

SoftwareRecht: Von der Entwicklung zum Export

Probleme und Lösungen nach deutschem und
amerikanischem Recht

Prof. Dr. Diana D. Chiampi Ohly, LL.M. (Duke)
Hochschule Darmstadt

5., überarbeitete Auflage 2026

Alle im Buch verwendeten Begriffe verstehen sich geschlechterneutral. Aus Gründen der besseren Lesbarkeit wird auf eine geschlechtsspezifische Differenzierung verzichtet – entsprechende Begriffe gelten im Sinne der Gleichbehandlung grundsätzlich für alle Geschlechter. Die verkürzte Sprachform hat lediglich redaktionelle Gründe und beinhaltet keine Wertung.

Band 3 der Schriftenreihe des Instituts für Informationsrecht der Hochschule Darmstadt

Herausgeber:innen:

Prof. Dr. Diana D. Chiampi Ohly, LL.M. (Duke), Attorney-at-Law (New York)

Prof. Dr. Felix Hermonies, LL.M. (R.L.), Mag. rer. publ.

Prof. Dr. Gisela Jung-Weiser

Prof. Dr. Caroline Volkmann

Prof. Dr. Thomas Wilmer

Institut für Informationsrecht (i2r) an der Hochschule Darmstadt

Haardtring 100

64295 Darmstadt

<https://i2r.h-da.de/>

Bibliografische Information Der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.de> abrufbar.

I S B N 9 7 8 - 3 - 8 0 0 5 - 2 0 8 0 - 0

dfv' Mediengruppe

© 2026 Deutscher Fachverlag GmbH, Fachmedien Recht und Wirtschaft, Mainzer Landstr. 251, 60326 Frankfurt am Main, buchverlag@ruw.de

www.ruw.de

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Druck: Beltz Grafische Betriebe GmbH, 99947 Bad Langensalza

Printed in Germany

Vorwort

Die fortschreitende Digitalisierung hat das Recht der Informationstechnologien nicht nur vor neue Fragen gestellt, sondern es in einen permanenten Dialog mit der Technik gebracht. Kaum ein Bereich verdeutlicht dies so eindrücklich wie das Zusammenspiel von Recht und Informatik: Trainingsdaten werden im Zeitalter der KI zu einem Wirtschaftsgut, Algorithmen generieren Texte, Software „erfindet“ technische Lösungen – und Juristinnen, Juristen sowie Informatikerinnen und Informatiker stehen gemeinsam vor der Aufgabe, diese Entwicklungen rechtlich und technisch einzuordnen. Dieses Lehrbuch versteht sich als Begleiter genau an dieser Schnittstelle.

Ziel des Werkes ist es, Studierenden wie Praktikern aus Recht und Informatik einen systematischen, zugleich praxisnahen Zugang zu den zentralen Schutzrechten und Regelungen zu eröffnen, die im Kontext moderner Software- und KI-Systeme von Bedeutung sind. Behandelt werden jeweils das deutsche und das US-amerikanische Urheberrecht, das Patentrecht sowie das Geheimnisschutzrecht – stets mit Blick auf ihre jeweilige Funktion, ihre dogmatischen Grenzen und ihre praktische Durchsetzung. Ein besonderer Schwerpunkt liegt auf der Künstlichen Intelligenz: von den Grundlagen über urheber- und patentrechtliche Fragestellungen bis hin zu den aktuellen Regelungen der europäischen KI-Verordnung, einschließlich ihrer besonderen Bedeutung für Open-Source-Software.

Darüber hinaus widmet sich das Buch der häufig unterschätzten, in der Praxis jedoch relevanten Abgrenzung zwischen KI-Regulierung und Exportkontrollrecht. Wann wird Software zur kontrollierten Technologie? Welche Rolle spielen Trainingsdaten und Schnittstellen? Und weshalb kann ein vermeintlich harmloser Code im falschen Kontext plötzlich außenwirtschaftsrechtliche Aufmerksamkeit erregen?

Allen Kapiteln gemeinsam ist der konsequente Praxisbezug. Die dargestellten Rechtsfragen werden anhand aktueller, lebensnaher Fallbeispiele entwickelt und mit den jeweils einschlägigen gesetzlichen Vorschriften verknüpft und dann rechtsvergleichend den US-amerikanischen Regelungen des Fall- und Gesetzesrecht gegenübergestellt. Theorie soll hier nicht Selbstzweck sein, sondern Werkzeug. Entsprechend richtet sich das Buch nicht nur an Leserinnen und Leser, die sich erstmals mit der Materie befassen, sondern auch an diejenigen, die rechtliche Entscheidungen treffen müssen – etwa in Unternehmen, Kanzleien oder Entwicklungsabteilungen.

Die vorliegende fünfte Auflage trägt der dynamischen Entwicklung des Rechtsgebiets Rechnung. Gegenüber der ersten Auflage wurde das Werk nicht nur inhaltlich erweitert und aktualisiert, sondern auch strukturell weiterentwickelt. Zahlreiche neue Checklisten sollen dabei helfen, komplexe

rechtliche Probleme zu ordnen und typische Fallstricke frühzeitig zu erkennen. Sie sind als Einladung zu verstehen, Recht nicht nur zu lesen, sondern anzuwenden.

Dieses Buch erhebt keinen Anspruch darauf, jede zukünftige technische Innovation vorherzusehen. Es will jedoch die Grundlagen vermitteln, um neue Entwicklungen rechtlich einordnen – und dabei gelegentlich daran erinnern, dass selbst der raffinierteste Algorithmus am Ende doch wieder rechtliche Fragen stellt, die nur Menschen beantworten können. Wer also erwartet, dass Recht und Informatik stets widerspruchsfrei harmonieren, wird hier eines Besseren belehrt. Wer sich jedoch für genau diese Themenfelder interessiert, ist herzlich eingeladen, weiterzulesen.

März 2026

Diana D. Chiampi Ohly

Inhaltsverzeichnis

Vorwort	V
Vorwort zur 1. Auflage	VII
Fallübersicht	XIII
Checklisten	XIV
Übersichten	XV
Abkürzungsverzeichnis	XVI
Einleitung	1
Fallszenario – Ausgangsfall	2
I Die Softwareentwicklung nach deutschem Recht	6
1 Urheberrechtliche Betrachtung	6
1.1 Software als urheberrechtlich geschütztes Werk	6
1.2 Datenbankwerk und Datenbank und Analyse	11
1.3 Die Besonderen Bestimmungen für Computerprogramme .	20
1.3.1 Entwurfsphase	20
1.3.2 Abgrenzung zur Idee	21
1.3.3 Schutzzumfang	22
1.3.3 Smart Contracts und Blockchain	24
1.4 Urheber- und Rechtsinhaberschaft	30
1.4.1 Urheberschaft	31
1.4.2 Rechtsinhaberschaft im Arbeitsverhältnis	33
1.5 Künstliche Intelligenz	37
1.5.1 Erzeugnisschutz	37
1.5.2 Text und Data Mining	52
1.6 Der Rechtsstreit und -weg	54
1.5 Zustimmungsbefürftige Handlungen	57
1.5.1 Die Änderungsbefugnis	58
1.5.1.1 Versionierung	59
1.5.1.2 Urheberrechtsvermerk und Dauer des Urheberrechts	61
1.5.2 Die Sicherungskopie	64
1.5.3 Der bestimmungsgemäße Gebrauch	65
1.5.4 Der Erschöpfungsgrundsatz	70
1.5.4.1 Veräußerung	70
1.5.4.2 AGB- und kartellrechtliche Prüfung	72
1.5.4.3 Miete	79

1.5.4.4	Download und Gebrauchtssoftware	83
1.5.4.5	Volumenlizenzen	102
1.5.4.6	Pflege und Erschöpfungswirkung	110
1.5.4.7	Zweiterwerb durch Hardwareüberlassung. .	110
1.5.4.8	Weitere Verbote	113
1.5.4.9	Technische Maßnahmen	115
1.6	Dekompilierung	123
1.7	Code-Schutz als Geschäftsgeheimnis	132
2	Patentrechtliche Betrachtung	139
2.1	Zweck des Patents	140
2.2	Computerimplementierte Erfindungen	142
2.3	Voraussetzungen für Softwarepatente	144
2.3.1	Technizität.	144
2.3.1.1	Rechtsprechung zu computerimplemen- tierten Erfindungen	145
2.3.1.2	Rechtsprechung zu technischem Problem und technischer Leistung.	152
2.3.2	Neuheitsschädliche Vorveröffentlichung	164
2.3.3	Kriterien der erfinderischen Tätigkeit.	171
2.3.4	Gewerbliche Anwendbarkeit.	173
2.3.5	Maschinen als Erfinder.	173
2.4	Patentanmeldung	176
2.5	Arbeitnehmererfindung	179
2.6	Freie Erfindungen	188
2.7	Rechtsübertragung	191
2.8	Rechtsweg	196
2.8.1	Rechtsstreitigkeiten über Erfindungen	196
2.8.2	Beschwerdeverfahren	196
2.8.3	Klageverfahren wegen Nichtigkeit oder Zwangs- lizenz.	197
2.8.4	Zuständigkeit in Patentstreitsachen	198
II	Die Softwareentwicklung nach amerikanischem Recht	199
1	Betrachtung nach dem Copyright Law	199
1.1	Computerprogramme als Schutzgegenstand – Copyrightable Subject Matter	199
1.2	Schutzzumfang Computerprogramme – Scope of Protection	203
1.2.1	Abgrenzung Idee-Ausdrucksform – Idea-Expression Dichotomy	204
1.2.2	Umfang der Übereinstimmung mit dem Original – Substantial Similarity	207
1.2.3	Schutzzumfang	208

1.3	Schutzgegenstand der Sammelwerke und abgeleiteter Werke	224
1.3.1	Compilations	224
1.3.2	Derivative Works	227
1.3.3	Beispiele aus Open Source Verträgen	231
1.4	Rechtsinhaberschaft – Author- and Ownership	235
1.4.1	Miturheber – Co-Ownership	236
1.4.2	Works made for Hire	237
1.4.3	Sammelwerke	245
1.4.4	AI-Authorship	245
1.4.5	Rechtsübertragung – Transfer of Rights	251
1.5	Beschränkungen der Exklusivrechte	253
1.5.1	Berechtigte Benutzung – Fair Use	253
1.5.2	LLM und Fair Use	257
1.5.3	Partizipation des Rechtsinhabers – First Sale Doctrine	261
1.5.4	Weitere Beschränkungen der Exklusivrechte – Limitations on Exclusive Rights	273
1.6	Umgehung technischer Schutzmaßnahmen – Anti-Circumvention	275
1.7	Copyright Notice, Registrierung und Schutzdauer	280
1.8	Infringement, Remedies and Damages	285
1.8.1	Strafverfahren	287
1.8.2	Verfahren bei Anti-Circumvention	288
2	Patentierbarkeit – Patentability	289
2.1	Schutzgegenstand – Inventions patentable	289
2.2	Neuheit – Novelty Requirement	304
2.3	Priorität – Priority	314
2.4	Stand der Technik – Prior Art	316
2.5	AI-Inventor	318
2.6	Patentanmeldung – Application	319
2.7	Patentschutz – Patent Grant	325
2.8	Patentverletzung – Patent Infringement	327
2.9	Rechtsmittel – Remedies	328
3	Trade Secrets	332
III	Drittsoftware	337
1	Open Source Software	337
2	Open Source Software als Bestandteil proprietärer Software	340
2.1	Abgrenzung zu Public Domain Software, Freeware und Shareware	342
2.2	Copyleft-Effekt	345

Inhaltsverzeichnis

2.2.1	Lizenzen ohne Copyleft-Effekt.	347
2.2.2	Anwendbares Recht auf OSS-Verträge.	350
2.2.3	Lizenzen mit strengem Copyleft-Effekt.	357
2.2.4	Lizenzen mit beschränktem Copyleft-Effekt.	382
3	Kompatibilität.	389
4	Open Source Software und Künstliche Intelligenz.	394
5	IT-Compliance.	404
IV	Exportkontrolle.	406
1	Nationale und europäische Regelungen.	406
1.1	Genehmigungspflichten beim Export von Software.	407
1.2	Arten der Genehmigung.	417
1.2.1	Ausfuhr.	417
1.2.2	Allgemeine Genehmigung.	420
1.2.3	Einzelgenehmigung.	421
1.2.4	Sammelgenehmigung.	421
1.3	Compliance und Verstöße gegen Genehmigungspflichten.	422
2	U.S.-Exportkontrollvorschriften.	428
2.1	Exporte.	431
2.2	Produktklassifizierung.	433
2.3	Ausnahmen – Licence Exceptions.	436
2.4	De-minimis-Level.	440
2.5	Weitere Voraussetzungen.	446
2.6	Verstoß gegen die Exportkontrollvorschriften.	446
2.7	Berücksichtigung in Verträgen.	449
	Schlusswort.	452
	Literaturverzeichnis.	453
	Aufsätze.	453
	Kommentare.	462
	Handbücher, Lehrbücher, Monographien und Sammelbände.	464
	Stichwortverzeichnis.	469

Fallübersicht

- Fall 1: „Best of S“
- Fall 2: „Entwurfsklau“
- Fall 3: „PC-Springmaus“
- Fall 4: „Undone“
- Fall 5: „Smart Chain“
- Fall 6: „Geltungsbedürfnis“
- Fall 7: „Die versagte Zustimmung“
- Fall 8: „Internet und nicht weiter!“
- Fall 9: „Legal & Tech“
- Fall 10: „Programmierpionier“
- Fall 11: „Robotic Art“
- Fall 12: „KI-Überraschung – für Groß und Klein“
- Fall 13: „KI-easy!“
- Fall 14: „Ex-Arbeitnehmer auf Abwegen zum Rechtsweg“
- Fall 15: „DJ’s Mix“
- Fall 16: „Safe mit Backup“
- Fall 17: „Recht erschöpft oder AGB okay?“
- Fall 18: „Softwareüberlassung in der Cloud“
- Fall 19: „Softwareerwerb online“
- Fall 20: „Überlassungsmix“
- Fall 21: „Gepflegtes Volumen“
- Fall 22: „Ideentest“
- Fall 23: „Rolle rückwärts“
- Fall 24: „Alles im Handbuch“
- Fall 25: „TopCode Secret“
- Fall 26: „Alles neu?“
- Fall 27: „Patente KI: Better sAIfe than sorry“
- Fall 28: „Patente PC-Springmaus“
- Fall 29: „Quellcodeoffenbarung?“
- Fall 30: „Patente Arbeitnehmerin“
- Fall 31: „Frist vermisst?“

- Fall 32: „Fristlos“
Fall 33: „Weiterentwicklung“
Fall 34: „Patente Kooperation“
Fall 35: „Working together“
Fall 36: „Mark(e)t-Testing“
Fall 37: „Open Source inclusive“
Fall 38: „Open und Closed Source“
Fall 39: „Personalauswahl Kinder leicht“
Fall 40: „Soft-Where to Go?“

Checklisten

- Checkliste 1: Abgrenzung Datenbankwerk und Datenbank
Checkliste 2: Arbeitnehmerautor contra Rechtsinhaber
Checkliste 3: Softwarepatente
Checkliste 4: Patentanmeldung
Checkliste 5: Faktoren zur Vergütungsberechnung
Checkliste 6: Arbeitnehmererfindungsrecht
Checkliste 7: Lizenzumfang bei Integration
Checkliste 8: Kriterien zur Bestimmung des Umfangs der Verletzung
Checkliste 9: Registrierung
Checkliste 10: Drittsoftware
Checkliste 11: KI-Compliance unter Verwendung von OSS
Checkliste 12: Genehmigungspflicht der Ausfuhr
Checkliste 13: Unternehmensinternes Vorgehen beim Export

Übersichten

Übersicht 1: Computerprogramm-Zusammensetzung

Übersicht 2: Versionierung

Übersicht 3: Rechtsübertragung

Übersicht 4: Rechte am Programm

Übersicht 5: Vor- und Nachteile von Open Source Software

Übersicht 6: Drittsoftware und Integration

Übersicht 7: Lizenzumfang – Patent Grant bei OSS

Übersicht 8: Open Source und Integration in proprietäre Software

Einleitung

Von der Entwicklung bis zur Fertigstellung, dem Release und dem Einsatz einer Software im internationalen Kontext tauchen eine Reihe von juristischen Fragestellungen und Problemen auf, die im Vorwort ansatzweise dargestellt werden. Die Vertragsgestaltung spielt dabei nicht nur beim Vertrieb eine Rolle. Vielmehr ist zu berücksichtigen, dass bereits in der Entwicklungsphase – insbesondere beim Erwerb von Drittsoftware – Verträge zu schließen sind. Solche Verträge werfen vor allem im Open Source Bereich Probleme auf. Wenn eine Open Source Software in ein Produkt integriert wird, so können urheberrechtliche Beschränkungen den weiteren entgeltlichen Vertrieb behindern. Bereits hier zeigt sich die internationale Komponente, da Open Source Software vielfach in den U. S. A. entwickelt und von dort aus in Umlauf gebracht wird. Die Beurteilung der Vertriebsfähigkeit einer Software allein nach national geltenden Regelungen reicht daher nicht aus. Vielmehr ist sowohl bei der Entwicklung als auch beim späteren Vertrieb der Software das amerikanische Recht zu prüfen. Dies wird auch durch die im Ausgangsfall aufgeworfenen urheberrechtlichen und patentrechtlichen Probleme aufgezeigt, die nach Maßgabe des anwendbaren Rechts zu lösen sind und zu unterschiedlichen Ergebnissen – je nachdem, ob deutsches oder amerikanisches Recht anwendbar ist – kommen können.

Es werden daher die Fragestellungen untersucht, bei denen Softwarekomponenten – auch solche von Drittanbietern – entgeltlich und unentgeltlich in das Produkt integriert werden. Da in der Praxis der Bezug von Drittsoftware aus den U. S. A. eine große Rolle spielt, wird der Fokus der internationalen Betrachtung im Bereich der Entwicklung und der Verbindung der verschiedenen Softwarekomponenten auf der Untersuchung des amerikanischen Rechts, insbesondere des Copyright und Patent Laws, liegen.

Die gesetzlichen Vorschriften, die auf Computerprogramme anwendbar sind, werden anhand zahlreicher Fallbeispiele aus der Praxis dargestellt.

Im Fokus steht dabei die Frage, welche Komponenten Informatikerinnen und Informatiker der Forschungs- und Entwicklungsabteilungen in Deutschland wie auch in den U. S. A. in eine Software integrieren dürfen (insbesondere diejenigen, die sie nicht selbst entwickelt oder auch mittels Künstlicher Intelligenz entwickelt haben), wenn danach das Folgende gewährleistet werden soll:

- Die Software ist frei von Rechten Dritter.
- Auch keine freien Mitarbeitenden haben Rechte an der Software.
- Die Software soll entgeltlich ausgeliefert werden.
- Die Software soll in verschiedene Länder exportiert werden dürfen.

- Technische Erfindungen, die der Software zugrunde liegen, können zum Patent angemeldet werden; die Technik ist (noch) neu.

Dieses Lehrbuch widmet sich der rechtlichen Betrachtung von Software und ihrer Zusammensetzung von der Entwicklung bis hin zur Fertigstellung, insbesondere der Untersuchung einzelner Softwarekomponenten, die auf der Basis verschiedener – auch internationaler – Lizenzverträge zusammengesetzt und als ein einheitliches Softwareprodukt vertrieben werden sollen.

Die Fragen werden rechtsvergleichend unter Prüfung und Betrachtung des nationalen wie des amerikanischen Rechts, insbesondere des Urheber-, Patent- und Außenwirtschaftsrechts und der entsprechenden Praxis erörtert. Im Rahmen dessen werden ebenfalls die einschlägigen europäischen Richtlinien und Normen berücksichtigt.

Fallszenario – Ausgangsfall

Anhand des folgenden Fallszenarios soll die Komplexität der rechtlichen Analyse demonstriert werden. Es handelt sich dabei keineswegs um ein konstruiertes Beispiel. Vielmehr zeigt das Beispiel die rechtlichen Probleme der Entwicklung und des Softwarevertriebs auf und insbesondere, worauf zu achten ist, wenn eine Software – auch mittels KI – entwickelt wird und danach auf dem nationalen wie internationalen Markt verkauft und ggf. auch patentiert werden soll.

Die GoodWare4U AG ist eine Holding mit Sitz in Deutschland. Sie hat selbständige und unselbständige Niederlassungen in Deutschland und den U. S. A., dort die GoodWare4U, Corp. Sowohl in der AG als auch in der Niederlassung in Kalifornien, der Inc., befindet sich eine F&E-Abteilung, die sich vornehmlich mit der Entwicklung von Software beschäftigen.

Daneben ist der GoodWare4U Konzern weltweit auch ein Serviceunternehmen, das Software des jeweiligen Kunden individuell oder auch Software für den Kunden vollständig erstellt.

Wir haben also bisher vier Komponenten der Software. Komponenten umfassen dabei z. B. Module eines Programms, Funktionalitäten, Add-ons für zusätzliche Funktionalitäten, Programmbibliotheken sowie Treiber und Adapter:

1. von F&E programmierte Software,
2. von F&E in den U. S. A. programmierte Software,
3. Komponenten, die im Servicegeschäft der deutschen Niederlassung als Werk- oder Dienstleistung entstanden sind,

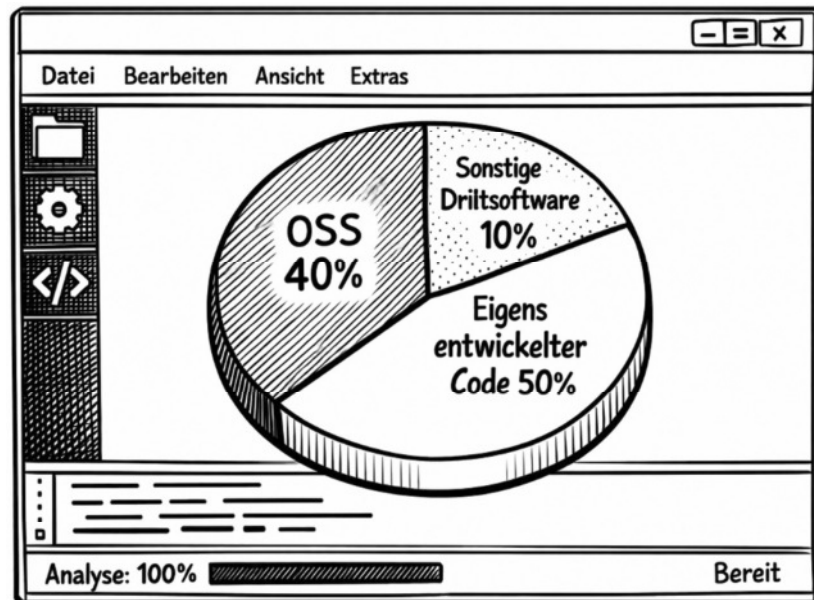
4. Komponenten, die im Servicegeschäft der U.S.-Niederlassung entstanden sind und in das Endprodukt der Software integriert werden sollen.

Nun zeigt die Praxis, dass nicht jede Softwarekomponente erst durch die Erstellungsleistungen, wie in den Ziffern 1–4 beschrieben, geschaffen wird. Vielmehr wird auch auf bereits existierende Routinen und Software zugegriffen und KI eingesetzt. Das bedeutet, dass auch Drittsoftware in das Produkt integriert wird. Hier sind daher die folgenden weiteren Komponenten denkbar:

5. entgeltlich erlangte Drittsoftware mit Quellprogramm (Open Source Software),
6. unentgeltlich erlangte Drittsoftware mit Quellprogramm (Open Source Software),
7. entgeltlich erlangte Drittsoftware ohne Quellprogramm,
8. unentgeltlich erlangte Drittsoftware ohne Quellprogramm (Freeware).

Ist das Computerprogramm – die neue Standardsoftware der GoodWare4U unter dem Namen „GoodWare2go“ – erst einmal technisch fertig gestellt, so stellt sich wiederum die Frage, wem welche Rechte an den jeweiligen Komponenten bzw. am Gesamtprogramm zustehen. Auch diese Frage soll unter der Betrachtung des deutschen wie des amerikanischen Rechts untersucht werden. Im Anschluss ist zu untersuchen, ob ein solches Produkt, bestehend aus den vorgenannten Komponenten, auch in jedem Land vertrieben und in jedes Land exportiert werden darf.

Zur Veranschaulichung der aufgeworfenen und zu untersuchenden Probleme soll die nachfolgende Abbildung dienen. Bei dieser handelt es sich um ein Computerprogramm. Die vorgestellten Fragestellungen, denen sich ein international operierendes IT-Unternehmen widmen muss, bevor eine Software für das Release freigegeben und dann vertrieben wird, werden dabei berücksichtigt. Die nachfolgende Abbildung enthält die im Ausgangsfall aufgeführten Problemstellungen. Im Zentrum ist ein Kreisdiagramm dargestellt, das die Zusammensetzung des Programms visualisieren soll. Der größte Anteil besteht aus eigens entwickeltem Code mit 50 %. Weitere 40 % entfallen auf Open-Source-Software (OSS), während sonstige Drittsoftware 10 % ausmacht.



Erzeugnis von ChatGPT vom 26.1.2026 nach Eingabe folgender Prompts: „Erstelle eine Abbildung von einem Computerprogramm! Sie soll zeigen, dass im Programm folgende Elemente enthalten sind: OSS zu 40 %, sonstige Drittsoftware zu 10 %, eigens entwickelter Code zu 50 %.“

Die Fragestellungen zum eigens entwickelten Code sowie zu sonstiger Drittsoftware werden im 1. Kapitel (nationales Recht) und im 2. Kapitel (US-amerikanisches Recht) behandelt, jeweils unter Berücksichtigung des Einsatzes von KI. Diejenigen zur Entwicklung mittels OSS werden im 3. Kapitel behandelt, einschließlich der Betrachtung der Regelungen der KI-VO zu OSS. Die Fragen der Zulässigkeit des Exports nach deutschem, europäischem und US-amerikanischem Recht werden im 4. Kapitel behandelt – auch in Abgrenzung zur KI-VO.

Zu den einzelnen Problemstellungen, welche unter diesen Ziffern bereits aufgezeigt werden, erfolgt eine vertiefte Auseinandersetzung. Dafür werden jeweils Einzelfälle aus der Praxis zur Verdeutlichung des Problems gebildet und Lösungswege hierzu vorgestellt. Anhand der Fallbeispiele werden zudem die für die Entwicklung, Erstellung und den sich anschließenden Vertrieb maßgeblichen Vorschriften unter Bezugnahme auf Rechtsprechung und Literatur dargestellt. Dieses Buch verfolgt daher auch den Zweck, ein

grundlegendes Verständnis für die Vielzahl der Detailprobleme – auch weit über den Ausgangsfall hinaus – zu schaffen.

Dieses Werk soll dem Rechtsvergleich dienen und die Probleme untersuchen, die durch eine Zusammenarbeit auf internationaler Basis mit der Geltung verschiedener Gesetze entstehen können, und Lösungsansätze praxisnah mit Checklisten anbieten. Das Werk enthält zahlreiche Fälle aus der IT-Praxis, anhand derer Rechtsprobleme aufgezeigt und die Konsequenzen für die Praxis dargestellt werden. Bei der Betrachtung des amerikanischen Rechts werden die Fälle, welche eingangs zur Erläuterung des nationalen Rechts dargestellt werden, auch wieder aufgegriffen. Die Fälle sind daher nummeriert, so dass auf sie auch an anderen Stellen wieder verwiesen werden kann. Wird der „Ausgangsfall“ zitiert, so ist damit das vorgenannte Fallszenario mit den verschiedenen aufgelisteten Problemen gemeint, deren Lösung sich dieses Buch widmet. Die Lösungswege sollen sodann aufzeigen, wie die unter den einzelnen Ziffern des Ausgangsfalls dargestellten Problemstellungen gelöst werden können.

Adressaten dieses Lehrbuches sind Studierende und Praktiker. Dabei richtet sich das Werk insbesondere an Studierende des Informations-, Lizenz- und IT-Rechts sowie der Informatik und Rechtsinformatik und an Unternehmensjuristinnen, juristische Berater, Teilnehmende des Fachanwaltslehrgangs IT-Recht sowie an Mitarbeitende und Selbständige, die an der Entwicklung von Computerprogrammen mitwirken.

Die Übersichten und Checklisten dienen zur schnellen Rekapitulation des Dargestellten und zur Erleichterung der praktischen Arbeit im Umgang mit und der rechtlichen Bewertung von Computerprogrammen und deren Bestandteilen.

I Die Softwareentwicklung nach deutschem Recht

1 Urheberrechtliche Betrachtung

1.1 Software als urheberrechtlich geschütztes Werk

Die Frage nach der Rechtsinhaberschaft an Teilen bzw. an der Software insgesamt bestimmt sich – wie in dem vorerwähnten Fallszenario – nach den Vorschriften des Urheberrechtsgesetzes, die vertraglich nur eingeschränkt abbedungen oder geändert werden können. Daher ist es erforderlich, zunächst die Vorschriften dieses Gesetzes zu betrachten und folgende Vorfragen zu klären:

- Genießt Software Urheberrechtsschutz?
- Welche Rechte stehen originär der Urheberin bzw. dem Urheber oder mehreren (Mit-)Urhebern zu?
- Gibt es Unterschiede, wenn ein freier Mitarbeiter im Gegensatz zu einer Angestellten Entwicklungs-, also Programmierleistungen erbringt?
- Wem stehen welche Rechte an der Software zu?

Das Urheberrecht basiert auf Art. 1 (Schutz der Menschenwürde), Art. 2 Abs. 1 (Recht auf freie Entfaltung der Persönlichkeit) sowie Art. 14 (Gewährleistung des Eigentums) des Grundgesetzes. Das bedeutet, es gibt demnach zwei Schutzbereiche: das Urheberpersönlichkeitsrecht, basierend auf Art. 1 und 2 Abs. 1 GG, sowie das Leistungsschutzrecht, basierend auf Art. 14 GG. Diese beiden Säulen des Urheberrechtsgesetzes zeigen sich schon im Titel: „Gesetz über Urheberrechte und verwandte Schutzrechte“.

Das Urheberpersönlichkeitsrecht zeigt seine Ausprägung u. a. in den folgenden Regelungen:

- Das Urheberrecht ist vererblich, § 28 Abs. 1 UrhG.
- Das Urheberrecht als Ganzes kann nicht übertragen werden, § 29 Abs. 1 UrhG; es können jedoch gemäß § 29 Abs. 2 UrhG Nutzungsrechte eingeräumt werden; diese bestimmen sich nach den §§ 31 ff. UrhG.
- Die Verwertungsrechte stehen einzig dem Urheber zu und umfassen das Recht, das Werk körperlich zu verwerten (für die IT insbesondere Vielfältigen, Verbreiten, öffentliche Zugänglichmachung, § 15 UrhG).

§ 11 UrhG zeigt die beiden Schutzgegenstände des Urheberrechtsgesetzes: Satz 1 betrifft das Urheberpersönlichkeitsrecht und Satz 2 das Recht auf Vergütung (Leistungsschutzrecht). Die §§ 12 und 15 ff. UrhG definieren sodann den Inhalt des Urheberpersönlichkeitsrechts.

Das Urheberrechtsgesetz vom 9. September 1965 (BGBl. I S. 1273) wurde zuletzt durch Artikel 25 des Gesetzes vom 23. Juni 2021 (BGBl. I S. 1858) geändert.

Der Urheberrechtsschutz für Computerprogramme hielt Einzug in das Urheberrechtsgesetz nach Umsetzung der Richtlinie 91/250/EWG des Europäischen Parlaments und des Rates vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen und wurde nach Umsetzung der Richtlinie 2009/24/EG des Europäischen Parlaments und des Rates vom 23. April 2009 über den Rechtsschutz von Computerprogrammen (ABl. L 111 S. 16, nachfolgend Computerprogramm-Richtlinie) ersetzt. Durch das zweite Gesetz zur Änderung des Urheberrechtsgesetzes vom 9. Juni 1993 wurden die Besonderen Bestimmungen für Computerprogramme in die neuen §§ 69a ff. UrhG eingeführt. Bis zur Novellierung des Urheberrechtsgesetzes im Jahr 1993 musste eine gewisse Schöpfungshöhe erreicht sein, um Computerprogramme als Werke charakterisieren zu können. Computerprogramme wurden daher gemäß § 2 Abs. 1 Nr. 1 UrhG als Sprachwerk – u. U. auch als Darstellung gemäß § 2 Abs. 1 Nr. 7 UrhG – geschützt. Für die Qualifizierung als Sprachwerk spricht zudem, dass ein Sprachwerk auch aus mathematischen Zahlen bestehen kann.¹ Voraussetzung nach der Rechtsprechung war aber, dass das Programm das Programmschaffen eines Durchschnittsprogrammierers „erheblich“ überstieg.² Die sogenannte „kleine Münze“ des Programmschaffens, also Werke von geringem schöpferischem Wert, waren damit noch vom Urheberrechtsschutz ausgeschlossen.³ Mit der Novellierung wurden die Computerprogramme – und somit ein weiteres Beispiel für ein Sprachwerk – in den Katalog der Werkarten unter den Oberbegriff der „Sprachwerke“ in § 2 Abs. 1 Nr. 1 UrhG explizit aufgenommen. Hierbei ist von einer niedrigen Schutzuntergrenze für Computerprogramme auszugehen, d. h. die Schöpfungs-/Gestaltungshöhe wird regelmäßig erreicht sein.⁴

Damit wurden die Innovationen der Hersteller von Computerprogrammen nun auch ausdrücklich unter den Schutz des Urheberrechtsgesetzes gestellt. Nach der Struktur des Gesetzes gibt es allgemeine und besondere Bestimmungen der §§ 69a ff. UrhG. Abschnitt 8 des ersten Teils des Gesetzes beinhaltet „Besondere Bestimmungen für Computerprogramme“. Diese in das Urheberrechtsgesetz eingefügten §§ 69a ff. UrhG gewähren Computerprogrammen, einschließlich des Entwurfsmaterials, urheberrechtlichen Schutz, sofern eine persönliche geistige Schöpfung vorliegt.

1 RGZ 121, 357, 358 – Rechentabellen; BGH, GRUR 1959, 251 – Einheitsfahrchein.

2 BGH, GRUR 1985, 1041, 1046 – Inkasso-Programm.

3 Hierzu *Loewenheim/Leistner*, in: *Schricker/Loewenheim*, § 2 Rn. 61 ff.

4 Zum Maßstab der Gestaltungshöhe der verschiedenen Werkarten s. *Nordemann*, in: *Schricker/Loewenheim*, § 9 Rn. 18 ff.

Eine Definition des Terminus Computerprogramm enthält das UrhG nicht, auch nicht in den „Besonderen Bestimmungen“. Es ist anzunehmen, dass der Gesetzgeber befürchtete, eine solche sei auf Grund der technischen Entwicklung schnell wieder hinfällig.

Die World Intellectual Property Organization definiert Computerprogramm wie folgt: „Computerprogramm ist eine Folge von Befehlen, die nach Aufnahme in einen maschinenlesbaren Träger fähig sind zu bewirken, dass eine Maschine mit informationsverarbeitenden Fähigkeiten eine bestimmte Funktion oder Aufgabe oder ein bestimmtes Ergebnis anzeigt, ausführt oder erzielt.“⁵

Nach dem Richtlinienvorschlag der EG-Kommission zum Rechtsschutz von Computerprogrammen aus dem Jahr 1989 ist ein Computerprogramm das „in jeder Form, Sprache und Notation oder in jedem Code gewählte Ausdrucksmittel für eine Folge von Befehlen, die dazu dient, einen Computer zur Ausführung einer bestimmten Aufgabe oder Funktion zu veranlassen.“ Hierbei sollen „alle menschlich wahrnehmbaren und maschinenlesbaren Formen von Programmen“ erfasst sein, aus denen „das Programm (...) entwickelt wurde oder entwickelt werden kann (...).“

Demgegenüber werden Material wie Bedienungsanleitungen oder Wartungshandbücher nicht als Teil oder „Ausdrucksformen des Computerprogramms angesehen.“⁶

Die DIN 44300 definierte Software als „eine nach den Regeln der verwendeten Sprache festgelegte syntaktische Einheit aus Anweisungen und Vereinbarungen, welche die zur Lösung einer Aufgabe notwendigen Elemente umfasst“.

Damit ist Software nicht allein der Quelltext, denn dieser ist nur Text, ohne dass er eine Maschine steuert.

Abgelöst wurde diese DIN durch die ISO/IEC 2382:2015, die Software kategorisiert. Die nachfolgende Definition von Computerprogrammen und Software enthält die Norm ISO/IEC/IEEE 24765:2017.

3.726 – computer program: 1. combination of computer instructions and data definitions that enable computer hardware to perform computational or control functions 2. syntactic unit that conforms to the rules of a particular programming language and that is composed of declarations and statements or instructions needed for a certain function, task, or problem solution (...)

5 Mustervorschriften, GRUR Int. 1978, 286, 290 ff.

6 Art. 1 Abs. 1 des Vorschlags für eine Richtlinie des Rates über den Rechtsschutz von Computerprogrammen, ABl. 89/C/91/05, 09.

3.3783 – software: 1. computer programs, procedures and possibly associated documentation and data pertaining to the operation of a computer system (...) 2. all or part of the programs, procedures, rules, and associated documentation of an information processing (...)

3. program or set of programs used to run a computer (...)

Beide Definitionen sehen im Computerprogramm ein Ausdrucksmittel (entsprechend dem Sprachwerk nach § 2 Abs. 1 Nr. 1 UrhG) als Folge von Befehlen zur Erreichung einer bestimmten Funktion oder Lösung einer Aufgabe.

Dabei wird der Begriff der Software nicht als Synonym für Computerprogramm verstanden; denn der Begriff „Software“ umfasst auch digitale Texte, Soundfiles, Grafiken oder sonstige Daten und Datenbanken, bei denen es sich nicht um Computerprogramme im technischen Sinne handelt.⁷ Ein weiterer Unterschied besteht darin, dass für Computerprogramme Steuerbefehle notwendig sind.⁸ Die Definition der WIPO hat dies ebenfalls demonstriert („Folge von Befehlen“). Zusammenfassend kann man dies auf die Kurzformel bringen: Software löst Probleme! Und sie ist überall: im Fernseher, im Tablet, in der Smartwatch, in der Spülmaschine, im Fotoapparat ...

Man spricht dann auch von embedded Software, wenn diese in Produkten enthalten – also eingebettet – ist.⁹

Es ist auch zu erwähnen, dass der Terminus des Computerprogramms keineswegs in allen – auf Computerprogramme anwendbaren – gesetzlichen Regelungen auftaucht. In den einzelnen Gesetzestexten taucht die folgende Terminologie auf:

- § 2 Abs. 1 Nr. 1 UrhG: „Computerprogramme“
- § 69e Abs. 1 UrhG: „Code“
- § 1 Abs. 3 Nr. 3 PatG: „Programme für Datenverarbeitungsanlagen“
- § 312g Abs. 2 Nr. 6 BGB: „Computersoftware“:
- § 327 Abs. 6 Nr. 6, 327e Abs. 2, S. 3 und 4, BGB: „Software“

In der Praxis – wie in diesem Buch – werden Begriffe wie Software, Programm, IT-System und Produkt synonym für Computerprogramm verwendet, wobei inhaltlich nur das unter dem jeweiligen Begriff verstanden wer-

⁷ Die SEVOCAB Datenbank enthält Definitionen aus der genannten Norm, [//pascal.computer.org/sev_display/index.action](http://pascal.computer.org/sev_display/index.action).

⁸ OLG Hamburg, CR 1998, 332, 333 f. – Computerspielerergänzung; OLG Rostock, CR 2007, 737 f.

⁹ Zur physischen Verbindung und Interaktion zwischen Soft- und Hardware s. ErwGrd 10 der Computerprogramm-Richtlinie.

den soll, was das Urheberrecht unter dem Begriff des Computerprogramms versteht.¹⁰

Es ist dann weiter zu differenzieren, welchen Zwecken die Software dienen soll und welche Vertragsart gewählt wird, auf deren Grundlage Rechte eingeräumt werden sollen.

Als Softwarearten gibt es grundsätzlich Standard- sowie Individualsoftware. Individualsoftware wird grundsätzlich auf den Erwerber „zugeschnitten“ oder zumindest auf seine Bedürfnisse hin angepasst (Customizing). Es gibt dabei verschiedene Methoden, Software als Individualsoftware zu entwickeln.¹¹ Dabei sollen hier zwei Methoden erläutert werden, die die Entwicklung in einem Kundenprojekt als Auftragsentwicklung betreffen.

Einmal gibt es die klassische Wasserfallmethode mit den aufeinanderfolgenden Phasen: Planung, Analyse, Design, Implementierung, Abnahme und Echtbetrieb. Da die Phasen nacheinander durchlaufen und abgearbeitet werden, muss immer wieder auch nach den vorherigen Phasen geschaut werden, damit nach Abschluss der Entwicklung auch alles noch aktuell ist. Deshalb werden in Praxisprojekten auch die für die jeweils weitere Entwicklung dienlichen Dokumente der Leistungsbeschreibung, die Pflichtenhefte, als begleitende Anleitung für das konkrete „Wie“ der Zielerreichung von den Parteien fortgeschrieben.

Bei der agilen Programmierung durch Scrum¹² wird im Rahmen der Entwicklung iterativ vorgegangen. Schon die Terminologie bei dieser Art der Softwareentwicklung zeigt Unterschiede zum Wasserfall. Beim klassischen Wasserfall werden regelmäßig nach Projektfortschritt Meilensteine eingelegt, verbunden mit dem Erreichen von Zwischenzielen, etwa dem Abschluss der Programmierung einer Teilfunktionalität des Computerprogramms. Beim Scrum dagegen werden Sprints eingelegt. D.h. nicht, dass bei der Scrum-Methode im Ergebnis alles schneller geht, es gibt aber Meetings in kürzeren Zeitabständen. Die Projektmitarbeiter beider Teams – der Auftraggeberin und der Auftragnehmerin – tauschen sich auch regelmäßig untereinander aus, es werden aber in kürzeren Zeitabständen Meetings abgehalten und Zwischentappen auf dem Weg zum Ziel in kleinere Abschnitte unterteilt, um den Projektfortschritt zu besprechen.

10 Wenn daher der Begriff „Software“ oder „Programm“ verwendet wird, dann soll dies als Synonym für „Computerprogramm“ verstanden werden. Dies gilt für alle Kapitel dieses Buches.

11 Eine Übersicht hierzu bietet auch *Sarre*, in: Auer-Reinsdorf/Conrad, § 1 Rn. 23 ff.

12 Bei Scrum handelt es sich um eine Technik der agilen Programmierung.