

# Open Source und Freie Software

–

## Gemeinsamkeiten und Unterschiede im historischen Kontext

Rico Warnke

Seminar „Wissen in der modernen Gesellschaft“

11.06.2020

# Gliederung

## 1. Begriffserklärung

- Proprietäre Software
- Freiheit
- Freie Software
- Open Source Software

## 2. Entstehungsgeschichte

- Vorgeschichte und das GNU-Projekt
- Bildung der Open-Source-Bewegung

## 3. Vergleich beider Modelle

- Gemeinsamkeiten
- Unterschiede

## 4. Gemeinschaftliche Diskussion

# 1. BEGRIFFSERKLÄRUNG

PROPRIETÄRE SOFTWARE

# Proprietäre Software

Software ist proprietär/unfrei  
(im Eigentum befindlich)  
wenn sie:

Das Recht und die Möglichkeit der  
**Anpassung,**  
**Änderung,**  
**Weiter- und Wiederverwendung**  
stark eingeschränkt

Möglichkeiten der Proprietärisierung:

- Softwarepatente
- Urheberrecht
- Lizenzbedingungen
- herstellerepezifischer Aufbau
- Quelltext als Betriebsgeheimnis

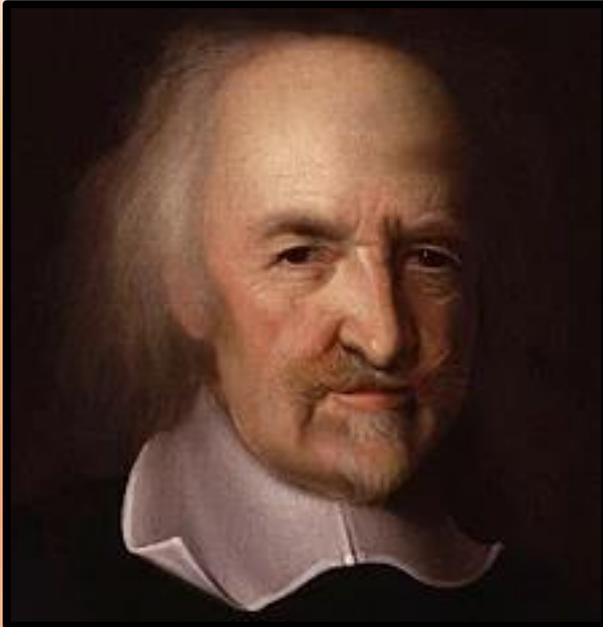
Beispiele: Microsoft Windows, MacOS

Quelle:  
[https://wiki.ubuntuusers.de/unfreie\\_Software/?rev=231304](https://wiki.ubuntuusers.de/unfreie_Software/?rev=231304)

# 1. BEGRIFFSERKLÄRUNG

## DER BEGRIFF DER FREIHEIT

# Freiheit nach Hobbes



Thomas Hobbes  
(1588-1679)

Quelle:

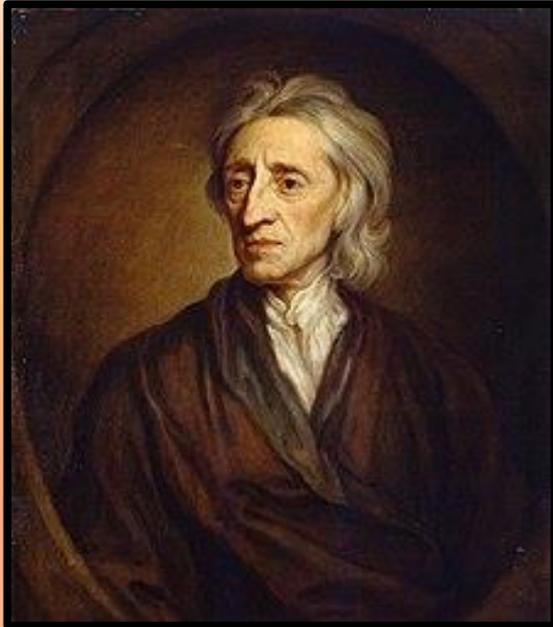
[https://de.wikipedia.org/wiki/Datei:Thomas\\_Hobbes\\_\(portrait\).jpg](https://de.wikipedia.org/wiki/Datei:Thomas_Hobbes_(portrait).jpg)

„**Freiheit bedeutet das Fehlen von Widerstand** und lässt sich nicht weniger auf unvernünftige und unbelebte Dinge als auf vernunftbegabte Geschöpfe anwenden.“

„Aber wenn man die Wörter **frei** und **Freiheit** auf **irgend etwas anderes als Körper** anwendet, mißbraucht man sie; denn was nicht der Bewegung unterliegt, unterliegt nicht der Behinderung; und wenn man daher sagt, der Weg ist frei, ist damit nicht die **Freiheit des Weges** gemeint, sondern jener, die **darauf gehen, ohne angehalten zu werden**. Und wenn wir sagen, ein Geschenk ist frei, so ist nicht die **Freiheit des Geschenks** gemeint, sondern **des Schenkenden, der nicht durch Gesetz oder Vertrag gehalten war, es zu geben**.“

Quelle: Hobbes: Leviathan S.177-178

# Freiheit nach Locke



John Locke  
(1632-1704)

Quelle:

[https://de.wikipedia.org/wiki/Datei:John\\_Locke.jpg](https://de.wikipedia.org/wiki/Datei:John_Locke.jpg)

„**Freiheit heißt frei sein von dem Zwang und der Gewalttätigkeit anderer**, was nicht sein kann, wo es keine Gesetze gibt; nicht aber, wie man sagt, eine Freiheit für jeden, zu tun, was er will; denn wer könnte frei sein, wenn die Laune jedes anderen ihn tyrannisieren dürfte? sondern **eine Freiheit, innerhalb der Grenzen der Gesetze**, unter welchen er steht, über seine Person, **Handlungen, Besitz** und gesamtes Eigentum zu verfügen und **damit zu tun, was ihm beliebt**, darin **nicht dem eigenmächtigen Willen eines anderen unterworfen zu sein**, sondern **frei dem eigenen zu folgen.**“

Quelle: [http://www.welcker-online.de/Texte/Locke/Locke\\_2.pdf](http://www.welcker-online.de/Texte/Locke/Locke_2.pdf) S.145

# Freiheit nach Kant



Immanuel Kant  
(1724-1804)

Quelle:

[https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/Kant\\_gemaelde\\_1.jpg/1280px-Kant\\_gemaelde\\_1.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/Kant_gemaelde_1.jpg/1280px-Kant_gemaelde_1.jpg)

„Die **Freiheit** im praktischen Verstande ist die **Unabhängigkeit der Willkür von der Nötigung durch Antriebe der Sinnlichkeit.**“

Quelle: Kant: Kritik der reinen Vernunft S.627

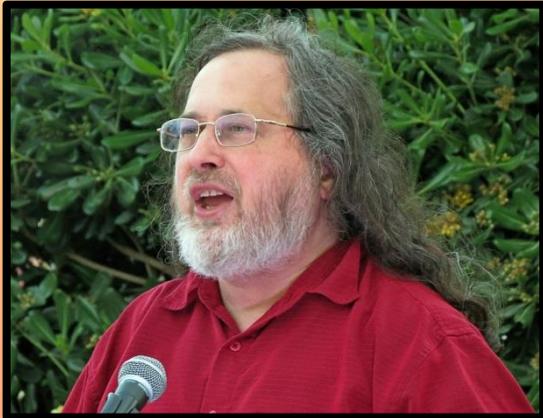
Eine Handlungsweise wird als **Willkür** bezeichnet, wenn der **Betroffene in einem Abhängigkeitsverhältnis zum Handelnden steht** und **der Handelnde sein Verhalten weder nach allgemeinen Regeln ausrichtet noch sein Verhalten gegenüber dem Betroffenen zu begründen beabsichtigt.**

Quelle: <https://www.spektrum.de/lexikon/philosophie/willkuer/2236>

# 1. BEGRIFFSERKLÄRUNG

FREIE SOFTWARE

# Freie Software



Richard M. Stallman

Quelle:

[https://de.wikipedia.org/wiki/Datei:Richard\\_Stallman\\_at\\_CommonsFest\\_Athens\\_2015\\_2.JPG](https://de.wikipedia.org/wiki/Datei:Richard_Stallman_at_CommonsFest_Athens_2015_2.JPG)

„Free as in free speech, not free beer“

„Frei wie in Redefreiheit, nicht wie in Freibier“

## Erste Definition von 1986

1. Freiheit, ein Programm zu kopieren und weiterzugeben
2. Freiheit, ein Programm zu ändern d.h. Zugang zum Quelltext

Quelle: <https://www.gnu.org/bulletins/bull1.txt>

# Freie Software

## Freiheit 0

Die Freiheit, die Software auszuführen wie man möchte, für jeden Zweck

## Freiheit 1

Die Freiheit, Programmfunktionen zu analysieren und anzupassen. Zugang zum Quelltext ist Voraussetzung

Freie Software muss für die kommerzielle Nutzung, Vertrieb und Entwicklung zugänglich sein. Man erhält die Freiheiten unabhängig davon, wie man eine Kopie des Programms erhalten hat.

## Freiheit 2

Die Freiheit, die Software weiterzugeben und damit seinen Mitmenschen zu helfen

## Freiheit 3

Die Freiheit, die Software zu modifizieren und Verbesserungen zu veröffentlichen

Quelle: <http://www.gnu.org/philosophy/free-sw>

# 1. BEGRIFFSERKLÄRUNG

OPEN SOURCE SOFTWARE

# Open Source Software



Eric S. Raymond

Quelle:  
[https://de.wikipedia.org/wiki/Datei:Eric\\_S\\_Raymond\\_portrait.jpg](https://de.wikipedia.org/wiki/Datei:Eric_S_Raymond_portrait.jpg)



Bruce Perens

Quelle:  
[https://de.wikipedia.org/wiki/Datei:3541705843\\_f896546bec\\_o\\_crop.jpg](https://de.wikipedia.org/wiki/Datei:3541705843_f896546bec_o_crop.jpg)

# Open Source Software

Open Source beinhaltet nicht nur Zugriff auf den Quelltext, auch die Weitergabe muss folgende Anforderungen erfüllen:

## **Freie Weiterverteilung**

Die Lizenz darf niemanden hindern, Software als Bestandteil einer aggregierten Software-Distribution, zu verkaufen oder zu verschenken. Dabei darf keine Lizenzgebühr erhoben werden.

## **Offener Quelltext**

Software muss im Quelltext verfügbar sein. Verbreitung im Quelltext als auch in kompilierter Form ist erlaubt. Zwischenformen sind nicht zulässig.

## **Abgeleitete Werke**

Die Lizenz muss Modifikationen und abgeleitete Werke zu lassen. Verbreitung muss unter gleichen Bedingungen wie die Originalsoftware erfolgen.

## **Integrität des Urheber-Quelltextes**

Lizenz muss Verbreitung von Software aus modifizierten Quelltext ausdrücklich erlauben. Lizenz darf verlangen, dass abgeleitete Werke einen anderen Namen oder andere Versionsnummern tragen.

# Open Source Software

## **Keine Benachteiligung von Personen/Gruppen**

Lizenz muss allen Personen oder Gruppen die Verwendung der Software gestatten.

## **Keine Einschränkung der Verwendung**

Lizenz darf niemanden hindern, die Software in einem bestimmten Bereich zu nutzen.

## **Verteilung der Lizenz**

Mit Programm verbundene Rechte müssen für alle gelten, ohne dass zusätzliche Lizenzen erworben werden müssen

## **Keine Produktspezifikation**

Mit Programm verbundene Rechte dürfen nicht davon abhängen, dass das Programm Teil einer bestimmten Software-Distribution ist.

## **Keine Einschränkung anderer Software**

Lizenz darf keine Einschränkungen für andere Software enthalten, die zusammen mit der lizenzierten Software vertrieben wird.

## **Technologieneutralität**

Keine Bestimmung der Lizenz darf auf individueller Technologie oder Art der Schnittstelle basieren

Quelle: <https://opensource.org/docs/osd>

# 2. ENTSTEHUNGSGESCHICHTE

## VORGESCHICHTE UND DAS GNU-PROJEKT

# Vorgeschichte und das GNU Projekt

## Bis 1969:

Es gilt als gängige Praxis, Software inklusive Quelltext mit Computersystemen auszuliefern. Software wird als kostenlose Beigabe für teure Hardware angesehen.

Offenlegung des Quelltext → Hersteller erhält vielerlei Fehlerkorrekturen und Modifikationsvorschläge

# Vorgeschichte und das GNU Projekt

## Juni 1969:

IBM kündigt an, Software aus den bisherigen Hardware-Nutzungsbedingungen zu entbinden und separat durch Lizenzverträge urheberrechtlich zu schützen.

Weiterentwicklung und Wartung ihrer Software ist nicht länger eine freie Dienstleistung.

→ eigener Wirtschaftsmarkt wird begründet

Weitere Unternehmen führen in den späten 1970er Jahren ebenfalls Softwarelizenzen ein

Möglichkeiten der Modifikation und Verteilung der Software werden beschränkt.

Auslieferung der Software erfolgt im Binärcode, zum Schutz der Software als Betriebsgeheimnis

Quelle: <https://www.heise.de/developer/meldung/Vor-50-Jahren-Unbundling-Impuls-fuer-eine-unabhaengige-Softwareindustrie-4453306.html>

# Vorgeschichte und das GNU Projekt

## Hacker

„Ein Hacker ist jemand, der versucht einen Weg zu finden, wie man mit einer Kaffeemaschine Toast zubereiten kann.“ ( Wau Holland, ein Gründer des Chaos Computer Clubs )

„**Hacken**“ ist also der Versuch, die Grenzen des Machbaren zu erkunden.

## 1970er Jahre:

Richard Stallman → Abteilung für künstliche Intelligenz am MIT

Gruppe von Programmierern/Hackern lebt eine Philosophie des unbegrenzten Informationsflusses

Eigene Software-Modifikationen und deren Quelltext zu teilen gilt als ganz natürlich

# Vorgeschichte und das GNU Projekt

Einführung proprietärer Software in den Laboren des MIT

Stallman beginnt mit der Entwicklung alternativer Software → Emacs (Texteditor)

Er empfindet den Verlust der Kontrolle des Benutzers über die Software als Einschränkung ihrer Freiheit.

## **1983:**

Bislang gilt Unix als quelloffenes Betriebssystem und die Entwicklung wurde maßgeblich durch US-Universitäten vorangetrieben

AT&T (American Telephone and Telegraph Company) vermarktet UNIX System V. , eine proprietäre Version ihres Unix

→ Zahllose Unix-Mitentwickler sind erzürnt

# Vorgeschichte und das GNU Projekt

## Stallman Schlüsselereignis

Anfang 1980er Jahre am MIT

Nutzung eines Xerox-Netzwerkdruckes in einem anderen Raum

→ Papierstau oder Drucker ging aus

Stallmans Vorhaben bestand in der Implementierung einer Funktion, die den Druckerstatus am Arbeitsplatz anzeigt

Seine Anfrage nach dem Quelltext des Druckertreibers wurde verweigert

Die Mauer „geistiges Eigentum“ macht eine Lösung des Problems unmöglich

→ Idee einer neuen Gemeinschaft zur freien Nutzung von Software

→ 1983 Initialisierung des GNU-Projektes

Quelle: Freie Software. Zwischen Privat- und Gemeineigentum S. 211-222

# Vorgeschichte und das GNU Projekt

## Das GNU-Projekt

Zur Nutzung eines Computers liegt seine Priorität auf einem Betriebssystem.

Als Grundlage dieses Betriebssystem wählt Stallman Unix, da es sich bereits bewährt hatte und über eine große Nutzergemeinschaft verfügt.

Die Bezeichnung GNU, steht als rekursives Akronym für „**G**NU is **n**ot **U**nix“.

### Ziel:

- Äquivalent zu Unix, ohne eine Zeile geschützten Unix-Code zu verwenden
- Freie Weiterentwicklung
- Freie Verwendung durch die Allgemeinheit

Quelle: <https://www.gnu.org/gnu/manifesto>

# Vorgeschichte und das GNU Projekt

## Die Free Software Foundation (FSF)

Gründung durch Richard Stallman im Jahr 1985

Zweck: Förderung freier Software und Finanzierung des GNU-Projektes

### Hauptaufgabe:

Juristische, personelle, technische und finanzielle Unterstützung des GNU-Projektes

Verwaltung der Lizenzen:

- General Public License (GPL, 1989)
- Lesser General Public License (LGPL, 1991)
- Affero General Public License (AGPL)
- Free Documentation License (FDL)

Quelle: <https://www.fsf.org/about/>

# Vorgeschichte und das GNU Projekt

## Bereitstellung Freier Software

Entlassung in die Gemeinfreiheit **Public Domain**, unterliegt nicht dem Urheberrecht

→ Ermöglicht teilen, Gefahr der Umwandlung in proprietäre Software

## **Copyleft-Methode**

→ Ermöglicht freie Nutzung und verlangt zusätzlich, dass alle erweiterten und modifizierten Versionen ebenfalls frei nutzbar sind

→ Anwendung der Copyleft-Methode in den GNU-Lizenzen

Quelle: <https://www.gnu.org/licenses/copyleft.html>

# Vorgeschichte und das GNU Projekt

## Die GNU-Lizenzen

- GPL: Gestattet lizenzgebührenfreie Verbreitung, Vervielfältigung und das Recht beliebige Veränderungen vorzunehmen. Dazu muss der Quelltext offengelegt werden. Des Weiteren fordert die GPL, dass jede veränderte Version ebenfalls der GPL unterliegt.
- LGPL: Die LGPL gestattet es, LGPL-Software in proprietäre Software einzubinden. Meist in Form einer dynamischen Programm-Bibliothek um Trennung von LGPL- und proprietärer Software zu gewährleisten.
- AGPL: Räumt das Recht ein, eine Downloadmöglichkeit für den Quelltext einer Software zu erhalten, auch wenn diese nur als Dienst auf einem Server betrieben wird.
- FDL: Copyleft-Lizenz für freie Dokumentation. Stellt Literatur die im Rahmen des GNU-Projektes geschaffen wurde unter eine ähnliche Lizenz wie die Software. Gestattet Vervielfältigung, Verbreitung und Veränderung des Werkes gegen die Verpflichtung den Autor zu nennen und abgeleitete Werke ebenfalls unter die FDL zu stellen.

Quelle: <https://www.gnu.org/licenses>

# Vorgeschichte und das GNU Projekt

## Linux

Bis 1990 waren die wichtigsten Komponenten des GNU-Systems entwickelt, mit Ausnahme eines Kernel.

Kernel: Betriebssystemkern, der Prozess- und Datenorganisation festlegt. Bildet die unterste Softwareschicht mit direktem Zugriff auf die Hardware. Grundlage der Software-Bestandteile des Betriebssystems.

Als Kernel war 1990 GNU HURD geplant, Version 0.1 erschien letztendlich im September 1996

Linus Torvalds kam GNU HURD zuvor und entwickelt auf Grundlage von Minix und mit Hilfe der GNU-Werkzeuge 1991 **Linux**, einen freien unixoiden Betriebssystemkern.

1992 stellte er diesen unter die GNU GPL

Quelle: Freie Software. Zwischen Privat- und Gemeineigentum S. 226-229

# Vorgeschichte und das GNU Projekt

Fast vollständiges GNU-System + Linux-Kernel → GNU/Linux Version 1.0 (1994)

zunehmende Beliebtheit

Jahr	Anzahl der Nutzer
1995	500.000
1996	1.500.000
1997	3.500.000
1998	7.500.000

Quelle: <https://www.schreiben10.com/referate/Informatik/6/Die-Entstehung-und-Entwicklung-des-Betriebssystems-Linux-reon.php>

Stallmans ursprüngliches Ziel eines freien Betriebssystems war erreicht, aber die Entwicklung schreitet weiter voran

# 2. ENTSTEHUNGSGESCHICHTE

## BILDUNG DER OPEN-SOURCE-BEWEGUNG

# Bildung der Open-Source- Bewegung

## Open Source als Entwicklungsmodell

Eric S. Raymonds Essay „The Cathedral & the Bazaar“ (1997)  
Schlüsseltext für begriffliche Wende

Stellt zentral gelenktes Projekt des Kathedralenbaus ( Microsoft Windows )  
dem selbstorganisierenden Gewirr eines Basars ( GNU/Linux ) gegenüber

Quelle: [https://www.selflinux.org/selflinux/pdf/die\\_kathedrale\\_und\\_der\\_basar.pdf](https://www.selflinux.org/selflinux/pdf/die_kathedrale_und_der_basar.pdf)

Auslöser: Netscape veröffentlicht Quelltext ihres Browsers „Netscape Navigator“

Quelle: <https://www.heise.de/newsticker/meldung/Netscape-Browser-gratis-Quelltext-frei-10711.html>

1998: Raymond und Perens gründen die „Open Source Initiative“

Neuer Marketing-Begriff „Open Source“ soll Wettbewerbsfähigkeit freier Software steigern

Ziel: Darstellung freier Software als geschäftsfreundlich, weniger ideologisch und frei von ethischen  
und sozialen Komponenten

Quelle: <https://opensource.org/history>

# 3. VERGLEICH BEIDER MODELLE

## GEMEINSAMKEITEN

# Gemeinsamkeiten

- Zwei eigenständige Bewegungen, mit eigenen Sichtweisen
- Vermittelt den Eindruck es handelt sich um kostenlose Software
- Öffentlich zugänglicher Quelltext
- Keine rechtlichen Nutzungseinschränkungen
- Geringe bzw. keine Initialkosten motivieren zu freiwilliger Mitarbeit
- Meist direkte Entwicklung von Endnutzer
- Volle Kontrolle über Anpassung des Endproduktes
- Effiziente Wartung und Weiterentwicklung
- Proprietäre Software wird als „Feind“ angesehen
- Deckt viele Bereiche ab, die für proprietäre Entwickler nicht wirtschaftlich wären
- Effiziente Verteilung der Entwicklungskosten und Risiken
- Entwicklung leistungsstarker und zuverlässiger Software, oft auch in Kooperation

# 3. VERGLEICH BEIDER MODELLE

UNTERSCHIEDE

# Unterschiede

## Freie Software

- Konzept spricht hauptsächlich Programmierer/Entwickler
- Stellt den Wert der Freiheit und dessen Vermittlung in den Vordergrund
- Proprietäre Software ist eine ethische Frage und stellt ein soziales Problem dar
- Qualifiziert sich auch als Open Source Software

## Open Source Software

- Ökonomisches Entwicklungsmodell, spricht hauptsächlich Geschäftsleute an
- Zeigt unmittelbar praktische Vorzüge der kommerziellen Nutzung ohne die Themen Freiheit und Ethik
- Proprietäre Software ist eine praktische Frage und stellt eine suboptimale Lösung dar
- Schließt zwar freie Software in, aber auch einige proprietäre Programme

# Unterschiede

Einige Open Source Lizenzen sind beschränkt

Beispielsweise Nichtgestattung der Erstellung modifizierter Versionen

Sollte der Quelltext eine Lizenz ohne Copyleft ausweisen, können ausführbare Dateien zusätzliche Bedingungen enthalten

Damit gelten sie als Open Source, wenn sie dem freigegebenem Quelltext entsprechen, nicht aber als freie Software

Tivosierung:            Software-Produkt erhält Prüfsignatur zur Beschränkung der Installation abweichender ausführbarer Dateien  
→ modifizierte Version nicht ausführbar  
Beispiel: Microsoft Visual Studio

Anfangs war man der Annahme, dass Open Source Software nicht in der Lage sein würde die Bedürfnisse des Marktes nach neuer Technologie zu befriedigen.

Quelle: <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>

Quelle: <http://www.gnu.org/philosophy/open-source-misses-the-point.html>

Vielen Dank für ihre  
Aufmerksamkeit

# 4. GEMEINSCHAFTLICHE DISKUSSION

Wirre Ideologie  
oder  
fortschrittlicher Entwicklungszweig?