

## **Frequently Asked Questions (FAQ) zu Rechtsfragen bei der Lizenzierung von Inhalten als Freie Software und Open Content für das Förderprogramm Kultur Digital**

*Dr. Till Jaeger, Fachanwalt für Urheber- und Medienrecht*

**1.**

### **Welche rechtlichen Voraussetzungen gibt es, um ein Programm als Open Source Software oder ein Video bzw. anderes urheberrechtlich geschütztes Werk als Open Content unter Creative Commons lizenzieren zu können?**

Durch eine Creative Commons Lizenz oder eine Open Source Lizenz werden zwar nur einfache Nutzungsrechte eingeräumt, aber dafür sehr umfassend für alle Nutzungsrechte und an jedermann. Daher ist es erforderlich, dass der Lizenzgeber selbst eine Rechtsposition besitzt, die die Einräumung derart umfassender Nutzungsrechte erlaubt.

Dies ist zum einen dann der Fall, wenn der Lizenzgeber selbst der Urheber ist und weder Dritten Nutzungsrechte eingeräumt hat noch Angestellter oder Bediensteter ist. Zum anderen kann der Auftraggeber oder Arbeitgeber ein Werk unter eine Open Source oder Creative Commons Lizenz stellen, wenn er genügende Nutzungsrechte vom Urheber erworben hat. Dies ist bei angestellten Programmierern aufgrund der gesetzlichen Regelung des § 69b) UrhG regelmäßig der Fall:

„Wird ein Computerprogramm von einem Arbeitnehmer in Wahrnehmung seiner Aufgaben oder nach den Anweisungen seines Arbeitgebers geschaffen, so ist ausschließlich der Arbeitgeber zur Ausübung aller vermögensrechtlichen Befugnisse an dem Computerprogramm berechtigt, sofern nichts anderes vereinbart ist.“

Bei allen anderen Kreativen bedarf es einer entsprechenden Regelung im Arbeits- oder Auftragsvertrag. Am besten ist es, wenn dabei die Zulässigkeit einer Lizenzierung unter einer Open Source Lizenz oder Creative Commons Lizenz explizit bestätigt wird.

Wenn mehrere Kreative an einem Werk mitgewirkt haben, müssen diese alle einer freien Lizenzierung zustimmen bzw. diejenigen, denen die Verwertungsrechte übertragen wurden. Dies kann aufwändige Prüfungen erfordern, etwas bei Videos, an denen Musiker, Schauspieler, ein Regisseur, ein Drehbuchautor und ggf. weitere Kreative mitgewirkt haben. Dann ist zumeist ein sog. „Rechte-Clearing“ erforderlich.

## **2.**

### **Wie setzt man eine Open Source Lizenzierung praktisch um?**

Es gibt keine zwingende Form für eine Open Source Lizenzierung, aber es hat sich in der Praxis bewährt,

a) jede Datei, die eine urheberrechtlich schutzfähige Programmierung enthält, mit einem sog. Lizenzheader auszustatten, der einen Urheber- und/oder Copyright-Vermerk enthält [zu dem Unterschied zwischen Urhebervermerk und Copyright-Vermerk siehe unten FAQ Nr. 3] sowie einen Hinweis auf die anwendbare Open Source Lizenz, und

b) eine Datei mit der Bezeichnung „LICENSE“ oder „COPYING“ mit dem Lizenztext der Open Source Lizenz auf der obersten Ebene des Quellcodeverzeichnisses abzulegen.

Die Information in jeder Datei hilft insbesondere dann, wenn Dritte nicht das gesamte Programm nachnutzen möchten, sondern nur Teile davon oder sogar nur einzelne Funktionen.

Zudem setzt es sich mehr und mehr durch, in jede Datei einen sog. SPDX-Identifizier einzufügen, d.h. die standardisierte Kurzbezeichnung von der „SPDX License List“ (<https://spdx.org/licenses/>), um auch ein automatisiertes Auslesen der Lizenzinformationen zu ermöglichen.

Beispiel:

```
# SPDX-FileCopyrightText: [year] [copyright holder] <[email address]>
#
# SPDX-License-Identifier: [identifier]
```

Weitere praktische Beispiele und Hilfestellungen gibt die REUSE-Initiative:  
<https://reuse.software>

### 3.

#### **Was ist der Unterschied zwischen einem Urhebervermerk und einem Copyright-Vermerk?**

Urhebervermerk und Copyright-Vermerk müssen sich im Einzelfall nicht unterscheiden, können es aber. Urheber ist immer der Kreative – also Programmierer, Autor oder Regisseur – der ein konkretes Werk geschaffen hat und dem damit von Gesetzes wegen das Urheberrecht zukommt. Wenn er auch der Inhaber der ausschließlichen Verwertungsrechte ist, unterscheiden sich Urheber- und Copyright-Vermerk nicht.

Wenn der Urheber jedoch angestellter Programmierer ist oder Dritten die ausschließlichen Verwertungsrechte vertraglich eingeräumt hat, dann ist der Arbeitgeber oder Dritte (oftmals der Auftraggeber) als Inhaber des „Copyrights“ anzusehen. Dies beruht auf der internationalen Praxis, entsprechend dem US Copyright-System den Inhaber der ausschließlichen Verwertungsrechte als Copyright Inhaber zu bezeichnen. Denn anders als in Europa ist der Urheber selbst dort bei Auftragswerken („work made for hire“) kein Inhaber von Urheberrechten, sondern nur der Auftraggeber oder Arbeitgeber.

Nach europäischem Urheberrecht bleibt der Urheber stets Inhaber seines Urheberpersönlichkeitsrechts, selbst wenn die ausschließlichen Verwertungsrechte bei einem Dritten liegen. Wenn der Urheber sein Urheberpersönlichkeitsrecht auf Namensnennung wahrnehmen möchte, aber nicht Inhaber der ausschließlichen Verwertungsrechte ist, dann kann man dies wie folgt zum Ausdruck bringen:

```
# SPDX-FileCopyrightText: 2020 Auftraggeberin GmbH, author: Sascha Musterfrau
<sascha.musterfrau@auftraggeberin.com>
#
# SPDX-License-Identifier: Apache-2.0
```

#### 4.

#### **Was ist Lizenzkompatibilität und wie wird diese geprüft?**

Es ist nicht unüblich, dass bei der Programmierung einer neuen Software auch vorbestehende Komponenten oder einzelne Code-Bestandteile übernommen werden. Gerade Open Source Lizenzen erlauben einen solchen „Reuse“. Wenn diese externen Code-Bestandteile andere Open Source Lizenzen als die Neuentwicklung haben, stellt sich die Frage, ob diese Kombination von Programmcode unter verschiedenen Lizenzen zulässig ist.

Während sog. „Permissive Licenses“ [siehe FAQ Nr. 5 und 6] untereinander stets kompatibel sind, weil sie keine Vorgaben für die Lizenz eines neuen oder bearbeiteten Programms machen, so dass die Lizenzpflichten bei verschiedenen Lizenzen einfach kumuliert werden, ist dies bei sog. „Copyleft Lizenzen“ [siehe FAQ Nr. 5 und 6] nicht der Fall. Hier ist Lizenzkompatibilität nur gegeben, wenn eine Copyleft Lizenz eine spezielle Kompatibilitätsklausel enthält, die die Nutzung unter einer anderen Copyleft Lizenz gestattet. Ein Beispiel ist Ziffer 3 LGPL-2.1<sup>1</sup>:

*„You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library.“*

Damit kann Code unter der LGPL-2.1 auch mit Code unter der GPL-2.0 kombiniert und das neugeschaffene Programm insgesamt unter der GPL-2.0 lizenziert werden.

Bei der (häufigen) Kombination von Programmcode unter einer „Permissive License“ und einer „Copyleft Lizenz“ kommt es auf den Einzelfall an, nämlich darauf, ob die Permissive License Lizenzpflichten enthält, die die Copyleft Lizenz nicht vorsieht. Denn Copyleft Lizenzen gestatten es nicht, dass Bearbeitungen (d.h. soweit

---

<sup>1</sup> LGPL-2.1 ist der SPDX-Identifizier für die GNU Lesser General Public License, Version 2.1, d.h. die Abkürzung des SPDX Standards (<https://spdx.org/licenses/>) für diese Lizenz.

das Copyleft reicht) weitergehenden Lizenzpflichten unterstellt werden als dies die Copyleft Lizenz selbst vorsieht. Die Prüfung kann im Einzelfall komplex sein.

Hilfestellung bietet die Lizenzliste der Free Software Foundation (<https://www.gnu.org/licenses/license-list.html#SoftwareLicenses>) und die Lizenzmatrix des OSADL (<https://www.osadl.org/fileadmin/checklists/matrix.html>).

## 5.

### **Was bedeuten "Copyleft" und "ShareAlike"?**

Allen Lizenzen für Open Source Software ist es gemeinsam, dass jedermann unentgeltlich das Nutzungsrecht erwerben kann, die Software in allen Nutzungsarten zu verwenden, insbesondere zu vervielfältigen, weiterzuverbreiten und bearbeiten.

Der wichtigste Unterschied zwischen den verschiedenen Open Source Lizenzen besteht hinsichtlich den Lizenzbedingungen für Weiterentwicklungen und andere Bearbeitungen. Während es dem Lizenznehmer beim Lizenztyp der "Permissive Licenses" freisteht, ob er seine Bearbeitungen der Ursprungssoftware wieder als Open Source Software freigibt oder "proprietär" (z.B. gegen Lizenzgebühren) lizenziert, sehen sog. "Copyleft Lizenzen" die Pflicht vor, Weiterentwicklungen - jedenfalls sofern diese nicht nur intern genutzt werden – wiederum der Ursprunglizenz zu unterstellen und damit ebenfalls an jedermann als Open Source Software zu lizenzieren. Damit soll sichergestellt werden, dass alle, die von Open Source-Software profitieren, ihre Weiterentwicklungen auch der Allgemeinheit zur Verfügung stellen. „Copyleft“ ist also eine Lizenzvertragsklausel, die dazu verpflichtet, Bearbeitungen bei der Weitergabe wieder der Ursprunglizenz zu unterstellen. Open Source bleibt damit Open Source, es entsteht eine „Software-Allmende“. Typisches Beispiel für eine Copyleft-Klausel ist Ziffer 2b) GPL-2.0:

*„You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.“*

Das "Copyleft" kann unterschiedlich weitgehend ausgestaltet sein und entweder alle Bearbeitungen erfassen (dann als "Strenges Copyleft" oder „Strong Copyleft“ bezeichnet) oder die Lizenzierungspflicht nur für bestimmte, definierte Fälle

vorsehen (dann als „Schwaches Copyleft“ oder „Weak Copyleft“ bezeichnet). So verlangt die Lesser General Public License (LGPL) nur, dass Änderungen der der LGPL unterstellten Programmbibliothek<sup>2</sup> selbst wieder der LGPL unterstellt werden. Das auf die Bibliothek zugreifende Programm kann wiederum beliebig lizenziert werden. Bei der GPL mit ihrem strengen Copyleft müssen hingegen auch verlinkte Komponenten wieder unter der GPL (oder einer kompatiblen Lizenz) lizenziert werden.

„ShareAlike“ ist die Übertragung des Gedankens des „Copyleft“ auf andere Werkgruppen als Software. Das Prinzip ist jedoch dasselbe: Bearbeitungen eines Fotos, eines Textes oder einer Komposition müssen wiederum der ursprünglichen Creative Commons Lizenz unterstellt werden (zumeist CC-BY-SA 4.0, <https://creativecommons.org/licenses/by-sa/4.0/>).

## 6.

### **Was sind die Vor- und Nachteile von Copyleft Lizenzen bzw. Permissive Licenses?**

Copyleft Lizenzen haben die folgenden Vorteile:

- Dritte können nicht einseitig von einem freien Inhalt profitieren, ohne ihre Weiterentwicklungen auch der Allgemeinheit zur Verfügung zu stellen. Das Modell sichert urheberrechtlich die Einhaltung der „Spielregeln“ ab.
- Kollaborative Werkerstellung ist einfacher, wenn alle Werkteile unter derselben Lizenz stehen und somit das Gesamtwerk einheitlich genutzt werden kann.
- Es wird das Risiko reduziert, dass eine Open Source Lizenzierung für die öffentliche Hand haushaltsrechtlich oder wettbewerbsrechtlich unzulässig ist.
- Copyleft-Lizenzen können die rechtliche Basis für Entwickler-Gemeinschaften bilden.

Copyleft Lizenzen haben folgende Nachteile:

- Copyleft Lizenzen sind rechtlich komplexer und in der Anwendung damit komplizierter.

---

<sup>2</sup> Programmbibliotheken oder Libraries sind Softwarekomponenten mit Funktionalitäten, die von zahlreichen anderen Programmen benötigt werden und daher nicht immer neu programmiert, sondern als vorbestehende Komponente eingebunden werden (z.B. „neues Fenster in der graphischen Oberfläche öffnen“), vgl. <https://de.wikipedia.org/wiki/Programmbibliothek>.

- Copyleft Lizenzen können Inkompatibilitäten mit anderen Open Source Lizenzen erzeugen (entsprechend für andere freie Inhalte). Da die moderne Programmierung regelmäßig Übernahmen von vorbestehenden Komponenten unter anderen Lizenzen vorsieht, können komplexe Prüfungen der Lizenzkompatibilität erforderlich sein. [siehe dazu FAQ Nr. 4]
- Die Kombination mit herkömmlich lizenzierter „proprietärer Software“ wird erschwert.
- Die Einhaltung der Lizenzpflichten ist für Nachnutzer schwieriger und verlangt eine eingehende Überprüfung.

Permissive Licenses haben folgende Vorteile:

- Sie sind zumeist einfach, kurz und unkompliziert.
- Permissive Licenses sind untereinander stets kompatibel und vielfach auch mit Copyleft Lizenzen.
- Die umfassende Freigabe von Urheberrechten bei gleichzeitiger Pflicht zu Namensnennung ist möglich.

Permissive Licenses haben folgende Nachteile:

- Dritten können „schmarotzen“ und die Software oder anderes Werk verwerten, ohne dass der Urheber davon profitiert.
- Komplexes Nebeneinander von proprietären Lizenzen und Open Source Software kann entstehen, das die Vorteile der Open Source Lizenzierung hinfällig macht.
- Sofern der Lizenzgeber den Vorgaben der Bundeshaushaltsordnung oder einer Landeshaushaltsordnung unterliegt, ist die Vereinbarkeit mit § 63 BHO (oder vergleichbaren Regelungen der Landeshaushaltsordnungen) schwieriger, da das Verschenken von Vermögenswerten des Bund nur im Einzelfall zulässig ist.

**7.**

**Welche Vor- und Nachteile haben die verschiedenen Creative Commons Lizenzen?**

Creative Commons hält Templates für 6 Musterlizenzen vor, mit denen der Lizenzgeber sein Werk unterschiedlich weitgehend lizenzieren kann

(<https://creativecommons.org/share-your-work/licensing-examples/>). Daneben gibt es noch „CC0“, ein Text zum Verzicht auf das Copyright bzw. die weitgehende und lizenzpflichtenfreie Lizenzierung von urheberrechtlichen Nutzungsrechten.

Die Lizenz CC-BY entspricht einer Permissiven License von Open Source Software oder Open Content und CC-BY-SA einer Copyleft-Lizenz [siehe FAQ Nr. 6]. Daneben gibt es noch Lizenzvarianten, die die Bearbeitung und/oder die kommerzielle Nutzung untersagen. Von den CC-Lizenzen, die eine kommerzielle Nutzung untersagen (CC-BY-NC, CC-BY-NC-SA und CC-BY-NC-ND) ist im Regelfall abzuraten, da der Ausschluss der kommerziellen Nutzung wegen der Unsicherheit über die Auslegung dessen, was als „nicht-kommerziell“ angesehen werden kann, nur einen geringen rechtssicheren Nutzungsspielraum belässt (vgl. [https://mirrors.creativecommons.org/defining-noncommercial/Defining\\_Noncommercial\\_fullreport.pdf](https://mirrors.creativecommons.org/defining-noncommercial/Defining_Noncommercial_fullreport.pdf)).

Die Lizenz CC-BY-ND ist dann geeignet, wenn jegliche Veränderung des Werkes verhindert werden soll, weil es in sich abgeschlossen ist und keiner Aktualisierung bedarf.

Bei CC0 müssen keine Lizenzpflichten erfüllt werden und es erfolgt keine Urhebernennung. Die Verwendung von CC0 ist vor allem für Daten (z.B. Metadaten) geeignet, die unter dem Datenbankherstellerrecht geschützt sind, oder wenn kein Interesse an einer Zuordnung zu einem Rechteinhaber oder Urheber besteht. Die Nachnutzung ist dann unkompliziert und die Aggregation von Daten wird beispielsweise nicht durch die Kumulierung von Lizenzbedingungen erschwert.

## **8.**

### **Wer haftet für Open Source Software?**

Bei der Haftung ist zwischen der Haftung für Sach- und Rechtsmängel sowie der Produkthaftung zu unterscheiden.

Für Sachmängel gewährleistet derjenige der die Software verbreitet. Da Open Source Lizenzen sowohl den Verkauf als auch die unentgeltliche Weitergabe zulassen, richtet sich der Umfang der Gewährleistung nach dem Vertragstyp, dem die Weitergabe zugrunde liegt. Bei der unentgeltlichen Weitergabe findet nach herrschender Meinung



eine Schenkung statt, die Haftung nur für arglistiges Verschweigen von Mängeln vorsieht.

Für Rechtsmängel gewährleisten sowohl der Lizenzgeber als auch der Distributor, die dafür einstehen müssen, dass die in einer Open Source Lizenz versprochenen Nutzungsrechte auch eingeräumt werden können. Da die Nutzungsrechte unentgeltlich eingeräumt werden, findet auch hier Schenkungsrecht Anwendung und die Haftung ist auf Arglist beschränkt.

Ob und in welchem Umfang das allgemeine Produkthaftungsrecht auf Open Source Software Anwendung findet, ist noch nicht gerichtlich geklärt. Wenn dies der Fall wäre, läge der Schwerpunkt auf sicherheitsrelevanter Software und Gefährdungen von Leib, Leben und Eigentum. Hier hat derjenige, der die Software in Deutschland in Verkehr bringt, für Vorsatz und grobe Fahrlässigkeit gem. § 521 BGB zu haften, wenn er seinen Produktbeobachtungspflichten nicht nachkommt, also etwa von Gefahren für Leib und Leben Kenntnis erlangt, aber darüber den Nutzerkreis (z.B. auf der Downloadseite) nicht informiert. Fälle dazu sind bislang nicht bekannt.

Die in den Open Source Lizenzen enthaltenen Haftungs- und Gewährleistungsausschlüsse sind überwiegend vor dem Hintergrund des US Rechts gestaltet worden und können für Lizenzgeber aus den USA wirksam sein. Für Lizenzgeber aus Deutschland steht einer Wirksamkeit das AGB-Recht entgegen, so dass die allgemeinen Regelungen des BGB Anwendung finden, wie oben ausgeführt.

## 9.

### **Was ist bei einem eigenen Open Source Projekt rechtlich zu beachten?**

Die Eröffnung eines Accounts auf einer Plattform für Software wie GitHub, GitLab oder Sourceforge ist einfach möglich. Dennoch gibt es einige Aspekte aus rechtlicher Sicht, die zu beachten sind:

- Bei der Namensfindung sind Marken und Namensrechte Dritter zu beachten. Es sollte insbesondere vermieden werden, seinen Namen an bestehende Programme mit ähnlicher Funktionalität anzulehnen.

- Es ist zu überlegen, welche Lizenz als Projektlizenz geeignet ist [siehe FAQ Nr. 6] und die Software entsprechend zu lizenzieren [siehe FAQ Nr. 2], wenn dazu die erforderliche Rechtsmacht besteht [siehe FAQ Nr. 1].
- Die Nutzungsbedingungen des Plattformbetreibers sind zu prüfen und zu beachten.
- Wenn urheberrechtlich geschützte Entwicklungsbeiträge Dritter in das eigene Projekt aufgenommen werden sollen, ist zu prüfen, ob der Dritte die Software auch selbst geschrieben hat und zur Einräumung von Nutzungsrechten befugt ist. Bei angestellten Entwicklern ist dies bei Programmierungen im Rahmen ihrer beruflichen Tätigkeit nur mit Zustimmung des Arbeitgebers der Fall. Bewährt hat sich die Verwendung eines „Developer Certificate of Origin“ (DCO), vgl. <https://developercertificate.org/>.
- Wenn die Software später umlizenziert oder parallel unter verschiedenen Lizenzbedingungen angeboten werden soll, kann sich die Verwendung eines „Contributor License Agreements“ (CLA) empfehlen, vgl. <http://contributoragreements.org/>. Viele Entwickler stehen einem Abschluss eines CLA – anders als einem DCO – kritisch gegenüber.
- Entwicklungsbeiträge aus anderen Open Source Projekten sind daraufhin zu überprüfen, ob die Lizenz kompatibel mit der Projektlizenz ist [siehe FAQ Nr. 4], und die Nutzer sind an zentraler Stelle (z.B. der Datei „LICENSES“) zu informieren, welche Lizenzen einschlägig sind.
- Auf Anfragen oder Hinweise zu rechtlichen Problemen sollte reagiert werden können [siehe FAQ Nr. 8], und das Projekt nicht verwaisen. Wenn eine Pflege nicht möglich ist, sollte es geschlossen und die Software bei „Software Heritage“ hinterlegt werden (<https://www.softwareheritage.org/>).

## 10.

### **Was unterscheidet Open Content und Open Access?**

Anders als für „Open Source Software“ existiert für „Open Content“ keine allgemein anerkannte Definition. Zumeist wird der Begriff weiter verstanden und umfasst auch solche Jedermann-Lizenzen, die wie die Creative Commons Lizenz „CC-BY-NC“ die kommerzielle Nachnutzung verbietet, oder wie „CC-BY-ND“ Bearbeitungen des Ursprungswerks. Immer muss jedoch die nicht-kommerzielle Vervielfältigung und Verbreitung von Open Content kostenlos gestattet sein. Der Begriff „freie Inhalte“ wird

oftmals synonym zu „Open Content“ verwendet, zum Teil wird der Begriff „freie Inhalte“ enger als Entsprechung zu „Freier Software“ verstanden, so dass auch die kommerzielle Nutzung und die Bearbeitung gestattet sein muss.

Der Begriff „Open Access“ wurde für den speziellen Fall des freien Zugangs zu Wissenschaftsveröffentlichungen im Internet geprägt und ist eine Reaktion auf mangelnden Zugang zu Wissen aufgrund steigender Preise von wissenschaftlichen Zeitschriften bei gleichzeitiger Kürzung der Budgets von Bibliotheken. Er verhält sich dabei zu Open Content wie überschneidende Kreise: Open Content ist einerseits weiter, weil hier nicht nur Wissenschaftsveröffentlichungen erfasst werden, sondern alle urheberrechtlich geschützten Werke, andererseits enger, weil für Open Access keine Open Content Lizenzierung erforderlich ist, sondern das kostenlose Angebot im Internet ausreichen kann. Allerdings ist auch hier das Verständnis nicht einheitlich, wie verschiedene Definitionen für „Open Access“ zeigen („Budapester Erklärung“, „Berthesda Statement“ und „Berliner Erklärung über offenen Zugang zu wissenschaftlichem Wissen“). Viele Open Access Journals verwenden allerdings Open Content Lizenzen wie die Creative Commons Lizenzen.

## **11.**

### **Was versteht man unter „Public Domain“?**

Der Begriff „Public Domain“ stammt aus dem anglo-amerikanischen Recht und bezeichnet Werke, die keinem Urheberschutz unterliegen. In Deutschland spricht man von „gemeinfreien Werken“. Dies kann der Fall sein, weil ein Urheberrechtsschutz von vornherein ausgeschlossen ist (z.B. amtliche Werke wie Gesetzestexte), die urheberrechtliche Schutzfrist angelaufen ist (im Regelfall 70 Jahre nach dem Tod des Urhebers) oder wirksam auf den Urheberrechtsschutz verzichtet wurde. Solch ein Rechtsverzicht ist z.B. in den USA für das Copyright zulässig, aber nicht im deutschen oder französischen Urheberrecht, weil dort das Urheberrecht eine unverzichtbare persönlichkeitsrechtliche Komponente besitzt.

Wenn in einem Staat der Urheberrechtsschutz gesetzlich ausgeschlossen ist (z.B. für Werke von staatlichen Stellen), bedeutet dies nicht zwingend, dass die Nutzung dann auch in anderen Staaten ohne Lizenz zulässig ist. Es ist also immer auf die urheberrechtliche Lage in dem Staat abzustellen, in dem das Werk genutzt werden

soll. Soll ein Werk durch Verzicht in die Public Domain gegeben werden (z.B. durch die Erklärung „dedicated to the public domain“), dann kann dies in Ländern, in denen ein Rechtsverzicht nicht möglich ist, als eine permissive Lizenz verstanden werden, die keine besonderen Lizenzpflichten vorsieht. CC0 folgt diesem Konzept ausdrücklich.

## 12.

### **Was ist bei der Nachnutzung von Werken unter Creative Commons Lizenzen zu beachten?**

Bei der Nachnutzung von Creative Commons lizenzierten Werken ist zweierlei zu beachten. Zum einen ist zu prüfen, welche Lizenz Anwendung findet und in welchem Umfang damit Nutzungsrechte eingeräumt werden. Nicht alle Creative Commons Lizenzen erlauben die Bearbeitung und die kommerzielle Nutzung der lizenzierten Werke, einige Lizenzen verlangen, dass bearbeitete Werke nur unter der Ursprungslizenz weitergegeben dürfen („ShareAlike“).

Zum anderen müssen die jeweils anwendbaren Lizenzpflichten befolgt werden, die sich wiederum nach Lizenz und Lizenzversion unterscheiden können und davon abhängen, in welcher Weise die Nachnutzung erfolgt. Um nicht in jedem Nutzungsfall eine aufwändige Einzelprüfung der Lizenzpflichten vornehmen zu müssen, kann der von Wikimedia Deutschland entwickelte „Lizenzhinweisgenerator“ (<https://lizenzhinweisgenerator.de/>) verwendet werden. Auch wenn dieser unmittelbar nur für Fotos aus Wikimedia Commons und der Wikipedia anwendbar ist, können die Ergebnisse auf Werke mit den entsprechenden Creative Commons Lizenzen einfach übertragen werden. Zumeist sind folgende Informationen mit dem nachgenutzten Werk anzugeben:

- Urhebervermerk des Rechteinhabers
- Titel des Werkes (wenn vorhanden)
- Link auf den Lizenztext
- Link auf die Online-Quelle (wenn angegeben)
- Hinweis auf eine Bearbeitung (wenn vorgenommen)