

Rechtsdurchsetzung bei Freier Software

Exemplarisch gezeigt am Urteil des Oberlandesgerichts Hamburg (Urteil vom 28.02.2019, Az. 5 U 146/16)

Falk Zscheile

Chemnitzer Linuxtage, 14. März 2021

- 1 Einleitung
- 2 Abgrenzung der Urheberrechte
 - Einzelurheber*in
 - Miturheber*in
 - Bearbeitungsurheber*in
 - Freie Benutzung im Urheberrecht
- 3 Urheberrechtsschutz und Durchsetzung
 - Beurteilung von Programmierleistungen
 - Dokumentation der Urheberschaft
- 4 Zusammenfassung
 - Ergebnis
 - Abschluss
 - Literatur

Gliederung

- 1 Einleitung
- 2 Abgrenzung der Urheberrechte
- 3 Urheberrechtsschutz und Durchsetzung
- 4 Zusammenfassung

FOSS-Lizenzen eine (rechtliche) Erfolgsgeschichte?

- Licht:
 - FOS(S)-Lizenzen werden eingesetzt
 - FOS(S)-Lizenzen sind durch die Rechtsordnung anerkannt.
 - Copyleft-Klauseln sind wirksam.
 - Juristen auf den FOSSDEM 2020 „feiern“ FOSS – Heise-Newsticker: „Wir haben gewonnen. Und jetzt? [...]“ [8]
- Schatten
 - Ist FOS(S) wirklich im Recht angekommen?
 - (Unter-)Stützt die Gesetzeslage des Urheberrechtsgesetzes (UrhG) FOS(S)?
 - Lassen sich FOS(S)-Lizenzen rechtlich gut durchsetzen?
 - Berücksichtigt das Recht Eigenheiten der FOSS-Entwicklung?
 - Bedrohen Cloudmodelle FOS(S)?

OLG Hamburg, Urteil vom 28.02.2019, Az. 5 U 146/16

Besonderheiten des Urteils

- Sachverhalt und Urteilsgründe ca. halbe-halbe
- Sowohl prozessuale als auch materielle Rechtsfragen
 - Prozessrecht
 - Anforderungen an Beweisangebote
 - Neuer Tatsachenvortrag in der Berufungsinstanz
 - Beweiserleichterungen (Beweislastumkehr, Anscheinsbeweis)
 - Materielles Recht
 - Ist Urheberrechtsschutz gegeben?
 - Wenn ja – welcher?
 - Wer darf die Urheberrechte geltend machen?
- Manche materiellen Fragen mussten nicht behandelt werden.
- Dennoch obiter dicta.

Gliederung

- 1 Einleitung
- 2 Abgrenzung der Urheberrechte
 - Einzelurheber*in
 - Miturheber*in
 - Bearbeitungsurheber*in
 - Freie Benutzung im Urheberrecht
- 3 Urheberrechtsschutz und Durchsetzung
- 4 Zusammenfassung

Ausgangspunkt: Einzelurheber*in

- Historische Ausgangslage
 - Das Urheberrechtsgesetz ist im Kern „analog“.
 - Das Urheberrechtsgesetz ist „alt“.
 - Die individuellen Urheberrechtseinhabenden stehen jeweils im Mittelpunkt.
- Konsequenzen:
 - Ausgangspunkt ist der/die Einzelurheber*in, § 7 UrhG.
 - Miturheberschaft ist ein Sonderfall.
 - Bearbeitungsurheberschaft ist ein Sonderfall.
 - Ausnahmeregelungen für Gruppenwerke gibt es nur für ausübende Künstler*innen, § 80 Abs. 2 UrhG.

Urheberrecht

- Das Urheberrecht entsteht immer bei einer natürlichen Person, § 2 Abs. 2 UrhG i. V. m. § 3 UrhG
- Arbeits- oder Dienstverhältnis
 - Urheber*innen im Arbeits- oder Dienstverhältnis: Unternehmen bzw. Dienstbehörden erhalten nur Nutzungsrechte an der Leistung, § 43 UrhG
 - Bei programmierenden Personen werden Kraft Gesetzes alle Rechte dem Unternehmen bzw. der Dienstbehörde übertragen, § 69b UrhG.

Miturheberschaft

- Was ist Miturheberschaft?
 - Alle an der Werkvollendung schöpferisch Beteiligten werden zu Personen der Miturheberschaft.
 - Alle Beitragenden müssen einen schöpferischen Beitrag leisten.
 - Auf Zeitpunkt, Umfang und Bedeutung des Beitrags vor Werkvollendung kommt es nicht an^[2].
- Welche Fälle kennt der Jurist?
 - Bücher
 - Film- und Multimediawerke
 - Computerprogramme
 - BGH. Urteil vom 9. Mai 1985. Az. I ZR 52/83, Inkasso-Programm
 - BGH. Urteil vom 14. Juli 1993. Az. I ZR 47/91, Buchhaltungsprogramm

Miturheberrecht

- Gemeinsame Schöpfung eines Werkes – gemeinsame Tätigkeit mehrerer Personen.
- Jeder Beitrag muss für sich wieder eine persönlich-geistige Schöpfung sein.
- Alle müssen ein gemeinsames Werk wollen.
 - Das wirt für FOSS zum Teil bezweifelt[7, S. 351]!
- idealtypische Vorstellung: gleichzeitige Tätigkeit am Werk
- Zeitlich versetzte Beiträge sind möglich.
 - Aber: Unterordnung jedes Beitrags unter die Gesamtidee notwendig![5, Rn. 155]

Sicht des Rechts auf die Software(-entwicklung)

Serielle Sichtweise des Urheberrechts:

Zusammenarbeit – Werkvollendung – Verwertung – (Bearbeitung) – Verwertung

BGH, Urteil vom 9. Mai 1985, Az. I ZR 52/83

Problemanalyse – Datenflussplan/Programmablaufplan – Kodierung als Programm[1]

Urheberrecht

Jedes Softwarerelease ist ein abgeschlossenes Werk.

Vertragsrecht

Jedes Softwarerelease (Standardsoftware on premise) ist ein Kaufgegenstand.

Sicht von FOSS auf die Software(-entwicklung)

Zyklische Sichtweise der Softwareentwicklung (inkrementell und iterativ):

planen – erstellen/umsetzen – testen – bewerten – planen usw.

FOSS Softwareentwicklung

Jedes Softwarerelease ist nur ein Zwischenstand auf dem Weg zu (noch) besserer Software.

Bearbeitungsurheberrecht, welche Fallgestaltungen kennen Juristen?

- Bearbeitungsrecht ist kein originäres Urheberrecht, sondern ein Verwertungsrecht (Nutzungsrecht).
- Normalfall: Fertiggestelltes Werk wird von einer anderen Person verändert.
 - Veröffentlichung einer Umgestaltung nur mit Einwilligung des Urhebers, § 3 UrhG
 - Viel enger bei Computerprogrammen: Verbot jedweder Umarbeitung, § 69c Nr. 2 UrhG
- Sonderkonstellation: Fortsetzung fremder Werke
 - Abgrenzungsproblem Bearbeitung oder Miturheberschaft?
 - Besonders kritisch: Die Fortsetzung, insbesondere Vollendung, fremder Werke.
- Sonderfall FOSS: Das Bearbeitungsrecht wird immer mit der Lizenz eingeräumt, sonst ist es keine FOSS!

Linux-Beitrag: Bearbeitungsurheberrecht oder Miturheberrecht?

- Im Ausgangsverfahren[4] (LG Hamburg) wurde nur Bearbeitungsurheberrecht diskutiert.
- Ein mögliches Miturheberrecht wurde erst in der Berufungsinstanz (OLG Hamburg) und damit zu spät[5, Rn. 155] vorgetragen (Präklusion), § 531 Abs. 2 ZPO.

Das Bearbeitungsurheberrecht

- Es setzt ein vorbestehendes geschütztes Werk (persönlich-geistige Schöpfung) voraus.
- Das Bearbeitungsurheberrecht entsteht nur am bearbeiteten Auszug des vorbestehenden Werkes.
- Der bearbeitete Auszug muss selbst eine persönlich-geistige Schöpfung enthalten.

Werk des Bearbeitungsurhebers

Konsequenz 1:

Je mehr Bearbeitungsurheber, desto geringer die Chance auf urheberrechtlichen Schutz!

Konsequenz 2:

Je öfter ein Werk verbessert wird, desto geringer seine Chance auf Schutz.

Freie Benutzung

- Die schöpferische Eigentümlichkeit des Originals muss im neuen verblassen! [5, Rn. 170]
- Das Original dient nur als Anregung für das neue.
- Ist das Original schon wenig originell, „verblasst“ es schneller.

Meinung:

Diese Kriterien machen bei Kunst Sinn, aber nicht bei Programmcode, wenn dieser 1-zu-1 übernommen wird.

Freie Benutzung von Codebestandteilen

- Die Verwendung nur kleiner Programmbestandteile ist eine freie Benutzung, § 24 Abs. 1 UrhG.[5, Rn. 169].
 - Entnahme vieler geringwertiger Teile bleibt freie Bearbeitung.
 - Anders im Datenbankrecht: Viele kleine Auszüge sind ein großer lizenzpflichtiger Auszug!
 - Fluch für FOSS: Die individualistischen Konzeption der Urheberrechts.
- Das OLG Hamburg setzt die Einzelbeiträge ins Verhältnis zum Gesamtcode und begründet so eine freie Benutzung.[5, Rn. 171]
 - Fremder Code kann im selbst geschaffenen Code verblassen!
 - Wesentliches Indiz ist der geringe quantitative (!) Anteil am Gesamtcode.

Gliederung

- 1 Einleitung
- 2 Abgrenzung der Urheberrechte
- 3 Urheberrechtsschutz und Durchsetzung**
 - Beurteilung von Programmierleistungen
 - Dokumentation der Urheberschaft
- 4 Zusammenfassung

Urheberrechtsschutz für Computerprogramme

- 1985 GNU-Manifest[6]
- Bis 1985 wurden Computerprogramme in Deutschland als allgemeines Sprachwerk kategorisiert (Textzusammenstellung), § 2 Abs. 1 Nr. 1 UrhG.
- Seit 1985 werden Computerprogramme ausdrücklich in § 2 Abs. 1 Nr. 1 UrhG erwähnt.
- Bis 1993: Schutz nur für Leistungen, die das handwerkliche Können eines Durchschnittsprogrammierers deutlich übersteigen[1].
- Seit 1993 müssen Computerprogramme nur eine individuelle Werke als eigene geistige Schöpfung des Urhebers sein, § 69a Abs. 3 Satz 1 UrhG
 - Die „kleine Münze“ wird geschützt.
 - Nur einfache routinemäßige Programmierleistungen bleiben schutzlos – alle machen es gleich oder ähnlich.

Beweiserleichterung für umfangreiche Programmierleistungen

- Grundsatz im Zivilprozessrecht: Wer für sich ein Recht geltend macht, muss es im Zweifel beweisen.
- Konkret: Beim Bestreiten muss für jede Programmierleistung die Schutzfähigkeit nachgewiesen werden (individuelles Werk mit geistiger Schöpfung), § 69a Abs. 3 Satz 1 UrhG i. V. m. § 2 Abs. 1 Nr. 1 UrhG.
- BGH: Vermutung für Urheberrechtsschutz bei umfangreicher Programmierleistung[3] – Beweislastumkehr.
 - Im konkreten Fall des BGH: 1 Hauptentwickler sowie 2 weitere Entwickler.
- OLG Hamburg[5, Rn. 166]: Keine Erstreckung auf kollaborativ erstellte Werke bei
 - vielen Einzelbeiträgen unterschiedlicher Personen,
 - die für sich nicht „komplex“ sind.

Nachteile für FOSS aus der Rechtsprechung des OLG Hamburg

Gefahr 1:

Gefahr 1: FOSS wird gegen (proprietärer) Einzelleistung benachteiligt: „Nur Einzelkämpfer können Großartiges leisten.“

Gefahr 2:

FOSS wird faktisch schutzlos gestellt: „Jeder Beitragende weist bitte seine Leistung konkret nach.“

Gefahr 3:

Microservices-Konzepte kommen eventuell nicht in den Genuss der Beweiserleichterung – wenn die Services nicht „komplex“ sind.

Urheberschaftsvermutung

- Grundsatz im Zivilprozessrecht: Wer für sich ein Recht geltend macht, muss es im Zweifel beweisen.
- Vermutung der Urheberschaft zu Gunsten der Person, die als Urheber*in im Werk bezeichnet wird, § 10 Abs. 1 UrhG i. V. m. § 292 ZPO.
- Beweislastumkehr zu Gunsten der genannten Person.
- Problem: Die Nennung muss auf dem Werk oder der Werkskopie erfolgen § 10 Abs. 1 UrhG.
- Klassisches Verfahren bei Software: Die programmierende Person schreibt ihren Namen in den Programmcode.

Urheberschaftsdokumentation bei Softwarecode

- Klassisches Verfahren bei Software: Dokumentation im Programmcode
 - Nachteil 1: Ist kein Programmcode, stört also ggf. bei Beteiligung vieler.
 - Nachteil 2: Kennzeichnung des jeweiligen Anteils bei Beteiligung vieler sehr umständlich.
 - Nachteile 3: Kennzeichnung bei Bearbeitung an verschiedenen Stellen (bei vorbestehenden Code) fast unmöglich.
 - Nennung im Programmcode ist nur *Textform*.
- Versionsverwaltung (Git):
 - Jeder Beitrag und jede Änderung wird zeilengenau dokumentiert.
 - Beitrag und Änderung werden mit einer Person verknüpft.
 - Änderungshistorie ist dank kryptographischer Verfahren manipulationssicher nachvollziehbar (Integrität)[vgl. auch 7, S. 357].
 - Nachweis einer Autorenschaft mit *fortgeschrittener* oder *qualifizierter elektronischer Signatur* möglich.
 - Problem: Die Dokumentation erfolgt *außerhalb* des Programmcodes/Werkstücks.

Sicht des OLG Hamburg und LG Hamburg

OLG Hamburg, Urteil vom 28.02.2019, Az. 5 U 146/16, Rn. 129

Selbst wenn in den Versionsgeschichten jedoch der Kläger als letzter Bearbeiter benannt werden würde, hat die Beklagte zutreffend geltend gemacht, dass Versionshistorien keine Vervielfältigungsstücke des betreffenden Computerprogramms sind, so dass von ihnen keine Vermutungswirkungen nach § 10 UrhG ausgehen können.

Kritik an den Ausführungen:

- ausschließliche Orientierung am Wortlaut des [§ 10 Abs. 1 UrhG](#)
 - Andere juristische Auslegungsmethoden, insbesondere Sinn und Zweck, werden außer acht gelassen.
 - Allerdings Umsetzung einer europäischen Richtlinie.
- fortschritts- und technikfeindliche Auslegung
- unreflektiertes Festhalten am Alten und Bekannten

Gliederung

- 1 Einleitung
- 2 Abgrenzung der Urheberrechte
- 3 Urheberrechtsschutz und Durchsetzung
- 4 Zusammenfassung
 - Ergebnis
 - Abschluss
 - Literatur



Ergebnis

- Besonderheiten bei der Entstehung von FOSS werden bisher im Recht nicht hinreichend gewürdigt bzw. wirken sich sogar nachteilig aus.
- Dem Urheberrecht ist altruistisches kollaboratives Zusammenwirken fremd.
- Urheberrechtliche Regeln für Computerprogramme bevorzugen die proprietäre Softwareentwicklung.
- FOS(S) steht bei der Würdigung im Urheberrecht immer noch am Anfang.
- FOS(S) ist dem Urheberrechtsgesetz weitgehend fremd.
- Beim kollaborativen Zusammenwirken steckt das Urheberrecht in einer Sackgasse.
- Die Vorstellungswelt der Gerichte spiegelt zeitgemäße Methoden und Techniken in der Softwareentwicklung nicht wider.

Vielen Dank für die Aufmerksamkeit!

Fragerunde

Literaturnachweise I

-  BGH. Urteil vom 9. Mai 1985. Az. I ZR 52/83, Inkasso-Programm.
-  BGH. Urteil vom 14. Juli 1993. Az. I ZR 47/91, Buchhaltungsprogramm.
-  BGH. Urteil vom 3. März 2005. Az. I ZR 111/02, Fash 2000.
-  LG Hamburg. Urteil vom 8. Juli 2016. Az. 310 O 89/15.
-  OLG Hamburg. Urteil vom 28. Feb. 2019. Az. 5 U 146/16.
-  Richard Stallman. *GNU-Manifest*. 1985. URL: <https://www.gnu.org/gnu/manifesto.de.html> (besucht am 06.03.2021).
-  Marcus von Welser und Alexander Hoppen. „Nachweis von Urheberrechten an Source Code mittels Git. Was Online-Plattformen wie GitHub zur urheberrechtlichen Wertung beisteuern können“. In: *CR* (2020), S. 350–360.

Literaturnachweise II



David Wolski. *FOSDEM Debatte: Ist die Definition von Open Source noch zeitgemäß?* 2. Feb. 2020. URL: <https://www.heise.de/newsticker/meldung/FOSDEM-Debatte-Ist-die-Definition-von-Open-Source-noch-zeitgemaess-4651498.html> (besucht am 05.02.2020).