

Praxen Freier Software. Wie Freie Softwareprojekte funktionieren

Vortrag im Seminar
„Wissen in der modernen Gesellschaft“

Yuan Zhang, 11.06.2009

Was ist freie Software?

- Freie Software ist Software, die für jeden Zweck genutzt, studiert, bearbeitet und in ursprünglicher oder veränderter Form weiterverbreitet werden darf.
 - Die Software darf ohne Einschränkungen benutzt werden.
 - Der Quellcode freier Software ist verfügbar darf studiert und aus ihm darf gelernt werden.
 - Sie darf ohne Einschränkungen und ohne Zahlungsverpflichtungen kopiert und weitergegeben werden.
 - Sie darf verändert und in veränderter Form weitergegeben werden.

Was ist freie Software?

- Jedes größere Softwareprojekt wird von Gruppen von Entwicklern erstellt. Auch in der Industrie hat man heute Teams,
 - die über eine kreative Selbstständigkeit verfügen,
 - die ihren Code synchronisieren und in regelmäßigen Abständen das gemeinsame Produkt stabilisieren,
 - In denen hierarchische Verfahren unter einem Chefprogrammierer weitgehend abgelöst sind
- Die freie Software hat dank der genannten Eigenschaften und dank des Internet die Zusammenarbeit auf alle ausgeweitet, die sich beteiligen möchten.

Die Leistungen Freier Software

- 70% aller Webserver laufen auf dem freien Apache.
- Qualität und Zuverlässigkeit Freier Software ist meist höher als bei vergleichbaren kommerziellen Produkten.

Quellcode und Objektcode

- Ein Programm hat immer zwei Sätze von Code, Quellcode und Objekt-Code.
 - Quellcode
 - Quellcode ist das, was man vorher erstellt, die Sprache, die der Programmierer verwendet, um entsprechende Anweisungen an den Compiler zu erteilen, wie das Programm laufen soll. Quellcode ist Voraussetzung für Veränderungsfreiheit
 - Objektcode
 - Objektcode ist das Ergebnis des Compilers aus der Transformation dieser Quellcode-Anweisungen.

Die Begriffe sind intuitiv, der Quellcode ist das Mittel zum Zweck. Mit anderen Worten, der Quellcode ist der Anfang, oder die Quelle, der Betrieb und der Objekt-Code ist das gewünschte Ergebnis, oder das Objekt, das Ziel der gesamten Übung. Objekt-Code wird durch den Compiler erzeugt, in Dateien gespeichert und kann als das ultimative Ende der Arbeit des Programmierers betrachtet werden.

Wie funktioniert ein Projekt der freie Software?

- Notwendigkeit ist die Mutter aller Erfindungen
 - Jedes gute Softwarewerk beginnt damit, dass ein Entwickler ein ihn persönlich betreffendes Problem angeht (Raymond, 1998)
 - David Harris arbeitete an einer Universität in Dunedin, die 1989 ein Novell NetWare-Netzwerk installierte, nur um festzustellen, dass es kein E-Mailsystem enthielt. **Aber** die kommerziellen E-Mailpakete waren sehr teuer, also schrieb Harris in seiner Freizeit ein einfaches Programm namens >>Pegasus Mail<<, das er seither zu einem mächtigen und komfortablen Werkzeug weiterentwickelt hat und weiterhin verschenkt.

Wie funktioniert ein Projekt der freie Software?

- Freie Software entsteht also zunächst nicht auf Anweisung eines Vorgesetzten oder Auftraggebers.
 - Sie ist viel mehr eine eigen-motivierte Tätigkeit, angetrieben von dem Wunsch, ein auf der Hand liegendes Problem bei der Arbeit oder Forschung zu lösen.

Core-Team und Maintainer

- Core-Team
 - Wenn die Zahl der Mitentwickler und Anwender wächst, bildet diese Gruppe das zentrale Steuerungsgremium des Projekts.
 - Ein solches Team rekrutiert sich meist aus den Leuten, die entweder schon am längsten daran arbeiten oder sehr viel daran gearbeitet haben oder derzeit am aktivsten sind.
 - Innerhalb des Core-Team werden Entscheidung über die allgemeine Richtung der Entwicklung gefällt, über Fragen des Designs und über interessante Probleme, an denen weitergearbeitet werden soll.
 - Große Projekte werden in funktionale Einheiten, in "Packages" oder Module gegliedert, für die jeweils ein oder mehrere Maintainer zuständig sind.
 - Z.B.
 - XFree86 hat ein Core-Team von elf Personen und eine Community von etwa 600 Entwickler
 - Beim Apache umfasst es 22 Leute aus sechs Ländern.

Core-Team und Maintainer

- Maintainer
 - Die Rolle eines Maintainers ist es zunächst, Ansprechpartner für die jeweilige Software zu sein.
 - Häufig handelt es sich um altgediente Coder, die sich eine Reputation erworben haben.
 - Als treibende Kraft wirken, die Gemeinschaft koordinieren.
 - An jedem Einzelprojekt arbeiten mehrere Dutzend bis hundert Entwickler weltweit mit. Änderungen werden an das Core-Team geschickt und von diesem in den Quellcode integriert.

Die Community

- jeder macht, wozu er Lust hat.....
 - Die neueren Projekte wie Linux messen der Entwicklergemeinschaft einen größeren Stellenwert bei.
 - Stallman: >>GNU ..ist..das Softwaresystem, das ich schreibe, um es frei weiterzugeben mehrere andere Freiwillige helfen mir dabei<<
 - Der Pflege der Community, die die Basis eines Projektes bildet, wird mehr Aufmerksamkeit gewidmet.
 - Das Debian Team besteht aus etwa 500 Mitarbeitern weltweit.
 - Bei Xfree86 sind es rund 600
 - Bei den meisten freien Softwareprojekten gibt es keine festgelegte Aufgabenverteilung.
 - Jeder macht das, was ihn interessiert und programmiert oder implementiert das, wozu er Lust hat.

Entscheidungsfindung: rough consensus and running code

- Philosophie in den meisten freien Softwareprojekten

>>wir wollen keine Könige, Präsidenten und Wahlen. Wir glauben an einen groben Konsens und an ablauffähigen Code<< (Das Credo der Internet-Entwicklergemeinde)

Die gleiche Philosophie herrscht auch in den meisten freien Softwareprojekten. »Grober Konsens, ablauffähiger Code« drückt gut aus, wie ein Projekt zur Programmierung Freier Software startet.

- Grober Konsens : keine formalen Abstimmungen
- *ablauffähiger Code*: Implementation neuer Technologien statt nur Beschreibung


Code-Forking

- Die Freiheit, eigene Wege zu gehen
 - Das eben Gesagte bedeutet nicht, dass Konflikte prinzipiell vermieden werden können.
 - Es hat verschiedentlich Spaltungen von Projekten gegeben, bei denen sich mit der Entwicklungsgruppe natürlich auch die Code-Basis verzweigt.
 - im schlimmsten Fall bedeutet dies den Tod eines Projektes, oder es entstehen daraus zwei ähnliche Projekte, die um Entwickler und Anwender konkurrieren
 - die Gesamtbemühungen verdoppeln sich
 - im günstigsten Fall kann eine Spaltung fruchtbar sein
 - die Möglichkeit, sich jederzeit von einem Projekt abzusetzen und es in eine eigene Richtung weiterzutreiben, wird auch als heilsam erachtet.
 - wenn die Entwicklung am Bedarf von ausreichend vielen vorbei geht, kommt es irgendwann zur Verzweigung

Die Werkzeuge

- Die zentralen Kommunikationsmittel für die weltweit ortsverteilte Kooperation sind E-Mail, genauer Mailinglisten sowie Newsgroups.
 - Für Echtzeitkommunikation verwenden einige Projekte auch den IRC (Internet Relay Chat)
- Die Projekte präsentieren sich und ihre Ressourcen auf Websites.
- CVS-Server (Concurrent Versions System)
 - Das zentrale Instrument zur kooperativen Verwaltung des Quellcode
 - CVS ist ein mächtiges Werkzeug für die Revisionsverwaltung von Softwareprojekten, das es Gruppen von Entwicklern erlaubt, gleichzeitig an denselben Dateien zu arbeiten, sich zu koordinieren und einen Überblick über die Veränderungen zu behalten.
 - CVS ist Standard bei freier Software

Wie CVS funktioniert

- Jeder kann lesen weltweit auf die "Quellcodebäume" zugreifen und sich aktuellste Version einer Software auf seine lokale Festplatte kopieren
 - wer die Software weiterentwickeln möchte, muss sich registrieren, um für andere Entwickler ansprechbar zu sein.
 - Dateien liegt in einem gemeinsamen Verzeichnis dem "Repository"
 - Man führt den "Checkout-Befehl" aus, um mit der Arbeit zu beginnen
 - Und übergibt den Verzeichnispfad oder den Namen des Moduls, an dem man arbeiten möchte
 - CVS kopiert dann die letzte Fassung der gewünschten Datei aus dem Repository in einen Verzeichnisbaum auf der lokalen Festplatte.
 - Der Entwickler kann diese Dateien nun mit einem Editor seine Wahl verändern, sie in eine Output-Datei >>bauen<< und das Ergebnis testen.
 - Ist er mit dem Ergebnis zufrieden, schreibt er es mit dem "Commit-Befehl" in das Repository zurück und macht damit seine Änderungen anderen Entwicklern zugänglich.
- 

Wie CVS funktioniert

- Wenn andere Entwickler zur selben Zeit dieselben Dateien bearbeiten, werden die verschiedenen neuen Versionen lokal mit dem "Update-Befehl" verschmolzen.
 - Lässt sich das Ergebnis korrekt bauen und testen, werden die verschmolzenen Dateien gleichfalls in den CVS-Baum zurückgeschrieben.
 - Ist das nicht automatisch möglich, müssen sich die beteiligten Entwickler ueber die Integration ihrer Änderungen untereinander verständigen.

Vorteil

Die Methode >>kopieren – verändern – zusammenfügen<< erfordert keine Sperrung der Quelldateien, die gerade von einem Entwickler bearbeitet werden.

Debugging

- Software enthält Fehler (bugs)
 - Einer Informatiker-Faustregel zufolge ist in jedem Programm pro 100 Zeilen Code ein Bug
 - Das Verfahren von Proprietärer Software
 - Neal Stephenson :
 - >>kommerzielle Betriebssysteme müssen die gleiche offizielle Haltung gegenüber Fehlern einnehmen, wie sie die kommunistischen Länder gegenüber der Armut hatten. Aus doktrinären Gründen war es nicht möglich zuzugeben, dass Armut in den kommunistischen Ländern ein ernstes Problem darstellte, weil der Kommunismus sich ja gerade zum Ziel gesetzt hatte, die Armut zu beseitigen.....<<

Debugging

- Software enthält Fehler (bugs)
 - Das Verfahren von Freien Softwareprojekten mit Bugs umzugehen stellt einen der wichtigsten Vorteile gegenüber proprietärer Software dar.
 - Die Stabilität dieser Software, d.h. ihr Grad an Fehlerfreiheit, verdankt sie nicht der Genialität ihrer Entwickler, sondern der Tatsache, dass jeder Anwender Fehler an den Pranger stellen kann und die kollektive Intelligenz von Hunderten von Entwicklern meist sehr schnell eine Lösung findet.
 - Dadurch können sich Tausende von Nutzern am Auffinden von Bugs und Hunderte von Entwicklern an ihrer Lösung beteiligen.
 - Das Debugging kann hochgradig parallelisiert werden, ohne dass der positive Effekt durch erhöhten Koordinationsaufwand und steigende Komplexität konterkariert würde.
 - Wer z.B. in GNU/Linux einem Bug begegnet : E-Mailbericht an die Entwickler schicken.

Die Releases

- **Schnelle Entwicklerversionen und stabile Produktionsversionen**
 - Durch die beschleunigten Innovationszyklen in der Informations- und Kommunikationstechnologie und hier besonders in der Software werden Produkte in immer kürzeren Abständen durch neue Versionen obsolet gemacht.
 - Z.B.
 - Version 2 auf Version 3 → markiert ein ganz neues Design
 - Version 3.3.1 zu Version 3.3.2 → stellt einen kleineren Integrationsschritt dar.
- **Freie Software ist kein Produkt, sondern ein Prozess**
 - Freie Software wird heute als kontinuierlicher Prozess verstanden.
 - Z.B. eine Entwicklergemeinschaft kann ihre Software über lange Zeit in hoher Qualität unterstützen und weiterentwickeln.
 - Abhängig von der Geschwindigkeit werden täglich oder im Abstand von Wochen oder Monaten neue Releases herausgegeben.

Die Motivation: Wer sind die Leute und warum machen die das..., wenn nicht für Geld?

- Selbsttätiges Schaffen als Wert an sich
- Freie Zeit für freie Software
- Die Lust des Lernens
- Kritik und Bestätigung von der Community
- Selbstmotivation und Geld verdienen schließen sich nicht aus.
- Lieber freischaffend als angestellt

Softwarezyklus: Entwickler, Power-User, Endnutzer

Man kann drei große Rollen in der freien Softwareprojekts unterscheiden.

- **Entwickler:**
 - Die Software ist noch nicht mehr als eine rohe Skizze. Jeder kann sich die Software beschaffen, aber nur wer selbst am Quellcode mitarbeiten möchte.
- **Power-User**
 - Die Software hat ein Stabilität erlangt, die sie für *Power-User* (z.B. Systemadministratoren) interessant macht, die sie nicht primär weiterentwickeln, sondern einsetzen wollen.
 - Von den Hackern zu den Anwendern
 - Installation, Integration und Wartung erfordern auch jetzt noch weiter reichende Programmierfähigkeiten, doch diese Nutzergruppe erhält dafür eine Funktionalität, Flexibilität und ein Maß an Kontrolle über die Software, die sie zu einer attraktiven Alternative zu proprietärer Software macht.
 - Hat die Software einen Grad an Stabilität erreicht, die sie für den praktischen Einsatz geeignet macht, gibt das Projekt ein offizielles Release heraus.

Softwarezyklus: Entwickler, Power-User, Endnutzer

- Endnutzer
 - Die Software erhält Merkmale wie Installationshilfen und Dokumentation, die sie schließlich auch für technisch weniger bedarfte Nutzer zugänglich macht.
 - Dieser Schritt wird oft nicht mehr von den selbst motivierten Teilnehmern des freien Projekts durchgeführt, sondern von Dienstleistungsfirmen, die oft eigens für den Support von freier Software gegründet wurden.

Ich danke Ihnen
für Ihre Aufmerksamkeit.