Softwarelizenzen richtig vergeben (Teil 2)

Qual der Wahl

FOSS-Lizenzen verdeutlichen anderen Anwendern, wie sie unsere Software nutzen können. Dabei hilft es, wenn sich die Lizenz automatisch auslesen lässt. Frank Hofmann, Veit Schiele

Basis dieses Beitrags ist Teil 1 dieser Artikelserie, in dem wir uns mit dem Erkennen und dem Auswählen eines Softwarepakets auf Grundlage der vergebenen Lizenzen auseinandersetzten. Nun nehmen wir die umgekehrte Perspektive ein: Wir sehen uns an, wie wir Lizenzen auswählen und so in unser Softwarepaket einbinden, dass andere dessen Verwendungsmöglichkeiten schnell und einfach erfassen können. Wir beleuchten, welche Vorgehensweisen und Werkzeuge sich im Alltag dafür bewährt haben und wie wir diese in den Entwicklungsprozess eigener Software integrieren.

Die Autoren

Frank Hofmann arbeitet zumeist unterwegs als Entwickler, Trainer und Autor. Er gehört zu den Verfassern des Debian-Paketmanagement-Buchs. **Veit Schiele**, Gründer und Geschäftsführer der Softwareberatung Cusy GmbH, ist Autor des Jupyter- und des PyViz-Tutorials.

Warum lizenzieren?

Ohne die Angabe einer Lizenz wissen Dritte nicht, unter welchen Bedingungen sie Ihr Werk oder Ihre Software überhaupt verwenden können. Zudem bleibt unklar, was Sie beachten müssen, wenn Sie das Werk auf die Bedürfnisse ihres konkreten Anwendungsfalls zuschneiden, es also anpassen und beispielsweise Änderungen am Quellcode vornehmen.

Die Lizenz bringt dabei Ihren Willen als Entwickler zum Ausdruck, regelt eindeutig den Umgang mit dem Werk für alle Seiten und beugt Missverständnissen und Streitigkeiten vor. Voraussetzung ist, dass alle Beteiligten die gewählte(n) Lizenz(en) und die damit verknüpften Regeln kennen und respektieren.

Welche Lizenzen?

Es gibt keine ideale Lizenz, die Sie in jedem Fall verwenden können. Die Wahl hängt von verschiedenen Faktoren ab:

- Welche Lizenzen verwenden abhängige Werke oder (Software-)Pakete?
- Welche Lizenzen bevorzugen andere Teammitglieder?
- Wie einfach und freizügig (permissiv) soll Ihr Werk sein?
- Wie wichtig ist es Ihnen, dass Erweiterungen und Änderungen ebenfalls wieder geteilt werden?

Für die interaktive Auswahl bietet sich der Joinup Licensing Assistant [1] der Europäischen Kommission an, den

Sie per Webbrowser bedienen. Durch Auswahl der jeweiligen Eigenschaften filtern Sie sukzessive die Softwarelizenzen heraus, die alle Ihre Bedürfnisse erfüllen. Die passenden Lizenzen werden am Ende unterhalb der Tabelle sichtbar. Ihre Sortierung erfolgt absteigend nach dem Grad der Übereinstimmung mit den zuvor gesetzten Kriterien. Je genauer Sie Ihre Vorgaben wählen, desto präziser fallen die Ergebnisse in der Liste aus.

Lizenzangaben

Nun ist die Frage zu beantworten, wie und insbesondere wo die Lizenzangaben konkret zu hinterlegen sind. Viele Open-Source-Lizenzen verlangen, dass das Lizenzdokument dem Werk beiliegt. Üblicherweise erfolgt dies in einer separaten Datei namens `LICENSE`. Dienste wie Github und Gitlab erwarten eine solche Lizenzdatei auf oberster Ebene 2.

In dieser Datei stehen der Name der Lizenz sowie alle Lizenzbedingungen. Damit aus der `LICENSE`-Datei die Lizenz automatisch ausgelesen werden kann, muss sie die passenden Formulierungen oder Schlüsselworte enthalten. Github verwendet dafür eine eigene Bibliothek, auf die Sie mittels einer Rest-API zugreifen. Weitere Quellen sind beispielsweise Spdx.org, Choose A License und Gnu.org. Erzeugen Sie via Github ein neues Repository für die Versionskontrolle für das Projekt, geben Sie die Lizenz gleich mit an 3 – wohlgemerkt nur eine einzige Lizenz, da Github Mehrfachlizenzierung (bislang) nicht unterstützt.

Zusatzinfos

Komplexer wird es bei einer Mehrfachlizenzierung für unterschiedliche Nutzungszwecke oder auch, wenn die einzelnen Bestandteile des Gesamtprojekts unter verschiedenen Lizenzen stehen. Nicht alle Werkzeuge zur Paketverwaltung und Auswertung kommen damit zurecht, wenn Sie mehrere `LICENSE`-Dateien beilegen oder mehrere Lizenzen in einer einzigen Lizenzdatei ausliefern. Diesbezüglich hat sich zwar mittlerweile einiges getan, insgesamt herrscht jedoch immer noch ziemlicher Wirrwarr.

Sowohl das DEB- als auch das RPM-Format unterstützen mehrere Lizenzen pro Paket. Für DEBs, die sogar mehrere Lizenzangaben für jede im Paket enthaltene Datei erlauben, beschreibt das Debian Packaging Manual das Vorgehen. In Listing 1 sind über Wildcards alle Dateien mit Pfadangabe und jeweiliger Lizenz aufgeführt. Listing 2 zeigt, wie sich mehrere Lizenzen für eine Datei angeben

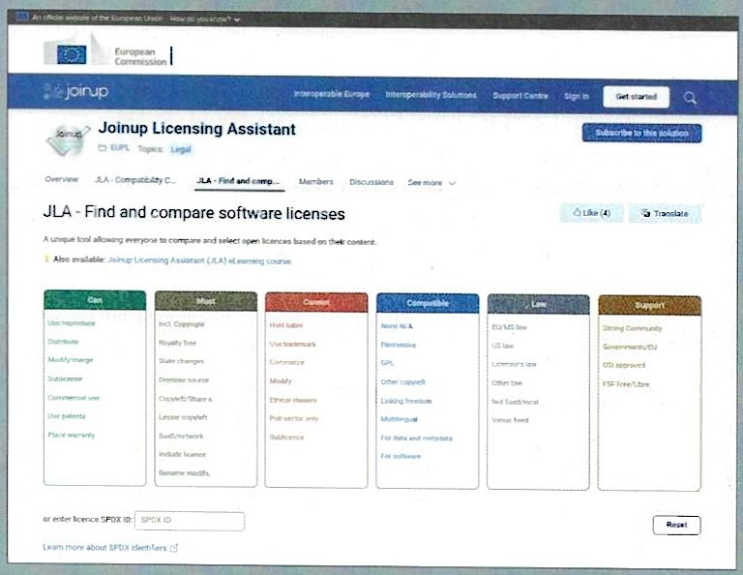
Listing 1: Lizenzangabe pro Datei

```
Files:
 *
Copyright: 1975-2010 Ulla Upstream
License: GPL-2+

Files:
 debian/*
Copyright: 2010 Daniela Debianizer
License: GPL-2+

Files:
 debian/patches/fancy-feature
Copyright: 2010 Daniela Debianizer
License: GPL-3+

Files:
 */*.1
Copyright: 2010 Manuela Manpager
License: GPL-2+
```



1 Der Joinup Licensing Assistant der Europäischen Kommission hilft beim Aussuchen einer passenden Lizenz.

2 So sieht eine automatisch generierte Lizenzangabe bei Github aus.

lassen. Im Beispiel stehen die Javascript-Dateien sowohl unter der Mozilla-Lizenz (MPL) als auch unter der GPLv2 bereit. Im Abschnitt `License:` hinterlegen Sie dann den vollständigen Lizenztext.

Für RPM-Pakete beschreiben der RPM Packaging Guide sowie die Fedora Licensing Guidelines, wie Sie die Lizenz angeben müssen. Hier kommt das SPDX-Format zum Einsatz. Eine fein abgestufte Lizenzangabe pro Datei im Paket wie in Listing 1 klappt damit allerdings nicht. So bleibt Ihnen als Paket-Maintainer nur der Ausweg, die Software in mehrere einzelne Pakete zu zerlegen, um die Lizenz genau angeben zu können.

Listing 2: Mehrere Lizenzen pro Datei

```
Files: src/js/editline/*
Copyright: 1993, John Doe
           1993, Joe Average
License: MPL-1.1 or GPL-2

License: MPL-1.1
 Vollständiger Lizenztext

License: GPL-2
 Vollständiger Lizenztext
```

Etwas einfacher wird es, wenn der komplette Quellcode eines Pakets dual lizenziert ist. In der SPEC-Datei legen Sie fest, aus welchen Komponenten sich das RPM-Paket zusammensetzt. Listing 3 enthält einen Vorschlag zur Angabe der Lizenz in der SPEC-Datei des RPM-Pakets. Laut RPM-Spezifikation müssen sich die beiden Lizenztexte in separaten Dateien befinden statt direkt in der SPEC-Datei.

Oft gibt es in Paketen zudem sprachspezifische Dateien mit Metainformationen. Für moderne Python-Pakete verwendet man ein File namens `pyproject.toml`. Als Lizenzangabe dient entweder der Name der Lizenz, der Verweis auf eine Lizenzdatei oder eine Angabe

Interview mit Lina Ceballos

Linux-Magazin: *Wie würdest du deine Rolle innerhalb der FSFE und des REUSE-Projekts beschreiben?*

Lina Ceballos: Ich bin Politik-Projektmanager bei der FSFE, einer gemeinnützigen Organisation, die Nutzern die Möglichkeit gibt, Technologien zu kontrollieren. Ich habe Erfahrung in rechtlichen Fragen und der Einhaltung von Lizenzen sowie in der Überwachung von Gesetzgebungsprozessen auf europäischer Ebene, wo ich mit verschiedenen Interessenvertretern und Entscheidungsträgern zusammenarbeite. Im Rahmen des REUSE-Projekts konzentriere ich mich hauptsächlich darauf, die Bedeutung des Projekts in unserer Gemeinschaft hervorzuheben und es anderen zu zeigen, die davon stark profitieren können, aber vielleicht noch nichts davon gehört haben.

LM: *REUSE hat die Rechtssicherheit erhöht. Bietet dies deiner Erfahrung nach eine klare Antwort auf alle Lizenzierungsfragen für Urheber? Oder ergeben sich daraus neue Lizenzierungsfragen?*

LC: Lass uns zunächst darüber sprechen, worum es sich bei REUSE eigentlich handelt. REUSE umfasst eine Reihe von Best Practices, die darauf abzielen, die Kommunikation von Lizenz- und Copyright-Informationen von FOSS-Projekten für jedermann einfacher zu machen. Das ultimative Ziel ist, dass für jede einzelne Datei in einem Projekt diese rechtlichen Informationen in einer eindeutigen und für Mensch und Maschine gleichermaßen lesbaren Weise vorliegen. Die REUSE-Spezifikation sowie ein passendes Hilfsprogramm machen den Prozess quasi zum Kinderspiel. Wir stellen zudem ein Tutorial und eine FAQ bereit, die den Einstieg erleichtern und Antworten auf eine Reihe von Fragen geben, die bei der Einführung von REUSE auftreten können.

In dieser Hinsicht macht es REUSE also für jeden einfacher, die rechtlichen Informationen anzuzeigen und zu finden. Allerdings lassen sich nicht alle Lizenzierungsfragen standardmäßig lösen. Deswegen offeriert die FSFE neben der REUSE-FAQ eine weitere Zusammenstellung häufig gestellter Fragen zu freier Software, Urheberrecht und Lizenzierung. Für diejenigen, die vielleicht mehr Hilfe benötigen, als die FAQ bieten können, gibt es eine Mailing-Liste für Lizenzierungsfragen. Dort beantworten unsere freiwilligen Experten für rechtliche Fragen zu Freier Software komplexere Fragen.

Im Rahmen von REUSE bieten wir daneben individuelle Einschätzungen sowie direkte Unterstützung an, was dazu beitragen kann, viele der Zweifel auszuräumen, die Entwickler und Betreuer freier Software haben können. Genau wie freie Software basiert auch das REUSE-Projekt auf seiner Gemeinschaft und den Menschen, die es unterstützen und annehmen. Daher betreiben wir eine weitere Mailing-Liste, um andere Nutzer kennenzulernen und die Weiterentwicklung von REUSE zu fördern.

Zurzeit gibt es Tausende Projekte, die REUSE implementierten. Dazu zählen Projekte wie KDE, GNU Health, Curl und teilweise der Linux Kernel. Die Mehrzahl der Projekte des Next Generation Internet hat ebenfalls REUSE übernommen. Ich möchte daher alle Entwicklerinnen und Entwickler ermutigen, REUSE in ihren FOSS-Projekten zu nutzen. Damit helfen sie uns dabei, REUSE zum Standard zu machen, um rechtliche Informationen auf einfachere Weise darzustellen.

LM: *Wie kann man die FSFE und das REUSE-Projekt im Einzelnen unterstützen?*

LC: Die FSFE ist eine gemeinnützige Organisation. Wir verdanken also einen Großteil unserer Arbeit der Unterstützung all unserer Freiwilligen und Mitwirkenden. Sie unterstützen uns in unseren Arbeitsbereichen, organisieren Veranstaltungen zur Förderung freier Software und helfen uns bei Übersetzungen. Auch REUSE hat ein wunderbares und kompetentes Team von Freiwilligen, die das Projekt vorantreiben und sich über Beiträge anderer freuen. Die Mitwirkungsmöglichkeiten reichen von der Verbesserung unserer Dokumentation über Hilfe bei der Überprüfung von Problemen bis hin zu Code-Checks in offenen Pull Requests.

Außerdem kann man unsere Arbeit an REUSE mit Spenden oder als Unternehmen via Sponsoring unterstützen. Wenn deine Firma also daran interessiert ist, auf diesem Weg zu REUSE beizutragen, sprich uns bitte an. Möchtest du dich uns als Freiwillige(r) anschließen, abonniere unsere REUSE-Mailing-Liste. Auf unserer Website findest du weitere Informationen über unsere Initiative und Möglichkeiten, mit uns in Kontakt zu treten.

LM: *Ganz herzlichen Dank für diesen Einblick in die Arbeit von REUSE!*

mit dem Trove Classifier `License :: ...`. Abbildung 4 zeigt als Beispiel die Angaben für die Bibliothek Matplotlib.

Während die `LICENSE`-Datei alle Lizenzen nennt, lässt der Trove Classifier nur eine einzige Lizenz daraus zu. Das ist insofern unglücklich, als der Classifier auch an anderen Stellen ausgewertet wird, beispielsweise im Python Package Index. Dort erscheint dann ebenfalls nur diese eine Lizenz, und die anderen Lizenzen, die in der `LICENSE`-Datei angegeben sind, fallen unter den Tisch.

Ärgerlich sind obendrein die unterschiedlichen Bezeichnungen für die Lizenzen. Sie machen eine Recherche und einen Vergleich der ausgewählten Lizenzierungen komplizierter als eigentlich notwendig. Eine Vereinheitlichung tut hier dringend Not.

Richtig implementieren

Um Klarheit zur Lizenz zu bekommen, hilft das Konzept rund um SPDX. Wie im ersten Teil der Serie schon angerissen, bietet es eine standardisierte Lösung für die beschriebenen Probleme.

Jede Quellcodedatei enthält am Anfang einen passenden Eintrag als Kommentar. Er beginnt mit dem Text `SPDX-License-Identifier:`, darauf folgt die gewählte Lizenz für den Programmcode. Listing 4 zeigt das für die Nutzung der GNU Public License v3 oder später. Diese Schreibweise erlaubt dank der standardisierten Formulierung das maschinelle, automatisierte Auslesen. Das schließt eine Fehlinterpretation der Lizenz aus.

Externe Unterstützung

Unterstützung bei der Lizenzauswahl ist stets hilfreich, da die große Vielfalt an Lizenzmodellen das Verständnis erschwert. Hier helfen sowohl das GNU-Projekt als auch Wikipedia und Choose A License mit Hinweisen und Übersichten weiter.

Die Free Software Foundation Europe (FSFE) lässt sich diesbezüglich ebenfalls nicht lumpen und bietet sich als kompetenter Anlaufpunkt an. Wir haben Lina Ceballos von FSFE dazu befragt, wie die Hilfe im Rahmen des REUSE-Projekts konkret aussieht (siehe Kasten Interview mit Lina Ceballos).

Nicht zu vergessen sind zu guter Letzt auch die auf Lizenzrecht spezialisierten Beratungsfirmen und Rechtsanwälte, die weitere Klarheit bringen können.

Listing 3: Lizenz (SPEC-Datei)

```
License: MPL-1.1 OR
GPL-2.0-or-later
```

Fazit

Bislang lag unser Blick im Rahmen dieser Artikelreihe primär auf Software und ihre korrekte Lizenzierung. Was noch fehlt, aber keinesfalls übersehen werden darf, sind Daten und Datenströme, Bilder, Dokumentation, Mediadaten und Hardware. Diesem Thema widmen wir uns im nächsten Teil dieser Serie. (*jlu*)

Danksagung

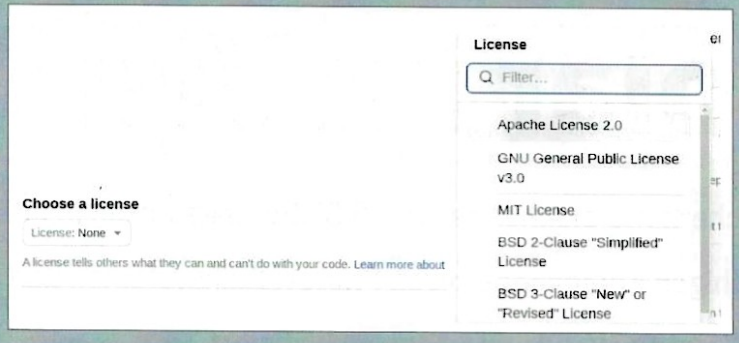
Die Autoren bedanken sich bei Axel Beckert für seine Kritik und Anregungen bei der Vorbereitung des Artikels.

Weitere Infos und interessante Links

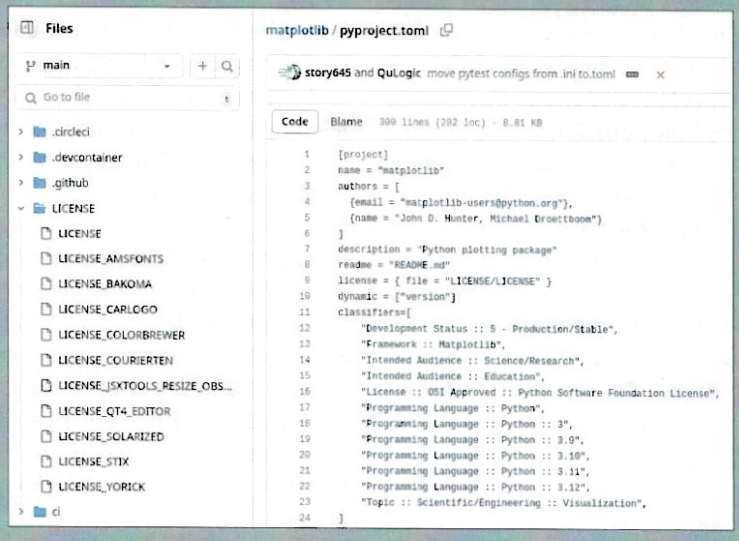
www.lm-online.de/qr/52165

Listing 4: SPDX-Lizenzangabe

```
# SPDX-License-Identifier: GNU General Public License v3.0 or later
```



3 So unkompliziert funktioniert die Lizenzauswahl bei Github.



4 So sehen die Lizenzangaben für die Bibliothek Matplotlib aus.