§ 2 Software als Wirtschaftsgut

Gemeinhin wird angenommen, dass Software besondere Eigenschaften aufweist, die sich maßgeblich auf Herstellung, Vertrieb und letztlich auch auf die Industriestruktur und den Softwaremarkt auswirken.[4] Diese Eigenschaften waren auch für die Entwicklung der urheberrechtlichen Bestimmungen für Software prägend.[5] Aus diesem Grund soll im Folgenden auf die wichtigsten technischen und wirtschaftlichen Aspekte eingegangen werden.

I. Begriffsdefinition „Software“

Software ist ein Sammelbegriff für ausführbare Computerprogramme und die dazugehörigen Daten.[6] Sie dient dazu, mithilfe eines Computers Aufgaben zu erledigen. Ihren Kern bilden Computerprogramme, aufeinanderfolgende Instruktionen, die den Regeln einer Programmiersprache genügen und als syntaktische Einheit dazu geeignet sind, eine bestimmte Funktionalität zur Verfügung zu stellen.[7] In einer Software sind Computerprogramme untrennbar mit zur Ausführung erforderlichen Daten verknüpft. Dabei handelt es sich etwa um Grafiken, Texte und Töne, die Teil der grafischen Benutzeroberfläche (engl. „Graphical User Interface“, GUI) sind, über die ein Nutzer mit den Programmen interagiert. Auch Dokumentation kann in digitaler Form Teil dieser Daten sein und ist dem Nutzer häufig als interaktive Hilfedatei während der Softwarebenutzung zugänglich. Der Begriff Software

[4] *Buxmann/Diefenbach/Hess*, Softwareindustrie, 3.
[5] BT-Drucks. 12/4022, S. 7 f.
[6] *Lassmann*, Wirtschaftsinformatik, 127. Vgl. auch IEEE Std 610.12: „Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.“
[7] Vgl. ISO/IEC 2382-1:1993: „A syntactic unit that conforms to the rules of a particular programming language and that is composed of declarations and statements or instructions needed to solve a certain function, task, or problem.“

J. Becher, *Der Sekundärmarkt für Software,* Ökonomische Analyse des Rechts,
DOI 10.1007/978-3-658-08849-1_2, © Springer Fachmedien Wiesbaden 2015

wurde in diesem Sinne bereits erstmals 1958 von dem Statistiker *John W. Tukey* verwendet.[8] Er erlangte jedoch erst in den 1970er Jahren wesentliche Bedeutung als Unternehmen wie *Microsoft* und *SAP* gegründet wurden, die ausschließlich Software herstellten. Vormals waren die Funktionalitäten von Computern auf deren Schalttafeln überwiegend „vorverdrahtet", also unveränderlich, wie dies auch heute noch bei Taschenrechnern und Spielautomaten der Fall ist. Eine Software hat die Aufgabe, eine strukturell unveränderliche Hardware für unterschiedliche Verwendungen nutzbar zu machen.

Einige Autoren bevorzugen im Kontext der urheberrechtlichen Diskussion den Begriff Computerprogramm, der in der Tat häufig synonym zum Begriff Software verwendet wird. Ihrer Ansicht nach handelt es sich bei Computerprogrammen um eine Teilmenge der Software.[9] Software könne wiederum lose definiert werden als „alles, was keine Hardware darstellt".[10] Sie umfasse daher auch digitale Medien wie Texte, Grafiken und Audiodateien, die für sich genommen keine Computerprogramme im Sinne der § 69a ff. UrhG enthielten und nicht dem urheberrechtlichen Themenkomplex der gebrauchten Software zuzuordnen seien. Diese Sichtweise basiert auf der uneingeschränkten, aus dem Wortsinne abgeleiteten Definition von Software (leicht veränderbare, „weiche" Komponenten) als Gegenstück zu Hardware (unveränderliche, physische Komponenten). Ihr zufolge könnten als Software auch funktionslose Datenbestände verstanden werden, die auf einer Hardware gespeichert sind und erst mit der Verarbeitung durch ein Computerprogramm nutzbar werden. Diese Definition ist aber sowohl im Sprachgebrauch als auch in der technischen und betriebswirtschaftlichen Literatur ungebräuchlich. Dies ist schon daran zu erkennen, dass bedeutende, anderweitige Begrifflichkeiten auf dem Verständnis von Software im Sinne eines *Softwareprodukts* aufbauen, also eines Computerprogramms und dessen eng verbundener Ressourcen. So wird zum Beispiel *Systemsoftware*,

[8] *Tukey*, Am. Math. Mon. 65-1 (1958), 1, 2: „Today the 'software' comprising the carefully planned interpretive routines, compilers, and other aspects of automative programming are at least as important to the modern electronic calculator as its 'hardware' of tubes, transistors, wires, tapes and the like."

[9] *Seitz*, Gebrauchte Software, 7 ff.

[10] *Seitz*, Gebrauchte Software, 9.

die für die grundsätzliche Funktion eines Computers benötigt wird, von der *Anwendungssoftware* abgegrenzt, die dem Benutzer die Ausführung bestimmter Aufgaben ermöglicht.[11] Eine künstliche Übertragung solcher Begriffspaare auf den Programmbegriff, wie sie *Seitz* vornimmt, indem er von „System-Programmen“ und „Anwender-Programmen“ spricht,[12] soll in dieser Ausarbeitung aber vermieden werden.

Ferner wird noch *Individualsoftware* von *Standardsoftware* unterschieden.[13] Bei Individualsoftware handelt es sich um Software, deren Entwicklung ein Unternehmen bei einem Hersteller in Auftrag gibt. Sie ist genau auf die Strukturen, Prozesse und Anforderungen eines bestimmten Unternehmens zugeschnitten. Derartig maßgeschneiderte Software eignet sich normalerweise nicht mehr für einen Einsatz in anderen Unternehmen. Urheberrechtsfragen treten deswegen bei Individualsoftware oft völlig in den Hintergrund. Bei Standardsoftware, im Englischen auch „packaged software“ oder „commercial off-the-shelf“ (COTS) genannt, handelt es sich um vorgefertigte Softwareprodukte, die für den Massenmarkt vorgesehen sind. Sie deckt einen klar definierten Anwendungsbereich ab, kann heutzutage aber häufig über Zusatzpakete (sog. *Plug-ins*) und Schnittstellen noch angepasst oder erweitert werden.[14] Die Frage nach einem gebrauchten Verkauf einer Software stellt sich nur für Standardsoftware und nicht für Individualsoftware oder individuell entwickelte Plug-ins. Gegenstand der folgenden Ausarbeitung ist stets Standardsoftware, der Einfachheit halber wird aber nur der Begriff Software verwendet. Keine Einschränkung erfolgt dagegen in Hinblick auf System- und Anwendungssoftware. Beide Gattungen werden zwar als Standardsoftware vermarktet, weisen aber bezüglich ihrer ökonomischen Charakteristika bedeutende Unterschiede auf, die für diese Untersuchung von Bedeutung sind.

---

[11] *Mertens* et al., Wirtschaftsinformatik, 21.
[12] *Seitz*, Gebrauchte Software, 9.
[13] *Buxmann/Diefenbach/Hess*, Softwareindustrie, 5 ff.; *Mertens* et al., Wirtschaftsinformatik, 21 f.
[14] *Mertens* et al., Wirtschaftsinformatik, 33 f.

II. Herstellung, Nutzung und Wartung

Die Herstellung von Software wird als Softwareentwicklung (*Software Engineering*) bezeichnet.[15] Es handelt sich dabei um einen komplexen, arbeitsteiligen Prozess, der grundsätzlich mit der ingenieursmäßigen Entwicklung anderer technischer Produkte vergleichbar ist.[16] Der Entwicklungsprozess gliedert sich traditionell in mehrere inhaltlich und zeitlich abgrenzbare Phasen, insbesondere Planung, Entwurf, Programmierung und Test, deren strikte Trennung in den letzten Jahren aber zunehmend zugunsten flexibler Verfahrensweisen aufgegeben wird, die unter dem Oberbegriff *agile* Softwareentwicklung zusammengefasst werden.[17] Kerntätigkeit der beteiligten Softwareentwickler ist die Erstellung, Erweiterung und Anpassung des Quellcodes, dem von Menschen lesbaren Text eines Computerprogramms. Dieser Code kann aber erst dann auf einem Computer ausgeführt werden, wenn er in einen binären Maschinencode übersetzt worden ist. Dazu wird der Quellcode durch einen *Compiler* übersetzt und in ausführbarer Form gespeichert.[18] Eine umgekehrte Übersetzung des Maschinencodes in einen Quellcode ist nur eingeschränkt möglich und in der Regel mit prohibitivem Aufwand verbunden. Das vermarktete Softwareprodukt wird regelmäßig nur in Form der durch den Menschen nicht lesbaren Maschinensprache vertrieben. Für den Käufer präsentiert sich die erworbene Software als reiner Datenbestand, dessen ausführbare Programmelemente zwar anhand ihres Erscheinungsbildes identifizierbar, aber nicht verständlich sind. Die zugrunde liegenden Funktionsweisen und die Qualität der enthaltenen Programme sind nicht erkenn- oder bewertbar. Eine im Binärcode vorliegende Software kann regelmäßig weder angepasst noch erweitert werden.

Um die Software nutzen zu können, muss sie gewöhnlicherweise auf dem

[15] *Mertens* et al., Wirtschaftsinformatik, 153.
[16] *Ludewig/Lichter*, Software Engineering, 34.
[17] *Buxmann/Diefenbach/Hess*, Softwareindustrie, 130 ff.
[18] In einigen Fällen wird der Maschinencode durch einen Interpreter zur Laufzeit, also vor und während der Ausführung einer Software, erstellt. Dieses Verfahren ist aufgrund der geringeren Geschwindigkeit sowie der Einseh- und Kopierbarkeit des Quellcodes bei Standardsoftware unüblich.

Computer des Nutzers mithilfe eines beigefügten Installationsprogramms installiert werden. Dies ist häufig schon deshalb erforderlich, da die Software vor der Auslieferung komprimiert wird und das Installationsprogramm die Dekomprimierung vornimmt. Ferner kann der Nutzer im Zuge der Installation die zu installierende Software üblicherweise noch konfigurieren und die zu installierenden Softwaremodule auswählen. Auch wenn die Software aus dem Internet heruntergeladen wurde, ist regelmäßig eine Installation erforderlich. Im Anschluss an die Installation befindet sich die Software in ausführbarem Zustand auf dem Computer. Das Ausgangsmedium, also z.B. die DVD oder die aus dem Internet heruntergeladene Installationsdatei, wird für die Nutzung der Software dann üblicherweise nicht mehr benötigt.[19] Die nun auf der Festplatte des Computers befindliche installierte Form der Software wurde aber im Zuge der Installation auf die Umgebungseigenschaften des Computers zugeschnitten und eignet sich nicht für die Verwendung auf einem anderen Computer. Möchte ein Zweiterwerber die Software auf dem eigenen Computer nutzen, so benötigt er zwingend den vom Hersteller übermittelten installierbaren Ausgangsdatenbestand.

Aufgrund der Distribution im Binärformat ist der Anwender hinsichtlich der Wartung der Software auf den Hersteller angewiesen, der dem Anwender Verbesserungen seiner Software im Form sog. *Updates* liefert, mit denen der Nutzer den installierten Softwaredatenbestand aktualisieren kann. Zwar unterliegt Software als immaterielles Gut keinerlei Abnutzung oder Verschleiß, erfordert aufgrund der sog. *Softwarealterung* aber regelmäßige Wartungsanpassungen und hat letztlich eine begrenzte Lebensdauer.[20] Aufgrund der hohen Komplexität des ihr zugrunde liegenden Quellcodes treten im Laufe des Nutzungszeitraums stets Fehler auf, die korrigiert werden müssen. Besondere Bedeutung kommt dabei Sicherheitslücken zu, die dazu führen können, dass ein vernetzter Rechner verwundbar für Angriffe über das Internet wird. Zudem verändert sich im Laufe der Zeit die technische Umgebung, in der die Software eingesetzt wird. Vor dem Aufkommen des Internets wurden

[19] In einigen Fällen muss aus Kopierschutzgründen der vom Hersteller gelieferte Datenträger während der Nutzung in einem Laufwerk des Computers eingelegt sein.

[20] Vgl. allgemein *Parnas*, ICSE Conf. 16 (1994), 279.

mehrere Updates gebündelt und den Nutzern als sog. *Service Packs* auf Datenträgern zur Verfügung gestellt. Mit zunehmender Verbreitung des Internets wurde es jedoch erforderlich, dass wichtige Sicherheitsupdates einzeln und kurzfristig über das Internet verfügbar gemacht oder sogar automatisch installiert werden. Stellt der Hersteller die Versorgung mit Updates für ein Softwareprodukt ein, so kann es binnen kurzer Zeit für den produktiven Einsatz im Unternehmen unbrauchbar werden. Die Software muss ersetzt werden. Man spricht dann vom *End-of-life* dieses Softwareprodukts. Der Zeitablauf von Markteinführung einer Software bis zum End-of-life wird daher auch als *Softwarelebenszyklus* bezeichnet.[21]

III. Typische Vertriebsmodelle

Hersteller greifen auf verschiedene Vertriebsmodelle zurück, die häufig auch kombiniert werden (sog. *Channel Mix* bzw. *Multi-Channel-Vertrieb*), um ihre Softwareprodukte zu vermarkten.[22] Die Vertriebsarten bei Standardsoftware lassen sich grob den folgenden drei Vertriebsmodellen zuordnen.

1. Der gestufte Vertrieb

Hersteller von Software vermarkten ihre Produkte in der Regel weltweit und greifen dafür auf ein Netz von Vertragshändlern bzw. Vertriebspartnern zurück.[23] Der Softwarehersteller entwickelt die Software und beliefert die Vertragshändler mit Kopien der Software auf Datenträgern, gegebenenfalls in einer Verpackung, die auch ein gedrucktes Handbuch enthält. Der Vertragshändler vertreibt diese Softwarepakete anschließend an die Endnutzer. Das Verhältnis zwischen Softwarehersteller und Vertragshändler regelt ein Softwarevertriebsvertrag. Je nach Natur des Vertrages handelt der Vertragshändler auf eigene Rechnung als Großhändler oder auf Rechnung des Softwareherstellers als Handelsvertreter.[24] Agiert der Vertragshändler als

[21] *Mertens* et al., Wirtschaftsinformatik, 141.
[22] *Buxmann/Diefenbach/Hess*, Softwareindustrie, 91 f.
[23] *Buxmann/Diefenbach/Hess*, Softwareindustrie, 89 f.

Handelsvertreter, bedeutet dies für den Softwarehersteller ein höheres Risiko, da er zum direkten Vertragspartner des Erwerbers wird und ihn Gewährleistungspflichten direkt treffen. Allerdings erhoffen sich Softwarehersteller, die einen handelsvertreterrechtlichen Vertrieb bevorzugen, ein höheres Maß an Kontrolle über die eingeräumten Nutzungsrechte.[25]

Wird der Verkauf des Datenträgers durch einen Vertragshändler übernommen, versuchen Softwarehersteller häufig, die Einräumung von Nutzungsrechten von dem zwischen Händler und Käufer geschlossenen Verkaufsgeschäft zu entkoppeln. Die Käufer sollen bei Inbetriebnahme der Software noch einen „Lizenzvertrag“ mit dem Hersteller abschließen, um die benötigten Nutzungsrechte für die erworbene Software zu erlangen. Dazu soll sich der Käufer entweder schon durch das Auspacken der Software den Bedingungen des Rechtsinhabers unterwerfen (sog. *Shrink-Wrap-Licensing* oder *Schutzhüllenvertrag*) oder diese durch das Bestätigen einer entsprechenden Abfrage bei der Installation der Software akzeptieren (sog. *Enter-Vereinbarungen*). Auf diese Weise soll den unabdingbaren Rechten des Käufers, die sich im Fall einer dauerhaften Überlassung aus dem Kaufrecht und Bestimmungen des Urheberrechts ergeben, ausgewichen werden.[26] Nach h.M. haben Shrink-Wrap- und Enter-Vereinbarungen allerdings keinen Einfluss auf das Vertragsverhältnis zwischen Käufer und Vertragshändler. Ein einfaches Nutzungsrecht ist zudem nach Maßgabe des § 69d UrhG bereits unabdingbarer Teil des Kaufvertrages.[27] Zusätzliche Beschränkungen und Bestimmungen, die sich aus Shrink-Wrap- oder Enter-Vereinbarungen des Herstellers ergeben sollen, sind für Käufer nicht bindend.[28]

[24] *Redeker*, IT-Recht, Rn. 818 ff.
[25] *Sahin/Haines*, CR 2005, 241.
[26] *Seitz*, Gebrauchte Software, 30; *Redeker*, IT-Recht, Rn. 577; *Hoeren/Foersterling*, MMR 2012, 642, 645.
[27] Vgl. S. 166.
[28] *Marly*, Softwareüberlassungsverträge, 218-225.

2. Direkt- und Onlinevertrieb

Im Direktvertrieb ist der Softwarekäufer direkter Kaufvertragspartner des Softwareherstellers, der ihn direkt anspricht, berät und seine Produkte selbst liefert. Da persönliche Ansprache und Beratung einen hohen Vertriebsaufwand verursachen, kommt diesem Vertriebsmodell traditionell nur im Großkundengeschäft eine hohe Bedeutung zu. Beim Softwarehersteller betreuen speziell geschulte Vertriebsmitarbeiter, die sog. *Key-Account-Manager*, Stammkunden über einen längeren Zeitraum und bauen Kundenbeziehungen auf.[29] Je nach Größe des Kundenunternehmens und des IT-Budgets bemühen sich die Key-Account-Manager verstärkt darum, durch *cross-selling* neue Produkte beim Kunden zu platzieren oder durch *up-selling* neue Einsatzmöglichkeiten für bestehende Produkte zu finden, um zusätzliche Lizenzen verkaufen zu können. Hersteller wie *Microsoft* bieten eine diversifizierte Palette untereinander kompatibler Softwareprodukte an und versuchen, Produkte unterschiedlicher Gattung im Unternehmen zu verankern. Gelingt dies, kann der Softwarehersteller mit kontinuierlichen Einnahmen über einen Zeitraum mehrerer Jahre rechnen, da der Austausch des gesamten Produktsystems eines Herstellers für den Kunden mit enormem Aufwand verbunden sein kann.

Das *World Wide Web* hat die Kosten für die unmittelbare Kundenansprache und -kommunikation erheblich reduziert. Die Softwarehersteller bedienen daher zunehmend auch kleine und mittlere Unternehmen (KMU) sowie Privatanwender direkt. Auf diesem Weg können die Hersteller eine Beteiligung der Vertragshändler vermeiden und höhere Gewinnmargen erzielen. Statt der persönlichen Ansprache der Kunden über Key-Account-Manager erfolgt die Kundenkommunikation regelmäßig nur über die Website des Herstellers, sodass auch von einem *Onlinevertrieb* gesprochen werden kann. Potenzielle Kunden müssen selbst die benötigten Informationen abrufen und geeignete Produkte auswählen. Eine individuelle Beratung erfolgt nicht. Für sie ergibt sich jedoch der Vorteil, dass das online erworbene Produkt sofort

[29] *Buxmann/Diefenbach/Hess*, Softwareindustrie, 92 f.

bezahlt, heruntergeladen und genutzt werden kann. Dem Softwarehersteller bietet der Onlinevertrieb den Vorteil, dass er im direkten Vertragsverhältnis zum Nutzer steht und seine Nutzungsbedingungen in Form Allgemeiner Geschäftsbedingungen zum Vertragsbestandteil machen kann. Im Gegensatz zu den Shrink-Wrap- und Enter-Vereinbarungen gestaltet sich die wirksame Einbeziehung solcher Bedingungen hier unproblematisch.[30] Diese deutlich bessere Stellung des Softwareherstellers im Vergleich zum gestuften Vertrieb dürfte allerdings den wenigsten Käufern bewusst sein.

3. Miete und Leasing

Sowohl im direkten als auch im indirekten Vertrieb erwirbt ein Unternehmen regelmäßig eine dauerhafte Nutzungsmöglichkeit und einen selbstständig nutzbaren Datenbestand gegen ein einmaliges Entgelt. Der Softwarekauf bindet jedoch aus Sicht der Unternehmen viel Kapital und belastet die Liquidität, da die zukünftige Nutzung der erworbenen Software bereits vorab finanziert werden muss. Die steuerliche Behandlung kann diesen Effekt noch verstärken, wenn die gesetzlichen Abschreibungsmöglichkeiten nicht der tatsächlichen Nutzung im Unternehmen entsprechen. Dieser Nachteil soll durch mietähnliche Vertriebsmodelle, bei denen befristete Nutzungsrechte gegen fortlaufende Zahlungen eingeräumt werden, vermieden werden. Aus betriebswirtschaftlicher Sicht kommt es zum sog. *Pay-as-you-earn*-Effekt, wenn die Erträge aus der Nutzung der Software die fortlaufenden Miet- bzw. Leasingzahlungen decken. Zudem können die Zahlungen in der Regel sofort steuerlich geltend gemacht werden.

Eine solche Gebrauchsüberlassung auf Zeit kann entweder auf dem Weg der *Softwaremiete* oder des *Softwareleasings* erfolgen. Im Fall der Miete treffen den Vermieter umfassende Gewährleistungspflichten. Im Gegensatz zum Softwarekauf bestehen diese Gewährleistungspflichten zudem nicht nur befristet, sondern während der gesamten Mietdauer.[31] Ist der Software-

[30] *Marly*, Softwareüberlassungsverträge, 217; *Schumacher*, CR 2000, 641, 641.
[31] Vgl. § 536 BGB.

hersteller nicht selbst Vermieter, wird in vielen Fällen das *Softwareleasing* bevorzugt. Dabei handelt es sich im Gegensatz zur Miete nicht um einen gesetzlich geregelten Vertragstypus. Wie bei der Miete entsteht ein Dauerschuldverhältnis zwischen Leasinggeber und -nehmer. Kennzeichnend für das Leasing ist jedoch, dass der Leasingnehmer auf seine Gewährleistungsrechte dem Leasinggeber gegenüber verzichtet, der im Gegenzug dafür die eigenen Rechte gegenüber dem Softwarehersteller an den Leasingnehmer abtritt.[32]

4. Cloud Computing und Software-as-a-Service

Aufgrund der zunehmenden Vernetzung technischer Infrastruktur werden die klassischen Vertriebsmodelle bei Software und Hardware, also Kauf, Miete und Leasing, zunehmend durch neuartige vernetzte Vertriebsformen ergänzt, die unter dem Oberbegriff *Cloud Computing* zusammengefasst werden.[33] Beim Cloud Computing werden Software oder Hardware dem Kundenunternehmen nicht vorübergehend oder dauerhaft überlassen, sondern in Form einer abstrahierten Nutzungsmöglichkeit über Netzwerke, vor allem das Internet, bereitgestellt. Wird Software auf diesem Weg angeboten, spricht man von *Software-as-a-Service* (*SaaS*). Die Interaktion von Nutzer und Software erfolgt dabei überwiegend mit einem gewöhnlichen Internetbrowser.[34] SaaS-Anbieter betreiben und warten dabei selbst die dafür notwendige Soft- und Hardwareinfrastruktur, deren technische und räumliche Ausgestaltung für den Nutzer undurchsichtig ist. Eine Installation durch den Nutzer ist regelmäßig nicht erforderlich. Es entfällt daher die Notwendigkeit der bei der Miete erforderlichen Kontrolle der Nutzungsdauer, die durch Anbieter und Kunden schwierig zu realisieren ist.[35] Je nach Geschäftsmodell zahlen die Nutzer ent-

[32] *Marly*, Softwareüberlassungsverträge, 102.

[33] *Sjurts*, Medienwirtschaft, 79. Der Begriff geht höchstwahrscheinlich auf die Darstellung von Netzwerken in Form einer Wolke in standardisierten Netzwerkdiagrammen zurück. Vgl. `http://en.wikipedia.org/w/index.php?title=Cloud_computing\&oldid=519308086` (Abruf: 23.10.2012).

[34] *Sjurts*, Medienwirtschaft, 565.

[35] In großen Unternehmen ist häufig von zentraler Stelle aus kaum feststellbar, welche Softwareprodukte auf den Computern der Mitarbeiter installiert sind. Die bloße Rückgabe der Datenträger am Ende der Mietdauer ist somit nicht ausreichend.

http://www.springer.com/978-3-658-08848-4

Der Sekundärmarkt für Software
Eine ökonomische Analyse des urheberrechtlichen Erschöpfungsprinzips
Becher, J.
2015, XIX, 250 S. 2 Abb., Softcover
ISBN: 978-3-658-08848-4