

Handout zum Vortrag:

# FLOSS vs. Closed Source Software

## Beweggründe der FLOSS Gemeinschaft

Closed Source Software grenzt die Freiheit eines Benutzers, sei es ein Entwickler oder ein Nutzer, ein. Mit dieser Tatsache konfrontierte in den 80er Jahren Richard Stallman, der Gründer der Free Software Foundation, als neuerdings im AI Lab des MIT proprietäre Software benutzt wurde. Freiheiten, wie das Teilen, Weitergeben, Kopieren oder Anpassen einer Software waren durch proprietäre Lizizenzen und Geheimhaltung des Quellcodes nicht mehr erlaubt. Stallman verzichtete darauf, so weiter zu machen und seine Freunde zu verraten, indem er mit ihnen nicht mehr teilen konnte und kündigte deswegen 1984 seinen Job am MIT. Er startete das GNU Projekt mit der Hoffnung, dass ein Programmierer etwas Gutes tun und die Freiheit jedes Benutzers schützen könnte.

Eine weitere wichtige Bewegung fand bei Linus Torvalds und seinem Linux Projekt statt. Im Jahr 1992 benutzte er nämlich das sogenannte Basarmodell bei Linux, was zu einem Erfolg führte. Das von der Open Source Welt bekannte Basarmodell zielt auf die Entwicklung eines Projekts mit frühen und häufigen Releases. Dabei ist der Quellcode immer zugänglich. Eric S. Raymond schrieb in seinem Werk „The Cathedral and the Bazaar“ über seine Erfahrungen beim Verwenden des Basarmodells bei seinem Projekt „fetchmail“. Er machte also Torvalds nach und hoffte auf Erfolg, den er tatsächlich auch erlangen konnte.

## Was unterscheidet die beiden Modelle?

Aus den Beweggründen von FLOSS können die Unterschiede zwischen den beiden Modellen recht gut verdeutlicht werden:

FLOSS/OSS	Closed Source Software
Quelloffen	Nicht quelloffen
Unter einer freien Lizenz (z.B. GPL)	Proprietäre Lizenz
Meist viele Entwickler (Basarmodell)	Wenig Entwickler (ausser grosse Firmen)
Frühe und häufige Releases	Lange Release Zyklen
Darf modifiziert werden	Nur Hersteller kann modifizieren (Macht über Software und Benutzer)
Kostenlose Updates und volle Funktionalität	Oft mit kostenpflichtigen Funktionen und Updates

In der letzten Zeile ist ein neuer, häufig gesehener Unterschied dazugekommen. Manch einer könnte bestimmt ein Lied davon singen, ginge es um kommerzielle Software, bei der viele Funktionen erst mit einem zusätzlichen Kauf freigestellt werden.

Welcher wichtige Punkt die beiden Modelle jedoch nicht unterscheidet, ist der Preis. FLOSS bedeutet nämlich nicht, dass die Software gratis sein muss. Eine Kopie der Software darf also gegen eine Gebühr ausgeteilt werden.

## Vor- und Nachteile der beiden Modelle

Möchte man die Vor- und Nachteile der beiden Modelle anschauen, sprechen die Rechte auf Freiheit und Zusammenarbeit klar für FLOSS. Denn nicht nur der offene Quellcode und die freie Lizenz gewähren und erleichtern einem Benutzer, Anpassungen an der Software vorzunehmen, sondern

auch das erlaubte Teilen und Weitergeben von Modifikationen. Bei einer Closed Source Software sind solche Freiheiten entweder gar nicht erst möglich, weil der Quellcode nicht frei verfügbar ist, oder durch eine proprietäre Lizenz explizit verboten.

Oft sind viele Entwickler, nicht selten auch viele Mitentwickler aus dem Benutzerkreis, an einem FLOSS Projekt beteiligt (Basarmodell), was das schnelle und leichte Auffinden und Fixen von Bugs ermöglicht. Außerdem sind die Entwickler aus eigenem Interesse dabei. Wenn das nicht ein Grund für Innovation und Effektivität bei einer kreativen Arbeit ist.

Zu guter Letzt kann FLOSS durch den offenen Quelltext keine verdeckte Spionage führen.

Auf der Closed Source Software Seite gelten als wichtige Vorteile ein fast rund um die Uhr verfügbarer, professioneller Support, eine Garantie auf Planung und Verträge und ein geringerer Schulungsaufwand durch die hohe Verbreitung und oft benutzerfreundlichere Software. Des Weiteren bieten kommerzielle Produkte eine bessere Kompatibilität auf andere kommerzielle Software oder Hardware. Folglich haben viele Unternehmen Schwierigkeiten bei der Einführung und Integration von Open Source Software mit vorhandener kommerzieller Software.

## Bugs

Auf die Frage, welches Modell weniger Bugs hat, kann nicht einfach eine generell richtige Antwort gegeben werden. Es hängt auch von vielen Faktoren ab, wie die Teamgrösse, das Management etc. Aber man geht davon aus, dass beim Basarmodell, also in der Open Source Welt, Bugs schneller gefunden und gefixt werden.

## Codequalität

Viele Meinungen, auch wenn hauptsächlich von der FLOSS Gemeinschaft stammend, unterstützen die Annahme, dass Open Source Software die bessere Codequality hätte. Aber auch hier kann man keine allgemeingültige Antwort definieren. Empirische Studien, wie die Reporte von Coverity, messen die Fehlerdichte (Fehler pro 1000 Codezeilen) von Projekten durch das automatisierte Scannen des Quellcodes. Dabei sollen beispielsweise Ressourcenlecks, Sicherheitslücken, Abstürze und andere Fehler erkannt werden. Solche Studien sollte man aber mit Vorsicht geniessen.

## Sicherheit

Leider ist auch beim Thema Sicherheit nicht ein klarer Gewinner ersichtlich. Absolut hemmungslos kann man aber sagen, dass beide Modelle Sicherheitslücken haben können. Die Sicherheit eines Projekts hängt nicht nur einfach vom Entwicklungsmodell ab, sondern auch von vielen anderen Dingen, wie vom Qualitätsmanagement oder von der Teamgrösse und deren Aktivität.

## Literaturverzeichnis

Diedrich, O. (04. Februar 2009). *Trendstudie Open Source*. Von heise online:

<http://www.heise.de/open/artikel/Trendstudie-Open-Source-221696.html> abgerufen

Raymond, E. S. (1999). *The Cathedral and the Bazaar*. O'Reilly Media.

Stallman, R. (1985). *GNU-Manifest*. Von <http://www.gnu.org/gnu manifesto> abgerufen

Stallman, R. (1998). *Über das Projekt ‚GNU‘*. Von <http://www.gnu.org/gnu/the gnuproject> abgerufen

*Studie zur Softwarequalität*. (24. April 2014). Von heise online:

<http://www.heise.de/newsticker/meldung/Studie-zur-Softwarequalitaet-Open-Source-schlaegt-proprietaer-2175954.html> abgerufen