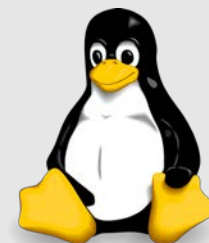


FLOSS vs. Closed Source Software

SEMINAR: OPEN SOURCE SOFTWAREENTWICKLUNG

AKAN YILMAZ



- ▶ Was unterscheidet die beiden Modelle?
- ▶ Beweggründe der FLOSS Gemeinschaft
- ▶ Was spricht für FLOSS?
- ▶ Was spricht für Closed Source Software?
- ▶ Hat FLOSS weniger Bugs als proprietäre Software?
- ▶ Ist die Codequalität besser bei Open Source?
- ▶ Ist FLOSS sicherer als proprietärer Code?
- ▶ Ist die Kontinuität von FLOSS unsicherer als die von proprietärem Code?

Unterschiede

- ▶ Werden deutlich, wenn man sich die Beweggründe der FLOSS Gemeinschaft anschaut

Beweggründe von FLOSS

- ▶ Die guten alten Zeiten im AI Lab des MIT, 1971
- ▶ Plötzlich proprietäre Software bei neuen Rechnern
- ▶ Richard konnte das nicht ertragen und fing an zu handeln
- ▶ 1984 kündigte er am MIT und startete das GNU Projekt



Richard Stallman

Beweggründe von FLOSS

6

«Es mag fair sein, Atemluft pro verbrauchten Liter zu berechnen, aber Tag – Nacht Gasmasken mit Messeinrichtungen zu tragen ist unerträglich, selbst wenn der Luftpreis für jeden erschwinglich ist. Und allgegenwärtige Fernsehkameras, die überwachen, ob jemand die Maske abnimmt, sind ungeheuerlich. Es ist besser, die Luftanlage mit einer Kopfsteuer zu finanzieren und die Masken wegzuwerfen.

Kopieren aller oder Teile eines Programms ist für einen Programmierer so natürlich wie das Atmen, und so produktiv. Es sollte genauso frei sein.»

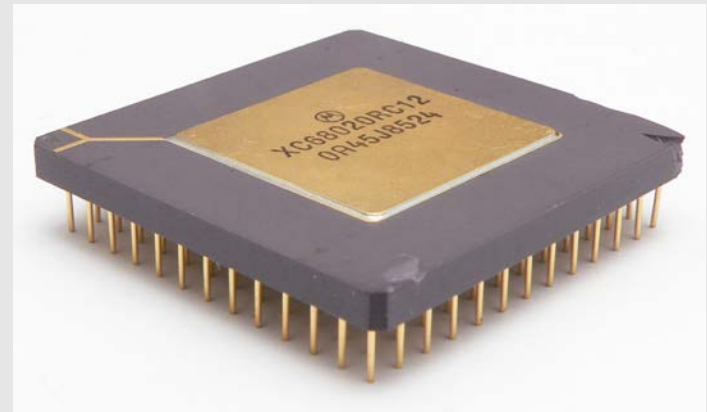


Richard Stallman

Beweggründe von FLOSS

7

- ▶ Der 68020 lief mit proprietärer Software
- ▶ Vertraulichkeitsvereinbarung unterzeichnen
- ▶ Zusammenarbeit verboten
- ▶ Hersteller hat Macht über die Software und ihre Benutzer
- ▶ Druckersoftware im AI Lab
- ▶ Was musste das GNU Projekt tun, damit ein Programmierer seine Freunde nicht verrät?



Motorola 68020
32-bit microprocessor, 1984

Beweggründe von FLOSS

- ▶ Es braucht Freiheit!
- ▶ Freiheit beim Ausführen!
- ▶ Freiheit beim Teilen!
- ▶ Freiheit beim Anpassen!
 - ▶ Quellcode offen!
- ▶ Und die GPL, damit GNU auch GNU bleibt



Beweggründe von FLOSS

- ▶ Linus Torvalds benutzt 1992 das Basarmodell
- ▶ Der Quellcode ist immer offen einsehbar
- ▶ Früh freigeben und oft freigeben
- ▶ Linux hat grossen Erfolg



Linus Torvalds

Beweggründe von FLOSS

10

- ▶ Eric S. Raymond versucht Linus nachzumachen
- ▶ Er versucht das Basarmodell bei seinem Projekt «fetchmail»
- ▶ Und hat Erfolg
- ▶ Er meint aber auch, dass man im Basarmodell eine nette und gute Führung braucht



Eric S. Raymond

Unterschiede

FLOSS/OSS

- ▶ Quelloffen
- ▶ Unter GPL u.a.
- ▶ Meist viele Entwickler (Basarmodell)
- ▶ Frühe und regelmässige Releases
- ▶ Darf modifiziert werden
- ▶ Kostenlose Updates und volle Funktionalität

Closed Source

- ▶ Nicht quelloffen
- ▶ Proprietäre Lizenz
- ▶ Wenig Entwickler (ausser grosse Firmen)
- ▶ Lange Releasezyklen
- ▶ Nur Hersteller kann modifizieren (Macht über Software und Benutzer)
- ▶ Oft mit kostenpflichtigen Funktionen und Updates

Was spricht für FLOSS?

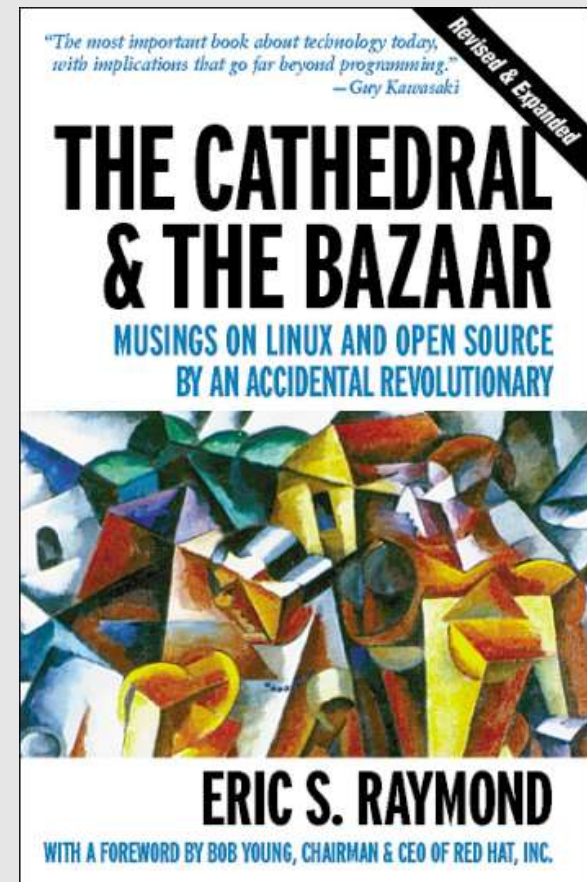
12

- ▶ Einige Punkte wurden bereits aus den Beweggründen deutlich
- ▶ Worin sieht sich die FLOSS Gemeinschaft sonst noch im Vorteil?

Was spricht für FLOSS?

13

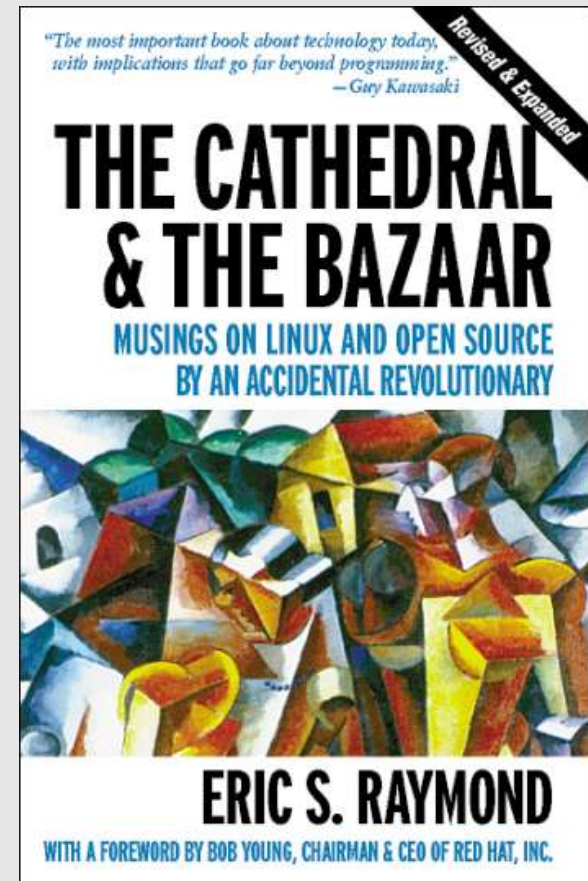
- ▶ Eric S. Raymond stellte 1999 in seinem Werk «The Cathedral and the Bazaar» einige Vorteile (und Nachteile) für Open Source vor



Was spricht für FLOSS?

14

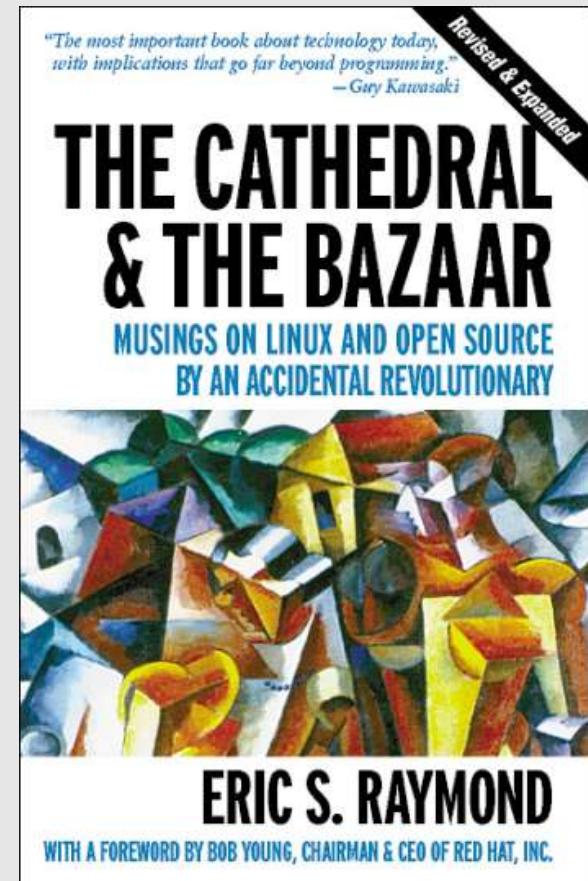
- ▶ Code sollte auch offen sein, damit «grossartige Programmierer» bestehende Software benutzen und darauf bauen können
- ▶ FLOSS kann weitergegeben und wiederverbraucht werden
- ▶ Viele Nutzer sind auch Hacker → Mitentwickler und Effizienz



Was spricht für FLOSS?

15

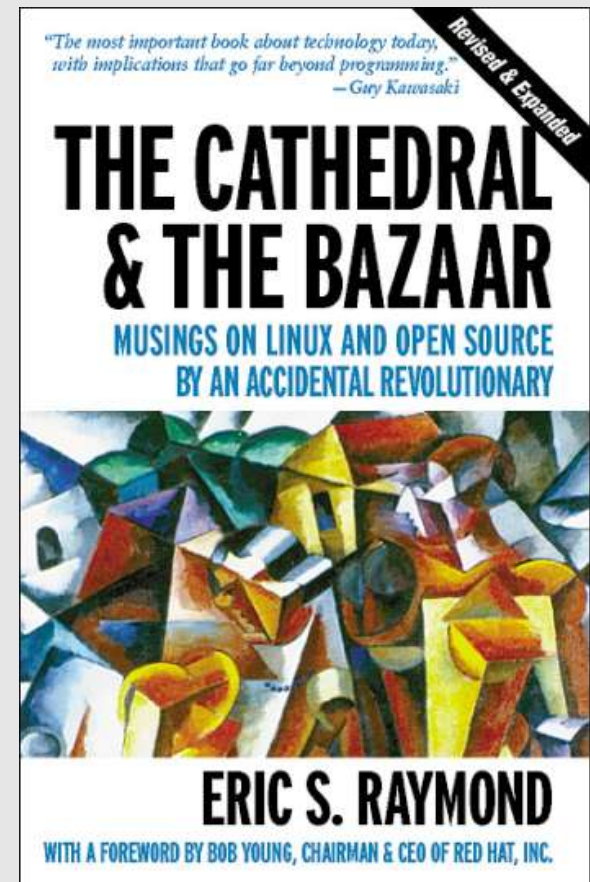
- ▶ «Alle Bugs sind trivial, wenn man nur genügend Entwickler hat.»
- ▶ Viele Benutzer und Mitentwickler tragen also zu Korrekturen sehr gut bei
- ▶ Mitentwickler können eigene Ideen und (Code) Vorschläge präsentieren → Nur möglich bei offenem Code



Was spricht für FLOSS?

16

- ▶ Auch auf höherer Ebene, beim Entwurf eines Projekts, können viele Mitentwickler helfen. (Ameisen beim Essenssuchen)
- ▶ Open Source Entwickler sind aus eigenem Interesse am Projekt
- ▶ Open Source ist also ökonomisch effektiver



Was spricht für FLOSS?

- ▶ Die Open Source Gemeinschaft glaubt zu wissen, dass die Codequalität besser ist.
- ▶ Keine versteckte Spionage
- ▶ Zu guter Letzt ist FLOSS meistens auch günstiger, vor allem dann, wenn man keine Dienste braucht. Bei proprietärer Software muss man meistens alleine schon für die Software viel zahlen.

Was spricht für Closed Source?

- ▶ Garantie und Support
- ▶ Kontinuität ist oft sicherer (Planungssicherheit)
- ▶ Weite Verbreitung und benutzerfreundlicher → Weniger Schulungsaufwand
- ▶ Weit bessere Kompatibilität und Treibersupport mit kommerziellen Hardwareherstellern
 - ▶ Linux schwächelt z.B. bei Grafiktreiber
- ▶ Einfachere Integration mit vorhandener (kommerzieller) Software
- ▶ «Ein Sicherheitssystem ist nur so sicher wie seine Geheimnisse.» - Eric S. Raymond

FLOSS vs. Closed Source im Überblick

FLOSS

- ▶ Quelloffen
- ▶ Anpassbar
- ▶ Teilen
- ▶ Oft günstiger
- ▶ Bugs schneller gefunden und gefixt
- ▶ Entwickler haben Spass
- ▶ Mangelnder Support
- ▶ Unsichere Kontinuität

Closed Source

- ▶ Garantie & Support
- ▶ Planungssicherheit
- ▶ Weit verbreitet
- ▶ Benutzerfreundlicher und einfacher
- ▶ Geheimer Code für Sicherheitssystem
- ▶ Nicht anpassbar
- ▶ Oft teuer
- ▶ Geheime Spionage möglich

Was ist mit der Codequalität?

Hat FLOSS weniger Bugs?

20

- ▶ Schwierig zu sagen
- ▶ Aber man geht davon aus, dass beim Basarmodell Bugs schneller gefunden und gefixt werden.
- ▶ Durch frühe und häufige Releases blamiert man sich nicht, auch wenn Bugs im Code sind

Codequalität

21

- ▶ Coverity misst Codequalität durch Scannen von Code
- ▶ Viele Projekte werden gescannt
- ▶ **Frage:** Wie ernst soll man das nehmen?



Codequalität

DEFECTS WE FIND INCLUDE



Crashes



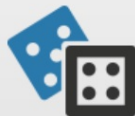
Security
Vulnerabilities



Concurrency



Memory
Corruption



Uninitialized
memory



Error handling



Resource leaks

LANGUAGES WE SUPPORT

C/C++

Java

C#

Codequalität

- ▶ Coverity Studie zur Codequalität
- ▶ Fehlerdichte (Fehler pro 1000 Codezeilen)
- ▶ Fehlerdichte unter 1.0 gilt als hohe Qualität
- ▶ Report 2013: Qualität quelloffener Software leidet ab einer Million Zeilen. Proprietäre Software ab dann besser.
- ▶ Report 2014: Open Source Software zeigt grundsätzlich bessere Resultate als Proprietäre.
- ▶ 1500 Projekte überprüft, darunter 700 quelloffen.
- ▶ C/C++ Fehler werden ernster genommen als bei Java.

Codequalität: Coverity Report 2014

Open Source

▶ Fehlerdichte:

0.59

Proprietär

▶ Fehlerdichte:

0.72

Ist FLOSS sicherer?

- ▶ Kann man nicht so einfach sagen
- ▶ Heartbleed (27 Monate bestehend)
- ▶ Beide Modelle können Fehler und Sicherheitslücken haben
- ▶ Hängt auch vom Qualitätsmanagement ab oder von der Teamgröße

Literatur und Quellen

- ▶ GNU Manifest (<http://www.gnu.org/gnu/manifesto>)
- ▶ Über das Projekt «GNU» (<http://www.gnu.org/gnu/thegnuproject>)
- ▶ The Cathedral and the Bazaar
(<http://www.catb.org/esr/writings/cathedral-bazaar/>)
- ▶ http://wiki.opensourceecology.de/Vorteile/Nachteile_von_Open_Source_Software
- ▶ <http://dellekom.de/blog/warum-open-source-besser-ist>
- ▶ <http://www.heise.de/newsticker/meldung/Studie-zur-Softwarequalitaet-Open-Source-schlaegt-proprietar-2175954.html>
- ▶ <http://www.coverity.com/press-releases/coverity-finds-python-sets-new-level-of-quality-for-open-source-software/>
- ▶ <http://www.heise.de/open/artikel/Hindernisse-224510.html>
- ▶ <http://www.coverity.com/why-coverity/#Developers>
- ▶ <http://de.wikipedia.org/wiki/Heartbleed>

Bilderquellen

- ▶ <http://icons.iconarchive.com/icons/icons-land/sport/256/Boxing-Gloves-icon.png>
- ▶ http://upload.wikimedia.org/wikipedia/commons/c/c7/Windows_logo_-_2012.png
- ▶ http://upload.wikimedia.org/wikipedia/en/4/45/Mac_Finder_icon_%28OS_X_Yosemite%29.png
- ▶ <http://upload.wikimedia.org/wikipedia/commons/3/35/Tux.svg>
- ▶ http://upload.wikimedia.org/wikipedia/en/1/19/Ffmpeg_new_logo.png
- ▶ <http://download.blender.org/institute/logos/blender-plain.png>
- ▶ http://upload.wikimedia.org/wikipedia/commons/3/38/VLC_icon.png
- ▶ http://upload.wikimedia.org/wikipedia/de/thumb/8/83/Avira_Antivirus_Logo.svg/2000px-Avira_Antivirus_Logo.svg.png
- ▶ <http://seattlewriter.com/wp-content/uploads/2013/01/Microsoft-Office-logo-2012.png>
- ▶ https://wiki.documentfoundation.org/images/7/74/LibreOffice_Initial-Artwork-Logo_ColorLogoBasic_2000px.png
- ▶ http://upload.wikimedia.org/wikipedia/commons/f/fc/iTunes_12_logo.png
- ▶ http://upload.wikimedia.org/wikipedia/commons/e/ee/Internet_Explorer_logo.png
- ▶ http://upload.wikimedia.org/wikipedia/fr/archive/d/df/20130818094908!Firefox_2013_logo.png

Bilderquellen

- ▶ <http://teknodevir.com/wp-content/uploads/2014/08/Microsoft-DirectX-logo.png>
- ▶ <http://upload.wikimedia.org/wikipedia/de/thumb/e/e9/OpenGL-logo.svg/512px-OpenGL-logo.svg.png>
- ▶ https://stallman.org/Portrait_-_Denmark_DTU_2007-3-31.jpg
- ▶ http://upload.wikimedia.org/wikipedia/commons/a/a3/XC68020_top_p1160084.jpg
- ▶ http://www.aguaplus.ch/files/media/02_Unternehmen/Leitbild/Freiheit.jpg
- ▶ http://upload.wikimedia.org/wikipedia/commons/thumb/e/e2/Eric_S_Raymond_portrait.jpg/220px-Eric_S_Raymond_portrait.jpg
- ▶ <https://plus.google.com/u/0/+LinusTorvalds/posts?pid=5627237652700051890&oid=102150693225130002912>
- ▶ <http://www.seattlestar.net/wp-content/uploads/2013/12/Cathedral-and-the-Bazaar-book-cover.jpg>
- ▶ http://upload.wikimedia.org/wikipedia/de/thumb/3/3b/Coverity_Prevent_logo.svg/2000px-Coverity_Prevent_logo.svg.png

Ende

Diskussion