

Nachdem wir im ersten Kapitel einige grundlegende Spielregeln der Softwareindustrie dargestellt haben, wenden wir uns im Folgenden den ökonomischen Prinzipien dieser Branche zu. Ausgangspunkt sind die Eigenschaften digitaler Güter (Abschn. 2.1). Im nächsten Schritt untersuchen wir Netzeffekte auf Softwaremärkten (Abschn. 2.2) und das damit eng verbundene Standardisierungsproblem (Abschn. 2.3). Darüber hinaus beschäftigen wir uns mit den Aspekten der Transaktionskosten- (Abschn. 2.4) und Principal-Agent-Theorie (Abschn. 2.5), die für die Softwareindustrie von besonderer Bedeutung sind.

---

## 2.1 Eigenschaften digitaler Güter

Zu Beginn des ersten Kapitels haben wir bereits einige ökonomische Eigenschaften von digitalen Gütern und Softwareprodukten vorgestellt. Diese Überlegungen sollen nun vertieft werden. Ein wesentliches Merkmal digitaler Güter besteht darin, dass die Erstellung der First Copy in der Regel zu hohen Kosten führt, die Reproduktion jedoch zu sehr geringen variablen Kosten möglich ist. Sehen wir uns diesen Zusammenhang für die Softwareindustrie einmal näher an. Die Entwicklung einer Software (genauer gesagt: des Quellcodes als First Copy) erfordert häufig erst einmal hohe Investitionen. Dabei ist jedoch keineswegs gewährleistet, dass ein Entwicklungsvorhaben auch tatsächlich zum Erfolg führt. Ein Individualsoftwareanbieter ist daher geneigt, die Finanzierung der Entwicklungskosten sowie das Risiko ganz oder zumindest teilweise auf seinen Auftraggeber abzuwälzen (siehe hierzu auch Abschn. 2.5). Einem Standardsoftwareanbieter bietet sich diese Option hingegen nicht; er trägt das komplette Risiko, kann im Erfolgsfall aber auch überproportional profitieren. Bei den Entwicklungskosten handelt es sich darüber hinaus um so genannte „sunk costs“, d. h. um Kosten, die unwiderruflich entstanden sind, nicht mehr beeinflusst werden können und damit auch nicht mehr entscheidungsrelevant sind.

Bereits an dieser Stelle wollen wir jedoch schon darauf hinweisen, dass die Theorie digitaler Güter die Realität in ihren Annahmen hinsichtlich der gegen Null tendierenden variablen Kosten stark und in vielen Fällen wohl auch zu stark vereinfacht. So gehen die variablen Kosten lediglich für Softwarelizenzen gegen Null. Nicht vernachlässigbar sind diese Kosten jedoch für die Dienstleistungen, die rund um die Softwareprodukte angeboten werden, beispielsweise für Beratung, Wartung und Support. Wir werden insbesondere im dritten Kapitel auf diese Zusammenhänge zurückkommen.

Eine weitere wichtige Eigenschaft digitaler Güter besteht darin, dass sie einfach und ohne Qualitätsverluste kopiert werden können. Aus dieser einfachen Kopierbarkeit resultieren unter anderem die niedrigen Kosten der Reproduktion. Kopien digitaler Güter werden auch als perfekt bezeichnet, da zwischen dem Original und dem Duplikat keinerlei Unterschiede mehr bestehen.

Die daraus resultierenden Probleme sind etwa aus der Musikindustrie bekannt. Auf verschiedenen Peer-to-Peer-Plattformen werden Musikstücke zum Teil illegal zum kostenlosen Download angeboten. Diese Problematik betrifft auch die Softwareindustrie und hier insbesondere jene Anbieter, deren Produkte zur Nutzung nicht oder nur wenig angepasst werden müssen und die auch für private Anwender interessant sind. So werden beispielsweise Office-Anwendungen und Computerspiele häufig illegal kopiert bzw. über Filesharing-Netze ausgetauscht. In mehreren Studien wurde versucht, die Umsatzeinbußen der Softwarefirmen durch Raubkopien zu schätzen, beispielsweise in der jährlich erscheinenden „Piracy Study“ des Industrieverbands BSA (Business Software Alliance). Die Methodik und die diesen Studien zugrunde liegenden Annahmen sind allerdings umstritten. So wird vielfach einfach unterstellt, dass jede illegal bezogene Kopie zu einem Umsatzverlust führt, also jeder Raubkopierer das entsprechende Softwareprodukt ansonsten auch gekauft hätte. Dies führt zu immensen (rechnerischen) Umsatzeinbußen, die in dieser Größenordnung wenig realistisch erscheinen, obgleich es unstrittig ist, dass die entsprechenden Softwareanbieter tatsächlich Verluste durch die Anfertigung illegaler Kopien hinnehmen müssen.

Zum Schutz des geistigen Eigentums bzw. zum Verhindern illegaler Kopien können Anbieter digitaler Güter grundsätzlich Digital-Rights-Management-Systeme einsetzen (Hess und Ünlü 2004). Diese stellen mittels Hard- oder Softwareimplementierungen entsprechende Schutzverfahren bereit. Dabei kann es sich etwa um Zugangs- und Nutzungskontrollen, den Schutz der Authentizität und Integrität, die Identifikation über Metadaten, die Anwendung eines Kopierschutzes oder ein bestimmtes Bezahlssystem handeln.

---

## **2.2 Netzeffekte auf Softwaremärkten: The Winner Takes it All**

In diesem Abschnitt beschäftigen wir uns mit Netzeffekten, die auf Softwaremärkten eine wesentliche Rolle spielen. Denn neben der Funktionalität einer Software hat insbesondere auch deren gegenwärtige und zukünftig erwartete Verbreitung einen großen Einfluss auf den Nutzen für die Anwender. Dieser Zusammenhang wird im Rahmen der Theorie der

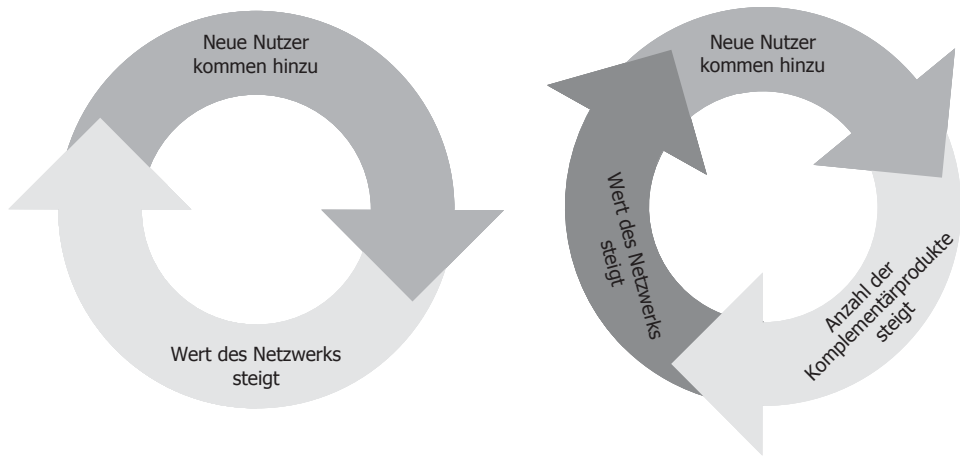
positiven Netzeffekte beschrieben und analysiert. Der erste Teil dieses Abschnitts enthält wesentliche Grundlagen und Definitionen (Abschn. 2.2.1). Danach beschreiben wir, warum sich nicht immer die besten Standards und die besten Softwarelösungen auf Netzeffektmärkten durchsetzen und speziell kleine Softwarefirmen und Startup-Unternehmen einen schweren Stand haben (Abschn. 2.2.2). In diesem Zusammenhang ziehen wir zum einen den so genannten Pinguineffekt als Erklärungsmodell heran. Zum anderen wird ein Modell von Arthur (1989) vorgestellt, das aufzeigt, wie kleine zufällige Ereignisse den Erfolg von Standards und Softwarelösungen auf Netzeffektmärkten bestimmen können. Im nächsten Schritt untersuchen wir, warum auf Netzeffektmärkten häufig die „Winner-takes-it-all-Regel“ gilt, was letztlich die Begründung für eine Vielzahl von Unternehmensübernahmen in der Softwarebranche darstellt (Abschn. 2.2.3). Darauf aufbauend wollen wir zeigen, wie Softwareanbieter die Existenz von Netzeffekten als Wettbewerbsfaktor ausnutzen können (Abschn. 2.2.4). Danach beschreiben wir in Abschn. 2.2.5 mit digitalen Spielen ein Anwendungsbeispiel, mit dem sich sowohl Plattformstrategien als auch zweiseitige Netzeffekte erklären lassen. Der Abschnitt schließt mit einigen Überlegungen zu den Grenzen und Erweiterungsmöglichkeiten der Netzeffekttheorie (Abschn. 2.2.6).

### 2.2.1 Netzeffekte: Grundlagen und Definitionen

Michael Katz und Carl Shapiro definieren das Konzept der Netzeffekte folgendermaßen: “The utility that a given user derives from the good depends upon the number of other users who are in the same network as he or she“ (Katz und Shapiro 1985, S. 424). Netzeffekte liegen also vor, wenn sich der Nutzen eines Gutes für einen Konsumenten dadurch erhöht, dass andere Konsumenten das Gut ebenfalls nutzen. Je größer das Netzwerk dabei ist, umso besser ist dies in der Regel für die Anwender. Dabei wird zwischen direkten und indirekten Netzeffekten unterschieden.

Direkte Netzeffekte entstehen, da die Anwender durch die gemeinsame Nutzung von Softwarestandards oder allgemeiner Technologien einfacher und damit kostengünstiger miteinander kommunizieren können. Das klassische Beispiel für direkte Netzeffekte ist das Telefon: Je mehr Personen ein Telefon besitzen, umso vorteilhafter ist diese Technologie für die Nutzer. Das gleiche Prinzip gilt etwa auch für XML-Vokabulare, wie xCBL, deren Nutzung ebenfalls umso vorteilhafter ist, je mehr andere Anwender das betreffende Vokabular nutzen. Auch im Bereich von ERP-Systemen spielen Netzeffekte auf zwischenbetrieblicher Ebene eine zunehmend wichtige Rolle. Durch die Nutzung standardisierter Formate wird beispielsweise der Austausch von Geschäftsdokumenten zwischen verschiedenen ERP-Systemen vereinfacht. Aus diesem Grund üben etwa in der Automobilindustrie die starken Partner in der Wertschöpfungskette häufig Druck auf kleinere Unternehmen aus, ein kompatibles und oftmals sogar identisches ERP-System zu verwenden. Ein weitergehender Schritt ist eine Prozessstandardisierung über Unternehmensgrenzen hinweg.

Indirekte Netzeffekte resultieren demgegenüber aus der Abhängigkeit zwischen dem Konsum eines Basisgutes und dem Konsum komplementärer Güter und Dienstleistungen.



**Abb. 2.1** Der Kreislauf der positiven Feedbacks – direkt und indirekt. (modifiziert nach Bansler und Havn 2002, S. 819)

Sie treten also dann auf, wenn die größere Verbreitung eines Gutes ein größeres Angebot an Zusatzgütern und Diensten nach sich zieht und damit wiederum der Nutzen des Basisgutes erhöht wird. Solche indirekten Netzeffekte treten beispielsweise auf bei Standardsoftware sowie komplementären Beratungsleistungen, bei Betriebssystemen mit dazu passender Anwendungssoftware oder wenn wir an die Verfügbarkeit von Experten und/oder Tools für Programmiersprachen denken.

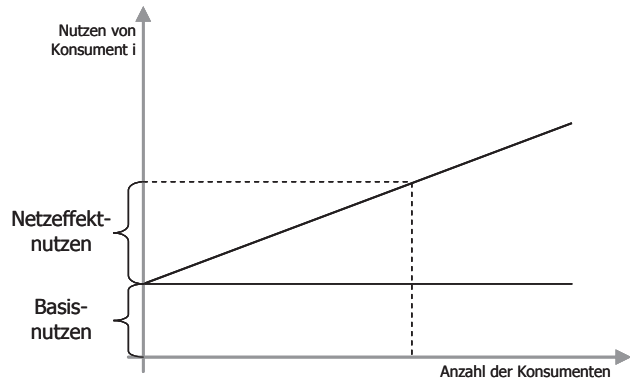
Netzeffekte führen zu nachfrageseitigen Skaleneffekten und zu so genannten positiven Feedbacks bzw. Increasing Returns. Shapiro und Varian fassen es wie folgt zusammen: „Positive Feedback makes the strong get stronger and the weak get weaker“ (Shapiro und Varian 1998, S. 175). Dieser sich positiv verstärkende Kreislauf ist in der folgenden Abb. 2.1 sowohl für direkte als auch für indirekte Netzeffekte dargestellt.

Diese Netzeffekte sind ein starkes Argument für Anwender, auf Softwareprodukte mit hoher Verbreitung zu setzen bzw. sich für Anbieter zu entscheiden, die ihnen starke Netzeffekte anbieten können.

Darüber hinaus können weitere Punkte für die Wahl eines großen Anbieters sprechen: An erster Stelle ist hier die Investitionssicherheit zu nennen, die insbesondere vor dem Hintergrund hoher Switching Costs (siehe hierzu auch Abschn. 2.2.4) eine zentrale Rolle spielt. Zudem gilt wohl auch, dass niemand seinen Job verlieren wird, wenn er bei der Auswahl der Softwarelösungen auf den Marktführer, etwa SAP oder Microsoft, setzt. Das Scheitern der Einführung einer Open Source Software dürfte demgegenüber für einen Entscheidungsträger wesentlich schwieriger zu begründen sein und eher zu beruflich negativen Konsequenzen führen.

Von diesem Netzeffektnutzen wollen wir im Weiteren den Basisnutzen unterscheiden, den eine Software stiften kann. Dabei handelt es sich um den Nutzen, den eine Software unabhängig von der Nutzung durch andere Anwender bietet. Ein Beispiel für ein Gut, das

**Abb. 2.2** Der Zusammenhang von Basis- und Netzeffektnutzen



sowohl einen Netzeffekt- als auch einen Basisnutzen stiftet, ist etwa ein Tabellenkalkulationsprogramm. Der Basisnutzen resultiert aus den angebotenen Funktionalitäten, der Netzeffektnutzen ergibt sich demgegenüber aus den Möglichkeiten, mit anderen Anwendern Dateien auszutauschen (direkter Netzeffekt) oder sich Tipps für den Umgang mit der Software zu holen (indirekter Netzeffekt). Dagegen ist ein E-Mail-System ein Beispiel für eine Software, die ausschließlich einen Netzeffektnutzen stiftet, denn was kann man schon als einziger Nutzer mit einem solchen System anfangen?

Abbildung 2.2 zeigt die Nutzenfunktion eines Anwenders. Dabei ergibt sich der Gesamtnutzen aus der Summe von Basis- und Netzeffektnutzen. In dieser Darstellung wird ein linearer positiver Zusammenhang zwischen dem Netzeffektnutzen und der Anzahl der Anwender angenommen.

Zur Messung der Stärke des Netzeffektnutzens im Vergleich zum Basisnutzen führen wir einen Netzeffektfaktor  $Q$  ein. Dabei bezeichnet im Weiteren  $c$  den Netzeffektnutzen und  $b$  den Basisnutzen einer Software. Den Netzeffektfaktor definieren wir nun wie folgt:

$$Q = \frac{c}{c + b}$$

Damit ist  $Q$  zwischen Null und Eins normiert. Je höher der Wert von  $Q$  ist, umso größer ist die Rolle, die Netzeffekte im Vergleich zum Basisnutzen spielen. Beispielsweise würde der Netzeffektfaktor einer Standardsoftware wie Microsoft Excel einen Wert zwischen Null und Eins annehmen. Ein EDI-Standard hingegen stiftet keinen Basisnutzen; insofern würde der Netzeffektfaktor in diesem Fall genau Eins werden. Wir werden auf diesen Netzeffektfaktor im Zusammenhang mit der Bewertung von Preisstrategien zurückkommen (siehe hierzu Abschn. 3.3 dieses Buches).

Im Folgenden wollen wir nun untersuchen, welche Auswirkungen die Existenz von Netzeffekten auf Softwaremärkte hat.

## 2.2.2 Auswirkungen von Netzeffekten auf Softwaremärkte

Beginnen wir mit einem einfachen Beispiel: Angenommen, einer neu gegründeten Firma würde es gelingen, eine Software zu entwickeln, die bessere Funktionalitäten hat als das Office-Paket von Microsoft. Die Software bietet den Kunden also einen höheren Basisnutzen. Würde sich dieses Startup-Unternehmen im Markt etablieren können? Vermutlich würden wir nicht viel Geld auf den Erfolg dieser Firma wetten. Die Schwierigkeit besteht für das kleine Unternehmen darin, dass der Marktführer – in diesem Fall Microsoft – seinen Kunden bereits den Vorteil hoher Netzeffekte bieten kann.

Aus gesamtwirtschaftlicher Sicht bestünde die optimale Lösung – unter der Bedingung, dass die Kosten für die Umstellung nicht zu hoch sind – darin, dass sich alle Anwender für die neue Software entscheiden würden. Denn sie stiftet, wie dargestellt, den höheren Basisnutzen und wenn alle Anwender einen Wechsel vornähmen, würden sich nach einer gewissen Zeit auch die entsprechenden Netzeffekte einstellen. Dennoch ist es unwahrscheinlich, dass ein solcher Wechsel tatsächlich stattfinden wird, da die Existenz dieser Netzeffekte zu einem „Lock-in“ in eine aus technischer Sicht nicht optimale Technologie führen kann (siehe hierzu auch David (1985) sowie Liebowitz und Margolis (1994)). Damit hat ein Unternehmen, das auf einem Netzeffektmarkt über eine installierte Basis verfügt, einen erheblichen Wettbewerbsvorteil, der von Konkurrenten nur schwer aufgeholt werden kann. In unserem Beispiel ist es der Microsoft-Konzern, der diesen Wettbewerbsvorteil hat.

### 2.2.2.1 Der Pinguineffekt

Dass die installierte Basis in vielen Fällen den Wechsel zu einem technisch überlegenen Standard verhindert, ist nach Farrell und Saloner auf Informations- und daraus resultierende Abstimmungsprobleme zurückzuführen (Farrell und Saloner 1985). Sie argumentieren wie folgt: Die Summe der Nutzen aller Marktteilnehmer könnte erhöht werden, wenn diese sich für einen Übergang zu dem neuen (technisch überlegenen) Standard entscheiden – in unserem Beispiel für die Programme der neu gegründeten Softwarefirma. Die Anwender sind jedoch unsicher, ob ein solcher Übergang tatsächlich stattfindet. Das Problem besteht für einen potenziellen Umsteiger bei unvollkommener Information darin, dass die anderen Marktteilnehmer möglicherweise nicht folgen und der Zugewinn an Basisnutzen durch die Anwendung des neuen Standards den entgangenen Netzeffektnutzen nicht kompensieren kann. Die Unsicherheit bezüglich der Reaktion der anderen Marktteilnehmer kann dazu führen, dass die Unternehmen in ihrem bisherigen Zustand verharren. Dieses Abstimmungsproblem wird auch als Pinguineffekt bezeichnet, wobei der Namensgebung folgende nette Analogie zugrunde liegt: Hungerige Pinguine stehen am Rande einer Eisscholle. Aus Angst vor Raubfischen hoffen sie, dass andere Pinguine zuerst ins Wasser springen, um das damit verbundene Risiko – nämlich Opfer eines Raubfisches zu werden – auszuloten. Sobald einige Pinguine den Sprung gewagt haben, hat sich die Gefahr für die anderen Pinguine verringert und die „Trittbrettinguine“ folgen nach (Farrell und Saloner 1987).

Aufgrund dieses Pinguineffektes sehen sich insbesondere Softwareunternehmen einem Startup-Problem ausgesetzt. Auch wenn natürlich jede Firma, die neu in einen Markt einsteigt, mit Schwierigkeiten zu kämpfen hat, gilt dies für Unternehmen auf Netzeffektmärkten in besonderer Weise. Das Startup-Unternehmen steht hier nicht nur vor der Aufgabe, das Produkt selbst überzeugend anzupreisen, sondern auch glaubhaft zu versichern, dass es sich flächendeckend durchsetzen und Netzeffekte generieren wird. Die Auswirkung dieses Startup-Problems aufgrund des Pinguineffekts wird in der angelsächsischen Literatur auch mit *excess inertia* (d. h. eine Unterversorgung mit Standards) bezeichnet. Daneben kann es grundsätzlich auch zu einer Überversorgung (*excess momentum*) mit Standards kommen.

Dieser Pinguineffekt ist ein wesentlicher Grund dafür, dass sich in vielen Fällen ein aus technischer Sicht nicht-optimaler Standard durchgesetzt hat; hierzu gibt es eine Vielzahl von Beispielen:

- Der Videostandard VHS war lange Zeit marktbeherrschend, obwohl Betamax aus technischer Sicht besser einzuschätzen war.
- OSI-Protokolle konnten sich trotz hoher technischer Qualität nicht flächendeckend auf allen Ebenen durchsetzen. Stattdessen dominieren heute die Internetprotokolle den Markt.
- Der Tastaturanordnung im so genannten QWERTY-Design wurde nachgesagt, weniger effizient als andere – wie zum Beispiel die später nach ergonomischen Gesichtspunkten entwickelte Dvorak-Tastatur – zu sein. Eine Umstellung auf eine möglicherweise effizientere Tastenanordnung wird aufgrund der Existenz von sowohl direkten als auch indirekten Netzeffekten verhindert.

Wir lernen daraus, dass sobald sich ein Standard erst einmal durchgesetzt hat, er nicht mehr so leicht aus dem Markt gedrängt werden kann.

### 2.2.2.2 Diffusionsprozesse: Das Modell von Arthur

Welcher Standard sich letztlich durchsetzt, kann auch durch zufällige kleine Ereignisse beeinflusst werden (Arthur 1989). Der Begriff Pfadabhängigkeit beschreibt diese Auswirkungen von frühen und gegebenenfalls zufälligen Ereignissen im Diffusionsprozess auf die spätere Marktstruktur (David 1985, S. 322). Das im Folgenden dargestellte Modell zeigt auch anschaulich, wie der Wettbewerb zwischen zwei Technologien stattfindet, die beide Netzeffekte erzeugen.

Als Ausgangspunkt dienen uns hierbei zwei Technologien A und B. Darüber hinaus existieren zwei Anwendertypen, die Arthur als R-Akteure bzw. S-Akteure bezeichnet. Der Gesamtnutzen einer Technologie ergibt sich dabei aus der Summe des Basis- und des Netzeffektnutzens. Der Basisnutzen wird in dem Modell durch die folgenden Parameter abgebildet:

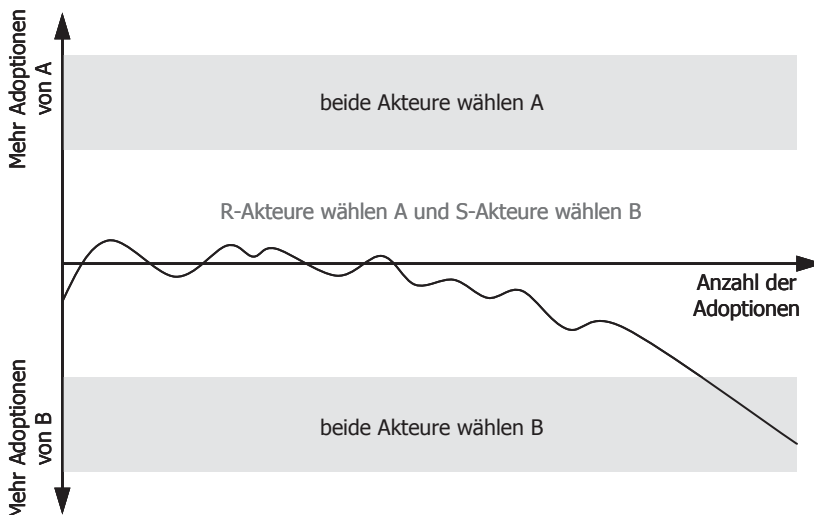
- $a_r$  Basisnutzen der Technologie A für R-Akteure
- $a_s$  Basisnutzen der Technologie A für S-Akteure
- $b_r$  Basisnutzen der Technologie B für R-Akteure
- $b_s$  Basisnutzen der Technologie B für S-Akteure

Dabei soll gelten, dass die Technologie A für R-Akteure einen höheren Basisnutzen hat als die Technologie B ( $a_r > b_r$ ). Umgekehrt stiftet Technologie B für die S-Akteure einen höheren Basisnutzen als Technologie A ( $a_s < b_s$ ).

Der Netzeffektnutzen wächst, wie in Abb. 2.2 bereits dargestellt, annahmegemäß linear mit der Anzahl der Nutzer. Dabei bezeichnet  $n_a$  die Anzahl der Nutzer von Technologie A und  $n_b$  entsprechend die Anzahl der Anwender von Technologie B.

Arthur zeigt nun in seinem Modell auf der Basis von Entscheidungen der Akteure, wie sich eine Technologie am Markt durchsetzt. Dazu entwickelt er ein Simulationsmodell, in dem sich zufällig ein R- oder ein S-Akteur zwischen den beiden Technologien entscheidet. Dabei wählen die Akteure die Technologie aus, die ihnen den höchsten Gesamtnutzen stiftet. Der Gesamtnutzen der Technologie A beträgt für die R-Akteure ( $a_r + r \times n_a$ ), der entsprechende Gesamtnutzen von Technologie B ( $b_r + r \times n_b$ ). Der Parameter  $r$  steht hierbei für den Grenznutzen jedes weiteren Anwenders. Die S-Agenten entscheiden annahmegemäß nach dem gleichen Kalkül. Abbildung 2.3 zeigt die Ergebnisse der Anwendung des Simulationsmodells.

In dem in der Abbildung weiß dargestellten Bereich wählen R-Akteure Technologie A und S-Akteure Technologie B, d. h. jeweils die Technologie, die ihnen den höchsten Basisnutzen stiftet. Wenn sich jedoch beispielsweise viele S-Agenten für Technologie B entscheiden, so steigt entsprechend der Netzeffektnutzen von Technologie B an. Dies führt



**Abb. 2.3** Pfadabhängigkeit aufgrund von „Increasing Returns“. (Arthur 1989, S. 120)



ab einer gewissen Nutzeranzahl dazu, dass auch R-Agenten sich nun für die Technologie B entscheiden, da die höheren Netzeffekte dieser Technologie den für die R-Akteure geringeren Basisnutzen überkompensieren.

Mit Hilfe dieses einfachen Modells lässt sich anschaulich zeigen, dass zum einen den so genannten „Early Adopters“ eine entscheidende Rolle bei dem Kampf um die Marktanteile zukommt. Zum anderen wird klar, dass es häufig zufällige Ereignisse sind, die die Verbreitung einer Technologie auf Netzeffektmärkten bestimmen.

### 2.2.3 Struktur von Softwaremärkten

Seit langem haben Ökonomen erkannt, dass Netzeffekte oftmals zu Monopolen führen. Daher werden Märkte, die starke positive Netzeffekte und Feedbacks aufweisen, häufig als „tippy“ bezeichnet: „When two or more firms compete for a market where there is strong positive feedback, only one may emerge as a winner. It's unlikely that all will survive.“ (Shapiro und Varian 1998, S. 176). Verschiedene Standards oder Technologien existieren häufig also nicht nebeneinander, sondern es setzt sich ein dominanter Standard bzw. eine dominante Technologie durch (Besen und Farrell 1994, S. 118). Es handelt sich deshalb um so genannte „Winner-takes-it-all-Märkte“.

Diese theoretischen Überlegungen werden durch Beobachtungen heutiger Softwaremärkte gestützt. Betrachten wir z. B. Märkte für Standardsoftwarelösungen: Hier ist es auffällig, dass die Anzahl der Anbieter deutlich zurückgegangen ist. So gab es vor einigen Jahren beispielsweise im Bereich von Browsern oder Office-Lösungen noch Alternativen zu Microsoftprodukten, wie etwa den Netscape Navigator, Word Perfect im Bereich Textverarbeitung oder 1-2-3 zur Tabellenkalkulation. Mittlerweile existiert auf diesen Märkten im Wesentlichen nur noch eine sehr starke Konkurrenz für Microsoft: die Open-Source-Gemeinde.

#### **Der Browserkrieg**

Der so genannte Browserkrieg entwickelte sich 1995 zwischen Microsoft und der Firma Netscape. Netscape ist mit seinem Internet Browser Netscape Navigator bekannt geworden und besaß vor 1995 einen Marktanteil von über 80 %. Als Microsoft schließlich die wachsende Bedeutung des Internets erkannte und daraufhin sein Konkurrenzprodukt, den Internet Explorer (IE), entwickelte, begann der „Krieg“ zwischen den beiden Firmen. In dessen Verlauf gab es zwei besonders interessante Aspekte, die im Folgenden näher beschrieben werden: zum einen die Bedeutung von Open-Source-Produkten als Konkurrenz zu kommerzieller Software, zum anderen die Auswirkungen von Produktbündelungen für ein Unternehmen.

#### ***Phase I: Die Anfänge des WWW***

Einer der wichtigsten Internetbrowser, der zu Beginn der rasanten Entwicklung des Internets Anfang der 90er Jahre auf den Markt kam, war MOSAIC. Er besaß als erster

Browser eine grafische Oberfläche, mit der es möglich war, im Internet zu „surfen“. Ende 1993 wurde dann die Firma Netscape gegründet, die bereits 1994 die erste Version des Browsers Netscape Navigator auf den Markt brachte. Bis zum Jahre 1995 hatte Netscape auf dem Browsermarkt schließlich eine Monopolstellung aufgebaut.

### ***Phase II: 1995–1998***

Als Microsoft entschied, in den Internetmarkt einzusteigen, wurde die erste Version des Internet Explorers entwickelt. Das Unternehmen hatte dabei zwei entscheidende Vorteile gegenüber Netscape: Zum einen standen Microsoft wesentlich mehr finanzielle Mittel für die Entwicklung der Browsersoftware zur Verfügung. Zum anderen hatte Microsoft die Möglichkeit der Produktbündelung und begann, den Internet Explorer in seine Programme einzubinden. Ab Windows 98 war der Internet Explorer dann fest in das Betriebssystem integriert (nach dem Motto: „Was einmal installiert ist, benutzen die Leute auch“). Nachdem Windows auf ca. 95 % aller neu verkauften PCs installiert war, zahlte sich die Bündelungsstrategie für Microsoft auch tatsächlich bald aus. Die Marktanteile des Internet Explorers stiegen in der Folgezeit von ursprünglich weniger als 3 % auf über 95 %. Schließlich musste sich Netscape geschlagen geben, als sein Marktanteil 1998 auf unter 4 % gesunken war. Netscape veröffentlichte daraufhin den Quellcode des Navigators und machte ihn damit zu dem Open-Source-Projekt „Mozilla“.

### ***Phase III: 2004 bis heute***

Diese Phase wird auch als die zweite Runde des Browserkrieges bezeichnet. 2004 wurden zunehmend Sicherheitslücken des Internet Explorers bekannt, was Mozilla die Chance bot, wieder Marktanteile zu gewinnen. Die Mozilla Community veröffentlichte im November 2004 den Firefox 1.0 (ein abgespeckter Browser). Die Gründe für die daraufhin sehr schnelle Verbreitung des Firefox liegen nicht nur in den Sicherheitsmängeln des Internet Explorers. Der Firefox bietet vielmehr auch eine Reihe praktischer Funktionalitäten (Tabbed Browsing, Find-as-you-type, Download Manager etc.), die von den Open-Source-Entwicklern integriert wurden. Microsoft konnte die veraltete Technologie seines Internet Explorers 6.0 nicht schnell genug aktualisieren, um zu verhindern, dass der moderne Open Source Browser Firefox ihm wichtige Marktanteile streitig machte. Erst mit seiner Version 7.0 (in der viele der neuen Funktionen beim Firefox abgeschaut sind) konnte sich der Internet Explorer wieder mit einem positiven Image auf dem Markt platzieren. In 2013 schneidet der Internet Explorer mit einem weltweiten Marktanteil von 30,7 % besser ab als Firefox mit 21,2 %.

*Quellen: heise-online (2002): Nicht Trustworthy – IE gefährdet Rechner und Netz. <http://www.heise.de/ct/02/25/100/>; Netplanet: Die Geschichte des Webbrowser. <http://www.netplanet.org/www/browser.shtml>; heise-online (2013): Marktanteile: Chrome und Safari legen zu, Internet Explorer und Firefox verlieren.*

*<http://www.heise.de/newsticker/meldung/Marktanteile-Chrome-und-Safari-legen-zu-Internet-Explorer-und-Firefox-verlieren-1775342.html>*

Die Softwareindustrie

Ökonomische Prinzipien, Strategien, Perspektiven

Buxmann, P.; Diefenbach; Hess, Th.

2015, XIV, 293 S. 134 Abb., Hardcover

ISBN: 978-3-662-45588-3