

# Handreichung

zur Entwicklung von  
Open-Source-Software  
in Kommunen

## Impressum

Originalausgabe Dezember 2024

© Fraunhofer IESE

Das Werk ist urheberrechtlich geschützt.  
Sämtliche, auch auszugsweise Verwertung  
bleibt vorbehalten.

## Auftraggeber und Herausgeber



Fraunhofer-Institut für  
Experimentelles Software Engineering IESE  
Fraunhofer-Platz 1  
67663 Kaiserslautern

Telefon: +49 631 6800-0  
E-Mail: [info@iese.fraunhofer.de](mailto:info@iese.fraunhofer.de)  
[www.iese.fraunhofer.de](http://www.iese.fraunhofer.de)

Design und Layout: **Annika Zepp**

## Autorinnen und Autoren

**Christoph Müller**, arbeitet als technischer Mitarbeiter in der Abteilung »Smart City Engineering« am Fraunhofer-Institut für Experimentelles Software Engineering IESE. Sein Fokus liegt auf der Entwicklung und Wartung von Webanwendungen sowie der Bereitstellung und dem Betrieb von Diensten in Cloud-Umgebungen. Spezielle Expertise weißt er beim Thema »Open-Source-Software« auf.

**Anna Schmitt**, arbeitet als Senior Digital Innovation Designer in der Abteilung »Smart City Design« am Fraunhofer-Institut für Experimentelles Software Engineering IESE. Ihre Kernbereiche liegen auf Projekten zur Entwicklung von Geschäfts- und Betreibermodellen für Digitale Ökosysteme in ländlichen sowie urbanen Gebieten, in denen sie mitunter als Projektleiterin fungiert.

**Anne-Marie Kilpert**, leitet die Abteilung »Smart City Design« am Fraunhofer-Institut für Experimentelles Software Engineering IESE in Kaiserslautern. Ihr Tätigkeitsbereich umfasst die kooperative Entwicklung von Softwarelösungen für verschiedene Bereiche der Daseinsvorsorge. Berufserfahrung sammelte sie in der freien Wirtschaft beim Einwerben von Fördermitteln des Landes Rheinland-Pfalz und der EU für Entwicklungsprojekte im ländlichen Raum.

**Thomas Zehler**, arbeitet als Expert Engineer und Projektleiter in der Abteilung »Smart City Design« am Fraunhofer-Institut für Experimentelles Software Engineering IESE in Kaiserslautern. Sein Fokus liegt auf Projekt- und Qualitätsmanagement sowie Prozessverbesserung. In seiner bisherigen Karriere leitete er verschiedene strategische Qualitätsverbesserungsprojekte, war in zahlreichen Prozess-Assessments tätig und übernahm die Projektmanagementunterstützung in einem großen Rüstungsprojekt der Bundeswehr.

# Inhaltsverzeichnis

<b>1. Was ist Open-Source-Software?</b>	<b>1</b>
1.1. Welche OSS-Lizenzmodelle gibt es und welche Implikationen haben diese?	2
1.2. Wie setzen Entwickelnde ein OSS-Projekt auf?	3
1.3. Was ist der Unterschied zwischen Copyright und Copyleft?	4
1.4. Welche Kosten bzgl. OSS entstehen für Entwickelnde?	4
1.5. Was bedeutet Community Building im Zusammenhang mit OSS?	5
1.6. Müssen Entwickelnde ihr eigenes OSS-Projekt moderieren?	5
<b>2. Wie sieht die Zukunft eines OSS-Projekts aus?</b>	<b>6</b>
2.1. Werden Entwickelnde eines OSS-Projekts über Forks informiert?	6
2.2. Können Entwickelnde ihr OSS-Projekt löschen?	6
2.3. Dürfen Entwickelnde ihr OSS-Projekt an jemand anderen übergeben?	6
<b>3. Checkliste: Für welches Lizenzmodell soll ich mich entscheiden?</b>	<b>7</b>
<b>Quellen</b>	<b>10</b>

# Vorwort

Der Einsatz von Open-Source-Software (OSS) gewinnt in der öffentlichen Verwaltung zunehmend an Bedeutung, insbesondere in Kommunen. Durch die offene Natur und die gemeinschaftsbasierte Entwicklung von OSS bietet sich ein enormes Potenzial, um innovative, transparente und flexible Lösungen zu entwickeln und zu nutzen. Der Charakter von OSS ermöglicht es Software aktiv mitzugestalten und anzupassen.

Für Kommunen empfiehlt es sich daher die Grundlagen von OSS zu kennen und zu verstehen, um von deren Möglichkeiten zu profitieren. Denn zu Beginn eines OSS-Projekts stehen wichtige strategische Entscheidungen an: Welches Lizenzmodell wird gewählt? Wie kann eine Community für ein OSS-Projekt aufgebaut und eingebunden werden? Wie wird zukünftig mit dem OSS-Projekt umgegangen?

Diese Handreichung dient als praxisnaher Leitfaden, um Kommunen eine Orientierung in der Welt der OSS zu geben. Sie vermittelt grundlegende Informationen darüber, was OSS ist, welche Grundgedanken dahinterstehen, welche Lizenzmodelle es gibt und wie OSS-Projekte aufgesetzt werden können. Hierzu bietet sie Kommunen eine Checkliste und einen Entscheidungsbaum zur Unterstützung bei der Auswahl eines passenden Lizenzmodells für ihr OSS-Projekt. Darüber hinaus widmet sich die Handreichung der Bedeutung von OSS-Communities und der Moderation des OSS-Projekts. Diese spielen eine Schlüsselrolle bei der Entwicklung, Pflege und Unterstützung von OSS-Projekten. Der Aufbau einer aktiven und engagierten Community kann für den langfristigen Erfolg eines Projekts entscheidend sein. Ziel ist es, den Weg zur Nutzung und Entwicklung von OSS so verständlich und handhabbar wie möglich zu gestalten, um Kommunen zu befähigen, ihre digitalen Transformationen nachhaltig und zukunftsorientiert zu gestalten.

# Glossar

## BUGFIX

Korrektur eines Softwarefehlers (»Bug«)

## DISTRIBUTION

Weg, wie eine Software den Endnutzer erreicht

## FORK

Abspaltung eines OSS-Projekts, das auf einem Stand der ursprünglichen Version aufbaut und eine vom Ursprung abweichende Entwicklung verfolgt. Forks werden i. d. R. von anderen Maintainern geleitet als das ursprüngliche OSS-Projekt.

## MAINTAINER

Personengruppe oder Einrichtung, die die Leitung eines OSS-Projekts übernimmt und federführend steuert

## PROPRIETÄR

Lizenzierungsmodell, bei dem der Quellcode der Öffentlichkeit nicht zugänglich gemacht wird

## PUBLIC CODE

Quellcode, der aus Projekten entstammt, die öffentlich finanziert wurden

## PUBLIC MONEY

Öffentliche Gelder

## QUELLCODE

Software-Quelltext, der in menschenlesbarer Form vorliegt

## QUELLCODE-HOST

Software-Plattform, auf der Quellcode abgelegt, modifiziert und diskutiert werden kann

# Was ist Open-Source-Software?

Der Grundgedanke von Open-Source-Software (OSS) ist eine offene, gemeinschaftsbasierte Entwicklung eines Software-Entwicklungsprojekts. OSS ermöglicht Nutzenden, die Software weiter zu verwenden und eigene Anforderungen umsetzen zu können. Änderungen an der OSS sollen im besten Fall in das Projekt selbst zurückfließen, bspw. in Form von Bugfixes, der Integration von neuen Features oder Ideen, um dieses gemeinschaftlich weiterzuentwickeln. In der Regel besteht die Gemeinschaft eines OSS-Projekts aus dem Projektteam, den Nutzenden und (externen) Freiwilligen, die Änderungen besteuern.

## BEISPIEL

**Eine Open-Source-Software soll mit anderem Farbschema (z. B. Hintergrund) genutzt werden. Nutzer können den Quellcode dieser Software beziehen und das Farbschema nach eigenen Wünschen anpassen und die so modifizierte Software nutzen.**

Die Entwicklung von OSS zeichnet sich insbesondere durch die Wahl der Lizenz sowie durch den Aufbau und die Pflege der Community. Aktuell gibt es eine Vielzahl von OSS-Lizenzen, die im Kern den Grundgedanken von OSS unterstützen, aber dennoch unterschiedliche Intentionen verfolgen. Für die (kommerzielle) Weiterentwicklung können OSS-Lizenzen verschiedene Regelwerke bereitstellen, die Bedingungen für die Bearbeitung und Wiederverwendung des Quellcodes steuern.

Neben dem gemeinschaftlichen Ansatz ist OSS jedoch auch ein Imagethema. Ein offenes Projekt bietet Nutzenden einen hohen Grad an Transparenz und bietet Möglichkeiten zur Mitbestimmung. Bei einer proprietären Software ist alles, was ein Unternehmen macht, Teil der Geschäftsgeheimnisse des Unternehmens. Bei OSS ist es Teil der Gemeinschaft, die Entwicklung und Diskussion wird dabei offen und transparent geführt.

## TIPP

Ein detaillierte Einführung über "Was ist OSS?" findet sich hier: [Open-Source-Leitfaden: Praxisempfehlungen für Open-Source-Software. Version 3.0](#)

OSS-Projekte geben ihren Quellcode frei, sodass Dritte auf diese zugreifen können. Neben der Beteiligung am Projekt selbst können Dritte sich dazu entscheiden, eine modifizierte Variante des Projekts unter Eigenregie getrennt weiterzuentwickeln. Diese modifizierte Variante wird hierbei als Abspaltung (»Fork«) oder Derivat bezeichnet. Ein Gedanke hinter OSS-Lizenzmodellen ist hierbei, den Umgang mit Abspaltungen zu regeln. Diese Regeln beinhalten unter anderem Pflichten für Dritte, die nach der Abspaltung erfüllt werden müssen, z. B. Weiterlizenzierung oder Erhaltung als OSS. Der Umfang dieser Regeln unterscheidet sich bei den OSS-Lizenzmodellen.

## 1.1. Welche OSS-Lizenzmodelle gibt es und welche Implikationen haben diese?

Im Wesentlichen lassen sich OSS-Lizenzen basierend auf ihren Regelungen und Lizenzbestimmungen in vier verschiedene Kategorien einordnen: *Permissiv*, *schwaches Copyleft*, *starkes Copyleft* und *Network Protective*. Generell bieten OSS-Lizenzen Freiheiten bei der Nutzung und Bearbeitung und koppeln diese lediglich an unterschiedliche Pflichten. Im Folgenden werden Eigenschaften verschiedener OSS-Lizenzmodelle beim Einsatz angepasster Versionen durch Dritte erläutert.

OSS-LIZENZEN WERDEN VON VERSCHIEDENEN INITIATIVEN GEPFLEGT UND AKTUALISIERT. DIES SIND UNTER ANDEREM DIE FREE SOFTWARE FOUNDATION (FSF) ODER DIE OPEN SOURCE INITIATIVE (OSI).

### PERMISSIV

Permissive Lizenzen zeichnen sich dadurch aus, dass sie wenige Auflagen bei der (kommerziellen) Weiterverwendung und bei Veränderungen der OSS hinsichtlich des zu verwendenden Lizenztyps bei Modifikationen auferlegen. Damit können veränderte Versionen des OSS-Projekts auch in proprietärer Software eingesetzt werden, ohne den Quellcode zu veröffentlichen.

### SCHWACHES COPYLEFT

Lizenzen mit schwachem Copyleft ermöglichen Entwickelnden ein eigenes, vom Quellcode abgeleitetes OSS-Projekt zu starten, ohne die gleiche Lizenz oder die gleiche Copyleft-Stärke zu verwenden. Lizenzen mit schwachem Copyleft erlauben somit bspw., dass eine modifizierte Variante eines OSS-Projekts unter anderen (offenen) Lizenzen lizenziert werden darf. Weiterverwender des Quellcodes haben daher weniger Restriktionen bzgl. der Weiterverwertbarkeit des OSS-Projekts. So fließen auch Änderungen ggf. nicht zwingend in das Hauptprojekt zurück, da diese nicht unbedingt angegeben werden müssen.

### STARKES COPYLEFT

Bei Lizenzen mit einem strengen Copyleft-Effekt sind Entwickelnde verpflichtet, von der ursprünglichen Software abgeleitete Werke nur unter den Bedingungen der Ursprungslizenz weiterzuverbreiten und diese anzugeben. Bei solchen Lizenzen steht der OSS-Gedanke und dessen Erhaltung der Software als Teil der Gemeinschaft im Vordergrund.

### NETWORK PROTECTIVE

Bei diesem Lizenzmodell handelt es sich um eine Erweiterung des starken Copyleft. Es regelt, dass die webbasierte Bereitstellung von Software (z. B. Hosting) eine Distribution des OSS-Projekts darstellt und somit angegeben und unter der gleichen Lizenz veröffentlicht werden muss.

Generell ist festzuhalten, dass sich die Lizenz und ihr eigenes

Regelungen voneinander unterscheiden, auch wenn die Lizenzen in die gleiche Kategorie der Lizenzmodelle fallen. Unter diesen Bedingungen ist zwingend zu beachten, dass nicht alle Lizenzen miteinander kompatibel sind. Neben den ausführlichen Lizenzverträgen mit allen Details, wie bspw. von der Open Source Ini-

tiative (OSI) aufgelistet<sup>1</sup>, bieten verschiedene Organisationen auch Zusammenfassungen der Lizenzen online an. Einen solchen Überblick erstellt zum Beispiel die Initiative TLDRLegal<sup>2</sup>.

## 1.2. Wie setzen Entwickelnde ein OSS-Projekt auf?

Eine OSS-Entwicklung kann zwei unterschiedliche Startpunkte aufweisen:

Im ersten Fall setzen Entwickelnde ein OSS-Projekt selbst auf, sodass ein neuer Quellcode entsteht, der vorher noch nicht existiert hat. Hierbei müssen sie eine Entscheidung über zu verwendende OSS-Lizenz treffen. Diese Entscheidung hat Implikationen auf die zukünftige Verwertung ihres OSS-Projekts durch Dritte. Unabhängig von der Lizenz müssen sie jedoch immer eine Sache tun, und zwar den Quellcode veröffentlichen.

Im zweiten Fall nutzen Entwickelnde zur Entwicklung ihres OSS-Projekts ein bestehendes OSS-Projekt, um ihren Dienst, ihren Service oder ihr Produkt bereitzustellen. Hierbei entwickeln sie keinen eigenen Quellcode, sondern nutzen existierenden Quellcode, d. h. ein bestehendes OSS-Projekt, und führen „lediglich“ Modifikationen am Quellcode durch. Inwieweit diese Modifikationen veröffentlicht werden müssen, hängt wiederum von der Lizenz des Quellcodes ab.

### PROJEKT ALS OSS ENTWICKELN

In diesem Fall setzen Entwickelnde ihr OSS-Projekt von Anfang an auf einer Quell-Code-Hosting-Plattform, bspw. Open CoDE<sup>3</sup> oder GitHub<sup>4</sup>. Dies setzt voraus, dass sie sich auf eine OSS-Lizenz für ihr OSS-Projekt festgelegt haben und der Entwicklungs-Workflow, z. B. die Entwicklungsumgebung, vorbereitet ist.

### PROJEKT ALS OSS VERÖFFENTLICHEN

Entwickelnde setzen ein OSS-Projekt auf und laden es am Ende der Projektlaufzeit auf eine offene Plattform wie Open CoDE oder GitHub hoch.

Detaillierte Informationen zu dieser Unterscheidung finden sich hier: [Open Source in der Öffentlichen Verwaltung: Wir zeigen geeignete Open-Source-Prozesse für \(Bestands-\)Software](#)



<sup>1</sup> <https://opensource.org/licenses>

<sup>2</sup> <https://tldrlegal.com/browse>

<sup>3</sup> <https://opencode.de/de>: Auf Open CoDE gelten Regeln, die beachtet werden müssen, z. B. sind nur OSS-Lizenzen erlaubt.

<sup>4</sup> <https://github.com/>

### 1.3. Was ist der Unterschied zwischen Copyright und Copyleft?

Copyleft dient der Ermöglichung einer offenen, gemeinschaftsbasierten Entwicklung. Im Gegensatz hierzu steht das Copyright, dessen Funktion darin besteht, Rechte an geistigem Eigentum zu schützen und eine Rechtsgrundlage herzustellen, diese auch durchsetzen zu können. Auch Copyleft-Lizenzen können Regelungen beinhalten, um eine geeignete Attribution bei der Weiterverwendung zu steuern. Sie dienen aber dem grundsätzlichen Zweck der transparenten, freien Entwicklung von Projekten.

Die OSS-Lizenz stellen einen entsprechenden Baukasten bereit, der genutzt werden kann, um OSS in die Richtung zu formulieren, für die sie genutzt werden soll (kommerziell, Marktabdeckung, Teil der Gesellschaft, OSS-Grundgedanke, ...). Dabei hat jede OSS-Lizenz auf gewisse Weise einen Copyright-Anteil. Autorinnen und Autoren treten dabei bestimmte Rechte ab, um die offene Beteiligung zu ermöglichen. Hierzu zählt zum Beispiel der Verzicht auf eine geschlossene Entwicklung von Software.

### 1.4. Welche Kosten bzgl. OSS entstehen für Entwickelnde?

Die Kostentreiber bei OSS hängen von der Ausrichtung ab:

**(1)** Wird OSS eingesetzt, um es etwa in Form von Dienstleistungen Bürgerinnen und Bürger anzubieten, entstehen hier primär Kosten durch die Bereitstellung der Dienstleistung, etwa in Form von Server- und Betreuungskosten. Hier sollte jedoch beachtet werden, dass bei der Nutzung von OSS als zentrale Komponenten für Dienstleistungen auch eine Verantwortung der Kommunen besteht, die OSS zu unterstützen. Die Art und Weise möglicher Unterstützung, in diesem Fall vor allem durch finanzielle Beiträge, hängt dabei vom jeweiligen Projekt ab.

**(2)** Nimmt eine Kommune eine aktive Rolle in einem neuen oder bereits bestehenden OSS-Projekt ein, bspw. durch das Beistuern neuer Funktionen oder durch Bugfixes, so entstehen weitere Kosten durch das eingesetzte Personal. Diese aktive Beteiligung an OSS und den dadurch anfallenden Kosten ist dabei zusätzlich zu etwaigen Kosten durch die Bereitstellungen von auf OSS basierten Dienstleistungen zu kalkulieren.

Es ist wichtig, rechtliche Aspekte bei OSS-Projekten individuell zu prüfen, daher sollte in jedem Fall ein Anwalt hinzugezogen werden.

Sowohl Copyright als auch Copyleft sind rechtliche Konstrukte, die Entwickelnde nutzen können, um ihrem OSS-Projekt einen rechtlichen Basisrahmen zu geben. Im Gegensatz zu den Rechten des Urhebers mit Copyright, bei dem das Kopieren und die Verbreitung eines Werks eingeschränkt oder gar verboten werden kann, ist die Vervielfältigung des Werks bei Copyleft auch nach dessen Bearbeitung zulässig. Es erlaubt dem Urheber des ursprünglichen Werks auf die Art und Weise der Verbreitung des bearbeiteten Werks, sprich dessen Lizenz, Einfluss zu üben.

Eine Kommune kann auch beide Arten der OSS-Nutzung kombinieren. Für die Erhaltung des OSS-Gedankens sollte die Möglichkeit der Beteiligung, egal in welcher Form, bei Nutzung von OSS durch eine Kommune in Betracht gezogen werden. Nur durch die Beteiligung an OSS können die Projekte nachhaltig weiterentwickelt werden. Die freiwillige Beteiligung an OSS-Projekten ist zwar nicht direkt quantifizierbar, ist jedoch für das OSS-Ökosystem von großer Wichtigkeit. Die Fairness besagt: »Ich nutze dein OSS-Projekt, also gebe ich dir etwas zurück. Du hast dafür den Aufwand der Wartung.«

Grundsätzlich gibt es bei OSS kein Preisschild, dennoch aber eine Form von Invest, z. B. einen Server, um das Angebot bereitzustellen, oder die ethische Verantwortung ggü. des OSS-Gedankens. In jedem Falle bedarf es den Einsatz von Ressourcen, die sich in unterschiedlicher Art und Ausprägung niederschlagen können.

## 1.5. Was bedeutet Community Building im Zusammenhang mit OSS?

Die Community besteht bei OSS-Projekten aus mehr als einer reinen Nutzendengemeinschaft, wie sie etwa bei geschlossener Software vorzufinden ist. So gehören zur Community Nutzende aber auch entwickelnde Personen, die ein gemeinsames Interesse an der Weiterentwicklung und Verbesserung der OSS haben. Des Weiteren können auch Stakeholder aus der Politik Teil der Community sein, um am OSS-Projekt mitzuarbeiten.

Ausschlaggebend für eine Community ist die Möglichkeit einer transparenten Diskussion, in der Entscheidungen und Entwicklungen nachvollziehbar gestaltet sind. Die Grundregeln der Zusammenarbeit und die Vision des OSS-Projekts werden hierbei in einem »Code of Conduct« festgehalten. Es empfiehlt sich das OSS-Projekt langfristig mit einer passenden Organisationsform zu verbinden, etwa in Form eines Vereins.

Kommunen sollten von Anfang an Verantwortliche benennen, die die Community zusammenhalten, die Roadmap im Blick behalten und für den Aufwuchs einer Community proaktiv Netzwerk- und Werbemaßnahmen ergreifen und vorantreiben. Wird die Gemeinschaft größer, ist ein Community-Manager eine Option, um die Überlastung einzelner verantwortlicher Teammitglieder zu vermeiden.

### TIPP

Detaillierte Informationen zum Aufbau von Communities finden sich hier: [Das Ein-Mal-Eins zum Aufbau erfolgreicher kommunaler Open-Source-Communities | Smart City Dialog \(smart-city-dialog.de\)](#)

## 1.6. Müssen Entwickelnde ihr eigenes OSS-Projekt moderieren?

Die Moderation von OSS-Projekten spielt eine wichtige Rolle für deren Erfolg. Wenn Entwickelnde ihr OSS-Projekt nicht moderieren, kann es passieren, dass ihr OSS-Projekt keine Bekanntheit in der jeweils relevanten Öffentlichkeit erlangt und dadurch auch nicht genutzt wird. Wer das OSS-Projekt moderiert, entscheidet das Projekt. Bei größeren Projekten gibt es einen Vorstand oder »Head of XY«. Das Projekt entscheidet dabei, wie groß der Vorstand ist, wie sich dieser zusammensetzt oder was die Aufgabengebiete sind. Es gibt wie in einem Unternehmen oder in einer Verwaltung auch Abteilungen, die sich um ein bestimmtes Thema kümmern, bspw. Security, Safety, etc.

# Wie sieht die Zukunft eines OSS-Projekts aus?

## 2.1. Werden Entwickelnde eines OSS-Projekts über Forks informiert?

Es gibt aus Lizenz-Sicht keinen Mechanismus, der automatisch darüber benachrichtigt, wer den entsprechenden Quellcode modifiziert. Entwickelnde erfahren es entweder über Dritte, finden es selbst heraus oder finden es nicht heraus. Eine »OSS-Aufsicht« gibt es nicht.

## 2.2. Können Entwickelnde ihr OSS-Projekt löschen?

Jeder Entwickelnde hat das Recht, das Projekt einzustellen. So kann das eigene Projekt zum Beispiel auf Hosting-Plattformen für Quellcode gelöscht oder archiviert werden. Abspaltungen (»Forks«) bleiben jedoch dabei weiterhin mit Inhalt (Quellcode etc.) erreichbar.

## 2.3. Dürfen Entwickelnde ihr OSS-Projekt an jemand anderen übergeben?

Ja, eine Übergabe an Dritte ist möglich und wird in der Praxis oft durchgeführt. Eine Übergabe wird durch gewisse Qualitätskriterien erleichtert, die das OSS-Projekt erfüllen sollte. Qualitätskriterien sind u. a. ein gewissenhaft geschriebener Code, eine ausführliche Code-Dokumentation sowie eine vorhandene Community. Vor allem das Community Building ist zeitaufwändig und daher meist ein Fulltime-Job, gehört jedoch zu den Aufgaben eines OSS-Projekt-Besitzenden. Im Grunde gibt es drei Übergabemöglichkeiten:

- Entwickelnde finde einen neuen Maintainer bspw. in Form eines Unternehmens oder einer Privatperson.
- Entwickelnde gliedern ihr Projekt unter eine Dachorganisation ein, bspw. wie LibreOffice unter der gemeinnützigen Stiftung »The Document Foundation«.
- Entwickelnde übergeben der Community ihr Projekt, welche die Rolle des Maintainers übernimmt.

Vor der Übergabe müssen die Entwickelnden aufzeigen, welche Lizenzen sie benutzt haben und welche Lizenzen welche Abhängigkeiten haben (Software-Bibliotheken, Komponenten etc.). Dies kann in einem »readme«, in der App oder anderweitig abgelegt werden. Oftmals gibt es auch einen Menüpunkt. Die Form ist hierbei nicht relevant. Es muss jedoch an einer beliebigen Stelle kenntlich gemacht werden.

# Checkliste: Für welches Lizenzmodell soll ich mich entscheiden?

Um sich für eine Lizenz zu entscheiden, müssen sich Entwickelnde zunächst über einige, ihr OSS-Projekt betreffende Aspekte Gedanken machen. Haben sie für jede dieser Fragen eine Antwort gefunden, wissen sie, in welche Richtung sich ihr OSS-Projekt langfristig entwickeln soll und welches Ziel sie mit dem Projekt verfolgen. Dazu müssen folgende Dinge bereits vorm Start der Entwicklung geklärt sein müssen:

1.

## WAS WILL ICH MIT DER OSS LANGFRISTIG ERREICHEN?

- Marktabdeckung
- Kommerzielle Nutzung
- Erfüllung des OSS-Grundgedankens
- Public Money, Public Code

2.

## HABE ICH MIR BEREITS DOMAINS UND EINEN PROJEKT-NAME GESICHERT?

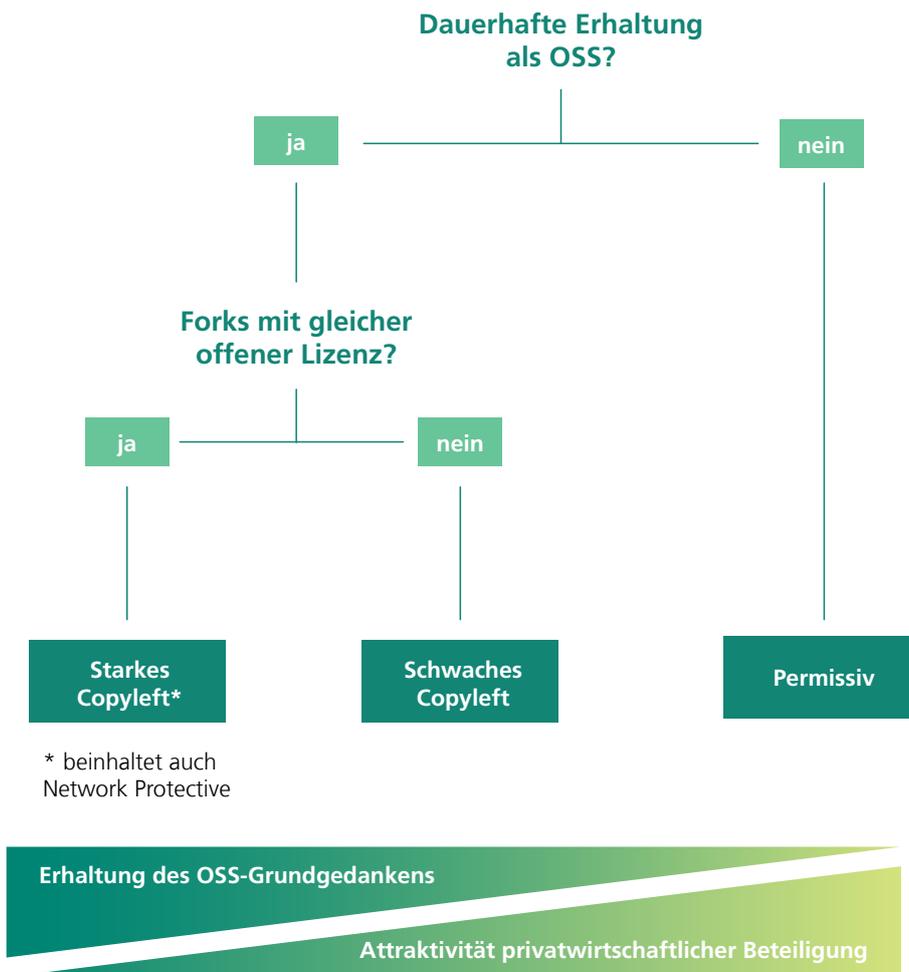
- Webseite für OSS-Projekt zur Dokumentation
- Domain-Sicherung für Webseite bei Domain-Anbieter
- Name für OSS-Projekt für Promotion und Marketing
- Branding für OSS-Projekt zum Aufbau der Projektmarke

3.

## WER SOLL DAS OSS-PROJEKT LANGFRISTIG BETREUEN?

- Unternehmen wird Maintainer
  - Ziel: kommerzielle Nutzung, Marktabdeckung
  - Ansporn: monetärer Art (Gewinn, Umsatz)
  - Lizenzmodell: Schwaches Copyleft, Permissiv
- Kommune wird Maintainer
  - Ziel 1: Public money, public code => Starkes Copyleft
  - Ziel 2: Projekt soll attraktiv für Dritte sein
  - Ansporn: Förderung
  - Lizenzmodell: Schwaches Copyleft, Permissiv
- Community oder Privatpersonen wird Maintainer
  - Ziel: Austausch mit Community, neue Ideen & Feature
  - Ansporn: Persönliches Interesse
  - Lizenzmodell: Starkes Copyleft

Auf Basis der vorgenannten Punkte und unter Zuhilfenahme des Entscheidungsbaums wählen Entwickelnde die Art ihrer Lizenz aus. Die Aspekte der Wirtschaftlichkeit und des OSS-Grundgedankens sowie deren Ausprägung werden im Entscheidungsbaum verdeutlicht.



Das OSS-Projekt kann maßgeblich von der Lizenzwahl beeinflusst werden. Dies bedeutet, dass die Kommune die Natur ihres Projekts kennen muss, um eine passende Lizenzwahl zu treffen. Dazu muss sie das Ziel, das mit dem OSS-Projekt erreicht werden soll, vor Projektstart definieren. Bei der Konzipierung eines OSS-Projekts sollten organisato-

risch-technische Aspekte mit bedacht werden, die während des Projekts und auch nach Projektende auftreten werden, wie bspw. die Finanzierung des nachhaltigen Betriebs und die Weiterentwicklung, Wartung, Pflege und Moderation des OSS-Projekts. Das OSS-Projekt wird sich nach Auslaufen der Förderfinanzierung nicht von selbst aufrechterhalten.

Ziele von OSS: Bei OSS steht die Fairness im Mittelpunkt, garantiert diese jedoch nicht. Bei der Veröffentlichung von OSS-Projekten ist es daher denkbar, dass Dritte von dem Projekt ohne Gegenleistung profitieren. Striktere Lizenzen können ein Mittel sein, die Weiternutzung zu reglementieren, es bleibt jedoch immer bei der freien Weiterverwendung für Dritte ohne Restriktionen. Je freizügiger die Lizenz ist, desto einfacher können auch kommerzielle Akteure das Projekt nutzen.



# Quellen

## OSS-ALLGEMEIN

[Open Source Leitfaden](#)

## LIZENZIERUNGSLEITFRAGEN

[Open CoDE - Lizenzierungsleitfragen](#)

## GOOD PRACTICE

[BWI / BundesMessenger / Info · GitLab \(opencode.de\)](#)

## OSS-COMMUNITY

[Das Ein-Mal-Eins zum Aufbau erfolgreicher kommunaler Open-Source-Communities | Smart City Dialog \(smart-city-dialog.de\)](#)

## OSS UND VERWALTUNG

[Open-Source-Software in Kommunen \(bund.de\)](#)

[Open Source in der Öffentlichen Verwaltung: Wir zeigen geeignete Open-Source-Prozesse für \(Bestands-\)Software - Blog des Fraunhofer IESE](#)

[Regelungen zu Open Source für \*\*Modellprojekte\*\* Smart Cities | Smart City Dialog \(smart-city-dialog.de\)](#)

## ÜBERSICHT ÜBER LIZENZEN

<https://opencode.de/de>

<https://choosealicense.com/>

<https://www.ifross.org/>

Fraunhofer-Institut für  
Experimentelles Software Engineering IESE  
Fraunhofer-Platz 1  
67663 Kaiserslautern

Telefon: **+49 631 6800-0**  
E-Mail: [info@iese.fraunhofer.de](mailto:info@iese.fraunhofer.de)  
[www.iese.fraunhofer.de](http://www.iese.fraunhofer.de)



## KONTAKT

### **Steffen Heß**

Division Manager

Digital Innovation & Smart City

Telefon: **+ 49 631 6800-2275**

E-Mail: [steffen.hess@iese.fraunhofer.de](mailto:steffen.hess@iese.fraunhofer.de)