

Enterprise Resource Planning Software:
Der Einfluss der Usability auf die Total Cost of Ownership

DIPLOMARBEIT

zur Erlangung des akademischen Grades
einer Magistra
der Sozial- und Wirtschaftswissenschaften

eingereicht bei Herrn
A. Univ.-Prof.
Mag. Dr. Kurt Promberger

Institut für Strategisches Management, Marketing und Tourismus
Tätigkeitsbereich Verwaltungsmanagement
Fakultät für Betriebswirtschaft
der Leopold-Franzens-Universität Innsbruck

von
Katrin Kneissl
Traunstraße 44
4650 Lambach

Innsbruck, Oktober 2006

KURZFASSUNG

Enterprise Resource Planning (ERP) Software konnte, indem sie erhebliche Verbesserungen für die Integration der Geschäftsprozesse und damit verbundene Vorteile für Unternehmen ermöglicht, zu Beginn der 1990er Jahren wie auch in den letzten Jahren einen unglaublichen Erfolg am Softwaremarkt verbuchen. Gleichzeitig ist zu beobachten, dass im Zusammenhang mit der Einführung von ERP-Systemen beachtliche Probleme auftreten. Ein mangelhaftes Projektmanagement resultiert nicht selten in einem gescheiterten Einführungsprojekt. Konkrete Probleme stellen beispielsweise die Budgets bei weitem überschreitende Kosten oder ein unüberbrückbarer Widerstand der Mitarbeiter dar.

Beinahe unbeachtet blieb bislang der Zusammenhang zwischen der Usability (dt. Gebrauchstauglichkeit) einer betriebswirtschaftlichen Standardsoftware und ihrem erfolgreichen Einsatz in der Praxis. Nun führt eine mangelnde Software Usability nicht sogleich zum totalen Misserfolg eines Implementierungsprojekts; sie wirkt sich aber beträchtlich auf die Kosten der Einführung und der Benutzung einer ERP-Software aus - eine effiziente und effektive Erledigung von Aufgaben wird erschwert, die Erreichung von Zielen, die für den ERP-Einsatz formuliert wurden, eventuell verhindert.

Diese Arbeit versuchte einen Beitrag zur Beantwortung der Frage zu leisten, inwieweit eine schlechte Usability von ERP-Systemen Auswirkungen auf ihre Total Cost of Ownership (TCO, dt. Gesamtbetriebskosten) hat. Es wurden Benutzungsprobleme infolge mangelhafter Usability aufgezeigt; daraus wurden konkrete Usability-Anforderungen an ERP-Softwaresysteme abgeleitet. Überdies wurde der Frage nachgegangen, für welche Unternehmen die ERP-Usability eine besonders zentrale Rolle spielt.

INHALTSVERZEICHNIS

KURZFASSUNG	I
INHALTSVERZEICHNIS	III
ABBILDUNGSVERZEICHNIS	V
TABELLENVERZEICHNIS	VI
ABKÜRZUNGSVERZEICHNIS	VII
I EINFÜHRUNG	1
1 Problemstellung	2
2 Zielsetzung	3
3 Aufbau und Vorgehensweise	4
II SOFTWARE USABILITY	5
1 Begriffsabgrenzungen	7
1.1 User Interface	7
1.2 Human Computer Interaction	10
1.3 Software Ergonomie	12
1.4 Software Usability	13
2 User Centred Design	17
2.1 Prozess des User Centred Design	18
2.2 Prinzipien des User Centred Design	20
3 Usability Testing	25
3.1 Motive für den Einsatz von Usability Testing	25
3.2 Ziele von Usability Testing	26
3.3 Methoden zur Beurteilung der Software Usability	27

4	Nutzen von Software Usability	32
4.1	Nutzen für Anwenderunternehmen	33
4.2	Nutzen für Softwarehersteller	35
III	ENTERPRISE RESOURCE PLANNING UND TOTAL COST OF OWNERSHIP	38
1	Enterprise Resource Planning	38
1.1	Definition	38
1.2	Merkmale von ERP-Systemen	39
1.3	Eckdaten zum ERP-Markt	42
1.4	Nutzen des Einsatzes von ERP-Systemen	45
2	Total Cost of Ownership	52
2.1	Verwendung der TCO als Instrument des IT-Controllings	54
2.2	Ursprung der TCO	55
2.3	Beschreibung der TCO	56
2.4	Bewertung des TCO-Konzepts	62
IV	EINFLUSS DER USABILITY VON ERP-SYSTEMEN AUF IHRE TOTAL COST OF OWNERSHIP	65
1	Usability im ERP-spezifischen Kontext	66
1.1	Überblick über die bisherige Forschung	67
1.2	Anforderungen an die Usability von ERP-Systemen	71
2	Total Cost of Ownership im ERP-spezifischen Kontext	76
2.1	Allgemeines zu den TCO von ERP-Systemen	76
2.2	TCO-Modell für ERP-Systeme	79
2.3	Fallstudien über die Gesamtkosten von ERP-Software	82
2.4	Potentiale zur Senkung der TCO bei ERP-Systemen	86
3	Usability von ERP-Systemen und der Effekt auf die Total Cost of Ownership	87
3.1	Theoretische Grundlagen	87
3.2	Konkrete Auswirkungen der ERP-Usability auf die TCO	92
3.3	Relevanz der ERP-Usability in Anwenderunternehmen	95
V	ZUSAMMENFASSUNG UND AUSBLICK	98
	BIBLIOGRAFIE	100

ABBILDUNGSVERZEICHNIS

Abbildung 1: User Interface	7
Abbildung 2: Seeheim-Modell für User Interfaces	8
Abbildung 3: Human Computer Interaction	10
Abbildung 4: Anwendungsrahmen der Software Usability nach DIN EN ISO 9241-11	15
Abbildung 5: DIN EN ISO 13407 - Benutzerorientierte Gestaltung interaktiver Systeme	18
Abbildung 6: User Centred Design Prozess: Schritt 4	20
Abbildung 7: Kategorisierung der Usability-Normen	22
Abbildung 8: Verhältnis zwischen Software, Benutzer und Aufgabe	26
Abbildung 9: Task Based Testing	29
Abbildung 10: Kosten der Fehlerbeseitigung im Softwarelebenszyklus	36
Abbildung 11: Integration der verschiedenen Unternehmensbereiche in einem ERP-System	40
Abbildung 12: Typische Komponenten eines ERP-Systems	41
Abbildung 13: Prognostizierte Entwicklung des Markts für Enterprise Resource Planning	43
Abbildung 14: Verbreitung von ERP-Systemen	45
Abbildung 15: Die fünf wichtigsten Nutzenarten von SAP R/3 aus Sicht der Unternehmen	48
Abbildung 16: Informationstechnologie als Hebel für Prozessinnovationen	49
Abbildung 17: Strategien bei der Einführung eines ERP-Systems	50
Abbildung 18: Aufteilung der Total Cost of Ownership (TCO) auf ihre Basiskostenfaktoren	57
Abbildung 19: Direkte und indirekte Kosten	59
Abbildung 20: Kostenkategorien und ihre Berücksichtigung bei der ERP-Auswahl	77
Abbildung 21: Aufteilung der ERP-Implementierungskosten	78
Abbildung 22: Effizienz von Usability Engineering im Verhältnis zur Benutzerpopulation	88
Abbildung 23: Information Systems Success Model nach DeLone und McLean	91
Abbildung 24: Nutzen der ERP-Usability für Anwenderunternehmen	93
Abbildung 25: Wichtigkeit der Unterstützung gelegentlicher interner Nutzer für Unternehmen	97

TABELLENVERZEICHNIS

Tabelle 1: User's Bill of Rights nach Karat	6
Tabelle 2: Elemente der Benutzeroberfläche	8
Tabelle 3: Vorteile des Graphical User Interface (GUI)	9
Tabelle 4: Themen der Disziplin Human Computer Interaction	11
Tabelle 5: Usability-Komponenten	14
Tabelle 6: Vier Grundregeln des User Centred Design	17
Tabelle 7: Entwurfsprinzipien nach Preim	21
Tabelle 8: Internationale Normen zur Software Usability	24
Tabelle 9: Methoden des Usability Testing	27
Tabelle 10: Nutzen von Software Usability für Anwenderunternehmen	34
Tabelle 11: Nutzen von Software Usability für Softwarehersteller	37
Tabelle 12: Nutzenkategorien von ERP-Systemen	47
Tabelle 13: Gründe für ein betriebliches IT-Controlling	54
Tabelle 14: Total Cost of Ownership (TCO) - Direkte Kosten	60
Tabelle 15: Total Cost of Ownership (TCO) - Indirekte Kosten	61
Tabelle 16: Literatur zur Usability von ERP-Software	67
Tabelle 17: ERP-Benutzungsprobleme infolge mangelnder Usability	70
Tabelle 18: Anforderungen an die Usability von ERP-Software	75
Tabelle 19: TCO-Modell nach SAP - Grundlegende Kostenkategorien	80
Tabelle 20: Total Cost of Ownership (TCO) von ERP-Systemen - Direkte Kosten	81
Tabelle 21: Total Cost of Ownership (TCO) von ERP-Systemen - Indirekte Kosten	82
Tabelle 22: Direkte Kosten von ERP-Systemen - Ergebnisse der betrachteten Fallstudien	85
Tabelle 23: Usability Cost-Benefit Models: Nutzen der Usability für Anwenderunternehmen	90
Tabelle 24: Auswirkung der Usability auf die TCO einer ERP-Software	94

ABKÜRZUNGSVERZEICHNIS

ACM SIGCHI	Association for Computing Machinery, Special Interest Group on Computer-Human Interaction
BPR	Business Process Reengineering
bzw.	beziehungsweise
CIO	Chief Information Officer (dt. IT-Verantwortlicher)
CRM	Customer Relationship Management
dt.	deutsch
ERP	Enterprise Resource Planning
etc.	et cetera
FH	Fachhochschule
GUI	Graphical User Interface (dt. Grafische Benutzeroberfläche)
HCI	Human Computer Interaction (dt. Mensch-Computer-Interaktion)
i.S.	im Sinne
IS	Information System (dt. Informationssystem)
ISO	International Organization for Standardization
IT	Information Technology (dt. Informationstechnologie)
k.A.	keine Angabe
MA	Mitarbeiter
MIS	Management Information Systems (dt. Managementinformationssystem)
PC	Personal Computer (dt. Arbeitsplatzrechner)
ROI	Return on Investment (dt. Rendite)
SOA	Service-Oriented Architecture (dt. Service-orientierte Architektur)
TCO	Total Cost of Ownership (dt. Gesamtkosten einer Investition über den gesamten Lebenszyklus)
TU	Technische Universität
u.a.	unter anderem
UCD	User-Centred Design
UI	User Interface (dt. Benutzeroberfläche)
UIMS	User Interface Management System
UT	Usability Testing
z.B.	zum Beispiel

I EINFÜHRUNG

Enterprise Resource Planning (ERP) verspricht erhebliche Verbesserungen für die Integration der Geschäftsprozesse und damit verbundene Vorteile für Unternehmen, indem es Wertschöpfungsketten in Unternehmen sowie über Unternehmensgrenzen hinweg unterstützt. Die für ERP-Systeme in den verschiedensten Branchen getätigten Ausgaben sind beträchtlich - und nach wie vor im Wachsen begriffen: Bereits 1998 wurde jeder zweite weltweite Software-Dollar in ERP-Systeme investiert.¹ Der Umsatz des Marktes betrug 2004 etwa 21 Milliarden US-Dollar; darin sind ausschließlich die Umsätze der ERP-Softwarehäuser durch Lizenzen, Beraterleistungen und Wartungsdienstleistungen enthalten.² Rund 67,6 Prozent der österreichischen Großunternehmen vertrauen auf ein ERP-System; unter den kleineren und mittleren Unternehmen sind es 15,5 Prozent, die ihre Geschäftsprozesse mit einem ERP-System steuern. Durchschnittlich hat jedes fünfte österreichische Unternehmen ein ERP-System im Einsatz.³

Den überwältigenden Marktdaten zum Trotz treten im Zusammenhang mit der Einführung von ERP-Systemen beachtliche Probleme auf: 85 Prozent der Unternehmen geben in einer Befragung zum Erfolg von ERP-Projekten an, sie hätten ihre inhaltlichen Projektziele nicht erreicht; 28 Prozent der ERP-Projekte wurden von gravierenden Problemen gestoppt.⁴ Nach Studien, die Beispiele von misslungenen Implementierungen aufzeigen, muss man nicht erst lange suchen.⁵ Warum sich ERP-Projekte für Unternehmen derart problematisch gestalten und warum die Einführung von ERP-Systemen in Unternehmen oft sehr stiefmütterlich behandelt wird, ist unklar. Denn ERP-Systeme stellen nicht bloß eine große anfängliche Investition dar; sie verursachen über ihren gesamten Lebenszyklus enorme Kosten für das Unternehmen. Ein Grund für die Vernachlässigung von ERP-Projekten mag jener sein, dass sie in Unternehmen eben nicht an der Tagesordnung stehen. Durchschnittlich steht nur alle acht bis zehn Jahre im Unternehmen ein ERP-Implementierungsprojekt an. Nicht selten resultiert ein mangelhaftes Projektmanage-

¹ vgl. Corbitt/Mensching (2000): S. 7015

² vgl. Forrester Research (2005) nach Karcher (2005): S. 27

³ vgl. Bernroider/Hampel (2005): S. 307f

⁴ vgl. Sontow (2006a): S. 42

⁵ vgl. Davenport (1998); Martin (1998); Krasner (2000); Buckhout/Frey/Nemec (1999); Trunick (1999)

ment schließlich in einem gescheiterten Einführungsprojekt. Konkrete Probleme stellen dann beispielsweise die Budgets bei weitem überschreitende Kosten oder ein unüberbrückbarer Widerstand der Mitarbeiter dar. Allerdings kann eine gut überlegte Implementierung einer ERP-Software einen gewaltigen Nutzen für Unternehmen mit sich bringen. ERP-Softwarehäuser haben in zahlreichen Beispielen bewiesen, dass die Integration der Geschäftsprozesse in einer betriebswirtschaftlichen Standardsoftwarelösung die Basis für beachtliche Kosteneinsparungen bilden kann.⁶

Beinahe unbeachtet blieb bislang der Zusammenhang zwischen der Usability (dt. Gebrauchstauglichkeit) einer betriebswirtschaftlichen Standardsoftware und ihrem erfolgreichen Einsatz in der Praxis. Während zur Software Usability im Allgemeinen viele Arbeiten existieren, findet die Usability von ERP-Systemen in der gegenwärtigen Forschung nur wenig Beachtung. Die Literatur zu diesem Thema, mag sie auch dürftig sein, stimmt jedoch darin überein, dass die Usability auch im Fall einer ERP-Software eine wesentliche Determinante für einen erfolgreichen und kosteneffizienten Software-Einsatz darstellt.

1 Problemstellung

“People who use business software might dispise it, but they are paid to tolerate it. This changes the way people think about software. Getting paid for using software makes users far more tolerant of its shortcomings because they have no choice, but it doesn't make it less expensive. Instead - while the costs remain high - they become very difficult to see and account for.”⁷

Ziele bezüglich der Projektkosten sind jene, die am seltensten eingehalten werden: Die Budgets übersteigende Kosten sind keine Seltenheit bei ERP-Einführungsprojekten.⁸ Unter den 1.017 Unternehmen, welche an der Schweizer ERP-Zufriedenheitsstudie 2005 teilnahmen, gaben 13,41 Prozent höhere als geplante Kosten als ein Hauptproblem während der Implementierung an.⁹ Für ERP-Projekte ist es nicht untypisch, das Budget bei weitem zu überschreiten: Die *Standish Group* berichtet in einer Studie darüber, dass Kosten in der Höhe von etwa 180 Prozent der budgetierten Kosten keine Seltenheit sind.¹⁰ Im Rahmen derselben Studie wurden die dringlichsten Probleme der Anwenderunternehmen erhoben; fast 17 Prozent waren der Meinung, die ERP-Betriebskosten wären zu hoch; als größeres Problem wurde lediglich die mangelnde Flexibilität von ERP-Software wahrgenommen.¹¹

Doch wie sollten Unternehmen in der Auswahlphase, nachdem mehrere ERP-Systeme evaluiert und bereits jene ausgeschlossen wurden, welche nicht die notwendigen Funktionalitäten zur Verfügung stellen, die übrig gebliebenen Softwareprodukte untereinander vergleichen, wenn die

⁶ vgl. Bishu/Kleiner/Drury (2001): S. 148

⁷ Cooper (2004): S. 40

⁸ vgl. Knolmayer/von Arb/Zimmerli (1997): S. 50

⁹ vgl. ERP-Zufriedenheitsstudie 2005 Schweiz nach Sontow (2006a): S. 43

¹⁰ vgl. Parry (2005): S. 22

¹¹ vgl. ERP-Zufriedenheitsstudie 2005 Schweiz nach Scherer (2006): S. 6

wahren Kosten für sie verborgen bleiben. Eine Berechnung der Kosten, die mit der Anschaffung und dem Betrieb eines ERP-Systems einhergehen, ist nur möglich, wenn etwa der Trainings- und Supportaufwand oder die Auswirkung des ERP-Einsatzes auf die Produktivität der End-User bekannt sind. Diese die Kosten beeinflussenden Faktoren können allerdings nur sinnvoll abgeschätzt werden, wenn bekannt ist, wie es um die Usability des ERP-Systems steht. Und das ist in Unternehmen, die nach einem neuen ERP-System Ausschau halten, zumeist eben nicht der Fall.

Stellt die Softwareergonomie zwar mitunter ein Kriterium für die Auswahlentscheidung dar, ist man sich den folgeschweren Auswirkungen allzu oft nicht bewusst. Nun führen Probleme mit der Usability der ERP-Software freilich nicht direkt zum totalen Projektmisserfolg, sie wirken sich aber mit Sicherheit auf die Produktivität der einzelnen Anwender und somit des Teams als Ganzes aus. „Usability is not a luxury but a basic ingredient in software systems: People's productivity and comfort relate directly to the usability of the software they use.“¹² Eine mangelhafte Usability erschwert eine effiziente und effektive Erledigung von Aufgaben und in der Folge die Erreichung von Zielen, die durch den ERP-Einsatz sichergestellt werden sollten.¹³

Somit birgt die Usability von Softwareprodukten auch ein enormes volkswirtschaftliches Potential: Das Studieren von Handbüchern, der Schulungsaufwand, das Einarbeiten in Softwaresysteme - all das würde durch eine bessere Software Usability effizienter werden. Ebenso würden adaptive Benutzerschnittstellen, die den aktuell angebotenen Funktionsumfang jeweils der Aufgabe anpassen, die der Benutzer gerade erledigen möchte, die Kosten der Softwarenutzung deutlich reduzieren - bis dorthin ist allerdings noch ein weiter Weg in der Entwicklung von ERP-Systemen zurückzulegen.¹⁴

2 Zielsetzung

Das Thema der vorliegenden Arbeit, der Einfluss der Usability von ERP-Systemen auf ihre Gesamtkosten, wird in Expertenkreisen nicht erst neuerdings als höchst brisant eingestuft. So kann man in einem Newsletter der *Special Interest Group on Human-Computer Interaction* der *Association for Computing Machinery (ACM SIGHCI)* die Frage finden, inwieweit eine schlechte Usability Auswirkungen auf die Total Cost of Ownership (TCO, dt. Gesamtbetriebskosten) hat.¹⁵ Nur mit dem Wissen um die Auswirkungen auf die Gesamtkosten ist es für Softwarehersteller und Usability-Experten möglich, das Kosten-Nutzen-Verhältnis von Maßnahmen zur Unterstützung und Verbesserung der Usability umfassend und sinnvoll zu beurteilen. Bislang wurde das Thema allerdings weder in Theorie noch Praxis genügend behandelt. Vor allem in der Praxis bleibt die Tatsache, dass sich die Usability direkt auf die Total Cost of Ownership auswirkt, meist völlig unbeachtet.

¹² Juristo/Windl/Constantine (2001): S. 20

¹³ vgl. Topi/Lucas/Babaian (2005): S. 128

¹⁴ vgl. o.V. (2006): S. 8

¹⁵ vgl. ACM SIGHCI (2003), online

Diese Arbeit sollte einen Beitrag dazu leisten, dass die ERP-Usability und ihre Wirkung auf Kosten und Erfolg von ERP-Systemen mehr Beachtung finden. Indem sie den bisherigen Erkenntnisstand zur ERP-Usability durchleuchtet und auf konkrete monetäre Effekte der ERP-Usability für Anwenderunternehmen hinweist, ist ein erster Schritt in diese Richtung getan.

3 Aufbau und Vorgehensweise

Um die zentralen Fragestellungen zu diskutieren, nämlich welche Usability-Anforderungen an ein ERP-System gestellt werden sollten, wie die ERP-Usability die Kosten für die Einführung und den Betrieb eines ERP-Systems beeinflusst, und für welche Unternehmen die ERP-Usability in der Folge eine besonders zentrale Rolle spielt, bedarf es einiger Grundlagen, die im Kapitel II Software Usability sowie im Kapitel III Enterprise Resource Planning und Total Cost of Ownership aufgearbeitet werden sollten. Grundlage für die Bearbeitung der in dieser Arbeit besprochenen Fragestellungen stellt eine fundierte Literaturanalyse dar.

Den Beginn von Kapitel II Software Usability bilden begriffliche Definitionen; User Interface, Human Computer Interaction, Software Ergonomie und Software Usability werden beschrieben und voneinander abgegrenzt. Auf User Centred Design wird in der Folge eingegangen, ebenso auf seinen zentralen Bestandteil das Usability Testing. Abschließend werden die Nutzeneffekte einer ausgereiften Software Usability für Anwenderunternehmen und Softwarehersteller erörtert.

Grundlagenwissen zu betriebswirtschaftlicher Standardsoftware sowie zu den Gesamtkosten des Einsatzes von Softwaresystemen findet sich in Kapitel III Enterprise Resource Planning und Total Cost of Ownership. Zunächst wird auf grundlegende Merkmale von ERP-Systemen eingegangen. Anschließend werden der ERP-Markt sowie Nutzenpotentiale des ERP-Einsatzes beleuchtet. An die Gesamtkosten einer IT-Komponente nähert sich die Arbeit, indem sie auf die Eignung der TCO als Instrument des IT-Controllings hinweist. Neben einer Auseinandersetzung mit dem Ursprung der TCO werden die Gesamtkosten mit ihren Kostenkategorien ausführlich beschrieben. Eine Bewertung des TCO-Konzepts bildet den Abschluss.

Im Rahmen von Kapitel IV Einfluss der Usability von ERP-Systemen auf ihre Total Cost of Ownership wird auf der Basis des gesammelten Grundlagenwissens versucht, die zentralen Fragestellungen dieser Arbeit zu beantworten. Der erste Teilabschnitt beschäftigt sich mit der ERP-Usability; es wird ein Überblick über die zu diesem Thema verfügbare Literatur gegeben, ebenso werden Anforderungen an die Usability von ERP-Software festgelegt. Die Gesamtkosten von betriebswirtschaftlicher Standardsoftware werden daran anschließend thematisiert. Auch an dieser Stelle findet sich eine Zusammenfassung von wesentlicher Literatur. Ferner wird ein TCO-Modell für ERP-Systeme entwickelt. Fallstudien zu den TCO von ERP-Software sowie eine Erörterung der Potentiale zur Senkung der ERP-Gesamtkosten finden ebenso ihren Platz. Wie sich der Effekt der ERP-Usability auf die TCO konkret gestaltet und inwieweit die Usability für ein bestimmtes Anwenderunternehmen relevant ist, wird im dritten Teilabschnitt geklärt.

Um den Lesefluss nicht zu beeinträchtigen, wird bei den Personenbezeichnungen die männliche Form verwendet. Die entsprechende weibliche Form ist aber stets mitgedacht.

II SOFTWARE USABILITY

Bei der Entwicklung von Softwaresystemen rückte die Qualität der Interaktion zwischen Mensch und Computer zunehmend in den Fokus der Softwareanbieter. Das lässt sich alleine an der Tatsache messen, dass der Anteil des User Interface (UI; dt. Benutzerschnittstelle) an der Software ständig angewachsen ist. Dies bezieht sich sowohl auf den Kodierungsumfang¹⁶ als auch auf den zeitlichen Aufwand bei der Entwicklung der Software. Die Qualität der Benutzerschnittstelle ist in einem kompetitiven Markt für die Auswahl von Softwaresystemen relevant.¹⁷ Die Akzeptanz einer Software bei den tatsächlichen wie potentiellen Anwendern bestimmt sich neben ihrem Funktionsumfang immer mehr durch ihre Nutzungsqualität. Eine hohe Qualität des User Interface ist demzufolge neben der Aufgabenangemessenheit¹⁸ und einem stabilen Betrieb eine entscheidende Determinante für den Erfolg einer Software.¹⁹ Den Anwendern ist es kaum möglich zwischen Funktionalität und User Interface zu unterscheiden. Nehmen die Anwender das User Interface als gebrauchstauglich wahr, so werden sie auch das gesamte Softwaresystem als gebrauchstauglich einschätzen.²⁰ Software Usability, oder Gebrauchstauglichkeit, wie sie in der deutschen Übersetzung genannt wird, dient als Maßstab zur Bestimmung der Nutzungsqualität einer Software.²¹

Der Stellenwert, den Software Usability heute in der Entwicklung von Softwaresystemen genießt, hat sich anfänglich daraus entwickelt, dass die Benutzer ihrem Wunsch nach einer verbesserten Software Ergonomie zunehmend Ausdruck verliehen. Der Druck von Benutzerseite diente gleichsam als Katalysator für die breite Akzeptanz von Usability. Bis Mitte der 1980er Jahre setzten sich ergonomische Erkenntnisse kaum durch. Die Benutzer waren mit unzuverlässigen Systemen und langen Antwortzeiten konfrontiert. Erheblich zum Usability-Fokus der Herstellerfirmen beigetragen hat die zunehmende Nachfrage nach Produkten mit

¹⁶ vgl. ACM SIGCHI (1996), online; Usability Forum (o.J. a), online

¹⁷ vgl. Holzinger (2001b): S. 16

¹⁸ Die Aufgabenangemessenheit ist gegeben, sofern die Software die Erledigung einer Aufgabe des Anwenders unterstützt. Vgl. Rohr (1988): S. 27

¹⁹ vgl. Preim (1999): S. 1f; ACM SIGCHI (1996), online

²⁰ vgl. Dray (1995): S: 18

²¹ vgl. Nielsen nach Eichinger (o.J. a), online

verbesserter Ergonomie. Beispielhaft sei hier die Initiative des Interessensverbands der IBM-Anwender, *SHARE*, genannt: Sie stellten im *Report of the Interactive Systems Task Force* konkrete Forderungen an *IBM*. Darin machten sie deutlich, dass die Benutzerfreundlichkeit wichtiger sei als die Funktionen an sich. Angesichts der Tatsache, dass zukünftig Softwarebenutzer nicht ausschließlich EDV-Experten wären, sollten die Systeme konsistent, integriert, robust, leistungsfähig, konfigurierbar, leicht zu erlernen und zu benutzen sein.²²

Karat, Human Interface Designerin am *Thomas J. Watson Research Center* der *IBM*, verlieh ihrer Forderung nach einer erhöhten Usability von Software-Produkten in ihren User's Bill of Rights Ausdruck, die in Tabelle 1 wiedergegeben sind.²³

Tabelle 1: User's Bill of Rights nach Karat²⁴

I. The user is always right. If there is a problem with the use of the system, the system is the problem, not the user.	VI. The user has the right to a system that provides clear, understandable, and accurate information regarding the task it is performing and the progress toward completion.
II. The user has the right to easily install software and hardware systems.	VII. The user has the right to be clearly informed about all system requirements for successfully using software or hardware.
III. The user has the right to a system that performs exactly as promised.	VIII. The user has the right to know the limits of the system's capabilities.
IV. The user has the right to easy-to-use instructions for understanding and utilizing a system to achieve desired goals.	IX. The user has the right to communicate with the technology provider and receive a thoughtful and helpful response when raising concerns.
V. The user has the right to be in control of the system and to be able to get the system to respond to a request for attention.	X. The user should be the master of software and hardware technology, not viceversa. Products should be natural and intuitive to use.

Der Lobbyismus von Benutzerseite artikulierte anfänglich die Forderung nach möglichst einfachen Systemen. Die heute übliche Definition von Software Usability erhebt die leichte Handhabbarkeit eines Systems jedoch nicht zur Maxime in der Softwareentwicklung. Lässt ein Softwareprodukt an Effektivität und Effizienz zu wünschen übrig, so wird der Benutzer die Software nicht als gebrauchstauglich empfinden, selbst wenn sie leicht zu bedienen ist. Irrführend ist die Annahme vom ahnungslosen Benutzer. Im Gegenteil, Benutzer erledigen meist Routineaufgaben und wollen dabei angemessen unterstützt werden.²⁵ Studien belegen dies auch: Usefulness (dt. Zweckmäßigkeit, Aufgabenangemessenheit) hat auf den faktischen Gebrauch größere Auswirkungen als Ease of Use (dt. Benutzerfreundlichkeit).²⁶

Einführend finden sich in diesem Kapitel zunächst terminologische Abgrenzungen; die Begriffe User Interface, Human Computer Interaction, Software Ergonomie und Software Usability

²² vgl. Hellbardt (2000), online

²³ vgl. Wildstrom (1998): S. 18

²⁴ eigene Darstellung, Daten entnommen aus Karat (o.J.) nach Wildstrom (1998): S. 18

²⁵ vgl. Maaß (1993): S. 194

²⁶ vgl. Au/Ngai/Cheng (2002); Gelderman (1998); Adams/Nelson/Todd (1992); Delone/McLean (1992)

werden beschrieben und voneinander abgegrenzt. Daran anschließend werden der Prozess sowie die Prinzipien und Methoden des User Centred Design thematisiert. User Centred Design zielt darauf, Systeme von möglichst hoher ergonomischer Güte zu entwickeln. Zentraler Bestandteil des User Centred Design ist das Usability Testing, das Methoden und Verfahren zur Überprüfung von Software Usability zur Verfügung stellt. Schließlich werden die Nutzeneffekte einer qualitativ hochwertigen Software Usability diskutiert - für Anwenderunternehmen wie Softwarehersteller.

1 Begriffsabgrenzungen

Das User Interface stellt für den Software-User den sichtbaren Teil des Systems dar; was dahinter an technischer Perfektion liegen mag, die überwiegende Mehrheit der Endbenutzer hat davon keine Ahnung. Deshalb ist die Bedeutung des User Interface für die vom Benutzer erfahrene Nutzungsqualität, die Software Usability, einzigartig.²⁷

Einleitend werden wesentliche Charakteristika des User Interface umrissen. Daran anknüpfend wird ein Einblick in jenes Teilgebiet der Informatik gegeben, welches sich mit der Kommunikation zwischen Mensch und Maschine befasst, die Human Computer Interaction (HCI). Es stellt die Grundlagen für die Forschung zur Software Usability bereit. Das mit der HCI verwandte Fach der Ergonomie wird anschließend vorgestellt, wobei sich die Erläuterungen dazu auf die Ergonomie von Softwareprodukten beschränken. Abschließend wird der Begriff Software Usability definiert.

1.1 User Interface

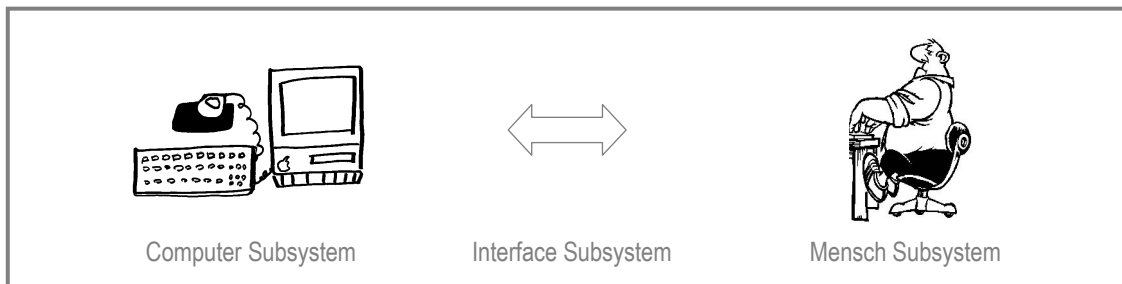


Abbildung 1: User Interface²⁸

Das User Interface (UI) bildet die zentrale Schnittstelle zwischen Mensch und Computer (siehe Abbildung 1).²⁹ Dabei geht es nicht ausschließlich darum, wie sich die visuelle Oberfläche des Softwaresystems gestaltet. Das UI umfasst alle Aspekte des Systemdesigns, welche die Interaktion zwischen Anwender und System beeinflussen, beispielsweise die Aufgabenangemessenheit des Systems, die Steuerung, die Navigation im System und die Integration

²⁷ vgl. Holzinger (2006a), online

²⁸ modifiziert nach Holzinger (2001a): S. 87

²⁹ vgl. Holzinger (2001a): S. 87

verschiedener Anwendungen.³⁰ In der Benutzeroberfläche integriert finden sich funktionale und ästhetische Elemente, welche die Interaktion zwischen Mensch und Maschine erleichtern. In Tabelle 2 wird auf diese näher eingegangen.

Tabelle 2: Elemente der Benutzeroberfläche³¹

	Beschreibung	Regel
Orientierungselement	Dient dem leichteren Zurechtfinden auf der Oberfläche. Beispiele: Übersichten, Gliederungen, Aufzählungszeichen, Hervorhebungen, Farbbereiche	Benutzer muss erkennen können, wo er sich befindet und aus welchen Bereichen die Software besteht.
Navigationselement	Ermöglicht dem Benutzer, sich auf dem User Interface zu bewegen und gezielt bestimmte Bereiche anzuspringen. Beispiel: Navigationsleiste	Die Navigation soll konsistent, logisch, übersichtlich und rasch sein.
Inhaltselement	Präsentiert Informationen, welche dem Benutzer vermittelt werden sollen. Beispiele: Texte, Bilder, Töne, Videos, Animationen	Die Inhaltselemente müssen für das System aufbereitet sein. Text sollte etwa kurz und prägnant sein.
Interaktionselement	Befähigt den Benutzer, bestimmte Aktionen zu erledigen. Beispiele: Auswahlmenüs, Buttons	Ein Interaktionselement muss den intuitiven Erwartungen des Benutzers entsprechen.
Motivationselement	Wirkt fördernd für die Benutzerakzeptanz und die Freude am Benutzen der Software. Beispiele: Bilder, Animationen	Ein Motivationselement soll die Neugier der Benutzer steigern.

Für die softwaretechnische Realisierung von User Interfaces existieren verschiedene Modelle:³² Unter den Schichtenmodellen ist das älteste und bekannteste das Seeheim-Modell, welches das UI in drei Schichten aufteilt (siehe Abbildung 2). Daneben gibt es Gerätemodelle, bei welchen Ein- und Ausgabegeräte im Vordergrund stehen. Demgegenüber begreifen objektorientierte Modelle das User Interface als ein System von Interaktionsobjekten.

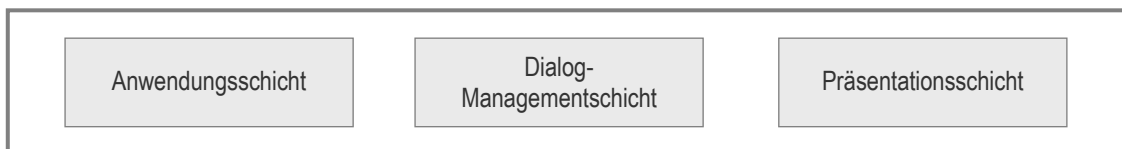


Abbildung 2: Seeheim-Modell für User Interfaces³³

³⁰ vgl. Dray (1995): S. 18

³¹ eigene Darstellung, Daten entnommen aus Holzinger (2006a), online

³² vgl. Preim (1999): S. 276ff

³³ modifiziert nach Preim (1999): S. 277

Diese Modelle bilden die Grundlage für User Interface Management Systems (UIMS), Konstruktionswerkzeuge zur Unterstützung der Entwicklung von Benutzerschnittstellen.³⁴ Mit Hilfe von UIMS können einerseits das Layout, andererseits die Handhabung, also sowohl statische wie auch dynamische Komponenten von Softwaresystemen, spezifiziert werden.³⁵

Erst als Graphical User Interfaces (GUI's) kostengünstig hergestellt werden konnten, nahm ihre Bedeutung für Softwaresysteme zu. Mittlerweile ist ein gebrauchstaugliches User Interface unerlässlich.³⁶ GUI's „bezeichnen eine Generation Benutzerschnittstellen, die sich aus der Idee der Visualisierung von Daten- und Kontrollmanipulation entwickelt hat.“³⁷ Auch WIMP-Schnittstellen (Windows, Icon, Menue, Pointing Device) genannt, bestehen sie aus einer visuellen Anzeigeeinheit, Interaktionsformen - basierend auf Fenstern, Piktogrammen und Menus - und einem Zeigehilfsmittel. Zur visuellen Anzeigeeinheit zählen ein Bildschirm zur Anzeige von Informationen sowie eine Tastatur zur Eingabe von Kontroll- und Manipulationsfunktionen sowie Daten. Die Vorteile von GUIs sind in Tabelle 3 aufgelistet.

Tabelle 3: Vorteile des Graphical User Interface (GUI)³⁸

Hochauflösender Bildschirm	Präsentation und Manipulation von grafischer Information
Fenstersystem	Kontrolle mehrerer, einander überlappender Bildschirmausschnitte; Bewegung in unterschiedlichem Kontext mit geringem Interaktionsaufwand
Piktogramm	Vereinfachte Darstellung für Daten-, Manipulations- und Kontrolleingaben
Menü	Einfache Auswahl von Eingaben aus einer Liste von Befehlen
Zeigegeräte	Auffinden, Selektieren und Manipulation von Information

Der Anteil des UI an der Entwicklung eines Softwaresystems, was Dauer und Kosten betrifft, ist ständig im Wachsen begriffen.³⁹ Daran lässt sich die Bedeutung erkennen, die der Benutzerschnittstelle heute beigemessen wird: Durchschnittlich beträgt der Anteil des UI am Sourcecode 47 bis 60 Prozent.⁴⁰ Knapp ein Drittel des Softwareentwicklungsbudgets ist für das Design des UI reserviert⁴¹, zeitlich beträgt der Anteil des Designs 40 Prozent des gesamten Entwicklungsprojekts.⁴²

Das UI bildet den grundlegenden Gestaltungsgegenstand in den Disziplinen Human Computer Interaction und Software Ergonomie⁴³, auf welche im Folgenden näher eingegangen wird.

³⁴ vgl. Preim (1999): S. 534; Hoppe (1988): S. 280

³⁵ vgl. Stary (1994): S. 291

³⁶ vgl. Holzinger (2001a): S. 79; Dray (1995): S. 18

³⁷ Stary (1994): S. 132

³⁸ eigene Darstellung, Daten entnommen aus Stary (1994): S. 132f

³⁹ vgl. Dray (1995): S. 18

⁴⁰ vgl. MacIntyre/Estep/Sieburth (1990) nach Karat (1994): S. 47; ACM SIGCHI (1996), online; Usability Forum (o.J. a), online

⁴¹ vgl. Rosenberg (1989) nach Karat (1994): S. 47f

⁴² vgl. Wixon/Jones (1992) nach Karat (1994): S. 48

⁴³ vgl. Maaß (1993): S. 199

1.2 Human Computer Interaction

Die Forschungsdisziplin der Human Computer Interaction (HCI, dt. Mensch-Computer-Interaktion) widmet sich dem Zusammenspiel zwischen und der Gestaltung der Interaktionsformen von Mensch und Computer. Zentrales Element einer Interaktion ist, dass der Anwender mit dem Computer interagiert, mit dem Zweck eine bestimmte Aufgabe zu erledigen.⁴⁴

Es handelt sich dabei um ein relativ junges Fachgebiet, das erst ab 1980 mit der Verbreitung von grafischen Benutzeroberflächen (Xerox Star 1983, Apple Lisa 1985) entstand.⁴⁵ Die Interessensgruppe *Special Interest Group on Computer Human Interaction* der *Association for Computing Machinery* (ACM SIGCHI) definiert HCI folgendermaßen:

„Human computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.“⁴⁶

Die Interaktion zwischen Mensch und Computer wird in Abbildung 3 dargestellt. Die daran anschließende Tabelle 4 soll zu einem besseren Verständnis beitragen.

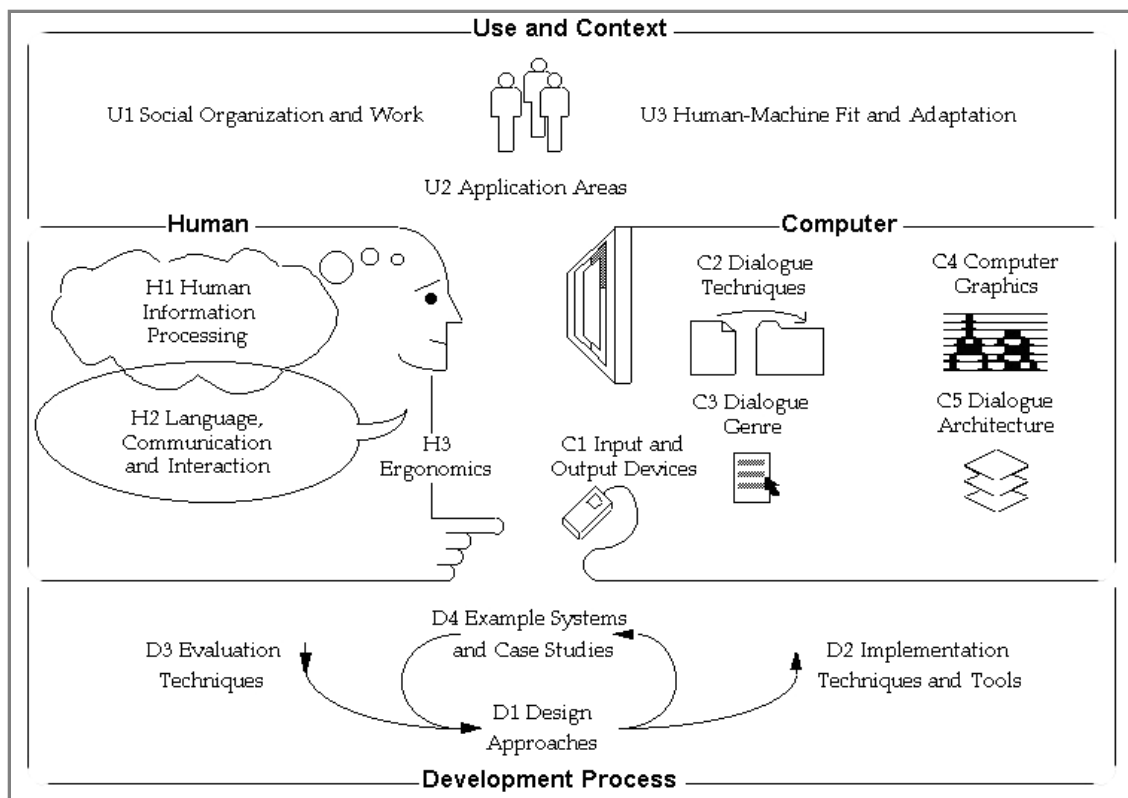


Abbildung 3: Human Computer Interaction⁴⁷

⁴⁴ vgl. Dix et al. (1993): S. 3; Geiser (1990): S. 9

⁴⁵ vgl. Holzinger (2006a), online

⁴⁶ ACM SIGCHI (1996), online

⁴⁷ Quelle: ACM SIGCHI (1996), online

Tabelle 4: Themen der Disziplin Human Computer Interaction⁴⁸

Use and Context	Die Anwendungen (Application Areas), welche mit Hilfe der Software ausgeführt werden, sind für das User Interface Design von entscheidender Bedeutung. Daneben können auch der soziale und organisationale Kontext (Social Organization and Work), beispielsweise das Image des Unternehmens, Einfluss auf das User Interface haben. Ziel der Gestaltung des User Interface ist die Entwicklung einer passenden Schnittstelle; das schließt allenfalls auch notwendige Anpassungen (Human Machine Fit and Adaptations) mit ein.
Human Characteristics	Grundlagen menschlicher Informationsverarbeitung (Human Information Processing) sowie menschlicher Kommunikation und Interaktion (Language, Communication and Interaction), aber auch das Wissen um anthropometrische und physiologische Charakteristika (Ergonomics) sind für die Disziplin der HCI von entscheidender Bedeutung.
Computer System and Interface Architecture	Computer sind mit speziellen Komponenten ausgestattet, um die Interaktion zu ermöglichen. Dazu zählen Ein- und Ausgabegeräte (Input and Output Devices), Dialogtechniken (Dialogue Techniques), Dialoggattung (Dialogue Genre), grafische Datenverarbeitung (Computer Graphics) und Dialogarchitektur (Dialogue Architecture).
Development Process	Die Gestaltung von User Interfaces beinhaltet gestalterische (Design Approaches) wie auch technische Aspekte (Implementation Techniques and Tools). Eine Evaluation der Benutzerschnittstelle (Evaluation Techniques) kann Verbesserungspotentiale aufdecken. Ebenso nützlich für die Gestaltung von User Interfaces erweisen sich Beispielsysteme und Fallstudien (Example Systems and Case Studies).

HCI als interdisziplinäres Fachgebiet verknüpft Erkenntnisse aus den Bereichen der Informatik - im Besonderen die Gestaltung von Anwendungen und User Interfaces -, der Psychologie - speziell aus der Kognitionswissenschaft und der Analyse des Anwenderverhaltens -, der Soziologie und der Anthropologie sowie des Industrial Design von Softwaresystemen. HCI vereint somit wissenschaftliche, technische und gestalterische Aspekte.

Die Fragestellungen, mit denen sich die Disziplin HCI auseinandersetzt, betreffen die Leistungsfähigkeit von Mensch und Computer, die Struktur der Kommunikation zwischen Anwender und Computer wie auch menschliche Fähigkeiten zur Benutzung von Maschinen, beispielsweise die Erlernbarkeit von User Interfaces. Algorithmen für und Programmierung von User Interfaces sowie weitere Aspekte des Software Engineering bezüglich der Gestaltung und der Implementierung von User Interfaces bilden zusätzliche Schwerpunkte.⁴⁹

Der Unterschied zwischen HCI und Software Ergonomie, jene Wissenschaft, welche im nächsten Abschnitt behandelt wird, besteht darin, dass die Forschungsrichtung der HCI ein größeres Themengebiet als die Software Ergonomie umfasst. So werden auch hardwareergonomische Aspekte, beispielsweise die Gestaltung von Tastaturen oder ergonomischen Bildschirmen, behandelt.⁵⁰

⁴⁸ eigene Darstellung, Daten entnommen aus ACM SIGHCI (1996), online

⁴⁹ vgl. ACM SIGHCI (1996), online

⁵⁰ vgl. Streitz (1988): S. 3

1.3 Software Ergonomie

Eng verwandt, aber nicht deckungsgleich mit der HCI ist die Ergonomie⁵¹. Diese Wissenschaft beschäftigt sich mit der Tätigkeit des Menschen mit technischen Hilfsmitteln und der Arbeitsumgebung im Allgemeinen. Anatomische, physiologische und psychologische Kenntnisse werden in der Ergonomie angewendet, um die zwei wesentlichen Ziele zu erreichen, welche einerseits die optimale Nutzung der menschlichen Ressourcen und andererseits die Sicherung von menschlicher Gesundheit und Wohlbefinden sind.⁵² Erforderlich für das Erreichen dieser Ziele ist das Anpassen der Software an die psychischen Eigenschaften der mit ihr arbeitenden Menschen.⁵³

Die Software Ergonomie erforscht die Merkmale benutzer- und aufgabengerechter Software und entwickelt Verfahren und Softwareunterstützung für die Gestaltung von User Interfaces. Es ist ihr Anliegen, das Zusammenwirken aller Komponenten zu optimieren, welche die Arbeitssituation der Anwender bestimmen - nämlich Mensch, Aufgabe, Technik und organisatorischer Rahmen. Softwareergonomische Fragestellungen beschränken sich nicht nur auf solche, welche die Präsentation interaktiver Software betreffen; technische Merkmale wie Zuverlässigkeit oder Effizienz werden bei softwareergonomischen Überlegungen vorausgesetzt. Das Fachgebiet der Software Ergonomie erzielt durch das Zusammenwirken verschiedener Forschungsströmungen Erkenntnisse für die ergonomische Systementwicklung und -gestaltung.⁵⁴

Die technisch orientierte Forschung versucht, durch technische Weiterentwicklungen Fortschritte in der Mensch-Computer-Interaktion zu erzielen. Systemgeführte Dialogtechniken wie Formulare und Menüs - als Alternative zur Kommando-Interaktion - sowie das Prinzip der direkten Manipulation - damit verbunden der Einsatz von Fenstertechnik - erleichterten die Computerhandhabung beträchtlich. Hypertext-Systeme, welche ein einfaches Bewegen in hochvernetzten Informationsstrukturen ermöglichen, oder neue Eingabe- und Ausgabetechniken, beispielsweise die Handschrifteneingabe, Gestenerkennung oder die akustische Sprachverarbeitung, haben ebenso ihren Anteil am verbesserten Zusammenspiel zwischen Mensch und Computer. Ein weiteres Teilgebiet der technisch orientierten Forschung zur Software Ergonomie entwirft Werkzeuge zur Entwicklung von User Interfaces, User Interface Management Systeme (UIMS).

Menschliche Wahrnehmung, Denken, Problemlösen und Lernen bilden die wesentlichen Aspekte, mit welchen die kognitiv-psychologische Forschung die Systemgestaltung betrachtet. Diese Forschungsströmung brachte etwa Hinweise zur optimalen Gestaltung von Piktogrammen als bildliche Darstellung von Systemobjekten, zum konsistenten Aufbau von Bildschirmmasken und zur maximalen Tiefe von Menübäumen hervor. Die Analyse von Fehlern in der Benutzung und Hilfeanforderungen von Anwendern lieferte neue Ansätze für die Unter-

⁵¹ vgl. Dix et al. (1993): S. 99; Streitz (1988): S. 3

⁵² vgl. Ergonomics Society (o.J.), online

⁵³ vgl. Bräutigam (2000), online

⁵⁴ vgl. Maaß (1993): S. 191ff

stützung der User durch Systemausgaben und Hilfefunktionen, sowie Softwaredokumentation und Training.

Die arbeitspsychologische Forschung betrachtet den Computereinsatz im Zusammenhang menschengerechter Arbeitsgestaltung. *Ulich*⁵⁵ definiert als Kriterien Beeinträchtigungslosigkeit, Schädigungsfreiheit, Persönlichkeitsförderlichkeit und Zumutbarkeit. Möglichst leicht handhabbare Systeme zu schaffen, eine Maxime in den Anfängen der Benutzerfreundlichkeitsforschung, wird nicht angestrebt. Menschliche Handlungs-, Gestaltungs- und Entscheidungsspielräume sind erwiesenermaßen wichtige Voraussetzungen für die Zufriedenheit mit der Arbeit und für eine qualifikationsfördernde Arbeitsumgebung.

Auf Software Usability, den Maßstab zur Bewertung der Software Ergonomie, soll im folgenden Abschnitt eingegangen werden.

1.4 Software Usability

"Usability is the measure of the quality of the user experience when interacting with something - whether a Web site, a traditional software application, or any other device the user can operate in some way or another."⁵⁶

Usability, oder Gebrauchstauglichkeit⁵⁷, wie es in der deutschen Übersetzung nach *DIN EN ISO 9241-11 1998* heißt, beschreibt die Qualität der Interaktion zwischen Mensch und Maschine. Trotz der vagen Definition von *Nielsen* bezieht sich Usability meist auf computerverwandte Kontexte.

Die Qualität interaktiver Systeme wird mit zahlreichen Begriffen beschrieben: Neben Usability dienen auch Bedienbarkeit, Bedienerfreundlichkeit, Benutzerfreundlichkeit, User Friendliness, Ease of Use, Gebrauchstauglichkeit, Handhabbarkeit, Nutzungsqualität sowie User Experience und Benutzerzufriedenheit der Darstellung der Qualität der Mensch Computer Interaktion.⁵⁸ Diese Begriffe werden in der Literatur größtenteils synonym verwendet. Zu den Begriffen Usability und User Friendliness etwa findet man unzählige Definitionen. Vergleicht man den Wortgebrauch der beiden Begriffe, zeigen sich jedoch keine bedeutenden Unterschiede.⁵⁹

Der Name Usability wurde ursprünglich geprägt, um den Begriff der User Friendliness abzulösen, welcher in den frühen 1980er Jahren mit einer Menge unerwünschter Nebenbedeutungen assoziiert wurde. Allerdings ist der Bedeutungsumfang von Usability heute ähnlich vage, wie es jener von User Friendliness war, den er eigentlich verdrängen sollte.⁶⁰

Software Usability kennzeichnet die Nutzungsqualität von Software für den Benutzer. Softwaretechnische Merkmale sind in diese Bewertung nur insofern eingeschlossen, als diese

⁵⁵ vgl. Ulich (1988): S. 49

⁵⁶ Nielsen nach Eichinger (o.J. a), online

⁵⁷ Deutsches Institut für Normung (2003): EN ISO 9241-11: 1998. S. 4

⁵⁸ vgl. Geis (o.J.), online

⁵⁹ vgl. Englisch (1993): 21ff

⁶⁰ vgl. Bevan/Kirakowski/Maisel (1991), online

Auswirkungen auf die Softwarenutzung haben. Anstatt der Bewertung der technischen Qualität der Software wird ihre Wirkung auf den Benutzer festgestellt. Es wird also beurteilt, ob und inwieweit ein Produkt am Arbeitsplatz des Benutzers anforderungsgemäß und zuverlässig funktioniert.⁶¹

Die erfahrene Nutzungsqualität setzt sich aus vielen teilweise interdependenten Faktoren zusammen, die in Tabelle 5 wiedergegeben sind.

Tabelle 5: Usability-Komponenten⁶²

I. System Performance	V. Language Translation	IX. Field maintenance and serviceability
a. Reliability	a. Reading Materials	
b. Responsiveness	b. User Interfaces	X. Advertising
II. System Functions	VI. Outreach Program	a. Motivating customers to buy
III. User Interfaces	a. End-user training	b. Motivating users to use
a. Organization	b. Online help system	
b. I/O Hardware	c. Hotlines	XI. Support-group users
IV. Reading Materials	VII. Ability for customers to modify and extend	a. Marketing people
a. End-user groups	VIII. Installation	b. Trainers
b. Support groups	a. Packaging and unpackaging	c. Operators
	b. Installing	d. Maintenance workers

Die grundsätzlichen Leitkriterien zur Software Usability finden sich in der Norm DIN EN ISO 9241-11 1998. Danach ist Gebrauchstauglichkeit wie folgt definiert:

„Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“⁶³

Demnach kann Usability als Attribut einer Interaktion zwischen Benutzer und Software verstanden werden. Usability ist immer in Abhängigkeit vom Produkt und der Situation, welche durch die Ziele des Benutzers sowie dem jeweiligen Kontext begründet wird, zu betrachten. DIN EN ISO 9241-11 präsentiert Effizienz, Effektivität und Benutzerzufriedenheit als Kriterien zur Bestimmung der Software Usability, die anschließend in Abbildung 4 veranschaulicht werden:⁶⁴

⁶¹ vgl. DATech (2004), online

⁶² modifiziert nach Gould/Boies/Ukelson (1997): S: 233

⁶³ Deutsches Institut für Normung (2003): EN ISO 9241-11: 1998. S. 4

⁶⁴ vgl. Eichinger (o.J. b), online

- Effizienz: Stehen die vom Benutzer eingesetzten Ressourcen in Relation zum Ergebnis?
- Effektivität: Ermöglicht die Software dem Benutzer das Erledigen seiner Aufgaben?
- Benutzerzufriedenheit: Wie zufrieden ist der Benutzer mit dem durch die Software erreichten Arbeitsziel? Ist der Benutzer gegenüber dem Softwareeinsatz positiv eingestellt?

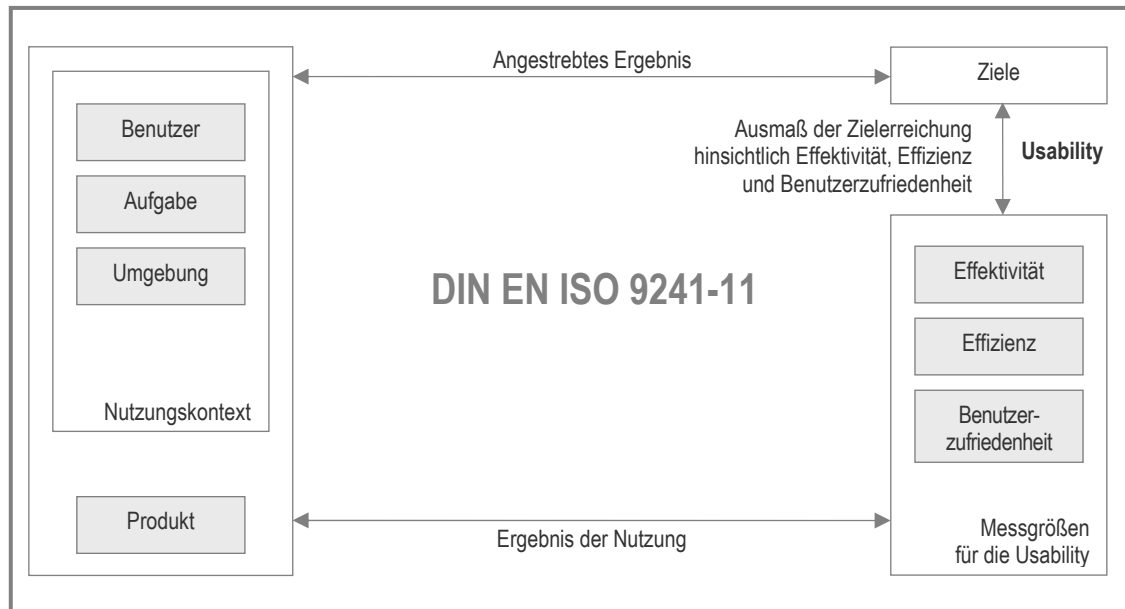


Abbildung 4: Anwendungsrahmen der Software Usability nach DIN EN ISO 9241-11⁶⁵

Zur Bewertung der Usability einer Software sind demnach folgende Elemente notwendig:⁶⁶

- Darstellung der Ziele der Interaktion
- Darstellung der Benutzer
- Darstellung der Aufgaben
- Darstellung der Ausstattung (insbesondere Software und Hardware)
- Darstellung der Umgebung, welche die Eigenschaften der sozialen und physikalischen Umwelt, beispielsweise die Organisationsstruktur oder die Raumtemperatur, beinhaltet
- Auswahl von Usability-Messgrößen für die Kriterien Effizienz, Effektivität und Zufriedenheit

Eine alternative Definition des Begriffes Usability geben *Dumas* und *Redish*: „Usability means that the *people who use the product* can do so *quickly and easily* to accomplish *their own tasks*.“⁶⁷ Diese Definition basiert auf folgenden vier Elementen⁶⁸:

⁶⁵ modifiziert nach Deutsches Institut für Normung (2003): EN ISO 9241-11: 1998. S. 6

⁶⁶ vgl. Deutsches Institut für Normung (2003): EN ISO 9241-11: 1998. S. 14; Eichinger (o.J. b), online

⁶⁷ Dumas/Redish (1999): S. 4

⁶⁸ vgl. Dumas/Redish (1999): S. 4ff

- Usability bedeutet das Fokussieren auf den Anwender. Ohne die Zusammenarbeit mit tatsächlichen bzw. potentiellen Anwendern kann keine ergonomisch gute Software entstehen.
- Menschen verwenden Produkte, um produktiv zu sein. Die Softwarehersteller müssen die Leistungsziele der Anwender verstehen, um eine hohe Usability sicherstellen zu können.
- Die Anwender sind Menschen, die das Erledigen ihrer Aufgaben anstreben. Sie verknüpfen Usability mit Leistungsfähigkeit. Eine Software ist ein Instrument, das Menschen bei der Durchführung ihrer Aufgaben helfen soll. Die Bereitschaft der Anwender, Zeit damit zu verbringen, das Benutzen solch eines Instruments zu erlernen, ist sehr beschränkt.
- Die Anwender entscheiden, ob ein Produkt leicht zu bedienen ist - nicht die Entwickler der Software. Die Anwender wägen Zeit und Mühe, die für das Erlernen eines Produkts erforderlich sind, mit dem erwarteten Nutzen ab. Bei Produkten mit geringer Usability nutzen Anwender zumeist nur einen geringen Teil der zur Verfügung stehenden Funktionalitäten.

Zunächst wurde Software Usability als Ergänzung des Nutzens (Utility) bzw. der Funktionalität eines Softwareprodukts betrachtet: Der funktionale Nutzen soll erfassen, ob die Software die für das Unternehmen benötigten Funktionen prinzipiell zur Verfügung stellt und somit die technischen Anforderungen des Unternehmens erfüllt. Demgegenüber würde Usability der Frage nachgehen, wie diese Funktionen bereitgestellt werden und wie gut diese durch den Anwender genutzt werden können. Funktionalität und Usability kombiniert, würden die Nützlichkeit (Usefulness) eines Softwareprodukts ergeben. Usability wäre demnach eine Qualität des Produktes, welche unabhängig vom Nutzungskontext erfasst werden könnte.⁶⁹

Die Definition von Software Usability, welche dieser Arbeit zugrunde liegt, orientiert sich an der Systembenutzung, am Benutzer und am Nutzungskontext.⁷⁰ Sie versteht Usability als Nutzungsqualität in dem Sinne, dass ein starker Zusammenhang zwischen Softwaresystem und Nutzungskontext besteht. Es muss folglich das Ziel der Softwareanbieter sein, Systeme mit hoher Effektivität und Effizienz zu gestalten. Denn ist eine Software effektiv und effizient, werden die Benutzer ohne Beeinträchtigung mit dem System arbeiten und eine positive Einstellung zu diesem entwickeln können. Die Anwender werden zufrieden sein. Im Mittelpunkt dieser Definition steht die Ziel- und Aufgabenorientierung. Diese Auffassung von Usability teilt auch die überwiegende Mehrzahl der Usability-Experten.⁷¹

⁶⁹ vgl. Burmester/Hassenzahl/Koller (2002): S. 32f

⁷⁰ vgl. Bevan/Kirakowski/Maissel (1991), online

⁷¹ vgl. Burmester/Hassenzahl/Koller (2002): S. 33

2 User Centred Design

Die zahlreichen ISO Qualitätsstandards zur Software Ergonomie erheben Usability zum höchsten Ziel für den gesamten Produktlebenszyklus eines Softwaresystems. Dadurch ergibt sich für Usability eine strategisch wichtige Rolle speziell im Entwicklungsprozess, aber auch in allen anderen Lebenszyklusphasen einer Software. Um das Metaziel, die Sicherstellung hoher ergonomischer Softwarequalität, zu erreichen, bedarf es eines User Centred Design Prozesses und des Gebrauchs geeigneter Techniken zum Evaluieren von Software Usability.⁷²

Neben dem User Centred Design hat sich in der Literatur der Begriff des Usability Engineering durchgesetzt. Unterschiede zwischen diesen beiden Begrifflichkeiten lassen sich kaum feststellen. *Preim*⁷³ definiert Usability Engineering als Prozess zur Entwicklung ergonomisch ausgereifter Benutzerschnittstellen. *Eichinger*⁷⁴ fasst den Begriff des Usability Engineering enger: Es sei ein Prozess, bei dem die Usability eines Produkts definiert, gemessen und verbessert wird und der somit alle Aktivitäten innerhalb eines Produktlebenszyklus umfasst, die einer gesteigerten Usability dienen. In dieser Arbeit werden die beiden Bezeichnungen synonym verwendet.

Gould, Boies und *Ukelson* haben folgende vier grundlegende Regeln für User Centred Design formuliert:

Tabelle 6: Vier Grundregeln des User Centred Design⁷⁵

Frühes und kontinuierliches Fokussieren der Benutzer	Gespräche mit Benutzern führen anstatt deskriptive Informationen über ihre Aufgaben zu sammeln
Empirische Überprüfung der Usability	Usability messen und Anpassungen am System machen, um den Entwicklungsprozess anzutreiben
Iterativer Designprozess	Notwendige Änderungen identifizieren und durchführen sowie eine grundlegende Bereitschaft für Änderungen zu zeigen
Integriertes Design	Usability-Aspekte gleichzeitig entwickeln anstatt einzelne Komponenten isoliert zu betrachten

In der Praxis zeigt sich die Bedeutung von Schwächen in der Software Usability. Während nur 15 Prozent der Softwareprobleme mit den Funktionalitäten einer Software zusammenhängen, sind etwa 60 Prozent die Folge mangelhafter Usability. Jene Probleme, welche aus einer falschen Definition der Anforderungen resultieren, sind die kostenintensivsten, was ihre Korrektur betrifft. Obwohl bekannt ist, dass User Centred Design die Wahrscheinlichkeit des Erfolgs eines Produkts drastisch erhöht, werden User Centred Design Methoden längst nicht von allen Softwareanbietern angewandt.⁷⁶

⁷² vgl. Bevan/Curson (1998), online

⁷³ vgl. Preim (1999): S. 534

⁷⁴ vgl. Eichinger (o.J. c), online; Usability Forum (o.J. b), online

⁷⁵ eigene Darstellung, Daten entnommen aus Gould/Boies/Ukelson (1997): S. 233ff

⁷⁶ vgl. Bevan (1999): S. 95

User Centred Design birgt große Kosteneinsparungschancen für Softwareanbieter. Je konsequenter ein User Centred Design Prozess verfolgt und je früher Usability Testing betrieben wird, desto günstiger gestaltet sich die allfällige Fehlerbehebung: Die Beseitigung eines Problems kostet in der Entwicklungsphase zehnmal mehr als in der Designphase, ist das Produkt bereits am Markt, kostet es hundertmal mehr als in der Designphase.⁷⁷

Nachfolgend wird der Prozess des User Centred Design beschrieben. Daran anknüpfend wird auf einige User Centred Design Prinzipien eingegangen.

2.1 Prozess des User Centred Design

Der Prozess des User Centred Design kann modellhaft als iterativer Ablauf von vier Aktivitäten betrachtet werden (siehe Abbildung 5).

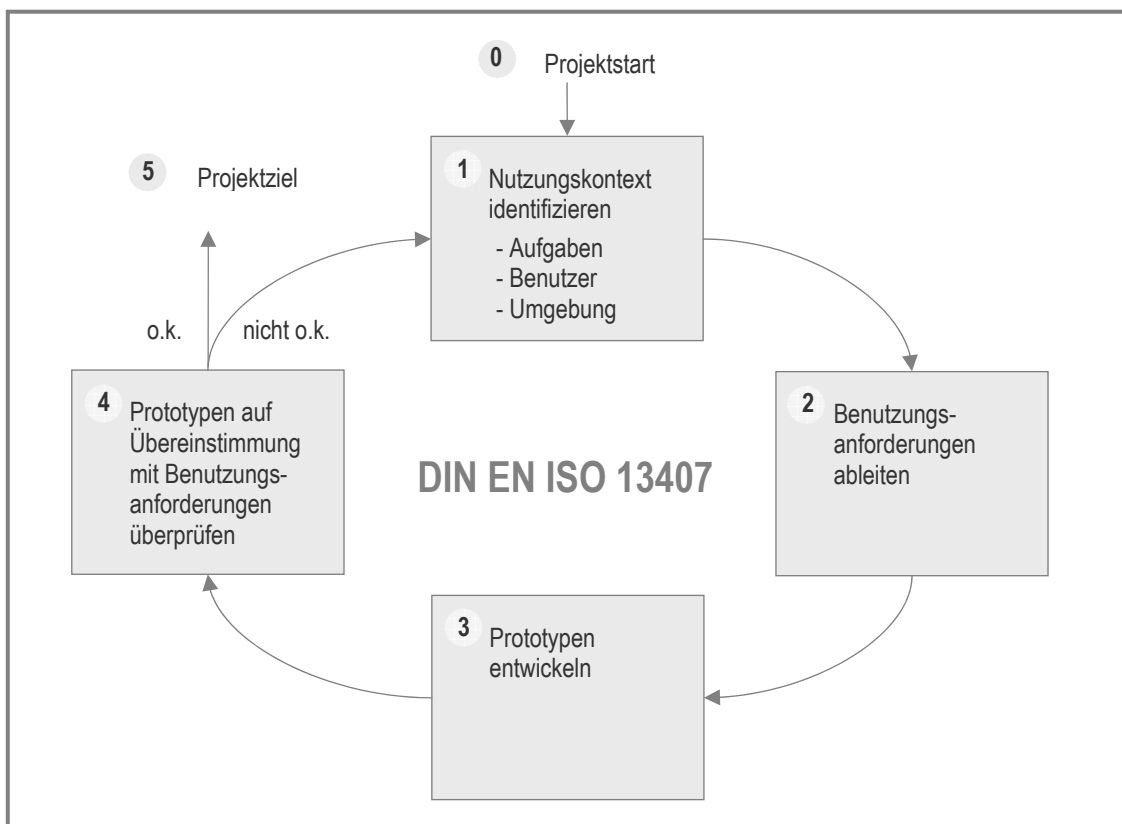


Abbildung 5: DIN EN ISO 13407 - Benutzerorientierte Gestaltung interaktiver Systeme⁷⁸

Zunächst wird der Nutzungskontext analysiert, welcher durch die zu erledigenden Aufgaben, die Benutzer sowie die Arbeitsumgebung charakterisiert ist. Daraus lassen sich die Benutzungsanforderungen ableiten. Auf der Grundlage dieser konkreten Anforderungen kann mit dem Entwickeln von Designlösungen begonnen werden. Anschließend werden die Designlösungen daraufhin getestet, inwieweit sie die vorher definierten Anforderungen erfüllen. Damit schließt

⁷⁷ vgl. Gilb (1988): S. 221

⁷⁸ eigene Darstellung, Daten entnommen aus Bevan (1999): S. 94; Geis/Hartwig (1998): S. 169

sich der Kreislauf des User Centred Design.⁷⁹ Dieser Kreislauf wird solange wiederholt, bis das Softwaresystem die Anforderungen erfüllt, wobei sich die Usability-Anforderungen im Laufe des Softwareentwicklungsprozesses zunehmend spezifizieren.⁸⁰

Die Festlegung der angestrebten Usability-Ziele ist eines der zentralsten Elemente im Rahmen des User Centred Design. Ohne konkrete Anforderungen an die Usability einer Software lassen sich keine Aussagen über die aktuelle Usability treffen (siehe Abbildung 6). Die Ziele werden in einem mehrstufigen - folgende Schritte umfassenden - Prozess gebildet.⁸¹

- Analyse der Aufgabe, der Benutzer und der Umgebung: Es gilt herauszufinden, zu welchem Zweck ein Produkt eingesetzt wird und in welchem Ausmaß dieser Zweck erreicht wird. Ziel der Aufgabenanalyse ist es, Aufgaben und ihre Häufigkeit zu bestimmen. Dies geschieht beispielsweise durch Beobachtung der Benutzer im Arbeitsumfeld, Befragung der Zielgruppe oder Videoanalyse des Arbeitsablaufs. Da Usability-Ziele auf die relevanten Benutzereigenschaften abgestimmt sein sollen, wird ein Benutzerprofil erstellt, das bei der Festsetzung der Ziele berücksichtigt wird.
- Festlegung der Usability-Attribute: Die Usability-Dimensionen - also Effizienz, Effektivität und Zufriedenheit - werden in diesem Schritt in Attribute zerlegt. Fehlerraten, Flexibilität oder Erlernbarkeit stellen mögliche Attribute dar. Entscheidend ist, dass diese Kriterien operationalisiert - sprich in messbare Größen verwandelt - werden können.
- Bestimmung eines kritischen Levels für die Usability-Attribute: Für jedes Usability-Attribut wird ein kritischer Wert festgesetzt, der die akzeptierten Ausprägungen bestimmt. Es kann dies ein absoluter (zum Beispiel: „Ein Benutzer sollte die Aufgabe mit nicht mehr als einem Fehler beenden.“) oder auch ein relativer Wert sein („Ein Benutzer sollte mit Produkt X innerhalb von fünf Minuten weniger Fehler machen als mit Produkt Y.“) Die Quellen für kritische Werte bilden frühere Usability Tests, Expertenurteile, Marktforschungsdaten oder Feldstudien.

Nach Festlegung der angestrebten Usability-Ziele wird das Messinstrument bestimmt. Ein Messinstrument beschreibt die Methode zur Erhebung der Werte der Usability-Attribute. Die quantitativen Ausprägungen können sowohl subjektive, etwa Benutzerzufriedenheit, wie auch objektive Werte, Zeitaufwand oder Fehlerhäufigkeit, darstellen. Zu den beliebtesten Messinstrumenten zählen Fragebögen, Protokollierungssoftware oder -videos und Benchmark-Aufgaben. Mit Hilfe der Messinstrumente können Usability-Probleme diagnostiziert werden.⁸² Sie bilden somit das Herzstück im Prozess des User Centred Design.⁸³

⁷⁹ vgl. Bevan (1999): S. 93; Bevan/Curson (1998), online

⁸⁰ vgl. Bevan (1999): S. 93

⁸¹ vgl. Eichinger (o.J. d), online

⁸² vgl. Eichinger (o.J. d), online

⁸³ Siehe dazu im Speziellen Kapitel II 3 Usability Testing.

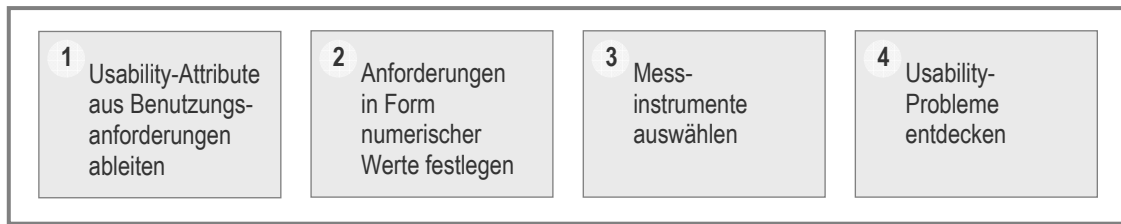


Abbildung 6: User Centred Design Prozess: Schritt 4 ⁸⁴

2.2 Prinzipien des User Centred Design

Prinzipien, Normen und Guidelines können Usability Testing, den wohl wichtigsten Bestandteil des User Centred Design, nicht ersetzen; sie können es jedoch sinnvoll ergänzen. Das Befolgen von Prinzipien garantiert keine idealen Produkte, es reduziert aber die Wahrscheinlichkeit, dass im Zuge von Usability Tests schwerwiegende Probleme auftreten.⁸⁵

Wesentliche Beachtung muss der Tatsache geschenkt werden, dass das Ziel einer qualitativ hohen Software Usability mit anderen Zielen abgewogen werden muss. Prinzipien des User Centred Design isoliert zu betrachten, wäre nicht sinnvoll. Beispielsweise stellen Passwortabfragen und andere Formen der Authentifizierung ganz offensichtlich eine Beschränkung der Benutzerfreundlichkeit einer Software dar; dennoch sind sie zweifellos nützlich.⁸⁶

Im Folgenden wird auf allgemeine Entwurfsprinzipien, internationale Usability Standards und Hersteller Styleguides eingegangen:

Ein allgemeines Entwurfsprinzip ist eine sehr allgemein formulierte Anweisung, auf Erkenntnissen der menschlichen Lernfähigkeit und des ergonomischen Arbeitens basierend. Als Beispiel sei hier das Prinzip der Konsistenz genannt: Bei der Auswahl der Worte, Formate, Grafiken und Vorgehensweisen innerhalb des Systems soll auf Konsistenz geachtet werden. Dieses Prinzip basiert auf der Tatsache, dass Menschen schneller lernen und sich Neuerlerntes besser aneignen, sofern ihre Eindrücke konsistent sind.⁸⁷

In der Literatur gibt es unzählige Checklisten mit Entwurfsprinzipien für User Interfaces. Die Bekannteste ist jene von *Smith* und *Mosier*⁸⁸, zugleich Ausgangspunkt für die Entwicklung der internationalen Normen durch die International Organization of Standardization.⁸⁹ Tabelle 7 gibt die kurzen und prägnanten Entwurfsprinzipien nach *Preim* wieder, welche auch die Grundsätze der Dialoggestaltung nach *EN ISO 9241-10* widerspiegeln.

⁸⁴ modifiziert nach Eichinger (o.J. c), online

⁸⁵ vgl. Dumas/Redish (1999): S. 55; Bevan/Kirakowski/Maissel (1991), online

⁸⁶ vgl. Preim (1999): S. 55

⁸⁷ vgl. Dumas/Redish (1999): S. 53

⁸⁸ vgl. Smith/Mosier (1986), online

⁸⁹ vgl. Bevan/Kirakowski/Maissel (1991), online

Tabelle 7: Entwurfsprinzipien nach Preim⁹⁰

I. Informiere dich über potentielle Benutzer und ihre Aufgaben.	IX. Kombiniere visuelle Interaktion mit sprachbasierter Interaktion.
II. Hilf Benutzern, ein mentales Modell zu entwickeln.	X. Vermeide, dass Benutzer sich zu viele Dinge merken müssen.
III. Sprich die Sprache des Benutzers.	XI. Ermögliche es, Fehler zu erkennen, zu diagnostizieren und zu beheben.
IV. Mach Systemzustände sichtbar und unterscheidbar.	XII. Vermeide es, den Benutzer zu überraschen.
V. Verdeutliche die jeweils möglichen Aktionen.	XIII. Beachte die wichtigsten Bedienhandlungen besonders.
VI. Strukturiere die Benutzungsschnittstelle.	XIV. Erkläre die Bedienung des Programms durch Beispiele und weniger durch Formalismen.
VII. Stelle eine erkennbare Rückkoppelung sicher.	
VIII. Gestalte die Schnittstelle adaptierbar.	

Hersteller Styleguides bilden Regelwerke, welche Softwareanbieter herausgegeben haben, um ein einheitliches Look & Feel ihrer Systeme zu beschreiben.⁹¹ Beispiele sind *Apple's Human Interface Guidelines*⁹² oder *SAA Common User Access* von IBM⁹³. Im Gegensatz zu Entwurfsprinzipien, sind Styleguides konkreter; sie enthalten Einzelheiten der Gestaltung und Benutzung von Bedienelementen und sind zumeist plattformspezifisch.⁹⁴ Ein allgemeines Entwurfsprinzip stellt ein Ziel dar; es werden aber keine Aussagen darüber getroffen, wie man das Ziel erreicht. Eine Guideline formuliert ein konkretes Ziel für spezifische Benutzer, Arbeitsumgebungen und Technologien. Für ein menügesteuertes Produkt beispielsweise könnte das Prinzip der Konsistenz in viele verschiedene Guidelines aufgespalten werden. Eine könnte sich etwa darauf beziehen, dass das Verlassen jedes einzelnen Menüs konsistent sein soll.⁹⁵

Hersteller Styleguides vereinen wissenschaftliche Forderungen und praktische Empfehlungen zur Gestaltung des User Interface. Der erste Teil, der generelle Entwurfsprinzipien für Schnittstellen enthält, welche von der Software Ergonomie in den letzten Jahren erarbeitet wurden, beispielsweise Einfachheit, Konsistenz und Benutzerkontrolle, wird um einen zweiten Teil ergänzt, in welchem detaillierte Angaben zum User Interface Design festgesetzt sind: Welche Interaktionselemente einzuplanen sind, wie diese aussehen sollen und wie sie zu bedienen sind.⁹⁶ Der Hauptteil umfasst also konkrete Richtlinien für die Verwendung von Interaktionstechniken, beispielsweise den Einsatz von Tastaturkürzeln, die Gestaltung von Icons und die Platzierung von Fenstern.⁹⁷

Sowohl allgemeine Entwurfsprinzipien wie auch Hersteller Styleguides treffen ausschließlich Aussagen zur Gestaltung der Systemhandhabung. Somit konzentrieren sie sich ausschließlich

⁹⁰ eigene Darstellung, Daten entnommen aus Preim (1999): S. 55ff

⁹¹ vgl. Maaß (1993): S. 199

⁹² vgl. Apple Computer (2005), online

⁹³ vgl. Berry (1988), online

⁹⁴ vgl. Preim (1999): S. 55

⁹⁵ vgl. Dumas/Redish (1999): S. 53

⁹⁶ vgl. Maaß (1993): S. 199

⁹⁷ vgl. Preim (1999): S. 532

auf einen Aspekt des User Centred Design. Die Aufgabenangemessenheit als Gestaltungsgrundsatz wird, anders als bei nationalen und internationalen Normen, nicht beachtet.⁹⁸

Usability Standards und Normen bestimmen für ein Softwaresystem wichtige Eigenschaften und enthalten zudem auch Richtlinien für den Entwicklungsprozess.⁹⁹ Je nachdem, welche Aspekte einer Software sie primär ansprechen, werden sie einer der vier Gruppen zugeordnet (siehe Abbildung 7): Zu den Standards für die Softwareanwendung (product use in context) zählen Normen zur Effektivität, Effizienz und Benutzerzufriedenheit. Für das Design von User Interfaces stehen Normen zur Produktqualität zur Verfügung. Normen zum Softwaregestaltungsprozess lassen auf den User Centred Design Prozess schließen. Schließlich zielen Normen zur prinzipiellen Fähigkeit eines Softwareanbieters, die Usability einer Software zu garantieren, darauf, die Usability im gesamten Softwarelebenszyklus zu sichern.¹⁰⁰

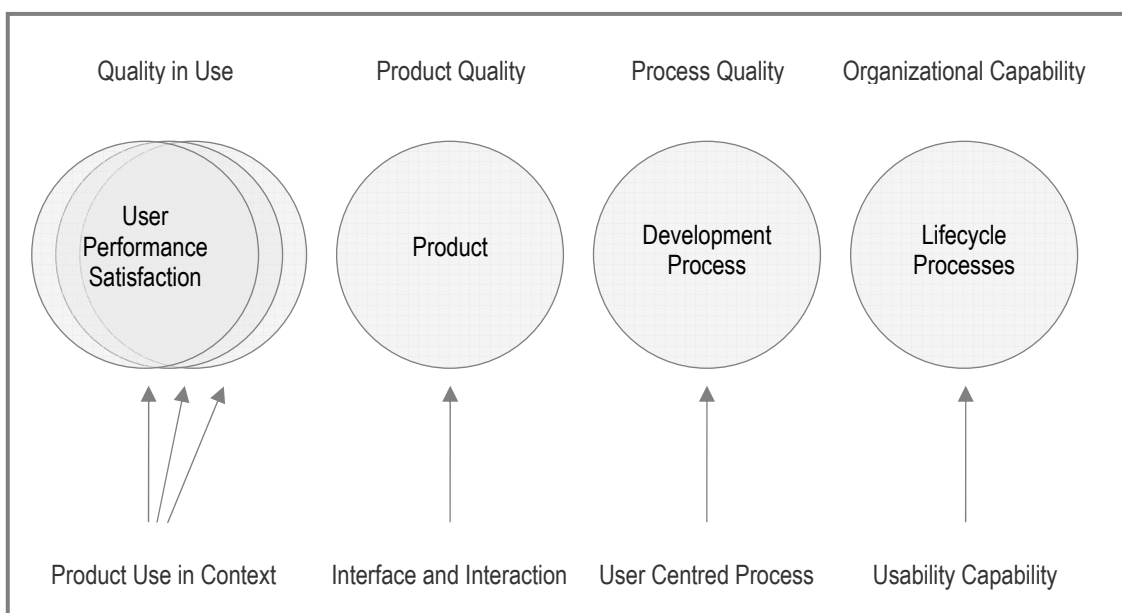


Abbildung 7: Kategorisierung der Usability-Normen¹⁰¹

Für die Softwaregestaltung ist Teil 10 der Norm *EN ISO 9241*¹⁰² der *International Organization for Standardization (ISO)* von überragender Bedeutung. Die Dialoggestaltung bezieht sich auf die Interaktion des Benutzers mit dem Softwaresystem. *EN ISO 9241-10* nennt sieben übergeordnete Grundsätze für die Dialoggestaltung¹⁰³, die während der Entwicklung mit einzubeziehen sind:

⁹⁸ vgl. Maaß (1993): S. 199

⁹⁹ vgl. Preim (1999): S. 529

¹⁰⁰ vgl. Serco (2001), online

¹⁰¹ modifiziert nach Serco (2001), online

¹⁰² vgl. Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 3ff

¹⁰³ Eine Ergänzung der Dialoggestaltungsgrundsätze findet sich in den detaillierten Normen der Teile 12 bis 17 der EN ISO 9241.

- Aufgabenangemessenheit: „Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen.“¹⁰⁴ Voraussetzung für die Erfüllung dieses Grundsatzes ist die Kenntnis von Benutzer und Aufgabe.¹⁰⁵ So sollten in einem Formular keine mühsam auszufüllenden Pflichtangaben verlangt werden, die für den relevanten Vorgang nicht wesentlich sind.¹⁰⁶
- Selbstbeschreibungsfähigkeit: „Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschritt durch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklärt wird.“¹⁰⁷ Ein Benutzer sollte stets Bescheid wissen, wo er sich im System befindet und was er machen muss, um sein Ziel zu erreichen.¹⁰⁸
- Steuerbarkeit: „Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.“¹⁰⁹ Bei einem Dateidownload soll es dem Benutzer beispielsweise möglich sein, diesen zu unterbrechen und später fortzusetzen.¹¹⁰
- Erwartungskonformität: „Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z.B. seinen Kenntnissen aus dem Arbeitsgebiet, seiner Ausbildung und seiner Erfahrung sowie den allgemein anerkannten Konventionen.“¹¹¹ Ziel ist es, das Erstaunen des Benutzers zu minimieren.¹¹² Sind Wörter unterstrichen, so soll es sich bei diesen um Hypertext-Links handeln.¹¹³
- Fehlertoleranz: „Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhaften Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.“¹¹⁴
- Individualisierbarkeit: „Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe sowie an die individuellen Fähigkeiten und Vorlieben des Benutzers zulässt.“¹¹⁵
- Lernförderlichkeit: „Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.“¹¹⁶ Für die Lernförderlichkeit sind eine gute Übersicht über verfügbare Optionen, eine ausreichende Rückkoppelung des Systems und Fehlermeldungen, welche leicht interpretiert werden können, wichtig.¹¹⁷

¹⁰⁴ Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 4

¹⁰⁵ vgl. Preim (1999): S. 73

¹⁰⁶ vgl. Wirth (2005), online

¹⁰⁷ Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 5

¹⁰⁸ vgl. Schneider (2000), online

¹⁰⁹ Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 6

¹¹⁰ vgl. Preim (1999): S. 73; Wirth (2005), online

¹¹¹ Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 6

¹¹² vgl. Preim (1999): S. 73

¹¹³ vgl. Wirth (2005), online

¹¹⁴ Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 7

¹¹⁵ Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 8

¹¹⁶ Deutsches Institut für Normung (2003): EN ISO 9241-10: 1996. S. 9

¹¹⁷ vgl. Preim (1999): S. 73f

Über die Summe und Themen der internationalen Normen der *ISO* zur Software Usability gibt Tabelle 8 einen Überblick:

Tabelle 8: Internationale Normen zur Software Usability¹¹⁸

	Prinzipien und Empfehlungen	Spezifizierungen
Kontextuelle Anwendung	ISO/IEC 9126-1: Software Engineering - Product quality - Part 1: Quality model ISO/IEC TR 9126-4: Software Engineering - Product quality - Part 4: Quality in use metrics ISO 9241-11: Guidance on Usability	ISO 20282: Usability of everyday products
User Interface und Interaktion	ISO/IEC TR 9126-2: Software Engineering - Product quality – Part 2 External metrics ISO/IEC TR 9126-3: Software Engineering - Product quality – Part 3 Internal metrics ISO 9241: Ergonomic requirements for office work with visual display terminals. Parts 10-17 ISO11064: Ergonomic design of control centres ISO 14915: Software ergonomics for multimedia user interfaces IEC TR 61997: Guidelines for the user interfaces in multimedia equipment for general purpose use	ISO 9241: Ergonomic requirements for office work with visual display terminals. Parts 3-9 ISO/IEC 10741-1: Dialogue interaction - Cursor control for text editing ISO/IEC 11581: Icon symbols and functions ISO 13406: Ergonomic requirements for work with visual displays based on flat panels ISO/IEC 14754: Pen-based interfaces - Common Gestures for text editing with pen-based systems ISO/IEC 18789: Information Technology - User interface for mobile tools ISO 18789: Ergonomic requirements and measurement techniques for electronic visual displays
Software-dokumentation	ISO/IEC 18019: Guidelines for the design and preparation of software user documentation	ISO/IEC 15910: Software user documentation process
Softwaregestaltungsprozess	ISO 13407: Human centred design processes for interactive systems ISO TR 16982: Usability methods supporting human centred design	ISO/IEC 14598: Information Technology - Evaluation of Software Products
Usability im Software-lebenszyklus	ISO TR 18529: Ergonomics of human system interaction – Human centred lifecycle process descriptions	
Andere	ISO 9241-1: Part 1: General Introduction ISO 9241-2: Part 2: Guidance on task requirements ISO 10075-1: Ergonomic principles related to mental workload - General terms and definitions ISO DTS 16071: Guidance on accessibility for human computer interfaces	

¹¹⁸ eigene Darstellung, Daten entnommen aus Serco (2001), online

Neben den internationalen Normen und Standards der *ISO* widmet sich auch eine EU-Richtlinie dem Thema Software Usability. Die 1990 erlassene *EU-Bildschirmrichtlinie (90/270/EWG)*¹¹⁹ setzt Mindestvorschriften und Pflichten für Arbeitgeber fest. Sie befasst sich vor allem mit der physischen Arbeitsumgebung, enthält aber auch Anforderungen betreffend der Aufgabenangemessenheit, der leichten Benutzbarkeit und der anzuwendenden Prinzipien der Software Ergonomie. Verglichen mit den *ISO*-Normen enthält die Richtlinie konkretere Vorschriften, welche sich allerdings hauptsächlich auf gesundheitliche Aspekte beziehen. Standards zur Effektivität und Effizienz eines Softwaresystems sind in der Richtlinie nicht festgelegt.¹²⁰

3 Usability Testing

„A usability test is a procedure for determining whether the quantitative goals defined in the usability specification have been achieved. The term “test” is a broad term that encompasses any method for assessing whether goals have been met.”¹²¹

Usability Testing bezeichnet also jenes Vorgehen, während dessen geprüft wird, ob die vorab definierten Usability-Ziele erreicht wurden. Testing ist dabei ein allgemeiner Name für viele verschiedene Methoden, die dabei Anwendung finden können.¹²²

Dumas und *Redish* formulieren den Begriff Usability Testing ein wenig provokant: Während sich das Testen der Funktionalität bzw. des Nutzens einer Software darauf konzentriert, ob das Produkt die gewünschten technischen Spezifikationen besitzt, rückt beim Usability Testing die Frage in den Mittelpunkt, inwieweit die Anwender die Funktionalitäten der Software überhaupt finden und mit diesen ihre Arbeit effektiv erledigen können.¹²³

3.1 Motive für den Einsatz von Usability Testing

Im Grunde genommen eignen sich die verschiedenen Methoden des Usability Testing in allen Phasen des Systemdesigns und der Systemnutzung, also im gesamten Lebenszyklus eines Softwareprodukts.¹²⁴ *Oppermann* hebt drei besondere Anlässe für den Einsatz von Evaluationsverfahren hervor: Zum einen kann die Software Usability während des Systemdesigns überprüft werden, die Usability eines bereits bestehenden Systems kann evaluiert werden oder es werden zum Zweck des Systemvergleichs mehrere Softwareprodukte miteinander hinsichtlich ihrer Usability verglichen.¹²⁵

Die Überprüfung der Software Usability in der Entwicklungsphase wird formative Evaluation¹²⁶ genannt. Die Evaluation von softwareergonomischen Eigenschaften durch den Softwarehersteller kann bereits in der Phase des Systementwurfs einsetzen. Mit Hilfe einer Prüfliste von

¹¹⁹ vgl. EUR-Lex (o.J.), online

¹²⁰ vgl. Bevan (1995), online

¹²¹ Wixon/Wilson (1997): S. 669

¹²² vgl. Eichinger (o.J. e), online

¹²³ vgl. Dumas/Redish (1999): S. 4

¹²⁴ vgl. Stary (1994): S. 317

¹²⁵ vgl. Oppermann et al. (1992): S. 7

¹²⁶ vgl. Holzinger (2001a): S. 224

angestrebten Usability-Eigenschaften wird bewertet, ob es sich um ein angemessenes Gestaltungskonzept handelt. Konkreter wird die Evaluation des Prototyps ausfallen, die sich von der Evaluation eines fertigen Systems im Grunde genommen nicht unterscheidet.¹²⁷

Mögliche Anwender der Evaluierung einer bereits entwickelten Software - auch als summative Evaluation¹²⁸ bekannt - sind Softwarehersteller, Unternehmen, welche die Software einsetzen (wollen), oder Software-Prüfstellen in Frage. Während bei der Evaluation eines Produktkonzepts bewertet wird, ob die softwareergonomischen Prinzipien beachtet und deren logische Konsistenz realisiert wurden, wird bei einer Produktevaluation darauf geachtet, ob diese Prinzipien durchgängig Anwendung finden.¹²⁹

Im Rahmen eines Systemvergleichs werden mehrere Softwareprodukte bewertet, beispielsweise im Zuge einer Auswahl einer bestimmten Software. Die Anwendergruppen sind in diesem Fall Softwarehersteller, Unternehmen als zukünftige Anwender oder Software-Prüfstellen.

3.2 Ziele von Usability Testing

Die Ziele, welche mit der Evaluation der Software Usability verfolgt werden, stehen im direkten Verhältnis zu den Beziehungen zwischen den drei Faktoren Benutzer, Aufgabe und Software (siehe auch Abbildung 8):¹³⁰

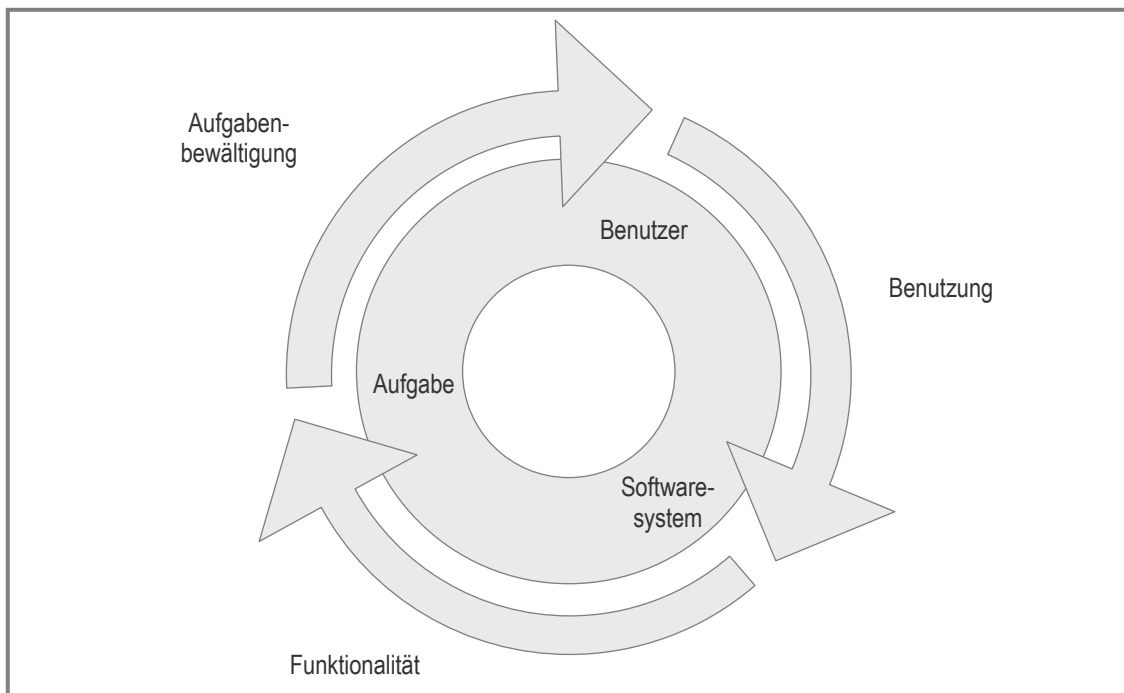


Abbildung 8: Verhältnis zwischen Software, Benutzer und Aufgabe¹³¹

¹²⁷ vgl. Oppermann (1988): S. 326

¹²⁸ vgl. Holzinger (2001): S. 224

¹²⁹ vgl. Oppermann (1988): S. 327

¹³⁰ vgl. Oppermann et al. (1992): S. 6f

¹³¹ modifiziert nach Oppermann et al. (1992): S. 6

- **Aufgabenbewältigung:** Bei der Evaluation von Softwareprodukten interessiert vor allem, inwieweit die Aufgabenerledigung durch die Software unterstützt oder behindert wird. Es wird versucht, das Ausmaß der Unterstützung des Benutzers an der Aufgabenbewältigung über die beiden folgenden Beziehungen - Benutzung und Funktionalität - zu erfassen.
- **Benutzung:** Es soll herausgefunden werden, wie groß der Aufwand der Systemnutzung ist. Einflussnehmende Faktoren sind beispielsweise die Erlernbarkeit der Software und die Anpassung des Systems an Benutzereigenschaften.
- **Funktionalität:** Die Aufgabenangemessenheit der Software wird durch das Ausmaß der Unterstützung der Aufgaben durch das System bestimmt. Ob das System die Aufgaben hinreichend genau abbilden kann oder inwieweit der Benutzer das System umgestalten kann, sind Aspekte, welche für die Funktionalität entscheidend sind.

3.3 Methoden zur Beurteilung der Software Usability

Die Einsatzmöglichkeiten von Evaluationsverfahren erstrecken sich über alle Phasen der Entwicklung und Nutzung von Softwaresystemen.¹³² Tabelle 9 vergleicht wichtige Methoden - darunter sowohl expertenzentrierte wie auch benutzerzentrierte - anhand einiger prägnanter Kriterien. Die Methoden, welche die Usability-Forschung einsetzt, sind für Betriebswirte oft keine unbekanntes. Interessant ist es, einen Vergleich zwischen den Methoden der Usability-Forschung und jenen der Marketing-Disziplin anzustellen - so unterschiedlich die beiden Disziplinen auf den ersten Blick anmuten, so groß ist die Übereinstimmung ihrer Methoden.¹³³

Tabelle 9: Methoden des Usability Testing¹³⁴

	Expertenzenzentrierte Methoden			Benutzerzentrierte Methoden		
	Heuristische Evaluation	Cognitive Walkthrough	Action Analysis	Thinking Aloud	Field Observation	Benutzerbefragung
Anwendbarkeit in Phasen	alle	alle	Design	Design	Finales Testen	alle
Benötigter Zeitaufwand	•	••	•••	•••	••	•
Benötigte Anzahl an Testpersonen	-	-	-	3+	20+	30+
Benötigte Anzahl an Evaluatoren	3+	3+	1-2	1	1+	1
Benötigte Ausstattung	•	•	•	•••	••	•
Benötigte Expertise	••	•••	•••	••	•••	•

¹³² vgl. Stary (1994): S. 317

¹³³ vgl. Haas/Thurnher (2006): S. 27

¹³⁴ modifiziert nach Holzinger (2005): S. 72

Unter den Inspection Methods (dt. expertenzentrierte Methoden) sind die am häufigsten eingesetzten Methoden die heuristische Evaluation, der Cognitive Walkthrough und die Action Analysis - auch bekannt unter Keystroke-Level Analysis.¹³⁵

Im Rahmen einer heuristischen Evaluation bewerten (etwa drei bis fünf) Experten - zunächst individuell -, ob die Dialogelemente des User Interface die Standards softwareergonomischer Prinzipien und Normen, sogenannte Heuristiken, erfüllen. Nachdem die einzelnen Usability-Experten die Evaluation durchgeführt haben, können die Ergebnisse untereinander diskutiert werden.¹³⁶ Heuristiken sind also ein Katalog von Qualitätsprinzipien, die entweder auf der Grundlage entsprechender Theorien oder empirischer Erkenntnisse von Wissenschaftlern entwickelt werden oder auf den Erfahrungen von Praktikern basieren.¹³⁷ Der Vorteil dieser Methode besteht darin, dass anerkannte Usability-Prinzipien angewandt werden und kleinere wie größere Probleme effektiv identifiziert werden. Eine heuristische Evaluation kann bereits früh im Entwicklungsprozess stattfinden und im gesamten Lebenszyklus der Software sinnvoll eingesetzt werden. Ein Nachteil dieser Methode ist zum einen, dass nicht bewertet werden kann, inwieweit die Bedürfnisse von weniger erfahrenen Benutzern respektiert werden, und zum anderen, dass eine ausgewogene Beurteilung des gesamten Produkts nicht sichergestellt ist, zumal die Experten möglicherweise den einen oder anderen Aspekt genauer fokussieren.¹³⁸

Cognitive Walkthrough bezeichnet jene Methode, in der wiederum jeder Experte unabhängig anhand von relevanten Aufgaben das Softwareprodukt untersucht.¹³⁹ Dabei wird das Verhalten von möglichen Benutzern simuliert, um zu überprüfen, ob das System den Erwartungen der User entspricht, ob beispielsweise Benutzer vom System bereitgestellte Funktionen als solche erkennen. Es werden insbesondere Erkenntnisse der kognitiven Theorie, etwa die menschliche Lernfähigkeit, beachtet. Die rasche Durchführbarkeit in allen Softwareentwicklungsphasen und die Tatsache, dass sich Experten in die Situation von Endanwendern zu versetzen versuchen, zählen zu den Vorteilen des Cognitive Walkthrough. Demgegenüber kritisch betrachtet werden müssen die Auswahl der überprüften Aufgaben und die Tatsache, dass die Simulation des Benutzerverhaltens nicht mit dem tatsächlichen Benutzerverhalten übereinstimmen muss.¹⁴⁰

Die Action Analysis zerlegt eine Aufgabe in Aktionen, die ein Softwarebenutzer durchführen muss, um eine Aufgabe erfolgreich abschließen zu können. Für die einzelnen Aktionen wird jeweils die dafür benötigte Zeitdauer bestimmt. Neben einer genauen Vorstellung darüber, wie lange die Erledigung einer Aufgabe braucht, profitieren die Softwarehersteller auch vom tiefen Einblick in das Benutzerverhalten. Jedoch ist diese Methode sehr zeitaufwändig und erfordert ein hohes Maß an Expertise.¹⁴¹

¹³⁵ vgl. Holzinger (2005): S. 72f

¹³⁶ vgl. Holzinger (2006b), online

¹³⁷ vgl. Fachrichtung Informationswissenschaft Saarbrücken (2006a), online

¹³⁸ vgl. Holzinger (2006b), online

¹³⁹ vgl. Usability Forum (2001), online

¹⁴⁰ vgl. Holzinger (2006b), online

¹⁴¹ vgl. Holzinger (2006b), online

Evaluationsverfahren, welche die Benutzer mit einbeziehen, sind die elementarsten Methoden und unabdingbar. Expertenzentrierte Methoden sollten immer um benutzerzentrierte ergänzt werden. Beim Einsatz von benutzerzentrierten Methoden können Informationen darüber gewonnen werden, wie Menschen Systeme einsetzen und auf welche Probleme sie dabei stoßen.¹⁴² Zu den bekanntesten der Test Methods (dt. benutzerzentrierte Methoden) zählen Befragungen sowie die Methode des Task Based Testing. Dabei kann beispielsweise die Durchführung bestimmter Aufgaben durch die Testpersonen anhand objektiv quantifizierbarer Messgrößen bewertet werden (z.B. Zeit für die Erledigung einer bestimmten Aufgabe). In sogenannten Benchmark-Tests werden standardisierte Aufgaben bei verschiedenen Softwareprodukten getestet.¹⁴³ Eine subjektive Bewertung stellen jene Evaluationen dar, welche von den Testpersonen neben dem Erledigen der Aufgabe lautes Denken (Thinking Aloud) verlangen.¹⁴⁴

Nicht von Holzinger (siehe Tabelle 9) berücksichtigt ist die Methode des **Task Based Testing**, das wohl dem klassischen Bild von Usability Tests am ehesten entspricht: Die Testpersonen bearbeiten Aufgaben am System. Diese Aufgabenerledigung wird bewertet und evaluiert. Wie solch ein Test vor sich geht, wird in Abbildung 9 veranschaulicht.

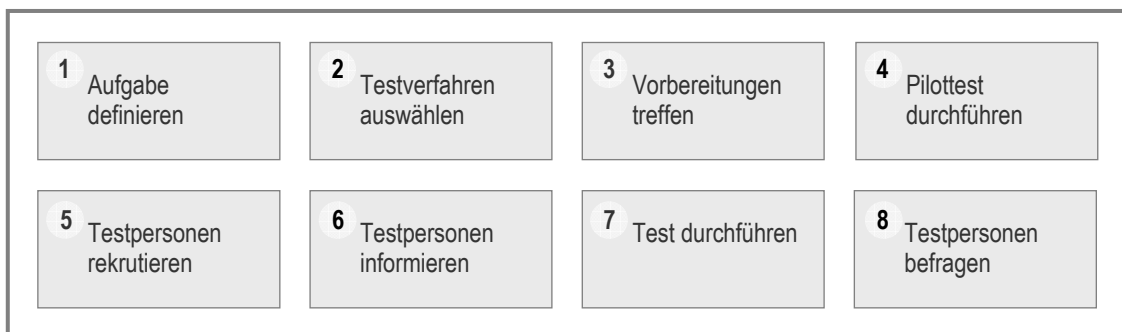


Abbildung 9: Task Based Testing¹⁴⁵

Grundlagen für den ersten Schritt der Aufgabendefinition bilden die relevanten Benutzereigenschaften sowie die Ergebnisse der Aufgabenanalyse.¹⁴⁶ Benutzer- und aufgabenspezifische Aspekte wurden bereits in der zweiten Phase des User Centred Design Prozesses betrachtet. Einerseits kann den Testpersonen das gewünschte Resultat gezeigt werden und sie werden aufgefordert, dieses zu reproduzieren - *Wixon* und *Wilson* bezeichnen solche Tests als „results-based tests“. Im Rahmen von „process-based tests“ hingegen wird eine Aufgabensequenz definiert, welche von den Testpersonen abgearbeitet werden soll.¹⁴⁷

Die zum Einsatz kommenden Methoden des Usability Testing werden ebenso auf der Basis der Usability-Ziele ausgewählt.¹⁴⁸ Man unterscheidet grundsätzlich zwischen zwei Messgrößen,

¹⁴² vgl. Holzinger (2005): S. 73

¹⁴³ vgl. Oppermann et al. (1992): S. 10ff; Holzinger (2005): S. 73f.

¹⁴⁴ vgl. Oppermann (1988): S. 328.

¹⁴⁵ modifiziert nach Eichinger (o.J. e), online

¹⁴⁶ vgl. Eichinger (o.J. e), online

¹⁴⁷ vgl. Wixon/Wilson (1997): S. 670

¹⁴⁸ vgl. Eichinger (o.J. e), online

welche bei einem Usability Test erhoben werden können: Zum einen Indikatoren, welche das Verhalten direkt abbilden (z.B. Zeit für die Erledigung einer Aufgabe, Anzahl an Fehlern), zum anderen solche, welche das Verhalten reflektierend abbilden (z.B. spontane Antworten der Testpersonen). Innerhalb eines Usability Tests können auch mehrere Methoden zum Einsatz kommen. Nachdem die Testpersonen beispielsweise eine Aufgabe erledigt haben, wird eine ähnliche Aufgabe gestellt, wobei dann Thinking Aloud als Methode angewandt wird.¹⁴⁹

Notwendige Vorbereitungen für einen Usability Test umfassen beispielsweise die Anfertigung von Fragebögen und Formularen. Der Pilottest sollte so früh im Voraus abgehalten werden, dass zwischen Pilottest und dem tatsächlichen Test genügend Zeit bleibt, um nötige Veränderungen an den Testmaterialien durchführen zu können. Welche Personen am Test teilnehmen sollten, ergibt sich aus der Benutzercharakterisierung im Rahmen des User Centred Design. Für eine Field Observation benötigt man relativ wenige Testpersonen; fünf bis zehn reichen etwa schon aus um die Software Usability umfassend beurteilen zu können.¹⁵⁰ Der Testleiter setzt die Teilnehmer am Anfang des Tests über Dauer und Zweck in Kenntnis; der Ablauf des Tests sollte für die Teilnehmer vorhersehbar sein. Ein mitunter wichtiger Hinweis an die Teilnehmer ist, dass ihr Verhalten nur hinsichtlich der Produkteigenschaften interpretiert wird und nicht das Ziel haben, Rückschlüsse über ihre Fähigkeiten zu machen.¹⁵¹ Nach Durchführung des Tests können Fragen an die Testteilnehmer hinsichtlich subjektiver Messgrößen (Zufriedenheit, erster Eindruck, etc.) gestellt werden.

Thinking Aloud ermöglicht herauszufinden, wie ein Benutzer mit dem Produkt umgeht und welche Überlegungen er bei der Benutzung anstellt. Während des Testdurchlaufs führt die Testperson eine Aufgabe durch; parallel zur Interaktion mit der Software soll die Testperson alles, was sie denkt und macht, verbal äußern. Falls Arbeitsschritte, die vom System bei der Aufgabenerledigung vorgegeben werden, von den durch die Testperson erwarteten Schritten abweichen, wird dies mit Hilfe von Thinking Aloud offensichtlich. Thinking Aloud macht offensichtlich, welche Teile des Dialogs die meisten Probleme verursachen. Der wesentliche Nutzen dieser Methode ist also ein vertieftes Verständnis vom mentalen Modell des Benutzers und dessen Umgang mit dem Produkt. Thinking Aloud kann leicht und kostengünstig zu jedem Zeitpunkt im Entwicklungsprozess eingesetzt werden. Verzerrend wirken kann mitunter das Problem der sozialen Erwünschtheit - wenn Testpersonen meinen, bestimmten Erwartungen entsprechen zu müssen. Ein relativ großer Zeitaufwand ist aufgrund der notwendigen Instruktion für die Testpersonen einzuplanen.¹⁵² Leicht abgewandelt können bei Thinking Aloud auch zwei Testpersonen gleichzeitig mit dem Lösen einer Aufgabe mit der Software beauftragt werden. Der Vorteil hier: Die Testsituation ist natürlicher, zumal die Testpersonen durch das gemeinsame Arbeiten am System ihre Gedanken gegenseitig austauschen und somit dazu

¹⁴⁹ vgl. Wixon/Wilson (1997): S. 671

¹⁵⁰ vgl. Wixon/Wilson (1997): S. 671f

¹⁵¹ vgl. Eichinger (o.J. e), online

¹⁵² vgl. Hunkirchen (2005), online; Fachrichtung Informationswissenschaft Saarbrücken (2006b), online; Holzinger (2006b), online

tendieren, viel mehr auszusprechen, als wenn eine Testperson dem Usability-Experten über ihre Eindrücke berichtet.¹⁵³

Die einfachste aller Methoden, wie *Holzinger* meint, ist die Field Observation. Usability-Experten beobachten Softwarebenutzer an ihren Arbeitsplätzen. Idealerweise sollte der Beobachter möglichst unsichtbar sein, um die gewohnte Arbeitsumgebung nicht zu stören. Eine unterstützende Erkenntnisquelle kann Data Logging darstellen, das Statistiken über die Benutzung des Systems durch einen Benutzer (z.B. Häufigkeit von Fehlermeldungen) bereitstellt. Diese einfache Methode, welche reelle Arbeitssituationen untersucht, ist erst sinnvoll anwendbar, wenn zumindest ein Prototyp existiert. Weitere Einschränkungen: Eine relativ große Anzahl an beobachteten Testpersonen ist nötig und vom Experten wird ein hohes Maß an Expertise erwartet.¹⁵⁴

Benutzerbefragungen, in Form von qualitativen Interviews oder standardisierten Fragebögen¹⁵⁵, welche von den Testpersonen auszufüllen sind, stellen eine indirekte Evaluationsmethode dar. Denn die Benutzer bewerten wird nicht das User Interface an sich; es werden ausschließlich Meinungen und Einstellungen der Benutzer zum User Interface gesammelt.¹⁵⁶ Die Fragen beziehen sich auf bestimmte Systemeigenschaften. Der Benutzer antwortet auf der Grundlage seiner Erfahrungen mit der Software. Als Vorteile einer Befragung gelten der im Vergleich zu anderen Testmethoden geringe Aufwand in der Durchführung, die Anwendbarkeit in allen Lebenszyklusphasen des Softwareprodukts sowie die Möglichkeit, unstrukturierte Probleme aufzufinden. Nachteile können daraus resultieren, dass sie anfällig für Übertreibungen sind, dass sie suggestive Fragestellungen enthalten können, und dass sie eine große Datenmenge produzieren, was den Aufwand der Auswertung in die Höhe treibt.¹⁵⁷ Um signifikante Aussagen treffen zu können, wird aber eine große Anzahl an Befragten benötigt: 30 Anwender gelten als das untere Limit für eine Studie.¹⁵⁸

Im Rahmen von Benutzerbefragungen wird vermeintlich nur die Benutzerzufriedenheit¹⁵⁹ mit der Software erhoben - dennoch stellt die Benutzerbefragung eine geeignete Methode dar, um auch die Effizienz und Effektivität einer Software zu bewerten. Das Konstrukt der User Satisfaction eignet sich also als übergreifender Maßstab für die Software Usability.

Cyert und *March* (1963) waren die ersten, welche das Konzept der User Satisfaction als Maßstab für den Erfolg von Softwaresystemen präsentierten. Heute stellt die Benutzerzufriedenheit

¹⁵³ vgl. *Holzinger* (2006b), online

¹⁵⁴ vgl. *Holzinger* (2006b), online

¹⁵⁵ vgl. Fachrichtung Informationswissenschaft Saarbrücken (2006b), online

¹⁵⁶ vgl. *Holzinger* (2005): S. 74

¹⁵⁷ vgl. *Oppermann et al.* (1992): S. 10

¹⁵⁸ vgl. *Holzinger* (2006b), online

¹⁵⁹ Der Begriff der User Satisfaction wird in *DIN EN ISO 9241-11* als positive Einstellung gegenüber der Benutzung der Software beschrieben. Vgl. Deutsches Institut für Normung (2003): *EN ISO 9241-11: 1998*. S. 4. Ähnlich auch die Definition von End-User Satisfaction nach *Doll* und *Torkzadeh*: „End-user satisfaction is [...] the affective attitude towards a specific computer application by someone who interacts with the application directly.“ *Doll/Torkzadeh* (1988): S. 261.

wohl den meist verbreiteten Ansatz zur Bewertung des Erfolgs einer Software dar.¹⁶⁰ Somers, Nelson und Karimi bewiesen 2003, dass das Konstrukt der User Satisfaction auch im Falle von ERP-Systemen eine der wichtigsten Determinanten für einen erfolgreichen Einsatz darstellt und eine geeignete Methode ist, um die Effizienz und Effektivität eines ERP-Systems zu beurteilen.¹⁶¹ 2005 stellten Sedera und Tan in ihrem Konferenzbeitrag „*User Satisfaction: An Overarching Measure for Enterprise System Success*“ ausdrücklich klar, dass sich das Konstrukt der User Satisfaction nicht als eine Dimension beim Evaluieren von betriebswirtschaftlicher Standardsoftware eignet. Vielmehr misst die Benutzerzufriedenheit den überspannenden Erfolg von ERP-Systemen.¹⁶² Diesen Erkenntnissen zufolge eignen sich Benutzerbefragungen, um den Erfolg des Einsatzes von Softwaresystemen im Allgemeinen und von ERP-Systemen im Speziellen zu beurteilen.

Nachdem User Centred Design und Usability Testing vorgestellt wurden, werden im nächsten Abschnitt die daraus resultierenden positiven Aspekte thematisiert: Die durch User Centred Design und Usability Testing geförderte Software Usability eröffnet - für Anwenderunternehmen wie für Softwarehersteller - erhebliche Nutzenpotentiale.

4 Nutzen von Software Usability

Schätzungen zufolge werden 20 Prozent der am Computer verbrachten Arbeitszeit auf Grund vermeidbarer Benutzungsprobleme nicht sinnvoll genutzt. Nicht nur die Kunden ergonomisch schlecht gestalteter Software haben Kosten infolge mangelnder Usability zu tragen; auf der Seite der Softwareanbieter sind es überlaufene Support-Hotlines und drohender Imageverlust; beides verursacht erhebliche Kosten.¹⁶³

Der Nutzen von Usability lässt sich in Erträgen des Einsatzes von Usability Engineering messen. Grundsätzlich setzt sich der Nutzen von Usability aus drei Komponenten zusammen: eine Steigerung der Produktivität beim Einsatz der Software, eine Reduktion anfallender Kosten bei ihrer Entwicklung und eine bessere Wettbewerbsfähigkeit des Produktes.¹⁶⁴ Der Wert der Usability bei Softwareprodukten, welche intensives und teures Training verlangen, was im Speziellen auf ERP-Systeme zutrifft, ist besonders groß.¹⁶⁵ Softwareanbieter bestätigen, dass die Nutzen von Usability die dafür aufzuwendenden Kosten deutlich überwiegen. Laut IBM, spart momentan jeder in Usability investierte Euro 10 bis 100 Euro.¹⁶⁶ Dieser Nutzen rechtfertigt die hohen Anteile am Softwareentwicklungsbudget, welche für Usability-Aspekte für gewöhnlich reserviert werden. Heute noch bei etwa 10 Prozent, werden diese zukünftig auf rund 20 Prozent anwachsen.¹⁶⁷

¹⁶⁰ vgl. Au/Ngai/Cheng (2002): S. 452

¹⁶¹ vgl. Somers/Nelson/Karimi (2003): S. 613

¹⁶² vgl. Sedera/Tan (2005): S. 973

¹⁶³ vgl. Geis/Hartwig (1998): S. 168

¹⁶⁴ vgl. Mutschler/Reichert (2004), online

¹⁶⁵ vgl. Black (2002), online

¹⁶⁶ vgl. IBM (o.J.), online

¹⁶⁷ vgl. Mutschler/Reichert (2004), online

4.1 Nutzen für Anwenderunternehmen

Die Qualität der Benutzerschnittstelle beeinflusst zunächst den entstehenden Schulungsbedarf und somit die Kosten, welche im Bereich Schulung aufgewendet werden müssen.¹⁶⁸ Die Höhe der Einsparungen ist vom konkreten Einzelfall abhängig: Eine Schätzung der gesparten Kosten lässt sich aus der Zeit ableiten, welche gespart wird, und aus der Anzahl der Personen, welche die Benutzung der Software pro Jahr erlernen.¹⁶⁹ Nielsen und Pernice Coyne geben an, dass bei verbesserter Usability eine Reduktion der Schulungskosten um 50 Prozent erwartet werden kann.¹⁷⁰ Neben den Erlernzeiten verkürzt Usability auch die Wiedererlernzeiten.¹⁷¹

Mitarbeiter arbeiten mit gesteigerter Produktivität, wenn sie von benutzerorientierter Software unterstützt werden.¹⁷² Der Zusammenhang zwischen Usability und Mitarbeiterproduktivität ist seit langer Zeit ein Thema: 1985 gaben in einer branchenübergreifenden Studie 90 Prozent der Manager an, dass sich die Produktivität ihrer Mitarbeiter sehr gesteigert hätte.¹⁷³ Bailey, Knox und Lynch veröffentlichten drei Jahre später eine Studie, in welcher sie einen beträchtlichen Unterschied in der Produktivität zwischen Anwendern zweier verschiedener Softwaresysteme nachweisen konnten.¹⁷⁴

Ergonomisch gestaltete Software reduziert die Fehlerrate, wie eine Fallstudie der New York Stock Exchange zeigt: Hier konnte die Fehlerrate um den Faktor 10 gesenkt werden.¹⁷⁵ Dray berichtet in einer weiteren Fallstudie von einem Unternehmen, das mit dem Re-Design der User Interfaces nach Usability-Prinzipien eine 25-prozentige Steigerung der Effizienz erreichte und die Benutzerfehler um 25 Prozent reduzierte.¹⁷⁶ In einer Studie mit dem Titel „*Rage Against the Machine. The Impact and Implications of Computer Rage on UK Business*“ wird angegeben, dass rund ein Viertel der End-User bei ihrer täglichen Arbeit von Computerfehlern verschiedenster Art gestört wird, wobei zwei Fünftel von ihnen dadurch bereits Abgabetermine versäumt haben. Ein Drittel gab für die durchschnittliche Dauer der Behebung des Problems mindestens eine Stunde an.¹⁷⁷

Zeichnet sich eine Software durch hohe Usability aus, sind die Anwender weniger häufig gezwungen, Hilfe bei den Mitarbeitern am Service-Desk oder bei ihren Kollegen zu suchen.¹⁷⁸ Die offensichtlichen Supportkosten, die hauptsächlich durch den Service Desk verursacht werden, können somit reduziert werden. Kann nur ein Service-Desk-Mitarbeiter eingespart werden, so ergeben sich für das Unternehmen bereits große finanzielle Vorteile: Denn das Gehalt eines Service-Desk-Mitarbeiters ist mit dem anderer IT-Spezialisten vergleichbar.¹⁷⁹ Die

¹⁶⁸ vgl. Preim (1999): S. 2; Donahue (2001): S. 34f; Kalbach (2003): S. 9

¹⁶⁹ vgl. Mantei/Teorey (1988): S. 435

¹⁷⁰ vgl. Nielsen/Pernice Coyne (2001), online

¹⁷¹ vgl. Usability Forum (o.J. c), online

¹⁷² vgl. Donahue (2001): S. 34

¹⁷³ vgl. Schneider (1985) nach Harrison/Henneman/Blatt (1994): S. 213

¹⁷⁴ vgl. Bailey/Knox/Lynch (1988): S. 207

¹⁷⁵ vgl. Gibbs (1997): S. 70f

¹⁷⁶ vgl. Dray (1995): S. 20

¹⁷⁷ vgl. MORI (1999), online

¹⁷⁸ vgl. Donahue (2001): S. 35

¹⁷⁹ vgl. o.V. (2000): S. 19

ökonomische Bedeutung des Peer-to-Peer-Support soll ebenso wenig unterschätzt werden: Einer Studie unter zehn großen US-amerikanischen Unternehmen zufolge, darunter auch Ford und Xerox, schlagen sich die Kosten der Hilfe unter Kollegen mit 6.000 bis 15.000 US-Dollar jährlich pro PC-Arbeitsplatz zu Buche.¹⁸⁰ Gut benutzbare Softwareprodukte schränken den Bedarf an Peer-to-Peer-Support ein und tragen somit zu einer erhöhten Produktivität des Personals bei.

Software Usability reduziert die Wartungskosten. 80 Prozent der Kosten, welche in einem Softwarelebenszyklus anfallen, sind der Wartung zuzurechnen. Der Großteil des Wartungsbedarfs entsteht auf Grund nicht beachteter Anforderungen der Anwender und anderer Probleme im Zusammenhang mit mangelnder Usability.¹⁸¹

Software, die auf die Bedürfnisse der Benutzer eingeht, steigert die Benutzerzufriedenheit deutlich.¹⁸² Neben der Zufriedenheit des Benutzers erweist sich auch die Benutzerfreude mit der Arbeit an einem Softwaresystem als entscheidendes Kriterium für die Bewertung der Software durch die Benutzer.¹⁸³

Aus einer gesteigerten Benutzerzufriedenheit resultiert auch eine höhere Arbeitszufriedenheit: Eine Studie über ergonomisch gestaltete Arbeitsumgebungen brachte folgendes Ergebnis: Jene Mitarbeiter, welche unter ergonomisch schlechteren Bedingungen arbeiteten, wiesen höhere Abwesenheitsraten, weniger Zufriedenheit mit der Arbeit und eine höhere Fluktuation auf.¹⁸⁴

Tabelle 10 fasst die wichtigsten Bereiche, durch welche Usability Kosten auf Seiten der Anwender spart, zusammen.

Tabelle 10: Nutzen von Software Usability für Anwenderunternehmen¹⁸⁵

Schulungskosten	Geringerer Schulungsbedarf	Geringe Zeitdauer für das Erlernen und Wiedererlernen eines Softwareprodukts
Support- und Wartungskosten	Geringerer Supportbedarf	Weniger Benutzerprobleme
	Geringerer Wartungsbedarf	Gesteigerte Effizienz bei der Wartung
Personalkosten	Geringere Fehlerrate	Gesteigerte Produktivität
	Erhöhte Benutzerzufriedenheit	Gesteigerte Produktivität
	Gesteigerte Arbeitszufriedenheit	Reduzierte Abwesenheitsraten, geringere Fluktuation

¹⁸⁰ vgl. Bulkeley (1992): S. B4

¹⁸¹ vgl. Pressman (1992) nach UPA (o.J.), online

¹⁸² vgl. Schneider (1985) nach Harrison/Henneman/Blatt (1994): S. 210

¹⁸³ *Burmester, Hassenzahl und Koller* entwickelten ein Zwei-Komponenten-Modell für die Bewertung der Nutzungsqualität einer Software, bestehend aus der wahrgenommenen pragmatischen und hedonistischen Qualität. Die erste der beiden setzt sich aus Usability und Nutzen zusammen und kann als aufgabenbezogene Qualität betrachtet werden. Die hedonistische Qualität hingegen bezieht sich auf die Wahrnehmung nicht-zielorientierter Aspekte, welche beispielsweise mit Attributen wie innovativ, originell, aufregend oder exklusiv beschrieben werden können. Vgl. *Burmester/Hassenzahl/Koller* (2002): S. 32ff

¹⁸⁴ vgl. Schneider (1985) nach Harrison/Henneman/Blatt (1994): S. 210

¹⁸⁵ eigene Darstellung

4.2 Nutzen für Softwarehersteller

Nicht nur die Kunden von Softwaresystemen profitieren schließlich von Software Usability. Auch für Softwareanbieter macht sich das Investieren in die Usability ihrer Produkte bezahlt.

Usability Engineering nimmt neue von Anwendern gewünschte Ideen auf. Somit entsteht Potential, Produktinnovationen zu lancieren, welche vom Markt nachgefragt werden.¹⁸⁶ Usability steigert somit die Erfolgssicherheit der auf den Markt gebrachten Software-Produkte. Eine benutzerorientierte Softwareentwicklung erhöht die Wahrscheinlichkeit für die Akzeptanz eines Produkts am Markt.¹⁸⁷

Ein Usability-Fokus führt zu einer verbesserten Planbarkeit des Softwareentwicklungsprojekts: Eine Studie über die Softwareentwicklungskosten zeigt, dass 63 Prozent der großen Softwareprojekte ihr Budgetziel nicht einhalten konnten. Die vier meist genannten Gründe für die höheren Ausgaben konnten auf eine mangelnde Benutzerorientierung im Entwicklungsprozess zurückgeführt werden; dazu zählten häufige Änderungsvorschläge durch User, übersehene Aufgabenstellungen, fehlendes Wissen des Users über seine Anforderungen an die Software und eine unzureichende Kommunikation zwischen Anwendern und Usability-Experten.¹⁸⁸

Ein frühes Erkennen von Problemen, zu Zeitpunkten, an denen die Korrektur noch leichter und günstiger ist, reduziert die Kosten der Fehlerbeseitigung.¹⁸⁹ Werden Methoden des Usability Testing in der Designphase angewandt, können bereits früh erforderliche Veränderungen durchgeführt werden, um spätere Probleme mit der Softwarenutzung zu verhindern. Damit können Softwarehersteller einen beachtlichen Nutzen erzielen: Die Software wurde noch nicht programmiert, Trainingshandbücher noch nicht publiziert und Benutzer noch nicht geschult. Verbesserungen, die erst in der Entwicklungsphase getätigt werden, erfordern möglicherweise das (teilweise) Neuprogrammieren der Software, das erneute Verfassen und Drucken der Handbücher sowie zusätzliche Trainingsmaßnahmen für bereits geschulte Endbenutzer.¹⁹⁰ Eine grobe Richtlinie zur Einschätzung des ökonomischen Nutzens eines frühen Usability-Fokus ist jene bereits 1988 veröffentlichte von *Gilb*: Ist ein Softwaresystem bereits in der Entwicklungsphase, muss für die Korrektur eines Problems zehnmal mehr ausgegeben werden, als dafür in der Designphase nötig gewesen wäre. Wird man erst nach Markteinführung des Systems auf einen Fehler aufmerksam, so kostet die Beseitigung rund hundertmal mehr als in der Designphase (siehe Abbildung 10).¹⁹¹ Ebenso 1988 erschienen ist der Artikel „*Cost/benefit analysis for incorporating human factors in the software lifecycle*“, in welchem die beiden Autoren *Mantei* und *Teorey* ein frühes Erkennen und Verbessern von Fehlern mit nur einem Viertel dessen bewerteten, was es zu einem späten Zeitpunkt kosten würde.¹⁹²

¹⁸⁶ vgl. Kalbach (2003): S. 11

¹⁸⁷ vgl. Kalbach (2003): S. 8; Eichinger (o.J. f), online

¹⁸⁸ vgl. Barker (2000), online

¹⁸⁹ vgl. Donahue (2001): S. 33; Kalbach (2003): S. 9

¹⁹⁰ vgl. Mayhew/Mantei (1994): S. 23

¹⁹¹ vgl. Gilb (1988): S. 221

¹⁹² vgl. Mantei/Teorey (1988): S. 435

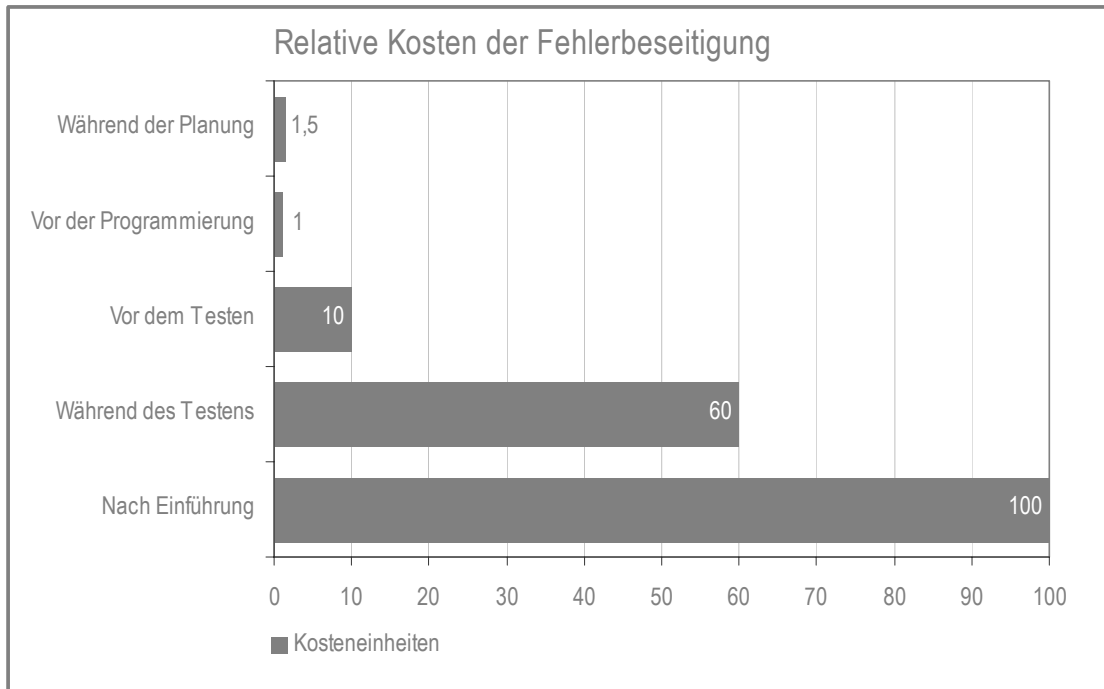


Abbildung 10: Kosten der Fehlerbeseitigung im Softwarelebenszyklus¹⁹³

Funktionalitäten, die für die Anwender von geringerer Wichtigkeit sind, müssen gar nicht erst entwickelt werden bzw. können auf später verschoben werden. Damit reduziert sich die Entwicklungszeit.¹⁹⁴

Ergonomisch gestaltete Software benötigt weniger Dokumentation. Außerdem gestaltet sich das Erstellen der benötigten Dokumentation als vergleichsweise einfach; es ist somit günstiger als bei nicht ergonomisch gestalteter Software.¹⁹⁵

Erfüllt die Software die Bedürfnisse der Zielgruppe, so fällt die Marktpositionierung und Neukundengewinnung leichter.¹⁹⁶ Softwarekunden betrachten die Leistungsfähigkeit und die Benutzungsfreundlichkeit als entscheidende Kriterien für die Kaufentscheidung.¹⁹⁷ Da Benutzer die Usability als wichtige Anforderung an ein Softwaresystem werten, steigert sie außerdem den Wert der Marke und die Loyalität der Kunden.¹⁹⁸ Ein Softwareprodukt, das die Anforderungen seiner Kunden erfüllt, erzielt eine höhere Kundenbindung. Softwareanbietern gelingt es, das Vertrauen der Kunden in das Unternehmen zu erhöhen.¹⁹⁹ Ferner bringt Usability Vorteile im Marketing und dient als Differenzierungsinstrument im Wettbewerb der Softwareanbieter.²⁰⁰

¹⁹³ modifiziert nach Gilb (1988): S. 221

¹⁹⁴ vgl. Kalbach (2003): S. 9

¹⁹⁵ vgl. Donahue (2001): S. 35

¹⁹⁶ vgl. Kalbach (2003): S. 10

¹⁹⁷ vgl. Harrison/Hennemann/Blatt (1994): S. 221

¹⁹⁸ vgl. Donahue (2001): S. 36

¹⁹⁹ vgl. Kalbach (2003): S. 10f

²⁰⁰ vgl. Mutschler/Reichert (2004), online; Donahue (2001): S. 36

Softwareanbieter, welche Produkte mit schlechter Usability auf den Markt bringen, riskieren negative Kritiken.²⁰¹

Einen Überblick über die ökonomisch bedeutenden Nutzeneffekte von Software Usability für Softwarehersteller gibt Tabelle 11.

Tabelle 11: Nutzen von Software Usability für Softwarehersteller²⁰²

Entwicklungskosten	Mehr Erfolgssicherheit	Bessere Kenntnis der Kundenbedürfnisse
	Kürzere Entwicklungszeit	Reduzierte Entwicklungszeit aufgrund detaillierten Wissens über Kundenbedürfnisse
	Effizientere Fehlerbeseitigung	Geringerer Aufwand der Fehlerbeseitigung in frühen Entwicklungsphasen
	Effizienteres Erstellen der Softwaredokumentation	Verminderter Bedarf an bzw. leichteres Erstellen der Softwaredokumentation
Risikopotential	Mehr Erfolgssicherheit	Erhöhte Wahrscheinlichkeit des Erfolgs eines Produktes am Markt
	Bessere Planbarkeit	Erhöhte Wahrscheinlichkeit, u.a. budgetäre Ziele des Entwicklungsprojekts einzuhalten
Marketingkosten	Effizientere Neukundengewinnung	Besseres Positionieren des Produkts am Markt
	Gesteigerter Markenwert	Höhere Kundenbindung
	Effizienteres Marketing	Differenzierteres Auftreten am Markt

²⁰¹ vgl. Kalbach (2003): S. 11; Donahue (2001): S. 36

²⁰² eigene Darstellung

III ENTERPRISE RESOURCE PLANNING UND TOTAL COST OF OWNERSHIP

Dieses Kapitel dient der Aufbereitung von Grundlagenwissen zu integrierter betriebswirtschaftlicher Standardsoftware sowie zu den Gesamtkosten des Einsatzes von Softwaresystemen.

1 Enterprise Resource Planning

Eine Enterprise Resource Planning Software, kurz ERP-Software genannt, bezeichnet eine integrierte betriebswirtschaftlich-administrative Standardanwendungssoftware.²⁰³ Neben einer Definition, was darunter zu verstehen ist, enthält dieses Kapitel einige Eckdaten zum ERP-Markt. Anschließend werden die Merkmale von ERP-Systemen sowie ihr Nutzen dargestellt.

1.1 Definition

Unter einem ERP-System wird betriebswirtschaftliche Standardsoftware verstanden, welche als Haupteigenschaften eine Prozessorientierung, die Zusammenführung aller wesentlichen kaufmännischen Geschäftsprozesse, die interne Integration aller Teilbereiche der Software und somit die Durchgängigkeit aller Geschäftsprozesse, die Speicherung der Daten in einer oder mehreren unabhängigen Datenbanken, die Einsetzbarkeit in vielen Branchen sowie die Anpassbarkeit an nationale Eigenschaften, eine einheitliche grafische Benutzeroberfläche und die Möglichkeit, die Software in einem gewissen Rahmen unternehmensindividuell anzupassen und zusätzliche Module selbst zu erstellen, aufweist.²⁰⁴

Im Wesentlichen werden mit ERP-Systemen Geschäftstransaktionen abgewickelt. Deshalb werden sie auch Transaktionssysteme oder in Anlehnung an den englischen Sprachgebrauch operative Informationssysteme - *day-to-day operations* bezeichnen den täglichen Betrieb - genannt. Operative Tätigkeiten eines Unternehmens sollen sich möglichst zeitnah steuern lassen, um Aktualität, Detaillierungsgrad und Genauigkeit der betrieblichen Daten sicherstellen zu können.²⁰⁵

²⁰³ vgl. Promberger/Schlager-Weidinger/Traxl (2003): S. 20; von Arb (1997), online

²⁰⁴ vgl. Steinbuch (1998): S. 214f

²⁰⁵ vgl. Hansen/Neumann (2005): S. 528

Eine ERP-Software ist ein Softwaresystem, das mehrere betriebswirtschaftliche Applikationen durch eine gemeinsame Datenbasis verbindet. All diesen Applikationen, welche Funktionalitäten aus Back Office und Front Office gleichermaßen abbilden, liegt eine prozessorientierte Sichtweise auf die Organisation zu Grunde. Indem sie den gesamten betrieblichen Ablauf in den verschiedenen Unternehmensbereichen steuern und kontrollieren, unterstützen ERP-Systeme das betriebliche Informationsmanagement.²⁰⁶ Stand in der ersten Generation der ERP-Systeme noch die interne Prozessunterstützung im Vordergrund, so sind Funktionalitäten des Customer Relationship Management sowie des Supply Chain Management in ERP^{II}-Systemen bereits integriert.²⁰⁷ Beispielsweise haben ERP-Anbieter ihre Systeme vielfach um Funktionen zur Unterstützung des Electronic Business ergänzt.²⁰⁸

1.2 Merkmale von ERP-Systemen

Betriebswirtschaftliche Software muss eine Reihe von Aspekten erfüllen, um als ERP-System klassifiziert zu werden. Zum einen unterstützen ERP-Systeme nahezu alle Geschäftsprozesse in der Wertschöpfungskette eines Unternehmens und optimieren somit den Datenfluss. Ein weiterer charakteristischer Aspekt ist der modulare Aufbau von ERP-Software. Daneben beschreiben die Merkmale der Standardisierung, der Flexibilität und der Internationalität ERP-Softwarelösungen. Im Folgenden soll auf die einzelnen Charakteristika näher eingegangen werden.

Integration bezieht sich sowohl auf die gemeinsame Verwendung von Daten (Datenintegration) als auch auf die durchgängige Verbindung mehrerer Geschäftsprozesse (Prozessintegration).²⁰⁹ Aufgrund der Tatsache, dass es nur eine Datenbasis für alle Module gibt, wird redundante Datenhaltung verhindert und Datenkonsistenz garantiert. Für Entscheidungen des strategischen und operativen Managements ist eine Datenintegration sehr wertvoll.²¹⁰ Eine Prozessintegration resultiert aus der IT-mäßigen Abbildung des ganzen Unternehmens mit seiner Aufbau- und Ablauforganisation im ERP-System (siehe Abbildung 11).²¹¹ Eine Integration findet sowohl zwischen den operativen Einheiten eines Unternehmens (beispielsweise den Niederlassungen) wie auch zwischen den verschiedenen Unternehmensfunktionen statt. Eine Transaktion in einem Teilsystem löst unmittelbar und automatisch ein Update in allen anderen Teilsystemen aus. Durch die Verlinkung aller Funktionen und Organisationseinheiten haben stets alle im Unternehmen Zugriff auf die relevanten Informationen.²¹² Der Vorteil der Integration liegt in der einheitlichen Datenbasis und infolgedessen darin, dass nur betriebswirtschaftlich konsistente Transaktionen durchgeführt werden können.²¹³

²⁰⁶ vgl. Promberger/Schlager-Weidinger/Traxl (2003): S. 39

²⁰⁷ vgl. Weston (2003): S. 49f

²⁰⁸ vgl. Gadatsch (2003): S. 394

²⁰⁹ vgl. Gadatsch (2003): S. 255f

²¹⁰ vgl. von Arb (1997), online

²¹¹ vgl. Davenport (1998): S. 124; Färbinger (2000), online

²¹² vgl. Gattiker/Goodhue (2000): S. 7017

²¹³ vgl. Friedl/Hilz/Pedell (2002): S. 164

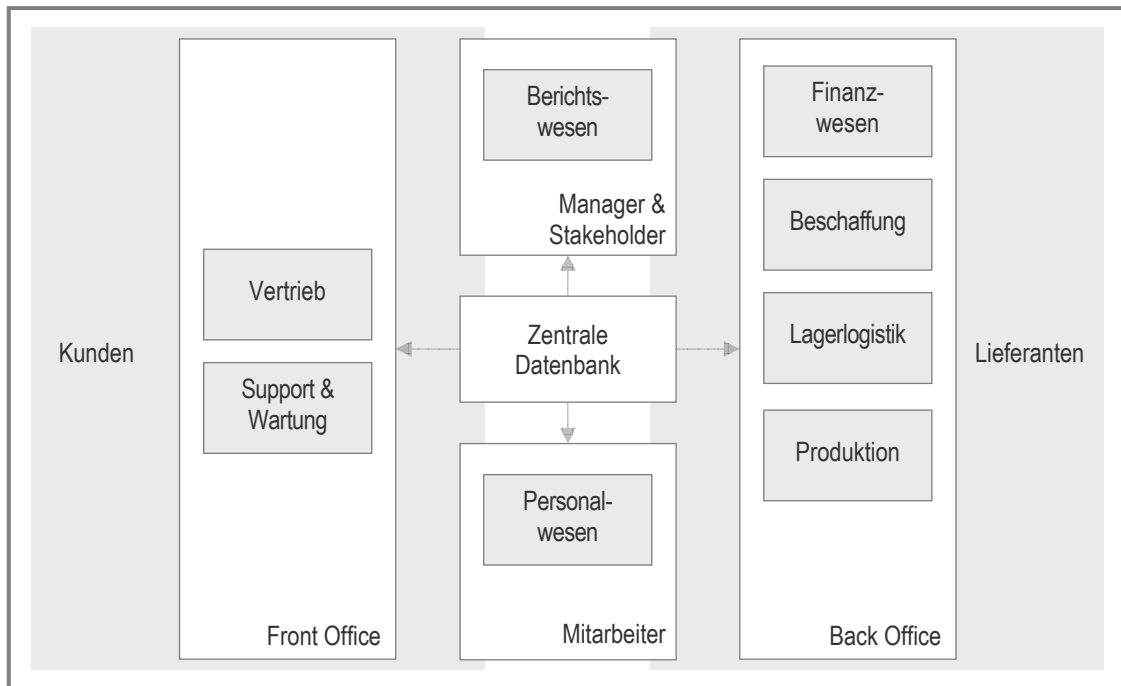


Abbildung 11: Integration der verschiedenen Unternehmensbereiche in einem ERP-System²¹⁴

Um die Komplexität von ERP-Software zu bewältigen, erfolgt eine Modularisierung. Hierbei werden einzelne kleine Einheiten (Module oder Komponenten) definiert, die über Schnittstellen miteinander kommunizieren.²¹⁵ Die einzelnen Module übernehmen für spezifische Arbeitsplatztypen eine funktionsorientierte Unterstützung. Typische Applikationen für ERP-Systeme sind Finanzwesen und Controlling, Produktionsplanung und -steuerung, Einkauf und Logistik, Vertrieb und Versand sowie Personalwesen.²¹⁶ Im Gegensatz zu ihren Vorgängern, Material Requirements Planning (MRP) und Manufacturing Resource Planning (MRP II) Systemen, sind sie branchen-neutral.²¹⁷ ERP-Module umfassen aber teilweise neben bestimmten branchenneutralen Funktionen auch branchenspezifische Aspekte.²¹⁸ Prozesse, welche strategische Wettbewerbsvorteile entstehen lassen, werden tendenziell in eigenentwickelten Systemen abgebildet und mit diesen gesteuert.²¹⁹ Um die Potenziale einer ERP-Software möglichst umfassend auszuschöpfen, sollten Unternehmen möglichst viele funktionale Bereiche im System abbilden.

Unternehmen können, wie in Abbildung 12 dargestellt, auf einem Basissystem die jeweils benötigten Module implementieren. Durch die Komponentenstruktur kann die Implementierung des Gesamtsystems in Teilschritten über einen größeren Zeitraum erfolgen.²²⁰

²¹⁴ modifiziert nach Davenport (1998): S. 124

²¹⁵ vgl. Fink/Schneiderei/Voß (2005): S. 211

²¹⁶ vgl. Friedl/Hilz/Pedell (2002): S. 164; Gadatsch/Mayer (2004): S. 70

²¹⁷ vgl. Hansmann/Neumann (2002): S. 327

²¹⁸ vgl. Fink/Schneiderei/Voß (2005): S. 214; Hansen/Neumann (2005): S. 589

²¹⁹ vgl. Gadatsch (2003): S. 299; Hansen/Neumann (2005): S. 534

²²⁰ vgl. Hansen/Neumann (2005): S. 534

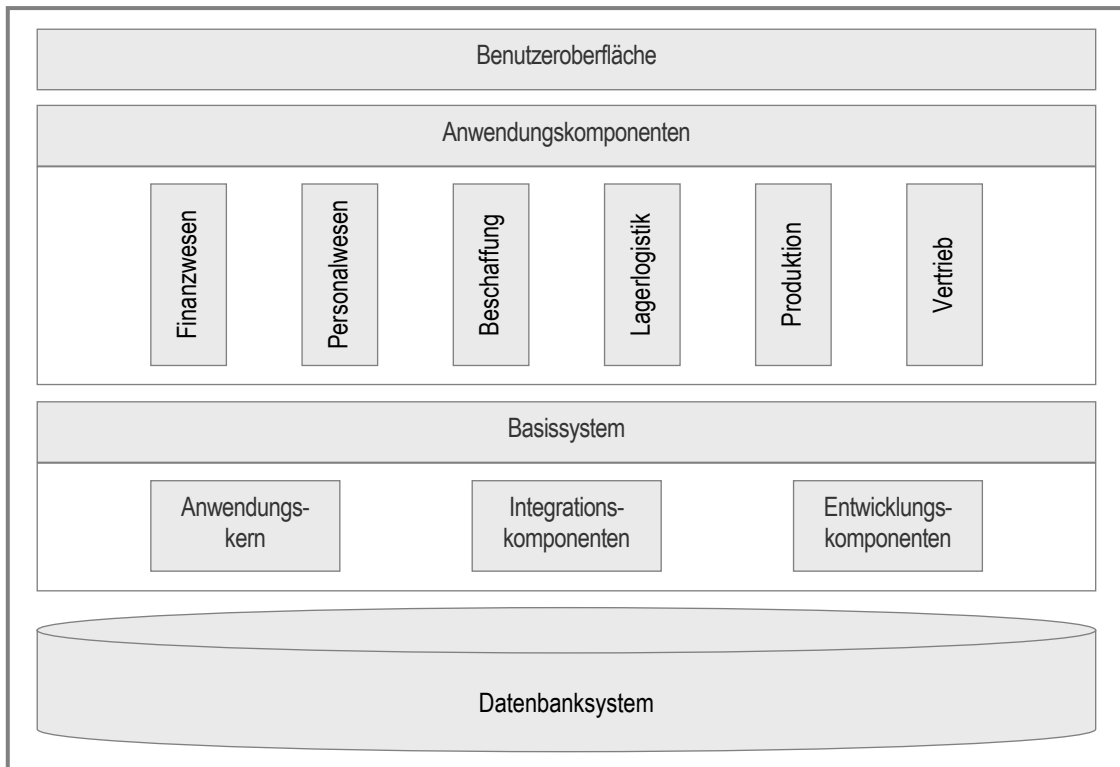


Abbildung 12: Typische Komponenten eines ERP-Systems²²¹

Ein konsistentes Design der Benutzeroberfläche in den einzelnen Modulen, was für den Bildschirmaufbau mit Fenstern, Menüs, Icons und Farben ebenso wie für die Tastenbelegung und Mausbedienung gilt, ist im Falle einer integrierten Standardsoftware wesentlich.²²²

Während die Modularisierung ein ERP-System bislang in betriebswirtschaftliche Funktionen unterteilt, wird aus heutiger Sicht für das kommende Jahrzehnt eine noch stärkere Modularisierung erwartet. Unter dem Namen service-orientierte Architektur (SOA) bekannt, wird bei der Gestaltung die Funktionalität des Systems in einzelne Dienste unterteilt, die jeweils eng umgrenzte Funktionen abbilden. Vordefinierte Schnittstellen dienen der Interaktion zwischen den einzelnen Diensten.²²³

Eine betriebswirtschaftliche Standardsoftware bezeichnet ein System, das auf Allgemeingültigkeit und Mehrfachverwendung ausgelegt ist und somit für den Einsatz in verschiedenen Branchen und in Unternehmen verschiedener Größe zur Verfügung steht.²²⁴ Die Standardisierung bezieht sich sowohl auf die Daten- als auch auf die Prozessstandardisierung.²²⁵ Meist wird Standardsoftware in der Praxis auf unternehmensindividuelle Gegebenheiten und Anforderungen angepasst. Diese Anpassung erfolgt durch Customizing und - falls notwendig - durch

²²¹ modifiziert nach Hansen/Neumann (2005): S. 530

²²² vgl. Hansen/Neumann (2005): S. 535

²²³ vgl. Fink/Schneiderei/Voß: S. 211; Hansen/Neumann (2005): S. 532

²²⁴ vgl. Gadatsch (2003): S. 298; Fink/Schneiderei/Voß (2005): S. 213

²²⁵ vgl. Nah (2001): 400

Ergänzungsprogrammierung.²²⁶ Was das Customizing betrifft, so boten frühe Generationen von Standardsoftware - wenn überhaupt - nur geringe Möglichkeiten der kundenspezifischen Individualisierung. Heute sind Releaseupdates idealerweise so gestaltet, dass die im Customizing vorgenommenen Einstellungen übernommen werden.²²⁷ Die Programmierung von zusätzlichen anwenderspezifischen Funktionen kann im Laufe der Zeit sehr aufwändig werden, da bei Versionswechsel die Anpassung notwendig werden kann. Es ist ratsam, die Geschäftsprozesse durch Business Process Reengineering (BPR)²²⁸ an die Software anzupassen und auf eine Adaptierung der Software weitestgehend zu verzichten.²²⁹

Betriebswirtschaftliche Standardsoftware kennzeichnet sich durch einen gewissen Grad an technischer und betriebswirtschaftlicher Flexibilität. So lassen sich ERP-Systeme auf mehreren Hardwareplattformen und Betriebssystemen einsetzen. Die Orientierung an Client/Server-Architekturen und offenen Standards schafft weitgehende Möglichkeiten der Skalierung.²³⁰ Wie eine betriebswirtschaftliche Anpassung vorgenommen werden kann, z.B. durch Parametersetzung (Customizing) oder durch Eingriffe an definierten Schnittstellen, wurde bereits oben erwähnt.

Unternehmen, welche ein ERP-Softwareprodukt in verschiedenen Ländern einsetzen wollen, profitieren von der Internationalität einer Standardsoftware. Der Einsatz unterschiedlicher Sprachen, Datumsformate, Kontenpläne, Währungen, Lohn- und Gehaltsabrechnungsmodalitäten, Steuern, gesetzlicher Rechnungslegungs- und Prüfungsvorschriften sowie anderer betriebswirtschaftlich relevanter nationaler Rechtsvorschriften wird unterstützt. Somit werden zentrale Anforderungen von international tätigen Unternehmen erfüllt.²³¹

1.3 Eckdaten zum ERP-Markt

2004 investierten Unternehmen weltweit 21 Milliarden US-Dollar²³² in ERP-Systeme. Bis 2008 wird ein Anstieg auf knapp 25 Milliarden erwartet. So die (prognostizierten) Zahlen von *Forrester Research* (siehe Abbildung 13).²³³ Die genauen Zahlen der verschiedenen Analysten variieren, dennoch lässt sich die ungefähre Größenordnung des Marktes erahnen: *Butters* berichtete für 2004 über 5,8 Milliarden US-Dollar an Lizenzgebühren - das liegt nahe bei jener Ziffer, welche auch *Forrester Research* erhoben hatte (nämlich knapp 6,2 Milliarden).²³⁴ *AMR Research* bezifferte das weltweite Marktvolumen der ERP-Softwarehäuser für 2004 mit 23,6 Milliarden US-Dollar.²³⁵ Die Einschätzung von *AMR Research*, dass bis 2009 am europäischen ERP-Markt ein jährlicher Umsatzzuwachs von 7 Prozent erwartet werden kann, fällt im

²²⁶ vgl. Hansen/Neumann (2005): S. 534; Buxmann/König (1997): S. 335; Fink/Schneiderei/Voß (2005): S. 214

²²⁷ vgl. Gadatsch (2003): S. 298; Hansen/Neumann (2005): S. 535

²²⁸ Das Abgleichen der betrieblichen Prozesse und jener des ERP-Systems ist einer der bedeutendsten Gründe für die Unzufriedenheit mit ERP-Systemen. Vgl. Bishu/Kleiner/Drury (2001): S. 149

²²⁹ vgl. Nah (2001): 401; Hansen/Neumann (2005): S. 535; Parry (2005): S. 24

²³⁰ vgl. von Arb (1997), online

²³¹ vgl. Hansen/Neumann (2005): S. 535; von Arb (1997), online

²³² Diese Zahlen spiegeln ausschließlich die Umsätze der ERP- Softwarehersteller wider. Zusätzliche Consulting-Dienstleistungen im Bereich ERP-Systeme bleiben unberücksichtigt, ebenso natürlich der interne Aufwand über den gesamten ERP-Produktzyklus für Anwenderunternehmen. Dies gilt für alle Zahlen in diesem Absatz.

²³³ vgl. *Forrester Research* (2005) nach Karcher (2005): S. 27

²³⁴ vgl. *Butters* (2005), online

²³⁵ vgl. *Reilly* (2005), online

Vergleich zur weltweit prognostizierten Marktentwicklung von *Forrester Research* deutlich positiver aus. Mit einem Marktvolumen von 8,8 Milliarden US-Dollar bildet der europäische Markt heute bereits etwa 37 Prozent des Weltmarkts.²³⁶ Es ist also davon auszugehen, dass das europäische Segment des ERP-Markts weiterhin seine Vorreiterrolle behält.

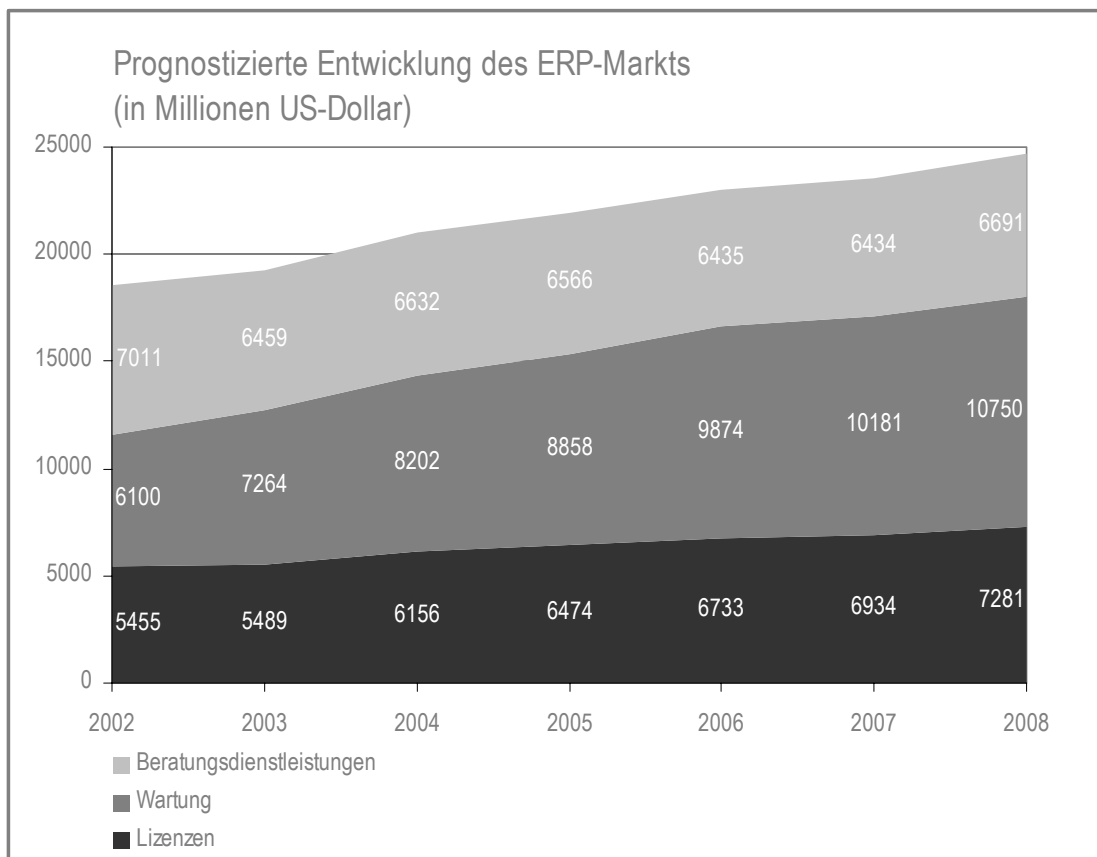


Abbildung 13: Prognostizierte Entwicklung des Markts für Enterprise Resource Planning²³⁷

Doch auf der Suche nach dem Marktvolumen von ERP-Softwaresystemen stößt man auf so manch unterschiedliche Ziffer; diese einzuschätzen und zu bewerten, fällt aufgrund fehlender Quellenangaben beziehungsweise einer fehlenden Angabe, welche Umsätze berücksichtigt wurden, nicht leicht. Als Beispiel sei *Davenport* genannt, der bereits 1998 in seinem viel zitierten Artikel „*Putting the Enterprise into the Enterprise System*“ angab, dass die globale Wirtschaft jährlich etwa 10 Milliarden US-Dollar für ERP-Systeme an sich und einen Betrag in ähnlicher Höhe zusätzlich für ERP-spezifisches Consulting ausgibt.²³⁸ Ebenso 1999 veröffentlicht, prognostizierte die Unternehmensberatung *Frost & Sullivan* einen Anstieg der Umsätze für ERP-Software alleine in Europa auf etwa 19 Milliarden US-Dollar.²³⁹ Man sieht, zwei sehr widersprüchliche Angaben.

²³⁶ vgl. Reilly (2005), online

²³⁷ eigene Darstellung, Daten entnommen aus Forrester Research (2005) nach Karcher (2005): S. 27

²³⁸ vgl. Davenport (1998): S. 122

²³⁹ vgl. Frost/Sullivan (1999) nach Gadatsch (2003): S. 252f

Laut einer von *Bernroider* und *Koch* durchgeführten Untersuchung des österreichischen ERP-Markts betragen die Kosten für die Implementierung einer betriebswirtschaftlichen Standardsoftwarelösung im österreichischen Markt durchschnittlich 944.750 Euro (Standardabweichung: 1,3 Millionen Euro) - bei einer mittleren Implementierungsdauer von 43 Wochen (39 Wochen). Zusätzlich kommen Kosten für den im Durchschnitt 26 Wochen (26 Wochen) dauernden Entscheidungsprozess in der Höhe von 63.660 Euro (116.280 Euro). Zwischen der Entscheidung und dem Beginn der Implementierung vergehen im Mittel 25 Wochen (32 Wochen).²⁴⁰

Eine Studie über die Verbreitung von ERP-Systemen am österreichischen Markt ergab, dass 19,3 Prozent aller Unternehmen ein ERP-System einsetzen beziehungsweise bereits eines zur Implementierung ausgewählt haben. Rund zwei Drittel der Großbetriebe (67,6 Prozent) steuern ihre Geschäftsprozesse mit einem ERP-System. Unter den kleinen und mittleren Unternehmen (KMU) liegt dieser Anteil bei 15,5 Prozent.²⁴¹

Den gesamten europäischen Markt für ERP betrachtend, ergibt sich folgendes Bild: Beinahe die Hälfte aller Industrieunternehmen in den von der *European Manufacturing Study 2003/2004* untersuchten Ländern setzt kommerzielle ERP-Softwarepakete ein.²⁴² Ähnliche Ergebnisse zeigt die *European e-Business Survey 2005*, welche die rasch zunehmende Verbreitung von ERP-Systemen in den meisten Sektoren bestätigt. Rund 28 Prozent aller Unternehmen in den sieben untersuchten EU-Staaten²⁴³ setzen ein betriebswirtschaftliches Standardsoftwaresystem ein. Lässt man kleine Unternehmen außer Acht, so ergibt sich für mittlere Unternehmen eine ERP-Verbreitungsrate von 33 Prozent, beziehungsweise 59 Prozent für Großbetriebe.²⁴⁴

Was die verarbeitende Industrie angeht, so nennt die *European Manufacturing Study 2003/2004* die Schweiz, Österreich und Deutschland als die führenden europäischen Länder, gemessen am Einsatz von ERP-Systemen in Industriebetrieben. 57 Prozent der österreichischen Unternehmen in der verarbeitenden Industrie setzen ein ERP-System für Informationsmanagementzwecke ein.²⁴⁵

Abschließend werden in Abbildung 14 ausgewählte Ergebnisse zur Verbreitung von ERP-Systemen aus der *European e-Business Survey 2005* sowie aus der *European Manufacturing Study 2003/04* dargestellt.

²⁴⁰ vgl. Bernroider/Koch (2000): S. 335

²⁴¹ vgl. Bernroider/Hampel (2005): S. 307f

²⁴² Zu den untersuchten Ländern zählen Deutschland, Frankreich, Italien, Kroatien, Österreich, die Schweiz, Slowenien, die Türkei sowie das Vereinigte Königreich. Vgl. Armbruster et al. (2005), online

²⁴³ Nämlich Deutschland, Frankreich, Italien, Polen, Spanien, Tschechien und das Vereinigte Königreich.

²⁴⁴ vgl. European Commission (2005), online

²⁴⁵ vgl. Armbruster et al. (2005), online

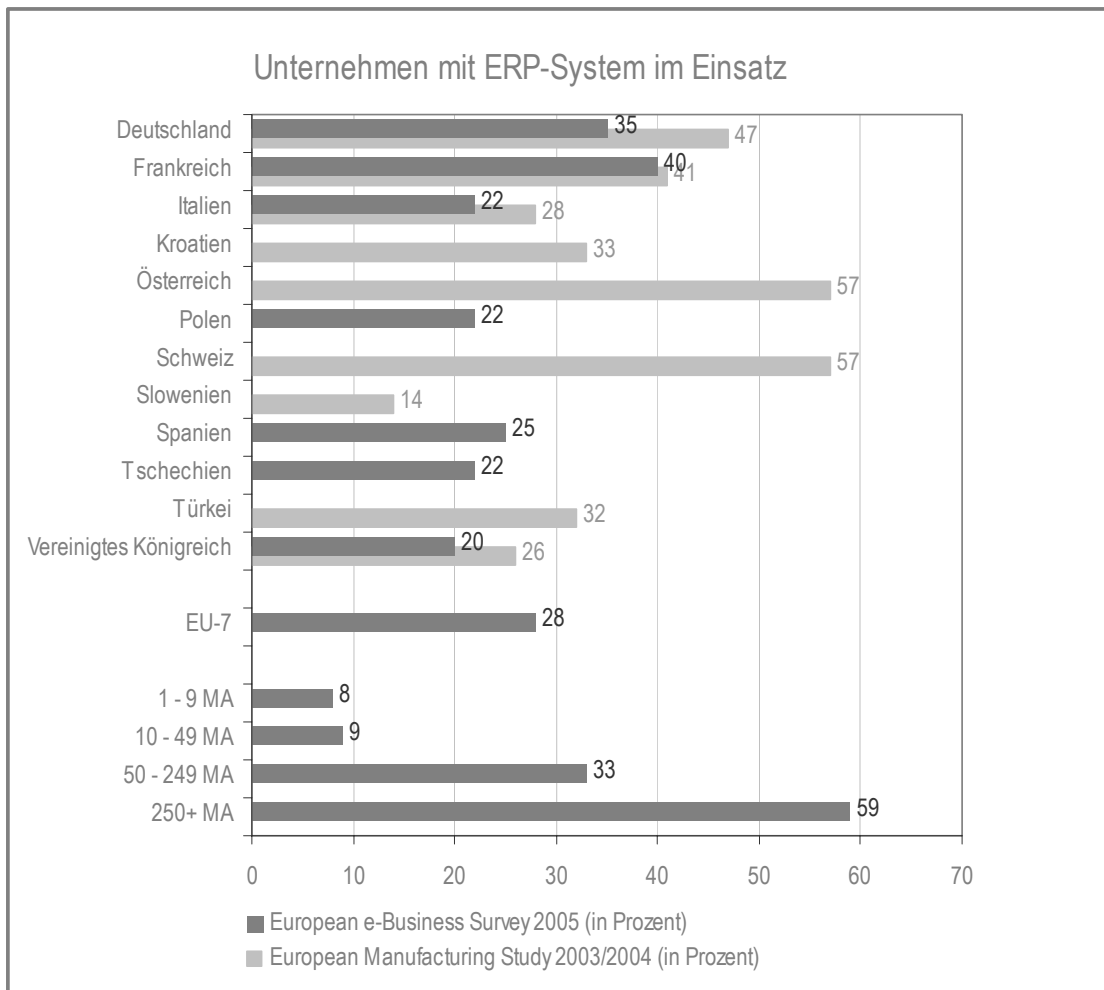


Abbildung 14: Verbreitung von ERP-Systemen²⁴⁶

1.4 Nutzen des Einsatzes von ERP-Systemen

Trotz des enormen Aufwands zeigt der Erfolg von ERP-Systemen, dass sich Unternehmen von ihrem Einsatz erhebliche Nutzenpotenziale versprechen.²⁴⁷ Das Management hofft, Produktion und Verwaltung effizienter zu gestalten; die Projektverantwortlichen fasziniert die Idee einer integrierten Datenbasis. Die Erwartung, der Nutzen des Systems werde sich wie von selbst einstellen, wird aber häufig enttäuscht. Denn ein ERP-System legt zwar die Basis für den Nutzen; dieser muss jedoch aktiv durch prozessgestaltende Maßnahmen, organisatorische Änderungen sowie Aus- und Weiterbildung der Mitarbeiter erarbeitet werden.²⁴⁸ Die technologische Innovation alleine führt zu keiner wesentlichen Leistungssteigerung; erst in Kombination mit Prozessinnovationen können Unternehmen mit Hilfe von ERP-Systemen Nutzen generieren.²⁴⁹

²⁴⁶ eigene Darstellung, Daten entnommen aus Armbruster et al. (2005), online; European Commission (2005), online

²⁴⁷ vgl. Gadatsch (2003): S. 372

²⁴⁸ vgl. Martin/Lempp/Mauterer (2003): S. 73

²⁴⁹ vgl. Clark/Stoddard (1996): S. 24

Zudem sei vorausgeschickt, dass eine quantitative Messung des Nutzens schwierig ist.²⁵⁰ Denn direkt monetär messbare Nutzeneffekte, wie Bestandssenkung, Durchlaufzeitverkürzung oder die Reduktion von Wartungskosten, bilden nur den kleineren Teil des Gesamtnutzens. Eine Studie mit 260 Unternehmen, die ERP-Systeme eingeführt haben, zeigt qualitativ messbare Nutzenaspekte als hauptverantwortlich für den Gesamtnutzen eines ERP-Einsatzes. Dieser substantielle Mehrnutzen gegenüber der Situation vor der ERP-Einführung resultierte beispielsweise aus einer erhöhten Liefertermintreue sowie einer verbesserten Prozessflexibilität und -transparenz.²⁵¹ Über das Ausmaß der einzelnen Nutzeneffekte, etwa einem verbesserten Prozessablauf²⁵² oder der Integration von Daten mit Geschäftspartnern²⁵³, eine Aussage zu treffen, ist jedoch schwierig.²⁵⁴

Unbestritten ist aber, dass der Einsatz von integrierter betriebswirtschaftlicher Standardsoftware ein großes Potential für Effizienzsteigerungen in Unternehmen birgt. Tabelle 12 dient dazu, einen Überblick über die diversen Nutzenpotentiale von ERP-Systemen zu geben, und orientiert sich dabei an der Systematik von *Martin, Mauterer* und *Gemünden*, deren 2002 erschienene Artikel „*Systematisierung des Nutzens von ERP-Systemen in der Fertigungsindustrie*“ erstmalig versucht, die in der Literatur verstreut genannten Nutzen von ERP-Systemen logisch einzuordnen und in Fallstudien auf ihre Richtigkeit zu überprüfen.²⁵⁵

Die Autoren konnten für die Kriterien Geschäftsprozesse, Kunden- und Marktorientierung, Produktivität und Wirtschaftlichkeit sowie Effizienz der Informationsgewinnung teilweise beachtliche Verbesserungen bei den untersuchten Unternehmen nachweisen. Bei der Mitarbeiter-ebene hing es vor allem vom Aufgabengebiet und der grundsätzlichen Arbeitseinstellung des einzelnen Mitarbeiters sowie von der aufgabengerechten Adaption des Systems ab, wie sich Motivation und Akzeptanz entwickelten. Das abschließende Urteil der Studie belegt, dass viele der beobachteten Nutzenpotenziale aufgrund der Standardisierung und Integration realisiert werden können. Um diese Vorteile ausschöpfen zu können, bedarf es jedoch höheren Anforderungen an die Datenstruktur und einer höheren Systemkomplexität, was wiederum dazu führt, dass Nutzenabstriche auf Mitarbeiter-ebene gemacht werden müssen. So zeigten sich Akzeptanz, Motivation und Leistungsfähigkeit der Mitarbeiter als zentrale Probleme bei der Einführung betriebs-wirtschaftlicher Standardsoftware.²⁵⁶

²⁵⁰ vgl. Martin (2003): S. 66; Schumann (1993): S. 168

²⁵¹ vgl. Martin (2003): S. 66

²⁵² vgl. Mukhopadhyay/Rajiv/Srinivasan (1997): S. 1646

²⁵³ vgl. Holland (1995): S. 132

²⁵⁴ vgl. Clark/Stoddard (1996): S. 24

²⁵⁵ vgl. Martin/Mauterer/Gemünden (2002): S. 109f

²⁵⁶ vgl. Martin/Mauterer/Gemünden (2002): S. 115

Tabelle 12: Nutzenkategorien von ERP-Systemen²⁵⁷

Prozesseffizienz (Geschäftsprozesse)	<p>Eine erhöhte Prozesseffizienz drückt sich in einer Verbesserung der Geschäftsprozesse hinsichtlich Kosten, Qualität und Zeit aus. ERP-Systeme unterstützen Unternehmen mit den in ihnen enthaltenen Prozessen, die als Best Practice angesehen werden können.</p> <p>In der Praxis gelingt es mit dem Einsatz eines ERP-Systems, die Durchlaufzeiten von Aufträgen zu reduzieren und die Liefertermintreue zu erhöhen. Daneben führt ein ERP-Einsatz zu einer verbesserten Prozessflexibilität hinsichtlich der Implementierung zusätzlicher Funktionalitäten und Technologien als Erweiterungen des Hauptsystems. Ebenso berichten Unternehmen von einer gesteigerten Prozesstransparenz; z.B. die für Pharma- und Lebensmittelindustrie wertvolle Chargenrückverfolgung.</p>
Markteffizienz (Kunden- und Marktorientierung)	<p>Eine gesteigerte Markteffizienz resultiert aufgrund einer verbesserten Nutzung von Chancen auf den Absatzmärkten sowie auf den Beschaffungsmärkten im Sinne eines koordinierten Auftretens gegenüber den Geschäftspartnern.</p> <p>Was die Lieferantenseite betrifft, kann das Ziel der Markteffizienz durch eine verbesserte Zusammenarbeit mit Lieferanten oder durch Bündelung der Nachfrage zur Stärkung der Marktposition angestrebt werden. Für den Kundenservice hat die Informationstechnologie eine ebenso wesentliche Bedeutung.</p>
Ressourceneffizienz (Produktivität und Wirtschaftlichkeit)	<p>Eine verbesserte Ressourceneffizienz wird durch eine gesteigerte Produktivität und Wirtschaftlichkeit im Sinne einer effizienteren Nutzung der Ressourcen (Mitarbeiter, Anlagen, Maschinen, Gebäude und Kapital) erzielt.</p> <p>Gemessen an Kennzahlen wie Kapazitätsauslastung in der Produktion, Lagerbestand oder Anzahl der benötigten Mitarbeiter können ERP-Systeme die Ressourceneffizienz steigern. Außerdem liefern sie die notwendige Informationsbasis bzw. Planungs- und Steuerungsfunktionen für Analysen zur Messung bzw. Verbesserung der Produktivität.</p>
Delegationseffizienz (Effizienz der Informationsgewinnung)	<p>Die Delegationseffizienz bezieht sich auf die Nutzung des Problemlösungspotentials hierarchisch übergeordneter Einheiten.</p> <p>Ein ERP-System senkt die Kosten der Informationsverarbeitung und -weitergabe durch eine höhere Geschwindigkeit, Qualität, Verlässlichkeit und unternehmensweite Vergleichbarkeit von Auswertungen und Analysen.</p>
Motivationseffizienz (Mitarbeiter)	<p>Motivationseffizienz bezeichnet die Fähigkeit eines Unternehmens, die Diskrepanz zwischen den Zielen des Unternehmens und jenen der Mitarbeiter zu überwinden, indem Voraussetzungen für ein unternehmenszielkonformes Verhalten der Mitarbeiter geschaffen werden.</p> <p>In der Praxis konnte festgestellt werden, dass der persönliche Nutzen für die Mitarbeiter stark von der individuellen Arbeitseinstellung und dem jeweiligen Aufgabenbereich abhängt. Ein Unterschied zwischen Mitarbeitern, die das ERP-System hauptsächlich zur Informationsgewinnung nutzen, und jenen, welche überwiegend für die Eingabe und Pflege bestimmter Daten zuständig sind, ist bemerkbar. Für letztere sind Aspekte der Usability besonders relevant. Das Top-Management zeigt überraschenderweise kurz nach Einführung der Software mehr Berührungsängste als die eigentlichen Anwender.</p>

Die wesentlichsten Nutzen des Einsatzes von ERP-Systemen können jedoch unter nur zwei Aspekten subsumiert werden: Zum einen profitieren Unternehmen vom Ende der Fragmentierung von Informationen.²⁵⁸ Zum anderen bringt ERP-Software vielschichtige Nutzeneffekte

²⁵⁷ eigene Darstellung, Daten entnommen aus Martin/Mauterer/Gemünden (2002): 111ff; Gadatsch (2003): S. 372f

²⁵⁸ vgl. Davenport (1998): S. 123

hinsichtlich der Geschäftsprozesse. So bietet ERP-Software eine sinnvolle Möglichkeit, die betrieblichen Prozesse zu verbessern, und wird deshalb zunehmend als Gestaltungselement für Prozessmanagement eingesetzt.²⁵⁹ Schließlich wird auch die Prozessflexibilität erhöht; Kunden und Lieferanten können mit vergleichsweise geringem Aufwand miteinbezogen werden.²⁶⁰

Die Ergebnisse einer Studie von *Buxmann* und *König*, welche im Zusammenhang mit der Organisationsgestaltung bei der Einführung betrieblicher Standardsoftware auch die Bedeutung der Nutzenarten einer ERP-Einführung - oder genauer der Einführung des Systems *SAP R/3* - aus Sicht der Unternehmen untersuchten, bestätigen diese beiden zentralen Nutzenaspekte. Abbildung 15 stellt die fünf wichtigsten - im Rahmen der Studie erhobenen - Nutzenarten dar.²⁶¹

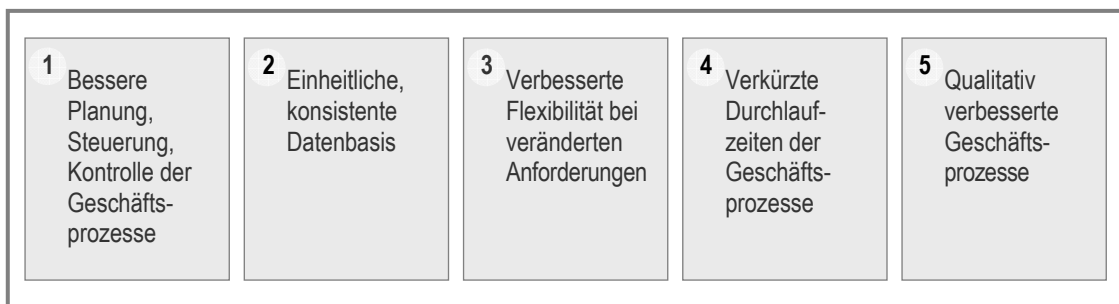


Abbildung 15: Die fünf wichtigsten Nutzenarten von SAP R/3 aus Sicht der Unternehmen²⁶²

Die in der Studie aufgezeigten Nutzeneffekte liegen also primär in einem verbesserten Prozessmanagement, welches sich nach Angabe der Unternehmen in einer besseren Planung, Steuerung und Kontrolle, einer höheren Flexibilität, einer verbesserten Qualität sowie kürzeren Durchlaufzeiten niederschlägt. In einer einheitlichen und konsistenten Datenbasis sehen die Unternehmen die zweitwichtigste Nutzenart des Einsatzes einer betrieblichen Standardsoftware. *Buxmann* und *König* vermerken, dass Rationalisierungseffekte, wie Kosteneinsparungen durch einen geringeren Materialverbrauch oder die Ersetzung von Humanressourcen durch maschinelle Aufgabenträger, als relativ unbedeutende Vorteile eingestuft wurden.²⁶³

Auf die beiden wesentlichsten Nutzenaspekte, die Prozessorientierung von ERP-Systemen und sich daraus ergebende Potentiale sowie die Integration der betrieblichen Daten in einer Datenbasis, soll im Folgenden noch etwas näher eingegangen werden.

ERP-Systeme werden immer häufiger als Hebel für Prozessverbesserungen verstanden. Während in der Vergangenheit mit der Einführung betriebswirtschaftlicher Standardsoftware insbesondere die IT-gestützte Abdeckung wichtiger Unternehmensfunktionen verbunden wurde, steht heute die Verbesserung der betrieblichen Geschäftsprozesse im Vordergrund.²⁶⁴

²⁵⁹ vgl. Harmon (2003): S. 282; Gadatsch (2003): S. 247

²⁶⁰ vgl. Hansen/Neumann (2005): S. 536

²⁶¹ vgl. Buxmann/König (1995): S. 163

²⁶² eigene Darstellung, Daten entnommen aus Buxmann/König (1995): S. 163

²⁶³ vgl. Buxmann/König (1995): S. 163

²⁶⁴ vgl. Gadatsch (2003): S. 372

Gefragt nach Prozessinnovationen, antworteten 45 Prozent der in der *European e-Business Survey 2005* untersuchten Unternehmen, dass sie im Jahr 2004 einen internen Prozess neu geschaffen beziehungsweise signifikant modifiziert haben. Die überwiegende Mehrheit (73 Prozent) dieser Prozessinnovationen in europäischen Unternehmen kann der betrieblichen IT zugeschrieben werden (siehe Abbildung 16).²⁶⁵

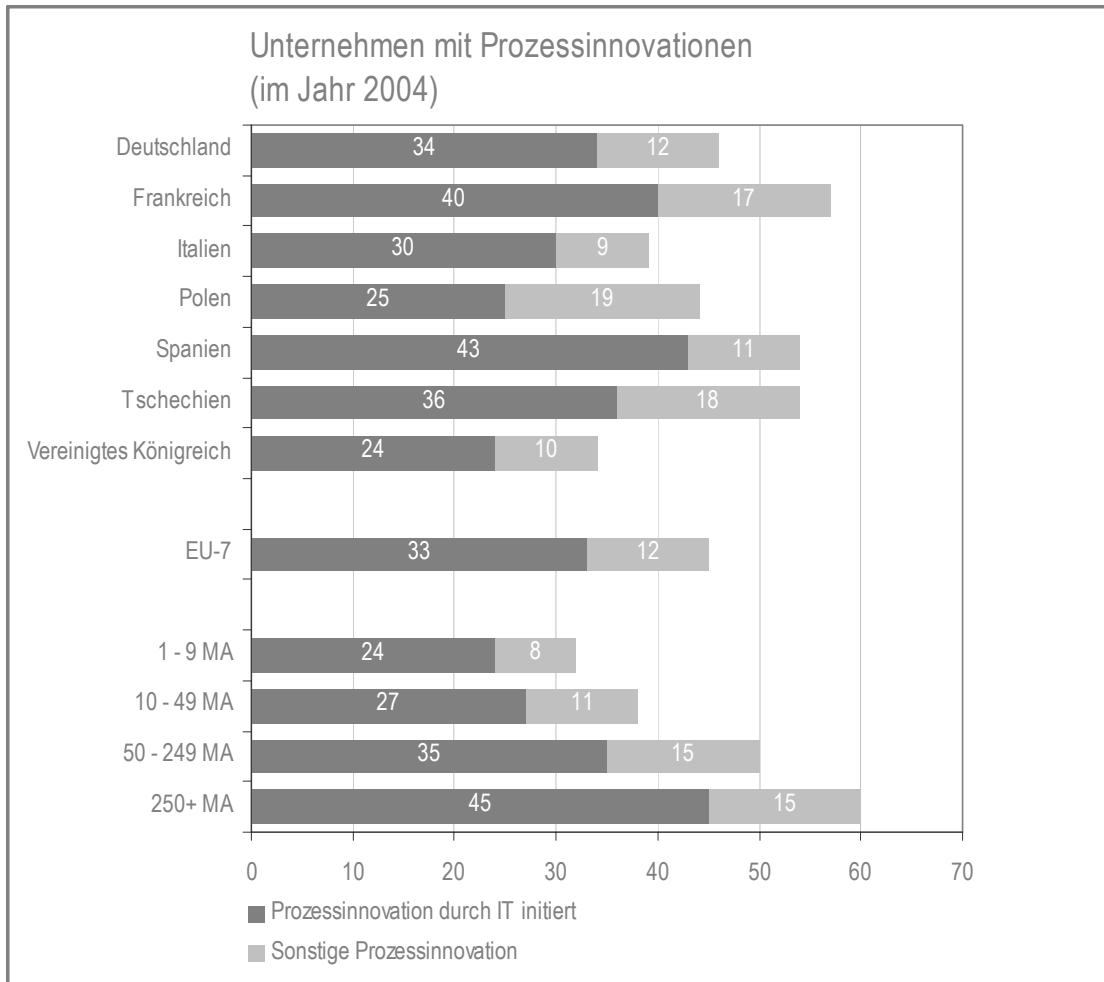


Abbildung 16: Informationstechnologie als Hebel für Prozessinnovationen²⁶⁶

Ähnlich auch die Ergebnisse der Studie von *Buxmann* und *König*: Rund 80 Prozent der befragten Entscheidungsträger in Unternehmen sehen für ihre Prozesse die Notwendigkeit eines Redesigns, jedoch wäre in lediglich 55 Prozent der Fälle eine Neustrukturierung auch ohne die Einführung einer Standardsoftware durchgeführt worden. Eine branchenspezifische Betrachtung dieser Fragestellung zeigt, dass diese Ergebnisse tendenziell auf alle Branchen übertragbar sind.²⁶⁷

²⁶⁵ vgl. European Commission (2005), online

²⁶⁶ eigene Darstellung, Daten entnommen aus European Commission (2005), online

²⁶⁷ vgl. Buxmann/König (1995): S. 164

Als Enabler für Business Process Reengineering (BPR)²⁶⁸ können ERP-Systeme deshalb betrachtet werden, weil sie wesentlich zu einer prozessorientierten Sichtweise in Unternehmen beitragen.²⁶⁹ Die Anpassung der Software an unternehmensspezifische Anforderungen ist vielfach Auslöser, die betrieblichen Abläufe zu analysieren und im Rahmen eines BPR neu zu gestalten.²⁷⁰ Der Vergleich des Istzustands mit dem gewünschten Sollzustand und den im Paket vorgesehenen Best-Practice-Möglichkeiten während der Einführung ermöglicht es, Schwachstellen in aktuellen Geschäftsprozessen aufzudecken.²⁷¹

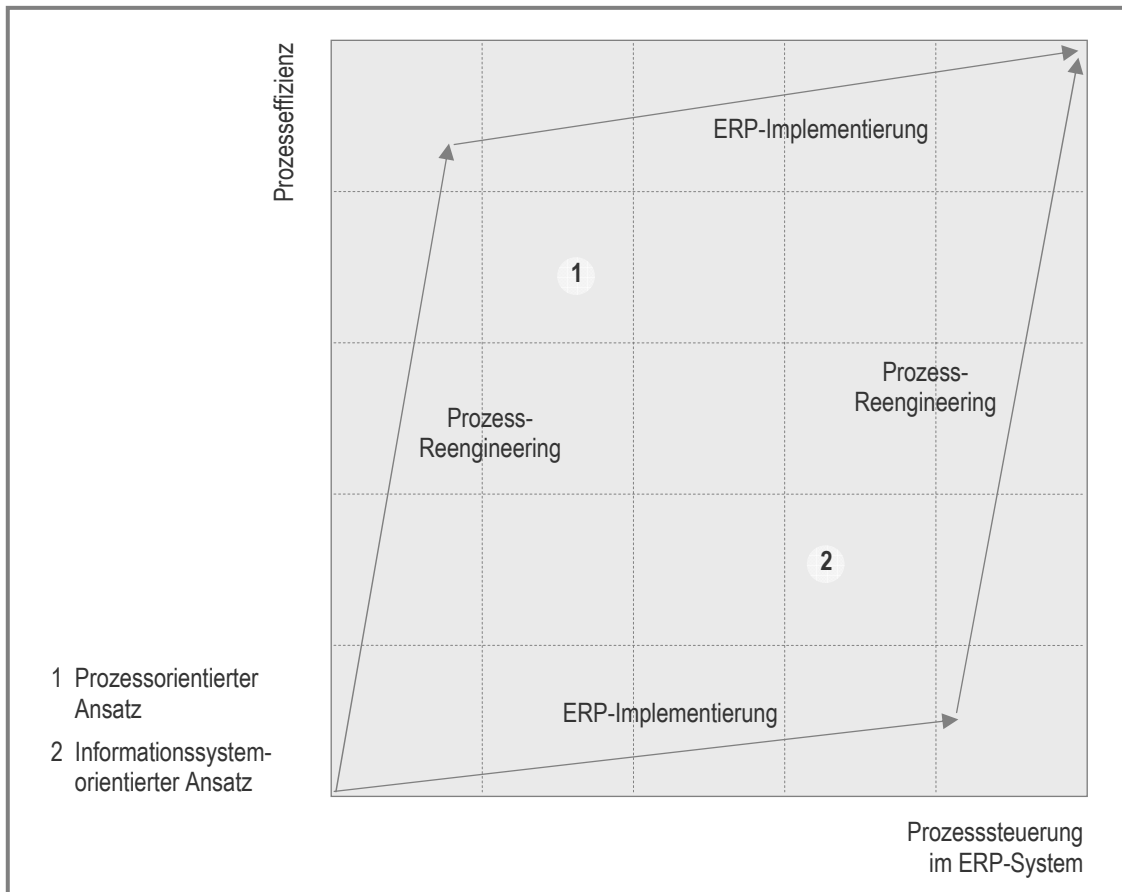


Abbildung 17: Strategien bei der Einführung eines ERP-Systems²⁷²

Dem organisatorischen Wandel im Zusammenhang mit einer sich verändernden betrieblichen IT begegnen Unternehmen auf unterschiedliche Weise. Zwei Strategien sind bei der Einführung von Standardsoftware vorherrschend (siehe Abbildung 17): Die erste Variante beschreibt jenen Prozess, in dem ein Unternehmen zunächst die Prozesse entwirft, diese dann im ERP-System abbildet. Die Prozessorientierung steht hier im Vordergrund. Die zügige Implementierung der Standardsoftware steht im Mittelpunkt der zweiten Variante; die Prozesse werden nicht bereits

²⁶⁸ vgl. Muschter/Österle (1999): S. 444; Buxmann/König (1995): S. 164; Gattiker/Goodhue (2000): S. 7017; European Commission (2005), online

²⁶⁹ vgl. Martin/Mauterer/Gemünden (2002): S. 113

²⁷⁰ vgl. Buxmann/König (1995): S. 163f; Legner/Österle (1999): S. 332

²⁷¹ vgl. Hansen/Neumann (2005): S. 536

²⁷² modifiziert nach Becker/Fleisch/Österle (1997): S. 23

im Vorfeld systematisch neu entworfen. Der informationssystemorientierte Ansatz verfolgt das Motto *Implement and Tune*; das Informationssystem bildet die Basis für eine Reorganisation.²⁷³

Die Prozessorientierung von ERP-Systemen birgt also ein großes Potential für die Steuerung und Optimierung von Geschäftsprozessen.²⁷⁴ Folgende Beispiele veranschaulichen dies:²⁷⁵

- Eine integrierte Datenbasis verhindert redundante und inkonsistente Daten. Aufgaben der Datenübermittlung oder der doppelten Datenerfassung, beispielsweise die Übernahme von Angebotsdaten in einen Kundenauftrag, entfallen somit gänzlich.
- ERP-Systeme erlauben auch die Automatisierung von Aufgaben, etwa eine durch die Warenausgangsbuchung automatisch ausgelöste Rechnungserstellung.
- Die effiziente und effektive Realisierung von organisationsübergreifenden Prozessen wird durch den Zugriff auf eine zentrale Datenbasis ermöglicht, denkt man zum Beispiel an eine Auftragsabwicklung, bei der Verkaufsorganisationen auf Basis aktueller Lagerbestände den Auftrag einem Distributionslager zuweisen und die Lieferung veranlassen.
- Eine Verlagerung und Dezentralisierung der Entscheidungskompetenz wird ebenso durch den Zugriff auf aktuelle Daten erreicht. Ein Auftragsbearbeiter kann dann beispielsweise einem Kunden den Liefertermin der bestellten Ware sofort mitteilen und den Auftrag direkt bestätigen.
- Schließlich treibt die Einführung eines integrierten Systems auch die Standardisierung von Geschäftsprozessen voran, was für global tätige Unternehmen ein attraktiver Vorteil ist.

Was den durch die gemeinsame Datenbasis generierten Nutzen anbelangt, so werden mit dem Einsatz einer Standardsoftware Redundanzen vermieden und unternehmensweite Daten zentral ausgewertet.²⁷⁶ Unternehmen, welche viele verschiedene Systeme aufrechterhalten, sind mit hohen Kosten konfrontiert: Die redundante Datenhaltung und der notwendige Datenaustausch bzw. das Programmieren von Schnittstellen für automatisierte Datentransfers beanspruchen betriebliche Ressourcen in einem nicht unbeträchtlichen Ausmaß. Ist das Vertriebssystem eines Unternehmens nicht an das Buchhaltungssystem geknüpft, gründen Entscheidungen nicht auf einem Verständnis der Profitabilität von einzelnen Produkten bzw. Kunden.²⁷⁷ Oder wie *Davenport* abschließend urteilt: „If a company's systems are fragmented, its business is fragmented.“²⁷⁸

²⁷³ vgl. Muschter/Österle (1999): S. 447

²⁷⁴ vgl. Allweyer (2005): S. 20; Färbinger (2000), online

²⁷⁵ vgl. Legner/Österle (1999): S. 332

²⁷⁶ vgl. Hansen/Neumann (2005): S. 536

²⁷⁷ vgl. Davenport (1998): S. 123

²⁷⁸ Davenport (1998): S. 123

2 Total Cost of Ownership

Unternehmerisches Handeln bedingt das Streben nach einem effizienten Einsatz von betrieblichen Ressourcen. Dazu zählt die in den letzten Jahrzehnten an Bedeutung stetig zunehmende IT-Infrastruktur gleichermaßen. Während die betriebswirtschaftliche Praxis bei Investitionen aller Art darauf bedacht ist, das Kosten-Nutzen-Verhältnis zu messen, zeigt sich, was IT-Investitionen betrifft, ein völlig anderes Bild. Speziell in mittelständischen Unternehmen sind IT-Controlling-Konzepte selten anzutreffen.²⁷⁹ Nur ein kleiner Teil der Unternehmen erhebt die Effizienz von IT-Investitionen. Folglich erweist es sich für ein Unternehmen ohne IT-Kosten-Nutzen-Analyse als schwierig, seine IT-Kosten gezielt einzudämmen.

Das Management der IT-Kosten, welches zumeist in den Händen der CIO's liegt, scheint keine beliebte Aufgabe zu sein: Laut einer Studie von *Price Waterhouse* halten 80 Prozent der IT-Manager TCO-Analysen für wertlos.²⁸⁰ Die logische Konsequenz: 78 Prozent der Unternehmen haben kein TCO-Konzept im Einsatz, so die Ergebnisse einer Studie von *Forrester Research*.²⁸¹ Eine von einer Vermögensmanagementfirma veröffentlichte Studie kommt zum Schluss, dass 43 Prozent der US-amerikanischen Unternehmen eine IT-Vermögensrechnung durchführten; nur 12 Prozent berechneten die Kosten ihrer PC-Arbeitsplätze.²⁸²

Dennoch beschränken sich die meisten Unternehmen auf die Betrachtung der Kostenseite, um ihre Kosten für IT einzuschränken. Die Notwendigkeit der Evaluierung des Nutzens von IT-Investitionen wird noch häufiger negiert.²⁸³ Nur ein Fünftel aller Unternehmen soll laut Schätzungen bewerten, inwieweit die Einführung von IT einen langfristigen unternehmerischen Nutzen bringt. Beispielsweise zeigt eine Studie unter Führungskräften in der Fertigungsindustrie, dass nur 15 Prozent der befragten Unternehmen eine ROI-Berechnung bei IT-Investitionen anstellen.²⁸⁴ Oftmals werden die Nutzenpotentiale von IT-Investitionen intuitiv bestimmt und teilweise auch bewusst überschätzt, um Entscheidungsträger zu überzeugen.²⁸⁵

Warum die Messung der Rentabilität von IT-Investitionen in der unternehmerischen Praxis nicht Fuß fassen konnte, liegt vor allem daran, dass das Ermitteln des Nutzens von IT kein leichter Job ist. Gemeinhin gilt das Identifizieren und Messen von IT-bezogenen Nutzeneffekten als die schwierigste Aufgabe beim Evaluieren von Informationssystemen.²⁸⁶ Speziell angesichts der ständig ansteigenden IT-Ausgaben.²⁸⁷ *Muschter* und *Österle* nennen als Gründe des nachlässigen IT-Nutzenmanagements in der Praxis große Defizite der bestehenden Ansätze zur

²⁷⁹ vgl. Spitta (1998): S. 431

²⁸⁰ vgl. Riepl (1998) nach Muschter/Österle (1999): S. 453

²⁸¹ vgl. Emigh (1999) nach Drury (2001): S. 830

²⁸² vgl. Mulqueen (1997) nach Drury (2001): S. 826

²⁸³ vgl. Dempsey et al. (1998): S. 127; Muschter/Österle (1999): S. 453

²⁸⁴ vgl. Taninecz (1996): S. 45f

²⁸⁵ vgl. Changchit/Joshi/Lederer (1998): S. 158

²⁸⁶ vgl. Seddon/Graeser/Willcocks (2001) nach Lin/Pervan (2003): S. 14

²⁸⁷ vgl. Silk (1990): S. 185

Nutzenbewertung und ein Fehlen von Methoden für ein systematisches Nutzenmanagement; daneben würden Unternehmen auch vielfach zu geringe Erwartungen an IT-Projekte stellen.²⁸⁸

Was die Beschäftigung mit der Kostenseite von IT-Infrastrukturen betrifft, wurden die Unternehmen allerdings im Zuge der starken Verbreitung von Client/Server-Architekturen wachgerüttelt²⁸⁹ - damit einhergehend veränderte sich der Fokus der IT-Kostenanalysen von der Unternehmensebene weg hin zum einzelnen PC-Arbeitsplatz.²⁹⁰ Dadurch wurde den Gesamtkosten eines PC-Arbeitsplatzes, für welche von nun an hauptsächlich der Begriff Total Cost of Ownership (TCO)²⁹¹ verwendet wurde, zunehmende Aufmerksamkeit zuteil. Das Interesse an der Berechnung dieser Gesamtkosten gründete vor allem im ungleichen Verhältnis zwischen den Anschaffungskosten und den Kosten der Betreuung und Wartung von Hard- und Software einer Client/Server-Architektur. Grundsätzlich kann festgestellt werden, dass die Kosten der Betreuung und Wartung deutlich über den Anschaffungskosten liegen.²⁹²

Hat man früher fast ausschließlich die Anschaffungskosten von Hardware und Software als Information für Investitionsentscheidungen herangezogen, so zeigen Untersuchungen, welcher geringer Anteil von den Gesamtkosten eines PC-Arbeitsplatzes auf diese fällt: Der Kaufpreis eines typischen Arbeitsplatzcomputers macht nur 14 bis 15 Prozent der TCO aus. Die restlichen 85 bis 86 Prozent verteilen sich auf unterschiedliche Positionen und werden auch in einer umfassenden Kostenanalyse meist nicht vollständig transparent.²⁹³

Als Beschaffungsmanagement-Tool zielen die TCO darauf, die wahren Kosten eines Produkts bzw. einer Dienstleistung zu ermitteln. Diese umfassen die Kosten der Anschaffung, des Besitzes, der Benutzung sowie der Entsorgung.²⁹⁴ Auf IT-Systeme angewandt, können sie als Modell für die IT-Gesamtkosten verstanden werden, das sämtliche Arbeitsplatzkosten berücksichtigt.²⁹⁵ Darunter fallen alle Kosten über die gesamte Nutzungsdauer einer IT-Komponente, welche in Verbindung mit Anschaffung und Betrieb, einschließlich Wartung und Benutzerbetreuung, sowie Upgrade und Entsorgung stehen.²⁹⁶

Zunächst wird auf den Ursprung der TCO ebenso wie die Beschreibung der Kostenarten, die sich in den meisten TCO-Modellen finden, eingegangen. Schließlich werden der Zweck des TCO-Konzepts aber auch Kritikpunkte daran diskutiert. Warum die TCO zum IT-Controlling einen wertvollen Beitrag leisten können, wird abschließend erörtert.

²⁸⁸ vgl. Muschter/Österle (1999): S. 453f

²⁸⁹ vgl. Wild/Herges (2000): S. 4; Treber/Teipel/Schwickert (2004): S. 11

²⁹⁰ vgl. Drury (2001): S. 826

²⁹¹ Die Thematik der Gesamtkosten prinzipiell ist aber keine revolutionäre. Beschaffungslehrbücher aus dem Jahre 1928 und vielleicht auch schon früher beschäftigen sich bereits mit der Bedeutung der Kosten, die neben dem eigentlichen Verkaufspreis anfallen und die von der Wahl eines bestimmten Lieferanten abhängig sind. Vgl. Ellram (1993): S. 163

²⁹² vgl. Wild/Herges (2000): S. 4

²⁹³ vgl. Conti (2000) nach Gadatsch (2004), online

²⁹⁴ vgl. Ellram (1995): S. 4

²⁹⁵ vgl. Masak (2005): S. 315

²⁹⁶ vgl. Hansen/Neumann (2005): S. 537; Muschter/Österle (1999): S. 452; Drury (2001), online

2.1 Verwendung der TCO als Instrument des IT-Controllings

Das betriebliche IT-Controlling plant, koordiniert und steuert die Informationstechnologie und ihre Aufgaben für die Optimierung der Geschäftsorganisation.²⁹⁷ Warum der Bedarf für IT-Controlling so rasant wächst, liegt auf der Hand: Unternehmen sind mit ständig steigenden IT-Budgets konfrontiert. Gleichzeitig bergen IT-Projekte ein großes unternehmerisches Risiko. Trotz dieses Risikos kann in vielen Unternehmen eine fehlende Transparenz der Kosten für die IT festgestellt werden, wobei sich die indirekten Kosten mangels Erfassung jeglicher Einflussnahme entziehen. Schließlich gibt auch die Vernetzung von unternehmens-übergreifenden Prozessen Anlass für ein betriebliches IT-Controlling-Konzept.²⁹⁸ Tabelle 13 fasst zusammen, warum dem IT-Controlling heute eine wesentliche Bedeutung beigemessen wird.

Tabelle 13: Gründe für ein betriebliches IT-Controlling²⁹⁹

Steigende IT-Budgets	Sinkende Hardware- und teils auch Software-Kosten können ein Ansteigen der IT-Budgets nicht verhindern. Durch ein IT-Controlling-Konzept können mitunter bedeutende Einsparungen erzielt werden.
Hohes Risiko von ERP-Projekten	Die Einführung von Standardsoftware ist mit hohen Risiken verbunden. Aufwändige Schnittstellen werden für die Datenmigration vom alten ins neue System benötigt. Die Wahl der Migrationsstrategie ³⁰⁰ erfordert etwa eine Unterstützung durch IT-Controller.
Fehlende Transparenz der Kosten einer IT-Infrastruktur	Die Kosten für die betriebliche IT werden oft nur unzureichend erfasst. Eine Untersuchung in mittelständischen Unternehmen zeigt, dass nur 17 Prozent ihre IT-Kosten genau kennen, rund der Hälfte sind sie gänzlich unbekannt. ³⁰¹ Ein IT-Controlling im Unternehmen umfasst die Aufgabe der vollständigen Kostenerfassung. Erst die systematische Erfassung der Kosten der betrieblichen IT ermöglicht es Unternehmen, ein Benchmarking durchzuführen. Somit lässt sich die Effizienz des IT-Managements im Vergleich mit anderen Unternehmen beurteilen. ³⁰²
Mangelnde Kontrolle über indirekte Kosten	Traditionelle Konzepte berücksichtigen die indirekten Kosten nicht; das sind jene Kosten, die der IT-Abteilung eines Unternehmens nicht zuzuordnen sind. ³⁰³ In der Folge bleiben die tatsächlichen Kosten einer IT-Infrastruktur verborgen. ³⁰⁴ Werden die indirekten Kosten nicht erfasst, so ist auch keine Einflussnahme des Managements möglich. Der nicht unwesentliche indirekte Kostenblock entzieht sich somit der Kostensteuerung.
Abteilungs- und unternehmens-übergreifende Prozesse	Das betriebliche IT-Controlling kann eine effiziente IT-Unterstützung für den abteilungs- bzw. unternehmensübergreifenden Gesamtprozess sicherstellen. Abteilungsegoismen und kostenintensive IT-Insellösungen können durch ein neutrales IT-Controlling verhindert werden.

²⁹⁷ vgl. Gadatsch/Meyer (2004): S. 53

²⁹⁸ vgl. Gadatsch/Meyer (2004): S. 66ff

²⁹⁹ eigene Darstellung, Daten entnommen aus Gadatsch/Meyer (2004): S. 66ff

³⁰⁰ Zwei verschiedene Einführungsstrategien können im Wesentlichen beobachtet werden: Zum einen existiert der sogenannte Big-Bang, die gleichzeitige Einführung aller gewünschten Module, zum anderen die Step-by-Step Einführung, bei welcher die einzelnen Komponenten hintereinander implementiert werden. Vgl. Jochem (1998): S. 63f

³⁰¹ vgl. o.V. (2000) nach Gadatsch/Meyer (2004): S. 67f

³⁰² vgl. Hurwicz (1998) nach Wild/Herges (2000): S. 6

³⁰³ vgl. Wolf/Holm (1998) nach Wild/Herges (2000): S. 6

³⁰⁴ vgl. Jansen (1998) nach Wild/Herges (2000), online

Zur Beurteilung von IT-Investitionsentscheidungen strebt das IT-Controlling nach der Bereitstellung von ganzheitlichen Kosteninformationen.³⁰⁵ Die vollständige Erfassung der Kosten im Rahmen eines TCO-Konzepts bietet eine sinnvolle Grundlage für das Kostenmanagement von IT-Komponenten.³⁰⁶ Somit erfüllt ein TCO-Ansatz die Funktion eines IT-Controlling-Instruments, er dient der Planung, Überwachung und Analyse von IT-Budgets und IT-Kosten.³⁰⁷ Die TCO können beispielsweise im IT-Controlling als Kennzahl für die finanzielle Perspektive eines IT-Kennzahlensystems eingesetzt werden.³⁰⁸

Die Berechnung der Gesamtkosten einer IT-Infrastruktur soll helfen, knapp bemessene Budgets effizient einzuteilen.³⁰⁹ Indem ein TCO-Konzept versucht, transparente IT-Kostenstrukturen zu schaffen, werden für IT-Controller Zusammenhänge und kostentreibende Faktoren sichtbar.³¹⁰ Hält man sich vor Augen, dass der größte Zuwachs der IT-Kosten außerhalb der IT-Budgets stattfindet - die indirekten Kosten haben sich im letzten Jahrzehnt verdreifacht -, so sind transparente Kostenberechnungen als Entscheidungsgrundlage für Investitionen nicht unerheblich.³¹¹

2.2 Ursprung der TCO

1987 weckte die *Gartner Group* großes öffentliches Interesse, als eine Studie über die Kosten eines PC-Arbeitsplatzes³¹² veröffentlicht wurde, in deren Rahmen neben den Anschaffungskosten eines Rechners auch diverse Kosten der Nutzung ermittelt wurden. Bis dahin wurden die Kosten einer IT-Infrastruktur nur mit dem jeweiligen finanziellen Anschaffungsaufwand kostenrechnerisch berücksichtigt. Die im laufenden Betrieb anfallenden Kosten wurden nicht beachtet.³¹³ Überraschenderweise stellte das IT-Consulting-Unternehmen damals fest, dass die Anschaffungskosten nur ein Fünftel der Gesamtkosten eines PCs betragen; vier Fünftel ergeben sich aus Löhnen und Gehältern im Lebenszyklus der IT-Vermögensgegenstände.³¹⁴ Als Konsequenz empfahl die *Gartner Group*, das besondere Augenmerk auf die Reduzierung der Personalkosten zu richten. Das bedeutete eine Abkehr von der üblichen Praxis, deren Fokus auf der Begrenzung der IT-Anschaffungskosten lag.³¹⁵

In der Folge entwickelte die *Gartner Group* ein Konzept zur Erfassung und Analyse der Gesamtkosten eines PC-Arbeitsplatzes, welche schließlich Total Cost of Ownership (TCO) benannt wurden und mit diesem Namen auch in die betriebswirtschaftliche Literatur ein-

³⁰⁵ vgl. Gadatsch/Meyer (2004): S. 92

³⁰⁶ vgl. Dempsey et al. (1998): S. 129

³⁰⁷ vgl. Treber/Teipel/Schwickert (2004): S. 40

³⁰⁸ vgl. Gadatsch (2006): S. 369; Zillich (2002), online

³⁰⁹ vgl. Ullerich (2004): S. 3

³¹⁰ vgl. Zillich (2002), online

³¹¹ vgl. Anzer Business Systems (o.J.), online

³¹² Während die *Gartner Group* anfänglich lediglich Arbeitsplatzrechner fokussierte, stellen heute auch Handhelds, Notebooks, Mainframes, Verteilte Systeme, Local Area Networks sowie Telekommunikationseinrichtungen Bezugsobjekte für TCO-Analysen dar. Vgl. Wild/Herges (2000): S. 3

³¹³ vgl. Wild/Herges (2000): S. 3

³¹⁴ vgl. Treber/Teipel/Schwickert (2004): S. 13; Bachmann (1998) nach Grob/Lahme (2004): S. 1

³¹⁵ vgl. Treber/Teipel/Schwickert (2004): S. 13f

gingen.³¹⁶ 1997 kam es mit Unterstützung von zwölf der führenden IT-Unternehmen -darunter Microsoft, IBM und Compaq - zur Vorstellung der neuen TCO-Methodik.³¹⁷

Bis dem TCO-Begriff breitere Aufmerksamkeit zuteil wurde, dauerte es allerdings bis Mitte der 1990er Jahre.³¹⁸ Angesichts des wachsenden Kostendrucks erlebte damals der Einsatz von TCO-Berechnungen eine Renaissance. Zu dieser Zeit etwa begannen IT-Unternehmen, Informationen zu den Gesamtkosten ihrer Produkte für Marketingzwecke einzusetzen.³¹⁹

Neben der *Gartner Group* haben auch viele andere bedeutende Unternehmen aus dem IT-Sektor (u.a. *META Group*, *Forrester Research*, *Fraunhofer-IAO*) Modelle zur Erfassung und Analyse der Kosten einer IT-Infrastruktur gebildet.³²⁰ Bislang konnte kein Konsens darüber erzielt werden, welchen Kosteneinflussgrößen Beachtung geschenkt werden soll, beziehungsweise wie diese zu den TCO verdichtet werden sollen. Somit entstanden mehr oder minder voneinander abweichende Kostenmodelle. Eine eindeutig definierte Kennzahl existiert nicht.³²¹

2.3 Beschreibung der TCO

Die Berechnung der TCO dient dazu, alle anfallenden Kosten von IT-Investitionsgütern abzuschätzen, indem nicht nur die Anschaffungskosten der Hardware und Software berücksichtigt, sondern auch alle Kostenaspekte der späteren Benutzung wie beispielsweise Schulungs-, Support- und Wartungskosten mit einbezogen werden. Unternehmen ist es somit möglich, bereits vor einer IT-Investition den gesamten Investitionsaufwand zu identifizieren.³²²

Im Gegensatz zu anderen betriebswirtschaftlichen Kennzahlen gibt es für die Berechnung der TCO keine verbindliche Formel. Es existieren viele unterschiedliche Konzepte, welche IT-Consulting-Unternehmen kreiert haben und welche ausschließlich deren Kunden zugänglich sind. Die der breiten Öffentlichkeit zugängliche Literatur zum Thema IT-Controlling - oder im speziellen zu TCO-Analysen - gibt nur einzelne Modellaspekte wieder, die in den gebräuchlichsten TCO-Konzepten ihren Niederschlag finden.³²³

Ein Vergleich der diversen Modelle lässt allerdings rasch erkennen, dass sich diese vornehmlich in strukturellen Details unterscheiden. Als Quintessenz kann man gemeinsame Basisfaktoren identifizieren (siehe Abbildung 18).³²⁴

³¹⁶ vgl. Wild/Herges (2000): S. 3

³¹⁷ vgl. Gartner Group (1997), online; Treber/Teipel/Schwickert (2004): S. 20

³¹⁸ vgl. Treber/Teipel/Schwickert (2004): S. 14

³¹⁹ Die *Deloitte & Touche Consulting Group* etwa erhielt von *Digital Equipment Corporation* und *Microsoft* den Auftrag, eine Studie über die TCO von Workstations zu erarbeiten. Der Einfluss der Betriebssysteme Windows NT und UNIX auf die Gesamtkosten im Lebenszyklus einer Workstation sollte ermittelt werden. Das Ergebnis: UNIX-Rechner verursachen im Vergleich zu Windows NT-Rechnern relativ höhere Wartungskosten. *Microsoft* und *Intel* propagierten in der Folge *Zero Administration for Windows (ZAW)*. Vgl. Grob/Lahme (2004): S. 1; Siering (1997): S. 20. Ein weiteres Beispiel für TCO im Marketing: Das Kostensenkungspotential von *Windows NT 5.2 (Windows Server 2003)* wurde mit dem österreichischen *Bundesministerium für Inneres* als Referenzkunden beworben. Vgl. Microsoft (2003), online. Siehe auch Ullerich (2004): S. 4; Hoffman (2002): S. 52

³²⁰ vgl. Wild/Herges (2000): S. 5

³²¹ vgl. Wild/Herges (2000): S. 5; Grob/Lahme (2004): S. 1

³²² vgl. Drury (2001), online

³²³ vgl. Wild/Herges (2000): S. 4f

³²⁴ vgl. Riepl (1998) nach Wild/Herges (2000): S. 8

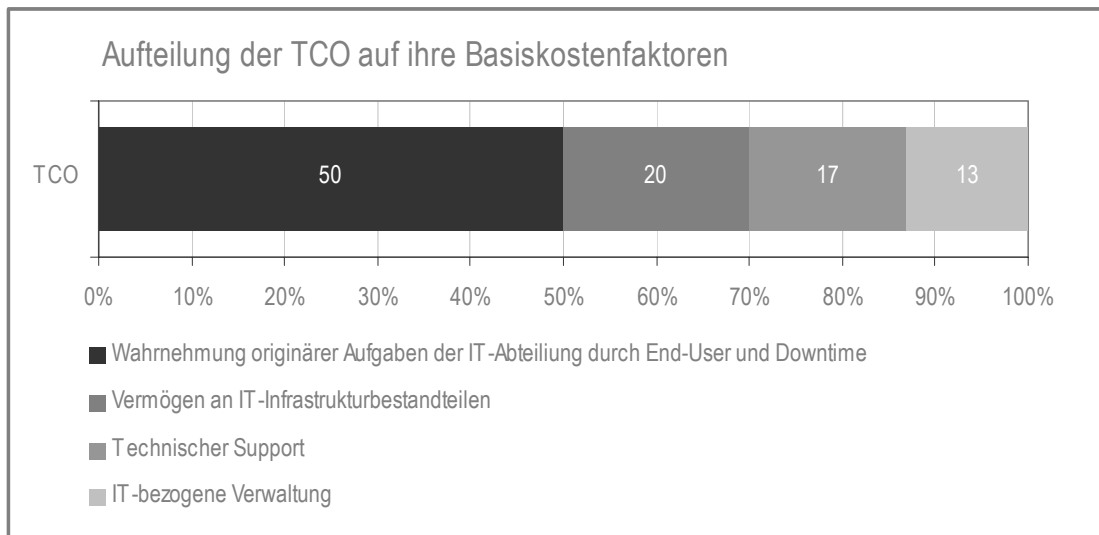


Abbildung 18: Aufteilung der Total Cost of Ownership (TCO) auf ihre Basiskostenfaktoren³²⁵

Rund die Hälfte der Gesamtkosten einer IT-Infrastruktur fällt auf Kosten, die in IT-Budgets typischerweise nicht auftauchen: Diese Kosten entstehen einerseits aufgrund der Wahrnehmung von Aufgaben, welche von der IT-Abteilung unzureichend erfüllt wurden, durch die End-User und andererseits aufgrund von Downtime (dt. Ausfallzeiten). Zu diesen originären Aufgaben der IT-Abteilung, welche nun die Endanwender übernehmen, sind vor allem Peer-to-Peer-Support - als Ausgleich zum mangelnden End-User-Training - und die eigene Datenverwaltung zu zählen. Mit der Downtime wird jene Zeit beschrieben, in der Teile einer IT-Infrastruktur nicht benutzbar sind. Der zweitgrößte Kostenblock mit einem Fünftel der TCO ist dem Vermögen an den verschiedenen Bestandteilen einer IT-Infrastruktur zuzurechnen. Der technische Support, beispielsweise die Systemadministration, verursacht 17 Prozent der Gesamtkosten einer IT-Infrastruktur, die IT-bezogene Verwaltung - die Verwaltung der IT-Abteilung und die Organisation von End-User-Training etwa - 13 Prozent.³²⁶

Zentrale Grundlage für das Verständnis der TCO bildet die Unterscheidung zwischen direkten und indirekten Kosten. IT-Arbeitsplätze verursachen neben den direkten - für die Verantwortlichen transparenten - Kosten beachtliche indirekte Kosten, die sich der Beeinflussung des Managements entziehen, weil sie im betrieblichen Rechnungswesen nicht sichtbar IT-bezogene Kosten darstellen.³²⁷

Alle Aufwendungen der IT-Abteilung, die ihr durch die Bereitstellung ihrer Leistungen gegenüber dem Unternehmen entstehen, werden als direkte Kosten bezeichnet.³²⁸ Diese beinhalten sämtliche Investitionen, Gebühren und Arbeitskosten, welche von der IT-Abteilung aufgewendet werden, um für das Unternehmen und seine Mitarbeiter die IT-Systeme anzuschaffen, bereitzu-

³²⁵ eigene Darstellung, Daten entnommen aus Riepl (1998) nach Wild/Herges (2000): S. 9; Wolf/Holm (1998) nach Wild/Herges (2000): S. 9

³²⁶ vgl. Riepl (1998) nach Wild/Herges (2000): S. 8

³²⁷ vgl. Gadatsch (2004), online

³²⁸ vgl. Wild/Herges (2000): S. 10

stellen und zu betreiben. Dazu zählen insbesondere Ausgaben für Hardware und Software, Kosten für Mitarbeiter der IT-Abteilung und allfällige Gebühren für ausgelagerte Dienste. Die Berechnung der direkten Kosten in den TCO-Modellen umfasst alle direkten Kosten, welche mit den Clients, den Servern, den Peripheriegeräten und dem Netzwerk in Verbindung stehen.³²⁹ Die Höhe der direkten Kosten lässt sich relativ genau bestimmen; sie sind in Form von Belegen, beispielsweise Rechnungen oder Lohn- und Gehaltslisten, dokumentiert und scheinen im Rechnungswesen typischerweise als IT-bezogene Kosten auf.³³⁰

Indirekte Kosten scheinen dagegen im betrieblichen Rechnungswesen nicht als IT-bezogene Kosten auf. Sie werden durch die direkten Kosten beeinflusst, oder wie es *Gartner Group* ausdrückt: „[They] measure the efficiency of IS in delivering expected services to end users.“³³¹ *Wild* und *Herges* beschreiben sie als den „Wertverzehr [...], der aus effizienzhemmenden Vorgängen im Rahmen der Nutzung einer IT-Infrastruktur erfolgt.“³³² Dieser Wertverzehr entsteht durch versteckte Benutzerkosten: Die Anwender beschäftigen sich mit der Infrastruktur statt mit ihrem Job, beispielsweise durch Arbeitszeit, während der Endanwender mit der Durchführung von Self- oder Peer-to-Peer-Support infolge von Benutzungsproblemen beschäftigt sind, Schulungen besuchen oder aufgrund von Systemausfällen (Downtime) die IT-Infrastruktur nicht nutzen können. In allen drei genannten Beispielen sind die Endanwender also für einen gewissen Zeitraum unproduktiv.³³³ Solche Zeiten drücken indirekt eine fehlende Effizienz der IT-Abteilung und ihrer entsprechenden IT-Leistungen aus, die für die Erbringung dieser Leistungen zuständig ist. Eine geringe Notwendigkeit für Self- und Peer-to-Peer-Support kann also als Indiz für ein effizientes Arbeiten der IT-Abteilung gelten.³³⁴ Die indirekten Kosten werden deshalb als Effekt zweiter Ordnung der (mangelnden) IT-Ausgaben betrachtet (siehe dazu Abbildung 19).³³⁵

Dieser Effekt kann zwar nicht direkt gemessen werden und es besteht nicht immer eine kausale Beziehung, aber man weiß, dass ein effizientes IT-Kostenmanagement einen direkten positiven Effekt auf die Endbenutzerproduktivität hat und dass Produktivitätsverluste infolge von Kürzungen in einem Unternehmen schlussendlich hohe - die Einsparungen weitaus übersteigende - Kosten verursachen können. An dieser Stelle sei nochmals auf Abbildung 18 verwiesen, in welcher der immense Anteil der indirekten Kosten - nämlich 50 Prozent - an den Gesamtkosten eines PC-Arbeitsplatzes bzw. einer IT-Infrastruktur dargestellt ist.³³⁶

³²⁹ vgl. Gartner Group (2003), online

³³⁰ vgl. Wild/Herges (2000): S. 11; Gartner Group (2003), online

³³¹ Gartner Group (2003), online

³³² Wild/Herges (2000): S. 11

³³³ vgl. Wild/Herges (2000): S. 11; Gadatsch (2004), online; Gartner Group (2003), online

³³⁴ vgl. Wild/Herges (2000): S. 11

³³⁵ vgl. Gartner Group (2003), online

³³⁶ vgl. Gartner Group (2003), online

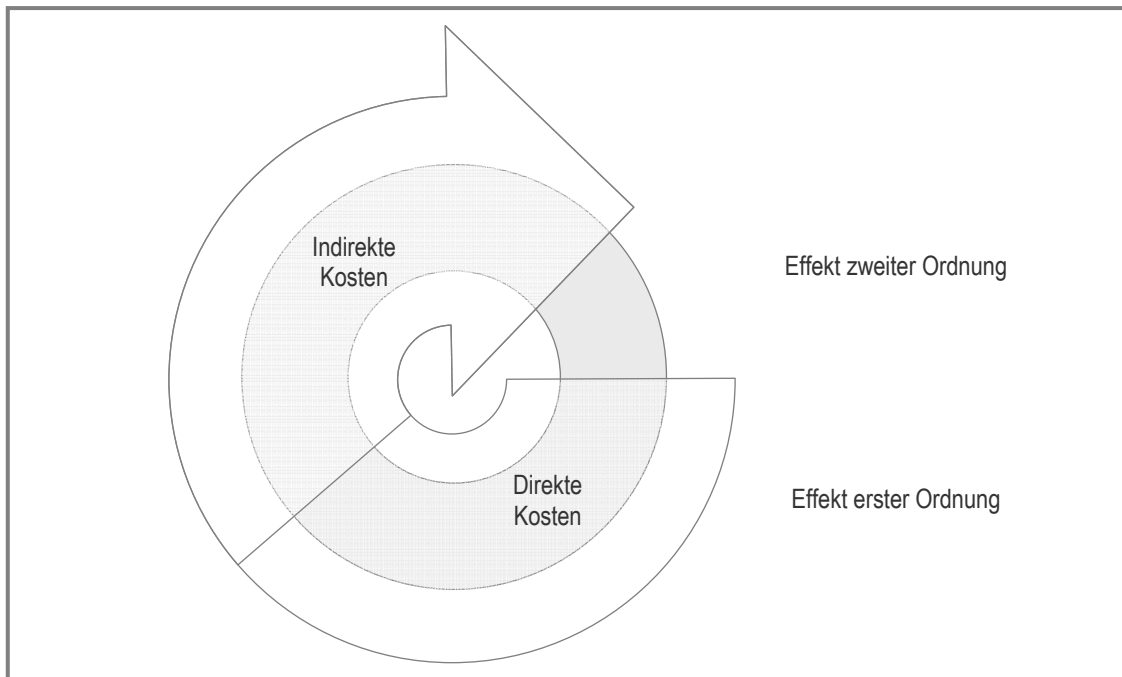


Abbildung 19: Direkte und indirekte Kosten³³⁷

Kostenmodelle werden benutzt, um die Gesamtkosten strukturiert und einheitlich erfassen und bewerten zu können.³³⁸ Im Folgenden wird der Aufbau des Kostenmodells der *Gartner Group* näher beschrieben. Die Wahl fiel nicht zuletzt deshalb auf dieses Modell, weil die *Gartner Group* Urheber dieser Methodik ist. Zudem finden sich zu anderen TCO-Modellen - beispielsweise von *Forrester Research* oder *META Group* - nur wenige Ausführungen in der Literatur.³³⁹ Bei der TCO-Berechnung werden bei den direkten wie bei den indirekten Kosten jeweils die jährlichen Kosten veranschlagt.³⁴⁰ Sich durch die Anschaffung einer beliebigen IT-Komponente ergebende Vorteile können kostensenkend bewertet werden.³⁴¹

Unter den direkten Kosten finden sich zunächst Ausgaben für Hardware und Software, von welchen jene Hardware- und Software-Ausgaben abgegrenzt sind, die IT-Prozesse unterstützen, nicht aber Geschäftsprozesse.³⁴² Weiters enthält das TCO-Modell der *Gartner Group* sämtliche Aufwendungen, die aus der Vergütung der internen und externen Mitarbeiter herühren, die unmittelbar den Betrieb der jeweiligen IT-Infrastruktur gewährleisten. Außerdem werden in den direkten Kosten alle Aufwendungen berücksichtigt, die mit der Organisation und Verwaltung einer IT-Abteilung entstehen.³⁴³ Tabelle 14 gibt eine grobe Einteilung der direkten Kosten wieder.

³³⁷ modifiziert nach Gartner Group (2003), online

³³⁸ vgl. Ullerich (2004): S. 7

³³⁹ vgl. Wild/Herges (2000): S. 16f

³⁴⁰ vgl. Gartner Group (2003), online

³⁴¹ vgl. Ullerich (2004): S. 7

³⁴² Die den IT-Prozessen zugerechneten Hardware- und Softwarekosten finden sich in Tabelle 9 als IS Hardware/Software. Vgl. Gartner Group (2003), online; Wild/Herges (2000): S. 12

³⁴³ vgl. Wild/Herges (2000): S. 12f

Tabelle 14: Total Cost of Ownership (TCO) - Direkte Kosten³⁴⁴

Hardware	Anschaffung bzw. Leasing	Anschaffungskosten für Server, Clients, Peripheriegeräte und Netzwerk-Hardware
	Upgrades	Kosten für Upgrades für Clients, Server, Peripheriegeräte und Netzwerk-Hardware
	Ersatzteile	Kosten für Laufwerke, Rechner, Motherboards etc.
	Zubehör	Kosten für Datensicherungsmedien, CD-ROMs, Toner etc.
Software	Personal Productivity Software	Kosten für Personal Productivity Software (z.B. Microsoft Office)
	Business Software	Kosten für ERP-Systeme bzw. für eigenständige Business Software (z.B. für Material-, Absatzwirtschaft, Finanzwesen)
	Datenbank- und Datenbankmanagement-Software	Kosten für Datenbank-Applikationen auf Server und Clients, Business Intelligence Tools und Report Writers
	Messaging und Groupware	Kosten für Messaging- und Groupware-Applikationen
IS Hardware	Siehe dazu Hardware	Sämtliche Kosten für Hardware, welche hauptsächlich IT-Prozesse unterstützt
IS Software	Netzwerk-, System- und Datensicherungsmanagement	Kosten für sämtliche Software, die für das Management des Distributed Computing Environment eingesetzt wird
	Service Desk Management	Kosten für Service Desk Software, um Endbenutzer-Anfragen bearbeiten zu können
	Training Software	Kosten für Computer Based Training Software
Operations	Technischer Service	Kosten für Personal und ausgelagerte Leistungen im Zusammenhang mit technischen Service-Aktivitäten für Clients, Server und Netzwerke (z.B. Benutzerverwaltung, Hardwarekonfiguration, Planung der Speicherkapazitäten, Datensicherung)
	IT-Planung	Kosten für Beschaffungsplanung, Aktivitäten im Zusammenhang mit Systems Research, Sicherheit und Virenschutz sowie Datensicherungsmanagement
	Datenbankmanagement	Kosten für alle datenbankspezifischen Tätigkeiten (ausgenommen der Erstellung von Datenbanken)
	Service Desk	Personalkosten für Service Desk Aufgaben
Verwaltung	Finanzielle und verwaltungstechnische Aufgaben	Kosten für das Management der IT-Abteilung (z.B. CIO's, MIS Directors)
	IT-bezogene Schulungen für die IT-Abteilung	Kosten für die Ausbildung der Mitarbeiter der IT-Abteilung
	IT-bezogenes End-User-Training	Kosten für die Planung und Durchführung der Schulungen der Endbenutzer

³⁴⁴ eigene Darstellung, Daten entnommen aus Gartner Group (2003), online; Wild/Herges (2000): S. 12f

Tabelle 15: Total Cost of Ownership (TCO) - Indirekte Kosten³⁴⁵

End-User Operations	Self- und Peer-to-Peer-Support	Kosten der für Selbsthilfe beziehungsweise informelle Hilfe von Kollege zu Kollege aufgewendeten Arbeitszeit ³⁴⁶
	Casual Learning	Kosten der für das bewusste und eigenständige Lernen verwendeten Arbeitszeit (z.B. Lesen von Instruktionen, der Online-Hilfe oder für das Probieren am System)
	Formales End-User Training	Kosten der für den Besuch von IT-Schulungsmaßnahmen aufgewendeten Arbeitszeit
	Datenverwaltung	Kosten der für Datenmanagement eingesetzten Arbeitszeit (z.B. Erstellen von Backups, Organisation der Datenverwaltung)
	Entwicklung von Software	Kosten für die Arbeitszeit, welche für die Entwicklung von Software, i.S. einer Anpassung verfügbarer Software an eigene Bedürfnisse, aufgewendet wird (z.B. Makroprogrammierung, Entwicklung von Datenbankanwendungen)
	Futzung	Kosten der Arbeitszeit, in welcher End-User die ihnen zur Verfügung gestellten IT-Infrastrukturbestandteile für private Zwecke nutzen (z.B. privates Web-Surfen, Computer-Spielen)
Ausfallzeiten	Geplante und ungeplante Ausfallzeiten	Kosten der Produktivitätsminderungen der Endbenutzer aufgrund der Antwortzeit der Mitarbeiter am Service Desk bei einem Benutzerproblem oder aufgrund des Ausfalls von Netzwerkressourcen (Clients, Server, Drucker, Applikationen, Kommunikationssysteme und Verbindungen), hervorgerufen durch geplante Wartungstätigkeiten oder durch unerwartete Ausfälle

Die Ursache für die Entstehung der indirekten Kosten ist die mangelnde Unterstützung der Benutzer durch die IT-Abteilung, welche einen Arbeitszeitverlust auf der Seite der Endbenutzer entstehen lässt. Die indirekten Kosten schließen die Wahrnehmung originärer Aufgaben einer IT-Abteilung durch die End-User, den Besuch von IT-bezogenen Schulungen durch End-User und den Ausfall von Teilen der IT-Infrastruktur ein, wobei interessant ist, dass alleine die Kosten des Casual Learning (dt. nicht-formales Lernen) die des formalen Trainings typischerweise übersteigen.³⁴⁷ Zur Berechnung der indirekten Kosten werden die jeweiligen Produktivitätsverluste der End-User mit den entsprechenden Lohn- und Gehaltskosten gewichtet.³⁴⁸ Tabelle 15 verschafft einen Überblick über die indirekten Kosten.

Indirekte Kosten zu quantifizieren, fällt Unternehmen schwer, so dass die allermeisten darauf verzichten und sich auf die Steuerung der direkten Kosten beschränken.³⁴⁹ Doch beim Versuch,

³⁴⁵ eigene Darstellung, Daten entnommen aus Gartner Group (2003), online; Wild/Herges (2000): S. 13f

³⁴⁶ Typische vom Endanwender durchgeführte Aufgaben sind etwa Installationen, Reparaturen und Support. Vgl. Treber/Teipel/Schwickert (2004): S. 28

³⁴⁷ vgl. Gartner Group (2003), online; Wild/Herges (2000): S. 13

³⁴⁸ vgl. Wild/Herges (2000): S. 13

³⁴⁹ vgl. Wild/Herges (2000): S. 11

die direkten Kosten zu senken, werden nicht selten die indirekten angehoben. Laut *Gartner Group* kann eine Einsparung von einer Kosteneinheit auf der Seite der direkten Kosten zu einem Anstieg der indirekten Kosten um bis zu vier Einheiten führen. Das IT-Consulting-Unternehmen rät eine Abschätzung der Höhe der indirekten Kosten durchzuführen und nennt eine Reihe von Kennzahlen, die sie für geeignet halten, um die indirekten Kosten hinreichend genau zu bestimmen. Wie oft pro Monat Unterstützung außerhalb des formalen Trainings gesucht wird und welcher Tätigkeit ein auf Hilfe wartender Mitarbeiter am häufigsten nachgeht, sind Beispiele aus dieser Kennzahlensammlung.³⁵⁰

2.4 Bewertung des TCO-Konzepts

Das grundlegende Motiv sämtlicher TCO-Modelle, die tatsächlichen Kostenstrukturen einer IT-Infrastruktur durch die Messung der direkten und indirekten Kosten transparent zu machen, bringt für Entscheidungsträger in Unternehmen einen entscheidenden Vorteil: Es wird ein Anstoß für mehr Kostenbewusstsein gegeben; die Aufmerksamkeit, was nicht-budgetierbare Kosten anbelangt, wird geweckt. Die ganzheitliche Erfassung der Kosten eines IT-Vermögensgegenstands über die gesamte wirtschaftliche Nutzungsdauer hinweg ist ein wichtiger Schritt zu einem effizienteren und damit betriebswirtschaftlicheren Umgang mit IT-Komponenten.³⁵¹ Deshalb sollte die TCO-Berechnung eine Standardaufgabe vor der Kaufentscheidung einer IT-Komponente darstellen.³⁵² Vorteilhaft für Unternehmen ist außerdem, dass relativ viele Vergleichszahlen ebenso wie eine Menge an detailliert beschriebenen Berechnungsmodellen verfügbar sind. Ansätze im Bereich von Wirtschaftlichkeitsanalysen, welche auch die Erträge zu quantifizieren versuchen, existieren bedeutend weniger.³⁵³

Ein besonderes Problem der TCO-Thematik ist ihre Intransparenz, welche sich im Fehlen von fundierten Arbeiten ausdrückt. Es sind die Kunden der jeweiligen IT-Consulter, denen ein breiter Zugang zu theoretischen Auseinandersetzungen und Studien vorbehalten bleibt. Zusätzlich beschäftigen sich viele praxisorientierte Artikel mit Studien, die von IT-Anbietern finanziert werden und ihnen in erster Linie für das Marketing der eigenen Produkte dienen.³⁵⁴

Grundsätzlich betonen Gesamtkostenmodelle den Wertverzehr durch die IT-Infrastruktur; sie konzentrieren sich ausschließlich auf die Kosten und vernachlässigen den Nutzen.³⁵⁵ Ein TCO-Vergleich könnte nur dann zu rationalen Investitionsentscheidungen führen, wenn das Nutzen-niveau für alle Alternativen tatsächlich gleich hoch wäre.³⁵⁶ TCO-Berechnungen eignen sich also nicht für Kaufentscheidungen. Eine Umfrage, die ergab, dass 30 Prozent der befragten Unternehmen TCO als einzige Kennzahl bei Investitionsentscheidungen zugrunde legen, veranlasste einen Analysten der *Gartner Group* zu der rhetorischen Frage, ob sie mit dem TCO-

³⁵⁰ vgl. Gartner Group (2003), online

³⁵¹ vgl. Treber/Teipel/Schwickert (2004): S. 39f

³⁵² vgl. o.V. (2005): S. 14; Sontow (2006a): S. 43

³⁵³ vgl. Muschter/Österle (1999): S. 452; Changchit/Joshi/Lederer (1998): S. 146f

³⁵⁴ vgl. Treber/Teipel/Schwickert (2004): S. 40f

³⁵⁵ vgl. Zillich (2002), online; Gadatsch (2004), online

³⁵⁶ vgl. Hoffman (2002): S. 52; Grob/Lahme (2004): S. 2

Modell ein Monster geschaffen hätten.³⁵⁷ Aus demselben Grund sei auch beim TCO-Vergleich mit anderen Unternehmen Vorsicht angebracht; die Option mit den niedrigsten TCO ist nicht immer die beste Variante.³⁵⁸ Es empfiehlt sich vielmehr, eine ROI-Analyse durchzuführen, um den Wertzufluss, sprich den Nutzen, abzubilden und anschließend die Rentabilität einer IT-Investition zu beurteilen. Die TCO-Berechnung bietet aber eine geeignete Grundlage für die Kostenseite des ROI. Um einen realitätsnahen ROI zu erhalten, ist es besonders wichtig, einen möglichst genauen Wert der Investition zu ermitteln. Wertvolle Aussagen können nur jene Unternehmen treffen, welche über ihre komplexen IT-Kostenstrukturen hinlänglich informiert sind.³⁵⁹ Während die *Gartner Group* noch 1998 den Anspruch an ihr TCO-Tool formulierte, ein Management-Instrument bereitzustellen, welches einen Regelkreis von TCO-Analyse, TCO-Verbesserung und TCO-Management durchläuft, stellte das IT-Consulting-Unternehmen in der Zwischenzeit deutlich, dass die Kostenbetrachtung der IT-Infrastruktur als Effizienzmaßstab unbedingt durch qualitative und quantitative Effektivitätsmaßstäbe ergänzt werden muss, um Entscheidungen über IT-Investitionen gerecht zu werden.³⁶⁰

Neben dem Mangel hinsichtlich der Nutzenbetrachtung einer IT-Infrastruktur widerstrebt die TCO-Systematik auch einer prozessorientierten Sicht der IT. Die IT-Komponenten bilden die Kostenträger, nicht aber die Prozesse. Wesentlich für die prozessorientierte Sicht wäre die Frage, wie durch den IT-Einsatz Geschäftsprozesse verbessert bzw. erst ermöglicht werden können.³⁶¹ *Liebmann* brachte es in einem 1999 erschienenen Artikel auf den Punkt:

„In fact, the real question facing any money-conscious IT-organization today is not the cost of installing and maintaining the individual elements of its infrastructure. What really matters is the cost of provisioning a service. [...] Companies don't buy PCs or routers for their own sakes. They buy them as part of a plan to implement business services that will make money.“³⁶²

Die TCO haben, indem sie sämtliche Anschaffungs- und Nutzenkosten vereinen, den Anspruch, ein langlebiges Modell zu sein. Eine langfristige Betrachtung soll jedoch nicht Kosten in den Mittelpunkt rücken. Auszahlungen sind langfristig relevant, weil die Veränderung der Kapitalbindung zur Bestimmung der Zinszahlung führt und Zinszahlungen im Allgemeinen ein relativ hohes Gewicht aufweisen. Das TCO-Modell der *Gartner Group* addiert aber die zu unterschiedlichen Zeitpunkten anfallenden Kosten einfach auf; Zinseffekte werden vernachlässigt. Außerdem fehlt zumeist der Ansatz von Ertragsteuern. Die von den Zahlungen abhängigen Steuern dürfen allerdings nicht unberücksichtigt bleiben, weil sonst das Prinzip der Vollständigkeit der Zurechnung von Zahlungen verletzt würde. Die TCO-Berechnung soll nach Ansicht von

³⁵⁷ vgl. Frick (1999) nach Treber/Teipel/Schwickert (2004): S. 42

³⁵⁸ vgl. Gartenberg (2000): S. 52; Hoffman (2002): S. 52; Treber/Teipel/Schwickert (2004): S. 42

³⁵⁹ vgl. Zillich (2002), online

³⁶⁰ vgl. Treber/Teipel/Schwickert (2004): S. 22f

³⁶¹ vgl. Treber/Teipel/Schwickert (2004): S. 45. Zu *IBM* und seiner prozessorientierten Bewertung von IT-Investitionen siehe Taninecz (1996): S. 46

³⁶² Liebmann (1999): S. 92

Grob und *Lahme* auch Opportunitätskosten für das anderweitig investierbare Eigenkapital sowie für den Arbeitseinsatz umfassen.³⁶³

Die Berechnung bzw. Abschätzung der indirekten Kosten gestaltet sich in der Praxis schwierig. Oft sind in der Folge Kosten wie beispielsweise die Ausbildung neuer Mitarbeiter oder der Peer-to-Peer-Support nicht umfassend enthalten.³⁶⁴ Dies trifft insbesondere auf nicht triviale IT-Komponenten zu; TCO-Analysen werden mit zunehmender Komplexität einer IT-Lösung (Update von Windows 98 auf Windows XP versus Einführung einer komplexen CRM-Lösung) schwieriger. Nicht nur die Anschaffungskosten fallen höher aus; es ergeben sich auch versteckte Kosten, die nur mittels einer genauen Analyse der TCO erhoben werden können. Zusätzlich ist die Dauer des Projekts bei komplexen IT-Lösungen viel schwerer planbar.³⁶⁵

Die TCO-Berechnung, aber auch die ROI-Analyse, stoßen bei Investitionsentscheidungen immer wieder an ihre Grenzen. Häufig sind IT-Investitionsentscheidungen in der Praxis nicht das Ergebnis von Wirtschaftlichkeitsanalysen. Als Beispiel sei der Druck, die IT-Systeme, was die technologische Entwicklung betrifft, jenen der Konkurrenz anzupassen, genannt.³⁶⁶ Eine Fixierung auf rein finanzielle Größen als Entscheidungsgrundlage bei IT-Investitionen wird aber ohnehin kritisch betrachtet. Es ist bekannt, dass der Erfolg von IT nicht ausschließlich von der Technik abhängt. Erst wenn die Endanwender die Technologie nutzenbringend einsetzen, wird das Unternehmen davon profitieren: Der User entscheidet über Erfolg oder Misserfolg.³⁶⁷

Abschließend sei noch erwähnt, dass eine Gesamtkostenberechnung niemals den Anspruch auf Objektivität stellen kann. Die monetäre Bewertung - im Speziellen der indirekten Kosten - wird stark von subjektiven Faktoren beeinflusst.³⁶⁸ Damit ist auch zum Teil erklärt, warum die Modelle der einzelnen IT-Consulter bei ein und derselben Fragestellung derart voneinander abweichen: Das Kostenmodell der *Gartner Group* etwa zeichnet sich durch die extensive Berücksichtigung der indirekten Kosten aus. *Treber*, *Teipel* und *Schwickert* meinen, dies sei die Hauptursache für die im Vergleich zu Konkurrenzmodellen höheren TCO-Werte.³⁶⁹

Trotz berechtigter Kritik am TCO-Modell, kann es als Teil einer umfassenden Bewertungssystematik sehr nützlich sein. Zunehmende Bedeutung gewinnt die TCO-Analyse zur Abbildung der Kostenseite im Rahmen von ROI-Berechnungen.³⁷⁰ Die TCO mit dem ROI zu verknüpfen, um sinnvolle Aussagen über den Beitrag der IT zum Unternehmenserfolg treffen zu können, das sollte auch das Ziel eines CIO sein.

³⁶³ vgl. *Grob/Lahme* (2004): S. 2

³⁶⁴ vgl. *Hoffman* (2002): S. 52

³⁶⁵ vgl. *Ullerich* (2004): S. 3f

³⁶⁶ vgl. *Zillich* (2002), online

³⁶⁷ vgl. *Zillich* (2002), online; *Eggert/Schmid* (2006): S. 52

³⁶⁸ vgl. *CSC* (2004), online

³⁶⁹ vgl. *Treber/Teipel/Schwickert* (2004): S. 27

³⁷⁰ vgl. *Treber/Teipel/Schwickert* (2004): S. 44

IV EINFLUSS DER USABILITY VON ERP-SYSTEMEN AUF IHRE TOTAL COST OF OWNERSHIP

„Badly designed business software makes people dislike their jobs. Their productivity suffers, errors creep into their work, they try to cheat the software, and they don't stay in the job very long. Losing employees is very expensive, not just in money but in disruption to the business, and the time lost can never be made up.“³⁷¹

Betriebswirtschaftliche Standardsoftwaresysteme helfen Unternehmen bei der Abwicklung ihrer Geschäftsprozesse. Mit ihrer Einführung sind häufig wirtschaftliche Erwartungen, vor allem der Erhalt der Wettbewerbsfähigkeit, verbunden.³⁷² Ein entscheidender Faktor für die Realisierung von Nutzeneffekten durch die ERP-Software ist jedoch ihre Usability. Warum der ERP-Usability eine solch wesentliche Rolle zukommt, lässt ein Blick auf ihre Auswirkungen auf die Softwarenutzung rasch erkennen: Anwenderunternehmen profitieren von verringerten Gesamtkosten der Software - einerseits durch den reduzierten Schulungsaufwand, andererseits durch eine effizientere Bedienung der Software. Ein ERP-System mit hoher Usability motiviert nicht zuletzt die End-User; sie verwenden dann mehr der angebotenen Funktionalitäten, auch selten auftretende Prozesse werden tendenziell eher im System abgebildet.³⁷³

Der Einfluss der Usability auf die Gesamtkosten eines ERP-Systems war für Anwenderunternehmen schon seit jeher von hoher Relevanz - wenn dieser auch bislang weitestgehend negiert wurde. Ein verstärkter Trend zur Dezentralisierung und damit einhergehend eine zunehmende Bedeutung von gelegentlichen Benutzern³⁷⁴ lassen die Usability von Unternehmenssoftware aber immer wichtiger werden. Im Falle von gelegentlichen Benutzern ist Usability ein unabdinglicher Faktor, um - ohne lange Wiedererlernzeiten - die Software effizient und effektiv bedienen zu können.

³⁷¹ Cooper (2004): S. 40

³⁷² vgl. Gadatsch (2003): S. 301

³⁷³ vgl. Plattner (1999): S. 82

³⁷⁴ vgl. Forrester Research (1999) nach Plattner (1999): S. 83

Zunächst soll die bisherige Forschung zur Usability thematisiert sowie Anforderungen an die Usability von ERP-Systemen konkretisiert werden. Anschließend gilt es, den Gesamtkosten in einem ERP-spezifischen Kontext auf den Grund zu gehen: Welche Kostenkategorien sind bei Implementierung und Einsatz von Bedeutung? Wie könnte ein TCO-Modell für ERP-Systeme aussehen? Welche Potentiale zur Senkung der TCO von ERP-Systemen gibt es? Der dritte Teil dieses Kapitels wendet sich schließlich der Frage zu, welche Auswirkungen die Usability von ERP-Systemen auf deren Gesamtkosten hat. Kosten-Nutzen-Modelle für Usability Engineering, welche den Zusammenhang zwischen einer erhöhten Usability und verringerten Kosten abstrahiert darstellen, werden zunächst vorgestellt. In der Folge werden konkrete Auswirkungen der ERP-Usability auf die TCO diskutiert. Abschließend wird auf unternehmensspezifische Aspekte eingegangen, welche über die Bedeutung von Usability im jeweiligen Anwendungskontext entscheiden.

1 Usability im ERP-spezifischen Kontext

Die Usability von ERP-Systemen wurde lange Zeit sehr stiefmütterlich behandelt.³⁷⁵ ERP-Softwareprodukte waren nicht selten aufgrund negativer Usability-Kritiken in den Schlagzeilen.³⁷⁶ Übliche Usability-Einwände an ERP-Produkten: Die User Interfaces sind nicht intuitiv gestaltet, sie frustrieren Benutzer und stehen einer erfolgreichen Implementierung und Benutzung der Software im Wege.³⁷⁷ ERP-Systeme muten den Anwendern viele unnötige Funktionalitäten zu. Die Funktionsbreite von betriebswirtschaftlicher Standardsoftware ist für den konstant steigenden Administrationsaufwand der IT-Abteilungen hauptverantwortlich. Unternehmensinterne IT-Abteilungen berichten von einer überproportional wachsenden Administration, Personal-mangel, steigender Instabilität sowie lange andauernden Projekten.³⁷⁸

Die Anwender würden jedoch zunehmend kritischer werden und die Hersteller von ERP-Software verstärkt auf Benutzerfreundlichkeit setzen. So berichtet eine Sprecherin der Fachgruppe Software-Ergonomie der Gesellschaft für Informatik, dass die Nachfrage nach Unterstützung beim Usability Engineering zunehme und Softwarehersteller gute Benutzbarkeit als umsatzfördernd erkannt hätten. Andere teilen diese Meinung nicht: Die Funktionen würden bei der Auswahl eines ERP-Systems im Vordergrund stehen; die Benutzungsqualität der Software würde erst beim Umgang mit dem Programm bemerkbar werden. Deshalb sei die Usability höchstens für den Wiederkauf entscheidend.³⁷⁹ Ähnlich auch das Urteil anderer Autoren: „[...] the importance of the ergonomist's role has not been realized either by the ERP community or addressed by the ergonomic community.“³⁸⁰

³⁷⁵ vgl. o.V. (2004): S. 9

³⁷⁶ Usability-Experten beanstandeten etwa die Zuweisung von Sicherheitsprofilen, die Konfiguration von Organisationsstrukturen und den Download von Patches. Vgl. o.V. (2004): S. 9. Zum steigenden Administrationsaufwand von ERP-Systemen siehe o.V. (1999): S. 1. Außerdem Chew/Orlov/Herbert (2003) nach Gilbert (2003), online.

³⁷⁷ vgl. Babaian/Lucas/Topi (2004): S. 164

³⁷⁸ vgl. o.V. (1999): S. 1

³⁷⁹ vgl. o.V. (2004): S. 9

³⁸⁰ Bishu/Kleiner/Drury (2001): S. 152

Zu beachten gilt es freilich, dass die Komplexität von betriebswirtschaftlicher Standardsoftware eine hohe Usability des Produkts tendenziell hemmt. ERP-Systeme umfassen eine große Anzahl von unternehmerischen Prozessen; zudem arbeiten mit ihnen unterschiedliche User mit ihren verschiedenartigen Bedürfnissen und einem nicht einheitlichen Grad an Expertise.³⁸¹ „By its nature, software that promises to integrate all the functions within a unit and all the units within an organization will be complex. In addition, [...] the software, with all its suite of functional modules, is sufficiently generic to be adaptable to the needs of a number of diverse organizations.“³⁸² Dass es eine spezielle Herausforderung darstellt, eine akzeptable Usability bei solch komplexen Systemen zu erreichen, sollte aber dennoch keine Entschuldigung für eine mangelhafte Usability sein. Denn schließlich handelt es sich um Softwareprodukte, die in so vielen Unternehmen im Einsatz sind wie nur wenig andere Systeme.³⁸³

1.1 Überblick über die bisherige Forschung

Literatur, welche sich explizit mit der Usability von ERP-Systemen auseinandersetzt, gibt es kaum. Ein 2001 erschienener Artikel, in dem *Esteves* und *Pastor* die bisherige ERP-Forschung zusammenfassen, beinhaltet keine einzige Referenz zu einer wissenschaftlichen Arbeit, die sich unmittelbar mit der Usability von betriebswirtschaftlicher Standardsoftware beschäftigt.³⁸⁴ In den letzten Jahren hat das Thema aber scheinbar an Popularität gewonnen und heute findet man bereits einige Forschungsarbeiten zur ERP-Usability. Dennoch fällt auf, dass es dieses Thema bislang in kein namhaftes Journal schaffte.

Tabelle 16: Literatur zur Usability von ERP-Software³⁸⁵

Bedeutung der ERP-Usability im Auswahlprozess	Bernroider/Koch (2000); Sontow (2006b)
User Satisfaction mit ERP-Software	Verdaasdonk/Oomen (2001); Wu et al. (2002); Sarpola (2003); Zviran (2003), online; Thavapragasam (2004); Kristiansen (2005)
Zusammenhang zwischen Usability und Erfolg von ERP-Systemen	Bishu/Kleiner/Drury (2001)
Usability Testing von ERP-Systemen	Chew/Orlov/Herbert (2003) nach Babaian/Lucas/Topi (2004)
Evaluierung von Schwächen der ERP-Usability	Bishu/Kleiner/Drury (2001); Babaian/Lucas/Topi (2004); Lucas/Babaian/Topi (2004), online; Topi/Lucas/Babaian (2005); Babaian/Lucas/Topi (o.J.)
Empfehlungen zur Verbesserung der ERP-Usability	Babaian/Lucas/Topi (2004); Lucas/Babaian/Topi (2004), online; Babaian/Lucas/Topi (o.J.)
Usability von ERP-Handbüchern	Scott (2005)

³⁸¹ vgl. Babaian/Lucas/Topi (o.J.): S. 19

³⁸² Bishu/Kleiner/Drury (2001): S. 148

³⁸³ Zur Verbreitung von ERP-Systemen siehe Kapitel III 1.3 Eckdaten zum ERP-Markt.

³⁸⁴ vgl. Esteves/Pastor (2001)

³⁸⁵ eigene Darstellung

Nachfolgend wurde der Versuch angestellt, einen Überblick über die zu diesem Thema verfügbare Literatur zu geben - Tabelle 16 fasst die in der Folge betrachteten Arbeiten zusammen. Es wurde auch Literatur berücksichtigt, welche das Thema nur als einen Teilaspekt behandelt. Ebenso werden im Folgenden Arbeiten thematisiert, welche die User Satisfaction von ERP-Systemen zum Gegenstand ihrer Untersuchung machten. Auf die Eignung des Konstrukts der User Satisfaction als übergreifender Maßstab zur Beurteilung der Software Usability wurde bereits in Kapitel II 3.3 Methoden zur Beurteilung der Software Usability eingegangen.

Zur Bedeutung der Usability von ERP-Produkten im Auswahlprozess der Software liegen zwei Studien vor. *Bernroider* und *Koch* untersuchten die Wichtigkeit von Entscheidungskriterien bei der ERP-Auswahl auf einer Skala von 1 („irrelevant“) bis 4 („sehr wichtig“). Obwohl Ergonomie als Kriterium von nur 1,7 Prozent der befragten Unternehmen als irrelevant eingestuft wurde, rangiert es - reiht man die Kategorien nach der Anzahl ihrer Einschätzungen als „sehr wichtig“ - nur auf Platz 20 unter den dreißig abgefragten Kategorien.³⁸⁶ Eine Untersuchung der *Trovarit AG* kommt zum Schluss, dass die Systemergonomie als Entscheidungskriterium besonders für Unternehmen unter 100 Mitarbeitern relevant ist. Etwa 45 Prozent der befragten Unternehmen in dieser Gruppe geben an, dass die einfache und verständliche Benutzerführung ein Grund für die Auswahl der Software war. Bei Unternehmen zwischen 100 und 500 Mitarbeitern liegt dieser Anteil bei knapp 35 Prozent, bei großen Unternehmen über 500 Beschäftigten sinkt er auf weniger als ein Viertel.³⁸⁷

Dass die User Satisfaction von ERP-Anwendern keine hohen Werte erreicht, darüber sind sich sämtliche Autoren einig. Eine Ausnahme bildet *Zviran*, der die Ergebnisse einer Umfrage unter 200 SAP-Anwendern in einem kanadischen Unternehmen präsentierte und von relativ hohen Zufriedenheitslevels mit dem System berichtete.³⁸⁸ *Wu et al.*, sie untersuchten die User Satisfaction von Endanwendern in Taiwan, konnten hingegen bei einigen der untersuchten Bereiche nur sehr geringe Zufriedenheitslevels feststellen: Dazu zählten auf der Seite der Key-User Bereiche wie die Systemintegration und das Projektmanagement durch den Consultant, das Training sowie das Verständnis für die Software. Die End-User bewerteten die Bereiche Verständnis für die Software, Systemintegration und das Gefühl, als End-User im Projekt involviert zu sein, schlecht.³⁸⁹ *Sarpola*, der in seiner empirischen Studie den finnischen Großhandel unter die Lupe nahm, stellte den Unternehmen die Frage, inwieweit der Einsatz der ERP-Software - auf vier Punkte bezogen, nämlich die Usability, die Reliabilität, die Integrationsfähigkeit mit unternehmensinternen IT-Systemen sowie die IT-Integrationsfähigkeit außerhalb der Unternehmensgrenze - von Erfolg gekrönt war. Usability schnitt unter den vier abgefragten Aspekten am schlechtesten ab.³⁹⁰ Eine qualitative Benutzerbefragung führte *Thavapragasam*

³⁸⁶ vgl. Bernroider/Koch (2000): S. 333

³⁸⁷ vgl. Sontow (2006b): S. 24

³⁸⁸ vgl. Zviran (2003), online

³⁸⁹ vgl. Wu et al. (2002): S. 9

³⁹⁰ vgl. Sarpola (2003): S. 75

an einer großen australischen Universität durch. Das Ergebnis: Nur zwei der neun befragten Anwender waren mit dem ERP-System zufrieden.³⁹¹

Ebenso äußerte sich *Kristiansen* kritisch zur User Satisfaction mit ERP-Systemen: „[...] many organizations adapting an ERP system have experienced problems stemming from end-user dissatisfaction with the new system being too complex and inefficient.“³⁹² Und weiters wies sie darauf hin, dass die komplexen ERP-Systeme die Definition von Usability nach DIN EN ISO 9241-11 wohl kaum erfüllen würden; ein Beispiel sind die komplexe hierarchische Menüstruktur von *SAP*. Die eigentlichen Vorteile von ERP-Systemen würden ihre Usability schwer in Mitleidenschaft ziehen und wären für viele gescheiterte Implementierungen mitverantwortlich. Integration und Branchenneutralität bewirkten eine große Komplexität, wie sie feststellte.³⁹³

Verdaasdonk und *Oomen* stellten die These auf, eine mangelnde Usefulness würde zur Unzufriedenheit von Softwareanwendern führen - dies sei auch im Fall von ERP-Systemen zu beobachten. In ihrer Fallstudie über ein Möbelunternehmen wurden Gründe für die Nicht-Benutzung des firmeneigenen ERP-Systems deutlich: Einerseits war man sich der Möglichkeiten der Softwarenutzung nicht bewusst, andererseits vertraute das Management den aus dem System abgerufenen Informationen nicht. Die Schwierigkeit für Anwender an die gewünschten Informationen zu kommen, führte laut *Verdaasdonk* und *Oomen* zu der beobachteten Unzufriedenheit mit der ERP-Software.³⁹⁴

Bishu, *Kleiner* und *Drury* warfen 2001 als eine der ersten einen Zusammenhang zwischen der Usability und dem Erfolg des Einsatzes von ERP-Systemen auf. System wie Implementierungsprozess wären zu technologie-orientiert und das End-User-Training im Allgemeinen nicht den Bedürfnissen der Endanwender angepasst. Dass ERP-Systeme ihren Erwartungen oft nicht gerecht werden, führten sie auf kostenintensive Implementierungen und das technologie-orientierte Paradigma zurück.³⁹⁵

Durchwegs schlechte Usability-Charakteristika von ERP-Systemen bestätigte auch *Forrester Research* 2003 in einer Studie, in welcher elf verschiedene Standardsoftwareprodukte namhafter ERP-Softwarehäuser - darunter *SAP*, *PeopleSoft*, *Oracle*, *J.D.Edwards* und *Microsoft* - evaluiert wurden. Die Durchführung von Standardaufgaben im System, welche ohne Training zu schaffen sein sollten, brauchte „inordinate patience and expertise“³⁹⁶; die Bewertung der Usability für die gesamte Software fiel für viele untersuchten Systeme schlecht aus. Nicht-intuitive User Interfaces führen zu einer verminderten Produktivität und erhöhten Kosten für Unternehmen, welche diese Produkte einsetzen - so das eindeutige Urteil der Autoren.³⁹⁷

³⁹¹ vgl. Thavapragasam (2004): S. 527

³⁹² Kristiansen (2005): S. 1

³⁹³ vgl. Kristiansen (2005): S. 2f

³⁹⁴ vgl. Verdaasdonk/Oomen (2001): S. 6ff

³⁹⁵ vgl. Bishu/Kleiner/Drury (2001): S. 152ff

³⁹⁶ Gilbert (2003), online

³⁹⁷ vgl. Chew/Orlov/Herbert (2003) nach Babaian/Lucas/Topi (2004): S. 164; Gilbert (2003), online

Anwenderunternehmen sollten eine bessere Usability von betriebswirtschaftlicher Standardsoftware einfordern: „In a buyer’s market, customers should be demanding better usability.“³⁹⁸

Um die Usability von ERP-Software entscheidend zu verbessern, empfahlen *Babaian, Lucas* und *Topi* in ihrem 2004 erschienen Konferenzbeitrag Grundsätze der Collaboration Theory³⁹⁹ als Guideline für das Design von ERP User Interfaces heranzuziehen.⁴⁰⁰ In einem weiteren Arbeitsbericht gaben die drei Autorinnen Einblick in die von mangelnder Usability ausgelösten Probleme von Endanwendern. Eine Evaluierung aktueller Benutzungsprobleme sei der erste Schritt für eine Verbesserung der ERP-Usability, so ihre Meinung. Die Fallstudie berichtet von zehn qualitativen Interviews, in welchen sie den ERP-Benutzungsproblemen in einem stark diversifizierten Fortune-500-Unternehmen auf den Grund gingen (siehe Tabelle 17). Die ERP-Software war zum Zeitpunkt der Befragung bereits zweieinhalb Jahre im Einsatz. Alle Interviews wurden in einer Abteilung einer Unternehmenssparte durchgeführt; unter den zehn befragten Personen waren neun End-User. Bei der integrierten Standardsoftware handelte es sich um eine für große Unternehmungen.⁴⁰¹

Tabelle 17: ERP-Benutzungsprobleme infolge mangelnder Usability⁴⁰²

Zurechtfinden im System	Es bedarf großer Anstrengung, bestimmte Funktionalitäten im System aufzuspüren. Neben Navigationsproblemen ist es für Endanwender schwierig, die für einen Gesamtprozess notwendigen Teilaufgaben zu verstehen bzw. sich an diese zu erinnern. End-User kritisieren den fehlenden systeminternen Support, was eine Unterstützung beim Verstehen der Geschäftsprozesse betrifft.
Erledigen von Transaktionen	Viele Benutzungsprobleme stehen mit dem Erledigen einer Transaktion im System im Zusammenhang. Komplexe User Interfaces behindern die End-User, eine Transaktion im System effizient durchzuführen. Überdies ist teilweise eine doppelte Eingabe von Daten notwendig. Welche Probleme eine redundante Datenhaltung mit sich zieht (gesteigerte Fehlerhäufigkeit, Inkonsistenz zwischen Teilsystemen, größere notwendige Speicherkapazitäten), ist bekannt. Transaktionen, die nur unregelmäßig durchgeführt werden, werden schließlich nicht im System abgebildet.
Systemoutput	Den gewünschten Output vom System zu erhalten, stellen Anwender als Problem dar. Das System bietet keine Hilfe, um zu den gewünschten Daten zu gelangen.
Fehlermeldungen	Die Fehlermeldungen des Systems sind unzureichend und teilweise irreführend. Oft sind solche Meldungen zu allgemein gehalten, um den Benutzern eine nützliche Information für die Problemlösung anzubieten.
Terminologie	Die Terminologie, wie sie das ERP-System verwendet, unterscheidet sich von jener seiner Benutzer entscheidend. Das schränkt auch die Nützlichkeit der Systemhilfe für End-User ein, weil sie den Inhalt in der dargebotenen Form nicht verstehen.
Systemkomplexität	Die allgemeine Komplexität des ERP-Systems wird von End-User negativ bewertet.

³⁹⁸ vgl. Orlov (o.J.) nach Gilbert (2003), online

³⁹⁹ Ein Modell, das eine Kollaboration zwischen System und Anwender in den Mittelpunkt stellt, sollte die Basis für das Design von User Interfaces sein. Ziel ist es, dem Anwender ein adäquates Ausmaß an Unterstützung bieten zu können. Eine Software sollte demnach derart gestaltet werden, dass sie als Partner des Anwenders bei der Bewältigung seiner Aufgaben fungiert. Während bislang der Benutzer als einziger mit Prozesswissen ausgestattet ist, sollte die Rolle des Systems ausgeweitet werden. Es sollte nicht länger nur auf Kommandos reagieren und den Anwender aktiv durch die Geschäftsprozesse führen. Zu Collaborative Systems im Allgemeinen siehe auch Grosz. Vgl. Babaian/Lucas/Topi (o.J.): S. 11ff; Grosz (1996): S. 67 ff

⁴⁰⁰ vgl. Babaian/Lucas/Topi (2004): S. 168

⁴⁰¹ vgl. Topi/Lucas/Babaian (2005): S. 129

⁴⁰² eigene Darstellung, Daten entnommen aus Topi/Lucas/Babaian (2005): S. 130ff; Lucas/Babaian/Topi (2004), online

Obwohl die Einführung des ERP-Systems in diesem konkreten Fall ein Erfolg war, deuten die von *Topi*, *Lucas* und *Babaian* dokumentierten Probleme mit der Benutzung der Software auf den beträchtlichen Zeitaufwand für das Erlernen des Systems, die hohe Fehlerrate, die zu einem großen Teil aus einem fehlenden Prozessverständnis resultierte, und die Frustration der Endanwender als Reaktion auf sinnlose Fehlermeldungen, unklare Anweisungen und das Fehlen von Hilfestellungen im System.⁴⁰³

Zu guter letzt sei auf einen Artikel verwiesen, in dem die Autorin die ERP-Usability um einen weiteren bislang vernachlässigten Aspekt bereicherte. *Scott* betonte in dem 2005 veröffentlichten Artikel „*Post-Implementation Usability of ERP Training Manuals: The User's Perspective*“, dass sich die ERP-Usability nicht nur auf die Gebrauchstauglichkeit des ERP-Systems an sich beschränkt, sondern auch die Usability der Softwaredokumentation mit einschließt. Probleme im Zuge einer ERP-Implementierung standen nicht selten mit Defiziten im End-User-Training im Zusammenhang.⁴⁰⁴ Laut einer Studie mit dreißig Industrieunternehmen stellt die Schulung der Endanwender das größte Problem für Unternehmen dar.⁴⁰⁵ Um Kosten zu sparen, werden jene für das Training meist als eine der ersten gekürzt⁴⁰⁶, obwohl eine Stunde effektives Training dem Anwenderunternehmen einen Nutzen von fünf Stunden beschert, so Berechnungen der *Gartner Group*.⁴⁰⁷ Der Rat von *Scott*: Unternehmen sollten die Wichtigkeit von gebrauchstauglichen Handbüchern beachten; sie wirken sich unmittelbar auf die aus der Investition in das ERP-System resultierenden Nutzen aus. Ist die Usability der Training Manuals mangelhaft, so ver(sch)wenden User typischerweise mehr Zeit, um Fehler zu korrigieren, den Help Desk mit Problemen zu konfrontieren oder Kollegen um Hilfe zu bitten. Die Autorin zeigt die Bedeutung von prozessorientierten Trainingshandbüchern auf; Mitarbeiter sollten im Handbuch vorzugsweise die Prozesse des eigenen Unternehmens wieder finden.⁴⁰⁸

1.2 Anforderungen an die Usability von ERP-Systemen

Nachdem im letzten Unterkapitel vor allem ERP-Benutzungsprobleme infolge mangelhafter Usability thematisiert wurden, wird im Folgenden versucht, möglichst konkrete Anforderungen an die Usability von ERP-Software zu definieren.

Das Beachten von softwareergonomischen Grundregeln wird auch bei der Entwicklung von Standardsoftware vorausgesetzt. Dies soll eine bestmögliche Unterstützung der Benutzer durch die Software bei der Erledigung ihrer Geschäftsprozesse sicherstellen. Beispielsweise sollten bei der Benutzung der Software keine wiederholten Fehler entstehen, die zu einem erheblichen Zeitaufwand führen und durch Training oder Erfahrung nicht eliminiert werden können. *Mayhew* und *Mantei* berichteten beispielsweise von einem Textverarbeitungsprogramm, das ein und denselben Funktionsschlüssel für das Schließen eines Dokuments in einem Kontext und für das

⁴⁰³ vgl. *Topi/Lucas/Babaian* (2005): S. 133

⁴⁰⁴ vgl. *Scott* (2005): S. 67

⁴⁰⁵ vgl. *Duplaga/Astani* (2003): S. 71

⁴⁰⁶ vgl. *Slater* (1998), online

⁴⁰⁷ vgl. *Aldrich* (2000) nach *Scott* (2005): S. 67

⁴⁰⁸ vgl. *Scott* (2005): S. 75f

Löschen der Eingaben in einem anderen Kontext verwendet hatte. Selbst erfahrene Benutzer können in so einem Fall nicht erkennen, in welchem Kontext sie sich bewegen.⁴⁰⁹ Ein weiteres Beispiel für allgemeine softwareergonomische Regeln bilden jene zur Systemgestaltung, die das menschliche Wahrnehmen, Denken und Lernen berücksichtigen; sie sollten beim Design von ERP-Systemen ebenso Anwendung finden. Dank ihnen kann der Aufwand für das End-User-Training bedeutend eingeschränkt werden. Kein unwesentlicher Faktor, wenn man bedenkt, dass Schulungsmaßnahmen durchschnittlich für zehn bis fünfzehn Prozent der Implementierungskosten einer ERP-Software verantwortlich sind.⁴¹⁰

Mit dem Einsatz von Methoden des Usability Engineering bei der Gestaltung von ERP-Systemen können ERP-Softwarehersteller überprüfen, ob software-ergonomische Grundregeln bei der Entwicklung des Systems beachtet werden. Die Komplexität und Ausdehnung dieser Softwaresysteme erschwert jedoch die Aufgabe für Softwarehersteller, eine hohe Usability ihrer Produkte zu garantieren, und für Usability-Experten, die Gebrauchstauglichkeit der Software zu analysieren. Denn es genügt nicht, diese für die verschiedenen User Interfaces der angebotenen Funktionalitäten sicher zu stellen - eine ERP-Software mit hoher Usability zeichnet sich gerade auch durch die Konsistenz des Designs der verschiedenen User Interfaces aus. Ein konsistentes Design der Benutzeroberfläche in den einzelnen Modulen gilt für den Bildschirmaufbau mit Fenstern, Menüs, Icons und Farben ebenso wie für die Tastenbelegung und Mausbedienung.⁴¹¹ Ein ERP-System, welches das Kopieren und Einfügen von Daten zwischen zwei Anwendungen teilweise ermöglicht und teilweise nicht, erfüllt den Anspruch auf Konsistenz offensichtlich nicht.⁴¹² Die Anforderung, ein konsistentes ERP-System zu entwerfen, wirkt sich auch grundlegend auf das Usability Testing von ERP-Systemen aus. Einzelne Anwendungen individuell zu beurteilen, reicht nicht aus; es bedarf einer Analyse der integrierten Nutzung der Systemfunktionalitäten, um sinnvolle Aussagen zur Usability eines ERP-Systems treffen zu können.⁴¹³

Babaian, Lucas und *Topi* haben sich mit der Frage, was die Usability von ERP-Systemen ausmacht, gründlicher befasst. Aus ihrer Analyse von Benutzungsfehlern lassen sich zentrale softwareergonomische Notwendigkeiten ableiten. Grundlegend ist für Benutzer ein problemloses Zurechtfinden im System. Ein ERP-System soll dem Benutzer zum einen Hilfestellungen bei der Suche nach bestimmten Funktionalitäten anbieten⁴¹⁴; zum anderen soll in einer Anwendung ein unkompliziertes Erreichen von verwandten Aufgaben und solchen, die zuvor erledigt werden müssen, möglich sein.⁴¹⁵ Ebenso wichtig erscheint eine Unterstützung des ERP-Benutzers mit geschäftsprozessspezifischen Anleitungen und Hinweisen. Die Anzeige von noch zu erledigenden sowie bereits erledigten Schritten im Geschäftsprozess stellt eine wert-

⁴⁰⁹ vgl. Mayhew/Mantei (1994): S. 22

⁴¹⁰ vgl. o.V. (2004): S. 9

⁴¹¹ vgl. Hansen/Neumann (2005): S. 535

⁴¹² vgl. Lucas/Babaian/Topi (2004), online

⁴¹³ vgl. Babaian/Lucas/Topi (o.J.): S. 10

⁴¹⁴ vgl. Topi/Lucas/Babaian (2005): S. 130

⁴¹⁵ vgl. Babaian/Lucas/Topi (o.J.): S. 12

volle Unterstützung dar; der Benutzer kann so mühelos erkennen, auf welcher Prozessstufe er sich gerade befindet.⁴¹⁶ Das System ist in den Augen der drei Autorinnen verantwortlich, den Benutzer gleichsam durch die Geschäftsprozesse zu leiten; „the system [...] should assume responsibility for guiding them through the interrelated processes and helping find the relevant data. One way to achieve this is by having the system make the recipes for complex business tasks available to its users.“⁴¹⁷

Viele der von *Babaian*, *Lucas* und *Topi* beobachteten Benutzungsprobleme standen mit dem Bearbeiten einer Transaktion im Zusammenhang. Nicht wenige Benutzer beschwerten sich über komplexe Interfaces. Die in Anwenderunternehmen häufig selbst erstellten Spickzettel, welche End-User selbst lange nach ihrer Ausbildung verwenden, sind ein sichtbares Ergebnis der zu komplexen Interfaces. Für die Benutzer ist es oft nicht erkenntlich, welche Dateneingaben optional - und somit nicht unbedingt nötig - sind; die Software hingegen weiß, welche Felder optional sind, und sollte dieses Wissen auch an die Benutzer weitergeben. Daten, die in aufeinander folgenden Screens mehrfach einzugeben sind, kosten Zeit für die Mitarbeiter und liefern bekanntlich eine zusätzliche Fehlerquelle.⁴¹⁸ Für selten durchzuführende Prozesse zeigt sich, dass diese schließlich nicht im System abgebildet werden.⁴¹⁹ Dass auf diese Weise wertvolle Informationen für das Unternehmen verloren gehen können und das Nutzenpotential der ERP-Software nicht ausgenutzt wird, ist offensichtlich.

Die Ausgabe der vom Benutzer gewünschten Daten in einem geeigneten Format ist ein weiterer wesentlicher Aspekt für eine anwenderfreundliche Benutzung eines ERP-Systems. Neben der Schwierigkeit für Benutzer, einen bestimmten Output zu erhalten, stellten das Komprimieren der Daten, das teilweise nur in externen Anwendungen möglich war, und ebenso die Komplexität von Abfragetools im System schwere Hindernisse für eine akzeptable Usability dar.⁴²⁰ Ebenso berichteten *Verdaasdonk* und *Oomen* in ihrer Fallstudie von den Unzulänglichkeiten, relevante Daten aus dem ERP-System zu erhalten. Resümierend meinten sie, dass die beobachtete Unzufriedenheit der Benutzer mit dem ERP-System auch auf die mangelhafte Usability der Output-Funktionalitäten zurückgeführt werden kann.⁴²¹

Im Falle von Benutzungsfehlern und einer darauf folgenden Unsicherheit des Benutzers, wie mit der Aufgabe fortgefahren werden muss, ist es entscheidend, dass das ERP-System den Benutzer über Ursachen und mögliche Lösungen des Problems aufklärt - anstatt ausschließlich darüber zu informieren, dass ein Fehler aufgetreten ist. Unzureichende oder irreführende Fehlermeldungen sollten vermieden werden.⁴²² Effektive Kommunikation ist - im Übrigen nicht bloß in Fehlersituationen - ein wesentlicher Faktor für eine erfolgreiche Zusammenarbeit zwischen System und Benutzer. Um eine effektive Kommunikation zu ermöglichen, ist es

⁴¹⁶ vgl. Topi/Lucas/Babaian (2005): S. 130; Babaian/Lucas/Topi (o.J.): S. 14

⁴¹⁷ Babaian/Lucas/Topi (o.J.): S. 15

⁴¹⁸ vgl. Topi/Lucas/Babaian (2005): S. 131

⁴¹⁹ vgl. Lucas/Babaian/Topi (2004), online

⁴²⁰ vgl. Topi/Lucas/Babaian (2005): S. 131; Lucas/Babaian/Topi (2004), online

⁴²¹ vgl. Verdaasdonk/Oomen (2001): S. 9

⁴²² vgl. Babaian/Lucas/Topi (o.J.): S. 16; Topi/Lucas/Babaian (2005): S. 131

unabdingbar, die Begrifflichkeiten des Systems an jene des Anwenderunternehmens anzupassen. Die Terminologie des ERP-Systems und jene des Anwenderunternehmens sollten also möglichst deckungsgleich sein.⁴²³ Dass diese Anforderung jedoch keine leicht zu lösende ist, liegt darin, dass auch die Begrifflichkeiten von Unternehmen zu Unternehmen abweichen - es gibt keine einheitliche Sprache in den verschiedenen Anwenderunternehmen, die von ERP-Softwarelösungen aufgegriffen werden könnte.⁴²⁴ Dennoch stellen Probleme infolge unklarer Begrifflichkeiten eine wichtige Herausforderung für ERP-Softwarehersteller dar.

Im Zusammenhang mit benutzerfreundlicher Kommunikation lassen sich weitere Anforderungen an die Usability eines ERP-Systems definieren: Der Benutzer sollte, nachdem er eine Aufgabe erledigt hat, über die Fortschritte der Bearbeitung durch das System informiert werden. Ferner ist es wünschenswert, dass ein Benutzer rechtzeitig über nicht offensichtliche Auswirkungen einer von ihm getätigten Aktion aufgeklärt wird.⁴²⁵ Die bloße Existenz von Hilfeoptionen macht noch keine ERP-Software mit hoher Usability aus. Für einen ungeübten Benutzer sind diese häufig nicht offensichtlich; der Benutzer soll vom System davon unterrichtet werden, welche systeminternen Hilfestellungen ihm zur Verfügung stehen.⁴²⁶

Bislang wurden ausschließlich systemeigene Charakteristika definiert, die als Anforderungen für eine hohe Usability von betriebswirtschaftlicher Standardsoftware gelten. Abschließend soll der Fokus auf die Bedeutung der dem End-User zur Verfügung stehenden Softwaredokumentation gelenkt werden. Diese hat keinen unmittelbaren Einfluss auf die Software Usability an sich, hat aber laut *Scott* bedeutende Auswirkungen auf den Erfolg oder Misserfolg einer ERP-Implementation: „[...] improving the design of ERP documentation [...] would decrease the pain and cost of ERP implementations.“⁴²⁷ Bedeutend ist ein gebrauchstaugliches Handbuch, wenn der End-User mit einem Problem bei der Anwendung konfrontiert ist. Ein prozessorientiertes auf das Anwenderunternehmen abgestimmtes Softwarehandbuch hilft effizient bei der Lösung eines Benutzungsproblems.⁴²⁸

Eine Zusammenfassung der an dieser Stelle diskutierten Anforderungen an die ERP-Usability gibt Tabelle 18. Manche dieser Anforderungen mögen übertrieben erscheinen; für bestimmte Anwenderunternehmen, beispielsweise für solche, welche viele gelegentliche Benutzer zu ihren Mitarbeitern zählen, sind sie aber unerlässlich für einen erfolgreichen ERP-Einsatz. Es wird bereits an dieser Stelle deutlich, dass die Usability von ERP-Software für verschiedene Unternehmen unterschiedlich bedeutsam ist. Ausführlicher wird dieser Aspekt in IV 3.3 Relevanz der ERP-Usability in Anwenderunternehmen diskutiert.

⁴²³ vgl. Babaian/Lucas/Topi (o.J.): S. 16

⁴²⁴ vgl. Topi/Lucas/Babaian (2005): S. 132

⁴²⁵ vgl. Lucas/Babaian/Topi (2004), online

⁴²⁶ vgl. Babaian/Lucas/Topi (o.J.): S. 13

⁴²⁷ Scott (2005): S. 75

⁴²⁸ vgl. Scott (2005): S. 75

Tabelle 18: Anforderungen an die Usability von ERP-Software⁴²⁹

Allgemeine software-ergonomische Grundregeln	Die Beachtung allgemeiner softwareergonomischer Grundregeln wird auch bei der Entwicklung von ERP-Systemen vorausgesetzt.
Konsistentes Design	Ein konsistentes Design ist bei ERP-Systemen von zentraler Bedeutung für ihre Usability. Um die Arbeit für den Benutzer zu vereinfachen, ist eine gleichartige Gestaltung der Benutzeroberfläche in den einzelnen Modulen unumgänglich.
Einfache Navigation	Das System sollte ein für den Benutzer müheloses und effizientes Auffinden von spezifischen Funktionalitäten unterstützen. Nicht erwünscht ist etwa, dass Funktionalitäten nur mit der Eingabe von bestimmten Codes ansteuerbar sind oder sie in einem tief verschachtelten Menü gesucht werden müssen.
Geschäftsprozess-spezifischer Systemsupport	Die ERP-Software sollte dem Benutzer Anleitungen für das Bearbeiten von Geschäftsprozessen geben. Dazu sollte es erkennen, in welchem Kontext sich der Benutzer im System bewegt und welches Ziel er erreichen will, um den Benutzer schließlich erfolgreich durch den mehrstufigen Geschäftsprozess leiten zu können.
Effiziente Interfacegestaltung	Optionale Datenfelder sollten als solche erkenntlich sein; der Benutzer kann sich bei der Eingabe sicher sein, dass er alle relevanten Felder ausgefüllt hat. Für eine effiziente Interfacegestaltung ist es außerdem unabdingbar, eine wiederholte Abfrage von gleichen Datensätzen in aufeinander folgenden Interfaces zu vermeiden.
Leicht zu erstellender Systemoutput	Für Benutzer sollte es ohne großen Aufwand möglich sein, gewünschte Daten aus dem System zu extrahieren. Der Export von Rohdaten in andere Applikationen wie beispielsweise Microsoft Excel, um so an die gewünschten Daten zu gelangen, stellt keine zufrieden stellende Lösung dar.
Support in Fehlersituationen	Tritt ein Fehler bei der Benutzung des Programms auf, so sollte der Benutzer sowohl über die Ursachen wie auch über mögliche Lösungswege informiert werden. Der bloße Hinweis, dass ein Fehler passiert sei, ist jedenfalls nicht ausreichend.
Verständliche Terminologie	Eine erfolgreiche Zusammenarbeit von System und Benutzer passiert in einem hohen Maße auf einer effektiven Kommunikation zwischen beiden Partnern. Verwenden diese unterschiedliche Begrifflichkeiten, so ist keine effektive Kommunikation möglich. Die Terminologie des Systems sollte möglichst nahe an jene des Anwenderunternehmens herangeführt werden.
Information über Bearbeitungsfortschritt	Hat der Benutzer eine Aktion getätigt, sollte das System ihn klar über den Fortschritt der systeminternen Bearbeitung informieren.
Aufklärung über nicht-offensichtliche Auswirkung einer Benutzeraktion	Hat der Benutzer eine Aktion getätigt, die schwerwiegende Folgen beispielsweise in der Bearbeitung des Geschäftsprozesses oder für das System als Ganzes verursacht, so sollte die ERP-Software dem Benutzer konkret über die Auswirkung seiner Aktion berichten.
Anzeige von Hilfeoptionen	Speziell für ungeschulte oder gelegentliche Benutzer ist es wichtig, dass eine ERP-Software über die von ihr angebotenen Hilfestellungen informiert.
Benutzerfreundliche Software-dokumentation	End-User profitieren von einem benutzerfreundlichen Trainingshandbuch, in dem Anleitungen zum Umgang mit der Software zu finden sind. Eine Darstellung anhand der Prozesse des Anwenderunternehmens ist für Hilfesuchende die bestmögliche Darstellungsform.

⁴²⁹ eigene Darstellung

2 Total Cost of Ownership im ERP-spezifischen Kontext

Dass sich hinter den TCO eines ERP-Systems mehr versteckt als die initialen Softwarelizenzkosten, sollte an dieser Stelle bereits klar sein. Ein TCO-Konzept für ERP-Systeme soll ebenso wie für andere IT-Komponenten einen ganzheitlichen Ansatz verfolgen. Die ERP-Gesamtkosten müssen demnach alle Ausgaben für Planung, Anpassung, Einsatz, Management und Wartung der Software erfassen.⁴³⁰ Kosten für externe Berater im Implementierungsprojekt, zusätzlich nötige Hardware, Wartungsgebühren, Personalkosten für die Implementierung und den Betrieb sowie Schulungskosten für die End-User sind einige Beispiele für konkrete Kostenkategorien einer ERP-Software.⁴³¹ Für Unternehmen gut abschätzbar ist die Höhe der Lizenzkosten, weniger gut abschätzbar sind beispielsweise notwendige Schulungen, zumindest kurzfristig auftretende Produktivitätsverluste, eine eventuell sinkende Mitarbeitermotivation und das Ausmaß an Wartungskosten. Das Konzept der TCO versucht, auch all diesen vordergründig nicht sichtbaren Kostenaspekten gerecht zu werden.⁴³² Dennoch gestaltet sich die Schätzung der Kosten - im Gegensatz zu jener des Nutzens - auch im Falle von ERP-Systemen verhältnismäßig einfach.⁴³³

Die Bedeutsamkeit der TCO-Berechnung wird heute zunehmend hervorgehoben, speziell wenn es um die Auswahl einer ERP-Software geht.⁴³⁴ Denn eine ERP-Software stellt mit ihrer Anschaffung wie auch im Betrieb bedeutende Ausgaben dar. Für kleine und mittlere Unternehmen können alleine die laufenden Supportkosten die größte jährliche IT-Ausgabe darstellen.⁴³⁵

2.1 Allgemeines zu den TCO von ERP-Systemen

Mit ERP-relevanten Auswahlkriterien beschäftigten sich *Bernroider* und *Koch*; dabei gingen sie auch der Frage nach, welche Kategorien der anfallenden Kosten von Anwenderunternehmen in die Auswahlentscheidung miteinbezogen wurden. Für die Quantifizierung der einzelnen Kostenkategorien wurde vor allem auf das Angebot zurückgegriffen oder alternativ eine grobe Abschätzung vorgenommen. Das erklärt auch die am häufigsten berücksichtigten Kostenkategorien: Sowohl die Lizenzkosten wie auch Hardware-Anschaffungskosten können relativ mühe-los Angeboten entnommen werden; sie werden im Auswahlprozess mit einbezogen. Weniger offensichtliche Kosten finden eine geringere Berücksichtigung, wenn es um die Entscheidung für ein ERP-System geht: Die Bewertung der Endbenutzer-, der Reise- und der Wartungskosten beispielsweise bedürfen einer aufwändigeren Ermittlung und Abschätzung und fehlen somit häufiger als Entscheidungsgrundlage. Nur 5 der 138 von *Bernroider* und *Koch* untersuchten Unternehmen - etwa 3,5 Prozent - gaben an, einen umfassenden TCO-Ansatz zu verfolgen. Abbildung 20 zeigt die Ergebnisse der Befragung im Detail.⁴³⁶

⁴³⁰ vgl. Dibbern/Günther/Teltzrow (2005): S. 19

⁴³¹ vgl. Ullerich (2004): S. 3; Gadatsch (2003): S. 301

⁴³² vgl. Bernroider/Koch (2000): S. 330

⁴³³ vgl. Dibbern/Günther/Teltzrow (2005): S. 17

⁴³⁴ vgl. o.V. (2005): S. 14; Powel/Barry (2005): S. 41; Sontow (2006a): S. 43

⁴³⁵ vgl. Powel/Barry (2005): S. 40f

⁴³⁶ vgl. Bernroider/Koch (2000): S. 334

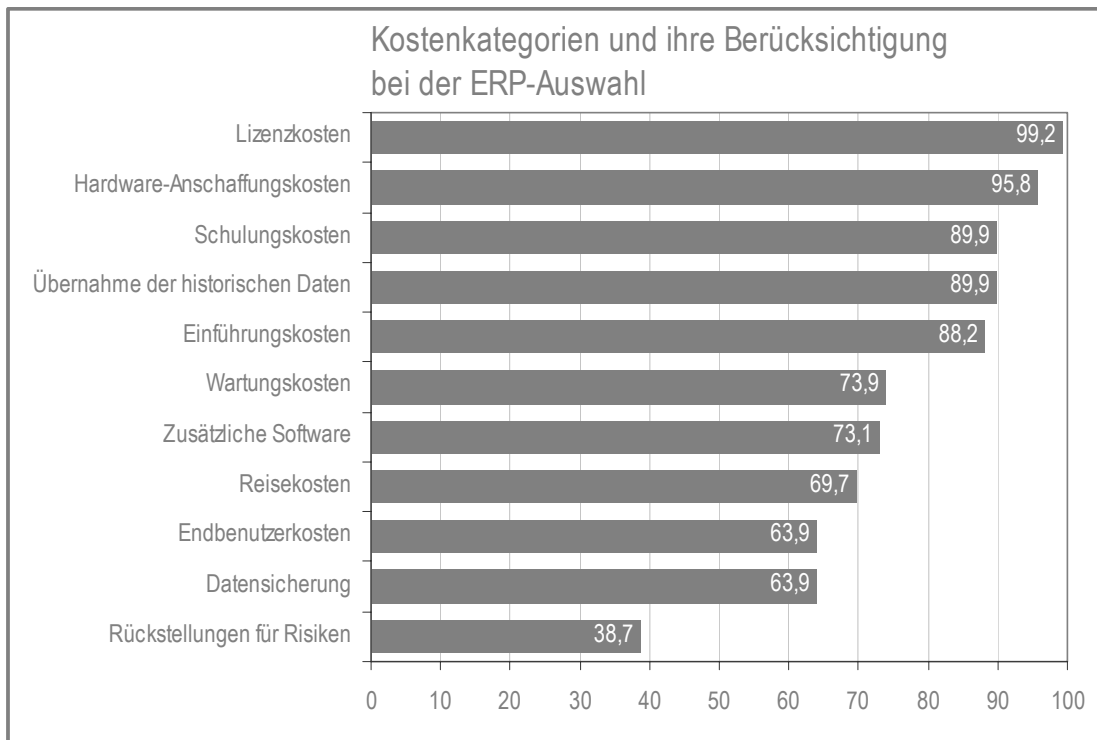


Abbildung 20: Kostenkategorien und ihre Berücksichtigung bei der ERP-Auswahl⁴³⁷

Grundlegend für die TCO von ERP-Systemen kann festgehalten werden, dass die Projektkosten wesentlich stärker von der Benutzerzahl als von der Anzahl der eingesetzten Module abhängen.⁴³⁸ Mit der Anzahl der Lizenzen, und dem damit verbundenen höheren Aufwand für Implementierung und Personal, steigen die ERP-Projektkosten.⁴³⁹ Zeitlich betrachtet, schlagen im ersten Quartal die einmaligen Anschaffungskosten zu Buche; die Kosten für den laufenden Betrieb und Änderungen fallen erfahrungsgemäß nach einiger Zeit auf ein konstantes Niveau.⁴⁴⁰ Für die Höhe der Gesamtimplementierungskosten gilt der Umfang der Ergänzungsprogrammierung als entscheidendes Kriterium: Sind die gewünschten Geschäftsprozesse vom System vollständig durch Parametrisierung der Einstellungen abbildbar, dauert die Implementierung einer ERP-Software vielleicht nur einige Monate; andernfalls kann sie sich über Jahre hinziehen. Die Kosten beschränken sich, wie bereits angesprochen, allerdings nicht auf die Ersteinführung, denkt man nur an die Anpassung des Systems an sich ändernde Geschäftsprozesse, veränderte Rahmenbedingungen und an eine wechselnde Systeminfrastruktur - Kostenfaktoren, die für den laufenden Betrieb eine große Rolle spielen.⁴⁴¹

Mabert, Soni und *Venkataramanan* gingen auf die Frage nach der Aufteilung der ERP-Implementierungskosten auf die wichtigsten Kostenkategorien ein: Die Ausgaben, welche Unternehmen für Consulting-Dienstleistungen tätigen, bilden durchschnittlich betrachtet, mit einem

⁴³⁷ eigene Darstellung, Daten entnommen aus Bernroider/Koch (2000): S. 334

⁴³⁸ vgl. Knolmayer/von Arb/Zimmerli (1997): S. 47

⁴³⁹ vgl. Dibbern/Günther/Teltzrow (2005): S. 19; Francalanci (2001): S. 44

⁴⁴⁰ vgl. Dibbern/Günther/Teltzrow (2005): S. 19

⁴⁴¹ vgl. Hansen/Neumann (2005): S. 537

Anteil von 30 Prozent, die größte Kostenkategorie; bei einigen betrachteten Unternehmen machte die externe Beratung allein 60 Prozent der Implementierungskosten aus. Rund ein Viertel der Gesamtkosten fällt auf Hardware und Infrastruktur. Auf Projektteam, Schulung und Ausbildung der End-User und die Software an sich fallen jeweils 15 Prozent. In Abbildung 21 ist die Aufteilung der Gesamtimplementierungskosten dargestellt; die Zahlen in den Klammern geben jeweils die von den drei Autoren beobachtete Abweichung an.

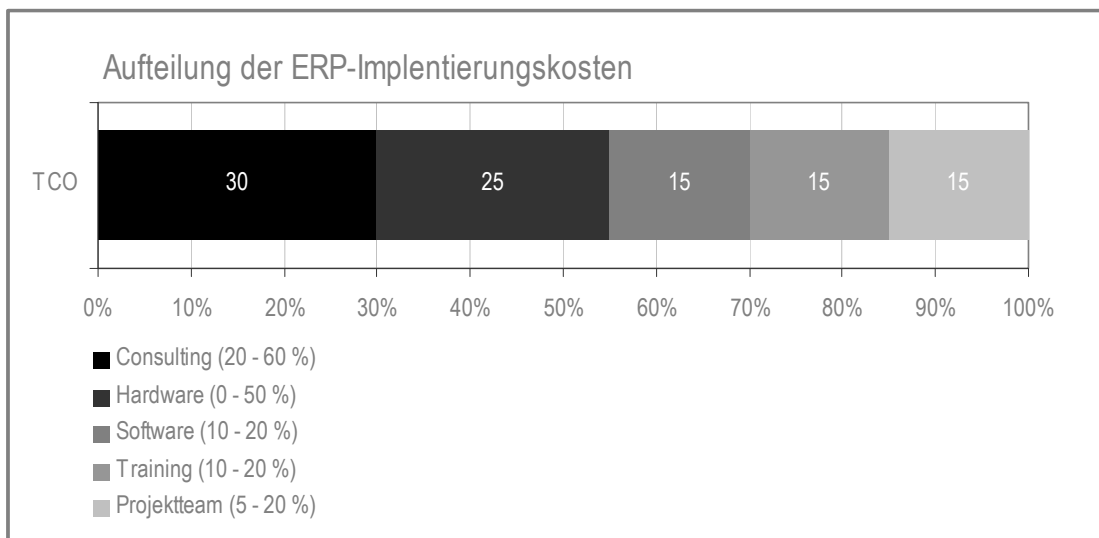


Abbildung 21: Aufteilung der ERP-Implementierungskosten⁴⁴²

Eine gemeinsame Studie der *FH Konstanz*, der *TU Berlin* und des Consulting-Unternehmens *Cap Gemini Ernst & Young* unterstreicht diese Ergebnisse: Consulting-Ausgaben brauchten rund 27 Prozent der Gesamtimplementierungskosten auf. In die Anschaffung von Hardware wurden knapp 17 Prozent investiert. Die Software schließlich kostete den Unternehmen circa 13 Prozent der Einführungskosten. Die Kosten für das Projektteam sowie für Schulung und Ausbildung der Mitarbeiter wurden nicht geschlüsselt: Der Kostenanteil der internen Mitarbeiter im ERP-Implementierungsprojekt betrug beinahe 43 Prozent.⁴⁴³

Etwas abweichend sind die Ergebnisse der Studie von *Knolmayer, von Arb* und *Zimmerli*, die Daten zu 71 Einführungsprojekten der ERP-Software *SAP R/3* in der Schweiz sammelten und auswerteten. Jeweils rund ein Viertel der Gesamtkosten wurde für den Kauf von Hardware und Softwarelizenzen benötigt, etwas mehr als die Hälfte der Kosten für die Einführung, von denen rund ein Fünftel für Training eingesetzt wurden. Die Autoren gaben also ein grobes Verhältnis von 1:1:2 für Hardware-, Software- und die übrigen Einführungskosten an. Sie stellten allerdings die Vermutung an, dass bei den Einführungskosten in der Regel nur die direkten Kosten und - nur selten - die indirekten Kosten einbezogen wurden.⁴⁴⁴ „Da die einführungsbedingten Kosten unter Berücksichtigung aller indirekten Kosten möglicherweise erheblich höher angesetzt

⁴⁴² eigene Darstellung, Daten entnommen aus Mabert/Soni/Venkataramanan (2001): S. 73

⁴⁴³ vgl. Martin/Mauterer/Lempp (2001): S. 6

⁴⁴⁴ vgl. Knolmayer/von Arb/Zimmerli (1997): S. 48

werden müssten, könnte ein in der Literatur angegebenes Kostenverhältnis von 1:1:5 eher realistisch sein⁴⁴⁵, so das Urteil der Autoren, das vom Ergebnis der beiden zuvor genannten Arbeiten nicht beträchtlich abweicht.

Der relativen Höhe der ERP-Projektkosten je nach Unternehmensgröße widmeten sich *Mabert, Soni* und *Venkataramanan* in ihrer Studie über den Einfluss der Organisationsgröße auf die ERP-Implementierung in der US-amerikanischen verarbeitenden Industrie. Der durchschnittliche Anteil der Implementierungskosten am Erlös betrug bei kleinen Unternehmen 5,53, bei mittleren 3,08 und bei großen Unternehmen 2,23 Prozent. Die Aufteilung der Implementierungskosten in verschiedene Kategorien brachte ebenso signifikante Unterschiede hervor, die mit der Größe der Unternehmen einhergehen: So investierten kleine Unternehmen relativ mehr Geld in die Software an sich; große Unternehmen hingegen gaben mehr für ihre Implementierungsteams aus. Überraschend war, dass der Anteil für Consulting und Training zwischen den drei Unternehmensgrößen keine bedeutenden Unterschiede zeigte. Die Kosten für externe Beratung machten etwa das Ein- bis Eineinhalbfache der Softwarekosten aus.⁴⁴⁶ Der Beratungsanteil fiel im Vergleich zu jenem in Abbildung 21 deutlich geringer aus.

Über die Kosten eines ERP-Systems in einem Konzern gibt das Modell von *Goodhue, Wybo* und *Kirsch* Aufschluss: Ein ERP-System in verschiedenen Bereichen eines Konzerns wird demnach kostengünstiger ausfallen, wenn die Unterschiede zwischen den einzelnen Unternehmen gering sind. Sind die Unterschiede zwischen den Unternehmen größer, steigt die Wahrscheinlichkeit, dass ein standardisiertes Informationssystem den jeweils spezifischen Anforderungen nicht genügt. Die drei Autoren berücksichtigen zwar Informationssysteme, welche den unterschiedlichen Ansprüchen genügen, stellen allerdings fest, dass diese hohe Kosten bei der Einführung und Wartung des Systems mit sich ziehen.⁴⁴⁷ *Gattiker* und *Goodhue* raten, während der Einführung den „ugly ducklings“ - jenen Sub-Units, welche einen höheren Grad an Diversifikation gegenüber den anderen Einheiten aufweisen und somit mit individuellen Problemen bei der Einführung zu kämpfen haben - mehr Ressourcen zukommen zu lassen. Andernfalls sind diese häufig mit Systemen konfrontiert, welche nicht zu ihren Anforderungen passen.⁴⁴⁸

2.2 TCO-Modell für ERP-Systeme

Das in Kapitel III 2.3 Beschreibung der TCO vorgestellte Kostenmodell der *Gartner Group* bedarf in einem ERP-spezifischen Kontext einer Adaptierung - so wie jedes TCO-Modell jeweils den spezifischen Gegebenheiten - auch bei seiner Anwendung in einem Unternehmen - angepasst werden sollte.⁴⁴⁹ Um dem Genüge zu tun, sollte in diesem Abschnitt ein für ERP-Systeme geeignetes Gesamtkostenmodell entwickelt werden.

⁴⁴⁵ Knolmayer/von Arb/Zimmerli (1997): S. 48

⁴⁴⁶ vgl. Mabert/Soni/Venkataramanan (2003): S. 239

⁴⁴⁷ vgl. Goodhue/Wybo/Kirsch (1992): S. 308

⁴⁴⁸ vgl. Gattiker/Goodhue (2000): S. 7017

⁴⁴⁹ vgl. Ullerich (2004): S. 8; Zillich (2002), online

Die TCO-Systematik der *Gartner Group* wird die Basis des an dieser Stelle zu entwickelnden Modells bilden. Zuvor soll allerdings noch das TCO-Modell von *SAP* aufgezeigt werden, das einen guten Überblick über die Kostenkategorien einer ERP-Softwarelösung bietet. Es trennt wie üblich in direkte und indirekte Kosten - sechs der sieben grundlegenden Kostenkategorien stellen direkte Kostenarten dar, hinter der letzten Kategorie verbergen sich indirekte Kosten. Ebenso trennt es zwischen anfänglicher Investition und laufendem Betrieb; eine umfassende Ermittlung der relevanten Kostenkategorien von ERP-Systemen bedarf zunächst einer Analyse der anfänglichen Investitionskosten sowie ferner einer Untersuchung der laufenden Betriebskosten. Das Modell mit seinen Kostenkategorien ist in Tabelle 19 näher beschrieben.

Tabelle 19: TCO-Modell nach SAP - Grundlegende Kostenkategorien⁴⁵⁰

Hardware/Software Investment	Diese Kategorie beinhaltet die Kapital- (Abschreibungen und Zinskosten) und Anschaffungskosten. Unter den Kapitalkosten finden sich auch Lizenzgebühren der Softwareausstattung und Anwenderlizenzen.
Implementation	Der gesamte Prozess von der Vorbereitungsphase bis hin zur produktiven Umsetzung des Projektvorhabens ist in dieser Kostenkategorie zusammengefasst.
Hardware/Software Ongoing Costs	Die laufenden Hardware- und Softwarekosten bildet dieser Bereich ab.
Operations	In dieser Kategorie schlagen sich alle Aufwendungen zu den betriebswirtschaftlichen Prozessen nieder. Sie umfasst den gesamten Supportbereich bis hin zum Service Desk und Wartungsbereich.
Continuous Improvement Projects	Diese Kostenkategorie beinhaltet Kosten für neue Software und Releases zur Umsetzung neuer Strategien sowie zur Verbesserung bestehender Prozesse.
Upgrade Projects	Alle Kosten im Zusammenhang mit dem Einsatz neuer Releases und dem Einspielen von Upgrades werden hier berücksichtigt, also beispielsweise auch jene Kosten, die aufgrund Vorbereitung, Setup, Training und Tests, entstehen.
End User Usage	Die Kostenkategorie End User Usage betrachtet die indirekten Kosten eines ERP-Einsatzes. Aufwände, die durch Ineffizienzen - etwa durch die Nicht-Verfügbarkeit eines Systems - ausgelöst werden, fließen hier ein.

Das im Folgenden präsentierte TCO-Modell übernimmt, vor allem bezüglich der Gliederung der Kosten in direkte und indirekte, die Systematik der *Gartner Group*. Die Kostenkategorien werden allerdings adaptiert, um den ERP-spezifischen Kontext ausreichend zu berücksichtigen. Die direkten Kosten des Einsatzes einer ERP-Software sind in fünf Kategorien eingeteilt (siehe Tabelle 20): Es sind dies die Anschaffungskosten für Hardware, die Lizenzgebühren für die Softwarenutzung, die Implementierungskosten, die sämtliche Kosten der Planung, Konzeption und Realisierung des ERP-Projekts beinhalten, ferner die Schulungs- und Ausbildungskosten sowie die Wartungs- und Supportkosten. Bei ERP-Investitionen stellen Implementierungs-, Wartungs- und Supportbereich üblicherweise die größten Kostenblöcke dar.⁴⁵¹

⁴⁵⁰ eigene Darstellung, Daten entnommen aus Ullerich (2004): S. 8ff

⁴⁵¹ vgl. Ullerich (2004): S. 8

Tabelle 20: Total Cost of Ownership (TCO) von ERP-Systemen - Direkte Kosten⁴⁵²

Hardware	Anschaffungskosten	Kosten der Anschaffung von zusätzlich notwendiger Hardware für den Betrieb des ERP-Systems
Software	Systemlizenzen	Kosten für Anwenderlizenzen
	Datenbanklizenzen	Kosten für Datenbanklizenzen
Implementation	Projektmanagement	Kosten für das Projektmanagement
	Projektvorbereitung	Kosten für Planung und Konzeption der Einführung
	Technischer Setup	Kosten für Installation und Konfiguration
	Business Setup	Kosten für die Definition der Geschäftsprozesse, für Add-ons, für Datenmigration, User Management etc.
	Realisierung	Kosten für die Anbindung von Schnittstellen
	Testing	Kosten des Testing und des Aufbaus eines Produktivsystems
	Go-live-Support	Kosten für den Support beim Go-Live
	Business Process Reengineering	Kosten für Aktivitäten des Business Process Reengineering
Training	Key-User Schulung	Kosten für die Ausbildung der Key-User
	End-User Schulung	Kosten für die Planung und Durchführung der Endanwenderschulung
Operations	Wartung	Kosten für die Wartung der Hardware und Pflege der Software
	End-User Support	Kosten für den Support der Endanwender, z.B. Personalkosten für Mitarbeiter des Service Desk

Mit Ausnahme der Kostenkategorie Implementation stellen alle übrigen Kategorien laufende Kosten dar. Auch wenn bestimmte Kostenkategorien nicht zyklisch anfallen, so sind sie dennoch den laufenden Kosten zuzurechnen, etwa Ausgaben für Hardware und ebenso jene für Schulung und Ausbildung. Ein Teil der laufenden Betriebskosten ist in Form der jährlichen Wartungsgebühr für die Software und Serviceleistungen vertraglich fixiert. Weitere wesentliche Aspekte sind bestmöglich abzuschätzen, beispielsweise der interne Aufwand für die Betreuung des ERP-Systems oder die Releasehäufigkeit und die mit einem neuen Release zu erwartenden Kosten.⁴⁵³ Bereits angedeutet, finden sich in den jeweiligen direkten Kostenkategorien sowohl Kosten für externe Produkte und Leistungen als auch interne Leistungsaufwendungen. Der Kostenbereich Wartung in der Kostenkategorie Operations umfasst beispielsweise die jährlichen Wartungsgebühren für Software und Serviceleistungen wie auch den internen Aufwand der Wartung der ERP-spezifischen IT-Komponenten.

Zu den indirekten Kostenkategorien zählen einerseits Endanwenderkosten - nämlich jene für Self- und Peer-to-Peer-Support, für das Casual Learning und für das End-User Training -,

⁴⁵² eigene Darstellung, Daten entnommen aus Gartner Group (2003), online; Ullerich (2004): S. 10f; Scherer (2005): S. 17

⁴⁵³ vgl. Scherer (2005): S. 17f

andererseits Kosten für Systemausfälle sowie Kosten, die anfallen, wenn End-User auf Supportdienste warten. Bei den indirekten Kosten ist zu beachten, dass die Kosten dafür jeweils mit den Personalkosten anzusetzen sind. So stellen die in Tabelle 21 dargestellten Trainingskosten die Kosten der Arbeitszeit dar, während der ein Mitarbeiter an Schulungsmaßnahmen teilnimmt. Da die indirekten Kosten wiederholt auftreten, stellen sie laufende Kosten dar.

Tabelle 21: Total Cost of Ownership (TCO) von ERP-Systemen - Indirekte Kosten⁴⁵⁴

End-User Operations	Self- und Peer-to-Peer-Support	Kosten der Selbsthilfe sowie der informellen Hilfe von Kollege zu Kollege
	Casual Learning	Kosten für bewusstes eigenständiges Lernen
	Formales End-User Training	Kosten für den Besuch von ERP-Schulungen
Ausfallzeiten	Geplante und ungeplante Ausfallzeiten	Kosten der Produktivitätsminderungen der Endbenutzer aufgrund der Wartezeit auf die Hilfe des Support oder aufgrund des Ausfalls eines Systems

Das Modell wird in Kapitel IV 3.2 Konkrete Auswirkungen der ERP-Usability auf die TCO erneut aufgegriffen, um die Auswirkungen der ERP-Usability auf die TCO zu veranschaulichen.

2.3 Fallstudien über die Gesamtkosten von ERP-Software

Um die im letzten Teilabschnitt vorgestellten Kostenkategorien eines ERP-Systems etwas mit Leben zu füllen, sollten an dieser Stelle einige Fallstudien und ihre Ergebnisse zu den Gesamtkosten von betriebswirtschaftlicher Standardsoftware wiedergegeben werden: Da die Fallstudien teils nur direkte Kosten beinhalten⁴⁵⁵, teils zwar die Kosten in direkte und indirekte Kosten aufteilen⁴⁵⁶, dies aber nicht der Methodik der *Gartner Group* entspricht und es sich bei diesen als indirekt identifizierten Kosten vornehmlich um direkte handelt⁴⁵⁷, muss auf die Betrachtung der indirekten Kosten verzichtet werden.

Zunächst werden die Ergebnisse von Fallstudien zu den Kosten von ERP-Einführungsprojekten aufgezeigt. *Becker, Fleisch* und *Österle* präsentierten in ihrer Fallstudie über *Kontron Elektronik*, ein mittelständisches Hightechunternehmen, die Projektkosten der Einführung von *SAP R/3*. Die extern entstandenen Kosten im Einführungsjahr 1994 betragen 1.858.000 Deutsche Mark. Investitionen in die Hardware waren in der Höhe von 245.000 Mark notwendig. Die Anschaffung der Software an sich kostete dem Unternehmen 650.000 Mark. Für 310 Tage an Beratungs- und Schulungsleistungen musste das Unternehmen 963.000 Mark in die Hand nehmen. Zusätzlich zu den externen Beratertagen investierte *Kontron Elektronik* im gleichen Zeitraum 960 Manntage in das Projekt. Zum Ablauf des End-User-Trainings ist zu ergänzen, dass die

⁴⁵⁴ eigene Darstellung, Daten entnommen aus Gartner Group (2003), online

⁴⁵⁵ vgl. Becker/Fleisch/Österle (1997)

⁴⁵⁶ vgl. Powell/Barry (2005)

⁴⁵⁷ Powell und Barry zählen jegliche interne Personalkosten im Zusammenhang mit dem ERP-Projekt zu den indirekten Kosten. Sämtliche Aufwendungen für operative Aufgaben, welche die IT-Abteilung erfüllt, werden aber nach dem TCO-Modell der *Gartner Group* zu den direkten Kosten gezählt, sind sie doch in Budgets als IT-bezogene Kosten enthalten. Siehe dazu im Speziellen Tabelle 14 und Tabelle 15.

Schulungen vom Unternehmen mit Unterstützung von externen Beratern größtenteils Inhouse vorgenommen wurden. Die für das Projekt engagierten Berater stellten spezifische Schulungen zusammen, um so die Inhalte am jeweils aktuellen Bedarf orientieren zu können.⁴⁵⁸

Dolmetsch et al. werteten die Daten von vier Unternehmen aus, die mit der Einführungsmethode *AcceleratedSAP*, kurz *ASAP*, die ERP-Software *SAP R/3* implementiert haben. Diese Methode hilft Unternehmen, das System auf Basis von vordefinierten Referenzprozessen zu implementieren.⁴⁵⁹ Aufgrund der Fokussierung auf die Implementierungsphase wurden jeweils ausschließlich die Projektkosten betrachtet. Das Projektbudget von *Crosfield*, ein mittelständisches Unternehmen der chemischen Industrie, betrug 1.200.000 Niederländische Gulden. Davon wurden 80.000 Gulden in die Hardware investiert; für ein Server-Upgrade nach Abschluss des Einführungsprojekts wurden 180.000 Gulden geplant. Bei den Lizenzgebühren für *SAP R/3* konnte das Unternehmen davon profitieren, dass auch die Konzernmutter diese ERP-Software im Einsatz hatte; 100.000 Gulden waren an Lizenzgebühren zu entrichten. Externe Beratungsleistungen machten 760.000 Gulden aus, wobei ein *SAP*-Beratertag in den Niederlanden durchschnittlich etwas mehr als 2.000 Gulden kostete. Für die Schulung der Systemadministratoren und Endanwender rechnete das Unternehmen mit 80.000 Gulden. Reisekosten, Bonusse und sonstiges wurden mit 180.000 Gulden veranschlagt. Neben den externen Beratertagen wurden im gleichen Zeitraum etwa zweieinhalb Mal so viele interne Mitarbeitertage - also etwa 950 - investiert.⁴⁶⁰

Mitek, ein medizinischer Gerätehersteller und das zweite von *Dolmetsch et al.* analysierte Unternehmen, plante 1,5 Millionen US-Dollar an externen Kosten in das Einführungsprojekt zu investieren. Tatsächlich beliefen sich die Beratungsleistungen auf 1,1 Millionen, die Softwarelizenzen kosteten 0,2 und die zusätzlich nötige Hardware 0,15 Millionen. Das Projekt konnte schließlich mehr als 3 Prozent unter Budget abgeschlossen werden.⁴⁶¹

Nokia SWP, ein Geschäftsbereich des finnischen Telekommunikationskonzerns, konnte das Projekt sogar 12 Prozent unter Budget abschließen, wobei darin nur die extern entstandenen Projektkosten berücksichtigt sind. Über die konkrete Höhe der Kosten machte das Unternehmen keine Angabe; die Verteilung ist wie folgt: 46 Prozent der Einführungskosten investierte das Unternehmen in Beratung, 29 Prozent in die Software, 18 Prozent in die Hardware und 7 Prozent in die Schulung. Die Höhe der Kosten für Beratungsleistungen und Schulung fällt im Vergleich gering aus: Grund für den niedrigen Beratungskostenanteil ist die überdurchschnittlich hohe Beanspruchung von internen Mitarbeiterkapazitäten im Projekt. Der externe Personalanteil machte nur 17 Prozent aus; insgesamt wurden 6950 Mitarbeitertage und 1465 Beratertage gezählt.⁴⁶²

⁴⁵⁸ vgl. Becker/Fleisch/Österle (1997): S. 10f

⁴⁵⁹ vgl. Dolmetsch et al. (1998): S. ii

⁴⁶⁰ vgl. Dolmetsch et al. (1998): S. 22

⁴⁶¹ vgl. Dolmetsch et al. (1998): S. 34

⁴⁶² vgl. Dolmetsch et al. (1998): S. 48f

Beim vierten von *Dolmetsch et al.* untersuchten Unternehmen handelte es sich um DMC Prints; das US-amerikanische Unternehmen koloriert und vertreibt Textilien. Das Projektbudget in der Höhe von 1,3 Millionen US-Dollar wurde bewusst knapp kalkuliert - mit dem Ergebnis, dass es schließlich um 46 Prozent überzogen wurde. Insgesamt investierte das Unternehmen folglich rund 1,9 Millionen US-Dollar. Zu gering budgetiert waren zum einen die Kosten für das Business Process Reengineering (BPR), zum anderen wurden bei der Budgeterstellung Spesen - wie etwa Reisetätigkeiten und Hotels - vernachlässigt. Die Aufteilung der gesamten Implementierungskosten gestaltete sich wie folgt: 49 Prozent für Beratungsleistungen, 14 Prozent für Software, 12 Prozent für Hardware, 10 Prozent für Reisespesen, weitere 7 Prozent für anderweitige Beratungstätigkeiten und jeweils 5 Prozent für Schulung und Schnittstellen. Jedoch sind die wahren Schulungskosten deutlich höher - auch diese wurden im Budget zu gering angesetzt; die zusätzlichen Kosten fielen allerdings erst nach dem Produktivstart an und sind daher nicht in den Projektkosten enthalten. Der externe Beratungsaufwand umfasste 603 Tage; jedem Beratertag stehen etwa drei interne Mitarbeitertage gegenüber.⁴⁶³

Powel und *Barry* berichteten in ihrem Artikel über die *Gonzaga University*. Die ERP-Implementierung, die von 1995 bis 1997 andauerte, wurde sechs Jahre später einer Beurteilung unterzogen. Die Kosten der ERP-Einführung werden mit 3,5 Millionen US-Dollar beziffert. Davon fallen rund 1,3 Millionen Dollar auf Kosten im Zusammenhang mit Hardware, Software und Beratungsdienstleistungen, nämlich genauer auf Implementierung, Schulung und Customizing. Budgetiert waren sämtliche Ausgaben für diese Bereiche nur mit knapp einer Million Dollar; das Budget wurde in der Höhe von einem Drittel des geplanten Budgets überschritten. Die restlichen 2,2 Millionen Dollar wurden für die Kosten der internen Projektteams aufgewendet, die Aufgaben im Zusammenhang mit der Migration vom alten ins neue System und mit der Implementierung des neu angeschafften ERP-Systems erledigten.⁴⁶⁴

Bemerkenswert ist, dass kaum Fallstudien zu den laufenden Kosten des Einsatzes von ERP-Software existieren. Die weitgehend fehlende Beschäftigung mit diesem Thema beschränkt sich aber nicht nur auf Fallstudien; auch wissenschaftliche Artikel zu den Folgekosten von ERP-Systemen sind Mangelware. Grund für dieses Phänomen dürfte wohl der sein, dass die Literatur zum Thema ERP wie auch die gesamte Forschung sehr praxisorientiert ausgerichtet sind und bislang viele Unternehmen damit beschäftigt waren, ERP-Systeme einzuführen. Nun wäre es aber an der Zeit, die Forschung zum Thema Betrieb und Einsatz von ERP-Systemen auszuweiten, wo doch viele Unternehmen eine Implementierung hinter sich gebracht haben und beispielsweise in Studien über Best Practices wertvolle Informationen für andere Unternehmen aufgearbeitet werden könnten. Zwei Fallstudien, die auch auf den laufenden Betrieb von ERP-Systemen und seine Kosten eingegangen sind, werden in der Folge vorgestellt.

⁴⁶³ vgl. Dolmetsch et al. (1998): S. 62

⁴⁶⁴ vgl. Powel/Barry (2005): S. 41ff

Bei *Kontron Elektronik* fielen Folgeinvestitionen für die Hardware an, deren Höhe sich in den Jahren 1995 und 1996 auf weitere 330.000 bzw. 250.000 Deutsche Mark belief. Zusätzlich sind die jährlichen Wartungskosten für die Hardware, die bei etwa 100.000 Mark liegen, zu berücksichtigen. Die Lizenzgebühren für die Softwarenutzung durch 230 Named User machten jährlich etwa 120.000 Mark aus. Weitere externe Beratungsleistungen schlugen in den Jahren 1995 und 1996 mit 158.000 bzw. 27.000 Mark zu Buche.⁴⁶⁵

Für die jährlichen Kosten des laufenden Betriebs an der *Gonzaga University* berechneten die Autoren, dass dieser drei Viertel der initialen Anschaffungskosten ausmachte, also etwa 2,58 Millionen Dollar. Diese gliederten sich folgendermaßen: Die Kosten für die für Support und Wartung zuständigen Mitarbeiter schlugen sich mit rund der Hälfte (52 Prozent) dieser Summe zu Buche. 29 Prozent fielen auf jährliche Lizenzgebühren und Supportverträge, 9 Prozent auf Abschreibungen für die Hardware-Komponenten, 6 Prozent auf internes User-Training und 4 Prozent auf den Hardware-Support. Die laufenden softwarebezogenen Ausgaben der *Gonzaga University* machten etwa ein Fünftel der ursprünglichen Softwarekosten aus.⁴⁶⁶

Einen Überblick über die ERP-Einführungskosten sowie die Folgekosten aus den diskutierten Fallstudien verschafft Tabelle 22. Das geplante Projektbudget sowie die tatsächlichen Kosten sind jeweils in US-Dollar angegeben. Bei den einzelnen Kostenkategorien ist jeweils ihr Anteil an den Projektkosten bzw. an den geplanten Projektkosten angegeben. Für die Folgekosten findet sich in der Tabelle ausschließlich die Summe der jährlich anfallenden Betriebskosten.

Tabelle 22: Direkte Kosten von ERP-Systemen - Ergebnisse der betrachteten Fallstudien⁴⁶⁷

	Kontron	Crosfield	Mitek	Nokia	DMC	Gonzaga
Projektdauer (in Monate)	8	6	5	4,5/10 ⁴⁶⁸	6	24
Anzahl User	230	40	40	500/120 ⁴⁶⁹	55	k.A.
Projektbudget (Plan)	k.A.	688.000	1.500.000	k.A.	1.250.000	k.A.
Projektkosten (Ist)	1.199.000	k.A.	1.450.000	k.A.	1.825.000	3.500.000
Beratungsleistungen	52 % ⁴⁷⁰	63 %	76 %	46 %	49 %	k.A.
Software	35 %	8 %	14 %	29 %	14 %	k.A.
Hardware	13 %	7 %	10 %	18 %	12 %	k.A.
Schulung	k.A.	7 %	k.A.	7 %	5 %	k.A.
Betriebskosten (jährlich)	392.000 ⁴⁷¹	k.A.	k.A.	k.A.	k.A.	2.580.000

⁴⁶⁵ vgl. Becker/Fleisch/Österle (1997): S. 11

⁴⁶⁶ vgl. Powel/Barry (2005): S. 43

⁴⁶⁷ eigene Darstellung. Zum Zwecke der Vergleichbarkeit werden alle Angaben zu monetären Werten in US-Dollar angegeben. Die für diese Tabelle verwendeten Wechselkurse zwischen Deutscher Mark und US-Dollar (1:0,645488) sowie zwischen Niederländischen Gulden und US-Dollar (1:0,572997) wurden am 20.10.2006 unter <http://www.xe.com> abgefragt. Die Zahlenwerte in der Tabelle wurden jeweils auf die Tausend gerundet.

⁴⁶⁸ Das Einführungsprojekt bei *Nokia SWP* war von Anfang an zweiphasig angelegt. Die erste Phase konnte nach 4,5 Monaten abgeschlossen werden, die zweite Phase nach 10 Monaten.

⁴⁶⁹ *Nokia SWP* hat 500 namentliche, davon 120 ständige User.

⁴⁷⁰ Dieser Anteil umfasst sowohl Beratungsleistungen wie auch Schulungsmaßnahmen.

⁴⁷¹ Diese Summe resultiert aus dem Mittelwert der Investitionen über die zwei Folgejahre.

2.4 Potentiale zur Senkung der TCO bei ERP-Systemen

Bei Investitionen in ERP-Systeme handelt es sich um IT-Investitionen, die einen positiven Ertrag erwirtschaften sollen.⁴⁷² Einige Autoren weisen ganz deutlich darauf hin, dass niedrigere TCO oft nicht die besseren sind; so meint etwa *Liebmann*: „Underlying most TCO discussions ist the idea that lower TCO is good.“⁴⁷³ Das Ziel der Senkung der Kosten von einer ERP-Softwarelösung kann also nur eines von vielen Zielen im Rahmen des ERP-Einsatzes darstellen. Warum dieses Ziel an dieser Stelle ein eigenes Unterkapitel verdient, hängt mit der Beschäftigung mit dem Thema TCO von ERP-Systemen zusammen und nicht damit, dass es einen besonderen Stellenwert einnehmen würde. Im Folgenden soll auf zwei zentrale Punkte für das Management der ERP-Gesamtkosten eingegangen werden - auf eine Best-Practice-geleitete ERP-Einführung zum einen und auf ein effizientes Lizenzmanagement zum anderen.

Eine Best-Practice-geleitete Einführung von betriebswirtschaftlicher Standardsoftware empfiehlt sich, um Zeit und auch Kosten zu sparen und die Qualität der Implementierung zu erhöhen. Die technischen und betriebswirtschaftlichen Aspekte von Implementierungserfahrungen dienen einer einfacheren Konfiguration von Anwendungen. Die TCO einer Best-Practice-basierten Lösung liegen - über drei Jahre betrachtet - durchschnittlich zehn Prozent unter denen einer Softwarelösung, die ohne Vorkonfiguration implementiert wurde.⁴⁷⁴

Eine Maßnahme, mit der bei den Projektkosten auf jeden Fall Ausgaben eingespart werden können, ist die möglichst genaue Ermittlung der Zahl der User und der Benutzungsintensität der verschiedenen Anwendergruppen.⁴⁷⁵ Auf Grundlage dieser Daten kann entschieden werden, wie viele ständige User das Unternehmen wirklich benötigt.

Den Sparstift bei den Schulungs- und Ausbildungskosten anzusetzen, mag verlockend sein; doch letztlich ist ein ERP-System nur so gut wie die End-User, die das System bedienen. Für Unternehmen ist es deshalb wichtiger, sich darüber klar zu werden, wie man den Nutzen eines ERP-Systems absichern kann, anstatt eine Senkung der TCO um jeden Preis anzustreben. An vorderster Front bei der Nutzenrealisierung von ERP-Systemen stehen hierbei Aus- und Weiterbildung der End-User sowie eine aktive Bereitschaft des Managements zur Veränderung. In der Praxis übersteigen die Lizenz- und Beratungskosten die Kosten für die Qualifizierung der Mitarbeiter und ein nachhaltiges Change Management deutlich. Doch Investitionen in diesen Bereichen tragen wesentlich zum Projekterfolg bei.⁴⁷⁶ Es werden nicht nur Kompetenz und Motivation der Mitarbeiter gefördert; End-User-Training hilft bei der Reduktion von Fehlern im laufenden Systembetrieb und somit von fehlerbedingten Kosten.⁴⁷⁷ Andernorts wird sogar geraten, dass Anwenderunternehmen die Einstellung der Mitarbeiter über den gesamten Projektzeitraum beobachten sollten, um eine erfolgreiche Einführung zu ermöglichen. Je früher

⁴⁷² vgl. Dibbern/Günther/Teltzrow (2005): S. 17

⁴⁷³ Liebmann (1999): S. 92

⁴⁷⁴ vgl. Viehmann (2005): S. 18. Zur Einführungsmethode *AcceleratedSAP* siehe auch Dolmetsch et al. (1998).

⁴⁷⁵ vgl. Scherer (2005): S. 17

⁴⁷⁶ vgl. Scherer (2005): S. 18

⁴⁷⁷ vgl. Eggert/Schmid (2006): S. 52; Scherer (2006): S. 8

Anwenderunternehmen einen Widerstand der Mitarbeiter gegen das neu einzuführende ERP-System aufspüren, desto eher können die Einstellungen und Haltungen der Mitarbeiter zum ERP-System noch positiv verändert werden.⁴⁷⁸ Denn es zeigt sich, dass eine negative Einstellung zum Einführungsprojekt für den Projekterfolg hinderlich ist: „Employee attitudes are a key factor in determining ERP implementation success or failure.“⁴⁷⁹ Diese Erkenntnis unterstützt wiederum die These, dass die Endanwender der wohl bedeutsamste Faktor für den Erfolg eines ERP-Projekts sind.⁴⁸⁰

3 Usability von ERP-Systemen und der Effekt auf die Total Cost of Ownership

Eine Verbesserung der Usability von ERP-Systemen bietet nicht nur individuelle Vorteile für die Endanwender, sondern hauptsächlich für das Unternehmen als Ganzes, das beispielsweise von einer verkürzten Ausbildungszeit der Anwender und von einer erhöhten Zufriedenheit der Mitarbeiter mit ihrer Arbeit profitiert.⁴⁸¹ Das wurde bereits an mehreren Stellen dieser Arbeit diskutiert. Nun soll die Auseinandersetzung mit der Usability von ERP-Systemen um weitere Aspekte bereichert werden.

3.1 Theoretische Grundlagen

Dass die Usability eine Auswirkung auf die Implementierungs- und Erhaltungskosten eines betrieblichen Informationssystems hat, ist keine bloße Mutmaßung. Ein Zusammenhang zwischen Usability und finanziellen Einsparungspotentialen wurde bereits Ende der 1980er Jahre im Rahmen von Usability Cost-Benefit Modells festgehalten. Die grundsätzliche Annahme dieser Modelle ist, dass gebrauchstaugliche User Interfaces zu konkreten und messbaren Nutzen führen; etwa dass die End-User Produktivität beachtlich steigt, wenn die Bearbeitungszeit für die Dateneingabe in eine Maske um eine Sekunde gesenkt werden kann, oder die Schulungskosten reduziert werden können, wenn die durchschnittliche Erlernzeit um zehn Trainingsstunden eingeschränkt werden kann.⁴⁸² Mit Hilfe der Modelle sollte der Einsatz von Methoden des User Centred Design sowie des Usability Testing begründet werden. Indem sie diverse Nutzenkategorien des Usability Engineering abschätzen und anschließend den Gesamtnutzen den Kosten gegenüberstellen, versuchen sie, den Nettonutzen zu quantifizieren.⁴⁸³ Es handelt sich bei den diversen Nutzenkategorien zwar um Schätzungen; diese basieren jedoch auf Erkenntnissen aus wissenschaftlichen Forschungsarbeiten, welche messbare und signifikante Vorteile bei Gestaltungsalternativen des User Interface nachgewiesen haben und somit einen Eindruck von der Bedeutung der Performanceunterschiede zwischen

⁴⁷⁸ vgl. Abdinnour-Helm/Lengnick-Hall/Lengnick-Hall (2003): S. 271

⁴⁷⁹ Abdinnour-Helm/Lengnick-Hall/Lengnick-Hall (2003): S. 271

⁴⁸⁰ vgl. Hare (1999) nach Wu et al. (2002): S. 3

⁴⁸¹ vgl. Babaian/Lucas/Topi (2004): S. 168

⁴⁸² vgl. Mayhew/Mantei (1994): S. 28

⁴⁸³ Usability Cost-Benefit Modells stellen neben den Nutzenkategorien für Anwenderunternehmen auch jene für Softwarehersteller dar. Die nachfolgenden kurzen Beschreibungen der einzelnen Modelle beschränken sich auf erstere.

optimaler und suboptimaler Interfacegestaltung geben.⁴⁸⁴ Einfache Analysetechniken zur Bewertung des Nettonutzens sind das Kosten-Nutzen-Verhältnis zum einen und die Bestimmung der Payback Period zum anderen; die Berechnung von Net Present Value und Internal Rate of Return stellen ausführlichere Techniken dar.⁴⁸⁵

Bereits 1988 erkannten *Mantei* und *Teorey*, dass Softwareentwicklungsmethoden, die softwareergonomische Faktoren berücksichtigen, das Produkt verbessern, seine Akzeptanz erhöhen und überdies den Wartungsaufwand verringern.⁴⁸⁶ Sie identifizierten eine Reduktion der Erlernzeiten und der Benutzungsfehler als direkte Nutzenkategorien⁴⁸⁷ eines Usability-Fokus für Anwenderunternehmen.⁴⁸⁸ *Mantei* und *Teorey* wiesen außerdem darauf hin, dass der Nettonutzen durch Usability Engineering mit der Population der User steigt, und empfahlen deshalb speziell in Entwicklungsprojekten für Software, die von vielen Benutzern angewandt werden sollte, entsprechende Methoden einzusetzen.⁴⁸⁹ Zur Effizienz von Usability Engineering in Abhängigkeit von der Anzahl der Benutzer siehe Abbildung 22.

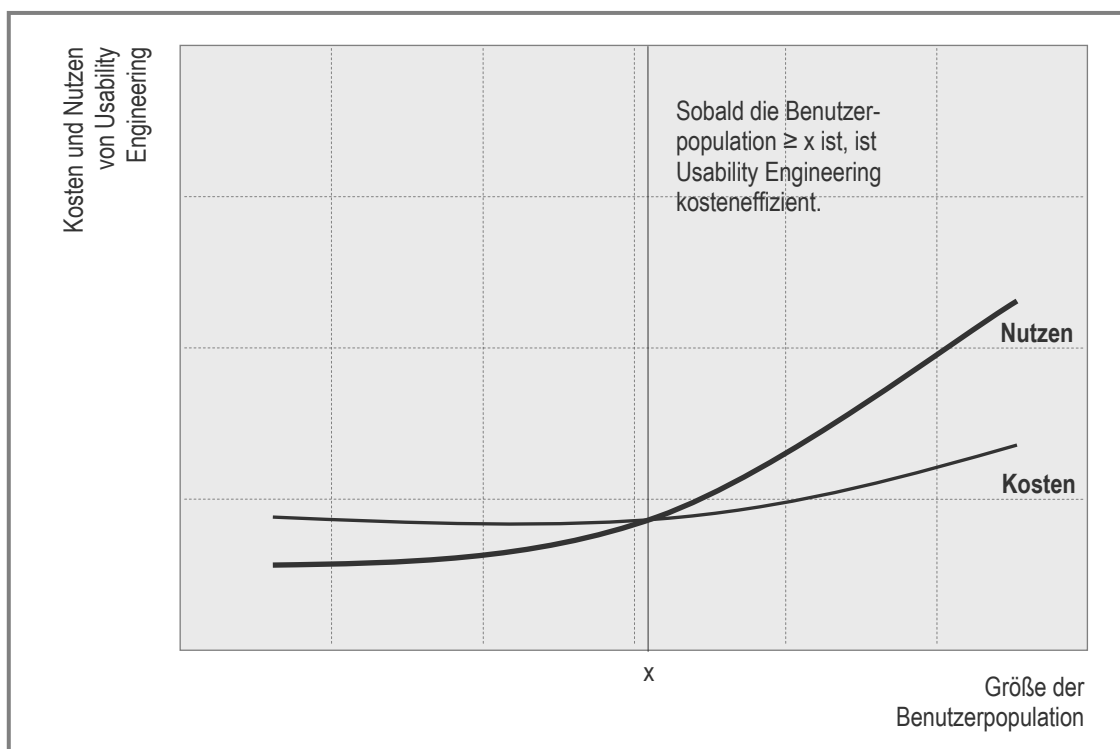


Abbildung 22: Effizienz von Usability Engineering im Verhältnis zur Benutzerpopulation⁴⁹⁰

⁴⁸⁴ vgl. Mayhew/Mantei (1994): S. 28ff

⁴⁸⁵ vgl. Karat (1994): S. 58ff

⁴⁸⁶ vgl. Mantei/Teorey (1988): S. 428

⁴⁸⁷ Da die beiden Autoren den Nutzen im Kontext einer unternehmensinternen Softwareentwicklungsabteilung betrachteten, stellten sie zusätzlich Einsparungspotentiale aufgrund einer frühen Fehlererkennung fest. Da diese Kategorie bei ERP-Systemen - diese werden nicht unternehmensintern entwickelt - jedoch keine Relevanz hat, bleibt sie in der Folge unbeachtet.

⁴⁸⁸ vgl. Mantei/Teorey (1988): S. 434

⁴⁸⁹ vgl. Mantei/Teorey (1988): S. 437f

⁴⁹⁰ modifiziert nach Mantei/Teorey (1988): S. 438

Das breiteste Spektrum an Usability Cost-Benefit Models enthält das 1994 von *Bias* und *Mayhew* editierte Buch „*Cost-Justifying Usability*“. *Mayhew* und *Mantei*, *Karat* sowie *Ehrlich* und *Rohn* stellten in ihren Beiträgen sowohl Kosten und Nutzen der Software Usability dar. Usability Engineering bei der Entwicklung von Softwaresystemen zu beachten, führe laut *Ehrlich* und *Rohn* zu einer Win-Win Situation für die Softwarehersteller und auch ihre Kunden. Die Autoren beschränkten sich in der Folge jedoch auf die Darstellung der Usability-Nutzen für Softwarehersteller. Auf eine Nutzenkategorie soll jedoch auch an dieser Stelle hingewiesen werden: Jene der verringerten Supportkosten für Herstellerfirmen hat nämlich auch Implikationen für Anwenderunternehmen.⁴⁹¹ Anrufe bei Customer Support Hotlines sind auch für sie mit Kosten verbunden; die Mitarbeiter der Anwenderunternehmen sind während der Zeitdauer des Telefonats unproduktiv - und oft schon lange zuvor, denn ein Anruf beim Hersteller ist gewöhnlich nicht der erste Schritt beim Versuch, ein Benutzungsproblem zu lösen.

Eine erhöhte Produktivität der End-User, eine verringerte Fehlerquote bei der Benutzung der Software, verringerte Schulungs- und Ausbildungskosten sowie ein reduzierter Supportaufwand wurden von *Mayhew* und *Mantei* als Nutzenkategorien⁴⁹² identifiziert.⁴⁹³

Karat näherte sich den Vorteilen von Software Usability, indem sie einige bekannte Nutzenkategorien mit Ergebnissen aus Fallstudien untermauerte. Konkrete Beispiele brachte *Karat* aber lediglich für zwei der für Anwenderunternehmen relevanten Nutzen, nämlich für eine gesteigerte Anwenderproduktivität durch eine verkürzte Bearbeitungszeit von Aufgaben im System und für reduzierte Personalkosten infolge einer verringerten Mitarbeiterfluktuation.⁴⁹⁴ Darüber hinaus berichtete die Autorin von gesunkenen Schulungs- und Supportkosten, geringeren Wartungskosten und einer verringerten Anzahl an Fehlern bei der Benutzung. *Karat* hielt auch fest, dass die finanziellen Vorteile für Anwenderunternehmen durch eine Kombination diverser Faktoren entstehen: Eine verbesserte Usability und eine erhöhte Benutzerzufriedenheit würden etwa die Mitarbeiterfluktuation einschränken.⁴⁹⁵

Einen weiteren bedeutenden Ansatz zur Bestimmung des Kosten-Nutzen-Verhältnisses von Usability Engineering findet man bei *Bevan*. In seinem 2000 veröffentlichten Bericht zählte er eine Reihe potentieller Nutzenkategorien von gebrauchstauglicher Software auf: Eine Software, welche es dem Benutzer ermöglicht, sich auf seine Aufgaben zu konzentrieren anstatt sich auf das Interface an sich konzentrieren zu müssen, steigere die Produktivität des Benutzers. Zudem seien Benutzungsfehler oft Resultat von schlecht gestalteten User Interfaces; Benutzerbedürfnisse zu beachten, führe demnach zu einer geringeren Anzahl an Benutzungsfehlern. Der durch Usability reduzierte Schulungsbedarf bewirke in der Folge eine Senkung der Schulungskosten. Außerdem führe Usability zu einer erhöhten Akzeptanz der Benutzer

⁴⁹¹ vgl. Ehrlich/Rohn (1994): S. 95

⁴⁹² Eine weitere Nutzenkategorie stellen wiederum Einsparungen aufgrund früher Änderungen in der Softwaredesignphase dar; sie werden in der Folge ebenso ignoriert.

⁴⁹³ vgl. Mayhew/Mantei (1994): S. 20

⁴⁹⁴ vgl. Karat (1994): S. 54ff

⁴⁹⁵ vgl. Karat (1994): S. 49f

gegenüber der Software. Dies hätte vor allem dann spürbare Auswirkungen, wenn der Gebrauch der Software nicht zwingend nötig ist, etwa wenn es sich um Aufgaben handelt, die nicht notwendigerweise mit Hilfe der Software erledigt werden müssen; im Fall von Software mit schlechter Usability vermieden die Benutzer, Aufgaben mit der Software zu bearbeiten. Überdies hätte Usability eine positive Auswirkung auf die Zufriedenheit der Mitarbeiter mit ihrer Arbeitssituation und folglich auf die Mitarbeiterfluktuation, die sich wiederum in niedrigeren Personalkosten niederschlägt. Ebenso stellte *Bevan* reduzierte Supportkosten als Nutzen eines Usability-Fokus fest, etwa weil weniger Benutzeranliegen von den Mitarbeitern des Service Desk bearbeitet werden müssen.⁴⁹⁶

Tabelle 23 vergleicht die soeben erörterten Usability Cost-Benefit Models hinsichtlich der in ihnen aufgezeigten Nutzenkategorien von Usability Engineering für Anwenderunternehmen. Das Modell von *Ehrlich* und *Rohn* wird aufgrund der Fokussierung auf die Usability-Nutzen für Softwarehersteller vernachlässigt.

Tabelle 23: Usability Cost-Benefit Models: Nutzen der Usability für Anwenderunternehmen⁴⁹⁷

		Mantei & Teorey	Mayhew & Mantei	Karat	Bevan
Einführung	Verringerte Kosten des Support	-	-	•	-
	Verringerte Kosten des End-User Training	••	••	•	•
Laufender Betrieb	Gesteigerte Produktivität	-	••	••	•
	Geringere Fehlerrate	••	••	•	•
	Erhöhte Akzeptanz	-	-	-	•
	Verringerte Mitarbeiterfluktuation	-	-	••	•
	Verringerter Aufwand des End-User Support	-	•	-	•
•• Identifizierung der Nutzenkategorie und Darstellung anhand von konkreten Leitlinien und Beispielen. • Identifizierung der Nutzenkategorie, jedoch keine konkreten Leitlinien und Beispiele. - Keine Identifizierung der Nutzenkategorie.					

⁴⁹⁶ vgl. Bevan (2000), online

⁴⁹⁷ eigene Darstellung

Keinen unmittelbaren Anhaltspunkt für die Unterstützung der These, dass Usability zu Kosteneinsparungen für Anwenderunternehmen führe, gaben *DeLone* und *McLean* in ihrem viel zitierten „*DeLone and McLean Model of Information Systems Success*“⁴⁹⁸, im Folgenden kurz IS Success Model genannt. Das Modell legt jedoch einen Zusammenhang zwischen der Usability und dem Nettonutzen eines Informationssystems nahe: Geringere Gesamtkosten des Einsatzes eines Informationssystems wirken sich unter der Bedingung eines konstanten Nutzenniveaus positiv auf den Nettonutzen⁴⁹⁹ aus.

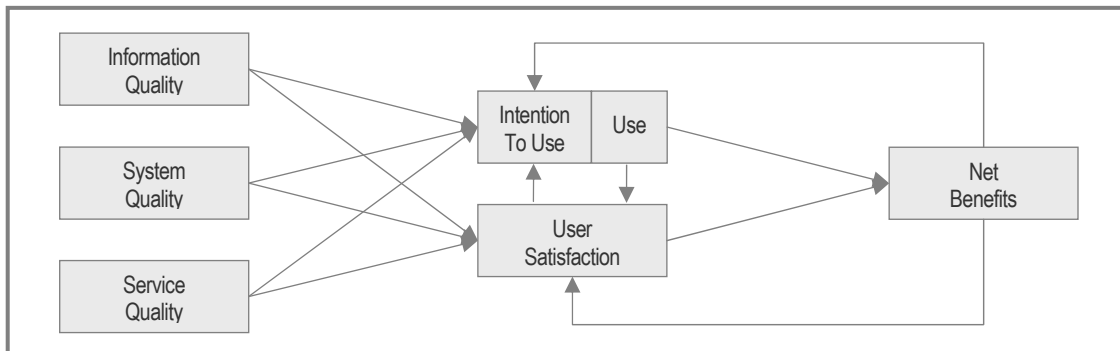


Abbildung 23: Information Systems Success Model nach DeLone und McLean⁵⁰⁰

Das IS Success Model (siehe Abbildung 23) umfasst drei verschiedene Qualitätskategorien zur Beschreibung eines Informationssystems - Information Quality, System Quality und Service Quality. Hinter allen dreien verstecken sich verschiedene Variablen; die Usability einer Software bildet etwa einen Aspekt der System Quality. Alle drei Qualitätskonstrukte haben sowohl Auswirkungen auf die Absicht der End-User, die Software auch tatsächlich zu benutzen, wie auch auf die Benutzerzufriedenheit. Diese wirkt sich wiederum auf die Benutzungsabsicht und die Benutzungsabsicht in der Folge auf die tatsächliche Benutzung aus. Die tatsächliche Benutzung und die Zufriedenheit der End-User haben schließlich einen Einfluss auf die Net Benefits des Informationssystems. Ein hoch qualitatives Softwaresystem - so das Modell - führt zu einem erhöhten tatsächlichen Gebrauch und zu einer gesteigerten Benutzerzufriedenheit, was wiederum einen positiven Nettonutzen begünstigt.⁵⁰¹ Auch die Software Usability als ein Element der System Quality⁵⁰² hat somit einen Einfluss auf die Net Benefits eines Informationssystems: Unter der Annahme, dass zwei Informationssysteme verglichen werden, bei denen alle Faktoren mit Ausnahme der Usability identisch sind, wird jenes den höheren Nettonutzen erzielen, das eine höhere Usability aufweist. Einerseits wird die höhere Usability für vergleichsweise

⁴⁹⁸ Von diesem Modell berichteten die beiden Autoren erstmals 1992 in ihrem Artikel „*Information Systems Success: The Quest for the Dependent Variable*“. Das Modell wurde in der Folge in rund 300 wissenschaftlichen Artikeln aufgegriffen und in diesen teilweise bestätigt, mitunter aber auch heftig kritisiert und weiterentwickelt; siehe dazu insbesondere *Seddon*. 2003 erschien schließlich der Artikel „*The DeLone and McLean Model of Information Systems Success: A Ten-Year Update*“, in welchem *DeLone* und *McLean* ein um einige Aspekte erweitertes Modell vorstellten. Vgl. *DeLone/McLean* (1992); *Seddon* (1997); *DeLone/McLean* (2003)

⁴⁹⁹ Der Nettonutzen drückt die Differenz zwischen Nutzen und Kosten des Informationssystems aus.

⁵⁰⁰ modifiziert nach *DeLone/McLean* (2003): S. 24

⁵⁰¹ vgl. *DeLone/McLean* (2003): S. 23f

⁵⁰² vgl. *DeLone/McLean* (2003): S. 26

niedrigere Gesamtkosten des Informationssystems verantwortlich sein, andererseits kann sie sich auch in einem höheren Nutzen für das Anwenderunternehmen niederschlagen. Mit diesem Beispiel sei veranschaulicht, dass das IS Success Model - obwohl es keine Aussage über eine explizite Beziehung zwischen Usability und Kosten einer Software trifft - eine Wirkung von Faktoren wie etwa der Usability auf den Nettonutzen, hinter dem sich gewissermaßen auch die Kostenseite einer IT-Infrastruktur versteckt, nachweist.

3.2 Konkrete Auswirkungen der ERP-Usability auf die TCO

Auf die Auswirkungen der Usability von ERP-Software auf ihre Gesamtkosten wurde von vielen Autoren hingewiesen. *Chew, Orlov* und *Herbert*, Analysten von *Forrester Research*, gaben nach einem Usability Test von elf ERP-Softwaresystemen an, dass nicht-intuitive User Interfaces von ERP-Systemen zu einer verminderten Produktivität der End-User und somit zu erhöhten Kosten für die Anwenderunternehmen führen.⁵⁰³ *Topi, Lucas* und *Babaian* kamen in ihrer Fallstudie⁵⁰⁴ zum Schluss, dass eine hohe ERP-Usability die Frustration auf Seiten der Endanwender hemmt und weniger frustrierte End-User mit einem besseren Verständnis der Benutzung des Systems dem Unternehmen Zeit und Geld sparen. Das Anwenderunternehmen kann in der Folge von geringeren Schulungskosten, einer raschen Implementierung - ein gebrauchstaugliches System ermöglicht ein vergleichsweise früheres effektives Arbeiten - und von einer intensiveren Benutzung des Systems profitieren. Die intensivere Benutzung ist darin begründet, dass die Software für alle Aufgaben, die sie lösen sollte, eingesetzt wird; auch eher selten anfallende Prozesse werden bei einer ERP-Softwarelösung mit hoher Usability tendenziell mit Hilfe der Software gelöst.⁵⁰⁵ *Cooper* widmete sich ebenso der Frustration der End-User, die schließlich dazu führe, dass die Endanwender mit ihrer Arbeit nicht länger zufrieden sind. Das wiederum wirke sich auf ihre Produktivität aus, auf die Fehlerhäufigkeit und schließlich auf die Mitarbeiterfluktuation. Mitarbeiter zu verlieren, ist für Anwenderunternehmen teuer: Denn es ziehe meist eine Unterbrechung für das Unternehmen mit sich; die verlorene Zeit könne niemals aufgeholt werden.⁵⁰⁶ Auf den Vorteil von geringen Erlernzeiten wurde auch andernorts hingewiesen: Die Schulungsmaßnahmen sind immerhin für 10 bis 15 Prozent der Implementierungskosten einer ERP-Software verantwortlich - je ausgereifter die Usability, desto geringer der Aufwand für das End-User-Training.⁵⁰⁷ *Ullerich* wies auf Probleme bei einem Wechsel des Releasestands oder beim Einspielen von Updates hin, die in unproduktiven Wartezeiten der End-User resultieren können.⁵⁰⁸ Dass auch das Design der ERP-Softwarehandbücher direkte Auswirkungen auf die Kosten einer ERP-Implementierung hat, bewies *Scott*.⁵⁰⁹

⁵⁰³ vgl. *Chew/Orlov/Herbert* (2003) nach *Babaian/Lucas/Topi* (2004): S. 164; *Gilbert* (2003), online

⁵⁰⁴ Eine Auseinandersetzung mit dieser Fallstudie findet sich im Unterkapitel IV 1.1 Überblick über die bisherige Forschung.

⁵⁰⁵ vgl. *Topi/Lucas/Babaian* (2005): S. 133

⁵⁰⁶ vgl. *Cooper* (2004): S. 40

⁵⁰⁷ vgl. o.V. (2004): S. 9

⁵⁰⁸ vgl. *Ullerich* (2004): S. 4

⁵⁰⁹ vgl. *Scott* (2005): S. 75

Eine Übersicht der angesprochenen Nutzenkategorien für Anwenderunternehmen durch eine hohe ERP-Usability ist in Abbildung 24 dargestellt. Die Pfeile zwischen den einzelnen Kategorien verdeutlichen die beobachteten Interdependenzen zwischen den verschiedenen Nutzen.

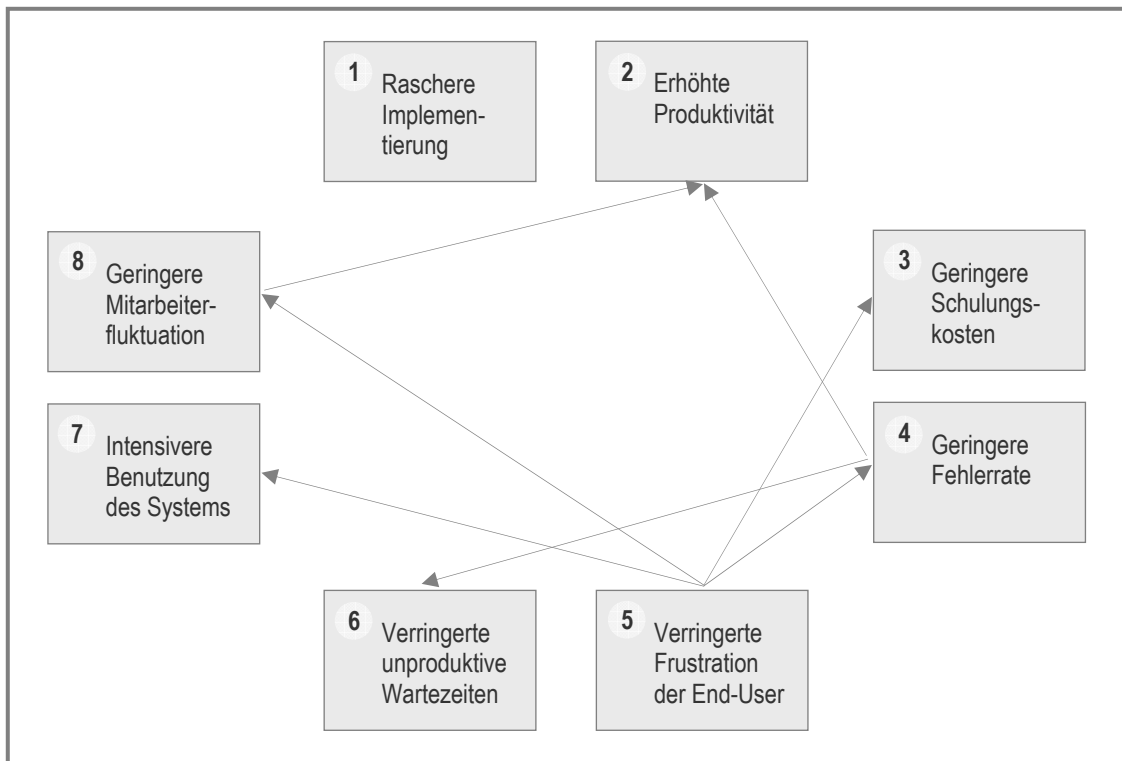


Abbildung 24: Nutzen der ERP-Usability für Anwenderunternehmen⁵¹⁰

Inwiefern sich die skizzierten Nutzenkategorien infolge einer hohen ERP-Usability auf die Gesamtkosten einer ERP-Software auswirken, gilt es nun zu überprüfen. Tabelle 24 gibt das im Unterkapitel IV 2.2 TCO-Modell für ERP-Systeme entwickelte Gesamtkostenmodell wieder. Neben den Kostenkategorien findet sich eine Einschätzung, inwieweit die Usability einen Einfluss auf den jeweiligen Kostenfaktor ausübt.

Für die Anschaffungskosten für Hardware-Komponenten sowie für die Softwarelizenzen spielt die Usability offensichtlich keine Rolle. Ebenso wenig für das Projektmanagement, die Projektvorbereitung, den technischen Setup, das Testing des Produktivsystems sowie eine eventuelle Neugestaltung der Geschäftsprozesse im Rahmen der Implementierung. Für den Business Setup hingegen ist die Usability bereits ein entscheidender Faktor, wenn es darum geht, wie viel Aufwand und somit Kosten etwa der Datenimport in das neue ERP-System oder auch das User Management mit sich bringt. Genauso wichtig ist ein gebrauchstaugliches ERP-System in der Realisierungsphase, wenn Schnittstellen an das System angebunden werden sollten. Ferner kann davon ausgegangen werden, dass der Aufwand und somit auch die Kosten des Go-Live-Support mit steigender Usability sinken.

⁵¹⁰ eigene Darstellung

Tabelle 24: Auswirkung der Usability auf die TCO einer ERP-Software⁵¹¹

Hardware	Anschaffungskosten	Kosten der Anschaffung von zusätzlich notwendiger Hardware für den Betrieb des ERP-Systems	-	
	Software	Systemlizenzen	Kosten für Anwenderlizenzen	-
		Datenbanklizenzen	Kosten für Datenbanklizenzen	-
Implementation	Projektmanagement	Kosten für das Projektmanagement	-	
	Projektvorbereitung	Kosten für Planung und Konzeption der Einführung	-	
	Technischer Setup	Kosten für Installation und Konfiguration	-	
	Business Setup	Kosten für die Definition der Geschäftsprozesse, für Add-ons, für Datenmigration, User Management etc.	●	
	Realisierung	Kosten für die Anbindung von Schnittstellen	●	
	Testing	Kosten des Testing und des Aufbaus eines Produktivsystems	-	
	Go-live-Support	Kosten für den Support beim Go-Live	●	
	Business Process Reengineering	Kosten für Aktivitäten des Business Process Reengineering	-	
	Training	Key-User Schulung	Kosten für die Ausbildung der Key-User	●
End-User Schulung		Kosten für die Planung und Durchführung der Endanwenderschulung	●	
Operations	Wartung	Kosten für die Wartung der Hardware und Pflege der Software	●	
	End-User Support	Kosten für den Support der Endanwender, z.B. Personalkosten für Mitarbeiter des Service Desk	●	
End-User Operations	Self- und Peer-to-Peer-Support	Kosten der Selbsthilfe sowie der informellen Hilfe von Kollege zu Kollege	●	
	Casual Learning	Kosten für bewusstes eigenständiges Lernen	●	
	Formales End-User Training	Kosten für den Besuch von ERP-Schulungen	●	
Ausfallzeiten	Geplante und ungeplante Ausfallzeiten	Kosten der Produktivitätsminderungen der Endbenutzer aufgrund der Wartezeit auf die Hilfe des Support oder aufgrund des Ausfalls eines Systems	●	
● Auswirkung der Usability auf spezifischen Kostenbereich zu erwarten. - Keine zu erwartende Auswirkung der Usability auf spezifischen Kostenbereich.				

Dass eine mangelnde Usability und der in der Folge größere Schulungsbedarf die Kosten für das Training in die Höhe treiben, wurde bereits gesagt. Dies gilt für den Schulungsbedarf für Key-User wie für End-User gleichermaßen. Ebenso bedeutend erweist sich die Usability für die

⁵¹¹ eigene Darstellung

beiden Bereiche innerhalb der Kostenkategorie Operations. Zum einen kann ein vergleichsweise geringer Wartungsaufwand bei ERP-Produkten mit passabler Usability erwartet werden; beispielsweise sollte sich das Einspielen von Patches und Updates einfach gestalten. Zum anderen sollte sich der Bedarf an End-User Support aufgrund einer geringeren Anzahl an Benutzungsproblemen in Grenzen halten. Das ist auch der Grund, warum die Kosten für Self- und Peer-to-Peer-Support bei gebrauchstauglicher ERP-Software tendenziell gering ausfallen. Anwenderunternehmen werden auch davon profitieren, dass Software mit hoher Usability ein effizientes Lernen ermöglicht; dadurch müssen die Mitarbeiter einen relativ kleinen Anteil ihrer Arbeitszeit für den Besuch von Schulungen sowie für das beiläufige selbständige Lernen aufwenden. Die bei gebrauchstauglicher Software zu erwartende geringere Fehlerrate führt zu großen Einsparungspotentialen im Bereich der Ausfallzeiten.

Jeden der an dieser Stelle aufgezeigten Effekte der Usability auf die Gesamtkosten einer ERP-Software findet man in der Literatur für sich belegt. Eine der Ausgangsfragen, nämlich jene zu den Auswirkungen der Usability auf die ERP-Gesamtkosten, kann somit beantwortet werden: Es zeigt sich deutlich, dass die Usability einen entscheidenden Einfluss auf wichtige Kostenkategorien der TCO eines ERP-Systems hat. Eine hohe ERP-Usability führt zu verringerten Kosten der Inbetriebnahme und der Nutzung von betriebswirtschaftlicher Standardsoftware. Anwenderunternehmen sparen durch eine gebrauchstaugliche ERP-Software, welche auch die Benutzer zufrieden stellt, bares Geld.

3.3 Relevanz der ERP-Usability in Anwenderunternehmen

Abschließend soll darauf eingegangen werden, warum nicht alle Anwenderunternehmen im selben Ausmaß von guter Usability bei ERP-Systemen profitieren. Diese Fragestellung sollte die Bedeutung der ERP-Usability jedoch nicht abschwächen. Im letzten Abschnitt wurde, indem konkrete Auswirkungen der ERP-Usability auf die Gesamtkosten betrachtet wurden, veranschaulicht, warum gebrauchstaugliche ERP-Software zu großen Einsparungspotentialen bei Anwenderunternehmen führt. Die genannten Vorteile gelten prinzipiell für jede Organisation, die eine ERP-Software einsetzt. Dennoch ist das Nutzensausmaß einer passablen Usability für die einzelnen ERP-Anwenderunternehmen verschieden. Grund dafür ist eine Reihe von Faktoren, die einen Einfluss auf die Relevanz der ERP-Usability nehmen. Auf zwei Faktoren - vielleicht sogar die beiden wesentlichsten - wird im Folgenden eingegangen.

Große Unternehmen mit einer funktionalen Organisationsstruktur und Mitarbeitern, die als End-User nur eine Hand voll Funktionalitäten in der ERP-Software bedienen müssen und sich in der Folge ein entsprechendes Wissen angeeignet haben, werden von einer hohen Usability nur eingeschränkt profitieren. Ebenso ungewiss ist, inwieweit die Endanwender eine verbesserte Usability - etwa hinsichtlich einer Prozessunterstützung - überhaupt schätzen würden. Mit der Zeit gewöhnen sie sich an weniger gebrauchstaugliche Aspekte der Software und sind nicht auf eine hohe Usability angewiesen, um mit dem Programm ihre Arbeit erledigen zu können. Anders ist die Situation für Mitarbeiter eines kleineren Betriebs, in dem die funktionale Trennung der Aufgaben aufgrund der Größe weit nicht so ausgeprägt ist. Ist ein Kollege erkrankt, ist es

durchaus üblich, dass ein anderer für ihn einspringt; gewisse Aufgaben müssen schließlich erledigt werden. Das hieße in der Folge, dass ein Mitarbeiter eines mittelständischen Betriebs sehr viele der vom Unternehmen eingesetzten Funktionalitäten beherrschen müsse, er also eine sehr umfassende ERP-spezifische Aus- und Weiterbildung benötigen würde.

Zudem ist mittlerweile eine von mittelständischen Unternehmen eingesetzte ERP-Software kaum weniger komplex als eine für große Unternehmen. Denn die Anforderungen von Mittelständlern weichen nicht wesentlich von jenen großer Konzerne ab: Auch mittelständische Betriebe müssen beispielsweise verschiedene Standorte mit einer ERP-Software verwalten - die Mehrmandantenfähigkeit des Softwaresystems ist dazu erforderlich. Sind die Firmenniederlassungen zudem in verschiedenen Ländern, was auch im Mittelstand häufig vorkommt, ist auch Internationalität gefragt. Die Bilanzierung hat nach voneinander abweichenden rechtlichen Grundlagen stattzufinden, ferner ist das Angebot von mehreren Sprachen im System wesentlich für einen Einsatz der ERP-Software in verschiedenen Ländern.

Die Komplexität des heute üblichen ERP-Systemstandards begründet auch, warum es für End-User in einem mittelständischen Unternehmen ohnehin schwierig ist, den überwiegenden Anteil aller Geschäftsvorfälle des Unternehmens mit Hilfe der Software bearbeiten zu können. Eine hohe ERP-Usability unterstützt jedoch die Endanwender und ermöglicht somit, dass diese bei konstantem Schulungsaufwand mehr Prozesse im System bearbeiten können.

Die Bedeutung der ERP-Usability für gelegentliche Nutzer bildet den zweiten der in diesem Abschnitt näher diskutierten Faktoren, die sich schließlich auf die Relevanz eines gebrauchstauglichen ERP-Systems in Anwenderunternehmen auswirken. Der Trend zur Dezentralisierung lässt die Bedeutung gelegentlicher Nutzer für Anwenderunternehmen anwachsen. Gelegentliche Nutzer profitieren sehr stark von einer hohen ERP-Usability, in der Folge profitieren auch ihre Arbeitgeber von einer leichter erlernbaren und intuitiveren Software - der Schulungsaufwand sinkt, daneben sinkt auch die Häufigkeit von Benutzungsproblemen mit all ihre möglichen Auswirkungen (Peer-to-Peer-Support, Ausfallzeit aufgrund Supportanfrage etc.).

Über den rasanten Bedeutungszuwachs von gelegentlichen Nutzern klärt Abbildung 25 auf: Dargestellt sind die Ergebnisse einer von *Forrester Research* durchgeführten Untersuchung; die Frage lautete, wie wichtig die Unterstützung der gelegentlichen Nutzer für die Firma sei. Es sind die Ergebnisse für gelegentliche interne Nutzer dargestellt; jene für gelegentliche externe Nutzer weichen kaum ab.⁵¹²

⁵¹² vgl. Plattner (1999): S. 82f

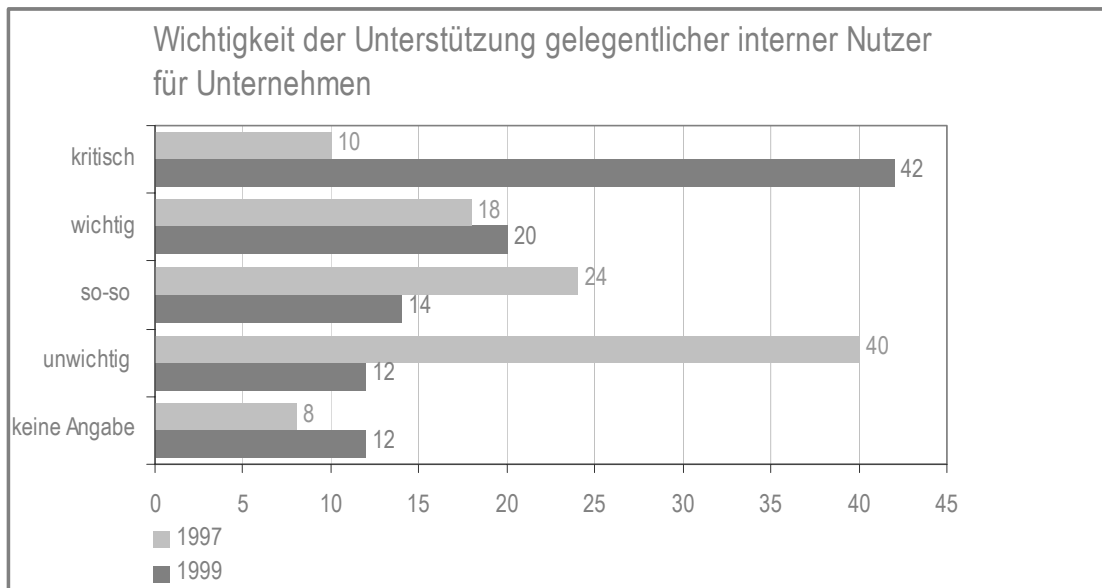


Abbildung 25: Wichtigkeit der Unterstützung gelegentlicher interner Nutzer für Unternehmen⁵¹³

Neben den zwei erörterten Aspekten existieren noch weitere beeinflussende Faktoren. So wirkt sich beispielsweise auch die Mitarbeiterfluktuation auf die Relevanz der ERP-Usability bei einem spezifischen Anwenderunternehmen aus: Bei einer durchschnittlich hohen Fluktuation ist die ERP-Usability umso wichtiger, um Schulungskosten einzuschränken. Ebenso ist die Anzahl der eingesetzten Module ein ausschlaggebender Faktor: Je mehr Module im Einsatz sind, desto wichtiger ist die Usability.

⁵¹³ eigene Darstellung, Daten entnommen aus Forrester Research (o.J.) nach Plattner (1999): S. 83

V ZUSAMMENFASSUNG UND AUSBLICK

Der Einfluss der Usability auf die Kosten von Informationssystemen wird bereits seit Ende der 1980er Jahre in der Usability-Forschung, etwa im Rahmen so genannter Usability Cost-Benefit Models, thematisiert. Die Erkenntnisse scheinen sich aber - zumindest was ERP-Systeme angeht - nicht ausreichend durchgesetzt zu haben. Die Literaturrecherche bestätigte die in der Einleitung gehegte Vermutung, dass es um die Usability von ERP-Systemen nicht gut steht. Die in dieser Arbeit diskutierte Literatur zur ERP-Usability trifft im Allgemeinen keine positiven Aussagen über die Gebrauchstauglichkeit von betriebswirtschaftlicher Standardsoftware. Vielmehr zeigt sie eine Reihe von akuten Benutzungsproblemen infolge mangelnder Usability auf. Als Reaktion auf die Probleme, die mit der Benutzung von ERP-Systemen im Zusammenhang stehen, wurde erörtert, wie konkrete Anforderungen an die Usability von betriebswirtschaftlicher Standardsoftware aussehen könnten. Es wäre für Anwenderunternehmen offensichtlich wünschenswert, wenn solchen Anforderungen beim Design von ERP-Systemen ausreichend Beachtung geschenkt werden würde.

Die Ergebnisse dieser Arbeit zeigen deutlich, dass sich eine hohe ERP-Usability positiv auf die Gesamtkosten der Implementierung sowie auf jene der Benutzung von ERP-Systemen auswirkt. Anwenderunternehmen können in der Folge von reduzierten Kosten des ERP-Einsatzes profitieren. Wie Unternehmen, die ERP-Systeme einsetzen, von einer hohen Usability dieser profitieren können, sei an dieser Stelle nochmals kurz skizziert: Eine hohe ERP-Usability lässt einen Nutzen für Anwenderunternehmen entstehen; dafür sind eine raschere Implementierung, eine gesteigerte Produktivität, reduzierte Schulungskosten, eine gesenkte Fehlerrate bei der Benutzung, eine verminderte Frustration unter den Endanwendern, weniger unproduktive Wartezeiten, eine intensivere Systembenutzung und eine verringerte Mitarbeiterfluktuation verantwortlich. Die diversen Nutzenkategorien wirken sich schließlich erheblich auf Teile der ERP-Gesamtkosten aus: Eine hohe ERP-Usability verringert die Kosten für Implementierung und Wartung des Systems, ebenso jene des End-User Support. Ferner ist der für Anwenderunternehmen positive Einfluss der Usability auf die ERP-Gesamtkosten vor allem in einer Reduktion der Endanwenderkosten begründet - die vom Mitarbeiter aufgewendete Zeit etwa für

Schulungsmaßnahmen oder auch für Peer-to-Peer-Support ist im Falle von gebrauchstauglicher ERP-Software beträchtlich eingeschränkt.

Im Zuge der Auseinandersetzung mit ERP-Systemen in den verschiedenen Lebenszyklusphasen wurde deutlich, dass die Forschung zum Einsatz von ERP-Systemen nach der Implementierungsphase sowie zu den Gesamtkosten der Benutzung eines ERP-Systems sehr dürftig ist. Ebenso wünschenswert wäre eine umfangreichere Forschung zum Thema ERP-Usability. Bislang liegt keine empirische Studie über das Ausmaß des Einflusses der ERP-Usability auf die Total Cost of Ownership vor. Wie das Design einer solchen Forschungsarbeit aussehen könnte, ist noch zu überprüfen: Eine experimentelle Studie, in der ein Unternehmen zwei oder mehrere ERP-Systeme einsetzt und Usability-Forscher und Controller gemeinsam den Einsatz beobachten und analysieren, inwieweit eine unterschiedliche Usability tatsächlich die ERP-Gesamtkosten beeinflusst, ist offensichtlich nicht möglich. Wäre solch eine experimentelle Anordnung möglich, würde sie wertvolle Ergebnisse liefern.

BIBLIOGRAFIE

Bücher, Zeitschriften und Zeitungen

- Abdinnour-Helm, Sue/Lengnick-Hall, Mark/Lengnick-Hall, Cynthia (2003): *Pre-implementation attitudes and organizational readiness for implementing an Enterprise Resource Planning system*. In: *European Journal of Operational Research* 2003. 146(2). S. 258-273.
- Adams, Dennis/Nelson, R. Ryan/Todd, Peter (1992): *Perceived Usefulness, Ease of Use and Usage of Information Technology: A Replication*. In: *MIS Quarterly* 1992. 16(2). S. 227-247.
- Allweyer, Thomas (2005): *Geschäftsprozessmanagement. Strategie, Entwurf, Implementierung, Controlling*. W3L-Verlag; Herdecke, Bochum; 2005.
- Au, Norman/Ngai, Eric/Cheng, Edwin (2002): *A critical review of end-user information system satisfaction research and a new research framework*. In: *Omega* 2002. 30(6). S. 451-478.
- Bernroider, Edward/Koch, Stefan (2000): *Entscheidungsfindung bei der Auswahl betriebswirtschaftlicher Standardsoftware - Ergebnisse einer empirischen Untersuchung in österreichischen Unternehmen*. In: *Wirtschaftsinformatik* 2000. 42(4). S. 329-338.
- Bevan, Nigel (1999): *Quality in use: Meeting user needs for quality*. In: *The Journal of Systems and Software* 1999. 49(1). S. 89-96.
- Bishu, Ram/Kleiner, Brian/Drury, Colin (2001): *Ergonomic Concerns in Enterprise Resource Planning (ERP) Systems and Its Implementations*. In: Mo, John/Nemes, Laszlo (Hrsg.): *Global engineering, manufacturing and enterprise networks*. Kluwer; Deventer; 2001. S. 146-155.
- Buckhout, Scott/Frey, Edward/Nemec, Joseph (1999): *Making ERP Succeed: Turning Fear into Promise*. In: *Journal of Strategy and Business* 1999. 17(2). S. 60-72.
- Bulkeley, William (1992): *Information Age: Study Finds Hidden Costs of Computing*. In: *The Wall Street Journal*. 02.11.1992. B4.
- Burmester, Michael/Hassenzahl, Marc/Koller, Franz (2002): *Usability ist nicht alles - Wege zu attraktiven Produkten*. In: *i-com* 2002. 1(1). S. 32-40.
- Buxmann, Peter/König, Wolfgang (1995): *Organisationsgestaltung bei der Einführung betrieblicher Standardsoftware*. In: *Management & Computer* 1995. 4(3). S. 161-168.
- Buxmann, Peter/König, Wolfgang (1997): *Empirische Ergebnisse zum Einsatz der betrieblichen Standardsoftware SAP R/3*. In: *Wirtschaftsinformatik* 1997. 39(4). S. 331-338.
- Changchit, Chuleepron/Joshi, Kshiti/Lederer, Albert (1998): *Process and reality in information systems benefit analysis*. In: *Information Systems Journal* 1998. 8(2). S. 145-162.
- Clark, Theodore/Stoddard, Donna (1996): *Interorganizational Business Process Redesign: Merging Technological and Process Innovation*. In: *Journal of Management Information Systems* 1996. 13(2). S. 9-28.

- Cooper, Alan (2004): *The Inmates Are Running The Asylum. Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Sams; Indianapolis; 2004.
- Davenport, Thomas (1998): *Putting the Enterprise into the Enterprise System*. In: *Harvard Business Review* 1998. 76(4). S. 121-131.
- Dempsey, Jed et al. (1998): *A Hard and Soft Look at IT Investments*. In: *The McKinsey Quarterly* 1998(1). S. 126-137.
- DeLone, William/McLean, Ephraim (1992): *Information system success: The quest for the dependent variable*. In: *Information Systems Research* 1992. 3(1). S. 60-95.
- DeLone, William/McLean, Ephraim (2003): *The DeLone and McLean Model of Information System Success: A Ten-Year Update*. In: *Journal of Management Information Systems* 2003. 19(4). S. 9-30.
- Deutsches Institut für Normung (2003): *Software-Ergonomie. Empfehlungen für die Programmierung und Auswahl von Software*. DIN Taschenbuch 354. Beuth; Berlin, Wien, Zürich; 2003.
- Dibbern, Peter/Günther, Oliver/Teltzrow, Maximilian (2005): *Produktivitätsmessung von ERP-Lösungen*. In: *ERP Management* 2005. 1(1). S. 17-20.
- Dix, Alan et al. (1993): *Human Computer Interaction*. Prentice Hall; New York u.a.; 1993.
- Doll, William/Torkzadeh, Gholamreza (1988): *The Measurement of End-User Computing Satisfaction*. In: *MIS Quarterly* 1988. 12(2). S. 259-274.
- Donahue, George (2001): *Usability and the Bottom Line*. In: *IEEE Software* 2001. 18(1). S. 31-37.
- Dray, Susan (1995): *The Importance of Designing Usable Systems*. In: *interactions* 1995. 2(1). S. 17-20.
- Dumas, Joseph/Redish, Janice (1999): *A Practical Guide to Usability Testing*. Intellect; Exeter, Portland; 1999.
- Duplaga, Edward/Astani, Marzie (2003): *Implementing ERP in Manufacturing*. In: *Information Systems Management* 2003. 20(3). S. 68-75.
- Eggert, Sandy/Schmid, Simone (2006): *Marktüberblick zum Support- und Schulungsspektrum von ERP-Anbietern*. In: *ERP Management* 2006. 2(2). S. 52-63.
- Ehrlich, Kate/Rohn, Janice (1994): *Cost Justification of Usability Engineering: A Vendor's Perspective*. In: *Bias, Randolph/Mayhew, Deborah (Hrsg.): Cost-Justifying Usability*. Academic Press; Boston u.a.; 1994. S. 73-110.
- Ellram, Lisa (1995): *Total cost of ownership. An analysis approach for purchasing*. In: *International Journal of Physical Distribution & Logistics* 1995. 25(8). S. 4-23.
- Ellram, Lisa/Siferd, Sue (1993): *Purchasing: The Cornerstone of the Total Cost of Ownership Concept*. In: *Journal of Business Logistics* 1993. 14(1). S. 163-184.
- Englisch, Joachim (1993): *Ergonomie von Softwareprodukten. Methodische Entwicklung von Evaluationsverfahren*. BI-Wissenschaftsverlag; Mannheim u.a.; 1993.
- Esteves, José/Pastor, Joan (2001): *Enterprise resources planning research: An annotated bibliography*. In: *Communications of the Association for Information Systems* 2001. 7(8). S. 1-52.
- Fink, Andreas/Schneiderei, Gabriele/Voß, Stefan (2005): *Grundlagen der Wirtschaftsinformatik*. Physica-Verlag; Heidelberg; 2005.
- Francalanci, Chiara (2001): *Predicting the implementation effort of ERP projects: empirical evidence on SAP/R3*. In: *Journal of Information Technology* 2001. 16(1). S. 33-48.
- Friedl, Gunther/Hilz, Christian/Pedell, Burkhard (2002): *Integriertes Controlling mit SAP-Software*. In: *Kostenrechnungspraxis* 2002. 46(3). S. 161-169.

- Gadatsch, Andreas (2003): *Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker*. Vieweg; Wiesbaden; 2003.
- Gadatsch, Andreas (2006): *IT-Controlling - operative und strategische Werkzeuge nutzen*. In: Tiemeyer, Ernst (Hrsg.): *Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis*. Hanser; München; 2006. S. 359-402.
- Gadatsch, Andreas/Mayer, Elmar (2004): *Grundkurs IT-Controlling. Grundlagen, Strategischer Stellenwert, Kosten- und Leistungsrechnung in der Praxis*. Vieweg; Wiesbaden; 2004.
- Gartenberg, Michael (2000): *Myths behind TCO*. In: *Computerworld*. 30.10.2000. 34(44). S. 52.
- Geis, Thomas/Hartwig, Ronald (1998): *Auf die Finger geschaut. Neue ISO-Norm für benutzergerechte interaktive Systeme*. In: *c't Magazin für Computertechnik* 1998(14). S. 168-172.
- Geiser, Georg (1990): *Mensch-Maschine-Kommunikation*. Oldenbourg; München, Wien, 1990.
- Gelderman, Maarten (1998): *The relation between user satisfaction and usage of information systems and performance*. In: *Information and Management* 1998. 34. S. 11-18.
- Gibbs, W. Wayt (1997): *Taking Computers to Task*. In: *Scientific American* 1997. 277(7). S. 64-71.
- Gilb, Tom (1988): *Principles of Software Engineering Management*. Addison-Wesley; Wokingham u.a.; 1988.
- Goodhue, Dale/Wybo, Michael/Kirsch, Laurie (1992): *The Impact of Data Integration on the Costs and Benefits of Information Systems*. In: *MIS Quarterly* 1992(3). S. 293-311.
- Gould, John/Boies, Stephen/Ukelson, Jacob (1997): *How To Design Usable Systems*. In: Helander, Martin/Landauer, Thomas/Prabhu, Prasad (Hrsg.): *Handbook of Human Computer Interaction*. Elsevier; Amsterdam u.a.; 1997. S. 231-254.
- Grosz, Barbara (1996): *Collaborative Systems*. In: *AI Magazine* 1996. 17(2). S. 67-85.
- Haas, Rainer/Thurnher, Bettina (2006): *Marketing und Usability. Ein Methodenvergleich und Synergiepotentiale*. In: *OCG-Journal* 2006. 31(1). S. 27-29.
- Hansen, Hans/Neumann, Gustav (2005): *Wirtschaftsinformatik 1. Grundlagen und Anwendungen*. Lucius & Lucius; Stuttgart u.a.; 2005.
- Hansmann, Holger/Neumann, Stefan (2002): *Prozessorientierte Einführung von ERP-Systemen*. In: Becker, Jörg/Kugeler, Martin/Rosemann, Michael (Hrsg.): *Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung*. Springer; Berlin u.a.; 2002. S. 327 - 372.
- Harmon, Paul (2003): *Business Process Change. A Manager's Guide to Improving, Redesigning and Automating Processes*. Morgan Kaufmann; San Francisco; 2003.
- Harrison, Mary/Henneman, Richard/Blatt, Louis (1994): *Design of a Human Factors Cost-Justification Tool*. In: Bias, Randolph/Mayhew, Deborah (Hrsg.): *Cost-Justifying Usability*. Academic Press; Boston u.a.; 1994. S. 203-241.
- Hoffman, Thomas (2002): *TCO. Flawed But Useful*. In: *Computerworld*. 12.02.2002. 36(49). S. 52.
- Holland, Christopher (1995): *Cooperative supply chain management: the impact of inter-organizational information systems*. In: *Journal of Strategic Information Systems* 1995. 4(2). S. 117-133.
- Holzinger, Andreas (2001a): *Basiswissen Multimedia. Band 3: Design*. Vogel; Würzburg; 2001.
- Holzinger, Andreas (2001b): *Mensch-Computer-Interaktion - ein interdisziplinäres Thema. OCG-Außenstelle Graz mit einem HCI-Syposium offiziell eröffnet*. In: *computer kommunikativ* 2001. 26(1). S. 14-16.
- Holzinger, Andreas (2005): *Usability Engineering Methods for Software Developers*. In: *Communications of the ACM* 2005. 48(1). S. 71-74.

- Hoppe, Heinz (1988): *Werkzeuge für die Prototypenentwicklung von Benutzerschnittstellen*. In: Balzert, Helmut et al. (Hrsg.): *Einführung in die Software-Ergonomie*. De Gruyter; Berlin, New York, 1988. S. 280-297.
- Jochem, Michael (1998): *Einführung integrierter Standardsoftware. Ein ganzheitlicher Ansatz*. Peter Lang; Frankfurt am Main; 1998.
- Juristo, Natalia/Windl, Helmut/Constantine, Larry (2001): *Introducing Usability*. In: *IEEE Software* 2001. 18(1). S. 20-21.
- Kalbach, James (2003): *Von Usability überzeugen*. In: Heinsen, Sven/Vogt, Petra (Hrsg.): *Usability praktisch umsetzen. Handbuch für Software, Web, Mobile Devices und andere innovative Produkte*. Carl Hanser Verlag; München, Wien; 2003. S. 7-21.
- Karat, Clare-Marie (1994): *A Business Case Approach to Usability Cost Justification*. In: Bias, Randolph/Mayhew, Deborah (Hrsg.): *Cost-Justifying Usability*. Academic Press; Boston u.a.; 1994. S. 45-70.
- Klaus, Helmut/Rosemann, Michael/Gable, Guy (2000): *What is ERP?* In: *Information System Frontiers* 2000. 2(2). S. 141-162.
- Krasner, Herb (2000): *Ensuring E-Business Success by Learning from ERP Failures*. In: *IT Professional* 2000. 2(1). S. 22-27.
- Legner, Christine/Österle, Hubert (1999): *Prozeßbenchmarking - Ein Methodischer Ansatz zur Prozeßentwicklung mit Standardsoftware*. In: Scheer, August-Wilhelm/Nüttgens, Markus (Hrsg.): *Electronic Business Engineering*. Physica-Verlag; Heidelberg; 1999. S. 331-352.
- Liebmann, Lenny (1999): *The TCO-Myth*. In: *Communications News* 1999(12). S. 92.
- Lin, Chad/Pervan, Graham (2003): *The practice of IS/IT benefits management in large Australian organizations*. In: *Information & Management* 2003. 41(1). S. 13-24.
- Maaß, Susanne (1993): *Software-Ergonomie. Benutzer- und aufgabenorientierte Softwaregestaltung*. In: *Informatik-Spektrum* 1993. 16(4). S. 191-205.
- Mabert, Vincent/Soni, Ashok/Venkataramanan, M.A. (2001): *Enterprise Resource Planning: Common Myths Versus Evolving Reality*. In: *Business Horizons* 2001. 44(3). S. 69-76.
- Mabert, Vincent/Soni, Ashok/Venkataramanan, M.A. (2003): *The impact of organization size on enterprise resource planning (ERP) implementations in the US manufacturing sector*. In: *Omega* 2003. 31(3). S. 235-246.
- Mantei, Marilyn/Teorey, Toby (1988): *Cost/benefit analysis for intercorporating human factors in the software lifecycle*. In: *Communications of the ACM* 1993. 31(4). S. 428-439.
- Martin, Michael (1998): *An Electronics Firm Will Save Big Money by Replacing Six People With One and Lose All This Paperwork, Using Enterprise Resource Planning Software. But Not Every Company Has Been so Lucky*. In: *Fortune* 1998. 137(2). S. 149-151.
- Martin, Reiner (2003): *Rechnen sich ERP-Systeme? Die Software allein ist nicht das entscheidende Erfolgskriterium*. In: *Neue Zürcher Zeitung*. 23.09.2003(220). S. 66.
- Martin, Reiner/Mauterer, Heiko/Gemünden, Hans-Georg (2002): *Systematisierung des Nutzens von ERP-Systemen in der Fertigungsindustrie*. In: *Wirtschaftsinformatik* 2002. 44(2). S. 109-116.
- Martin, Reiner/Lempp, Peter/Mauterer, Heiko (2003): *Wie Software wirklich Nutzen bringt*. In: *Harvard Business manager* 2003. 25(6). S. 71-77.
- Masak, Dieter (2005): *Moderne Enterprise Architekturen*. Springer; Berlin, Heidelberg; 2005.
- Mayhew, Deborah/Mantei, Marilyn (1994): *A Basic Framework for Cost-Justifying Usability Engineering*. In: Bias, Randolph/Mayhew, Deborah (Hrsg.): *Cost-Justifying Usability*. Academic Press; Boston u.a.; 1994. S. 9-43.
- Mukhopadhyay, Tridas/Rajiv, Surendra/Srinivasan, Kannan (1997): *Information Technology Impact on Process Output and Quality*. In: *Management Science* 1997. 43(12). S. 1645-1659.

- Muschter, Sebastian/Österle, Hubert (1999): *Investitionen in Standardsoftware: Ein geschäftsorientierter Ansatz zur Nutzenmessung und -bewertung*. In: Scheer, August-Wilhelm/Nüttgens, Markus (Hrsg.): *Electronic Business Engineering*. Physica-Verlag; Heidelberg; 1999. S. 443 - 468.
- Nah, Fiona (2001): *Characteristics of ERP-software maintenance: a multiply case study*. In: *Journal of Software Maintenance and Evolution: Research and Practice* 2001. 13(6). S. 399-414.
- Oppermann, Reinhard (1988): *Software-ergonomische Evaluationsverfahren*. In: Balzert, Helmut et al. (Hrsg.): *Einführung in die Software-Ergonomie*. De Gruyter; Berlin, New York, 1988. S. 323 - 342.
- Oppermann, Reinhard et al. (1992): *Software-ergonomische Evaluation. Der Leitfaden EVADIS II*. De Gruyter; Berlin, New York; 1992.
- o.V. (1999): *Standardpakete erschlagen professionelle Anwender*. In: *Computer Zeitung* 1999. (37). S. 1.
- o.V. (2000): *Helpdesk-Mitarbeiter brauchen starke Nerven und ein offenes Ohr*. In: *Computer Zeitung* 2000(3). S. 19.
- o.V. (2004): *Gute Benutzerschnittstellen zahlen sich aus*. In: *Computer Zeitung* 2004(9). S. 9.
- o.V. (2005): *SAP will Gesamtbetriebskosten senken*. In: *Computer Zeitung* 2005(23). S. 14.
- o.V. (2006): *Usability hat ein enormes Potenzial*. In: *Computer Zeitung* 2006(24). S. 8.
- Parry, Glenn (2005): *Counting the Cost*. In: *IEEE Manufacturing Engineer* 2005. 84(1). S. 22-25.
- Powel, Wayne/Barry, Jim (2005): *Planning for the Future by Looking Back. An ERP Post-Implementation Review*. In: *Educause Quarterly* 2005. 28(3). S. 40-46.
- Preim, Bernhard (1999): *Entwicklung interaktiver Systeme. Grundlagen, Fallbeispiele und innovative Anwendungsfelder*. Springer; Berlin, Heidelberg; 1999.
- Promberger, Kurt/Schlager-Weidinger, Norbert/Traxl, Markus (2003): *Verwaltungsmodernisierung durch Enterprise Resource Planning Systeme. Steigerung der Leistungsfähigkeit öffentlicher Verwaltungen durch den Einsatz betriebswirtschaftlicher Standardsoftware*. Neuer Wissenschaftlicher Verlag; Wien, Graz; 2003.
- Rohr, Gabriele (1988): *Grundlagen menschlicher Informationsverarbeitung*. In: Balzert, Helmut et al. (Hrsg.): *Einführung in die Software-Ergonomie*. De Gruyter; Berlin, New York; 1988. S. 27-48.
- Sarpola, Sami (2003): *Enterprise Resource Planning (ERP). Software Selection and Success of Acquisition Process in Wholesale Companies*. HeSE print; Helsinki; 2003.
- Scherer, Eric (2005): *Return on Investment von ERP-Investitionen*. In: Meyer, Heinrich (Hrsg.): *Netzguide Business-Software 2005*. Netzmedien AG; Basel; 2005. S. 16-19.
- Scherer, Eric (2006): *ERP-Systeme: Anwendung entscheidet über Erfolge*. In: Meyer, Heinrich (Hrsg.): *Netzguide Business-Software 2006*. Netzmedien AG; Basel; 2006. S. 6-8.
- Schumann, Matthias (1993): *Wirtschaftlichkeitsbeurteilung für IV-Systeme*. In: *Wirtschaftsinformatik* 1993. 35(2). S. 167-178.
- Scott, Judy (2005): *Post-Implementation Usability of ERP Training Manuals: The User's Perspective*. In: *Information Systems Management* 2005. 22(2). S. 67-77.
- Seddon, Peter (1997): *A Respecification and Extension of the DeLone and McLean Model of IS Success*. In: *Information Systems Research* 1997. 8(3). S. 240-253.
- Siering, Peter (1997): *Windows-XXL. Was die nächsten Windows-Versionen bringen*. In: *c't Magazin für Computertechnik* 1997(5). S. 20.
- Silk, David (1990): *Managing IS benefits fort he 1990s*. In: *Journal of Information Technology* 1990. 5(4). S. 185-193.

- Somers, Toni/Nelson, Klara/Karimi, Jahangir (2003): *Confirmatory Factor Analysis of the End-User Computing Satisfaction Instrument: Replication within an ERP Domain*. In: *Decision Sciences* 2003. 34(3). S. 595-621.
- Sontow, Karsten (2006a): *ERP-Evaluation - sicher über Investitionen entscheiden*. In: Meyer, Heinrich (Hrsg.): *Netzguide Business-Software 2006*. Netzmedien AG; Basel; 2006. S. 42-43.
- Sontow, Karsten (2006b): *Funktionen bestimmen ERP-Auswahl*. In: *is report* 2006. 10(3). S. 24-25.
- Spitta, Thorsten (1998): *IV-Controlling in mittelständischen Industrieunternehmen. Ergebnisse einer empirischen Studie*. In: *Wirtschaftsinformatik* 1998. 40(5). S. 424-433.
- Stary, Christian (1994): *Interaktive Systeme. Software-Entwicklung und Software-Ergonomie*. Vieweg; Braunschweig/Wiesbaden; 1994.
- Steinbuch, Pitter (1998): *Prozessorganisation - Business Reengineering - Beispiel R/3*. Kiehl; Ludwigshafen (Rhein); 1998.
- Streitz, Norbert (1988): *Fragestellungen und Forschungsstrategien der Software-Ergonomie*. In: Balzert, Helmut et al. (Hrsg.): *Einführung in die Software-Ergonomie*. De Gruyter; Berlin, New York, 1988. S. 3-24.
- Taninecz, George (1996): *What's the ROI?* In: *Industry Week*. 07.10.1996. S. 45-48.
- Trunick, Perry (1999): *ERP: Promise or pipe dream?* In: *Transportation and Distribution* 1999. 40(1). S. 23-26.
- Ulich, Eberhard (1988): *Arbeits- und organisationspsychologische Aspekte*. In: Balzert, Helmut et al. (Hrsg.): *Einführung in die Software-Ergonomie*. De Gruyter; Berlin, New York, 1988. S. 49-66.
- Ullerich, Tatjana (2004): *TCO-Modell für SAP-Systeme am Beispiel mySAP CRM mit SAP Enterprise Portal*. Galileo Press; Bonn; 2004.
- Viehmann, Robert (2005): *Best Practices verringern das Projektrisiko*. In: *Computer Zeitung* 2005(18). S. 18.
- Weston, Ted (2003): *ERP II: The extended enterprise system*. In: *Business Horizons* 2003. 46(6). S. 49-55.
- Wildstrom, Stephen (1998): *A Computer User's Manifesto*. In: *Business Week* 28.09.1998. 3597. S. 18.
- Wixon, Dennis/Wilson, Chauncey (1997): *The Usability Engineering Framework for Product Design and Evaluation*. In: Helander, Martin/Landauer, Thomas/Prabhu, Prasad (Hrsg.): *Handbook of Human Computer Interaction*. Elsevier; Amsterdam u.a.; 1997. S. 653-688.

Konferenzbeiträge

- Babaian, Tamara/Lucas, Wendy/Topi, Heikki (2004): *Collaborating to Improve ERP Usability*. In: *Proceedings of the 6th International Conference on Enterprise Information Systems*. Porto, Portugal; 14.04.2004 - 17.04.2004. S. 164-168.
- Bailey, Wayne/Knox, Stephen/Lynch, Eugene (1988): *Effects of interface design upon user productivity*. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. Washington D.C., USA; 15.05.1988 - 19.05.1988. S. 207-212.
- Bernroider, Edward/Hampel, Alexander (2005): *Enterprise Resource Planning and IT Governance in Perspective: Strategic Planning and Alignment, Value Delivery and Controlling*. In: *Proceedings of the Fifth International Conference on Electronic Business*. Hong Kong, China; 05.12.2005 - 09.12.2005. S. 306-308.
- Corbitt, Gail/Mensching, Jim (2000): *ERP Issues and Answers: Session Overview*. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Waikoloa, USA; 04.01.2000 - 07.01.2000. S. 7015.

- Drury, Donald (2001): *Determining IT TCO: Lessons and Extensions*. In: *The 9th European Conference on Information Systems*. Bled, Slowenien; 27.06.2001 - 29.06.2001. S. 825-836.
- Gattiker, Thomas/Goodhue, Dale (2000): *Understanding the Plant Level Costs and Benefits of ERP: Will the Ugly Duckling Always Turn Into a Swan?* In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Waikoloa, USA; 04.01.2000 - 07.01.2000. S. 7017.
- Kristiansen, Renate (2005): *Task-based tailoring of ERP System's User Interface*. In: *Proceedings of the 12th doctoral consortium on advanced information systems engineering*. Porto, Portugal; 13.06.2005 - 14.06.2005.
- Sedera, Darshana/Tan, Felix (2005): *User Satisfaction: An Overarching Measure of Enterprise System Success*. In: *9th Pacific Asia Conference on Information Systems*. Bangkok, Thailand; 07.07.2005 - 10.07.2005. S. 963-976.
- Thavapragasam, Xavier (2004): *ERP Systems and User Perceptions: An Approach for Implementation Success*. In: *Issues in Informing Science and Information Technology*. Rockhampton, Australia; 25.06.2004 - 28.06.2004. S. 521-531.
- Topi, Heikki/Lucas, Wendy/Babaian, Tamara (2005): *Identifying Usability Issues with an ERP Implementation*. In: *Proceedings of the 7th International Conference on Enterprise Information Systems*. Miami, USA; 25.05.2005 - 28.05.2005. S. 128-133.
- Verdaasdonk, Peter/Oomen, Toon (2001): *Explaining ERP users' (dis)satisfaction: a case study*. In: *5th EIASM International Seminar on Manufacturing Accounting Research*. Pisa, Italien; 06.06.2001 - 08.06.2001.
- Wu, Jen-Her et al. (2002): *An Examination of ERP User Satisfaction in Taiwan*. In: *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. Waikoloa, USA; 07.01.2002 - 10.01.2002.

Arbeitsberichte

- Babaian, Tamara/Lucas, Wendy/Topi, Heikki (o.J.): *Improving ERP Usability Through User-System Collaboration*. Bentley College (Waltham, MA), Computer Information Systems Department.
- Becker, Matthias/Fleisch, Elgar/Österle, Hubert (1997): *Business Process Reengineering mit SAP R/3*. Fallstudie Kontron Elektronik. Universität St. Gallen, Institut für Wirtschaftsinformatik.
- Dolmetsch, Ralph et al. (1998): *Accelerated SAP. 4 Case Studies*. Universität St. Gallen, Institut für Wirtschaftsinformatik.
- Grob, Heinz/Lahme, Norman (2004): *Total Cost of Ownership-Analyse mit VOFI*. Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik.
- Knolmayer, Gerhard/von Arb, Reto/Zimmerli, Christoph (1997): *Erfahrungen mit der Einführung von SAP R/3 in Schweizer Unternehmen*. Universität Bern, Institut für Wirtschaftsinformatik, Abteilung Information Engineering.
- Martin, Reiner/Mauterer, Heiko/Lempp, Peter (2001): *Der Nutzen von ERP-Systemen. Eine Analyse am Beispiel von SAP R/3*. Fachhochschule Konstanz, Technische Universität Berlin, Cap Gemini Ernst & Young.
- Treber, Udo/Teipel, Philip/Schwickert, Axel (2004): *Total Cost of Ownership - Stand und Entwicklungstendenzen 2003*. Justus-Liebig-Universität Gießen, Professur Betriebswirtschaftslehre - Wirtschaftsinformatik.
- Wild, Martin/Herges, Sascha (2000): *Total Cost of Ownership (TCO) - Ein Überblick*. Universität Mainz, Lehrstuhl für Allgemeine BWL und Wirtschaftsinformatik.

Vorlesungsunterlagen

Karcher, Andreas (2005): *Vorlesung Anwendungssysteme: Integrierte betriebliche Informationsverarbeitung. Kapitel 7 „Bereichsübergreifende Ansätze“*. Herbst-Trimester 2005. Universität der Bundeswehr München, Deutschland.

Plattner, Hasso (1999): *Blockvorlesung Technologie integrierter Informationssysteme*. Wintersemester 1998/1999. Universität Saarbrücken, Deutschland.

Internetquellen

ACM SIGCHI (1996): *Curricula for Human Computer Interaction. Chapter 2: Human-Computer Interaction*. Online in Internet. URL: <<http://sigchi.org/cdg/cdg2.html>>, [19.01.2006]

ACM SIGHCI (2003): *SIGHCI Newsletter. July 2003*. Online in Internet. URL: <http://sigs.aisnet.org/SIGHCI/newsletters/v2n1_0703.pdf>, [22.03.2006]

Anzer Business Systems (o.J.): *Total Cost of Ownership (TCO)*. Online in Internet. URL: <<http://www.anzer.com/tco.htm>>, [07.10.2000]

Apple Computer (2005): *Introduction to Apple Human Interface Guidelines*. Online in Internet. URL: <http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHI_Guidelines/>, [08.02.2006]

Armbruster, Heidi et al. (2005): *Techno-Organisational Innovation in the European Manufacturing Industry. Do European Countries Differ Regarding the Diffusion of Technical and Non-Technical Innovations in Manufacturing Companies?* Online in Internet. URL: <<http://www.systemsresearch.ac.at/getdownload.php?id=146>>, [02.06.2005]

Barker, Dean (2000): *Cost Benefits of Usability Engineering*. Online in Internet. URL: <<http://www.interfacearchitecture.net/articles/benefits.htm>>, [10.02.2005]

Berry, Richard (1988): *Common User Access - A consistent and usable human-computer interface for the SAA environments*. Online in Internet. URL: <<http://www.research.ibm.com/journal/sj/273/ibmsj2703E.pdf>>, [08.02.2006]

Bevan, Nigel (1995): *Human-Computer Interaction Standards*. Online in Internet. URL: <<http://www.usability.serco.com/papers/hcistd95.pdf>>, [02.02.2006]

Bevan, Nigel (2000): *Cost Benefit Analysis*. Online in Internet. URL: <http://www.cs.concordia.ca/~comp773/article/cost_benefits_analysis_bevan.doc>, [11.10.2006]

Bevan, Nigel/Curson, Ian (1998): *Planning and Implementing User-Centred Design*. Online in Internet. URL: <<http://www.usability.serco.com/papers/ucdtut97.pdf>>, [04.02.2006]

Bevan, Nigel/Kirakowski, Jurek/Maissel, Jonathan (1991): *What is Usability?* Online in Internet. URL: <<http://www.usability.serco.com/papers/whatis92.pdf>>, [02.02.2006]

Black, Jane (2002): *Usability Is Next to Profitability*. Online in Internet. URL: <http://www.businessweek.com/technology/content/dec2002/tc2002124_2181.htm>, [20.01.2006]

Bräutigam, Lothar (2000): *Software-Ergonomie*. Online in Internet. URL: <<http://www.sozialnetzhessen.de/ca/ph/het/Hauptpunkt/aaaaaaaaaaaaahfi/Unterpunkt/aaaaaaaaaaaaaiih/HauptframeID/aaaaaaaaaaaaaalul/HauptframeTemplate/aaaaaaaaaaaaaapq/>>, [19.01.2006]

Butters, Ingo (2005): *ERP-Markt übertrifft Erwartungen - Nischenanbieter mit kräftigem Wachstum*. Online in Internet. URL: <<http://www.cio.de/markt/uebersichten/812486/index.html>>, [28.05.2006]

Computer Sciences Corporation (CSC) (2004): *Open Source. Open for Business*. Online in Internet. URL: <http://www.csc.com/features/2004/uploads/LEF_OPENSOURCE.pdf>, [25.04.2006]

DATEch (2004): *DATEch-Prüfhandbuch Gebrauchstauglichkeit. Leitfaden für die ergonomische Evaluierung von Software auf Grundlage von DIN EN ISO 9241, Teile 10 und 11*. Online in Internet. URL: <http://www.datech.de/share/files/Pruefhandbuch_ISO_9241.pdf>, [23.01.2006]

- Eichinger, Armin (o.J. a): Usability - Vorbemerkungen. Online in Internet. URL: <<http://pcptpp030.psychologie.uni-regensburg.de/student2001/Skripten/Zimmer/u-vorbemerkungen.html>>, [18.01.2006]
- Eichinger, Armin (o.J. b): Usability - Definition. Online in Internet. URL: <<http://pcptpp030.psychologie.uni-regensburg.de/student2001/Skripten/Zimmer/u-definition.html>>, [18.01.2006]
- Eichinger, Armin (o.J. c): Usability Engineering. Online in Internet. URL: <<http://pcptpp030.psychologie.uni-regensburg.de/student2001/Skripten/Zimmer/u-engineering.html>>, [18.01.2006]
- Eichinger, Armin (o.J. d): Usability Engineering (2). Online in Internet. URL: <<http://pcptpp030.psychologie.uni-regensburg.de/student2001/Skripten/Zimmer/u-engineering2.html>>, [18.01.2006]
- Eichinger, Armin (o.J. e): Usability Tests. Online in Internet. URL: <<http://pcptpp030.psychologie.uni-regensburg.de/student2001/Skripten/Zimmer/u-tests.html>>, [18.01.2006]
- Eichinger, Armin (o.J. f): Usability - Kosten. Online in Internet. URL: <<http://pcptpp030.psychologie.uni-regensburg.de/student2001/Skripten/Zimmer/u-kosten.html>>, [18.01.2006]
- Ergonomics Society (o.J.): Ergonomics. Definition. Online in Internet. URL: <<http://www.ergonomics.org.uk/ergonomics/definition.htm>>, [26.01.2006]
- EUR-Lex (o.J.): Richtlinie 90/270/EWG des Rates vom 29. Mai 1990 über die Mindestvorschriften bezüglich der Sicherheit und des Gesundheitsschutzes bei der Arbeit an Bildschirmgeräten (Fünfte Einzelrichtlinie im Sinne von Artikel 16 Absatz 1 der Richtlinie 89/391/EWG). Online in Internet. URL: <<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31990L0270:DE:HTML>>, [11.02.2006]
- European Commission (2005): The European e-Business Report. 2005 edition. A portrait of e-business in 10 sectors of the EU economy. Online in Internet. URL: <<http://www.ebusiness-watch.org/resources/documents/eBusiness-Report-2005.pdf>>, [02.06.2006]
- Fachrichtung Informationswissenschaft Saarbrücken (2006a): Arbeitsbereich Usability Engineering. Methoden und Verfahren. Die Expertenzentrierten Methoden. Online in Internet. URL: <http://usability.is.uni-sb.de/methoden/experten_methoden.php>, [28.09.2006]
- Fachrichtung Informationswissenschaft Saarbrücken (2006b): Arbeitsbereich Usability Engineering. Methoden und Verfahren. Die Nutzerzentrierten Methoden. Online in Internet. URL: <http://usability.is.uni-sb.de/methoden/nutzer_methoden.php>, [02.02.2006]
- Färbinger, Peter (2000): Enterprise Resource Planning. Online in Internet. URL: <<http://ebusiness.or.at/erp.htm>>, [12.02.2006]
- Gadatsch, Andreas (2004): IT-Arbeitsplatzmanagement. Effiziente Gestaltung von IT-Prozessen. Online in Internet. URL: <[http://www.competence-site.de/gpm.nsf/5D1D5AF91DB513E4C1256EDC00690C37/\\$File/gadatsch_it_arbeitsplatzmanagement.pdf](http://www.competence-site.de/gpm.nsf/5D1D5AF91DB513E4C1256EDC00690C37/$File/gadatsch_it_arbeitsplatzmanagement.pdf)>, [17.06.2006]
- Gartner Group (1997): TCO Analyst. A White Paper on GartnerGroup's Next Generation Total Cost of Ownership Methodology. Online in Internet. URL: <http://www.go-eol.com/whitepapers/pdfs/Gart_tco.pdf>, [26.06.2006]
- Gartner Group (2003): Distributed Computing. Chart of Accounts. Online in Internet. URL: <http://www.gartner.com/4_decision_tools/modeling_tools/costcat.pdf>, [04.07.2006]
- Geis, Thomas (o.J.): Willkommen im Usability Begriffszoo. Online in Internet. URL: <<http://www.fit-fuer-usability.de/1x1/basics/begriffszoo.html>>, [19.01.2006]
- Gilbert, Alorie (2003): Business apps get bad marks in usability. Online in Internet. URL: <<http://news.com.com/2100-1017-980648.html>>, [21.09.2006]
- Hellbardt, Günter (2000): Vorlesung Software-Ergonomie. Texte und Folien. Einführung. Online in Internet. URL: <http://www1.informatik.uni-jena.de/lehre/SoftErg/vor_e100.htm>, [23.01.2006]
- Holzinger, Andreas (2006a): Usability - Erfolgsfaktor für Unternehmen. Interdisziplinäre Erkenntnisse aus Psychologie und Informatik bestimmen die Zukunft. Online in Internet. URL: <[http://www.competence-site.de/wissensmanagement.nsf/%208A644AB5EEF5063FC12570EC00351C51/\\$File/g47_holzinger_competencesite01.pdf](http://www.competence-site.de/wissensmanagement.nsf/%208A644AB5EEF5063FC12570EC00351C51/$File/g47_holzinger_competencesite01.pdf)>, [01.02.2006].

- Holzinger, Andreas (2006b): *Usability Engineering Methods (UEMs) for Software Developers*. Online in Internet. URL: <<http://user.meduni-graz.at/andreas.holzinger/holzinger/usability.html>>, [21.09.2006]
- Hunkirchen, Peter (2005): *Bitte laut denken: "Thinking Aloud"*. Online in Internet. URL: <<http://www.fit-fuer-usability.de/1x1/messen/thinking.html>>, [02.02.2006]
- IBM (o.J.): *Cost justifying ease of use. Complex solutions are problems*. Online in Internet. URL: <http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/23>, [20.01.2006]
- Lucas, Wendy/Babaian, Tamara/Topi, Heikki (2004): *Identifying ERP Usability Issues*. Online im Internet. URL: <<http://www.nottingham.ac.uk/cesr/Sem5WendyLucas.ppt>>, [08.03.2006]
- Microsoft (2003): *Das österreichische Bundesministerium für Inneres reduziert die TCO um € 7,26 Mio durch den Einsatz des Windows Server 2003*. Online in Internet. URL: <<http://www.microsoft.at/referenzen/referenz.asp?id=367>>, [26.06.2006]
- MORI (1999): *Employees get 'It' Out Of Their Systems. Frustration levels with technology soar, according to new survey*. Online in Internet. URL: <<http://www.mori.com/polls/1999/rage.shtml>>, [28.08.2006]
- Mutschler, Bela/Reichert, Manfred (2004): *Usability-Metriken als Nachweis der Wirtschaftlichkeit von Verbesserungen der Mensch-Maschine-Schnittstelle*. Online in Internet. URL: <<http://www.mutschler.info/downloads/MetriKon-Mutschler-Reichert-Usability-Metriken.pdf>>, [20.01.2006]
- Nielsen, Jakob/Pernice Coyne, Kara (2001): *A Useful Investment*. Online in Internet. URL: <http://www.cio.com/archive/021501/et_pundits.html>, [22.01.2006]
- Reilly, Kevin (2005): *AMR Research Releases Report Showing Overall European Market for ERP Vendors to Grow 7% Annually Through 2009*. Online in Internet. URL: <<http://www.amrresearch.com/Content/View.asp?pmillid=18386>>, [02.02.2006]
- Schneider, Wolfgang (2000): *Selbstbeschreibungsfähigkeit*. Online in Internet. URL: <<http://www.sozialnetz-hessen.de/ca/ph/het/Hauptpunkt/aaaaaaaaaaaaahfi/Unterpunkt/aaaaaaaaaaaaaiih/HauptframeID/aaaaaaaaaaaaaluw/HauptframeTemplate/aaaaaaaaaaaaaapq/>>, [05.02.2006]
- Serco (2001): *User centred design standards*. Online in Internet. URL: <<http://usabilitynet.org/trump/resources/standards.htm>>, [04.02.2006]
- Slater, Derek (1998): *The Hidden Costs of Enterprise Software*. Online in Internet. URL: <http://www.cio.com/archive/enterprise/011598_erp.html>, [28.09.2006]
- Smith, Sidney/Mosier, Jane (1986): *Guidelines For Designing User Interface Software*. Online in Internet. URL: <<http://hcibib.org/sam/>>, [11.02.2006]
- Usability Forum (2001): *Cognitive Walkthrough - eine effiziente Evaluationsmethode*. Online in Internet. URL: <<http://www.usability-forum.com/newsletter/archiv/02102001.shtml>>, [28.09.2006]
- Usability Forum (o.J. a): *HCI-Geschichte*. Online in Internet. URL: <<http://www.usability-forum.com/bereiche/einleitung/geschichte.shtml>>, [27.01.2006]
- Usability Forum (o.J. b): *Usability Engineering*. Online in Internet. URL: <<http://www.usability-forum.com/bereiche/usabilityengineering.shtml>>, [18.01.2006]
- Usability Forum (o.J. c): *Was bringt Usability?* Online in Internet. URL: <<http://www.usability-forum.com/bereiche/kosten.shtml>>, [18.01.2006]
- Usability Professionals' Organization (UPA) (o.J.): *The ROI of Usability*. Online in Internet. URL: <http://www.upassoc.org/usability_resources/usability_in_the_real_world/roi_of_usability.html>, [24.01.2006]
- von Arb, Reto (1997): *Vorgehensweise und Erfahrungen bei der Einführung von Enterprise-Management-Systemen dargestellt am Beispiel von SAP R/3*. Online in Internet. URL: <<http://www.digital-publications.ch/vonarb/vonarb.zip>>, [12.02.2006]

- Wirth, Thomas (2005): Die EN ISO 9241 - 10. Online in Internet. URL: <<http://kommdesign.de/texte/din.htm>>, [19.01.2006]
- Zillich, Christian (2002): Mit Rol und TCO auf der Scorecard balancieren. Online in Internet. URL: <<http://www.computerwoche.de/index.cfm?pid=695&pk=528137>>, [26.06.2006]
- Zviran, Moshe (2003): User satisfaction in ERP system: some empirical evidence. Online in Internet. URL: <<http://www.highbeam.com/library/docFree.asp?DOCID=1G1:113563625>>, [27.03.2006]

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Diese Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Katrin Kneissl

Innsbruck, Oktober 2006