OpenSource-Software-HowTo

Wie wird OpenSource-Software eigentlich entwickelt?

Linux-Cafe 2012-02-13

Referent:

Bernd Strößenreuther

<linux-cafe@stroessenreuther.net>

Was macht OpenSource aus?

- Die Freiheit, das Programm für jeden Zweck auszuführen.
- Die Freiheit, die Funktionsweise eines Programms zu untersuchen, und es anzupassen.
- Die Freiheit, Kopien weiterzugeben.
- Die Freiheit, ein Programm zu verbessern, und die Verbesserungen an die Öffentlichkeit weiterzugeben.

Quelle: http://fsfe.org/about/basics/freesoftware.de.html

Merkmale von OpenSource

- Quelltext verfügbar
- Meist: Versionskontrollsystem frei einsehbar
- Oft: Binaries verfügbar
- Teilweise: Pakete für Distributionen
- Oft: Direkt in den Repositories der Distributoren enthalten

Von der Idee zum Projekt

- Idee für ein Programm, das auch für andere nützlich sein könnte
- Gibt es schon etwas Ähnliches oder eine Vorstufe davon als OpenSource?
  - Falls nein: Ganz neues Projekt starten
  - Falls ja: Verbesserung oder Erweiterung beisteuern (contributen)
  - Falls nicht möglich oder gewünscht: Fork

Projekt-Hilfsmittel

- WebSite
- Download-Bereich
- Versionskontrollsystem
  - Meist: World-readable, Commit: wenige
- Projekt-Mailinglisten
  - Meist: Developer, User, Announce
- Meist öffentliches Bugtracking-System

Projekt-Organisation I

- Kleineres Projekt, z. B.
  - Ein Maintainer
  - Wenige Leute mit Commit-Rechten
  - Patches werden auf Dev-Mailingliste gepostet und dort diskutiert
  - Letzlich: Commit, Ablehnung oder Anfordern von Anpassungen

Projekt-Organisation II

- Große Projekte
  - Mehrstufige Verfahren
  - z. B. Kernel: Linus vertraut Subsystem-Maintainern sehr weitgehend
  - Teilweise formalisierte Verfahren zur Qualitätssicherung

How to contribute?

- Coding-Style des Projektes beachten
- Möglichst gleiches Versionskontrollsystem verwenden
- Rausfinden wie Patches einzureichen sind
- Meist: Patches im richtigen Format, in der richtigen Granularität an die richtige Mailingliste schicken

How to maintain?

- Maintainer zu sein bedeutet über lange Zeit Verantwortung zu tragen
- Motivieren, koordinieren, Konflikte lösen
- Technik ist oft nicht das Problem
- Projekt-Hilfsmittel auch nicht dank Sourceforge, GitHub, …
- Regeln dürfen mit der Zeit wachsen sollten aber immer klar kommuniziert sein

Vielen Dank...

... für die Aufmerksamkeit

Noch Fragen?