

Getting Started with Open Source Software Contributions

Erfahrungsbericht & Tipps

Christian Heusel

Über mich

- **Name:** Christian Heusel
- **Beruf:** Informatikstudent an der Uni in Heidelberg
- **Interessen:** Software Engineering, HPC & DevOps
 - Arbeite gerne mit Linux & Open Source
 - Betreue Pakete im Arch Linux User Repository
 - Arch Linux Package Maintainer, Bug Wrangler, DevOps-Team & Tester
- **Hobbys:** Schlagzeug & Engagement in CVJM und Jugendwerk
- **Kontakt:** christian@heusel.eu bzw. gromit@archlinux.org



Um was geht es in diesem Vortrag?

- Für wen ist dieser Vortrag gedacht?
 - a. Motivierte Anfänger, die nicht wissen wo anfangen
 - b. FOSS-Veteranen, die nach Inspiration suchen
 - c. alles dazwischen
- Was schauen wir uns an?
 - a. Grundlegendes
 - b. Möglichkeiten ohne Coding mitzumachen
 - c. Möglichkeiten mit n bisschen Coding mitzumachen
 - d. Exemplarisches Vorgehen / Erfahrungsbericht
 - e. Fragen, Anmerkungen etc.
- Format: Gerne Interaktiv :)

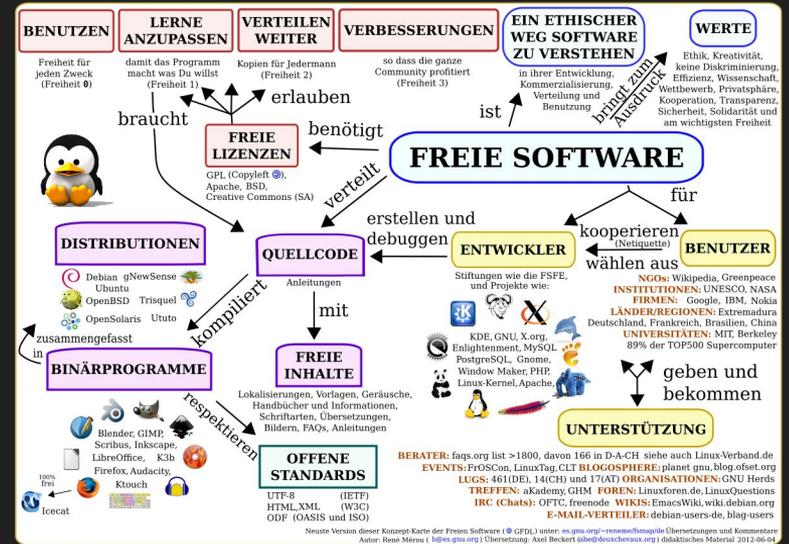
Publikum

- Erfahrung
 - a. Ich bin noch Anfänger
 - b. Ich kann schon ein bisschen was
 - c. Ich bin FOSS Veteran
- Betriebssystem
 - a. Linux
 - b. Windows
 - c. MacOS
 - d. anderes?
- Informatiker oder nicht?

Was ist Open Source / FOSS?

“Free software” means software that respects users' freedom and community. Roughly, it means that **the users have the freedom to run, copy, distribute, study, change and improve the software**. Thus, “free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer.”

– What is Free Software? - GNU Project



Such dir (d)eine Community!

- Worauf hast du Bock?
- Was nutzt du sowieso schon?
- Wo fühle ich mich wohl / mit wem arbeite ich gerne zusammen?
- Welcher Bug / Funktionalität nervt mich? Was würde ich gerne verbessern?
- Was sagen andere was Spaß macht?
- Ziel sollte nie eine Rolle oder Titel sein!
- Viele Communities haben Tipps zum Mitmachen:
 - [Get Involved :: Fedora Docs](#)
 - [Getting involved - ArchWiki](#) / <https://whatcanidofor.archlinux.org/>
 - [Get Involved - KDE Community Wiki](#)
 - <https://kernelnewbies.org>
 - [GIMP - Get Involved](#)
 - [...]

No Code Contributions: Bug Reporting / Bug Wrangling

- Bug Reporting:
 - Bug → Bug Report
 - Bugs forwarden
 - Qualitativ hochwertige Bugreports schreiben
- Bug Wrangling:
 - Triage von Bugs
 - Fehlende Informationen hinzufügen
 - Reproducer testen
 - Logs interpretieren
 - Reporter nerven/unterstützen falls nicht alle Infos da sind
 - andere Bugtracker & Foren durchwühlen

The 5 Stages of Debugging

At some point in each of our lives, we must face errors in our code. Debugging is a natural healing process to help us through these times. It is important to recognize these common stages and realize that debugging will eventually come to an end.



Denial

This stage is often characterized by such phrases as "What? That's impossible," or "I know this is right." A strong sign of denial is recompiling without changing any code, "just in case."



Bargaining/Self-Blame

Several programming errors are uncovered and the programmer feels stupid and guilty for having made them. Bargaining is common: "If I fix this, will you please compile?" Also, "I only have 14 errors to go!"



Anger

Cryptic error messages send the programmer into a rage. This stage is accompanied by an hours-long and profanity-filled diatribe about the limitations of the language directed at whomever will listen.



Depression

Following the outburst, the programmer becomes aware that hours have gone by unproductively and there is still no solution in sight. The programmer becomes listless. Posture often deteriorates.



Acceptance

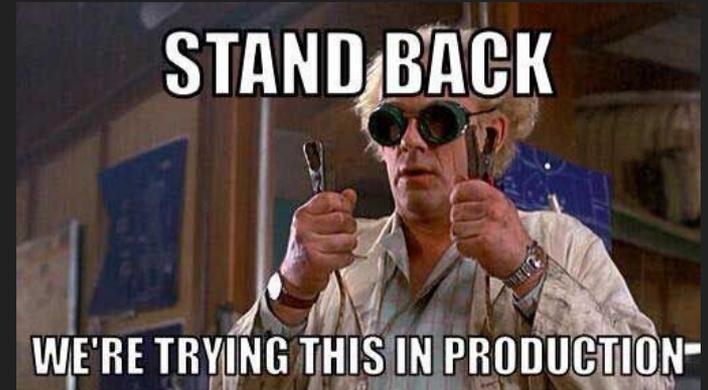
The programmer finally accepts the situation, declares the bug a "feature", and goes to play some Quake.

No Code Contributions: Donations

- Viele Open Source Projekte brauchen Geld um z.B. Infrastrukturkosten zu decken oder um mehr Zeit in das Projekt investieren zu können
- Geld löst viele Probleme bei FOSS Projekten nicht, keines zu haben schafft aber andere!
- Bitte unterstützt auch gerade die Projekte, die ihr kommerziell in der Firma nutzt! (Business-Case)

No Code Contributions: Testing

- Release candidate testing
- Feedback auf Merge Requests, einfach mal ausprobieren
- Testing Branches z.B. von der Distro ([Arch Testing Team - ArchWiki](#))
- Testing für spezielle Hardware



No Code Contributions: Community Work

- **Transparenz & Kommunikation** <https://monthly-reports.archlinux.page/2024/01/>
- **User Support:**
 - Foren
 - IRC, Matrix, Discord, Telegram
 - Github Discussions
 - Mailinglisten
- **Mentoring**
- **PR / Social Media**
- **Moderation**

No Code Contributions: Design

- Artworks
- Icons
- Fonts
- Accessibility
- UX Improvements
- Flyer & Corporate Designs



No Code Contributions: Region specific stuff

- Übersetzung
 - Dokumentation
 - Programme
 - Community
- Font configuration etc.
- Community für die eigene Sprache
- Proxys etc. für Spezielle Netzwerke

No Code Contributions: Dokumentation

- User Guides
- Wikis
- Blogposts
- Videos
- Schreibskills sind oft Mangelware!

(“No Code” Contributions: Infrastructure)

- Administration zahlreicher Services
- Migrationen bei Software- oder Servicearchitekturwechsel
- On- / Offboarding von Leuten beim Projekt
- Incident Response
- “Enabler” für andere Projekte sein

Beispiel: <https://gitlab.archlinux.org/archlinux/infrastructure>



Open Source Code: Erste Schritte für eigene Projekte

1. Erstelle ein Repo und mach es verfügbar
 - Github
 - Gitlab
 - Codeberg
 - Selfhosted
2. Erstelle eine Beschreibung, die sagt was das Projekt ist und welches Problem es löst
3. Code importieren
4. Projekt ordentlich lizensieren: <https://choosealicense.com/>

“As the old saying goes: ‘perfect is the enemy of the good’. Do not wait. Do not clean up your code first. Do not fall into the trap of ‘I just need to do this before I Open Source my project.’”

[https://un curl.dev/start/do-it](https://un	curl.dev/start/do-it)

Code Contributions: Erste Schritte für existierende Projekte

- Einfach anfangen und *einfach* anfangen!
 - Eigene Bugs als Motivation nehmen
 - Labels wie “easyfirst” “help wanted” etc. suchen
 - Beispiel: <https://github.com/zsh-users/zsh/pull/119>
- Mühe bzgl. der Community Regeln geben!
- Bereit sein Fehler zu machen
- Review:
 - Nicht erschrecken & nicht persönlich nehmen!
 - Als Möglichkeit für eigene Verbesserung sehen
 - Auf Review (zeitnah) eingehen
- Dran bleiben und weiter machen

Persönliches Vorgehen

- Mäuschen spielen → Mailingliste, Issues oder Bugtracker subscriben
- Nutzen & Testen
- Tools & Arbeitsweise kennen lernen
- Fehler von anderen Abschauen
- Entrypoints suchen
- Abfahrt!

Fragen?

Getting started with Open Source Software Contributions – Christian Heusel

Ergänzungen?

Getting started with Open Source Software Contributions – Christian Heusel