

Dr. Wolfgang Straub
Augsburger Deutsch & Partner
Postfach 5860
CH-3001 Bern

www.it-recht.ch
wolfgang.straub@ad-p.ch

Checklist Open Source Licenses

1. Wahl eines geeigneten Open Source Modells

Soll neue Software unter einem Open Source Modell verbreitet werden, sind bei der Auswahl des geeigneten Lizenztyps insbesondere folgende **Kriterien** in Betracht zu ziehen:¹

- **Kompatibilität** mit Lizenzen für Drittsoftware
- **Akzeptanz** der Lizenz bei der 'Zielcommunity'
- **Complianceaufwand** für die Einhaltung der betreffenden Lizenz
- Vorhandensein einer Institution zur Rechtsdurchsetzung

¹ Siehe dazu auch die Übersichtstabelle über gängige Open Source Lizenzen bei BEARDWOOD, JOHN P. / ALLEYNE, ANDREW C, The Price of Binary Freedom: The Challenge of Open Hybrid Software, CRi 2006, S. 97-104, S. 103f., sowie die Gegenüberstellungen bei THALHOFER THOMAS, Commercial Usability of Open Source Software Licenses: To what extent can software governed by GNU or alternative licenses be commercially exploited? CRi 2008, S. 129-136, S. 132ff.; FRÖHLICH-BLEULER GIANNI, Urheber- und vertragsrechtliche Aspekte der Open Source Software, in: Jörg Florian S./Arter Oliver (Hrsg.), IT-Verträge, 10. Tagungsband, Bern: Stämpfli 2007, S. 179-232, S. 218ff.

2. Copyleft Vermeidungsstrategien

Zur Vermeidung des Copyleft-Effekts können z.B. folgende **Strategien** verwendet werden:

- **Änderungen** nicht selbst durchführen, sondern **durch die Endbenutzer** vornehmen lassen. Statt eines modifizierten Codes wird diesen allenfalls eine Anleitung geliefert, wie sie das Programm ändern können.
- Programme **getrennt verbreiten**: Wenn die eigene Software keinen urheberrechtlich geschützten Code von Open Source Programmen enthält, kann diese grundsätzlich alleine vertrieben werden.
- **Technische Distanzierungsstrategien entwickeln**: Es wird z.B. ein besonderes Modul geschaffen, das zwischen die Open Source Software und das neue Programm gestellt wird. Nur das betreffende Modul soll vom Copyleft-Effekt erfasst werden. GPL v3 enthält allerdings eine Abwehrklausel, um technische Umgehungen des Copyleft zu vermeiden.²
- **Parallele Lizenzen** bei den Rechtsinhabern einholen. In diesem Zusammenhang ist allerdings zu beachten, dass Programmschöpfer und Rechtsinhaber auseinanderfallen können (z.B. bei Softwareentwicklung im Rahmen von Arbeits- oder Werkverträgen).
- Eine **Ausnahme** bei der Free Software Foundation (FSF) einholen. Dies setzt allerdings voraus, dass die FSF über die Rechte an der betreffenden Software verfügen kann. Das ist nur dann der Fall, wenn diese ihr von den Rechtsinhabern zumindest fiduziarisch eingeräumt wurden.

² Ziff. 5 Abs. 2 GPL v3. Siehe dazu auch BEARDWOOD, JOHN P. / ALLEYNE, ANDREW C., Open Source Hybrids and the Final GPLv3, CRi 2008, S. 14-20, S. 16.

3. Open Source Compliance

Beim Einsatz von Open Source Software in Entwicklungsprojekten sind insbesondere folgende Punkte zu klären und zu dokumentieren:³

- Erarbeiten einer **Open Source Strategie**: In welchen Bereichen und unter welchen Bedingungen soll Open Source Software überhaupt eingesetzt werden.
- **Welche Software** wird überhaupt verwendet? Welche Building Instructions, Installationsskripte, Compiler etc. werden eingesetzt? Konkret ist alles zu dokumentieren, was eine Fachperson braucht, um selbst ein lauffähiges Programm herzustellen. Alle Versionen der betreffenden Open Source Software sind im Rahmen eines konsequenten Release Managements zu dokumentieren.⁴
- **Welche Lizenzen** sind auf die verwendeten Softwarekomponenten anwendbar? Sind diese überhaupt miteinander kompatibel?⁵ Wie können allfällige Kompatibilitätskonflikte vermieden werden?
- In welcher Form wird Open Source Software mit eigenen Programmen oder mit Drittprodukten⁶ **verbunden** und **weitergegeben**?⁷
- Für welche Softwarekomponenten ist der **Copyleft-Effekt** eingetreten?⁸

³ Siehe dazu auch KUHNS BRADLEY M. / WILLIAMSON AARON / SANDLER KAREN M., A Practical Guide to GPL Compliance, New York 2008, online verfügbar unter www.softwarefreedom.org, S. 2ff.; BEARDWOOD / ALLEYNE, Binary Freedom, S. 102.

⁴ Dazu kann z.B. das Fossology System eingesetzt werden. Siehe www.fossology.org.

⁵ Siehe zur Kompatibilitätsproblematik STRAUB WOLFGANG, Softwareschutz: Urheberrecht, Patentrecht, Open Source, Zürich 2011, Rz. 675ff.

⁶ In diesem Zusammenhang ist zu beachten, dass beim Einsatz in Drittprodukten sowohl der Entwickler der Software als auch der Hersteller des Endprodukts die Weitergabepflichten für den Sourcecode einhalten müssen. Siehe zu deren Inhalt STRAUB, Softwareschutz, Rz. 697ff. Im Fall der Integration der Open Source Software in Endbenutzerprodukte muss zudem eine Installationsanleitung mitgeliefert werden, die es den Benutzern erlaubt, modifizierte Versionen der Software auf dem Produkt zu installieren (Ziff. 6 Abs. 3 GPL v3).

⁷ Wenn Pakete aus eigenständigen Softwarekomponenten gebildet werden, muss eindeutig gekennzeichnet werden, welche Komponente unter welcher Lizenz steht (z.B. durch entsprechend beschriftete Ordnerstruktur). Siehe zum Aggregieren von Software auch STRAUB, Softwareschutz Rz. 716.

- Sourcecodes korrekt **publizieren** und Verfügbarkeit aller Versionen (inkl. Building Instructions etc.) während der durch die jeweilige Lizenz vorgeschriebenen Mindestdauer sicherstellen.⁹
- **Aktives Bemühen um Compliance** im Fall eines Verletzungsvorwurfs zeigen.¹⁰

⁸ Siehe dazu STRAUB, Softwareschutz, Rz. 689ff.

⁹ Unter der GPL muss der gesamte Sourcecode mindestens 3 Jahre nach der letzten Weitergabelung verfügbar sein. Sofern die Software noch unter GPL v2 steht, müssen eventuell physische Datenträger bereitgehalten werden, welche zu Selbstkosten (Datenträger plus Versand) abzugeben sind. Siehe dazu KUHN / WILLIAMSON / SANDLER, S. 4ff.

¹⁰ Falls der Verletzer vom Rechtsinhaber erstmalig abgemahnt wurde oder innert 60 Tagen keine Ankündigung der Auflösung erfolgt, lebt die Lizenz definitiv wieder auf (Ziff. 8 Abs. 2 und Ziff. 3 GPL v3). Ein kooperatives Verhalten kann daher nicht nur zur Abwendung allfälliger Schadenersatzforderungen sinnvoll sein.