

# 1

## Warum Risiken?

*If you do not actively attack risks, they will actively attack you.*

*Tom Gilb*

Wir befinden uns mitten in einer Zeitenwende. Bisher stand die Produktion von Gütern im Mittelpunkt des wirtschaftlichen Interesses. Wer die Produktionsmittel besaß, musste fast zwangsläufig reich werden. Fabriken waren die Stätten, die in der Wirtschaft dominierten. Große Serien, gar Massenproduktion, sorgten für Umsatz und Gewinn.

Doch die Zeiten wandeln sich. Produktionen wandern nach China und Indien, wo sich konventionelle Güter wesentlich preisgünstiger herstellen lassen. Die Produktion steht nicht länger im Schweinwerferlicht. Sie muss Platz machen für das Informationszeitalter.

Information ist der Rohstoff der neuen Zeit. Damit wandeln sich die Regeln, an denen wir uns orientiert haben. Was Jahrzehnte richtig war, muss in Frage gestellt werden. Wer sich an die alten Regeln hält, gerät leicht ins Abseits. Ein derartiger Zeitenwandel ist zwangsläufig mit Risiken verbunden. Müssen wir uns auf diese Risiken einlassen? Können wir sie vermeiden? Oder lassen sie sich zwar umgehen, aber nur unter Bedingungen, die uns nicht gefallen werden?

Wenn wir diese Fragen beantworten wollen, hilft uns vielleicht ein Blick auf eine Epoche, in der ebenfalls eine Zeitenwende zu verzeichnen war.

### 1.1 Das Unbekannte

In alten Karten finden wir oftmals Drachen. Das bedeutet nicht, dass in diesen Ländern Drachen lebten, auf die Eindringlinge stoßen würden. Vielmehr wollten die Kartenmacher mit dem Symbol des Drachen darauf hinweisen, dass es sich um unerforschtes Gebiet handelte. In solche Länder vorzudringen hieß, das eigene Leben zu riskieren. Die Gefahr mochte in einer winzigen Fliege liegen, die ein unbekanntes tödliches Virus trug. Auf hoher See drohten die Seeleute zu verdursten, oder sie starben, weil sie an einem Mangel an Vitaminen litten. Gingen sie an Land, wurden sie möglicherweise von den Eingeborenen erschlagen.

Drachen fanden sie nicht, aber Gefahr in vielerlei Form. Sie mussten diese Gefahren meistern, die Drachen erschlagen. Nur wenn sie das taten, winkte ihnen der Erfolg. Die

Reichtümer, die zu Beginn der Neuzeit lockten, waren nicht unbeträchtlich. Nachdem die Seidenstraße, der uralte Handelsweg nach Indien und China, blockiert war, blieb gegen Ende des 15. Jahrhunderts nur der Seeweg, um diese fernen Länder zu erreichen. Es waren zwei Königreiche, die daran gingen, diese Routen zu erkunden: Spanien und Portugal.

»Eine Krippe als Wiege, als Grab die Welt«, lautet ein Sprichwort in Portugal. Im äußersten Südwesten Europas gelegen, ist Portugal ein recht kleines Land. Wer Karriere machen will, geht noch heute in die Fremde. Unter der Führung Heinrich des Seefahrers tasteten sich portugiesische Seefahrer ins Unbekannte vor. Entlang der Küste Afrikas fuhr man nach Süden, kam zu den Kapverdischen Inseln und entdeckte Afrikas Goldküste. Die Südspitze Afrikas wurde schließlich umrundet, und damit war der Weg frei in den Indischen Ozean. Man fand Städte, die von arabischen Händlern dominiert waren, etwa Mombasa im heutigen Kenia. In Indien wurde Goa zum Stützpunkt der Portugiesen, und von den Chinesen erwarb man Macao, nicht weit von Hongkong gelegen. Im Persischen Golf wurde Hormuz ein portugiesischer Handelsplatz. Und nicht zuletzt wurde in der Neuen Welt Brasilien Portugal zugeschlagen.

Die portugiesischen Entdeckungen fanden ihren Höhepunkt in der ersten Umrundung des Globus durch Ferdinand Magellan. Der Seefahrer wurde auf den Philippinen von Eingeborenen erschlagen, aber eines seiner Schiffe kehrte am 8. September 1522 in seinen Heimathafen zurück. Die *Victoria* hatte 17 spanische Seeleute und vier Indianer an Bord. Ihre Laderäume waren voller Gewürze. Diese Ladung war damals ein kleines Vermögen wert.

Wechseln wir die Bühne: Etwas weiter östlich, in Spanien, waren König Ferdinand und seine Gemahlin Isabella gerade dabei, die letzten Mohren aus ihrem Land zu vertreiben. Granada leistete am längsten Widerstand, doch am 2. Januar 1492 fiel die letzte Bastion der Mohren im Süden. Spanien wurde ein Bollwerk der Christenheit.

Christopher Columbus war ein Genueser, der sich zunächst bei den Portugiesen verdingte. Er nahm an Reisen nach Grönland, Irland und entlang der afrikanischen Küste teil. Heute wird der Genuese meistens als Angestellter der spanischen Krone portraitiert. Aber damit tut man ihm wohl unrecht. In erster Linie war Columbus ein Unternehmer. Er wusste, dass er sehr reich werden konnte, wenn er einen Seeweg nach Indien entdeckte. Im Gegensatz zu den portugiesischen Kollegen wollte er allerdings nicht um Afrika herum segeln, sondern direkt über den Atlantik nach Indien gelangen.

Ob Columbus nicht richtig rechnen konnte oder mit Absicht falsche Zahlen vorlegte, um die Risiken gegenüber seinen Geldgebern kleiner dazustellen, ist bis heute unklar. Jedenfalls verschätzte er sich beim Seeweg nach Indien um rund ein Drittel. Die Wissenschaftler am portugiesischen Hof konnten rechnen, und deswegen weigerte man sich dort, Columbus zu unterstützen. Er suchte daraufhin anderswo nach Sponsoren.

In Isabella, der katholischen Herrscherin von Aragon, Kastilien und Leon fand er eine Gönnerin. Am 3. August 1492 stach die kleine Flotte in See. Der Rest ist Geschichte ...

Was können wir aus diesen Beispielen kühnen Unternehmertums lernen? – Das Risiko war enorm. Columbus segelte gen Westen, obwohl er wusste, dass Wasser und Proviant für eine derart lange Reise kaum reichen konnten. Aber konnte er nicht einen enormen Gewinn einstreichen, wenn er Indien erreichen würde?

Portugal hatte den Indischen Ozean fast ein Jahrhundert für sich, bevor das britische Empire an Macht gewann und ihm die Herrschaft streitig machte. Spanien plünderte Mexiko aus. Enorme Mengen an Gold flossen nach Europa, und nicht zuletzt hatte Columbus einen Kontinent entdeckt, der besiedelt werden konnte: Amerika.

Die Risiken für Portugal und Spanien waren beträchtlich, aber hatten diese Königreiche eine Wahl? Bei Portugal kann man diese Frage sofort verneinen. Ein derart kleines Land ohne eigene Rohstoffe muss Handel treiben, seine Bürger können nur jenseits der eigenen Grenzen auf Reichtum hoffen.

Bei Spanien sah es vielleicht etwas anders aus, aber wir dürfen nicht vergessen, dass die beiden Reiche im Wettbewerb standen. Sie waren Rivalen, und das kleine Portugal schickte sich an, dem viel größeren Spanien den Rang abzulaufen. Unter diesen Umständen war es verständlich, dass das spanische Königspaar dem Abenteurer aus Genua eine Expedition in das Land der Drachen finanzierte.

Halten wir fest: Ins Unbekannte vorzustoßen ist mit hohen Risiken verbunden. Diese Risiken sind durchaus real. Wer versagt, kann ein Vermögen verlieren, und in früheren Zeiten bedeutete ein Misserfolg meist auch den Verlust des eigenen Lebens.

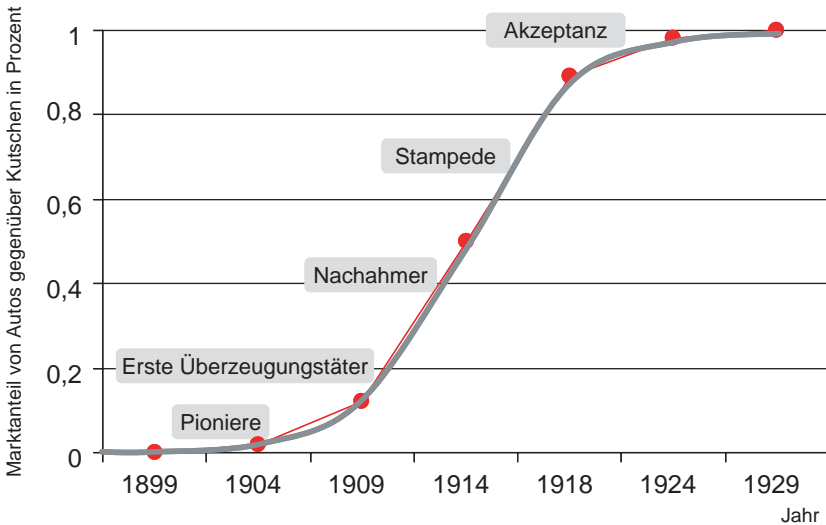
Diese Medaille hat allerdings auch eine Kehrseite, und diese heißt Chance. Hätten Portugal und Spanien ihre Chancen nicht genutzt, wären andere Völker in dieses Vakuum vorgestoßen. Vielleicht hätte Holland seine Chance bekommen, vielleicht wäre das britische Weltreich früher zur Blüte gelangt. Die Gelegenheit nicht am Schopfe zu packen, hätte weder Spanien noch Portugal zum Vorteil gereicht. Solche Chancen bieten sich nur einmal im Lauf der Geschichte.

## 1.2 Alte vs. neue Wirtschaft

Wie vor fünfhundert Jahren stehen wir heute wieder an einer Zeitenwende. Sie ist wie damals verbunden mit Risiken und Chancen, aber die Bedingungen sind gänzlich verschieden. Das liegt nicht zuletzt daran, dass inzwischen die industrielle Revolution ihren Lauf genommen hat. Sie war in ihren Anfangsjahren gekennzeichnet durch die Dampfmaschine, die zuerst in der Form der Lokomotive eine Applikation fand.

Ein wichtiges Produkt dieser Epoche ist das Automobil. Wenn wir seine Geschichte betrachten, können wir einige allgemein gültige Erkenntnisse gewinnen. Zunächst ist jedes industrielle Produkt in seiner Einführung und Vermarktung durch eine Glockenkurve gekennzeichnet. Das heißt, es existiert zunächst eine Phase, in der das neue Produkt – die Innovation – im Wettbewerb mit bestehenden Produkten seinen Markt erobern muss. Gelingt dies erfolgreich, so kann das Produkt über Jahre und Jahrzehnte hinweg den Markt beherrschen. Eines Tages wird jedoch ein besseres Produkt erfunden werden, und damit ist der Niedergang eingeleitet.

Befassen wir uns zunächst nur mit einer Hälfte dieser Glockenkurve. In Abbildung 1–1 ist dargestellt, wie sich das Automobil im Konkurrenzkampf mit den Pferdekutschen in der ersten Hälfte des 20. Jahrhunderts seinen Markt erobert hat.



**Abb. 1-1:** Vordringen des Automobils [1]

Am Anfang hat es jede neue Erfindung schwer. Zum einen ist sie einem breiten Publikum einfach nicht bekannt. Zum anderen zögern die potentiellen Käufer, ihre hart erarbeiteten Dollar für ein Produkt auszugeben, dessen Nutzen nicht bewiesen ist. Im Fall des Autos war allerdings der Nutzen, die individuelle Mobilität, so verlockend, dass das neue Gerät rasch Käufer fand.

Auf die ersten Anwender, die Pioniere, folgen jene Leute, die von der Erfindung durch Mundpropaganda gehört oder das Gerät auf der Straße gesehen haben. Noch immer gibt es relativ wenige Käufer, aber ihre Zahl wächst stetig. Schließlich wollen alle ein Auto haben. Diese Entwicklung führte beim Automobil dazu, dass zu Anfang der 30er Jahre die Pferdekutsche in den USA praktisch nicht mehr auf den Straßen zu finden war.

Für die Produktion von industriellen Gütern gibt es gewisse wirtschaftliche Gesetze. Entdeckt hat man diesen Zusammenhang in der 30er Jahren in der [1] Flugzeugindustrie. Wenn sich das Volumen der Produktion verdoppelt, sinken die Produktionskosten um 30 Prozent. Mit anderen Worten: Je höher die gefertigte Stückzahl, desto niedriger die Kosten.

Von Henry Ford ist der Ausspruch überliefert: »You can have it in any colour, as long as it is black.« Er spielte damit auf die Tatsache an, dass alle Autos von Ford zunächst schwarz waren. Vom Standpunkt eines Produzenten ist das durchaus vernünftig: Jede zusätzliche Farbe, jedes weitere Modell erhöht die Produktionskosten. Ideal wäre ein Modell, das in hohen Stückzahlen für alle Käufer mit der gleichen Ausstattung produziert wird.

Leider spielten da die Kunden nicht mit. Wer will schon genau das gleiche Modell wie der Nachbar vor der Haustür stehen haben? Deswegen war General Motors (GM) mit verschiedenfarbigen Autos und einer Reihe von Modellen im Markt erfolgreich. Alfred P. Sloan von GM erreichte die Kostensenkung durch andere Mittel. Er verwendete in allen Modellen gleiche oder ähnliche Subsysteme, etwa Motoren, Batterien, Bremsen oder Reifen. Diese wurden von Zulieferern gefertigt, die zwar GM gehörten, aber auf Effizienz getrimmt wurden. Auch auf diese Weise ist, um mit den Volkswirten zu sprechen, *Economics of Scale* erreichbar.

Wenn Boeing ein neues Flugzeug baut, dann dauert es ein paar Jahre, bevor die Entwicklungskosten wieder hereingeholt werden können. Sind rund achthundert Maschinen verkauft, ist der Punkt erreicht, ab dem der Hersteller einen höheren Gewinn macht. Die Entwicklungskosten sind vollständig abgeschrieben.

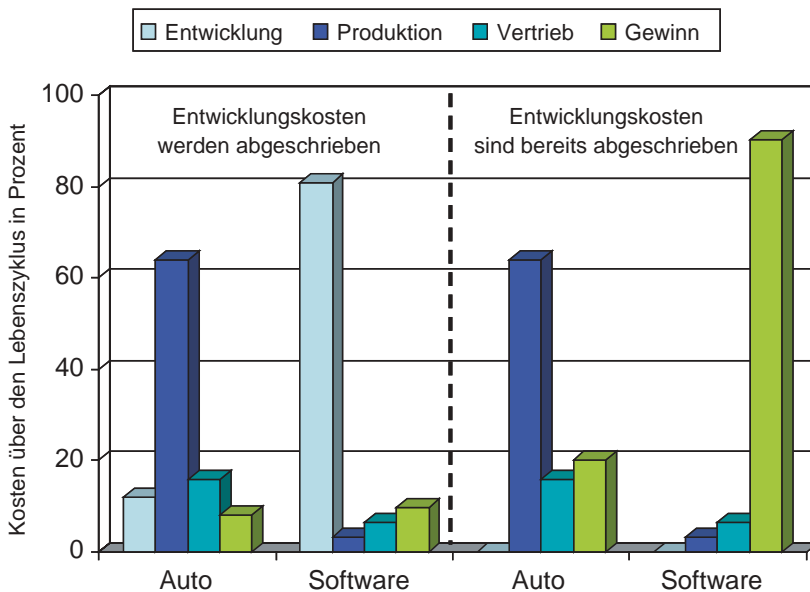


Abb. 1–2: Kosten über den Lebenszyklus eines Produkts

Ähnlich sieht es bei einem Auto aus. Der Zyklus für ein Modell von Opel oder VW beträgt rund fünf Jahre. In diesem Zeitraum werden die Entwicklungskosten abgeschrieben. Die höchsten Kosten fallen für die Produktion an, und daneben müssen wir mit Aufwendungen im Bereich Vertrieb und Marketing rechnen. Nicht zuletzt wollen wir Gewinn machen. Diese Tatsache ist in der ersten Säulengruppe in Abbildung 1–2 dargestellt.

Der größte Kostenblock entfällt auf die Fertigung. Das ist bei einem materiellen Produkt gewiss nicht verwunderlich. Anders sieht es hingegen aus, wenn wir uns der Software zuwenden. Denken wir zum Beispiel an ein populäres Produkt wie MS Windows oder eine ganze Gruppe von Programmen wie MS Office. Wenn wir hier an die Produktion denken, so besteht sie im Grunde darin, von einem Original auf einer CD-ROM Tausende von Ko-

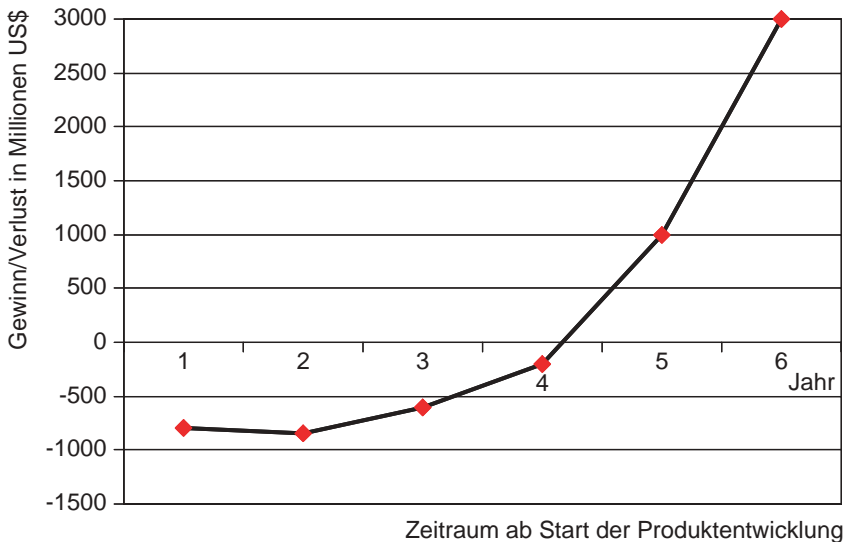
prien anzufertigen. Die Kosten dafür liegen im Bereich von ein paar Cents. Verglichen mit den Kosten, die für ein Auto anfallen, können wir sie praktisch vernachlässigen. Ins Gewicht fallen dagegen die Entwicklungskosten. Sie machen rund 80 Prozent der Gesamtkosten über den ersten Teil des Lebenszyklus aus.

Fragen wir uns nun, wie sich die Kosten entwickeln, nachdem vier oder fünf Jahre vergangen sind. In beiden Fällen werden nach diesem Zeitraum die Entwicklungskosten abgeschrieben sein. Wir können sie für die zweite Periode vernachlässigen. Damit kann sowohl der Automobilhersteller als auch der Verkäufer von Software billiger anbieten. Die Frage ist lediglich: Wird er das tun?

Verkauft Boeing seine Flugzeuge billiger, weil die magische Grenze von 800 Stück erreicht worden ist? Damit ist wohl kaum zu rechnen. In den meisten Fällen bleibt der Listenpreis gleich, und der Hersteller streicht einfach einen höheren Gewinn ein.

So verhalten sich auch unsere Automobilkonzerne und die Hersteller von Software. Allerdings ist zu bedenken, dass der Autofabrikant weiterhin rund sechzig Prozent des Listenpreises für die Fertigungskosten aufwenden muss. Sein Gewinn steigt, aber nicht so gewaltig wie bei der Software. Dort fällt der Kostenblock für die Entwicklung vollständig weg, und im gleichen Ausmaß explodiert der Gewinn.

Bill Gates ist der reichste Mann der Welt. Wer das oben gezeichnete Bild verstanden hat, der wundert sich darüber nicht. Es ist fast unausbleiblich, dass Bill Gates so reich wurde. Wer die ersten paar harten Jahre durchhält, wer diese Durststrecke übersteht, der kann hinterher absahnen. Diese Entwicklung ist in anderer Form in Abbildung 1–3 zu sehen.



**Abb. 1–3:** *Kosten über den Produkt-Lebenszyklus [1]*

Ist ein Software-Produkt also erst im Markt etabliert, dann explodiert der Gewinn geradezu. Da können in einem einzigen Jahr durchaus Profite im Bereich von Milliarden Dollar anfallen.

Der ein oder andere Leser mag einwenden, dass weitere Entwicklungskosten für die Produktpflege anfallen werden. Das ist richtig, aber fallen diese für den Hersteller wirklich ins Gewicht? Bei jedem Update werden ein paar neue Funktionen hinzukommen, aber gleichzeitig werden Fehler beseitigt. Wer einen Wagen kauft, bei dem der Lack abblättert, der beschwert sich beim Händler. Der Hersteller wird in der Folge den Schaden für den Kunden kostenlos beseitigen.

Anders sieht es dagegen im Bereich der Software aus. Hier zahlt der Kunde für ein Update hundert oder zweihundert Euro. Der Anbieter wird behaupten, dass dieser Preis für die vielen neuen Funktionen, die er erwirbt, durchaus angemessen ist. Aber seien wir ehrlich: Der Funktionsumfang in der alten Version war für unsere Bedürfnisse durchaus ausreichend. Im Grunde haben wir für die Mängelbeseitigung, die es in anderen Branchen umsonst gibt, noch Geld auf den Tisch gelegt: Wunderbare Welt der Software!

Wer nun als Entwickler glaubt, er könne fast zwangsläufig zum Millionär werden, täuscht sich allerdings. Die Regeln der alten Wirtschaft gelten in der neuen Wirtschaft nicht mehr. Was für Henry Ford richtig war, kann für Bill Gates und seine Nachahmer falsch sein.

## 1.3 Monopoly-Spieler

*Change in the industry isn't something to fear; it's an enormous opportunity to shuffle the deck, to replay the game.*

*Jack Welch*

In der neuen Wirtschaft wird ein Spiel gespielt, das durchaus Züge des populären Monopoly trägt. Microsoft würde das vehement bestreiten, aber es gab in der US-Regierung zu Zeiten von Bill Clinton durchaus Juristen, die dem Riesen aus Seattle gerade das vorwarfen. Die Konkurrenz stimmte in diese Klage ein, aber das hat keinen Vorstandsvorsitzenden jemals davon abgehalten, genau das gleiche Spiel wie Microsoft zu spielen.

In der alten Wirtschaft herrscht ein Gleichgewicht zwischen Angebot und Nachfrage. Gibt es in einem Jahr wenig Kirschen, weil die Blüten erfroren sind, steigen die Preise. Fällt hingegen die Ernte wider Erwarten prächtig aus, sinken die Preise, weil das Angebot die Nachfrage übersteigt. In der neuen Wirtschaft gelten dagegen andere Gesetze. Der Rohstoff für Computer-Chips ist *Silicon*, also im Grunde Sand. Den gibt es in der Erdkruste im Überfluss. Er kostet fast gar nichts. Was Chips teuer macht, ist der menschliche Geist, der in ihre Entwicklung fließt. Wenn allerdings diese Entwicklungskosten auf Millionen von Chips umgelegt werden können, dann wird die einzelne Scheibe Silicon wieder erschwinglich. Wenn wir im Bereich der Automobilindustrie den gleichen Preisverfall wie bei der Elektronik erleben würden, dürfte ein Mittelklassewagen heute nicht mehr als ein paar Euro kosten.

Allerdings können wir auch in diesem Sektor der Wirtschaft das Eindringen der Elektronik [2] nicht leugnen. Im Jahr 1994 enthielt ein Wagen von General Motors Stahl im Wert von 675 US\$. Dagegen beliefen sich die Kosten für elektronische Bauteile und Software auf US\$ 2 500. Wir finden sie in den Bremsen und der Zündung, im Radio und im

Navigationssystem. Der Anteil dieser Kosten steigt, während der Stahl weiter abnimmt. Kunststoffe sind dagegen auf dem Vormarsch, weil sie weniger wiegen als Metalle.

Sehen wir uns nun die Marktanteile für den PC im Jahr 1995 für die USA an (siehe Tab. 1–1).

Unternehmen	Gewinn in Milliarden US\$	Marktanteil [%]
Apple	11,1	11,5
Compaq	14,9	11,7
Dell	3,5	5,0
Gateway	3,7	5,1
IBM	71,5	8,2
Packard Bell	7,5	11,6

**Tabelle 1–1:** Profit und Marktanteil im PC-Markt [1]

Dieser Markt ist durch Fragmentierung gekennzeichnet. Keiner der aufgeführten Anbieter hat eine dominierende Position. Die Folge ist ein intensiver Preiskampf. Dies führt zu sinkenden Preisen für den Verbraucher, ist aber für die Anbieter nicht gerade positiv. Wer im Markt bleiben will, ist gezwungen, Jahr für Jahr die Preise zu senken. Das führt zwangsläufig zu knappen Margen und mageren Gewinnen.

Für die Anbieter ist die oben geschilderte Situation beim PC folglich nicht erstrebenswert. Was sie versuchen, wird in der neuen Wirtschaft *Mainstreaming* genannt. Das bedeutet, ein Produkt für den Verbraucher so attraktiv zu machen, dass er es haben will. Wenn ein Anbieter mit einer bestimmten Applikation der erste auf den Markt ist, hat er eine recht gute Chance, mit seinem Produkt der Marktführer zu werden. Microsoft hat es bei MS Windows und Office der Konkurrenz vorgemacht.

Das Spiel, das in der neuen Wirtschaft gespielt wird, bezeichnet man als Lanchester-Strategie. Die Spieler kann man in drei Gruppen einteilen.

- **Players:** Diese Anbieter müssen mindestens 26,1 Prozent Marktanteil besitzen. Ist ihr Marktanteil geringer, zählen sie nicht und laufen Gefahr, mangels Masse aus dem Markt zu verschwinden. Obwohl Player nicht unbedingt zu den Firmen gehören, die durch innovative Produkte glänzen, stellen sie für die übrigen Unternehmen in einem Marktsegment eine ernst zu nehmende Konkurrenz dar.
- **Leader:** In der alten Wirtschaft nahm man an, dass eine Firma über fünfzig Prozent Marktanteil besitzen müsse, um dieses Segment zu dominieren. In der neuen Wirtschaft genügt dagegen ein Anteil von 41,7 Prozent. Hat ein Anbieter diese Marke erreicht, ist er kaum mehr zu stoppen.
- **Monopolist:** Ab 73,9 Prozent Marktanteil sprechen wir von einem Monopolisten. Solch ein Unternehmen beherrscht sein Segment und kann die Preise diktieren. Es macht horrende Gewinne, weil es praktisch keine Konkurrenz mehr zu fürchten braucht.

Das Spiel funktioniert nicht nur bei Betriebssystemen und Anwendungsprogrammen für den PC, sondern auch im Bereich des Internet. Da gab es Mitte der 1990er Jahre [2] im US-

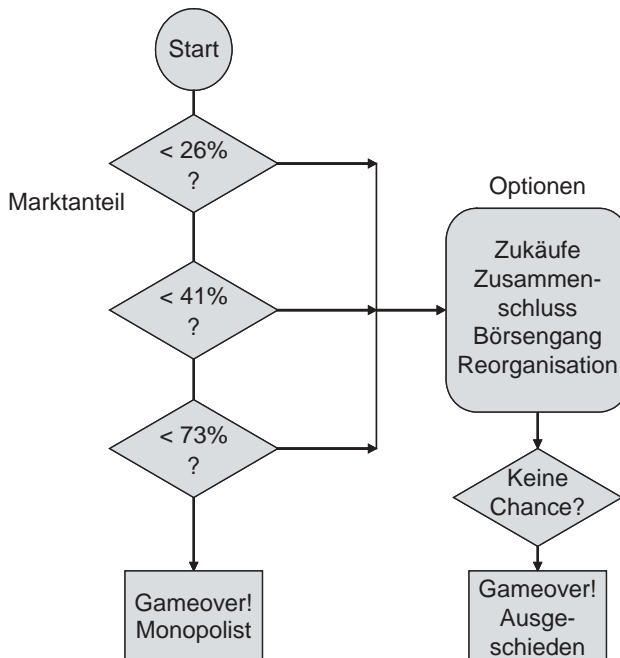
Markt noch ein paar Anbieter, die sich Hoffnungen machten, die Marktführerschaft zu eringen. Die Lage damals ist in Tabelle 1–2 dargestellt.

Unternehmen	Marktanteil [%]	Status
AOL	28	Player
CompuServe	25	Mitspieler, instabil
Prodigy	20	Mitspieler, instabil

**Tabelle 1–2:** Anbieter im Internet [2]

Inzwischen hat sich die Situation gewandelt. AOL hat CompuServe aufgekauft und ist damit zum unbestrittenen Marktführer geworden. Ähnliche Entwicklungen können wir im Marktsegment für Groupware beobachten. Dort ist IBM durch den Kauf von Lotus zu einer dominierenden Kraft geworden.

Wir können die Lanchester-Strategie auch in der Form eines Flussdiagramms aufzeigen. Damit ergibt sich Abbildung 1–4.



**Abb. 1–4:** Lanchester-Strategie [2]

Ein Spieler in diesem Markt hat eine Reihe von Optionen, um zum Gewinner zu werden. Wenn ein Unternehmen ein gutes Produkt hat, das verspricht, zu einer so genannten Killer-Applikation zu werden, dann werden am Anfang vermutlich hohe Kosten für die Entwicklung anfallen. Falls der Gründer des Unternehmens diese Anlaufkosten nicht aufbringen

kann, bietet sich ein Börsengang an. Das war bis vor wenigen Jahren wenigstens die gängige Art der Finanzierung. Inzwischen sind die Anleger skeptisch geworden, wenn sie von jungen Unternehmen im Bereich der Software hören. Das sollte allerdings niemanden abschrecken. Falls die Idee gut ist, bietet sich eine private Finanzierung als Alternative an.

Durchaus üblich in diesem Haifischbecken sind auch Zukäufe und Zusammenschlüsse. Wenn das Produkt des Konkurrenten überlegen ist, stellt eine Übernahme dieses Unternehmens einen gangbaren Weg dar, um zum Player oder Leader zu werden. Auch eine Reorganisation kann mitunter zum Erfolg führen. Denken wir etwa an die Ausgliederung eines Produkts in eine Tochterfirma oder an ein Joint Venture im begrenzten Rahmen mit einem Wettbewerber. Gegenwärtig versuchen Fujitsu und Siemens, auf diesem Weg im PC-Markt in Europa ihre Chancen zu verbessern.

Fassen wir zusammen: Wer nicht zumindest Player werden kann, scheidet bald aus dem Markt aus. Das heißt in anderen Worten: Insolvenz und Pleite. Das ist für Unternehmer kein ungewohntes Risiko, aber im Markt der Lanchester-Strategie wird mit harten Bandagen gespielt.

Auf der anderen Seite sind Gewinne in Milliardenhöhe nicht ungewöhnlich. Das ist der Preis, der den Siegern winkt.

## 1.4 Risiken anderer Art

*In the new economy, human invention increasingly makes physical resources obsolete... Even as we explore the most advanced reaches of science, we're returning to the age-old wisdom of our culture ... In the beginning was the spirit, and it was from this spirit that the material abundance of creation issued forth.*

*Ronald Reagan in einer Rede an der Universität von Moskau*

Während wir bisher die Risiken eher aus makroökonomischer Sicht betrachtet haben, wollen wir uns nun einzelnen Branchen und ihren Risiken zuwenden. Software ist ein Produkt des menschlichen Geistes, ein immaterielles Gut. Das birgt, verglichen mit traditionellen Gütern, ganz neue Arten von Risiken.

Die mit der Software-Entwicklung und ihrem Einsatz verbundenen Risiken sind vor allem auf zwei Trends zurückzuführen:

1. Programme werden immer größer und komplexer.
2. Die Software dringt in alle Anwendungsgebiete und alle Bereiche menschlicher Betätigung vor.

Der erste Trend ist wohl unvermeidlich. Die Anwendungen, bei denen mit geringem Aufwand an Software viel erreicht werden konnte, sind bereits im Einsatz. Jetzt werden zunehmend Applikationen erstellt, die sich mit nicht ganz leicht zu durchschauenden Sachverhalten beschäftigen. Entsprechend umfangreich und komplex wird das zugehörige Programm sein.

Der zweite Trend ist überall zu beobachten. Software steuert Sonden, die sich auf dem Weg zum Mars oder zur Venus befinden. Software ist für die Steuerung der Verkehrsleitsysteme einer Stadt verantwortlich, und wenn sie versagt, können sämtliche Ampeln ausfallen. Software ist dafür verantwortlich, bei Gefahr ein Atomkraftwerk abzuschalten. Software beherrscht den Autopiloten eines Verkehrsflugzeugs wie den Airbus, und wir finden sie auch im Herzschrittmacher, den einige Mitmenschen im Körper tragen.

Weil Software so allgegenwärtig ist, geht damit einher die Verbreitung des Risikos, das damit verbunden ist. Allerdings sind nicht alle Applikationen über einen Kamm zu scheren. Wer ein Videospiel für den PC erstellt, der mag als kritische Faktoren für den Markterfolg packende Grafiken und eine hohe Geschwindigkeit ausgemacht haben. Stürzt das Programm beim Einsatz ab, ist das zwar unangenehm, ein Risiko oder eine unmittelbare Gefahr für den Nutzer ist damit aber nicht verbunden.

Anders sieht es dagegen beim Flugzeug aus. Wenn der Autopilot ein falsch eingestelltes Messgerät nicht bemerkt, wenn die Maschine plötzlich in den Sturzflug übergeht, dann kann dies zum Verlust Hunderter von Menschenleben führen. Wir sprechen in diesem Fall von sicherheitskritischer Software.

In der Raumfahrt ist Software schon deswegen unentbehrlich, weil es bei langen Flügen nicht möglich ist, eine menschliche Besatzung mit ihren Bedürfnissen über Monate und Jahre hinweg zu unterhalten. Da bietet sich als Ausweg ein technisches System an. Um die Ausfallsicherheit zu erhöhen, setzt man wie beim Auto auf Redundanz. So ganz ohne Tücken ist jedoch auch diese Technik nicht, wie der folgende Fall zeigt.

### **Fall 1–1: Zweifach redundant [3,4,5]**

Ende April 2001 kam die International Space Station (ISS) in erhebliche Probleme, weil ihre drei Bordcomputer gleichzeitig ausfielen. Zu dieser Zeit war gerade die Besatzung der Raumfähre *Endeavour* dabei, den kanadischen *Space Arm*, eine Art Kran, zu montieren. Man glaubte ursprünglich, die neu hinzugekommenen Module für diesen Kran hätten mit dem Ausfall der Bordcomputer zu tun.

Für die NASA im Kontrollzentrum in Houston war es die größte Krise seit dem Jahr 1992. Die Probleme begannen am 25. April nach zwei Tagen erfolgreicher Montage des Krans. Die C&C genannten Computer sollen automatisch einspringen, wenn einer davon ausfällt. Man konnte die Geschehnisse im Nachhinein wie folgt rekonstruieren:

1. C&C-2: Dieser Computer bekam nach dem Ausfall von C&C-1 den Befehl, die Kontrolle über die Raumstation und ihre Systeme zu übernehmen. Als er hochlief, kam eine Warnung. Sie besagte, dass nicht auf die Festplatte zugegriffen werden konnte. Das Team in Houston sandte mehrmals Reset-Kommandos zur ISS, aber sie bewirkten gar nichts. Die Controller in Houston erfuhren später, dass dieser Computer immer klagte, dass seine Festplatte nicht arbeitete, obwohl sie in Ordnung war. Diese Tatsache war ihnen zu diesem Zeitpunkt aber nicht bekannt, und so folgten sie ihrem Verfahren.
2. C&C-3: Die Telemetriedaten von der dritten Einheit schauten normal aus. Der Computer fuhr aber nicht hoch. Offenbar hatte seine Festplatte bereits viel früher versagt. Das war allerdings im Kontrollzentrum in Texas nicht bekannt. Die Telemetriedaten zeigten Werte an, die offenbar veraltet und nicht mehr aktuell waren. Die Analyse deutete darauf hin, dass die Festplatte von C&C-3 zwei Wochen lang nicht mehr angesprochen worden war.

Die unmittelbare Folge des Computerausfalls war, dass einige größere Operationen auf der ISS nicht durchgeführt werden konnten. Die Ladung eines Containers, den die *Endeavour* mit zurück zur Erde nehmen sollte, wurde um zwei Tage verschoben. Zuletzt wechselte die Crew C&C-1 gegen eine identische Einheit aus, die an Bord war und für andere Zwecke eingesetzt wurde. Für den ausgefallenen C&C-3 Computer kam am 7. Mai ein Ersatzcomputer an Bord.

Die Festplatten der Computer fielen aus, weil die Schreib/Leseköpfe beschädigt waren. Dieser Schaden entstand, weil die Schreib/Leseköpfe bei Nichtgebrauch der Festplatte nicht wie vorgesehen in ihre Parkposition zurückfahren konnten. Dazu sollte kurz vor Ausschalten des Hard Drives die verbleibende Spannung benutzt werden. Diese reichte jedoch für die Bewegung nicht aus. Die Köpfe blieben hängen und beschädigten in der Folgezeit die Oberfläche der Festplatten.

Bemerkenswert ist, dass dieser Fehler erst im Weltraum bemerkt wurde. Was spricht dagegen, das Herunterfahren der Festplatte bei Spannungsabfall bereits auf der Erde zu prüfen?

In den 1990er Jahren stand die NASA unter erheblichem Druck. Ihre Missionen sollten erheblich billiger werden, ohne an der Qualität Abstriche zu machen. Ob dieser Ansatz erfolgreich war, kann jeder Bürger dieses blauen Planeten selbst beurteilen. Sehen Sie dazu das folgende Beispiel.

### **Fall 1–2: Störsignale [6]**

Im Jahr 1999 sandte die NASA eine Reihe von Sonden zum Mars. Darunter war auch der Mars Polar Lander (MPL). Diese Sonde sollte auf der Oberfläche des roten Planeten landen und für 90 Tage Daten zur Beschaffenheit der Oberfläche und des Klimas erfassen und zur Erde senden. Leider hörte man auf der Erde nichts mehr von der Sonde, als sie in die Atmosphäre des Mars eintrat.

Eine detaillierte Untersuchung zur möglichen Unfallursache durch die NASA brachte die folgenden Erkenntnisse: Mit hoher Wahrscheinlichkeit war es so, dass durch das Ausfahren der Beine des MPL Störsignale erzeugt wurden, die so interpretiert wurden, als wäre die Sonde bereits gelandet. Dadurch wurde der Raketenmotor abgeschaltet. Die Folge war, dass die Sonde auf der Oberfläche zerschellte.

Dazu heißt es im Untersuchungsbericht der NASA: »Es ist bei Sensoren für mechanische Teile nicht ungewöhnlich, dass von ihnen Störsignale ausgehen können. Im Fall des MPL bestand für die Software keine Forderung, Störsignale auszufiltern, bevor die Daten der Sensoren benutzt wurden, um eine erfolgreiche Landung anzuzeigen. Während des Systemtests war die Sonde falsch verdrahtet. Dies war auf einen Designfehler zurückzuführen. Die Folge war, dass der vorhandene Systemfehler während des Tests nicht identifiziert werden konnte. Ursächlich für die Zerstörung der Sonde ist ein frühzeitiges Ausschalten des Raketenmotors. Die eigentliche Ursache liegt allerdings in der Programmierung. Die Software war nicht in der Lage, Störsignale zu erkennen und zu verwerfen.«

Wer nun die NASA schelten will, sollte nicht vergessen, dass die europäischen Raumfahrtbehörden nicht besser sind. Stichwort: Ariane.

### Fall 1–3: Überlauf [7,8]

Im Frühjahr 1996 zerbrach die ARIANE 5, die neueste Rakete des europäischen Konsortiums, bei ihrem Jungfernflug wenige Sekunden nach dem Start. In die Neuentwicklung waren 8 Milliarden US\$ investiert worden. Die Betreiber sagten eine Zuverlässigkeit des Systems, basierend auf den Daten des Vorgängermodells ARIANE 4, von 98,5% voraus.

Im Juli stand das Ergebnis der Ermittlungen zur Fehlerursache fest. Ursächlich war ein Software-Fehler im Trägheitsnavigationssystem der Rakete. Dreißig Sekunden nach dem Start ging der zweite Computer der Rakete, ein redundantes System, außer Betrieb. Dies wurde durch einen Fehler in einem Unterprogramm der Software verursacht. Obwohl dieses Unterprogramm nur am Boden gebraucht wird und während des Flugs eigentlich unnötig ist, wird es periodisch ausgeführt. Bei der ARIANE 5 kalkulierte dieses Unterprogramm im Flug eine große horizontale Bewegung, die zu einem Überlauf (overflow) bei einer Variablen führte. Die Design-Philosophie bei der Rakete war es, bei einem derartigen Rechenfehler den Prozessor anzuhalten.

Nachdem das redundante Back-up-System ausgefallen war, fiel 50 Millisekunden später auch das Hauptsystem aus, das aus identischer Hard- und Software bestand. Dies führte dazu, dass das Trägheitsnavigationssystem nur Diagnosedaten an den Hauptcomputer der ARIANE 5 lieferte. Diese Daten wurden als gültige Steuersignale interpretiert. Die ARIANE versuchte, eine Abweichung von der Flugbahn zu korrigieren, die in der Realität gar nicht bestand. Durch die extremen Korrekturmanöver wurde die Rakete so überlastet, dass sie auseinanderbrach.

Das Unterprogramm, das zu dem Fehler führte, wurde von der ARIANE 4 übernommen. Dabei ging man davon aus, dass die Software in der ARIANE 4 fehlerfrei sei. Offensichtlich war bei der ARIANE 4 die horizontale Bewegung geringer, so dass bei diesem System der Fehler zwar auftrat, allerdings nicht zu einem Überlauf führte.

Dieser Verlust der Rakete hätte aus einer ganzen Reihe von Gründen vermieden werden können:

- Die Routine, die letztlich zum Absturz führte, war im Flug unnötig. Sie diente zur Positionsbestimmung am Boden. Durch eine simple Abfrage hätte der Aufruf nach dem Start vermieden werden können. Diese Programmierung hätte sogar Rechenzeit gespart.
- Als Programmiersprache kam Ada zum Einsatz. In dieser Sprache können Ausnahmbedingungen abgefangen werden. Natürlich bläht dies den Code auf, aber grundsätzlich besteht diese Möglichkeit.
- Die Routine zur Positionsbestimmung wurde deswegen als fehlerfrei betrachtet, weil sie bei der ARIANE 4 bereits im Einsatz war. Auf einen erneuten Test vor dem Einsatz wurde folglich verzichtet. Bei dem Vorgängermodell sind allerdings einige Parameter anders, und daher sind die dort herrschenden Bedingungen für die ARIANE 5 nicht in jedem Fall relevant.

Die europäische Raumfahrt wird durch Steuergelder finanziert, und deswegen sind solche kritischen Fragen durchaus legitim. Seit Jahrzehnten wird versucht, die alten Einheiten wie Pfund oder Meilen durch metrische Einheiten zu ersetzen. Noch ist dieser Kampf nicht gewonnen. Zu welchen Konsequenzen die Verwendung alter und neuer Einheiten im Messwesen führen kann, demonstriert der folgende Fall.

**Fall 1–4: Umrechnungen [9,10,11,12]**

Am 24. September 1999 ging die Raumsonde Mars Orbiter der NASA, die auf dem roten Planeten landen und die Oberfläche erkunden sollte, kurz vor dem Ziel verloren. Das war umso erstaunlicher, als die Sonde bis zu diesem Zeitpunkt ohne Probleme funktioniert hatte.

Als Ursache des Fehlers stellte sich schließlich heraus, dass die Umrechnung verschiedener Maßeinheiten nicht funktioniert hatte. Der Fehler lag in einer Umrechnungstabelle in der Support Software der Kontrollstation in den USA, die verwendet wurde. Dabei ging es darum, den Kurs der Sonde auf dem Weg zum Mars durch Korrekturmanöver zu berichtigen. Die am JPL in Kalifornien eingesetzte Software erwartete für den Schub die Einheit Newton, während man beim Hersteller Martin Lockheed offensichtlich mit Pfunden rechnete. Die Folge war, dass bei den Steuermanövern der Schub während der gesamten Flugzeit um 22 Prozent zu gering ausfiel.

Der Fehler war während des gesamten Flugs zum 220 Millionen Kilometer weit entfernten Mars gegenwärtig, wurde aber offensichtlich nicht bemerkt. Üblicherweise sind die Bahnen derartiger Sonden auf 10 Kilometer genau. Kritisch wurde es erst bei der Annäherung an den roten Planeten. Die Sonde kam der Oberfläche um 100 Kilometer (62 Meilen) zu nahe. Der Mars Orbiter, dessen Kosten mit 125 Mio. US\$ angegeben werden, ist auf der Marsoberfläche zerschellt.

Dieser Vorfall ist ein weiterer Beweis dafür, dass die NASA mit ihrem Konzept billiger Missionen in der Praxis gescheitert ist. Bei einer wachsenden Qualitätssicherung und einem ordentlichen Testprogramm hätte dieser Fehler bereits auf der Erde gefunden werden müssen.

Nach dem Unfall mit der Raumfähre Columbia am 1. Februar 2003 stand nur noch die russische Soyuz-Kapsel zur Verfügung, um die Besatzung der International Space Station (ISS) abzuholen und zu diesem Habitat im Erdorbit zu bringen. Leider war auch dieses System nicht ganz ohne Probleme.

**Fall 1–5: Harte Landung [13,21]**

Am 4. Mai 2003 sollte der Kosmonaut Nikolai Budarin sowie die Astronauten Donald Pettit und Kenneth Bowersox in einer Soyuz-Raumkapsel zur Erde zurückkehren. Um die hohe Geschwindigkeit von 25 Mach abzubremsen, benutzt die Kapsel die äußeren Schichten der Erdatmosphäre in 7 600 Kilometer Höhe als Reibungsfläche. Dabei dreht sich die Kapsel um ihre Längsachse, wobei das Hitzeschild leicht nach oben ragt. Dies ist notwendig, um ein schnelles Absinken in die dichteren Schichten der Atmosphäre zu verhindern.

Bei diesem Manöver muss der Bordcomputer wissen, was oben ist, um den Zielpunkt weit unten auf der Erde anpeilen zu können. Diese Berechnungen sind nicht besonders komplex, müssen aber mit der notwendigen Genauigkeit durchgeführt werden. Bei den Amerikanern waren derartige Berechnungen bei den Gemini- und Apollo-Programmen notwendig.

In diesem Fall kündigte der Autopilot der Soyuz der Besatzung überraschend an, dass er vergessen hatte, wo er sich befand. Die drei Passagiere konnten in diesem Augenblick wenig tun, sondern mussten diese Fehlermeldung einfach hinnehmen. Der Autopilot setzte ein alternatives Programm zur Landung ein. Allerdings war damit ein steiler Eintritt in die Erdatmosphäre verbunden, etwa wie bei einer ballistischen Rakete. Hinzu kam, dass durch ein Problem mit der Antenne die Besatzung in diesen kritischen Minuten nicht mit ihrer Bodenstation kommunizieren konnte.

Durch den steilen Abstieg zur Erde waren die Astronauten einer Belastung von bis zum achtfachen Wert der normalen Erdbeschleunigung (8 g) ausgesetzt. Ihre Zungen wurden weit in den Rachen gepresst, und sie hatten Schwierigkeiten mit der Atmung. Unter normalen Umständen beträgt die Belastung lediglich 4 g. Die Kapsel setzte 480 Kilometer entfernt vom erwarteten Landepunkt auf. Zur Suche nach der Raumkapsel wurden vier Helikopter vom Typ Mil Mi-8 sowie fünf Flugzeuge eingesetzt. Eine Antonov An-12 fand die Raumfahrer 2 Stunden und 20 Minuten nach der Landung. Es dauerte weitere 2 Stunden, bevor ein Helikopter die Crew bergen konnte. Pettit hatte der Abstieg so mitgenommen, dass er auf einer Trage in den Hubschrauber gebracht werden musste.

Bei der Luft- und Raumfahrt ist Software unverzichtbar. Wir sind inzwischen aber so weit, dass Software selbst in Chips einfließt. Mit anderen Worten: Selbst Hardware besteht zum Teil aus Software. Das zeigt das folgende Beispiel.

### Fall 1–6: Der Pentium Bug, noch ein Software-Fehler [14]

Im Sommer 1994 meldete Thomas Nicely, ein Professor für Mathematik am Lynchburg College im amerikanischen Bundesstaat Virginia, INTEL in Kalifornien einen Rechenfehler am neuesten Chip des Herstellers von Mikroprozessoren, dem PENTIUM. Im Silicon Valley nimmt man weder den Professor aus dem Süden der USA noch den Fehler so richtig ernst: INTEL beherrscht mit über 80 Prozent den Markt der Prozessoren für den PC, macht fabelhafte Umsätze und Gewinne. »INTEL inside« lautet der Werbespruch, mit dem die Kunden weltweit auf Prozessoren von INTEL verpflichtet werden sollen.

Monate später muss INTELs Management kleinlaut zugeben, dass ihr Mikroprozessor tatsächlich fehlerhaft ist. Zwar mag man darüber streiten, ob der Fehler nur alle 27 000 Jahre auftritt, wie INTEL behauptet, oder alle 24 Tage, wie IBM errechnet hat. Es bleibt ein Fehler, der sich bei vielen Anwendern zeigen kann. INTEL bietet schließlich allen Kunden den kostenlosen Umtausch des schadhafte Prozessors an und startet damit die bisher größte Rückrufaktion in der Computerbranche. Um die Kosten abzudecken, wird eine Rückstellung von 475 Mio. US\$ gebildet.

Dabei handelt es sich bei dem Fehler eigentlich um einen Irrtum in der Software. Um mit der verwendeten Architektur beim PENTIUM gegenüber den RISC-Architekturen der Konkurrenten bestehen zu können, entschloss man sich bei der Fließkommaarchitektur zur Verwendung einer *Lookup Table*. Das ist einfach eine Tabelle, wie man sie zum Beispiel in Logarithmentafeln findet. Bei diesem nach seinen Erfindern benannten STR-Algorithmus werden 1066 Werte benötigt, die zwischen -2 und +2 liegen. Ein Ingenieur berechnete diese Zahlen auf einem Computer und erstellte die benötigte Tabelle. Sie wurden mit einem *Script* in die Hardware, ein *Programmable Logic Array* (PLA), geladen.

Allerdings war das Script zum Laden der Werte selbst fehlerhaft. Ganze 5 von 1066 Zahlen wurden nicht übernommen. Diese Zellen hätten den Wert 2 enthalten sollen, blieben aber leer. Wenn die Floating Point Unit darauf zugreift, wird der Wert Null ausgelesen. Warum wurde man nun nicht früher auf den Fehler aufmerksam?

Das PLA wurde bei INTEL nicht auf korrekte Werte überprüft, und der PENTIUM ging damit in die Produktion. Im Feld fiel der Fehler deswegen nicht sofort auf, weil der Algorithmus rekursiv arbeitet. Zwar schaukelt sich der Fehler bei gewissen Rechnungen so stark auf, dass Rechenfehler in der vierten signifikanten Stelle einer Dezimalzahl auftreten können. In der Regel tritt die Abweichung aber erst in der neunten oder zehnten Stelle auf, und bei vielen Anwendungen spielt Dezimalarithmetik keine herausragende Rolle.

Bemerkenswert bleibt, dass es sich trotz des Gießens in Silicon letztlich um einen Fehler handelt, der in der Software begründet liegt.

Ein wichtiger Teil der Infrastruktur einer modernen Industriegesellschaft ist das Telefonnetz. Hier wird die höchste Zuverlässigkeit gefordert, nämlich eine Verfügbarkeit von nahezu 100 Prozent. Zuverlässigkeit heißt in diesem Fall nicht, dass jedes Gespräch notwendigerweise durchkommen muss. Die Nutzer würden gelegentlich ein abgebrochenes Gespräch tolerieren. Allerdings wird gefordert, dass das Telefonnetz als Ganzes jederzeit funktionsfähig bleiben muss.

In der Regel wird ein Netzwerk als recht zuverlässig und sicher gegen Ausfall betrachtet. Nicht zuletzt wegen dieser Eigenschaft wurde das Internet als Netzwerk konstruiert. Die Erfahrungen in den USA zeigen allerdings, dass durch Software eine neue Komponente in das Netz hereinkommen kann, die in Bezug auf die Ausfallsicherheit ein Risiko darstellt. Sehen wir dazu den folgenden Fall.

### **Fall 1–7: Dominoeffekt [15]**

Im Dezember 1989 installierte die größte amerikanische Telefongesellschaft, AT&T, ein neues und verbessertes Programm zur Vermittlung von Ferngesprächen in ihren Fernmeldeämtern. Am 15. Januar des folgenden Jahres signalisierte eine der Vermittlungsstellen einen Ausnahmezustand, sandte eine Nachricht an benachbarte Fernmeldeämter, dass keine weiteren Gespräche angenommen werden konnten und ging anschließend in den Initialisierungszustand. Nach einiger Zeit ging der Computer der Vermittlungsstelle wieder ans Netz.

Eine zweite Vermittlungsstelle im Netz von AT&T empfing die Nachricht von dem ersten Fehler und versuchte, sich selbst neu zu starten. Während dieser Initialisierungsphase wurde eine zweite Nachricht von der ersten Vermittlungsstelle empfangen. Dieses Signal konnte jedoch aufgrund eines Fehlers in der Software nicht richtig verarbeitet werden. Der Rechner der Vermittlungsstelle signalisierte seinen benachbarten Fernmeldeämtern einen Ausnahmezustand, sandte eine Nachricht an diese Vermittlungsämter, dass derzeit keine weiteren Telefongespräche angenommen werden konnten und ging anschließend in den Initialisierungszustand zurück. Nach einiger Zeit ging die Vermittlungsstelle wieder in Betrieb.

Obwohl der Rechner des Fernmeldeamts nach einiger Zeit wieder in Betrieb ging, pflanzte sich der Fehler im Netz der Telefonvermittlungen fort. Das Telefonnetz von AT&T im Norden der USA brach weitgehend zusammen. Dies gerade zu einer Zeit, als die Firma in großformatigen Anzeigen auf die besondere Zuverlässigkeit ihres Telefonservices hinwies!

Wie später eruiert wurde, trat der Fehler immer dann auf, wenn innerhalb von weniger als vier Sekunden eine weitere Nachricht in der Vermittlung eintraf. Letztlich wurde der Ausfall des Telefonnetzes auf einen Fehler in dem verwendeten Program, das in C geschrieben war, zurückverfolgt. Der Quellcode enthielt eine BREAK-Anweisung in einer IF-Abfrage, die wiederum Teil eines SWITCH-Statements war.

Die oben skizzierten Programmzeilen sind nicht gerade sehr gute Arbeit, aber im strikten Sinn verstoßen sie nicht gegen die Regeln von C. Ein Compiler wird sie also nicht bemängeln. Weil natürlich auf jedem Netzknoten dieselbe Software residieren wird, leidet die Ausfallsicherheit des Netzwerks. Das ist ein Nachteil, der durch das Eindringen der Software in diese Technik entsteht.

Wir sollten die Wichtigkeit der Verfügbarkeit des Telefonnetzes für eine moderne Gesellschaft nicht unterschätzen. Was wäre geschehen, wenn zur gleichen Zeit ein Kernreaktor Gefahr gelaufen wäre, durchzugehen? Kann es in solchen Fällen zu einer Panik unter der Zivilbevölkerung kommen?

Mancher mag einwenden, dass es noch den öffentlichen Rundfunk gibt. Das ist richtig. Aber es ist nicht gewährleistet, dass dieses Mittel auch genutzt wird. Bei der Flutkatastrophe an der Elbe im Sommer 2002 hätte der Rundfunk eingesetzt werden können, um die Bevölkerung zu warnen. Die verantwortlichen Politiker und Beamte haben auf diese Möglichkeit zur schnellen Unterrichtung der Bevölkerung verzichtet.

In anderen Fällen geht es nicht um Kommunikation, sondern um Geschäfte. Weil hier Geld im Spiel ist, kann der finanzielle Verlust mitunter beträchtliche Höhen erreichen.

### **Fall 1–8: Time-out für die Wall Street [16]**

Am 4. Juni 2001 war der Handel an der New York Stock Exchange, der weltweit wichtigsten Börse, für eineinhalb Stunden unterbrochen. Der Grund dafür lag in neu installierter Software. Obwohl an diesem Freitag der Handel später wieder aufgenommen werden konnte, stellte der Computerausfall doch einen schweren Schlag für die NYSE dar, die sich zunehmend in Konkurrenz zu anderen US-Börsen befindet.

Der Fehler in der Software führte dazu, dass der Handel um 10:10 Uhr Ortszeit unterbrochen werden musste. Es war das erste Mal seit dem Oktober 1998, dass der Handel ausgesetzt werden musste. Die meisten Firmen konnten ab 11:35 Uhr den Handel wieder aufnehmen, doch für rund 10 Prozent aller Aktien dauerte es noch Stunden, bevor sie wieder geordert oder verkauft werden konnten.

Richard Grasso, der Vorstandsvorsitzende der NYSE, erklärte, dass der Fehler durch die Software verursacht wurde, die in der Nacht zum Freitag eingespielt worden war. Die Börse wäre in der Lage gewesen, mit Ersatz-Computern den Handel in beschränktem Umfang aufrecht zu erhalten. Um aber gegenüber individuellen Investoren fair zu sein, wurde der gesamte Handel eingestellt. Große Unternehmen, darunter Fonds, wären wohl in der Lage gewesen, ihre Aufträge auch ohne die Computer der NYSE abzuwickeln.

In anderen Fällen schafft der Einsatz von Software unmittelbar Risiken für Menschen, ihre Gesundheit und ihr Leben. Das lässt sich mit dem folgenden Beispiel aus England demonstrieren.

### **Fall 1–9: Britische Fluglotsen können Computerschrift nicht lesen [66,67]**

Die Schrift auf den Computerbildschirmen im nagelneuen britischen Flugkontrollzentrum von Swanwick (Grafschaft Hampshire) ist so klein, dass Fluglotsen schon mehrfach Maschinen in die falsche Flughöhe dirigiert haben. Dies berichtete das Fachmagazin COMPUTER WEEKLY unter Berufung auf einen vertraulichen Report aus dem Zentrum.

Danach wurde in einem Fall ein für Glasgow (Schottland) bestimmtes Flugzeug nach Cardiff (Wales) geschickt, weil die Lotsen die Codenamen für die beiden Ziele auf dem Schirm nicht unterscheiden konnten. Wegen Software-Problemen war das Zentrum Swanwick ein ganzes Wochenende lang lahm gelegt worden. Stundenlange Verzögerungen und Flugausfälle waren die Folge.

Das für 623 Millionen Pfund (996 Millionen Euro) erbaute Zentrum mit dem Namen National Air Traffic Services (NATS) war im Januar 2003 in Betrieb genommen worden. Es wird privatwirtschaftlich geführt. Seitdem hat es drei Mal Probleme mit dem Computersystem gegeben.

Nach Angaben einer Fachzeitschrift sollen in Kürze Tests an einem neuen, besseren Schriftbild beginnen. Die zivile Luftaufsichtsbehörde (CAA) hat bestritten, dass das Computerproblem die Sicherheit des Luftraums gefährdet. Laut COMPUTER WEEKLY haben allerdings die Flugkontrolleure selbst das Problem als »sicherheitsbezogen« eingestuft.

Hier können wir ein Verhalten beobachten, das leider typisch ist für bestimmte Arten von Organisationen. Es sind gerade die Organisationen und Einrichtungen, die vorher nie Software eingesetzt haben, die bei ihrem ersten Versuch mit der neuen Technik auf die Nase fallen. Dagegen sind Organisationen, die mit der Software groß geworden sind, viel skeptischer. Das bewahrt sie zuweilen vor größeren Missgeschicken.

Die Teilstreitkräfte der USA, also die Armee, Marine und die Luftwaffe, haben den Einsatz von Software seit den 1950er Jahren in großem Umfang gefördert. Sie haben dabei durchaus Lehrgeld zahlen müssen, wie der folgende Fall beweist.

### Fall 1–10: Feind in Sicht? [17]

Am 3. Juli 1988 kreuzte das US-Kriegsschiff *Vincennes* im Arabischen Golf. Die Lage war explosiv: Der Iran und der Irak trugen ein paar hundert Meilen weiter nördlich einen blutigen Krieg aus. Rohöl war in diesem Konflikt eine nicht zu unterschätzende Waffe. Würde es dem Iran gelingen, den Strom des Rohöls aus dem Golf zum Erliegen zu bringen, so konnten sich daraus ganz erhebliche Schwierigkeiten für die Weltwirtschaft entwickeln. Die Industrieländer – allen voran natürlich die USA – waren deshalb stark daran interessiert, die Ölrouen zum Arabischen Golf offen zu halten. Die *USS Vincennes* war ein Teil der Streitmacht, die zur See diesem Ziel dienen sollte.

An diesem Tag war die Crew des Kreuzers in erster Linie mit einem Boot beschäftigt, das überprüft werden musste. Der Operator des Aegis-Raketenabwehrsystems tief im Inneren des Kriegsschiffes hielt einen startenden Airbus, Flug 655 der Iran Air, für einen Angriff auf das Schiff und leitete Abwehrmaßnahmen ein. Er verwechselte dabei den startenden Airbus, der rapide an Höhe gewann, mit einem F-14 Jagdflugzeug der iranischen Luftwaffe. Dieses Kampfflugzeug befand sich allerdings noch auf dem Runway, als der Airbus bereits in der Luft war.

Was den Operator hätte warnen können, war die zunehmende Höhe des vermeintlichen Angreifers. Ein Kampfflugzeug wäre zunächst tiefer geflogen. Obwohl der Operator im Kontrollzentrum an seinem Bildschirm die Höhe ablesen konnte, so wurde diese wichtige Information jedoch nicht auf dem Hauptbildschirm des Aegis-Systems übertragen. Der Kreuzer schoss eine Rakete ab. An Bord des iranischen Airbus starben 290 Menschen.

Hier kann man durchaus argumentieren, dass das Interface der Software zum menschlichen Benutzer fehlerhaft war. Es zeigte dem verantwortlichen Offizier nicht alle relevanten Informationen an.

Der Kalte Krieg liegt hinter uns. Er war nicht nur für Spione, sondern auch für die Zivilbevölkerung in den USA und Europa nicht ohne Risiken. Meistens haben wir von diesen

kritischen Momenten nichts erfahren, aber gelegentlich war die Panne so groß, dass sich der Vorfall nicht verheimlichen ließ.

### **Fall 1–11: Atomraketen im Anflug auf die USA [18]**

Es dauerte nur zwanzig Minuten, aber den Anwesenden erschien es wie eine Ewigkeit. Am 5. Oktober 1960 besichtigten der Präsident von IBM, Thomas J. Watson, sowie zwei weitere wichtige Unternehmensführer das Hauptquartier der North American Air Defense (NORAD) in der Nähe von Colorado Springs. Man zeigte ihnen die Kommandozentrale, in der Offiziere an ihren Pulten saßen und die Bildschirme beobachteten. An der Wand war eine riesige Karte angebracht, auf der große Teile Asiens und das Territorium der USA [3] zu sehen waren. Darüber waren Anzeigen montiert worden, die verschiedene Alarmstufen anzeigen konnten. Diese Anzeige war verbunden mit den Radaranlagen in Thule in Grönland, einem vorgeschobenen Beobachtungsposten in der Zeit des Kalten Krieges.

Man hatte den Besuchern erklärt, dass das Blinken einer Lampe bedeutete, dass sich nur wenige Objekte in der Luft befanden, die man erwartete. Mit anderen Worten: Das war Routine. Wenn zwei Lampen blinkten, befanden sich ein paar nicht identifizierte Objekte im Luftraum. Fünf Lampen standen dagegen für die höchste Alarmstufe. In so einem Fall befanden sich feindliche Atomraketen im Anflug auf Amerika.

Als die Gäste noch zuschauten, ging eine Lampe nach der anderen an. Als vier Lampen blinkten, kamen höhere Offiziere angerannt. Als fünf Lampen blinkten, komplimentierte man die Besucher hinaus. Sie wurden in ein Büro gebracht. Die drei Manager waren beunruhigt. Offenbar wurden die USA angegriffen, und sie hatten keinerlei Möglichkeit, ihre Familien zu warnen.

Zur selben Zeit befand sich General Laurence S. Kutter, der Befehlshabende von NORAD, in seiner C-118 in 4 500 Meter Höhe auf dem Rückweg nach Colorado Springs. Dieses Flugzeug war mit modernster Kommunikationstechnik ausgerüstet, und so gelang es seinem Stellvertreter, dem kanadischen Air Marshal C. Roy Slemon, rasch mit ihm Verbindung aufzunehmen. »Chef«, begann Slemon. »Das ist der Ernstfall. Wir bekommen viele Signale von unserem Frühwarnsystem herein.«

Kutter hörte aufmerksam zu. Es sah so aus, als hätte die Sowjetunion ihre Atomraketen von Sibirien aus gestartet. Der US-Generalstab wurde zu den Beratungen hinzugezogen. Slemon sprach mit den Offizieren in den Stäben in Washington, Ottawa und Nebraska. Das Strategic Air Command (SAC) musste seine Bomber und Raketen auf den Weg bringen, bevor sie am Boden zerstört werden konnten.

Die Lage war für vielleicht 20 Minuten äußerst gespannt. Bei NORAD stand man vor einer ungewohnten Situation. Es war zu 99,9 Prozent sicher, dass die Sowjetunion einen Angriff auf die USA gestartet hatte. Wenn das aber so war, mussten sich die Bahnen der Raketen verfolgen lassen, und inzwischen hätte man auch erkennen können müssen, welche Ziele in den USA die Russen angreifen wollten. Solche Bahnkurven gab es aber nicht.

Slemon wandte sich schließlich an den Verbindungsoffizier zum Geheimdienst bei NORAD. »Wo ist Chruschtschow?« rief er.

»In New York bei der Vollversammlung der UNO«, erwiderte Harris Hull.

Slemon überlegte, dass die Sowjetunion wohl kaum die USA angreifen würden, wenn der Vorsitzende der Kommunistischen Partei zu Besuch in den Vereinigten Staaten war. Er blies den Alarm ab.

Stunden später machte man sich beim US-Militär langsam ein Bild über die Ursache des Vorfalls. Was die Radaranlagen in Thule in Grönland entdeckt hatten, war der Mond, der

über Norwegen aufging. Bei der Entwicklung des Frühwarnsystems hatte niemand daran gedacht, dass dessen mächtiges RADAR ein Objekt wie den Mond, der ungefähr 443 000 Kilometer von der Erde entfernt war, als feindliches Objekt einstufen würde. Die Entwickler der Software rechneten mit Raketen in einer Entfernung von fünf- bis sechstausend Kilometern.

Das Programm war durch den Mond, ein sich langsam bewegendes Objekt, ziemlich verwirrt. Als die Radarwellen auf der Oberfläche des Mondes aufprallten und reflektiert wurden, interpretierte das System die Echos als weitere russische Raketen. Für das Frühwarnsystem begann ein massiver Angriff sowjetischer Atomraketen.

Hätte der Auftraggeber des Systems, das Department of Defense (DoD), etwas mehr Sorgfalt walten lassen, hätte es Objekte in großer Entfernung, etwa über einer Höhe von zehntausend Kilometern, als nicht feindlich eingestuft und bei der Zielerfassung ausgeschlossen. Dies hätte nicht einmal einen hohen Aufwand bei der Programmierung verursacht, denn RADAR steht für *Radio Detecting and Ranging*. Das heißt, dass mit derartigen Strahlen die Entfernung eines Objekts sehr leicht ermittelt werden kann. Diese Werte fallen routinemäßig an. Man hätte diese Messungen nur auswerten müssen, um den aufgehenden Mond von anfliegenden Atomraketen unterscheiden zu können.

So hätte also bereits in den 1960er Jahren ein Fehler in der Software beinahe dafür gesorgt, dass die Zivilisation, wie wir sie kennen, zum Untergang verdammt gewesen wäre. Zuweilen sind die mit Software zusammenhängenden Risiken durchaus lebensbedrohend. Nicht für den einzelnen Menschen, sondern sogar für alle Bewohner dieses Planeten.

## 1.4.1 Fehler ohne Zahl?

*Measurements are not to provide numbers but insight.*

*Ingrid Bucher*

Wer Fehler in der Software als unabänderlich hinnimmt, geht ein unkalkulierbares Risiko ein. Natürlich ist es nicht gerade die Gewohnheit der Anbieter, die Fehler in ihren Produkten herauszustreichen. Das sollte die Verantwortlichen für den Kauf und Einsatz von Software allerdings keinesfalls davon abhalten, sich ein Bild über das zu erwartende Risiko zu verschaffen.

In der Tat ist es so, dass wir Fehler in der Software berechnen können, wenn wir den Programmumfang kennen. Das ist gewiss keine exakte Wissenschaft, aber eine fundierte Schätzung dürfte für unsere Zwecke ausreichen. Neben den Fehlern, die im Verlauf der Entwicklung anfallen und durch Tests [19] gefunden werden, interessieren uns dabei in erster Linie die Restfehler. Das sind jene Fehler, die eben nicht gefunden werden und nach der Auslieferung der Software an den Kunden im Programm verbleiben.

Wir wollen deshalb kurz überschlagen, mit welchen Fehlerraten in Software zu rechnen ist. Dabei will ich Durchschnittswerte der Industrie zugrunde legen. Für die Entwicklung kann man dabei mit Fehlerraten [14] von 30 bis 50 Fehlern pro 1 000 Lines of Code rechnen.

Für ausgelieferten Code liegt diese Rate natürlich niedriger. Für gute Organisationen der Software-Industrie kann man derzeit mit 1 bis 3 Fehlern pro 1 000 Lines of Code rechnen. Mit Tabelle 1–3 können wir uns einen ersten Überblick verschaffen.

Programmumfang in Lines of Code [LOC]	Fehler während der Entwicklung	Restfehler	Schwerwiegende Restfehler
20 000	600 – 1 000	20 – 60	2 – 6
50 000	1 500 – 2 500	50 – 150	5 – 15
100 000	3 000 – 5 000	100 – 300	10 – 30
500 000	15 000 – 25 000	500 – 1 500	50 – 150
1 000 000	30 000 – 50 000	1 000 – 3 000	100 – 300

**Tabelle 1–3:** Fehler und Restfehler [14]

Während wir davon ausgehen können, dass der Großteil der Fehler in der Software beim Test am Schluss der Entwicklung gefunden und beseitigt wird, sind die Restfehler bei realistischer Betrachtung bei der Auslieferung der Software noch vorhanden. Allerdings ist nicht jeder Restfehler gleich ernsthafter Natur. Ein Tippfehler, der sich im Text eines Programms findet, bringt in der Regel kein Flugzeug zum Absturz. Aus diesem Grund können wir bei unserer Betrachtung davon ausgehen, dass rund zehn Prozent der Restfehler für das Programm ernsthafte Konsequenzen haben, bei ihrem Auftreten also zu einem Programmabsturz oder zu einer Verklemmung (*Deadlock*) führen können.

Die Restfehlerrate basiert dabei auf den Zahlen von Unternehmen, die derartige Statistiken erstellen, aufbewahren und veröffentlichen. Für den Rest der Branche dürften die Zahlen eher schlechter liegen.

Microsoft teilte seine Zwischenhändlern bei einer Konferenz hinter verschlossenen Türen im März 1998 mit, dass mit Windows 98 rund 5 000 Fehler in der Vorgängerversion Windows 95 beseitigt worden wären. Fünftausend Restfehler in einem Betriebssystem sind nicht gerade wenig. Da darf man sich als Anwender nicht wundern, wenn man täglich mit Problemen konfrontiert wird.

Nehmen wir als Beispiel das amerikanische Space Shuttle. Das Navigationssystem dieser Raumfähre enthält rund 500 000 Lines of Code (LOC). Wir können in diesem Fall gemäß Tabelle 1–3 also mit Restfehlern im Bereich von 50 bis 150 rechnen.

Das Space Shuttle ist durch seine Software nie in ernsthafte Gefahr geraten. Dabei ist allerdings zu bedenken, dass die Organisation, die dieses Programm liefert, eine der niedrigsten Restfehlerraten in der Industrie hat. Das bedeutet auf der anderen Seite nicht, dass diese Software fehlerfrei wäre. Mir ist zum Beispiel ein Fall bekannt, in dem die Kontrollzentrale in Houston die Besatzung des Space Shuttle zur Einleitung der Landung aufforderte. Der Kommandant leitete daraufhin das entsprechende Manöver ein.

Wenig später entschloss man sich in Houston wegen schlechten Wetters in Florida, die Landung zu verschieben. Dies wurde dem Kommandanten des Shuttle mitgeteilt. Als er den entsprechenden Befehl eingab, weigerte sich die Software, ihn anzunehmen. Im Programm

war nur *eine* Landung vorgesehen. Diese war bereits in Vorbereitung. Wie konnte da ein zweiter Befehl mit dem gleichen Inhalt akzeptiert werden?

Die Besatzung des Space Shuttle kam diesmal heil zur Erde zurück. Wir können allerdings bereits hier einen Schluss ziehen: Es ist besser, ein vorhandenes Risiko numerisch zu bestimmen, als einfach im Dunklen zu tappen.

## 1.5 Gesetzliche Anforderungen

*We are in bondage to the law in order that we may be free.*

*Cicero*

Ein guter Manager wird Risiken nicht außer Acht lassen, weil er dies als Teil seiner Aufgabe betrachtet. Es gehört einfach zum Job. Auf der anderen Seite darf nicht unerwähnt bleiben, dass der deutsche Gesetzgeber in jüngster Zeit die Bestimmungen in dieser Hinsicht verschärft hat.

Mit dem Artikel-Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG) wurden verschiedene Vorschriften im Wirtschaftsleben verschärft. Dieses Gesetz trifft für alle Wirtschaftsjahre zu, die nach dem 31. Dezember 1998 beginnen. In erster Linie gilt es für die Vorstände von Aktiengesellschaften. Auch die Geschäftsführer von Gesellschaften mit beschränkter Haftung (GmbHs) sind jedoch nach der einhelligen Meinung der Fachleute [20] betroffen. Damit ist davon auszugehen, dass alle Unternehmen durch das Gesetz zum Risikomanagement verpflichtet sind, die zwei der drei folgenden Kriterien erfüllen:

- Bilanzsumme über 3,44 Millionen Euro
- Umsatz über 6,87 Millionen Euro
- Mehr als 50 Mitarbeiter

Versäumt ein verantwortlicher Manager die Einrichtung eines geeigneten Systems zum Risikomanagement, so ist er im Schadensfall den Anteilseignern unter Umständen auch mit seinem persönlichen Vermögen haftbar. Hierzu bemerkt § 91, Absatz 2, des Gesetzes: »Der Vorstand hat geeignete Maßnahmen zu treffen, insbesondere ein Überwachungssystem einzurichten, damit den Fortbestand der Gesellschaft gefährdende Entwicklungen frühzeitig erkannt werden.«

Der deutsche Gesetzgeber hat zur Ausgestaltung des verlangten Risikomanagementsystems keine Vorgaben gemacht. Noch gibt es in der Bundesrepublik Deutschland auch keine Norm hierzu. Es ist allerdings damit zu rechnen, dass früher oder später eine deutsche Norm erarbeitet oder eine international geltende Norm zum Risikomanagement übernommen werden wird.

In unseren bewegten Zeiten werden wir nicht umhinkönnen, Risiken einzugehen. Dies gilt nicht zuletzt wegen der damit verbundenen Chancen. Man kann den Wettbewerb mit ein paar Rolltreppen vergleichen, die sich nach unten bewegen. Die verschiedenen Wettbewer-

ber stehen auf den Treppen. Wer sich nicht bewegt, wer nichts tut, der verliert schon alleine dadurch, dass sich die Treppe bewegt. Gewinnen kann nur derjenige, der sich aktiv nach oben orientiert.

Dazu gehört es, Chancen wahrzunehmen. Aber auch, die damit unweigerlich verbundenen Risiken aktiv zu verfolgen.