

# Advanced Topics of Information Science

Prof. Dr. Wolfgang Semar ([wolfgang.semar@htwchur.ch](mailto:wolfgang.semar@htwchur.ch))  
Prof. Dr. Rainer Kuhlen ([rainer.kuhlen@uni-konstanz.de](mailto:rainer.kuhlen@uni-konstanz.de))

Hochschule für Technik und Wirtschaft Chur  
Schweizerisches Institut für Informationswissenschaft  
[Creative Commons Attribution-NonCommercial-ShareAlike 2.5 Switzerland License](https://creativecommons.org/licenses/by-nc-sa/2.5/ch/)



## Inhalt

---

- Spezielle Themen der IW die sonst nicht bearbeitet werden.
  - Open Source Software
  - Informationsökonomie, Industrie 4.0
  - Informationsethik

## Aufbau und Ziel der Veranstaltung

---

- Aufbau
  - 2 Blocks: OSS
  - 2 Blocks: Informationsökonomie
  - 2 Blocks: Informationsethik
    - 25.11. und 26.11. (Anwesenheitspflicht)
- Leistungsnachweis
  - Kühlen: Hausarbeit (30%)
  - Semar: Abschlussklausur (70%)

## Aufbau und Ziel der Veranstaltung

---

- Ziel - Semar
  - Wissen um die Probleme beim Einsatz von OSS
  - Wissen um die veränderten ökonomischen Verhältnisse bei digitalen Gütern
  - Wissen um neue Geschäftsmodelle zu entwickeln

# Agenda - ATIS

---

- I. Open Source Software - Einführung
- II. OSS-Lizenzen und Creative Commons
- III. Geschäftsmodelle für den Einsatz von OSS
  
- IV. Informationsökonomie – Einführung
- V. Prinzipien der InfÖk
- VI. Ökonomische Besonderheiten – Industrie 4.0
- VII. Geschäftsmodelle
- VIII. Wertschöpfung in der InfÖk

## Advanced Topics of Information Science

### 1. OSS - Einführung

Prof. Dr. habil. Wolfgang Semar ([wolfgang.semar@htwchur.ch](mailto:wolfgang.semar@htwchur.ch))

Hochschule für Technik und Wirtschaft Chur  
Schweizerisches Institut für Informationswissenschaft

# Agenda - ATIS

---

## I. Open Source

### 1. Theoretischer Teil

#### 1. Grundlagen

#### 2. „Freie“ Software <-> Open Source Software

#### 3. Lizenzproblematik

### 2. Ökonomie

#### 1. Geschäftsmodelle für OS

- Unternehmenssichten (Hersteller, Anwender)
- Nutzersicht
- Entwicklersicht

### 3. Recht und Politik

# Warum beschäftigen wir uns mit OS

---

## ■ Motivation

- Sowohl aus ökonomischer als auch aus informationswissenschaftlicher Sicht sehr interessant
- Umsatz mit *Open-Source-Produkten* auf dem US-Markt (Marktforschungsinstitut IDC, 2009):
  - 2007: 5,6 Milliarden US\$
  - 2008: 7 Milliarden US\$
    - Steigerung um 1,4 Milliarden US\$
  - Für das Jahr 2013 wird eine Steigerung von 8,1 Milliarden US\$ prognostiziert
  - Der Einsatz von „freier Software“ wächst kontinuierlich
- Informationsethik; Informationsgesellschaft; ...

# Agenda: 1. Einführung

---

## I. Open Source

### 4. „Open“ geht weiter!

- Open Content <http://www.opencontent.org/>,  
<http://www.opencontentalliance.org/>
- Open Innovations <http://www.openinnovation.eu/>
- Open Linked Data <http://www.linked-open-data.de/>
- Openstreetmap <http://www.openstreetmap.org/>

# Lernziele: ATIS

---

## 1. Lernziele

1. Grundverständnis der verschiedenen SW-Lizenzen
2. Unterschied erkennen zwischen Free Software und Open Source Software
3. Kenntnis der geschichtlichen Entwicklungen und Zusammenhänge
4. Kennen der möglichen OS-Geschäftsmodelle
5. Kennen der Herausforderungen beim Einsatz von OSS in Unternehmen

Wissen besitzen, um selbständig unternehmerische OSS-Entscheidungen treffen zu können!

## Quellen

---

- Open Source Jahrbücher 2004-2008
  - <http://www.opensourcejahrbuch.de>
- GNU
  - <http://www.gnu.org>
  - <http://www.gnu.de>
- Free Software Foundation
  - <http://www.fsf.org>
- Open Source Initiative
  - <http://www.opensource.org>
- Institut für Rechtsfragen der Freien und Open Source Software
  - <http://www.ifross.de>
- Europäische EDV-Akademie des Rechts

## Quellen

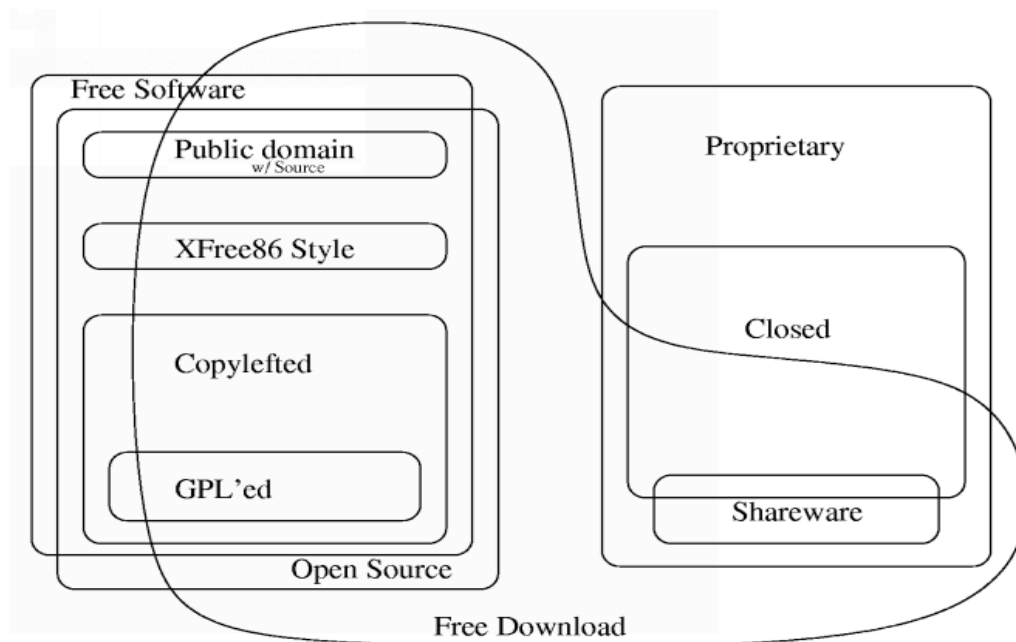
---

- Jens Mundhenke: Wettbewerbswirkungen von Open-Source-Software und offene Standards auf Softwaremärkten. Springer, Berlin et al., 2007
- Thorsten Wichmann: Linux- und Open-Source-Strategien. Springer, Berlin et al., 2005
- Lothar Gläßer: Open Source Software – Projekte, Geschäftsmodelle, Rechtsfragen, Anwendungsszenarien – was IT-Entscheider und Anwender wissen müssen. Publicis Corporate Publisching, Erlangen, 2004

# I. Theoretischer Teil - Grundlagen

- Begriffsdefinitionen - Softwarekategorien
- Unterschiedliche Lizenzen
- Open Source Software
- Open Source <-> Freie Software
- Lizenzmodelle
- Diskussion zur OS-Lizenz-Problematik
- Softwareentwicklung (The Cathedral and the Bazaar)
- Was ist beim Erwerb und Einsatz (Unternehmen, Endanwender) von OSS zu berücksichtigen?

## Verschiedene Softwarekategorien



## Womit alles begann: “Freie Software”

- Richard Stallman (\* 16. März 1953 in Manhattan, New York City; <http://stallman.org/>)
  - Postuliert 1985 “freie Software”
  - Freiheit:
    - „ ... free as in ‚free speech‘, not as in ‚free beer‘.“
    - Freie Software stellt den gesellschaftspolitischen Aspekt in den Vordergrund
  - Warum wurde dies 1985 von Stallman postuliert?
    - In den Jahren 1950-60 war alle Software “quelloffen”
    - Software war kein Wirtschaftsgut, sie wurde für eine spezielle Hardware entwickelt und mit ihr verkauft
    - Leitbild der Entwickler war: “Informationsfreiheit”, Prinzip der dezentralen Machtverteilung.



## Womit alles begann: “Freie Software”

- Entbündelung von Hard- und Software wurde 1969 durch US-Justizministerium (wegen IBM: 70% Marktanteile) eingeleitet.
- In Folge entwickelte sich ein eigenständiger Softwaremarkt (getrieben durch PC-Entwicklung)
  - 1974 Microsoft
  - 1977 Commodore und Apple
  - 1981 IBM PC
    - “IBM kompatibel” keine Bindung an spezifische Hardware mehr
- Zum Beginn der 80er war fast alle Software proprietär, kompiliert und mit Lizenzen ausgestattet.



## Womit alles begann: “Freie Software”

- Grossrechner brauchten ein anderes Betriebssystem
  - Unix war am Beginn 1969 (AT&T Bell Laboratories) ein offenes System.
    - AT&T war es aus kartellrechtlichen Gründen nicht erlaubt ausserhalb der Telekommunikation tätig zu sein.
  - 1984 wurde AT&T aufgespalten - Unix wurde mit sehr hohen Lizenzkosten selbst kommerziell verwertet
  - 1983 startete Stallman das GNU-Projekt
    - Stallman arbeitete bis 1983 am MIT
      - “Paradies für Hacker”

## Womit alles begann: “Freie Software”

- Ziele des GNU-Projekts:
  - Entwicklung eines vollständig freien Betriebssystems, genannt GNU („GNU is *not* Unix“- rekursives Akronym)



- Ein Modell von freier Software, welches die Offenheit der Software und die Möglichkeit des Tauschens gewährleisten soll.
- <http://www.gnu.de/documents/manifesto.de.html>

## Womit alles begann: “Freie Software”

---

- Warum ich (Stallman) GNU schreiben muss:
  - “Ich glaube, daß es das Gebot der Nächstenliebe verlangt, daß ich ein Programm, das mir gefällt, mit anderen teile, denen es ebenfalls gefällt. Software-Anbieter hingegen wollen die Anwender isolieren und beherrschen, wobei sie jeden Anwender dazu verpflichten, nicht mit anderen zu teilen. Ich weigere mich, die Solidarität mit anderen Anwendern in dieser Weise zu brechen. Ich kann nicht mit gutem Gewissen einen Nichtoffenbarungsvertrag oder einen Software-Lizenzvertrag unterzeichnen.”
    - Quelle: <http://www.gnu.de/documents/manifesto.de.html>

## Womit alles begann: “Freie Software”

---

- Warum ich (Stallman) GNU schreiben muss:
  - “Damit ich ehrlich bleiben und trotzdem weiterhin Computer benutzen kann, habe ich mich entschlossen, eine genügend große Sammlung von freier Software zusammenzustellen, so dass ich in der Lage sein werde, ohne jegliche nicht-freie Software auszukommen. Ich habe meinen Beruf im AI lab aufgegeben, um dem MIT keinen rechtlichen Vorwand zu bieten, mich daran zu hindern, GNU weiterzugeben.”
    - Quelle: <http://www.gnu.de/documents/manifesto.de.html>

## Womit alles begann: “Freie Software”

---

- Warum GNU Unix-kompatibel sein wird:
  - “Unix ist nicht mein ideales Betriebssystem, aber es ist nicht übel. Die wesentlichen Eigenschaften von Unix scheinen gut zu sein, und ich denke, daß ich fehlendes ergänzen kann, ohne die guten Eigenschaften zu verderben. Außerdem wird ein Unix-kompatibles System für viele Menschen eher annehmbar sein.”
- Aufgabe: Bitte durchlesen
  - Einige leicht zu entkräftende Einwände gegen die Ziele von GNU
  - <http://www.gnu.de/documents/manifesto.de.html>

## Womit alles begann: “Freie Software”

---

- Warum Unix?
  - Stallman war der Meinung, dass die meisten Unternehmen ein grundlegend neues Betriebssystem ablehnen.
  - Die Architektur von Unix ermöglichte eine schnelle, einfache und verteilte Entwicklung, da Unix aus vielen kleinen Programmen besteht, die größtenteils unabhängig voneinander entwickelt werden können.
  - Viele Bestandteile eines Unix-Systems waren frei für jeden erhältlich und konnten so direkt in GNU integriert werden
  - Fehlende Teile wurden von Grund auf neu geschrieben.

## Womit alles begann: “Freie Software”

- Warum ist es eigentlich “kein” Unix?
  - GNU brachte in den 80er eine Vielzahl an Programm-Paketen hervor, die in bestehende Unix-System eingebettet werden konnten.
  - Diese Komponenten wurden unter der GNU General Public Licence (GPL) veröffentlicht.
  - ABER es war kein komplettes Betriebssystem da noch ein entscheidender Teil fehlte. Welcher?
  - 1994 kam dieser Teil bekannt als “-Kernel hinzu.

## Womit alles begann: “Freie Software”

- Freiheit (Stallman 1985)
  - „... free as in ‚free speech‘, not as in ‚free beer‘.“
  - Viele Sprachen haben zwei verschiedene Worte für "frei" wie in Freiheit und "frei" wie kostenlos.
  - Das Französische zum Beispiel kennt "libre" (frei) und "gratuit" (gratis).
- Freie Software
  - Besitzt eine nicht widerrufbare Freiheit
  - Wenn ein Programm frei ist, kann es möglicherweise in ein freies System wie GNU aufgenommen werden oder in freie Versionen des GNU/Linux-Systems.
    - Problem der „offiziellen Definition“!
  - <http://www.gnu.org/philosophy/free-sw.html>

## Womit alles begann: “Freie Software”

---

- Free Software Foundation
  - 1985 gründete Stallmann die FSF ([www.fsf.org](http://www.fsf.org), [www.fsfe.org](http://www.fsfe.org))
    - logistischer, juristischer und finanzieller Rahmen für das GNU-Projekt
    - Die FSF beschäftigt auch Programmierer, um an GNU zu arbeiten, obwohl der wesentliche Teil der Arbeit von Freiwilligen geleistet wird.

## Womit alles begann: “Freie Software”

---

- Free Software Foundation
  - Free (Freie) Software?
  - Richard Stallman - What is free software?
    - <http://www.youtube.com/watch?v=uJi2rkHiNqg>

## Womit alles begann: “Freie Software”

- Von Richard Stallman stammt die Free Software Definition:
  0. Freiheit zur Nutzung eines Programms zu beliebigen Zwecken
  1. Freiheit zum Studium der Funktionsweise eines Programms und zur Anpassung an die eigenen Bedürfnisse
  2. Freiheit zum Weitervertrieb von Programmen, zur Hilfe des Nächsten.
  3. Freiheit zur Modifikation und Verbesserung von Programmen und zur Veröffentlichung solcher Modifikationen und Verbesserungen, um diese allgemein nutzbar zu machen
- Grundvoraussetzung ist die Verfügbarkeit des Quelltextes!

„Wenn es keinen Quellcode dazu gibt, ist es keine freie Software.“

Quelle: 12/2009; <http://www.gnu.org/philosophy/free-sw.de.html>

- Welche Probleme gibt es in/mit den Freiheiten?

## Freie Software und ihre Freiheiten

- Freiheit 0
  - „Die Freiheit, ein Programm zu benutzen, heißt, dass jede Person oder Organisation das Programm auf jeder Art von Computersystem für jede Tätigkeit benutzen kann, ohne dass verlangt werden kann, sich deshalb mit dem Entwickler oder irgendeiner bestimmten Stelle in Verbindung zu setzen.“

# Freie Software und ihre Freiheiten

---

## ■ Freiheit 1

- Freiheit zum Studium der Funktionsweise eines Programms und zur Anpassung an die eigenen Bedürfnisse.
- Voraussetzung: Der Quellcode ist vorhanden!

# Freie Software und ihre Freiheiten

---

## ■ Freiheit 2

- Die Freiheit, **Kopien** zu verteilen, muss sowohl für binäre – also ausführbare Formen – des Programms als auch für den Quellcode gelten, sowohl für veränderte wie für unveränderte Versionen.

## Freie Software und ihre Freiheiten

---

### ■ Freiheit 3

- **Veränderungen** an der Software können **privat** – bei der eigenen Arbeit oder beim Spiel – verwendet werden, sogar ohne zu erwähnen, dass es diese Änderungen gibt.
- „Wenn man seine Veränderungen veröffentlicht, sollte nicht verlangt werden, jemand bestimmten oder auf eine bestimmte Art zu informieren.“

## Freie Software und ihre Freiheiten

---

- „Der Zugang zum Quellcode ist **eine notwendige** Bedingung für Freie Software.“
- „Damit diese Freiheiten wirklich gelten, müssen sie unwiderruflich sein, so lange man nichts Unrechtes macht.“
  - **Wo ist das Problem?**



## Freie Software und ihre Freiheiten

---

- Jetzt wird's spannend!
  - „**Allerdings** sind bestimmte Regeln über die Art und Weise, wie Freie Software vertrieben werden kann, akzeptabel, sofern sie nicht in Widerspruch zu den grundlegenden Freiheiten stehen.“
- Das führt uns zum:
  - „**Copyleft** (vereinfacht gesagt) ist eine Regel, die verhindert, dass man beim Weiterverteilen des Programmes die grundlegenden Freiheiten anderer beschränkt. Diese Regel steht in keinem Widerspruch zu den grundlegenden Freiheiten, vielmehr schützt sie sie.“

## Copyleft

---

- Software unter Copyleft ist freie Software, deren Verbreitungsbedingungen (**Redistributoren**)
  - keine zusätzlichen Restriktionen hinzufügen lassen,
  - wenn sie diese (Software) weiterverbreiten oder verändern.
- Dies bedeutet, dass jede Kopie dieser Software, sogar wenn sie verändert wurde, **freie Software bleiben muss**.
  - Im GNU-Projekt steht fast alle Software unter Copyleft, da es das GNU-Ziel ist, jedem Benutzer die Freiheiten zu geben, die der Begriff "freie Software" bedingt.

## Copyleft

---

- „Copyleft“ ist ein generelles Konzept.
- Um ein Programm tatsächlich unter Copyleft zu stellen, muss den Verbreitungsbedingungen zugestimmt werden.
  - Wie kann das rechtsgültig gemacht werden?

## Copyleft

---

- Um ein Programm unter das Copyleft zu stellen, wird es zuerst unter das Copyright gestellt.
  - Welches Problem taucht hier auf?
- Danach werden als Rechtsmittel Vertriebsbestimmungen hinzugefügt.
  - Welche müssen das sein?

## Copyleft

---

- Weitere rechtliche Vertriebsbestimmungen:
  - Sie erlauben allen,
    - den Code des Programms oder jedes davon abgeleiteten Programms zu verwenden,
    - zu ändern und
    - weiter zu verteilen,
    - **aber nur** wenn die Vertriebsbestimmungen unverändert bleiben.
  - So werden der Code und die gewährten Freiheiten rechtlich untrennbar.

## Copyleft

---

- Und schon wieder ein Problem!
  - Was ist mit „nicht unter Copyleft stehender freier Software“?

## Nicht unter Copyleft stehende freie Software

---

- Nicht unter Copyleft stehende freie Software kommt vom Autor mit der Erlaubnis,
  - sie weiter zu verbreiten und
  - zu verändern **und**
  - **zusätzliche** Einschränkungen hinzuzufügen.
- Was bedeutet das?
  - Die Probleme hören nicht auf :-)

## Nicht unter Copyleft stehende freie Software

---

- Wenn ein Programm frei ist, aber nicht unter Copyleft steht,
  - **können manche Kopien oder veränderte Versionen überhaupt nicht frei sein!**
- Was bedeutet das?
  - Jetzt wirds langsam spannend, wer denkt noch mit?
  - :-)

## Nicht unter Copyleft stehende freie Software

---

- Wenn ein Programm frei ist, aber nicht unter Copyleft steht:
  - Eine Softwarefirma kann das Programm mit oder ohne Veränderungen kompilieren und die ausführbare Datei als proprietäres Softwareprodukt vertreiben!
  - Und das ist genau das was eigentlich der Idee der “freien Software” entgegen spricht! Oder?

## Nicht unter Copyleft stehende freie Software

---

- Wenn ein Programm frei ist, aber nicht unter Copyleft steht ist dies,
  - die freieste aller Formen für Software!
  - Dies widerspricht zwar dem “Open”-Gedanken, nicht aber dem “RechtsFrei”-Gedanken!
- Was folgt für GNU-Software zwangsläufig daraus?
  - ...

# GNU

---

- Da der Zweck von GNU darin besteht, frei zu sein, muss jede einzelne Komponente im GNU-System freie Software sein.
  - "GNU-Programme" ist gleichbedeutend mit „GNU-Software“.
  - Was ist mit Copyleft und GNU Software?
    - Das führt zu einer “eigenen” Lizenz!!!!!!

# GNU GPL (General Public License)

---

- Die GNU GPL (General Public License, generelle öffentliche Lizenz) ist ein spezifischer Satz an Vertriebsbedingungen, um ein Programm unter Copyleft zu stellen.
  - Für das GNU-Projekt sind die konkreten Vertriebsbestimmungen, die verwendet werden, in der GNU General Public License geregelt.

## GNU GPL (General Public License)

---

- Die GNU-Lizenz wurde 1988 als GNU General Public License (GPL) in einem juristischen Rahmen verankert und in Englisch veröffentlicht.
- Es existieren viele Übersetzungen in viele Sprachen, doch ist z.B. die deutsche Übersetzung nicht rechtsverbindlich.
- Deutsche Version der GNU GPL:  
<http://www.gnu.de/gpl-ger.html>

## GNU GPL (General Public License)

---

- Erfolge auf Rechtsebene
  - “Am 27.7.2010 hat ein New Yorker Gericht der Klage der Entwickler von Busybox stattgegeben und den Elektronikhändler Westinghouse wegen einer Urheberrechtsverletzung zu Unterlassung und Schadensersatz verurteilt. Damit hat zum ersten Mal ein US-amerikanisches Gericht in einem streitigen Verfahren die Gültigkeit der GPL bestätigt, nachdem bisher alle Verfahren durch Vergleiche erledigt worden waren. Auch wenn das Gericht in diesem Versäumnisurteil nicht auf die Begründetheit des Klägervortrags eingeht, kann es wohl doch als Bestätigung der Gültigkeit der Lizenz im US-amerikanischen Rechtssystem verstanden werden.”
  - <http://www.ifross.org/artikel/erste-streitige-durchsetzung-gpl-usa>

## GNU GPL (General Public License)

---

- Erfolge auf Rechtsebene:
  - “Das Landgericht München I hat am 02. April 2004 eine einstweilige Verfügung erlassen, welche die weitere Verbreitung der GPL-Software "netfilter/iptables" entgegen den Bestimmungen der Lizenz untersagt. Diese Entscheidung stellt, soweit ersichtlich, die weltweit erste gerichtliche Durchsetzung der Lizenz vor einem Gericht dar.”
  - <http://www.ifross.org/artikel/erste-gerichtliche-durchsetzung-gpl>
  - Sitecom Deutschland GmbH hat:
    - weder auf die Geltung der Lizenz hingewiesen, noch war eine Kopie der Lizenz beigelegt. Auch fehlte es an einer Offenlegung des Source Codes.

## GNU GPL (General Public License)

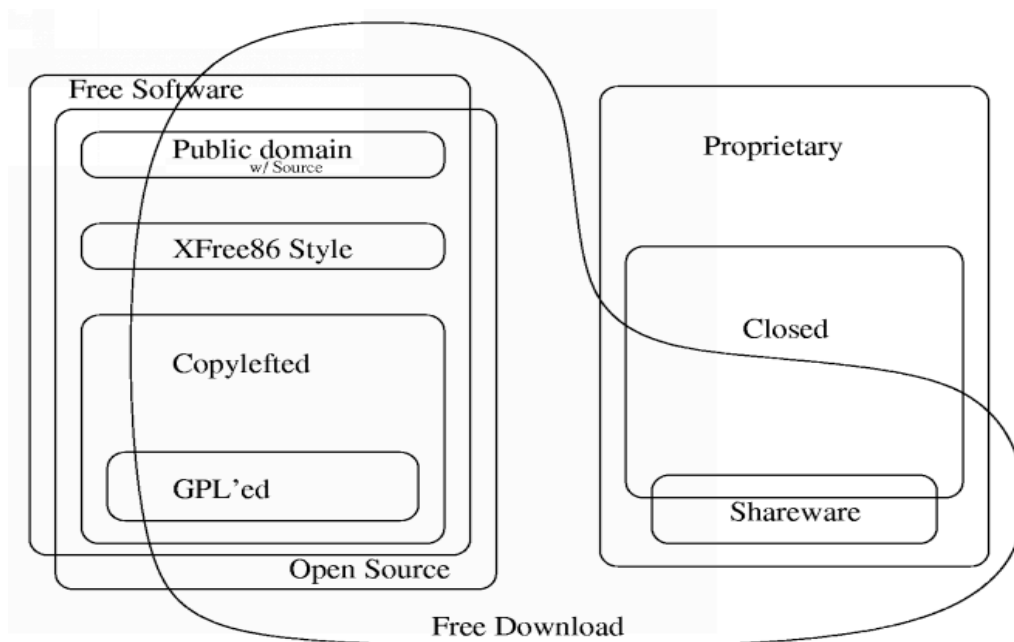
---

- Fazit zur GPL
  - Man kann Geld für eine Kopie von GNU Software bezahlen oder sie kostenlos erhalten haben.
  - Aber egal, wie man die Kopien erhalten hat, man hat immer die Freiheit, die Software:
    - zu kopieren, sie
    - zu verändern, sogar
    - Kopien davon zu verkaufen,
    - aber nicht das Copyleft zu verändern!
    - Quellcode muss kostenlos vorhanden sein!
  - In der GPL kommen die Freiheiten und das Copyleft zusammen!



## Verschiedene Softwarekategorien

## Verschiedene Softwarekategorien



## Public Domain Software

---

- Public Domain Software ist Software, die
  - kein Copyright (Urheberrechtsschutz) besitzt.
  - Das erlaubt es allen, das ursprüngliche Programm und verbesserte Versionen zu verteilen, wenn sie dies wollen.
  - Aber es erlaubt auch unkooperativen Personen, das Programm in proprietäre Software umzuwandeln.
    - Sie können kleinere oder größere Änderungen vornehmen und das Resultat als eigenes Produkt vertreiben.
    - Personen, die das Programm in der geänderten Form erhalten, haben nicht die Freiheit, die die ursprüngliche Autorin ihnen gab; die Person vor ihnen hat sie ihnen genommen.

## Public Domain Software

---

- Mit dem Begriff „Public Domain“ wird auch "frei" oder "gratis verfügbar" in Verbindung gebracht.
- Nichtsdestotrotz ist "Public Domain" ein rechtlicher Begriff und bedeutet genau genommen "nicht urheberrechtsgeschützt".
  - Gibt es auch andere Güter die "Public Domain" sind?

## Halbfreie Software

---

- Halbfreie Software ist Software:
  - die **nicht frei** ist (was auch sonst) ;-)
  - bei der es aber für Individuen die Erlaubnis gibt, sie
    - zu **benutzen**,
    - zu **kopieren** und
    - zu **verändern**
    - inklusive **Verbreitung** veränderter Versionen
  - für **nichtkommerzielle Zwecke**.
- Beispiel für halbfreie Software?
  - PGP

## Proprietäre Software

---

- Proprietäre Software ist Software:
  - die **weder frei**,
  - **noch halbfrei** ist.
    - Ihre **Weiterverbreitung** oder
    - **Veränderung** sind **verboten** oder
    - verlangen von Ihnen, dass Sie eine Erlaubnis dafür benötigen oder
  - sind so stark eingeschränkt, dass Sie sie effektiv nicht
    - frei verändern oder
    - verbreiten dürfen.

## Proprietäre Software

---

- Proprietäre Softwarehersteller benutzen manchmal den Ausdruck "freie Software", um sich auf den Preis zu beziehen.
  - Sie meinen damit, dass eine binäre Kopie ohne Entgelt zu erhalten ist.
  - Oder dass eine Kopie dem Computer, den Sie kaufen, beiliegt.

## Proprietäre Software

---

- Die Free Software Foundation folgt der Regel, dass
  - sie kein proprietäres Programm auf ihren Computern installieren,
  - außer temporär für den speziellen Zweck, einen freien Ersatz für eben dieses Programm zu schreiben.

## Proprietäre Software

---

- Es ist unmöglich, halbfreie Software in ein freies Betriebssystem einzubeziehen,
  - da dies die Vertriebsbedingungen des Systems als Ganzes halbfrei machen würde.
- Was bedeutet das für GNU-Projekte?

## Freeware

---

- Der Begriff "Freeware" hat keine klar anerkannte Definition,
  - wird jedoch gemeinhin für Pakete benutzt, welche die Weiterverbreitung,
  - nicht jedoch die Veränderung erlauben
  - darüber hinaus ist der Quellcode nicht verfügbar.

# Freeware

---

- Diese Pakete sind keine „freie Software“,
  - daher verwendet man **nicht** den Begriff „Freeware“, um sich auf „freie Software“ zu beziehen.
- Achtung, das wird sehr oft verwechselt!!

# Shareware

---

- Shareware ist Software:
  - die mit der **Erlaubnis** ausgestattet ist,
    - sie **weiterzuverbreiten**
  - Allerdings muss derjenige, der eine Kopie längerfristig benutzt, ein **Lizenzentgelt zahlen**.
  - Vorteil von Shareware?
    - Testversion
    - Reduzierter Funktionsumfang

# Shareware

---

- Was ist der Unterschied zu einer “Demoversion”?
  - Trailversion (gilt für Demoversion und zeitlich beschränkte Shareware)
- Frage: Ist Shareware freie oder halbfreie Software?

# Shareware

---

- Shareware ist keine freie Software, noch nicht einmal halbfreie. Dafür gibt es zwei Gründe:
  - Für die meiste Shareware ist der Quelltext nicht verfügbar; daher kann man das Programm überhaupt nicht verändern.
  - Shareware enthält keine Erlaubnis, eine Kopie zu erstellen und zu installieren, noch nicht einmal für Individuen, die sich in nicht profitablen Aktivitäten engagieren.
    - Wird aber dennoch oft gemacht!
    - <http://deutsche-shareware.de/download/20.htm>

## Shared Source

---

- Shared-Source-Initiative
  - 2001 von Microsoft gewährt in ausgewählte Produkte ausgewählte Teilnehmer (akademische und staatliche Institutionen, Softwareentwickler, Hardwareentwickler) Einsicht in den Quellcode seiner Produkte.
  - Weitergehenden Nutzungsrechte oder Modifizierungsrechte oder Weitergabe oder kommerzielle Nutzung von Veränderungen werden streng überwacht.
  - Hier wird „absichtlich“ eine „freie“ Strategie gefahren!
  - Seit 2005 „will“ sich MS an der GPL und BSD orientieren (Microsoft Reference License (Ms-RL)).

## Kommerzielle Software

---

- Kommerzielle Software ist Software
  - die von einer Firma mit dem Ziel entwickelt wird, aus der Benutzung dieser Software Geld zu machen.
- Frage: Ist kommerzielle Software proprietär?



## Kommerzielle Software

---

- Frage: Ist kommerzielle Software proprietär?
  - "Kommerziell" und "proprietär" ist nicht dasselbe!
    - Die meiste kommerzielle Software ist proprietär, aber es gibt auch
    - kommerzielle freie Software, und es gibt
    - nichtkommerzielle unfreie Software.

## Kommerzielle Software

---

- GNU Ada (Compiler, "GNAT"), zum Beispiel, wird immer unter den Bedingungen der GNU GPL vertrieben und jede Kopie ist freie Software:
  - aber ihre Entwickler verkaufen Supportverträge.
  - Wenn ihre Vertreter mit voraussichtlichen Kunden reden, sagen die Kunden manchmal, "Wir würden uns mit einem kommerziellen Compiler sicherer fühlen." Die Vertreter antworten: "GNU Ada ist ein kommerzieller Compiler; zufälligerweise ist er auch freie Software."

## GPL v3: Lizenzbedingungen

---

- Quelle: <http://www.gnu.de/documents/gpl.de.html>
  - 0. Definitionen
  - 1. Quelltext
  - 2. Grundlegende Genehmigungen
  - 3. Schutz von Anwenderrechten vor Umgehungsverbotgesetzen
  - 4. Unveränderte Kopien
  - 5. Übertragung modifizierter Quelltextversionen
  - 6. Übertragung in Nicht-Quelltext-Form
  - 7. Zusätzliche Bedingungen

## GPL v3: Lizenzbedingungen

---

- Quelle: <http://www.gnu.de/documents/gpl.de.html>
  - 8. Kündigung
  - 9. Annahme der Lizenz keine Voraussetzung für den Besitz von Kopien
  - 10. Automatische Lizenzsierung nachgeordneter Anwender
  - 11. Patente
  - 12. Keine Preisgabe der Freiheit Dritter
  - 13. Nutzung zusammen mit der GNU Affero General Public License

## GPL v3: Lizenzbedingungen

- Quelle: <http://www.gnu.de/documents/gpl.de.html>
  - 14. Überarbeitung dieser Lizenz
  - 15. Gewährleistungsausschuss
  - 16. Haftungsbegrenzung
  - 17. Interpretations von §§ 15 und 16
  - Ende der Lizenzbestimmungen

## Wie kann man GPL in eigene Programme anwenden?

- Quelle: <http://www.gnu.de/documents/gpl.de.html>

[eine Zeile mit dem Programmnamen und einer kurzen Beschreibung]

Copyright (C) [Jahr] [Name des Autors]

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 3 der Lizenz oder (nach Ihrer Option) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE IRGEND EINE GARANTIE, sogar ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK. Details finden Sie in der GNU General Public License.

Sie sollten ein Exemplar der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, siehe <<http://www.gnu.org/licenses/>>.

Fügen Sie auch einen kurzen Hinweis hinzu, wie Sie elektronisch und per Brief erreichbar sind.

## Wie kann man GPL in eigene Programme anwenden?

- Quelle: <http://www.gnu.de/documents/gpl.de.html>

Wenn Ihr Programm interaktive Befehle in einem Terminal entgegennimmt, sorgen Sie dafür, dass es nach dem Start einen kurzen Vermerk ausgibt:

[Programm] Copyright (C) [Jahr] [Name des Autors] Für dieses Programm besteht KEINERLEI GARANTIE; geben Sie "show w" für Details ein.

Dies ist freie Software, die Sie unter bestimmten Bedingungen weitergeben dürfen; geben Sie "show c" für Details ein.

Die hypothetischen Kommandos „show w“ und „show c“ sollten die entsprechenden Teile der GNU-GPL anzeigen. Natürlich können die von Ihnen verwendeten Kommandos auch anders lauten; für ein Programm mit graphischer Benutzeroberfläche werden Sie sicherlich eine „About-Box“ verwenden.

Soweit vorhanden, sollten Sie auch Ihren Arbeitgeber (wenn Sie als Programmierer arbeiten) oder Ihre Schule einen Urheberrechteverzicht für das Programm unterschreiben lassen. Für weitere Informationen darüber und wie Sie die GNU GPL anwenden und befolgen, siehe <http://www.gnu.org/licenses/>.

## Wie kann man GPL in eigene Programme anwenden?

- Quelle: <http://www.gnu.de/documents/gpl.de.html>

Diese General Public License gestattet nicht die Einbindung Ihres Programms in proprietäre Programme. Wenn Ihr Programm eine Funktionsbibliothek ist, dann kann es sinnvoller sein, das Linken proprietärer Programme mit dieser Bibliothek zu gestatten. Wenn dies Ihre Absicht ist, sollten Sie die GNU Lesser General Public License anstelle dieser Lizenz verwenden. Lesen Sie aber bitte vorher <http://www.gnu.org/philosophy/why-not-lgpl.html>.

\* \* \*

Copyright-Notiz siehe oben.

51 Franklin Street, Fifth Floor, Boston, MA 02110, USA

Es ist gebührenfrei gestattet, diesen Artikel als Ganzes und unverändert in beliebigen Medien zu kopieren und weiterzugeben, sofern dieser Hinweis erhalten bleiben.



"I'M SORRY, MR. CHRIST. YOU'RE ONLY LICENSED FOR FIVE LOAVES OF BREAD AND TWO FISH."

Verbatim copying and distribution are permitted without royalty in any medium provided this notice and the copyright notice are preserved.

**HTW** Chur

Wolfgang Semar

Seite 73

Quelle: Phil Garcia, Don Berry, 12/2009; <http://www.gnu.org/graphics/jesus-cartoon.html>

**HTW** Chur  
Schweizerisches Institut für  
Informationswissenschaft

**Vielen Dank für Ihre  
Aufmerksamkeit.**

**Wolfgang.Semar@htwchur.ch**

Hochschule für Technik und Wirtschaft Chur  
Schweizerisches Institut für Informationswissenschaft

[www.informationswissenschaft.ch](http://www.informationswissenschaft.ch)

