# Vorlesung Digitale Nachhaltigkeit

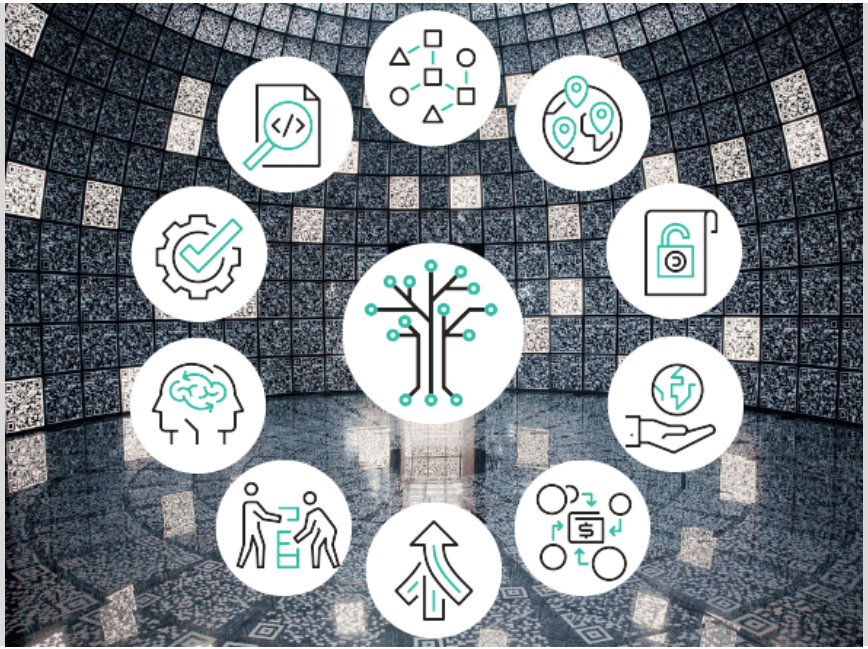
## Termin 7: Urheberrecht und Lizenzen

**28. Oktober 2020**

**Dr. Wolfgang Straub**

**PD Dr. Matthias Stürmer**

Forschungsstelle Digitale Nachhaltigkeit
Institut für Informatik
Universität Bern



UNIVERSITÄT BERN

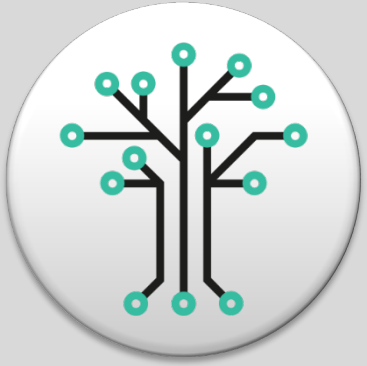
# Termine

| | | |
|---|---|---|
| 1. | **16. September 2020:** | Einführung und Überblick |
| 2. | **23. September 2020:** | Ökologische Nachhaltigkeit und Digitalisierung |
| 3. | **30. September 2020:** | Soziale Nachhaltigkeit und Digitalisierung |
| 4. | **07. Oktober 2020:** | Konzept der digitalen Nachhaltigkeit |
| 5. | **14. Oktober 2020:** | Datenschutz und Privatsphäre |
| 6. | **21. Oktober 2020:** | Ethische Fragestellungen bei KI |
| 7. | **28. Oktober 2020:** | Urheberrecht und Lizenzen |
| 8. | **04. November 2020:** | Open Source Software Development |
| 9. | **11. November 2020:** | Open Source Community Dynamics |
| 10. | **18. November 2020:** | Geschäftsmodelle in der IT-Branche |
| 11. | **25. November 2020:** | Digital nachhaltige Unternehmens-IT |
| 12. | **02. Dezember 2020:** | Digitale Transformation in der Schweiz und international |
| 13. | **09. Dezember 2020:** | Wildcard (Thema noch offen) |
| 14. | **16. Dezember 2020:** | Mündliche Präsentationen |

UNIVERSITÄT BERN

# Agenda

1. **Immaterialgüterrechtlicher Softwareschutz**
2. Lizenzen
3. Was ist Open Source Software?
4. Copyright vs. Copyleft
5. OSS Compliance
6. Erfolgsfaktoren für OSS Projekte



# Immaterialgüterrechtlicher Softwareschutz



**Immaterialgüterrechte**

- **Urheberrecht:** Werke der Literatur und Kunst sowie Computerprogramme
- **Patentrecht:** technische Erfindungen (auch Hard- und Software)
- **Halbleitertopografien** gemäss ToG
- **Designrecht:** grafische und dreidimensionale Gestaltungen (auch GUIs)
- Ergänzender Schutz durch das **Gesetz über den unlauteren Wettbewerb**

# Immaterialgüterrechtlicher Softwareschutz

**Urheberrecht: Schutzvoraussetzungen**
(Art. 2 URG):

- **Werk der Literatur oder der Kunst**
  (auch technische Dokumente) *oder* **Computerprogramm**
- **geistige Schöpfung**
- **individueller Charakter**

# Immaterialgüterrechtlicher Softwareschutz



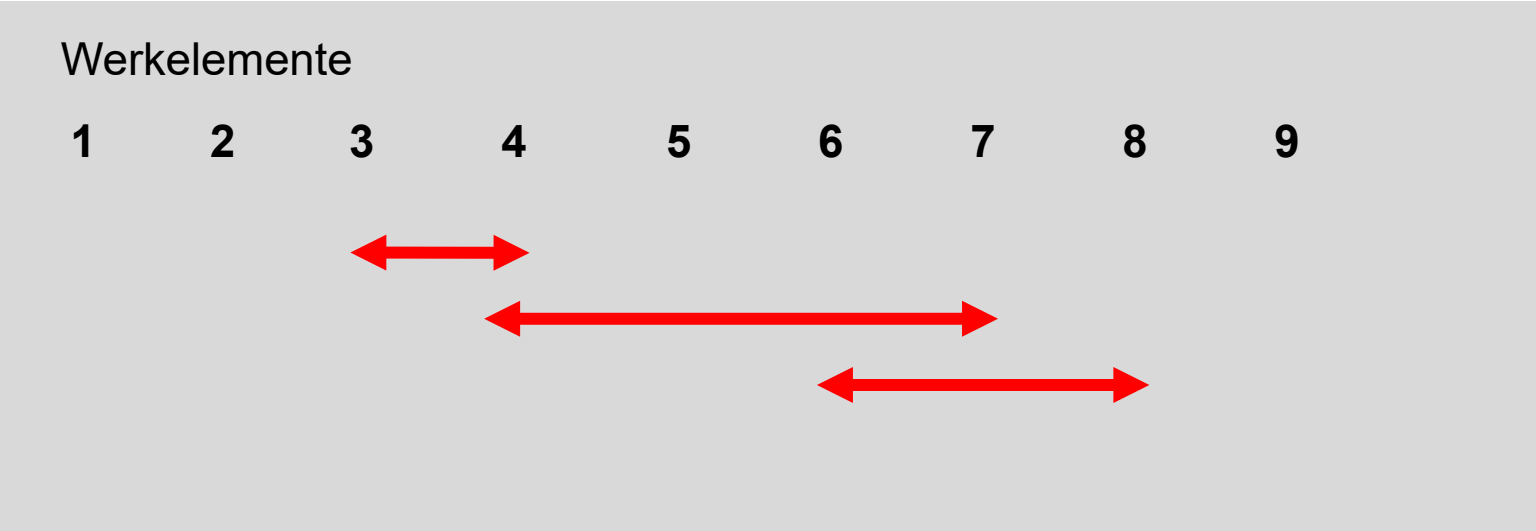
## Urheberrecht

Der **individuelle Charakter** ist gegeben, wenn vor die gleiche Aufgabe gestellte Personen aller Wahrscheinlichkeit nach nicht auf die identische Lösung kommen würden.

Theorie der ‚statistischen Einmaligkeit', Max Kummer 1968

# Immaterialgüterrechtlicher Softwareschutz

**Urheberrecht:** Individualität der einzelnen Sequenzen bestimmt den **Schutzumfang**



# Immaterialgüterrechtlicher Softwareschutz



**Urheberrecht: Schutzebenen**

- **Metastruktur** (z.B. Softwarearchitektur)
  → Problem der Abgrenzung zwischen
  nicht schützbarer **Idee** und geschützter **Ausdrucksform**
- **Text**, Codierung von Software etc.
- grafische **Darstellung**
- …

# Immaterialgüterrechtlicher Softwareschutz

**Urheberrechtlicher Schutz**

– Jeder Kopier- oder Änderungsvorgang erfordert die Zustimmung des Rechtsinhabers.

Ausnahmen:

– **Gebrauchs- und Weiterveräusserungsrechte** (Art. 12, Art. 21 und Art. 24 URG)

– ‚**Freie Benutzung**‘: untergeordnete Bedeutung der verwendeten Elemente im Verhältnis zum neuen Werk

– Zahlreiche weitere gesetzliche Ausnahmen für Sonderfälle

# Immaterialgüterrechtlicher Softwareschutz

**Urheberpersönlichkeitsrechte**

- **Erstveröffentlichungsrecht**
- Recht auf **Erwähnung als** Autor
- Recht, sich gegen **entstellende Werkänderungen** zu wehren

- Diese stehen nur natürlichen Personen zu (Arbeitnehmer des Vertragspartners).
- Sie sind vertraglich nicht übertragbar.
- Ein **Verzicht auf Ausübung** ist aber möglich.

# Immaterialgüterrechtlicher Softwareschutz

UNIVERSITÄT BERN

**Urheberrecht: Besonderheiten des Softwareschutzes**

- Programmnutzung stellt aufgrund von **Kopiervorgängen in den Arbeitsspeicher** eine Vervielfältigung dar.
- Vervielfältigungen und Modifikationen erfordern eine gesetzliche oder vertragliche Lizenz
- **Art. 12 Abs. 2 URG** gibt dem rechtmässigen Erwerber eines Computerprogramms ein zwingendes Gebrauchsrecht
- Wie weit dieser Gebrauch reicht, ist Gegenstand der Auslegung.

# Immaterialgüterrechtlicher Softwareschutz

Die meisten Programme werden in der Form eines **Sourcecodes** in einer höheren Programmiersprache geschrieben.

Beispiel für eine Routine in der Programmiersprache C, welche die Variable c als die Summe der Zahlen *a*=2 und *b*=3 berechnet:

```
int main()
{ int a = 2; int b = 3; int c = a + b; return c; }
```

# Immaterialgüterrechtlicher Softwareschutz

Anschliessend wird der Sourcecode durch einen Compiler in **Objektcode** umgewandelt und in dieser Form ausgeliefert, installiert und ausgeführt.

Beispiel der selben Programmroutine in Objektcodeform:

55 48 89 E5  C7 45 FC 02  C7 45 F8 03  8B 45 F8  8B 55 FC  01 D0  89 45 F4  8B 45 F4  5D C3

# Immaterialgüterrechtlicher Softwareschutz

- Zur Analyse und Weiterentwicklung des Programms ist der **Sourcecode** erforderlich
- Im Sourcecode liegt viel **Know-how der Entwickler**
- Der Sourcecode wird daher bei proprietärer Software meist nicht mitgeliefert
- **Reverse Engineering** zur **Dekompilation von Schnittstelleninformationen** ist unter bestimmten (engen) Voraussetzungen zulässig, aber aufwändig (Art. 21 URG/17 URV)

# Immaterialgüterrechtlicher Softwareschutz

**Reverse Engineering**

(Art. 21 URG und Art. 17 URV und Art. 6 EU ‘Softwareschutz’-RL 91/250 EWG)

- Nur Dekompilation von Schnittstelleninformationen
- Nur zur Schaffung interoperabler Programme
- Hersteller ist nicht bereit, diese zu angemessenen Konditionen zur Verfügung zu stellen
- Soweit nicht überwiegende Interessen des Rechtsinhabers entgegenstehen

- Dekompilation für andere Zwecke (z.B. Sicherheitsanalysen, Fehlerbehebung)?

# Immaterialgüterrechtlicher Softwareschutz



**Schutz von Datenbanken**

- Art. 10 Abs. 2 TRIPs verpflichtet die Mitgliedstaaten zum Schutz von Datenbanken
- EU: ‘Datenbankrichtlinie’ RL 96/9/EG
- CH: Urheberrechtlicher **Schutz als Sammelwerke**, wenn Auswahl oder Anordnung individuellen Charakter hat (Art. 4 URG)
- Kein urheberrechtlicher Schutz, wenn sich Struktur ohne weiteres aus dem Inhalt ergibt (z.B. alphabetisches Telefonbuch)

# Immaterialgüterrechtlicher Softwareschutz

**Patentrecht:** Voraussetzungen (Art. 1 PatG)**:**

- **Erfindung**
- **Gewerbliche Anwendbarkeit**
- **Neuheit (**weltweit!)
- Für eine **Fachperson** auf dem betreffenden Gebiet aufgrund des **aktuellen Standes der Technik nicht naheliegend**
- **Technische Natur**
- Form- und fristgerechtes **Eintragungsgesuch**

# Immaterialgüterrechtlicher Softwareschutz

**Patentrecht:** ‘Technizität’ (Art. 52 Abs. 2 EPUe)**:**

*«Als Erfindungen werden insbesondere nicht angesehen: (…) Programme für Datenverarbeitungsanlagen.(…)»*

Gleiche Funktionalität kann u.U. durch Hard- oder Softwarekomponenten realisiert werden.

# Immaterialgüterrechtlicher Softwareschutz

**Patentrecht:** ‘Technizität’ softwarebezogener Erfindungen

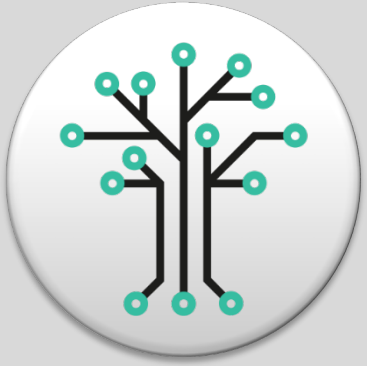
Softwarebezogene Erfindungen können in der Schweiz und der EU patentiert werden, wenn besondere Kriterien erfüllt sind, insbesondere:

- Die Erfindung betrifft die Steuerung der **internen Funktionsweise des Computers** selbst oder seiner Schnittstellen (z.B. Art der Speicherverwaltung durch Betriebssysteme).
- Die Erfindung betrifft die Steuerung **nachgelagerter technischer Komponenten** (z.B. Industrieanlagen, welche physikalische oder chemische Effekte erzielen).
- Die Erfindung betrifft die **Verarbeitung technischer Daten** (z.B. digitale Bilder oder verschlüsselte elektronische Nachrichten)

UNIVERSITÄT BERN

# Agenda

1. Immaterialgüterrechtlicher Softwareschutz
2. **Lizenzen**
3. Was ist Open Source Software?
4. Copyright vs. Copyleft
5. OSS Compliance
6. Erfolgsfaktoren für OSS Projekte



# Lizenzen

**Grundlage**

Verzicht des Lizenzgebers auf die Durchsetzung von Immaterialgüterrechten gegenüber dem Lizenznehmer

**eventuell ergänzt durch Nebenleistungspflichten** des Lizenzgebers (z.B. Schulung, Zurverfügungstellen von Know-how und Material)

UNIVERSITÄT BERN

# Lizenzen

**Erscheinungsformen/Abgrenzungen**

- **Lizenzverträge im engeren Sinn**: Dauerschuldverhältnisse
- ‘**Unechte Lizenzverträge**’: zeitlich unbeschränkte Nutzung gegen Einmalentgelt
- **Gesetzliche Lizenz** beim Erwerb eines Computerprogramms (Art. 12 Abs. 2 URG)
- **Open Source Lizenzen**
- ‘**Gemischte Lizenzen**’ an Immaterialgüterrechten und Know-how

UNIVERSITÄT BERN

# Lizenzen

**‘Konkursfestigkeit’/Durchsetzung gegenüber Rechtsnachfolgern?**

- Umwandlung ‘**echter Lizenzen’** im Insolvenzfall in Geldforderungen
- **’unechte Lizenzen**’ bleiben weiter bestehen (Analogie zum Kaufrecht)

→ Wenn möglich Weiterverwendbarkeit von Drittlizenzen über Vertragsbeendigung hinaus sicherstellen

Nutzung von Software erfordert allerdings oft auch Wartung und Support

UNIVERSITÄT BERN

# Lizenzen

**Lizenzarten**

- Einfache Lizenz
- Exklusivlizenz
- Semiexklusivlizenz
- Übertragbare Lizenz
- Lizenz mit Befugnis zur Unterlizenzierung
- Cross License
- Rücklizenz (sale & license back)
- …

UNIVERSITÄT BERN

# Lizenzen

**Beschränkungen**

- Personelle und örtliche (Named License/Site License)
- Zeitliche (z.B. Jahreslizenz)
- Technische (z.B. CPU/NPU-Klauseln)
- …

**Erweiterungen**

- Mehrere parallele Kopien (z.B. Enterprise License)
- Unterlizenzierung, Outsourcing/SaaS
- Änderung und Weiterentwicklung
- …

UNIVERSITÄT BERN

# Lizenzen

**Gebührenmodelle**

- **Fixe Komponenten**
- **Wiederkehrende Gebühren**, eventuell mit Ober- und Untergrenzen
- **Nutzungsabhängige Gebühren**
  (auch versteckt, z.B. über CPU-Klauseln)
  → wer misst?
- **...**

UNIVERSITÄT BERN

# Lizenzen

**End User Licensing Agreements (EULA)**

- **Erscheinungsformen** (send back, shrink wrap, clickdown etc.)
- Kontroverse um **rechtliche Verbindlichkeit**
- Problematik **Allgemeiner Geschäftsbedingungen**
- Haftungsbumerang?

# Lizenzen

**Umfang von Rechtsübertragungen**

- Urheberrecht als **Bündel von Befugnissen**
  → nicht übertragene Befugnisse bleiben beim Inhaber
- **Zweckübertragungstheorie** als Auslegungsregel
- Bei der **Entwicklung von Individualsoftware** bleiben die Urheberrechte in der Regel bei der Unternehmerin
- Wird ein **Computerprogramm in einem Arbeitsverhältnis** in Erfüllung vertraglicher Pflichten geschaffen, so sind die Arbeitnehmer zwar Urheber, aber der Arbeitgeber ist zur Ausübung der Urhebervermögensrechte berechtigt
  (siehe in Bezug auf Patente und Designs Art. 332 OR)

UNIVERSITÄT BERN

# Lizenzen

**Was insbesondere zu regeln ist:**

- **Rechtsübertragung oder Lizenz**?
- Beschreibung der **Nutzungsbefugnisse**
- **Umfang** von Lizenzen (Exklusivität, zeitliche und örtliche Beschränkungen, Unterlizenzierbarkeit etc.)
- Rechte an **Weiterentwicklungen**
- Art des **Entgelts**
- **Rechtsgewährleistung**

# Lizenzen

**Gemeinsam geschaffene Immaterialgüter**

- **Keine Regelung = gemeinschaftliche Berechtigung**
- Möglichkeit einer Replizierung von Befugnissen?
- ‚Sell & license back‘: Zuweisung an einen Partner und Lizenzerteilung an den anderen
- Kickback bei Drittkommerzialisierung
- Gründung einer gemeinsamen Gesellschaft zur Verwertung?
- Kostenrelevanz der Regelung

# Lizenzen

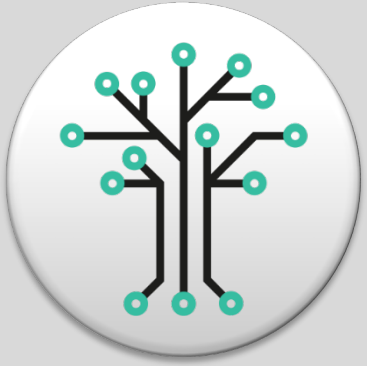
**Rechtsgewährleistung**

- **Es gibt keinen gutgläubigen Erwerb von Urheberrechten!**
- Sicherstellen, dass Vertragspartner selbst über die erforderlichen Rechte (z.B. von Arbeitnehmern) verfügt
- Informationspflicht über potenzielle Ansprüche
- Regeln über Prozessführung
- Eventuell Unzumutbarkeit von Modifikationen

UNIVERSITÄT BERN

# Agenda

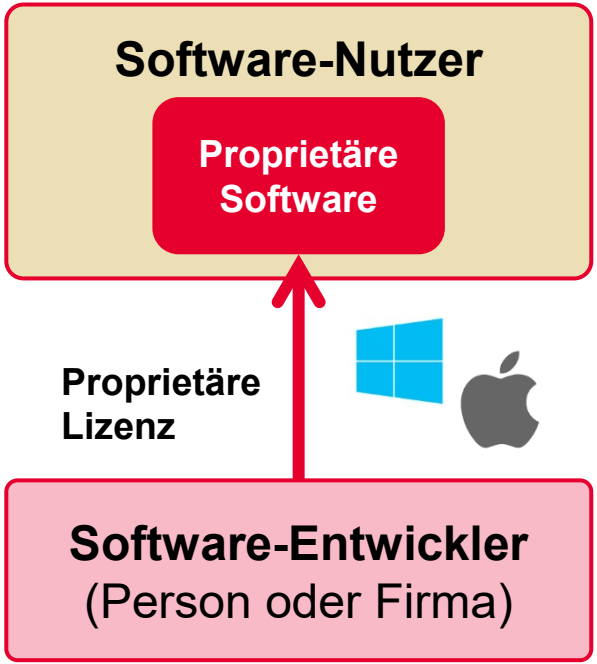
1. Immaterialgüterrechtlicher Softwareschutz
2. Lizenzen
3. **Was ist Open Source Software?**
4. Copyright vs. Copyleft
5. OSS Compliance
6. Erfolgsfaktoren für OSS Projekte



# Was ist Open Source Software?

**Proprietäre Software**

- Anbieter entwickelt Software und **verkauft Nutzungslizenz an Anwender**
- Anbieter behält **Urheberrecht** an der Software
- Anwender kann die Software gemäss **Lizenzbestimmungen** nutzen
- Proprietäre Lizenz gibt vor, dass Software nur von bestimmter **Anzahl Nutzer** verwendet werden kann oder nur für **bestimmte Prozessor-Art** etc.
- **Anwender kennt Software-Code nicht und darf Software nicht kopieren oder ändern**



UNIVERSITÄT BERN

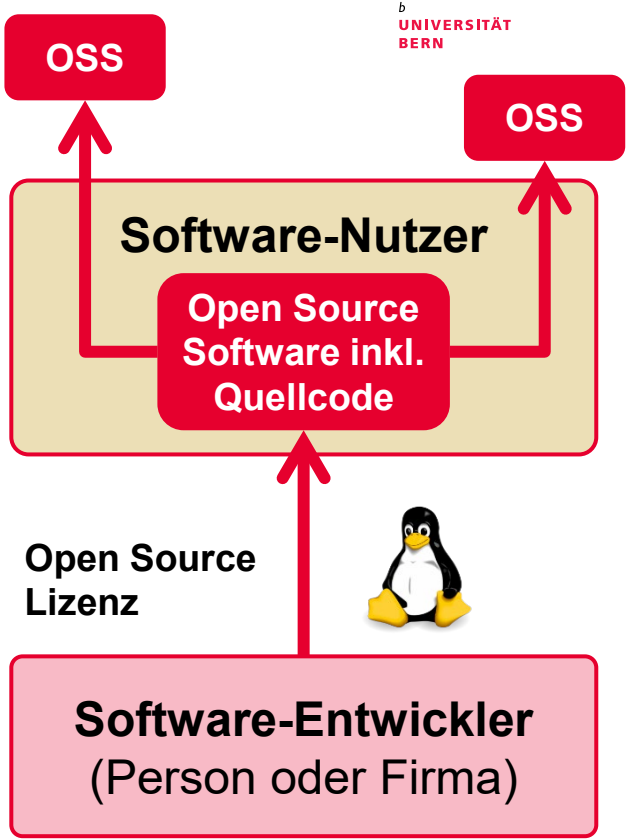
# Was ist Open Source Software?

Zu proprietärer Software sind verschiedene Alternativen entstanden:

- **Public Domain**: Verzicht auf Urheberrechte, eventuell auf Objectcode beschränkt
- **Freeware/Shareware**: kostenlose Vervielfältigung und Weiterverbreitung, keine Modifikation
- **Open Source Software** (OSS): Offenlegung des Sourcecodes, freie Weitergabe und Weiterentwicklung

# Was ist Open Source Software?

Ein Software-Produkt wird als Open Source Software bezeichnet, wenn es unter einer der rund 100 Lizenzen veröffentlicht ist, welche durch die **Open Source Initiative** (OSI) anerkannt sind.






Quelle: https://opensource.org/licenses/alphabetical

UNIVERSITÄT BERN

# Was ist Open Source Software?

**Open Source Definition** der Open Source Initiative (Zusammenfassung):

- **Freie Nutzung** und Weiterverbreitung
- **Offenlegung des Sourcecodes**
- **Recht zu Modifikationen**, wenn diese gekennzeichnet werden
- **Keine Diskriminierung** bestimmter Personengruppen oder Nutzungsarten
- **Weitergabe der Lizenzbedingungen** sowie des **Haftungsausschlusses** bei der Weitergabe

UNIVERSITÄT BERN

# Was ist Open Source Software?

- **Haftungsausschluss** ist sehr wichtig!
- Beispiel **Heartbleed** Bug von 2014: Programmierfehler in Verschlüsselungsprotokoll **OpenSSL**, konkret beim Heartbeat-Befehl
- Kleiner Fehler von **Robin Seggelmann,** Review durch **Stephen Henson**, dennoch blibe Fehler 2 Jahre unentdeckt
- Achtung: Haftungsausschluss für **absichtliche oder grobfahrlässige Verursachung** ist unwirksam, Art. 100 OR

https://www.welt.de/wirtschaft/webwelt/article126814584/Heartbleed-Programmierer-spricht-von-Versehen.html

UNIVERSITÄT BERN

# Was ist Open Source Software?

- **Jede Art von Software** kann unter einer OSS Lizenz veröffentlicht werden, auch solche welche nicht in Sourcecodeform entwickelt wird (z.B. JAVA)
- Auch **andere Werke** (z.B. Dokumentationen) können unter einer OSS publiziert werden, dafür stellen allerdings spezifischere Lizenzen zur Verfügung (z.B. GNU Free Documentation License, Creative Commons License).

UNIVERSITÄT BERN

# Was ist Open Source Software?

- OSS Lizenzen haben den Charakter von **Allgemeinen Geschäftsbedingungen**
  → Die Regeln zur Auslegung von AGBs sind auf sie anwendbar
- OSS Lizenzen dürfen in der Regeln nicht abgeändert werden, können aber **mit zusätzlichen vertraglichen Vereinbarungen verbunden** werden (z.B. Werkverträgen für die Erstellung der Software)

UNIVERSITÄT BERN

# Was ist Open Source Software?

- Das **Nutzungsrecht** an OSS ergibt sich aus Art. 12 Abs. 2 URG
- Das **Recht zur Modifikation** der Software ergibt sich jedoch aus der OSS Lizenz

# Was ist Open Source Software?



**Lizenzverletzungen**

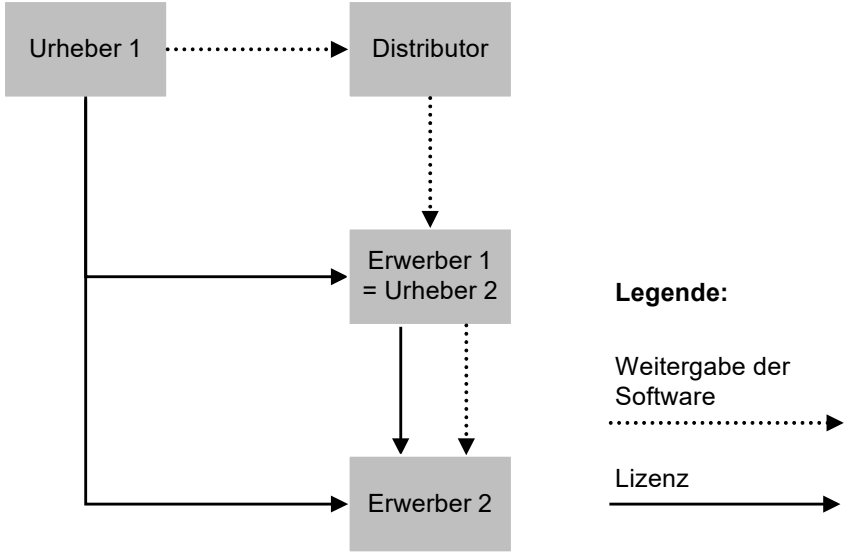
- OSS Lizenzen enthalten oft **auflösende Bedingungen für den Fall einer Lizenzverletzung** → das Modifikationsrecht erlischt
- Lizenzgeber können ihre **Rechtsansprüche** an eine Institution abtreten
- Wenn der Nutzer die Rechtsverletzung beendet hat, kann ein **neuer Lizenzvertrag** abgeschlossen werden. Die Folgen der Rechtsverletzung werden dadurch aber nicht beseitigt →Schadenersatzansprüche

UNIVERSITÄT BERN

# Was ist Open Source Software?

In der Regel **direkte Lizenzverhältnisse zu allen Urhebern**

→ fehlende Rechte in Veräusserungskette unproblematisch



# Open Source Software

**Lizenztexte** sind ebenfalls urheberrechtlich geschützt.

- Oft bestehen **Abänderungsverbote** für Lizenzen.
- Es können jedoch zusätzliche Vereinbarungen zwischen den Parteien getroffen werden.
- Zudem können individuelle Ausnahmen bei der betreffenden Organisation beantragt werden.

UNIVERSITÄT BERN

# Was ist Open Source Software?

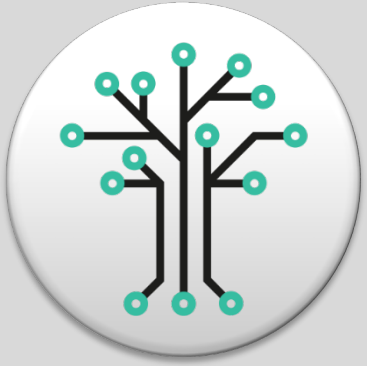
**Modularität:**
OSS verwendet oft modulare Strukturen

- Die Wiederverwendung von fertigen Komponenten **beschleunigt die Softwareentwicklung**
- Da es keinen Preiswettbewerb gibt, entsteht ein **Qualitätswettbewerb** zwischen Open Source Komponenten

UNIVERSITÄT BERN

# Agenda

1. Immaterialgüterrechtlicher Softwareschutz
2. Lizenzen
3. Was ist Open Source Software?
4. **Copyright vs. Copyleft**
5. OSS Compliance
6. Erfolgsfaktoren für OSS Projekte



UNIVERSITÄT BERN

# Copyright vs. Copyleft

Wortspiel: **Copyright vs. Copyleft** enthalten in **GNU GPL**

- **Copyright**: Urheberrechtlich relevante Nutzungshandlungen (Kopieren, Ändern etc.) setzen Zustimmung des Berechtigten voraus.

- **Copyleft**: Modifikationen von OSS müssen unter der gleichen OSS Lizenz weiterverbreitet werden.
  Beispiel: GNU GPL

UNIVERSITÄT BERN

# Copyright vs. Copyleft

**Copyleft**

- Wahrung der Freiheit als grundlegende Idee: **Freie Software bleibt für immer Freie Software**
- **Vorgabe:** Veränderte Software muss unter gleichen Bedingungen freigegeben werden
- **Viraler Effekt:** alle abgeleiteten Werke werden von Copyleft «infiziert»
- Aber **nicht alle Open Source Lizenzen** haben Copyleft-Eigenschaft

UNIVERSITÄT BERN

# Copyright vs. Copyleft

**Kategorien von Open Source Lizenzen:**

1. **Starkes Copyleft:** GNU General Public License (GPL), Affero GPL, SIK GPL
2. **Schwaches Copyleft:** GNU Lesser General Public License (LGPL) und Mozilla Public License MPL
3. **Permissive Lizenzen (kein Copyleft):** Berkley Software Distribution BSD, MIT License, Apache Software License etc.

open source TM



Link: https://opensource.org/licenses/

# Copyright vs. Copyleft

**Kompatibilität von OSS Lizenzen**

- Einzelne OSS Lizenzen sind nicht mit einander kompatibel (insbesondere solche mit und ohne Copyleft-Effekt).
- Beispielsweise enthält **GNU GPLv3** zusätzliche Optionen und Restriktionen **gegenüber GPLv2**.

Dies ist aber nur dann ein Problem, wenn der Lizenzgeber eine bestimmte Version der GPL für seinen Code vorgegeben hat

- Eventuell kann eine **parallele Lizenz** bei allen Rechtsinhaber eingeholt werden.

UNIVERSITÄT BERN

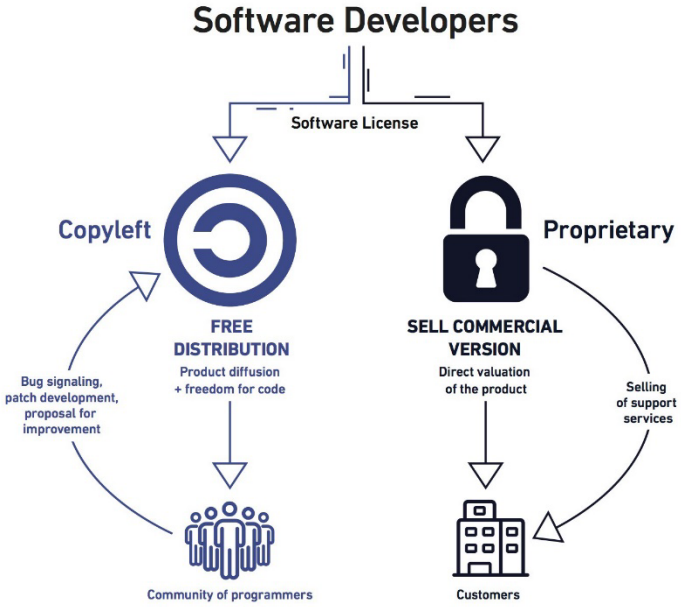
# Copyright vs. Copyleft

## Mehrfachlizenzierung / Dual Licensing

Zwei parallele Lizenzen durch denselben Eigentümer des Urheberrechts:

1. **GPL** für OSS-Anwendungen
2. **Enterprise License** für proprietäre Anwendungen

→ Geschäftsmodell mit Multi Licensing






https://medium.com/deconet/how-to-get-paid-for-open-source-6e13bb238a7f

UNIVERSITÄT BERN

# Copyright vs. Copyleft

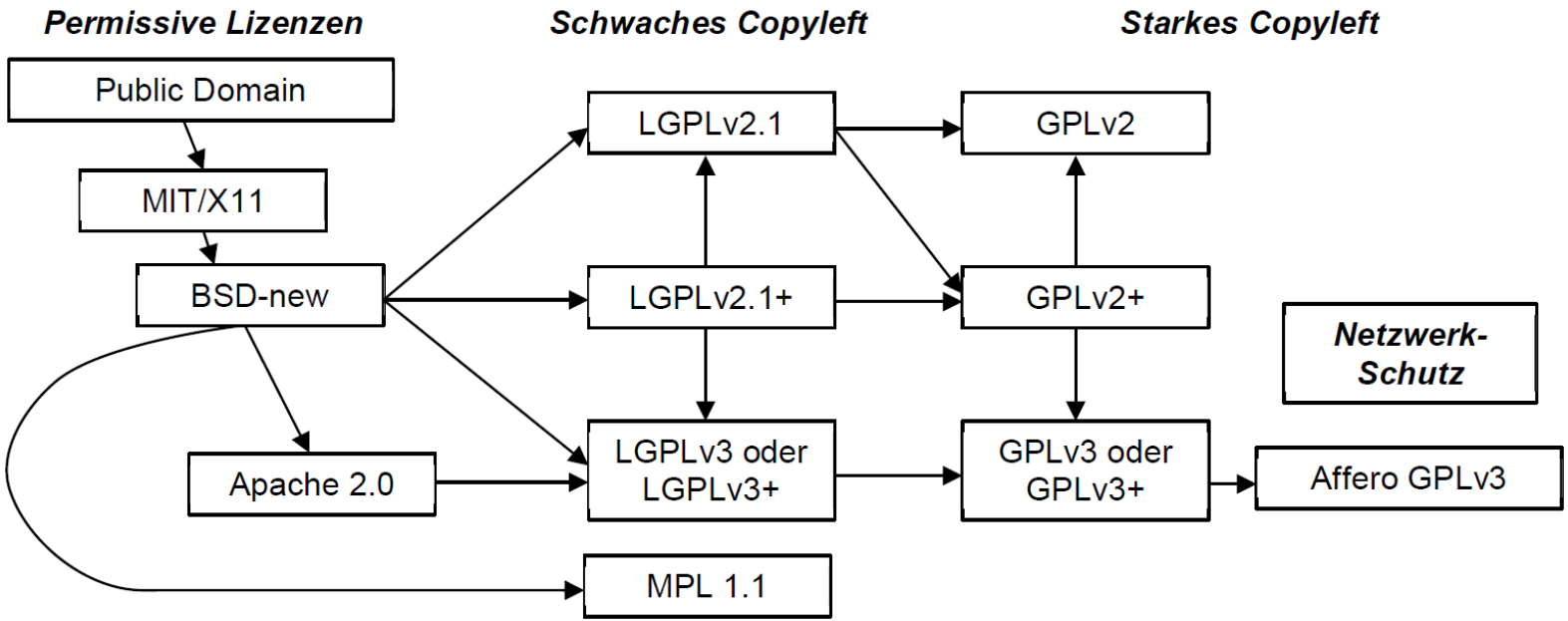
- Die **Free Software Foundation** (*www.fsf.org)* publiziert eine Liste der mit der GNU GPL kompatiblen Lizenzen

- **Lizenzkompatibilitäten** können z.B. bei Black Duck (*www.blackducksoftware.com*), FOSSA *(www.fossa.com) oder* FOSSology *(https://www.fossology.org)* geprüft werden.

UNIVERSITÄT BERN

# Copyright vs. Copyleft

## Kompatibilität von OSS Lizenzen






https://www.isb.admin.ch/isb/de/home/ikt-vorgaben/strategien-teilstrategien/sb004-ikt-teilstrategie-open-source.html

UNIVERSITÄT BERN

# Copyright vs. Copyleft

Der **Copyleft-Effekt** kann in folgenden Konstellationen eintreten:

- OSS wird modifiziert
- Urheberrechtlich geschützte OSS Codeteile werden in neue Programme **übernommen**
- OSS wird mit neuer Software in einer bestimmten Weise **verbunden**
- Die neue Software wird **zusammen mit OSS** in einer bestimmten Weise **verbreitet**

UNIVERSITÄT BERN

# Copyright vs. Copyleft

- **Der Copyleft-Effekt** tritt nur ein, wenn modifizierte Software **verbreitet** wird, d.h. wenn es zu einer Weitergabe ausserhalb der eigenen Organisation kommt.
- Nutzung im Rahmen **einer IT-Outsourcings** gilt grundsätzlich nicht als Weitergabe.

UNIVERSITÄT BERN

# Copyright vs. Copyleft

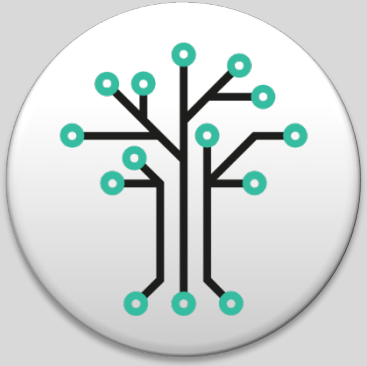
**Copyleft-Vermeidungsstrategien**

- **Parallele Lizenzen** bei allen Rechtsinhabern einholen (Achtung: Programmschöpfer und Rechtsinhaber können auseinanderfallen!). Bei zahlreichen Contributors nicht praktikabel
- **Änderungen** durch **Endbenutzer vornehmen** lassen. Statt eines modifizierten Codes wird eine Anleitung zur Programmänderung geliefert
- **Technische Distanzierungsstrategien**

UNIVERSITÄT BERN

# Agenda

1. Immaterialgüterrechtlicher Softwareschutz
2. Lizenzen
3. Was ist Open Source Software?
4. Copyright vs. Copyleft
5. **OSS Compliance**
6. Erfolgsfaktoren für OSS Projekte



# Open Source Compliance

Let's save the GNU! gpl-violations.org

gpl-violations.org | About | GPL | FAQ | Mailing Lists | News | Helping out | Contact

## gpl-violations.org project brings TomTom to GPL compliance

Harald Welte — 2004-10-24

The gpl-violations.org project (http://www.gpl-violations.org/) announces yet another successful enforcement of the GNU General Public License(GPL), a license commonly used with Free Software, such as the Linux Operating System Kernel. In an amicable agreement with TomTom B.V. (http://www.tomtom.com/), a Dutch vendor of Navigation Systems, the gpl-violations.org project was able to bring the TomTom GO product into GPL compliance.

The source code for the Linux Operating System Kernel, including TomTom's own modifications, is now made available at http://www.tomtom.com/gpl. Furthermore, other obligations of the GPL such as reproduction of the full license terms will be fulfilled by TomTom B.V.

As part of the agreement, TomTom will show it's appreciation of the Free Software and technology enthusiast movement by making a significant donation to the Chaos Computer Club (http://www.ccc.de/).

> "We are happy about the productive and friendly attitude of TomTom B.V. while resolving this licensing issue", says Harald Welte, one of the copyright holders of the Linux Operating System Kernel and founder of the gpl-violations.org project.



https://gpl-violations.org/news/20041024-linux-tomtom/

# Open Source Compliance

**Es gibt keinen gutgläubigen Erwerb von Urheberrechten!**

- Jede nicht durch Art. 12 Abs. 2 URG oder die OSS Lizenz abgedeckte Nutzung ist grundsätzlich eine Urheberrechtsverletzung.
- Auch die unbewusste Nutzung von versteckten OSS Komponenten in einem Softwarepaket kann zu einer Urheberrechtsverletzung führen.

# Open Source Compliance

**Inventarisierung**

- Welche Software wird verwendet?
- Welche Building Instructions, Installationsskripts Compiler etc. werden verwendet?

# Open Source Compliance

**Lizenzmanagement**

- **Welche Lizenzen** sind auf die verwendeten Softwarekomponenten anwendbar?
- Sind diese mit einander **kompatibel**?
- Bei komplexem Code eventuell externe Analyse des Codes (z.B. via Fossa, Black Duck, Fossology, Whitesource, Eclipse)
- Wie können allfällige **Kompatibilitätsprobleme vermieden** werden (z.B. Einholung einer weitergehenden Lizenz bei allen Rechtsinhabern)

# Open Source Compliance



**Release Management**

- Dokumentation aller verwendeter OSS Versionen, eventuell unter Verwendung eines Repository (z.B. eigener Github Account)
- Je nach Lizenz unterschiedliche Verfügbarkeitsdauern aller Versionen

# Open Source Compliance



**Weitergabe von OSS**

- Wird die Software organisationsextern weitergegeben?
- In welcher Form wird **OSS mit eigener Software oder Drittsoftware verbunden** (dynamische oder statische Verlinkung)?
- **Wie wird die Software weitergegeben** (z.B. Distribution von Softwarepaketen)?
- **Weitergabe der Lizenztexte**: Ist für Empfänger klar, welche Komponenten unter der OSS Lizenz stehen?
- Bei welchen Komponenten wurde **ein Copyleft-Effekt** ausgelöst?

# Open Source Compliance

**Empfehlungen**

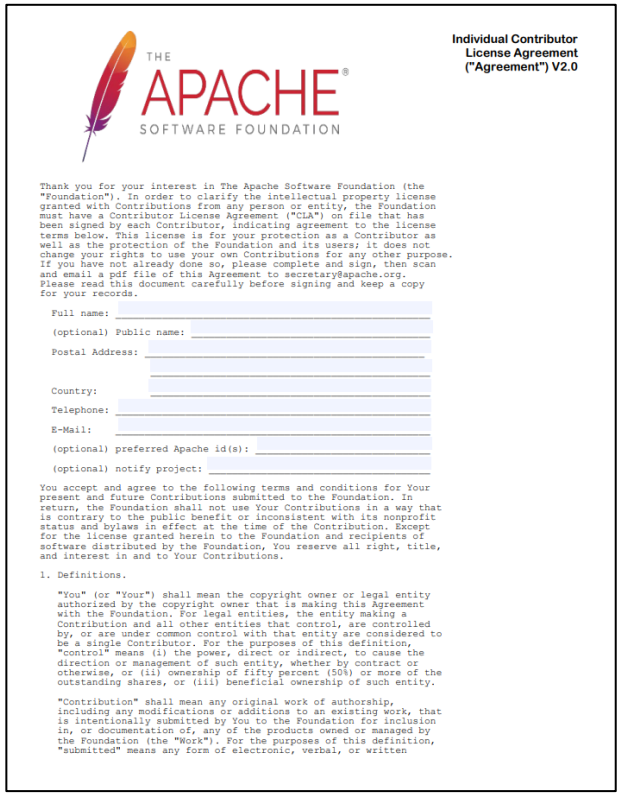
- Korrektes Publizieren der Sourcecodes
- Sicherstellen, dass alle Software Versionen (inkl. Building Instructions etc.) **während der erforderlichen Zeitdauer zur Verfügung** stehen (z.B. Publikation auf GitHub)
- Eigene Anstrengungen zur Compliance **dokumentieren**
- Sich im Fall eines Verletzungsvorwurfs **proaktiv um eine Lösung bemühen**

UNIVERSITÄT BERN

# Open Source Compliance

## Contributor Agreements (CLA)

- Zur Vermeidung von Lizenzstreitigkeiten
- Überträgt Rechte der Code-Contribution an Hauptentwickler
- Hauptentwickler kann bspw. später Änderung der Lizenz bestimmen
- Beispiel: «The Apache Software Foundation Individual Contributor License Agreement»



THE APACHE SOFTWARE FOUNDATION

**Individual Contributor License Agreement ("Agreement") V2.0**

Thank you for your interest in The Apache Software Foundation (the "Foundation"). In order to clarify the intellectual property license granted with Contributions from any person or entity, the Foundation must have a Contributor License Agreement ("CLA") on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of the Foundation and its users; it does not change your rights to use your own Contributions for any other purpose. If you have not already done so, please complete and sign, then scan and email a pdf file of this Agreement to secretary@apache.org. Please read this document carefully before signing and keep a copy for your records.

Full name: _____

(optional) Public name: _____

Postal Address: _____

Country: _____

Telephone: _____

E-Mail: _____

(optional) preferred Apache id(s): _____

(optional) notify project: _____

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to the Foundation. In return, the Foundation shall not use Your Contributions in a way that is contrary to the public benefit or inconsistent with its nonprofit status and bylaws in effect at the time of the Contribution. Except for the license granted herein to the Foundation and recipients of software distributed by the Foundation, You reserve all right, title, and interest in and to Your Contributions.

1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with the Foundation. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to the Foundation for inclusion in, or documentation of, any of the products owned or managed by the Foundation (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written

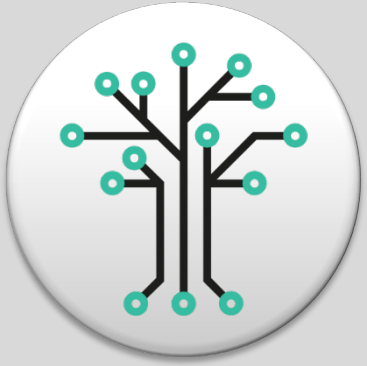


http://oss-watch.ac.uk/resources/cla http://apache.org/licenses/icla.pdf

UNIVERSITÄT BERN

# Agenda

1. Immaterialgüterrechtlicher Softwareschutz
2. Lizenzen
3. Was ist Open Source Software?
4. Copyright vs. Copyleft
5. OSS Compliance
6. **Erfolgsfaktoren für OSS Projekte**



UNIVERSITÄT BERN

# Erfolgsfaktoren für OSS Projekte

Kriterien zur Wahl der geeigneten OSS Lizenz

- **Kompatibilität mit anderen Lizenzen** (insbesondere mit Lizenzen von zu verbindender Software)?
- **Akzeptanz** der Lizenz **bei der Entwickler Community**
- **Aufwand für die Compliance** mit der Lizenz (z.B. Software Repository aller Software Versionen)
- Vorhandensein einer **Institution zur Durchsetzung** der Rechte im Fall von Lizenzverletzungen

UNIVERSITÄT BERN

# Erfolgsfaktoren für OSS Projekte

Wie lässt sich sicherstellen, dass Weiterentwicklungen an OSS allen Nutzern zugute kommen?

- **Verwendung einer Lizenz mit Copyleft**
- **Modularisierung**
- **Repository**
- Eventuell Qualitätskontrolle mit Instanz, welche darüber entscheidet, **welche Commits in den Standard aufgenommen** werden (z.B. Linux)
- Vermeiden von **Unfriendly Forks**

# Erfolgsfaktoren für OSS Projekte

UNIVERSITÄT BERN

## Weiterführende Links

- www.fsf.org
- www.opensource.org
- www.opensource.ch
- www.ossdirectory.com
- www.ch-open.ch
- https://choosealicense.com/licenses/ (Auswahlkriterien für OSS Lizenzen)
- https://timreview.ca/article/416 (gute Übersicht der relevanten OSS Lizenzen)
- https://gpl-violations.org/ (GPL violations Website von Harald Welte)
- https://spdx.dev/ (Standard für OSS Lizenz-Beschreibungen)

### Tools für OSS Compliance

- www.fossa.com
- www.fossology.org
- www.blackducksoftware.com
- www.whitesourcesoftware.com
- projects.eclipse.org/proposals/sw360

- www.it-recht.ch (Wolfgang Straub, Übersetzungen Lizenzen und Teil zu OSS aus dem Buch Softwareschutz)