Qualitätsanalyse und Teammanagement in Open-Source-Projekten

Andreas Tille

CLT 2012

Chemnitz, 17. März 2012

1. Wie kann man beurteilen, ob ein FLOSS-Projekt zum scheitern verurteilt ist

2. GSoC - Teammetrics

Quelle: Fail - O - Meter

- Folien sind im wesentlichen Übersetzung von *How to tell if a FLOSS project is doomed to FAIL*
- Inspiriert durch Untersuchung von Chromium, doch später verallgemeinert
- Offensichtliche Ausnahmen: Linux Kernel
- Ausnahmen funktionieren, weil sie klein begannen und die Community mit dem Code wachsen konnte
- Große Komplexität erfordert eine große Community
- Der Versuch ein hochkomplexes Projekt zu starten macht es außerordentlich schwer, eine funktionierende Community zu entwickeln

Größe

- Quellcode > 100MB [+5]
- Quellcode sogar komprimiert > 100MB [+5]

Quellcodeverwaltung

- Kein öffentliches Quellcoderepository (svn, git, . . .) [+10]
- Öffentliches Quellcoderepository, aber
  - Kein Webviewer [+5]
  - Keine Dokumentation für neue Nutzer wie es zu nutzen ist [+5]
  - Selbstgeschriebene Quellcodeverwaltung für diesen Code [+30]
  - Diese Quellcodeverwaltung wird nicht wirklich genutzt [+50]

Bauen des Quellcodes

- Keine Dokumentation zum Bauen aus den Quellen [+20]
- Dokumentation existiert, funktioniert aber nicht [+10]
- Quellcode ist konfiguriert:
  - mit einem handgeschriebenen Shellscript [+10]
  - durch Editieren von Textkonfigurationsdateien [+20]
  - durch manuelles Editieren von Headerdateien [+30]
- Quellcode ist nicht konfigurierbar [+50]
- Quellcode baut:
  - nur mit etwas Anderem als GNU Make [+10]
  - nur mit proprietären Werkzeugen [+50]
  - nur mit einem eigens erstellten Werkzeug [+100]
- Es existieren keine Tests für das Resultat [+5]

Java hat andere Eigenschaften hinsichtlich des Bauens:

- Kein Standardbauwerkzeug (Ant, Maven, Gradle) [+10]
- Laufzeitbestandteile mit maschinenabhängigen Teilen [+5]
- Es existieren keine Tests für das Resultat [+20]

Bündelung mit anderen unabhängigen Projekten

- Sourcecode beinhaltet fremde Projekte, von denen es abhängt [+20]
- Fremder Sourcecode muß zuerst gebaut werden, bevor eigener Code gebaut werden kann [+10]
- Der eingebundene Fremdcode wurde sogar modifiziert [+40]

Bibliotheken

- Code baut nur statische Bibliotheken [+20]
- Code baut auch zur dynamischen Bibliothek doch ohne Versionierung [+20]
- Code versucht nicht die vorhandenen Systembibliotheken zu benutzen [+20]

Für Java:

- Code enthält maschinenabhängige Teile [+5]

Installation auf dem Zielsystem

- Installation zu /opt oder /usr/local [+10]
- Es fehlt ein „`make install`“ [+20]
- Code funktioniert nicht außerhalb des Quellcodeverzeichnisses [+30]

Seltsame Codeeigenschaften

- Code benutzt Windows Zeilenvorschübe ("DOS formatiert") [+5]
- Code hängt von speziellen Compilereigenschaften ab [+20]
- Code hängt von speziellen Compilerfehlern ab [+50]
- Code hängt von Microsoft Visual Irgendwas ab [+100]

Kommunikation

- Neue Versionen werden nicht auf Mailingliste angekündigt [+5]
- Projekt hat keine Mailingliste [+10]
- Projekt hat kein Fehlerverfolgungssystem [+20]
- Projekt hat keine Webseite [+50]
- Projekt ist „SourceForge Vaporware“ [+100]

Releases und Versionierung

- Keine typische Versionierung (Major, Minor) [+10]
- Grundsätzlich keine versionierten Releases [+20]
- Überhaupt keine Releases [+50]
- Releases als Attachments in Webforum Posts [+100]
- Releases nur im .zip Format [+5]
- Releases nur im OSX .zip Format [+10]
- Releases nur im .rar Format [+20]
- Releases nur im .arj Format [+50]
- Releases nur in selbsterfundenem Archiv-Format [+100]
- Release entpackt nicht in versioniertes Verzeichnis (z.B. glibc-2.4.2/) [+10]
- Release entpackt nicht in Verzeichnis (z.B. glibc/) [+25]
- Release entpackt in zahlreiche Verzeichnissebenen (z.B. home/johndoe/glibc-svn/tarball/glibc/src/) [+50]

Geschichte

- Abspaltung eines anderen Projektes [+10]
- Hauptentwickler sind nicht im Mutterprojekt involviert [+50]
- Vor der freien Veröffentlichung war der Code proprietär für
  - 1-2 Jahre [+10]
  - 3-5 Jahre [+20]
  - 6-10 Jahre [+30]
  - 10+ Jahre [+50]

Lizensierung

- Code enthält nicht in jeder Datei Lizensbedingungen [+10]
- Code enthält widersprüchliche Lizensbedingungen [+20]
- Code enthält keine Bemerkung zur beabsichtigen Lizens [+30]
- Code enthält keine Kopie des Lizenstextes [+50]
- Code hat keine Lizens [+100]

Dokumentation

- Code hat keinen Changelog [+10]
- Code hat keine Dokumentation [+20]
- Webseite hat keine Anwendungsbeispiele [+20]
- Webseite hat keine Dokumentation [+30]

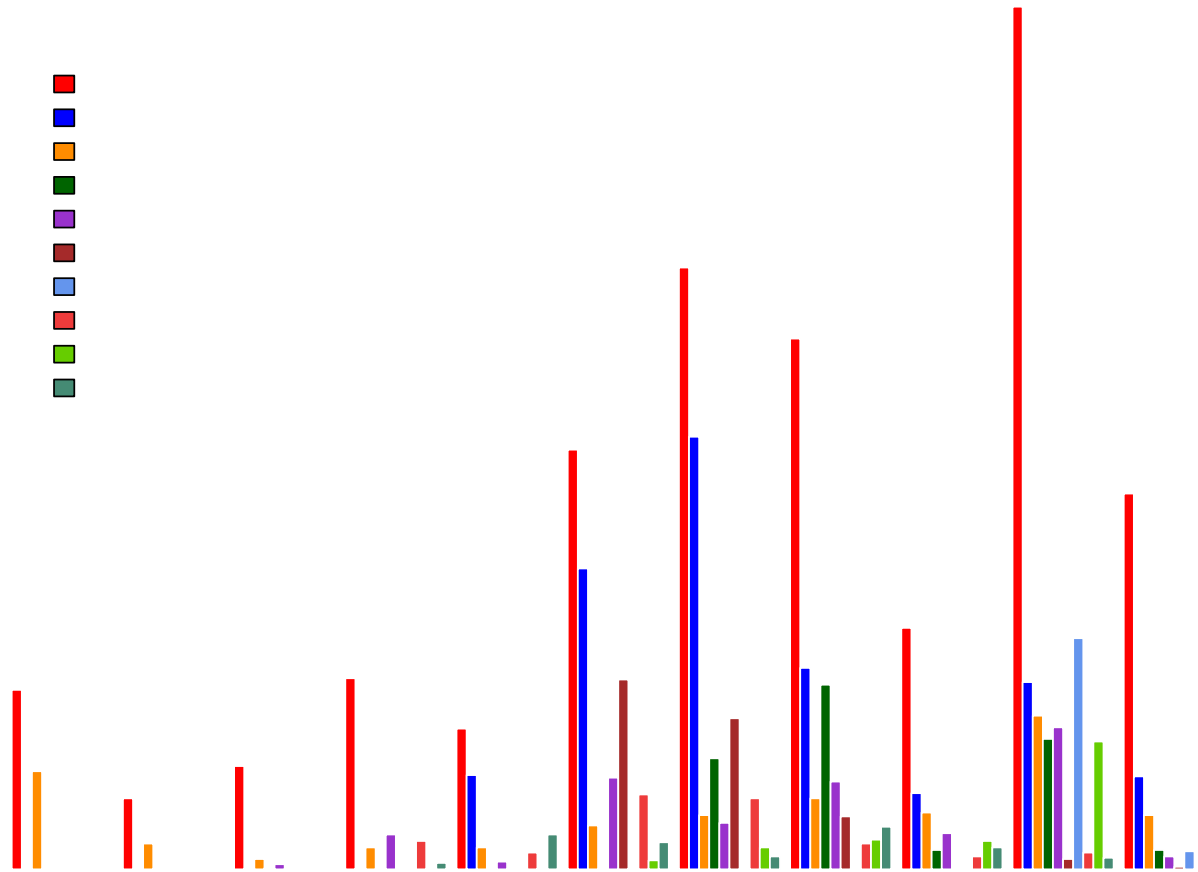
Maß für den wahrscheinlichen Mißerfolg

0	Perfekt! Alle Zeichen stehen auf Erfolg!
5-25	Scheint OK, auch wenn manche Aspekte verbesserungswürdig sind.
30-60	Babies beginnen zu weinen wenn der Code heruntergeladen wird.
65-90	Kätzchen sterben wenn der Code heruntergeladen wird.
95-130	Achtung, Achtung, das Schiff beginnt zu sinken.
135+	Der Code ist so kaputt, er könnte seine eigene Reality TV show haben

Teil II: GSoC - Teammetrics

- Erster Teil Code-orientiert, zweiter Teil Team-orientiert
- Wer ist im Team?
- Wer verließ das Team?
- Hat ein Team genug Mitglieder?
- Wächst oder schrumpft ein Team?
- Ursachenanalyse

Einfacher Ansatz: Aktivität auf der Mailingliste

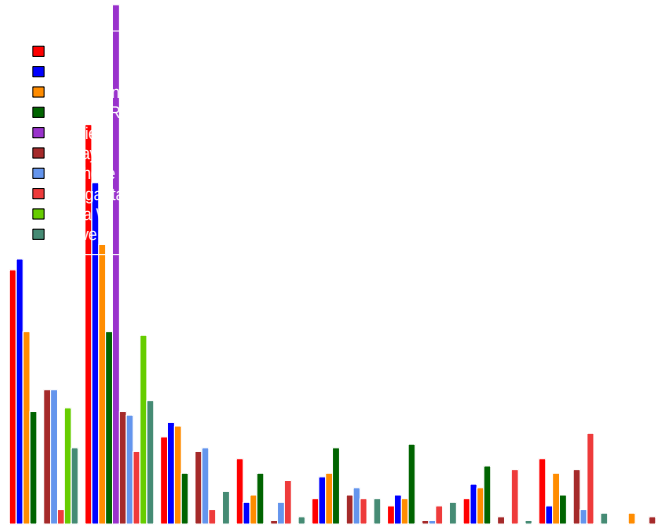


GSoC: Erweiterte Beobachtung

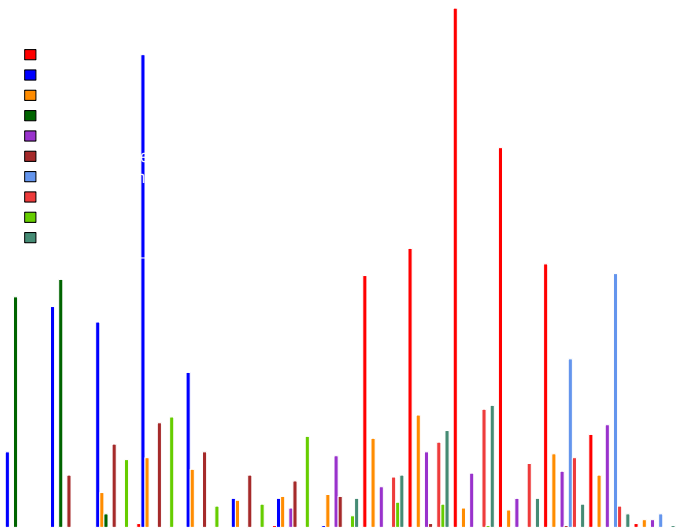
- Wer trägt mehr als nur durch „Schwatzen“ bei
  - Code Beiträge
  - Statistik über abgeladene Pakete
  - Statistik über gelöste Fehler
- Größere Flexibilität
  - Teams in Konfigurationsdateien
  - Bessere Handhabung von SPAM + Robots
  - Bessere Namenszuordnung
  - Prinzipiell nicht auf Debian begrenzt

Zweck des Teams erfüllt

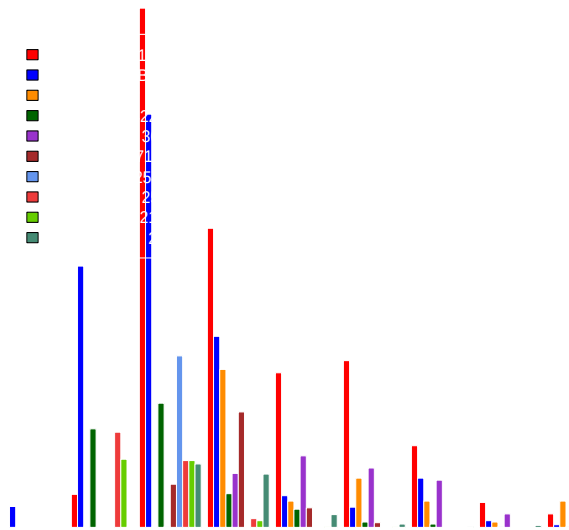
Debian Women →



↓ Portierung Arm

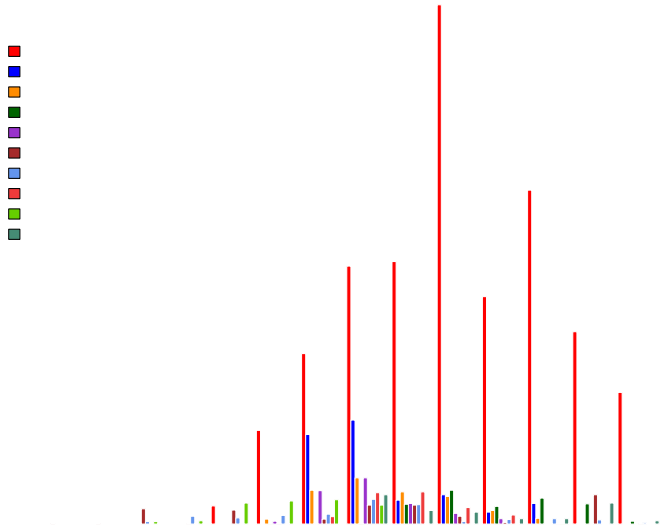


Portierung Amd64

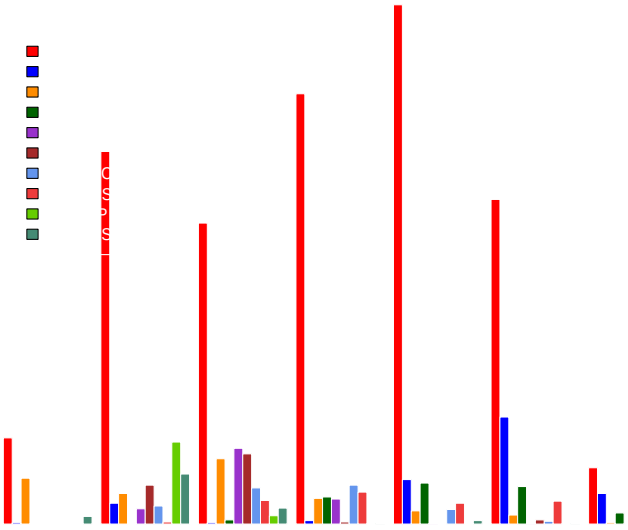


Dominiert durch eine Person

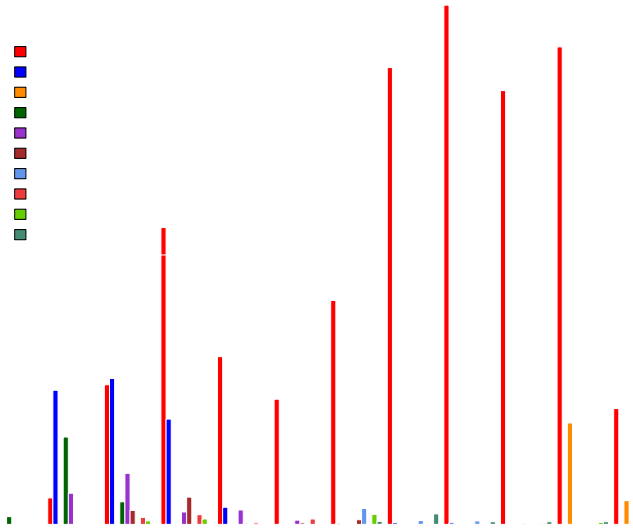
Debian i18n



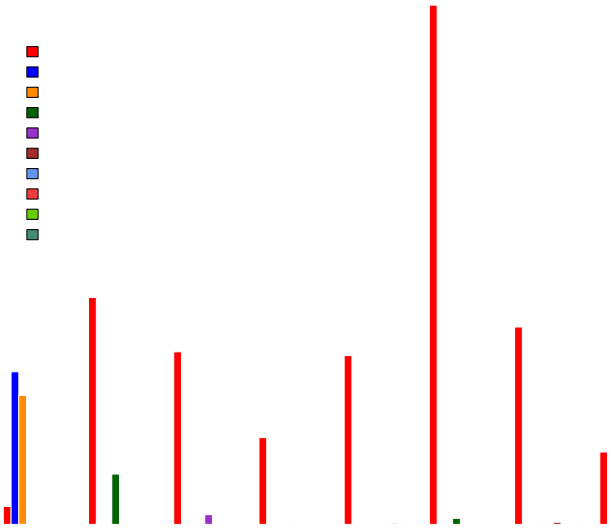
Debian Live



Openoffice Mailingliste

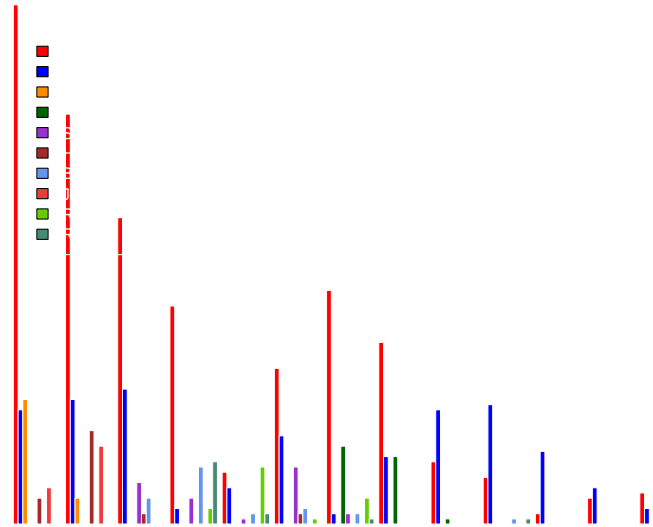


Openoffice Commits

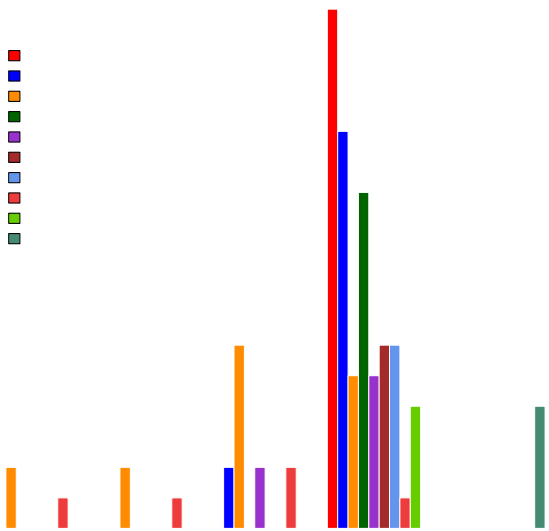


Verwaiste und nicht funktionierende Projekte

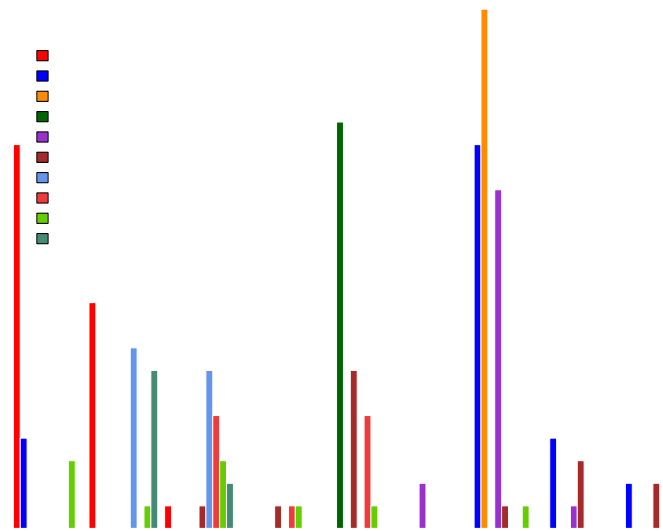
Debian Junior →



↓ Debian Enterprise

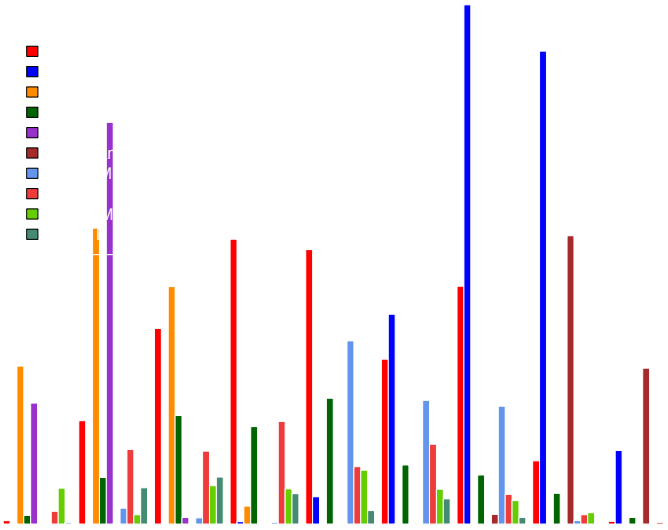


Debian Lex

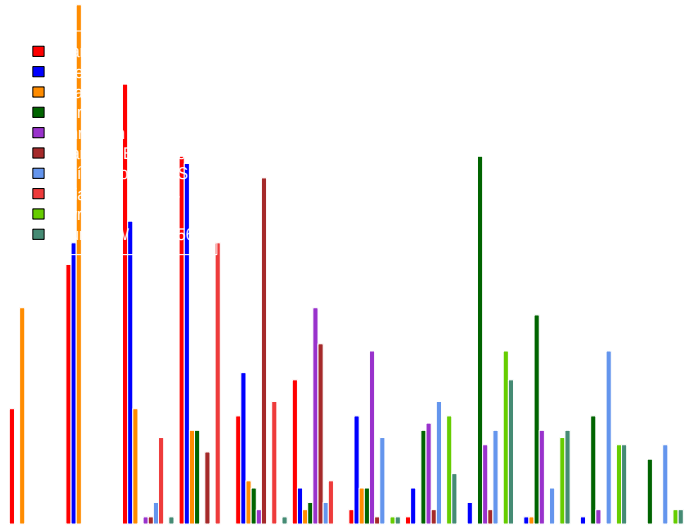


Wichtige Projekte - Übernahme der Projektleitung

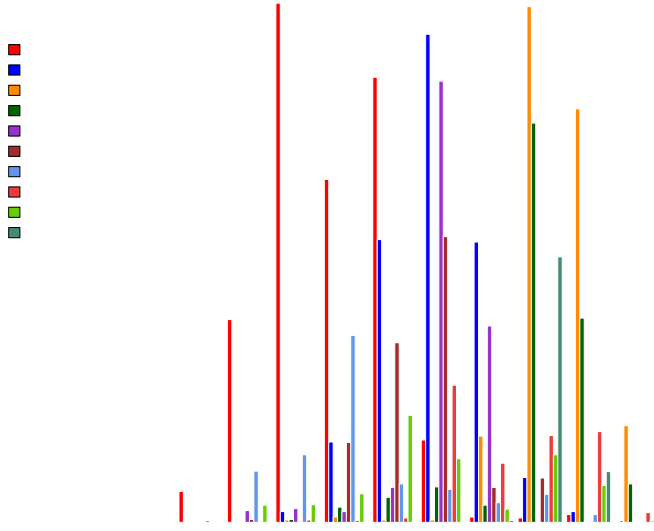
Debian Kernel



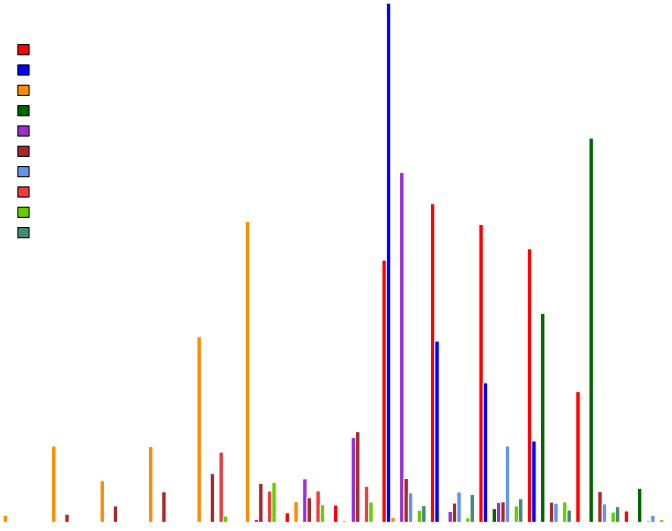
Debian New Maintainer



Debian Release

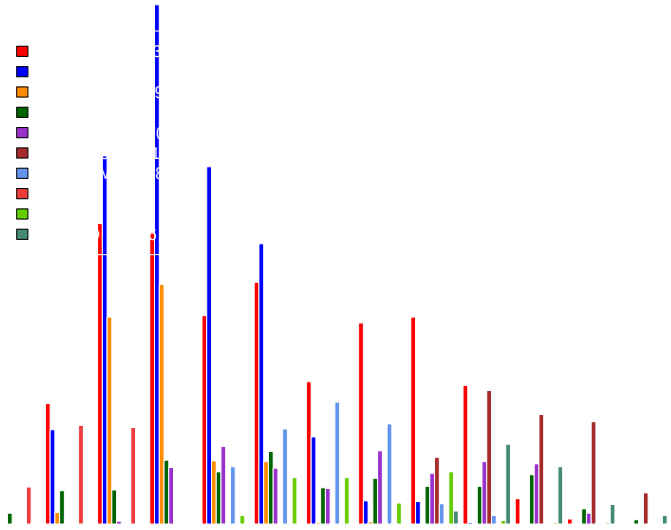


Debian X

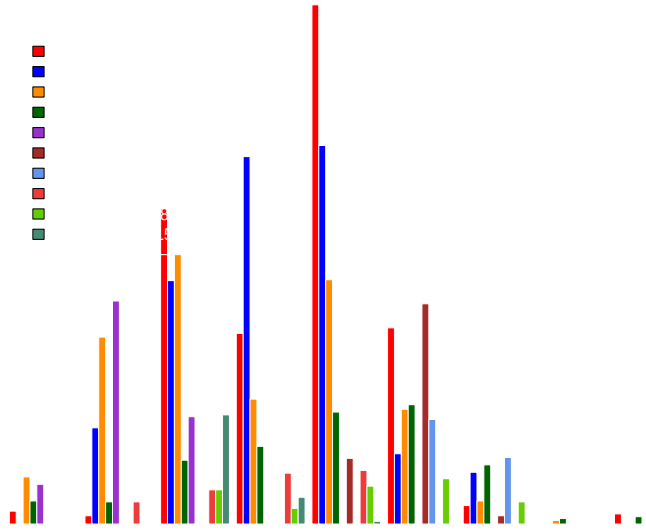


Projektleiter übernimmt andere Aufgabe (Ocaml Team)

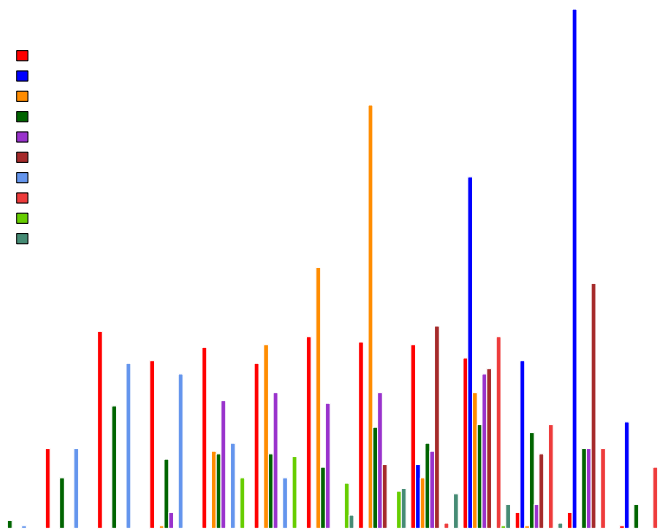
Mailingliste



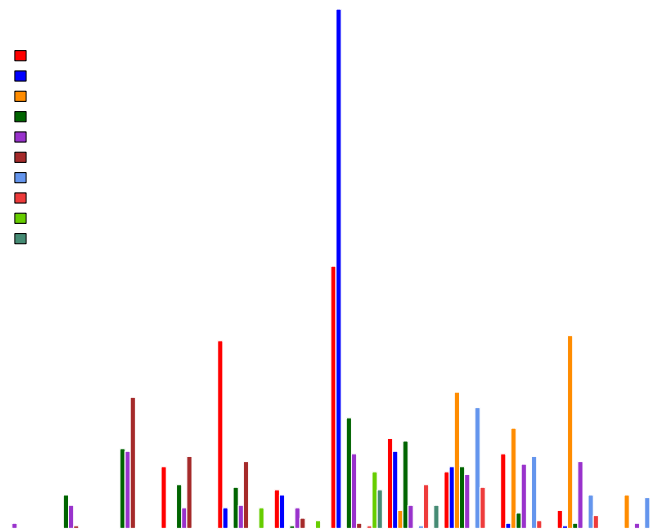
Commits



Paketuploads

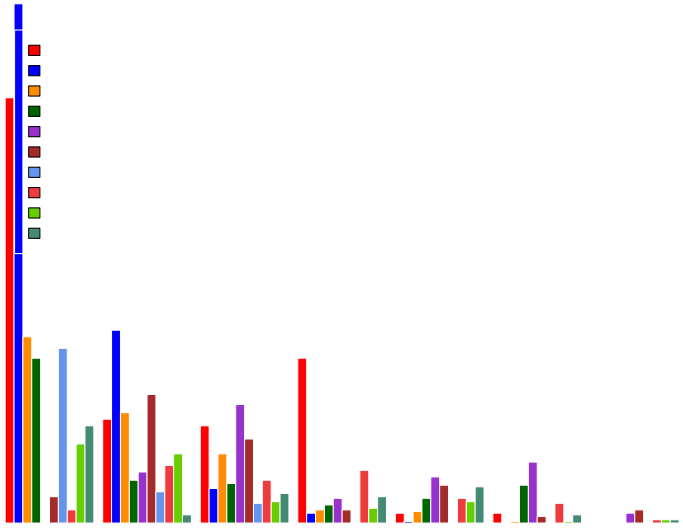


Behobene Fehler

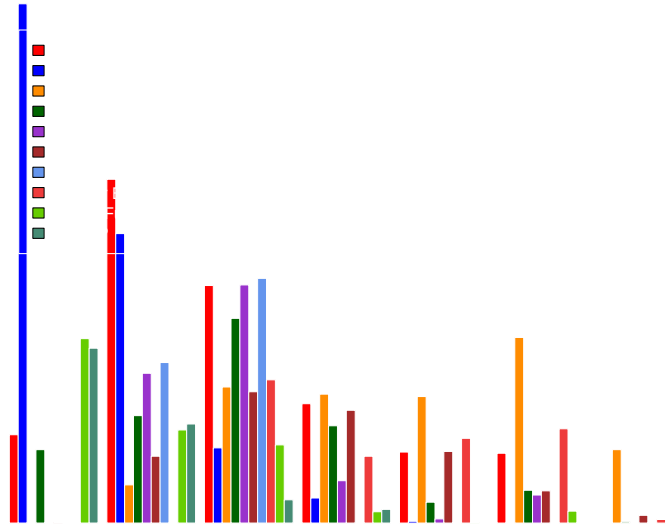


Diskussion unterrepräsentiert (Pkg-Games)

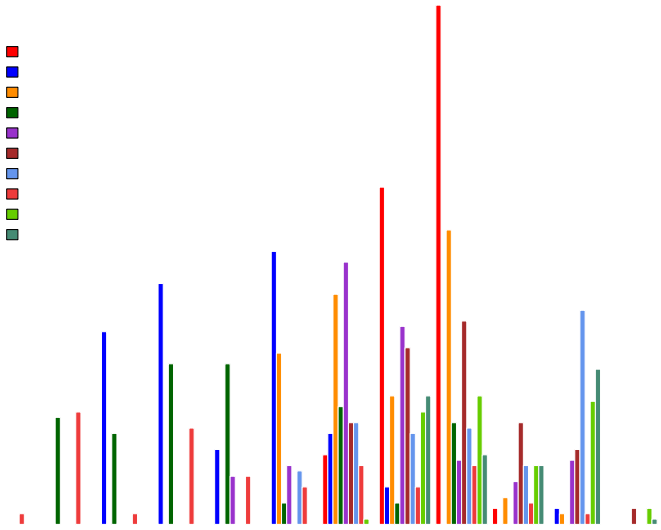
Mailingliste



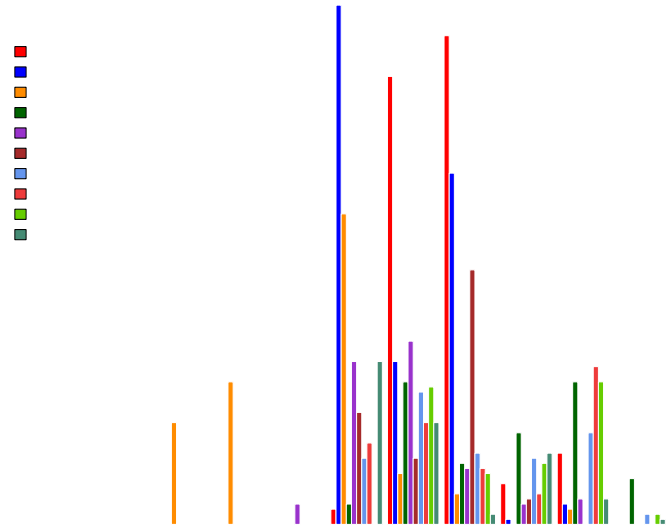
Commits



Paketuploads

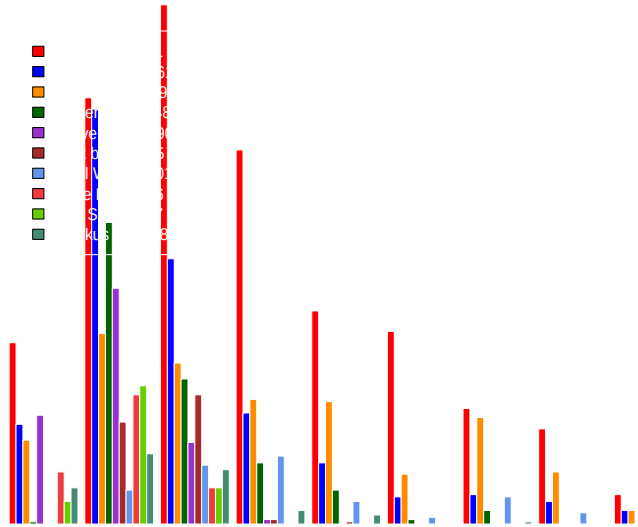


Behobene Fehler

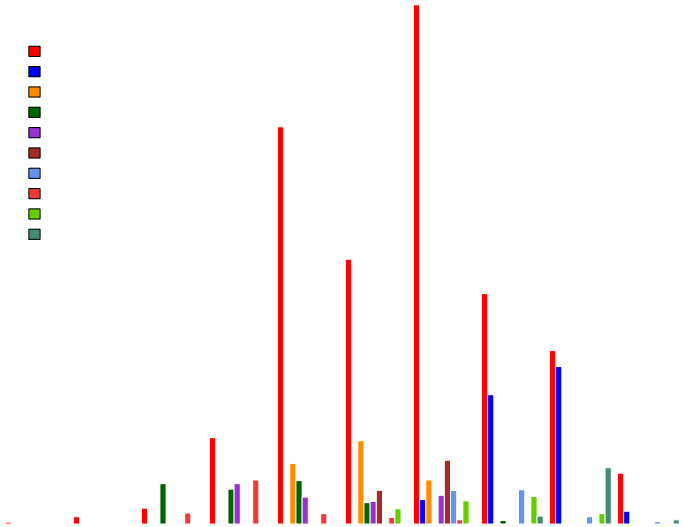


Mißverhältnis Diskussion - Code (Debian GIS)

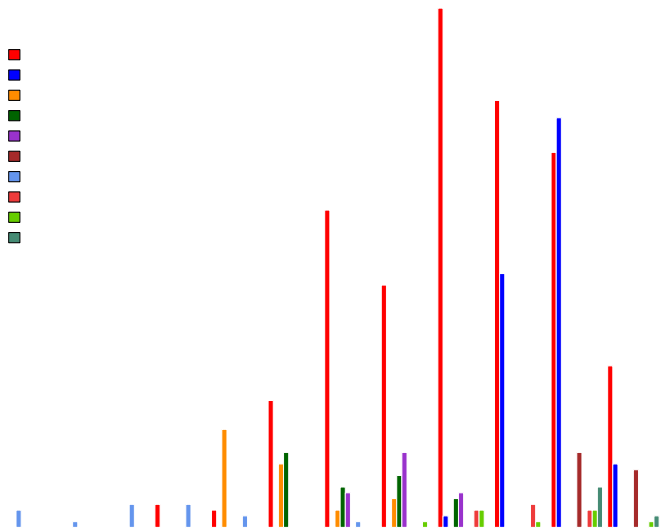
Mailingliste



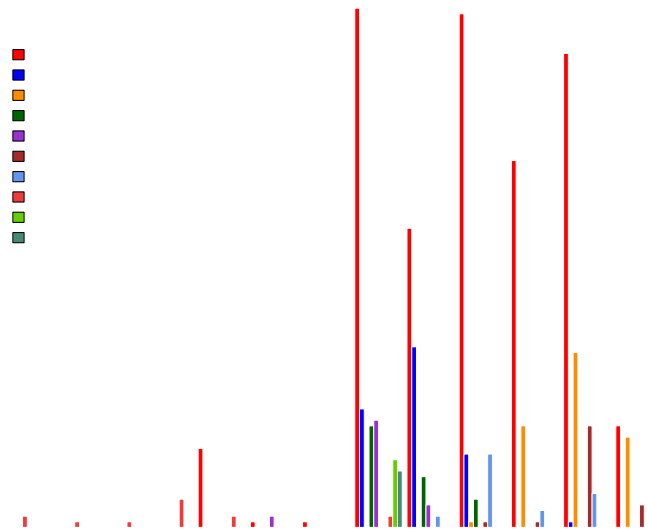
Commits



Paketuploads

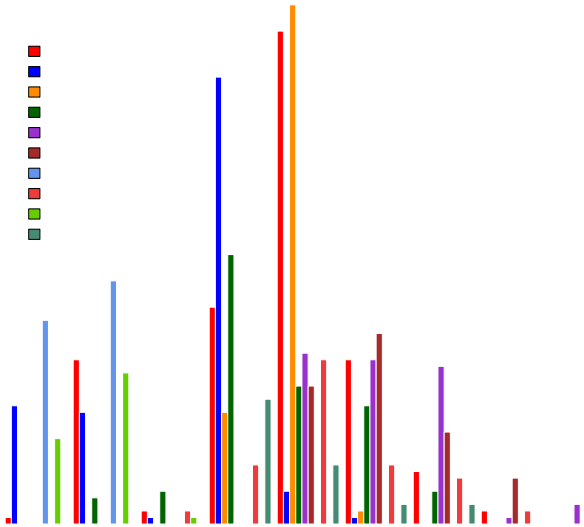


Behobene Fehler

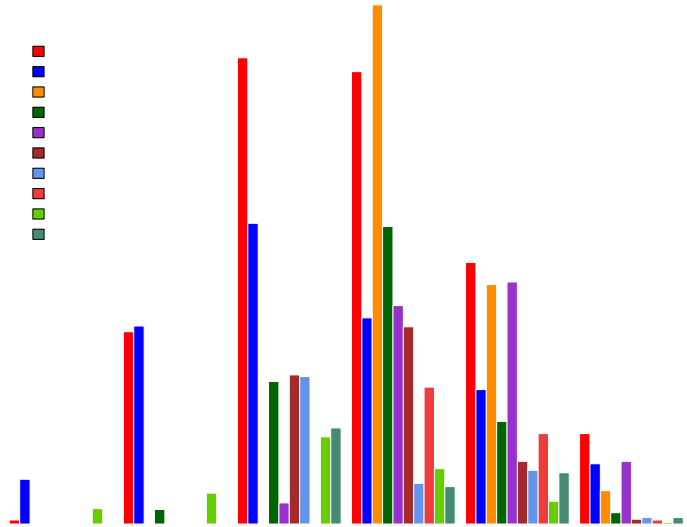


Gespaltenes Team (Debian Multimedia)

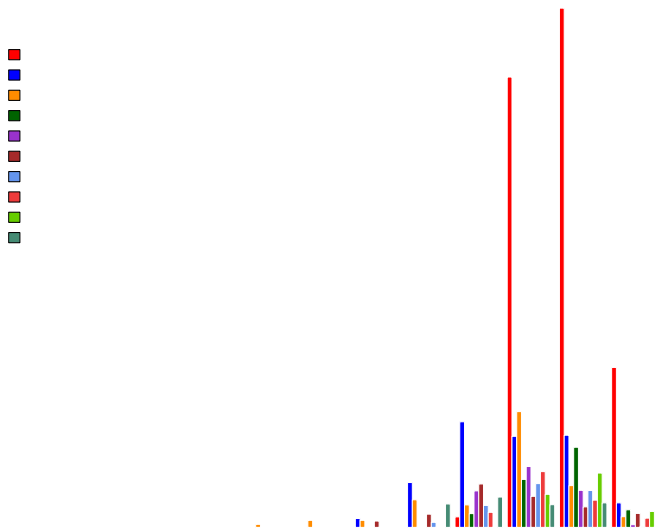
Mailingliste 1



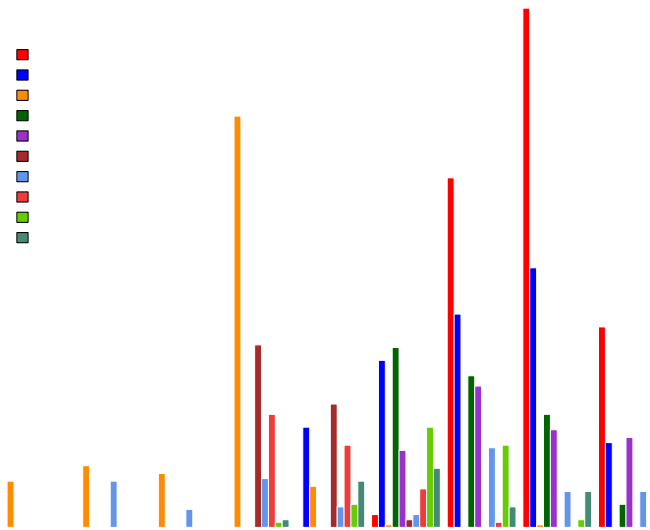
Mailingliste 2



Commits

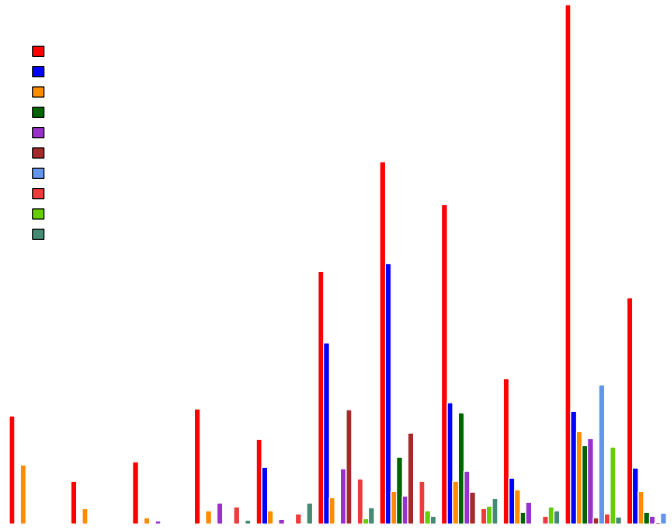


Behobene Fehler

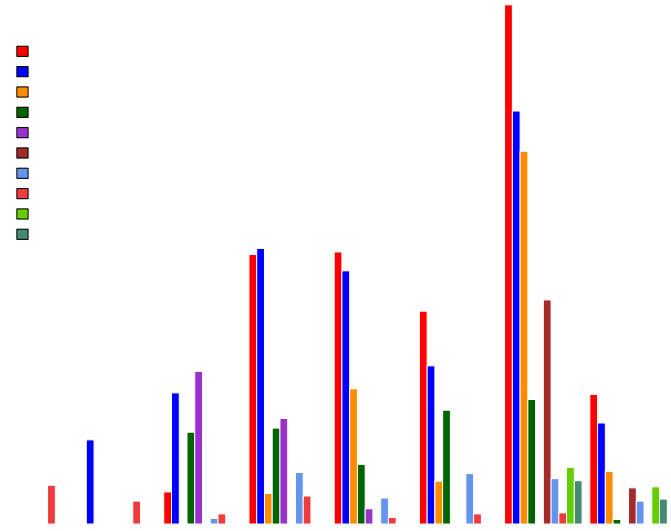


„Gesundes“ Team (Debian Med)

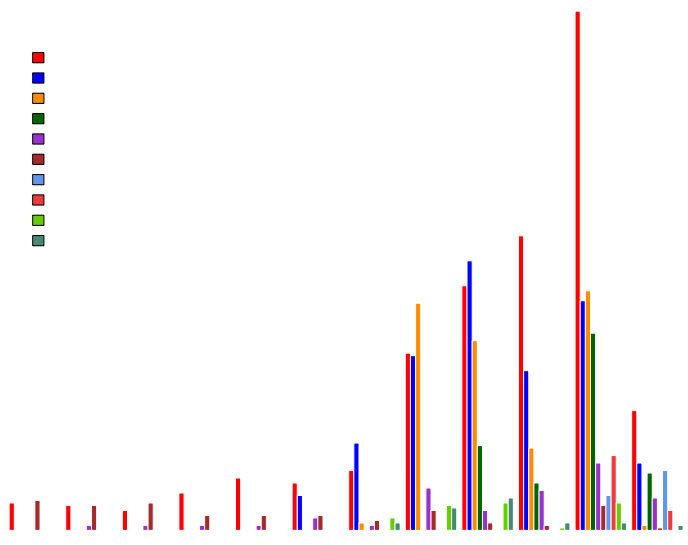
User Mailingliste



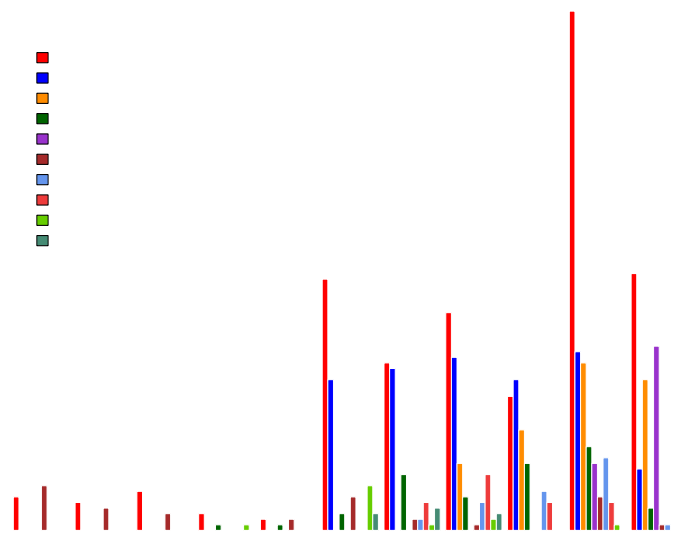
Commits



Paketuploads



Behobene Fehler



Quellen und weiterführende Literatur

- *How to tell if a FLOSS project is doomed to FAIL*
- *GSoC Metrics working group*
- *Debian Teams Activity Metrics*
- *Open Advise. What We Wish We Had Known When We Started*

This talk is available at
http://people.debian.org/~tille/talks/
Andreas Tille <tille@debian.org>