

# 20 Jahre als überzeugter FOSS-Entwickler

Harald Welte

Danke

# Ungewohnt

Ungewohnt: Keine technische Präsentation

# Ungewohnt: Deutsch

Frage

# Alleinunterhalter

# Überzeugungstäter



# Kindheit

# Stereoanlage

# Schreibmaschine

# Stadtbibliothek

# C64 bei Freunden

# Eigener 80286 / DR-DOS mit 10 Jahren

RLL-Festplatte Low-Level formatiert am 1. Tag :)

Mehr Spass am Save Game Hacking als am spielen



# Von QBasic zu QuickBasic zu Turbo Pascal

# x86 Assembler mit Stift + Block im Sommerurlaub

# 300-Baud Akkustikkoppler

# Mailboxen/BBSs

## Communities (Z-Netz, FIDO, ...)

# Shareware/Freeware

# KA9Q NOS

# Kommunikationstechnik (Modems, ISDN, Ethernet)



# Kommunikationsnetz Franken e.V.

1994: Linux

Hilfe zur Installation: SCSI bootdisk geht nicht mit IDE

# LUG-N / ALIGN

Es gibt Communities!

# Wissen Teilen: Kurse an VHS und im KNF

# Freie Software

Man bekommt Source Code



Man kann Software anpassen

# Lernen, Lernen, Lernen

# Ethische Aspekte Freier Software

# Freie Software als Hobby

Erste patches und contributions (smail, pppd, pptpd)

# Quersubventionierung durch bezahlten nicht-FOSS Job

Italienurlaub 1998: ip\_conntrack\_irc.c

Ziel: Hauptberufliche FOSS-Entwicklung



netfilter core team

## 2000: Conectiva in Brasilien

# Bester denkbarer Arbeitsvertrag

# dot-com Blase platzt

# Freiberufliche Linux Kernel Entwicklung in DE

# netfilter/iptables für Firewall-Appliance-Hersteller

# Treiberentwicklung für Hardwarehersteller

2002: Umzug nach Berlin; CCC; mehr IT-Security



## 2004: OpenEZX: Hackbare Linux-Telefone

2005: [gpl-violations.org](http://gpl-violations.org)

# Gerichtsverfahren zur Durchsetzung der GPL

# Copyleft-Lizenzen sind keine Einbahnstrasse

Wenn Regeln nie durchgesetzt werden, hält sich keiner dran

2006: Open Source RFID; erste Open Source Hardware Projekte

2007: Chief Architect System Level (HW+SW) bei  
Openmoko (Taiwan)

2008: Open Source Liaison bei VIA Technologies (Taiwan)



2008: bs11\_abis / bsc\_hack / OpenBSC

# 2010: Training der Samsung Linux-Kernel-Entwickler in Korea

# Quersubventionierung: FOSS Mobilfunk durch Linux-Kernel-Arbeit

# Ziel: Hauptberufliche FOSS-Entwicklung an Mobilfunkprotokollen

## 2010: OsmocomBB

2011: Gründung sysmocom GmbH in Berlin

# Vollzeit FOSS Mobilfunkinfrastruktur + Hardwareentwicklung

Geschäftsführer, aber trotzdem überwiegend  
Entwicklungstätigkeit



Herausforderung: Firma mit ERP, Buchhaltung, etc. und  
nur FOSS

die einzigen Windows-Systeme, mit denen ich seit 20  
Jahren zu tun habe?

# Oszilloskop und Vektor-Network-Analyzer

2018: 10 Jahre Osmocom: Dutzende Projekte, 140  
repositories auf [git.osmocom.org](https://git.osmocom.org)

# Freie Software vor 25 .. 20 Jahren

# Absolute Nische

Anwender: Privat, Forschung, KMU

Wir benutzten Kernel 2.3.99 produktiv in einer  
Grossbank...



...aber wir mussten es "UNIX" nennen, damit sich keiner erschrickt ;)

# RCS und CVS als Versionsverwaltung (Dateien, nicht Trees/Projekte)

Keine "continuous integration"

Grosse Projekte glibc oder X11 bauten (gefühlte) Tage

# Freie Software heute

# Grosskonzerne

# Silicon Valley

# Contributor License Agreements



Wer entwickelt FOSS und warum?

# Einzelpersonen aus Enthusiasmus

Wissenschaftler? gute contributions selten.  
Paper-orientiert, proof-of-concept reicht

Industrie? leider oft einseitig(forks, keine contributions upstream, ...)

# Negative Trends

# Licence Proliferation

# pseudo-FOSS-Lizenzen

# Openwashing



# anti-copyleft-campaigning aus Silicon Valley

9 Monate Vertragsverhandlungen keine Seltenheit

# sogenanntes Cloud-Computing

# Fokus der Linux-Entwicklung in letzten Jahren

# Konferenzen

Früher: kleiner, sehr technisch, nicht kommerziell; an/in  
Universitäten

Heute: Kommerziell, Linux Foundation Monokultur, in  
Hotels/Konferenzzentren

# Recht



Früher: Entwickler behalten Urheberrechte; Entspannte  
Verträge

# Heute: Anti-Copyleft-Attitude; Monatelange Vertragsverhandlungen

Level

Früher: Mehr C, systemnah. Gab ja noch kein Java,  
Python, PHP, Ruby, ...

# Komplexität

Komplexität nimmt ständig zu. Niemand versteht mehr als seinen kleinen Teil.

Natürlicher Reflex "Angst vor Komplexität" ausgestorben

# Fragmentierung



Communities Fragmentiert nach Programmiersprache,  
Framework, Industriesektor

# Lessons learnt

Keine "single points of failure"

keine proprietären Entwicklungstools/services

keine CLA / copyright assignments

Vorsicht vor open source ohne kollaborative  
Entwicklergemeinde

# Letzte Worte

Vergesst nicht, wo Freie Software herkommt



Industrie ist immer mit Vorsicht zu geniessen

Freiheit braucht Schutz, deshalb ist copyleft wichtig

Nachhaltige Softwareentwicklung funktioniert nur durch  
contributions von allen Seiten

EOF (End of File).

No packets were harmed during the making of this presentation.

Der Autor war stets bemüht.