorzugen, auszuschließen?
- Einheitliches Codeformat verbessert die Lesbarkeit und Verständlichkeit!

Beispiele

Einrückungsregeln („Indention Styles“)

- One True Brace Style (1TBS), auch bekannt als „West Coast“ oder „Kernel Style“
- Allman/BSD, auch bekannt als „East Coast“

Coding Standards

- PHP-FIG PSR-2 Coding Style Guide
  `https://www.php-fig.org/psr/psr-2/`
- Horde Coding Standard
  `https://www.horde.org/apps/horde/docs/CODING_STANDARDS`

Lösungen

- Alice konfiguriert ihren Editor passend
- Alice macht Whitespace-Zeichen nach dem letzten Druckzeichen sichtbar
- ein guter Editor kann Tabstops in eine gewünschte Anzahl Leerzeichen umwandeln
- Alice benutzt einen automatischen Linter/Checker, z. B. `php-CodeSniffer`

Grund 2: Das Changelog

Das Changelog

- das Changelog ist eine menschenlesbare Zusammenfassung, in welcher Version welche Änderung einging
- viele Softwareprojekte haben Regeln, ob der Committer Alice oder der Core-Entwickler Bob das Changelog pflegt
- auch das Format ist oft festgelegt: Präsens oder Vergangenheitsform, Strichpunkt oder Satz, mit oder ohne Punkt
- Nennung des Beitragenden ab einer gewissen „Fallhöhe“
- oft in YAML-Format oder durch Programme in json oder XML-Format generiert, z. B. bei PHP-Pear-Paketen
- das Changelog ist eine wichtige Grundlage für die Arbeit von Distributionsentwicklern

Lösungen

- Alice liest und versteht die Changelog-Regeln des Projektes
- Alice findet keine Regeln, liest das bisherige Changelog und leitet daraus ab, was vermutlich akzeptiert wird

Grund 3: Unit Tests fehlen oder scheitern in der CI

Unit Tests fehlen oder scheitern in der CI

- Softwareprojekte benutzen *Unit Tests*, um die korrekte Funktion von einzelnen Komponenten zu überprüfen
- viele Projekte verlangen bei Bugfixes zwingend Unit Tests, die den Bug nachweisen und zeigen, für welche Randbedingungen er gelöst ist
- die Unit-Test-Plattform ist verbindlich,
  z. B. JUnit, phpunit, Perl Test::More
- die Unit Tests werden meist automatisiert auf einer *Continuous Integration Plattform* (CI) bei jeder Änderung ausgeführt und decken verschiedene Standardsysteme ab
- bekannte CIs sind Travis (GitHub), Gitlab-CI (GitLab), Jenkins und Concourse

Beispiele

test/Horde/Auth/Unit/CascadingMockTest.php

```
52 +    public function testTransparent()
53 +    {
54 +        //throw exception if no backend provides transparent
55 +        $this->setExpectedException(Horde_Auth_Exception::class);
```

yunosh 16 days ago **Owner**

This make the Travis test fail on PHP 5.4.

Abbildung: `https://github.com/horde/Auth/pull/3`

- `https://github.com/horde/horde/pull/173`
- `https://github.com/chef/chef/pull/6476`

Lösungen

- Alice schreibt erst den Unit Test, der den Fehler für einen oder mehrere Randfälle nachweist
- Alice schreibt dann erst den Patch, der möglichst genau nur den Fehler behebt und sonst nichts ändert
- bei komplizierten Themen erklärt Alice sehr genau, warum das gezeigte Verhalten ein Fehler ist
- oft ist es auch ausreichend, nur den Unit Test einzureichen und den Fehler als Bug zu melden

Grund 4: Fehlende Code-Dokumentation

Fehlende Code-Dokumentation

- viele Sprachen haben einen De-Facto-Standard für Code-Dokumentation (z. B. perldoc, phpdoc, doxygen)
- Code-Dokumentation hilft Anwender besser verstehen, welche Parameter in welchem Format eine Funktion oder Methode akzeptiert
- aus speziell formatierten Kommentaren lassen sich Online-Dokumentationen erstellen, die man im Browser lesen kann

Lösungen

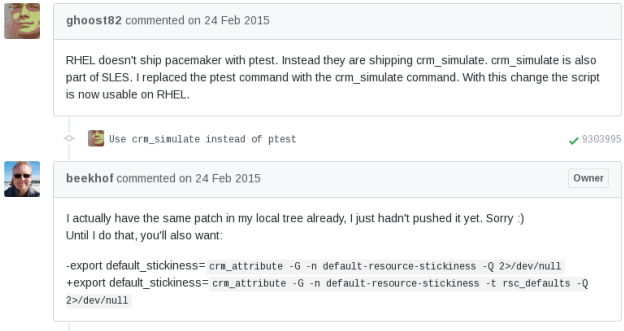
- Alice ergänzt Code-Dokumentation

Grund 5: Alice war zu langsam

Alice war zu langsam

- oft entdecken viele Benutzer zeitgleich ein Problem in einer neuen Version
- jemand anderes hat seine Lösung etwas schneller eingereicht oder seine ist etwas besser

Beispiele 1/2



ghoost82 commented on 24 Feb 2015

RHEL doesn't ship pacemaker with ptest. Instead they are shipping crm_simulate. crm_simulate is also part of SLES. I replaced the ptest command with the crm_simulate command. With this change the script is now usable on RHEL.

Use crm_simulate instead of ptest 9303995

beekhof commented on 24 Feb 2015 Owner

I actually have the same patch in my local tree already, I just hadn't pushed it yet. Sorry :)
Until I do that, you'll also want:

-export default_stickiness=`crm_attribute -G -n default-resource-stickiness -Q 2>/dev/null`
+export default_stickiness=`crm_attribute -G -n default-resource-stickiness -t rsc_defaults -Q 2>/dev/null`

Abbildung: https://github.com/ClusterLabs/pacemaker/pull/648

Beispiele 2/2

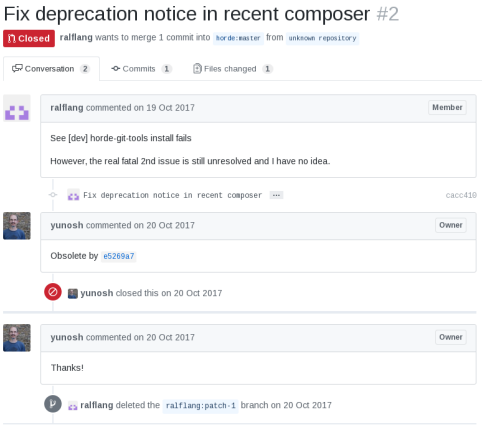


Abbildung: https://github.com/horde/git-tools/pull/2

Lösungen

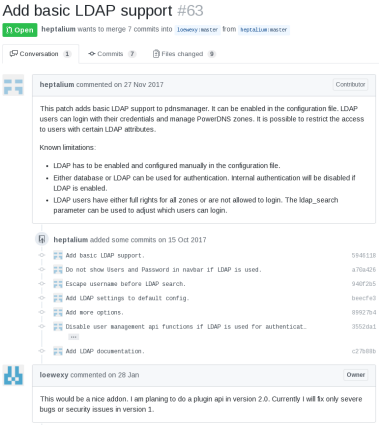
- Alice freut sich
- Alice fängt keinen Streit über die bessere Lösung an – Upstream hat schon entschieden

Grund 6: Feature passt nicht zur Architektur

Feature passt nicht zur Architektur

- Alice hat eine neue Funktion gebaut, doch die Funktion passt nicht in die Architektur der Anwendung
- eine größere Umstrukturierung (Refactoring) ist bereits geplant
- das Feature behandelt einen Nebenaspekt der Anwendung

Beispiel



Add basic LDAP support #63

Open heptalium wants to merge 7 commits into loewexy:master from heptalium:master

Conversation 1 · Commits 7 · Files changed 9

heptalium commented on 27 Nov 2017 — Contributor

This patch adds basic LDAP support to pdnsmanager. It can be enabled in the configuration file. LDAP users can login with their credentials and manage PowerDNS zones. It is possible to restrict the access to users with certain LDAP attributes.

Known limitations:

- LDAP has to be enabled and configured manually in the configuration file.
- Either database or LDAP can be used for authentication. Internal authentication will be disabled if LDAP is enabled.
- LDAP users have either full rights for all zones or are not allowed to login. The ldap_search parameter can be used to adjust which users can login.

heptalium added some commits on 15 Oct 2017

- Add basic LDAP support. 5946118
- Do not show Users and Password in navbar if LDAP is used. a70a426
- Escape username before LDAP search. 940f2b5
- Add LDAP settings to default config. beecfe3
- Add more options. 89927b4
- Disable user management api functions if LDAP is used for authenticat… 3552da1
  …
- Add LDAP documentation. c27b88b

loewexy commented on 28 Jan — Owner

This would be a nice addon. I am planing to do a plugin api in version 2.0. Currently I will fix only severe bugs or security issues in version 1.

Abbildung: https://github.com/loewexy/pdnsmanager/pull/63

Lösungen

- Alice baut das Feature neu als optionales Plugin, Addon oder Hook für die Grundsoftware
- Alice wartet das Refactoring ab und schreibt das Feature passend zur neuen Architektur neu

Grund 7: Änderung ist zu speziell

Änderung ist zu speziell

- bei der Anpassung von Software an bestehende Umgebungen sind Änderungen nötig
- Core-Entwickler Bob lehnt die Änderungen ab, weil sie für die Allgemeinheit nicht nützlich sind

Lösungen

- Alice verallgemeinert oder parametrisiert ihre Änderung
- Alice verwaltet einen Downstream-Patch, der dauerhaft nicht in die Grundsoftware eingeht

Beispiel

Source Files (show unmerged sources)

Show 25 entries — Search

Filename	Size	Changed	Actions
at-3.1.13-documentation-dir.patch	396 Bytes	8 months ago	
at-3.1.13-leak-fix.patch	725 Bytes	over 4 years ago	
at-3.1.13-massive_batch.patch	1.34 KB	8 months ago	
at-3.1.13-tomorrow.patch	627 Bytes	over 4 years ago	
at-3.1.14-joblist.patch	2.77 KB	8 months ago	
at-3.1.14-makefile-deps.patch	478 Bytes	about 4 years ago	
at-3.1.14-parse-suse-sysconfig.patch	2.59 KB	8 months ago	
at-3.1.14-usePOSIXtimers.patch	2.62 KB	8 months ago	
at-3.1.14.patch	6.63 KB	8 months ago	
at-3.1.16-handle_malformed_jobs.patch	1.41 KB	over 2 years ago	
at-3.1.8-denylist.patch	253 Bytes	over 7 years ago	
at-3.1.8-jobdir-mtime.patch	656 Bytes	almost 7 years ago	
at-adjust_load_to_cpu_count.patch	2.14 KB	over 2 years ago	
at-atq-timeformat.patch	2.63 KB	over 3 years ago	
at-backport-old-pnvs.patch	2.76 KB	over 4 years ago	
at-priddir.patch	244 Bytes	over 4 years ago	
at-secure_getenv.patch	1.16 KB	over 4 years ago	
at.changes	24.1 KB	3 months ago	
at.spec	5.05 KB	3 months ago	
at_3.1.20.orig.tar.gz	125 KB	8 months ago	
atd.pamd	251 Bytes	about 11 years ago	
atd.service	170 Bytes	almost 3 years ago	
sysconfig.atd	524 Bytes	over 2 years ago	

Showing 1 to 23 of 23 entries — Previous 1 Next

Latest Revision

buildservice-autocommit accepted request 545119 3 months ago (revision 105)
baserev update by copy to link target

Files changed | Browse Source

Abbildung: `https://build.opensuse.org/package/show/Base:System/at`

Grund 8: Patch passt nicht mehr

Patch passt nicht mehr

- Alice hat für ihren Patch viele Wochen gebraucht, bis er fertig entwickelt war
- Bob hat inzwischen andere Änderungen an derselben Codestelle gemacht
- die Versionsverwaltung kann die Änderung nicht mehr mit dem neuen Code zusammenfügen (Merge-Konflikt)
- passiert besonders häufig bei Meta-Dateien wie Paketmanifesten (Pear, Composer)

Beispiel

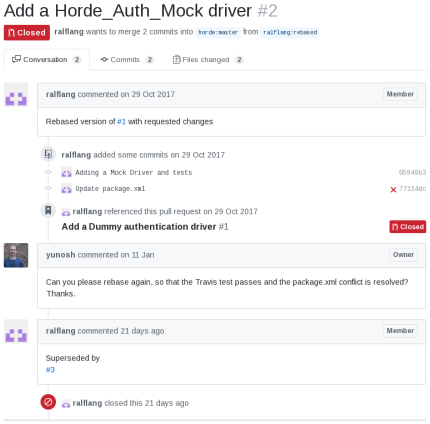


Abbildung: https://github.com/horde/Auth/pull/2

Lösungen

- Alice passt den Patch von Hand an den neuen Grund-Code an
- Alice generiert die Metadaten des Moduls neu und fügt ihr Changelog neu ein

Grund 9: Änderung ist zu umfangreich

Änderung ist zu umfangreich

- statistisch gesehen ist es wahrscheinlicher, dass ein Patch akzeptiert wird, wenn er wenige Zeilen umfasst und möglichst wenig ändert
- viele kleine Änderungen sind leichter zu verstehen als eine große Big-Bang-Änderung
- Merge-Konflikte sind für große zusammenhängende Änderungen wahrscheinlicher
- große Änderungen sind komplizierter zu bewerten und Core-Entwickler Bob braucht mehr Zeit auf einmal – Bob hat meist wenig Zeit
- die Core-Entwickler sind mit dem Ansatz oder der Architektur der Lösung möglicherweise nicht einverstanden

Lösungen – Einvernehmlich

- Alice diskutiert einen Lösungsansatz für eine grundlegende Erweiterung oder Änderung ausgiebig mit allen Core-Entwicklern
- Alice und Bob einigen sich, die Lösung zunächst in einem Feature-Branch aufzunehmen und dort zu stabilisieren und verfeinern
- Alice und Bob einigen sich, dass Alice bestimmte Features nur in einer Sonderversion (Fork) der Software für spezielle Nutzerkreise anbietet

Beispiel – Einvernehmliche Lösung

- „Enhanced SVN Builds“
  `https://www.dosbox.com/wiki/SVN_Builds`

Lösungen – Nicht so einvernehmlich

- Alice und Bob können sich über wichtige Fragen nicht einigen
- Alice macht sich mit einem dauerhaften Fork der Software selbständig
- manche Forks werden groß und wichtig, die meisten verlaufen im Sande

Beispiele – Nicht so einvernehmliche Lösung

→ Lizenzgründe, Firmenpolitik, allgemeine Projektorientierung

- OpenOffice-Fork LibreOffice
- MySQL-Fork MariaDB

→ Uneinigkeit über Entwicklungsrichtung des Projekts

- eGroupware-Fork Tine 2.0
- ownCloud-Fork Nextcloud

Grund 10: Neue Abhängigkeiten/Code Duplication

Neue Abhängigkeiten/Code Duplication

- große Softwareprojekte und Frameworks haben oft Abhängigkeiten zu vielen externen Bibliotheken
- für viele Teilprobleme bringen sie bereits eigenen Code mit
- externe Entwickler kennen sich mit den Details des Projekts nicht ausreichend aus
- Alice hat in ihrem neuen Login-Modul für ein LDAP-Backend eine neue Abhängigkeit geschaffen
- Bob lehnt die Änderung ab und verweist auf eigenen Code oder eine bereits bestehende Abhängigkeit
- verwendete Abhängigkeiten haben evtl. inkompatible Lizenzbedingungen

Beispiel

lib/Horde/Auth/Cascading.php

```
241 +                    $newPassword .= $characters[rand(0, $charactersLength - 1
242 +                }
243 +            }
244 +            $credentials = array('password' => $newPassword);
```

yunosh 16 days ago **Owner**

There already is Horde_Auth::genRandomPassword()

Abbildung: `https://github.com/horde/Auth/pull/3`

Lösungen

- Alice schreibt Teile ihres Codes um und benutzt die vom Projekt bereits benutzten Libraries
- Alice benutzt eine externe Library mit Lizenzbedingungen, die zum Projekt passen

Grund 11: Rechtsfragen

Rechtsfragen

- Alice hat einen Patch für eine führende Cloud-Software geschrieben, aber das Projekt akzeptiert Code nur, wenn Alice ein Contributor License Agreement unterschreibt
- ein anderes Softwareprojekt will jegliche Patches nur akzeptieren, wenn Alice das Copyright an ihrem Code dem Projekt überschreibt
- ein weiteres Softwareprojekt will eine schriftliche Bestätigung, dass auf dem Code keine Rechte Dritter liegen

Beispiel

Contributor License Agreement

For Individuals, See the Contributor License Agreement steps above

- Corporations
  - If you are contributing on behalf of a company or organization, you still need to sign the Individual CLA above but someone at your company with appropriate signing authority also needs to sign the Corporate Contributor License Agreement providing a list of people authorized to commit code to OpenStack. Request a printable copy by emailing the OpenStack Foundation. Check How to update the CCLA to provide changes to such list.
- US Government employees
  - Employees of the the U.S. Government do not sign the Individual CLA. Instead, someone with authority to sign on behalf of your agency should sign the U.S. Government Contributor License Agreement. Individual developers still need to create a Launchpad account, join the OpenStack Foundation and click the Sign In link on Gerrit. Please contact the OpenStack Foundation to initiate this process.

Check the FAQ for Contributor License Agreements if you have questions

Abbildung: `https://wiki.openstack.org/wiki/How_To_Contribute#Contributor_License_Agreement`

Lösungen

- Alice unterzeichnet die nötigen Bedingungen

Grund 12: Bob hat einfach keine Zeit

Bob hat einfach keine Zeit

- Alice sendet einen Pull-Request an das Projekt
  $\rightarrow$ keine Reaktion
- Alice sendet später einen Patch per E-Mail an Bob
  $\rightarrow$ nichts passiert
- Bob hat keine Zeit, ist mit anderen Aufgaben beschäftigt oder interessiert sich nicht mehr für die Software

Lösungen

- Alice bittet darum, das Projekt selbst zu übernehmen
- Alice hat auch darauf keine Antwort bekommen oder es ist nicht möglich
  $\rightarrow$ Alice macht einen Fork

Ein paar allgemeine Tipps

Ein paar allgemeine Tipps

- Kritik ist eine Chance, Ansätze zu überdenken
- die kleinste ausreichende Lösung hat die besten Chancen, akzeptiert zu werden
- bei Frust erst einmal Pause machen :)

Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an info@b1-systems.de oder +49 (0)8457 - 931096