

Interaktive Ad-hoc-Evaluation von Desktopsoftware durch Endbenutzer

Eine explorative Studie

Dissertation

zur

Erlangung des akademischen Grades

doctor rerum politicarum (Dr. rer. pol.)

der Wirtschafts- und Sozialwissenschaftlichen Fakultät

der Universität Rostock

vorgelegt von

Zimmermann, Jörg, geb. am 01. März 1972 in Wolgast

aus Rostock

Rostock, April 2008

Gutachter

Prof. Dr. Friedemann W. Nerdinger

Universität Rostock

Wirtschafts- und Sozialwissenschaftliche Fakultät

Lehrstuhl für Wirtschafts- und Organisationspsychologie

Prof. Dr. Hans Röck

Universität Rostock

Fakultät für Informatik und Elektrotechnik

Lehrstuhl für Wirtschaftsinformatik

Datum der Verteidigung

11. Dezember 2008

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
Abkürzungsverzeichnis	ix
Symbolverzeichnis	x
1 Einleitung und Problemstellung	1
1.1 Softwaredistribution per Internet	1
1.2 Der besondere Produktcharakter von Software	3
1.3 Ziel und Struktur der Arbeit	4
2 Terminologische Konkretisierung und Abgrenzung	6
2.1 Desktopsoftware	6
2.2 Endbenutzer	8
2.3 Interaktive Ad-hoc-Evaluation und -Auswahl	10
3 Stand der Forschung zur Evaluation und Auswahl von Software	13
3.1 Softwarequalität	13
3.2 Präskriptive Studien zur Evaluation und Auswahl von Software	18
3.3 Deskriptive Studien zur Evaluation und Auswahl von Software	23
3.4 Das Technology Acceptance Model	27
3.4.1 Modellentwicklung	27
3.4.2 Technology Acceptance Model und Softwareauswahl	32
3.5 Zusammenfassung und Schlussfolgerung	34
4 Zur Psychologie der Entscheidung	36
4.1 Grundlegende Begriffe und Konzepte	36
4.1.1 Entscheidung, Option, Attribut	36
4.1.2 Rationales Entscheiden vs. beschränkte Rationalität	37
4.1.3 Typologie des Entscheidungsverhaltens	39
4.2 Entscheidungsverhalten und Kognition	41
4.2.1 Schematheorie	41
4.2.2 Entscheidungsregeln	43
4.2.2.1 Merkmale von Entscheidungsregeln	43

4.2.2.2	Spezifische Entscheidungsregeln	45
4.2.3	Kognitive Determinanten des Entscheidungsverhaltens	47
4.2.3.1	Schwierigkeit des Entscheidungsproblems	47
4.2.3.2	Vorkenntnisse des Individuums	49
4.2.3.2.1	Speicherinteraktion	49
4.2.3.2.2	Kategorien	51
4.2.3.3	Kontext-Effekte	53
4.3	Entscheidungsverhalten und Emotion	55
4.3.1	Terminologische Abgrenzung	55
4.3.2	Einfluss von Stimmung als Hintergrundvariable	56
4.3.2.1	Stimmung als eindimensionales Konstrukt	56
4.3.2.2	Stimmung als mehrdimensionales Konstrukt	57
4.3.3	Emotion als Information	59
4.3.4	Emotion als Ergebnis des Entscheidungsproblems	62
4.4	Zusammenfassung und Schlussfolgerung	63
5	Empirische Untersuchung	65
5.1	Forschungsfragen	65
5.2	Forschungsmethode	66
5.2.1	Exploration und heuristische Beobachtung	66
5.2.2	Lautes Denken	68
5.3	Entwicklung und Aufbau der Untersuchung	69
5.3.1	Pilotphase	69
5.3.2	Bereitstellung der zu erprobenden Software	70
5.3.2.1	Softwarekategorie	70
5.3.2.2	Präsentation der Alternativen	73
5.3.2.3	Verwendete Programme	74
5.3.3	Aufgabenstellung	76
5.3.3.1	Vorüberlegungen	76
5.3.3.2	Ableitung der Aufgabenstellung	79
5.3.3.3	Wortlaut und Kommentar	82
5.3.4	Versuchspersonen	84
5.3.4.1	Auswahl und Rekrutierung	84
5.3.4.2	Vorbereitung	86
5.3.5	Datenerfassung	87
5.3.6	Geplanter Ablauf einer Sitzung	88
5.3.7	Durchführung	90
5.4	Auswertung	92
5.4.1	Beschreibung der Daten	92
5.4.2	Aufbereitung des Materials für die Analyse	93
5.4.2.1	Funktionalitätskategorien	93
5.4.2.2	Rahmenmerkmale und Orientierungsphasen	97
5.4.2.3	Abstraktion des Evaluationsverlaufes	98

5.4.2.4	Abstraktionsbeispiel	103
6	Ergebnisse der empirischen Untersuchung	105
6.1	Quantitative Auswertung	105
6.1.1	Auswahlhäufigkeit	105
6.1.2	Bearbeitungsreihenfolge	105
6.1.3	Bearbeitungsaufwand	106
6.1.4	Erfahrung	108
6.2	Qualitative Analyse I: Evaluationsstrategie	109
6.2.1	Ansatz: Empirisch begründete Typenbildung	109
6.2.2	Vorläufige Typen	111
6.2.3	Verwendete Merkmale und Merkmalsausprägungen	113
6.2.3.1	Komplexität von Abbildungen	113
6.2.3.2	Kennzahlen	114
6.2.3.3	Dichotomisierung der Merkmale	117
6.2.4	Typenbildung	119
6.2.4.1	Besetzte Merkmalskombinationen	119
6.2.4.2	Typ I – Die Aufgabenorientierten	122
6.2.4.3	Typ II – Die Minimalisten	123
6.2.4.4	Typ III – Die Verspielten	126
6.2.4.5	Typ IV – Die Unsicheren	128
6.2.5	Evaluationstyp und Auswahlresultat sowie Erfahrung	131
6.3	Qualitative Analyse II: Gründe für die Entscheidung	133
6.3.1	Ansatz: Heuristische Beobachtung	133
6.3.2	Analyse der Einzelfälle	136
6.3.3	Aggregation der Einzelfallhypothesen	138
7	Diskussion der Ergebnisse	144
7.1	Ergebnisse der Quantitativen Auswertung	144
7.2	Ergebnisse der Qualitativen Analyse I	149
7.2.1	Typologie	149
7.2.2	Zusammenhang von Typologie und Auswahlresultat sowie Erfahrung	154
7.3	Ergebnisse der Qualitativen Analyse II	159
7.4	Grenzen der Untersuchung	162
7.5	Implikationen für Forschung und Praxis	164
8	Zusammenfassung	168
A	Materialien	173
B	Versuchspersonen	186

C Quantifizierung der Interaktionsprotokolle	187
D Kennzahlen der Qualitativen Analyse I	191
Literaturverzeichnis	192

Abbildungsverzeichnis

2.1	Desktopsoftware	8
3.1	Qualitätsfaktoren, -kriterien und -metriken im McCall-Modell	15
3.2	Theorie des begründeten Handelns	28
3.3	Das Technology Acceptance Model	30
3.4	Das revidierte Technology Acceptance Model	31
4.1	Dominante psychische Prozesse und Entscheidungsverhalten	39
4.2	Simultankontrast	54
4.3	Dominanzverhältnisse und Attraktions- bzw. Kompromiss-Effekt	55
4.4	Circumplex-Modell der Emotion	58
5.1	Beispiel eines einfachen Flussdiagramms	72
5.2	Funktionalitätskategorien	96
5.3	Format des Interaktionsprotokollbogens (IPB)	100
5.4	Beispiel für ein Interaktionsprotokoll	103
6.1	Absolute und relative Auswahlhäufigkeiten nach Erfahrung	108
6.2	Bestimmung der Komplexität eines Diagramms	114
6.3	Diagramm der Kennzahlenwerte für alle Probanden	117
6.4	Diagramm der Werte für ϕX_{rel} und ϕK_{px}	119
6.5	Diagramm der Werte für $\phi K'_{rel}$ und ϕK_{px}	120
6.6	Diagramm der Merkmalswerte in dichotomer Ausprägung	121
6.7	Typische Probezeichnung eines Typ I-Probanden	124
6.8	Typische Probezeichnung eines Typ II-Probanden	125
6.9	Typische Probezeichnung eines Typ III-Probanden	128
6.10	Typische Probezeichnung eines Typ IV-Probanden	132
6.11	Absolute und relative Auswahlhäufigkeiten nach Evaluationstyp	132
6.12	Verteilung der Probanden nach Erfahrung und Evaluationstyp	133
7.1	Die Benutzeroberflächen von Flowcharter und Visio	148
7.2	Struktur der Typologie des Evaluationsverhaltens	150
7.3	Der moderierende Einfluss des Evaluationstyps im revidierten TAM	158
7.4	Der moderierende Einfluss von EoU auf U im revidierten TAM	162

Tabellenverzeichnis

5.1	Häufigkeiten der Anordnung der Programm-Icons	92
6.1	Verteilung der Bearbeitungspositionen für die angebotenen Programme .	106
6.2	Verteilung der Siegerpositionen	106
6.3	Absolute und relative Auswahlhäufigkeiten nach Bearbeitungsposition . .	107
6.4	Statistik der Bearbeitungszeiten nach verschiedenen Kriterien	107
6.5	Auswahlhäufigkeiten nach Erfahrung und Bearbeitungsposition	108
6.6	Durchschnittliche Bearbeitungszeiten nach Erfahrung	109
6.7	Besetzte Merkmalskombinationen bzw. Zellen des Merkmalsraums	120
6.8	Hierarchie der Entscheidungskriterien	142
B.1	Hintergrundinformationen zu den Probanden	186
C.1	Quantifizierung der Interaktionsprotokolle für SmartDraw	188
C.2	Quantifizierung der Interaktionsprotokolle für Flowcharter	189
C.3	Quantifizierung der Interaktionsprotokolle für Visio/DS	190
D.1	Werte der Kennzahlen der Qualitativen Analyse I	191

Abkürzungsverzeichnis

AVI	Audio Video Interleaved
CON	Konjunktionsregel
COTS	Commercial Off-The-Shelf
DBMS	Datenbankmanagementsystem
DIS	Disjunktionsregel
DOM	Dominanzregel
EBA	Elimination by Aspects (Entscheidungsregel)
ERP	Enterprise Resource Planning
FTP	File Transfer Protocol
GUI	Graphical User Interface
IEC	International Engineering Consortium
IPB	Interaktionsprotokollbogen
ISO	Internationale Organisation für Normung
J.Z.	Jörg Zimmermann (der Verfasser)
LAM	Linear Assignment Model (Entscheidungsregel)
LEX	Lexikografische Ordnung (Entscheidungsregel)
LTM	Long-term Memory
LWA	Linear Weighted Adding (Entscheidungsregel)
MAU(T)	Multi-Attribute Utility (Theory) (Entscheidungsregel)
MAX	Maximax-Regel
MBA	Master of Business Administration
MCDM	Multi-Attribute Decision Making
MW	Mittelwert
PC	Personal Computer
RSI	Repetitive Strain Injury
SAT	Satisficing-Regel
SD	Standardabweichung
STM	Short-term Memory
TAM	Technology Acceptance Model
TRA	Theory of Reasoned Action
VIE	Valenz Instrumentalität Erwartung
VL	Versuchsleiter
Vp	Versuchsperson
vTyp	vorläufiger Typus
WWW	World Wide Web

Symbolverzeichnis

A	Einstellung gegenüber einem Objekt (Attitude)
b_i	subjektive Überzeugung hinsichtlich der Eintrittswahrscheinlichkeit einer Konsequenz i
BI	Absicht, ein bestimmtes Verhalten auszuführen (Behavioral Intention)
e_i	subjektive Wertschätzung einer Konsequenz i
EoU	wahrgenommene Benutzungsfreundlichkeit einer Technologie (Ease of Use)
F	eine beliebige Funktionalität
K_i	Anzahl der über die Kernfunktionalitäten des i -ten Programms eingeholten Informationen
ϕK	durchschnittliche Anzahl der eingeholten Informationen über Kernfunktionalitäten
$\phi K'_{rel}$	durchschnittliches Verhältnis der Anzahl der wiederholt zu der der insgesamt eingeholten Informationen über Kernfunktionalitäten
Kpx_i	Komplexität des mit dem i -ten Programm erzeugten Flussdiagramms
ϕKpx	durchschnittliche Komplexität der erzeugten Flussdiagramme
M	Manipulieren eines Flussdiagramm-Objektes (Kernfunktionalität)
mc_j	Motivation, den Erwartungen einer Bezugsperson j zu entsprechen
O	Orientierungsphase
R	Rahmenwahrnehmung/-exploration
S_B	Beschriften eines Flussdiagramm-Symbols (Kernfunktionalität)
S_E	Erzeugen eines Flussdiagramm-Symbols (Kernfunktionalität)
SN	Soziale Norm (Social Norm)
U	wahrgenommene Nützlichkeit einer Technologie (Utility)
V_B	Beschriften einer Verbindung (Kernfunktionalität)
V_E	Erzeugen einer Verbindung mit Pfeilspitze (Kernfunktionalität)
X	Extrafunktionalität
X_i	Anzahl der über die Extrafunktionalitäten des i -ten Programms eingeholten Informationen
ϕX_{rel}	durchschnittliches Verhältnis der Anzahl der eingeholten Informationen über Extrafunktionalitäten zu jener über Kernfunktionalitäten

1 Einleitung und Problemstellung

1.1 Softwaredistribution per Internet

Die zunehmende Verbreitung von Mikrocomputern sowohl in Unternehmen als auch in jüngerer Vergangenheit insbesondere in privaten Haushalten¹ bedeutet einen wachsenden Bedarf an für die jeweiligen Aufgaben geeigneter Standardsoftware. Die daraus erwachsende Nachfrage bringt es mit sich, dass es für jedes Anwendungsgebiet bzw. jedes Problem, das mithilfe des Computers gelöst werden soll oder durch diesen überhaupt erst auftritt, eine Vielzahl von Software-Lösungen gibt. Dass das nicht immer so war soll hier jedoch ebenso wenig thematisiert werden wie der Umstand, dass in den frühen Tagen der elektronischen Datenverarbeitung Software nicht als eigenständiges Produkt angesehen wurde, sondern als notwendige Beigabe zur Hardware, um diese überhaupt absetzbar zu machen. Vielmehr geht es um mögliche Distributionswege und -konzepte sowie insbesondere die daraus folgenden Konsequenzen für Evaluation und Auswahl aus dem Angebot.

Software ist vom Produktcharakter her grundsätzlich immateriell, jedoch hatte dieser Umstand lange Zeit keine Auswirkungen auf deren Verbreitung, da sie dennoch eines materiellen oder physischen Trägermediums bedurfte. So war und ist Software in der Regel bereits beim Kauf eines Computers installiert oder auf Disketten bzw. CD oder DVD beigelegt. Ähnliches trifft auch auf Peripheriegeräte wie Maus, Scanner oder Drucker zu, die häufig durch Beigabe von thematisch passender Software aufgewertet werden, z. B. ein Scanner durch ein einfaches Programm zur Bildbearbeitung. Wurde über diese Ausstattung hinaus weitere Software gewünscht, musste diese, abgesehen von illegalen Formen der Beschaffung, wie ein materielles Gut bei einem entsprechenden Händler erworben werden. Die Kaufentscheidung musste, ebenfalls analog zu einem materiellen Gut, in der Regel auf rein äußerlichen Kriterien, wie dem Augenschein, Expertenempfehlungen oder den Zusicherungen des Herstellers bzw. Verkäufers beruhen.

Zu Beginn der Neunzigerjahre nahmen jedoch zwei Entwicklungen ihren Anfang, die die Distribution von Software grundlegend ändern sollten und auch freien Programmierern die Möglichkeit gaben, ihre Software kostengünstig zu verbreiten. Zum einen wurden PCs standardmäßig mit CD-ROM-Laufwerken ausgestattet, zum anderen das Internet für kommerzielle Zwecke geöffnet (Hahn, 1994; Kyas, 1994).

Das standardmäßige Vorhandensein eines CD-ROM-Laufwerks ist insofern bedeutsam, als auf einer CD-ROM im Vergleich zu den bis dahin üblichen Disketten die fast 500fache Datenmenge untergebracht werden kann. Dieser Umstand, verbunden mit den äußerst kompakten Ausmaßen einer CD-ROM, machte es möglich, Zeitschriften, naturgemäß ins-

¹ Im Jahr 2006 hatten laut einer OECD-Statistik 76,8 % der Haushalte in Deutschland Zugang zu einem Computer. 2001 waren es 53,0 % (OECD, 2007).

besondere Computer-Magazine, durch Beilage eines Datenträgers mit Programmsammlungen aufzuwerten und einen großen Nutzerkreis zu erreichen. Damit gewannen bisher zwar existente, aber gemessen an der Zahl der potenziellen Nachfrager kaum relevante Vertriebsformen für Software, insbesondere sog. Shareware sowie Demoware, an Bedeutung (Hahn, 1994).

Shareware ist Software, die nach den Vorgaben des Autors beliebig kostenlos kopiert und an Dritte weitergegeben werden darf, ohne dass dadurch Urheberrechte verletzt werden (Heinrich, Heinzl & Roithmayr, 2004). Dadurch kann ein Interessent diese Software über einen bestimmten Zeitraum hinweg testen und feststellen, ob das Programm den persönlichen Bedürfnissen und Wünschen entspricht, bevor er sich zum Kauf entschließt. Dabei vertraut der Autor darauf, dass der Interessent sich nach Ablauf des Testzeitraumes kostenpflichtig registriert, auf diese Weise also das Nutzungsrecht an dieser Software erwirbt oder aber die Nutzung beendet. *Demoware* verfolgt im Prinzip das gleiche Konzept, ist aber in ihrem Funktionsumfang eingeschränkt und/oder nach Ablauf der Testzeit nicht mehr funktionsfähig, sodass die missbräuchliche Weiterverwendung ohne Registrierung ausgeschlossen ist. Weitere häufig verwendete, z. T. ähnliche Vertriebskonzepte sind z. B. *Nagware*, *Crippleware*, *Freeware* sowie *Freie/Open Source Software*.²

Softwaredistribution mittels Datenträgern, die einer Zeitschrift beigelegt sind, ist jedoch für Nutzer, die am jeweiligen redaktionellen Teil nicht interessiert sind, mit dem Mangel behaftet, dass sie zunächst ein Entgelt entrichten müssen, um das Softwareangebot überhaupt sichten zu dürfen. Hinzu kommt, dass dieses Angebot auf die Inhalte der jeweiligen CD oder DVD beschränkt ist. Wesentlich größere Bedeutung im Hinblick auf den Umfang der Auswahlmenge und somit die Wahrscheinlichkeit, eine möglichst genau den individuellen Bedürfnissen entsprechende Software zu finden, dürfte die zweite der oben angesprochenen Entwicklungen haben, nämlich die Öffnung des Internets für kommerzielle Zwecke.

Das Internet diene von je her nicht nur zum Austausch von Informationen im engeren Sinn, sondern auch für die Verbreitung von Software. Der Nutzerkreis war jedoch lange Zeit auf Universitäten und Forschungsbehörden beschränkt. Ende der Achtziger-, Anfang der Neunzigerjahre begann die Privatisierung der universitären Infrastruktur, d. h. der Erhalt und Ausbau des Internets wurde von kommerziellen Internet-Service-Providern übernommen (Kyas, 1994). Dass das Internet aber tatsächlich auch unter Laien so populär geworden ist, ist in erster Linie der Entwicklung des World Wide Web (www)³, quasi der

² *Nagware* erinnert den Nutzer durch mehr oder weniger lästige Meldungen an die Registrierung (*nag* (engl.): nörgeln). *Crippleware* ist in ihrer Funktionsfähigkeit insofern eingeschränkt, als nicht alle Funktionalitäten verfügbar sind (*cripple* (engl.): Krüppel). *Freeware* wird vom Autor kostenlos und ohne Einschränkung zur Verfügung gestellt; häufig werden jedoch freiwillige Spenden erbeten. *Freie/Open Source Software* wird aus ideologischen und pragmatischen Gründen kostenlos, ohne Einschränkung und unter Offenlegung des Quellcodes zur Verfügung gestellt.

³ Im alltäglichen Sprachgebrauch wird das www oft mit dem Internet gleichgesetzt. Tatsächlich handelt es sich dabei aber nur um eine von vielen *Nutzungen* des Internets, wenn auch die für Laien vermutlich präsenteste. Ein weiterer Grund für die Gleichsetzung der Begriffe dürfte darin liegen, dass heutige www-Browser in der Regel auch eine Reihe anderer Nutzungen, z. B. die Dateiübertragung mittels FTP (File Transfer Protocol) ermöglichen (Klute, 1996).

grafischen Benutzeroberfläche des Internets zuzuschreiben. Dadurch hat sich der Nutzerkreis in den letzten 15 Jahren von wenigen Spezialisten auf praktisch jedermann erweitert. Verbunden mit der wachsenden Zahl von Nutzern war und ist eine Vielzahl von kommerziellen Angeboten (Klute, 1996).

Spätestens mit der verbreiteten Verfügbarkeit schneller Internetzugänge jenseits von Universitäten und anderen Forschungseinrichtungen, ob nun am Arbeitsplatz oder zu Hause⁴, sind auch die technischen Möglichkeiten für die Verbreitung von Software in eine neue Dimension eingetreten. Viele unabhängige Entwickler, aber auch etablierte Unternehmen, nutzen das Internet als Distributionskanal für ihre Programme, die mit verschiedenen Lizenzen (Demo-, Share-, Freeware usw.) ausgestattet über zahlreiche Downloadportale angeboten werden. So hat zurzeit z. B. winload.de fast 20 000 Programme in verschiedenen Kategorien wie „Grafik/Desktop“, „Office/Business“ oder „Spiele“ im Angebot. Weitere deutschsprachige Anbieter sind z. B. windows-software.de, shareware.de, winsoftware.de, heise.de. In englischer Sprache finden sich z. B. tu cows.com, download.com, soft14.com.

1.2 Der besondere Produktcharakter von Software

Ein besonderes Merkmal von Software ist ihr immaterieller Charakter, wodurch sie sich von klassischen Sachgütern unterscheidet. Der gesamte Wert einer Software als Produkt steckt deshalb in ihrer Entwicklung; der letzte Schritt, das Vervielfältigen und Verteilen des Ergebnisses dieser Entwicklung, z. B. das Pressen des ausführbaren Programms auf CD/DVD oder dessen Hochladen auf einen Server, kann zu vernachlässigbaren Kosten erfolgen. Dies steht im Gegensatz zu anderen Gütern, wie z. B. der Computer-Hardware. Hier müssen nach Abschluss der Entwicklung für jedes zu produzierende Exemplar die einzelnen Teile unter Einsatz von Rohstoffen, Maschinen und Arbeitskraft hergestellt, zusammengefügt und in jedem Fall physisch zum Abnehmer transportiert werden. Eine Kopie einer Software stellt dagegen einen nur minimalen materiellen Wert dar, nämlich den des Trägermediums, bzw. kommt im Falle der Verteilung über Computernetzwerke sogar ganz ohne dedizierten physischen Träger aus.

Dieser Umstand hat insbesondere Konsequenzen für die Verteilung von Software zu Zwecken der Evaluation. Einem Interessenten eine Kopie einer Software zur Erprobung zu überlassen, stellt für einen Softwarehersteller nur ein geringes Wagnis dar, weil damit keine Überlassung oder gar Aufgabe materieller Werte verbunden ist. Um die unkontrollierte Preisgabe des ungleich höheren immateriellen Wertes zu verhindern, können z. B. Funktionalitäten wie das Speichern oder Drucken der Ergebnisse des Softwareeinsatzes deaktiviert oder die Lauffähigkeit des Programms auf einen kurzen Zeitraum beschränkt werden. Dadurch ist kein produktiver Einsatz, wohl aber eine ansonsten uneingeschränkte Erprobung durch Bearbeitung realistischer Aufgaben möglich.

⁴ Nach einer OECD-Statistik gab es im Juni 2007 in Deutschland 17,472 Mio. Breitbandanschlüsse, was einer Versorgung von 33,5 % der privaten Haushalte entspricht. 2001 waren es lediglich 1,936 Mio. Anschlüsse (OECD, 2007).

Eine weitere Konsequenz der Immaterialität von Software besteht darin, dass die konkreten Eigenschaften eines Programms im Grunde völlig beliebig sind. Während sich bei Hardware, z. B. technischen Geräten wie DVD-Spielern, schnell Standards herausbilden, durch die bestimmte Parameter eindeutig und verbindlich beschrieben werden können, ist dies bei (Standard-)Software schwieriger. Es liegt einzig am Hersteller, welche Funktionalitäten und sog. Features⁵ er einbaut und was er jeweils darunter versteht. Insofern sind in Abhängigkeit von der Qualität der begleitenden Dokumentation viele Eigenschaften von Software überhaupt nur durch Ausprobieren zu ermitteln (Steinmann, 2007).

Die Erprobung eines materiellen Gutes durch einen Interessenten ist stets verbunden mit der Gefahr der Abnutzung, Beschädigung oder Zerstörung des Gutes oder dessen garantierbarem Verlust (z. B. Warenproben) und wird deshalb nur in Ausnahmefällen unter realistischen Bedingungen möglich sein. Selbstverständlich kann z. B. eine Hose vor dem Kauf anprobiert werden. Es ist aber in der Regel nicht möglich, sie 30 Tage lang im Alltag Probe zu tragen und sich dann erst für oder gegen einen Kauf zu entscheiden. Eine Software kann dagegen wie ein Eigentümern in Besitz genommen werden, da sie nicht in dem Sinne wie ein materielles Gut zerstört werden kann. Beschädigt wird allenfalls die überlassene Kopie, die aber, wie oben ausgeführt, praktisch keinen materiellen Wert hat. In ihrer risiko- und kostenlosen Erprobbarkeit unterscheidet sich Software auch von Dienstleistungen, deren unentgeltliche Erbringung stets mit einem Verlust für den Anbieter verbunden ist.

1.3 Ziel und Struktur der Arbeit

Der immaterielle Charakter von Software bedeutet in Verbindung mit den aktuellen Möglichkeiten des Internetzugangs, dass ein Nutzer mit Bedarf an einer Software zur Bearbeitung eines bestimmten Problems zum einen eine sehr große Auswahl aus verschiedenen Programmen hat und zum anderen die infrage kommenden Produkte im Zuge der Entscheidungsfindung auch ausprobieren kann. Dies gilt durch die allgemeine Verfügbarkeit von Breitbandanschlüssen sowie die geringen Anforderungen an die Benutzung eines Webbrowsers in besonderem Maße auch für Laien, die nicht mit anspruchsvollen, wissenschaftlichen Methoden zur Softwareauswahl vertraut sind, sondern mehr „aus dem Bauch heraus“ entscheiden.⁶

⁵ Unter einem Feature wird im Allgemeinen ein Ausstattungs- oder Leistungsmerkmal einer Software verstanden.

⁶ Was nicht heißen soll, dass bei Softwarebeschaffungen im professionellen Umfeld das Bauchgefühl grundsätzlich keine Rolle spielt: „Ich habe (in Unternehmen) zwei solche Erfahrungen gemacht: a) die Kriterienliste wurde dermaßen überfrachtet, dass keine Software dem gerecht werden konnte. Eine Priorisierung bzw. eine Beschränkung auf den wesentlichen Zweck unterblieb. Wegen der Lücken, die alle Anwendungen im Test aufweisen mussten, fiel die Entscheidung letztlich aus dem Bauch heraus. b) Wegen der Präferenz eines Beteiligten für genau eine Software wurde die Kriterienliste so gebaut, dass sie nur auf diese Software passen konnte. Die Entscheidung wurde somit über den Testplan vorweg genommen“ (Steinmann, 2007).

Völlig offen und weitgehend unerforscht ist jedoch, *wie* ein einfacher Nutzer in dieser speziellen Situation vorgeht und *warum* er sich für ein bestimmtes Programm entscheidet, vorausgesetzt, die gewünschte Funktionalität steht grundsätzlich zur Verfügung. Relevant ist diese Frage zum einen durch den großen Kreis der Computernutzer und damit Software-nachfrager, zum anderen durch die plausible Annahme, dass jeder, der seine Software öffentlich anbietet, ein Interesse an deren größtmöglicher Verbreitung hat, sei es aus ideologischen oder altruistischen Motiven, für den persönlichen Ruhm, aus Arglist und mit bösen Absichten oder ganz einfach, um damit Geld zu verdienen. Das Wissen darum, wie Nutzer beim Ausprobieren von Software vorgehen und welche Gründe für die Entscheidung ausschlaggebend sind, ermöglicht eine entsprechende Optimierung bzw. Differenzierung des Angebotes. Das Ziel dieser Untersuchung besteht somit in der Klärung folgender Fragen:

1. Auf welche Weise erfolgt die Evaluation, d. h. Erforschung und Prüfung, bzw. der Vergleich der zur Auswahl stehenden Programme?
2. Wovon hängt es ab, welche Software ausgewählt wird?

Mit der wissenschaftlichen Bearbeitung dieser Fragen wird weitgehend Neuland betreten. Deshalb ist die vorliegende Arbeit als explorative Studie angelegt. Basierend auf Beobachtung von Individuen bei der Evaluation und Auswahl von Standardsoftware werden als Ergebnis der Untersuchung Hypothesen generiert, die vorläufige Antworten auf die gestellten Fragen liefern.

Im ersten Teil dieser Arbeit (Kapitel 2 bis 4) werden zunächst die im Titel verwendeten Begriffe erläutert und damit der Untersuchungsgegenstand konkretisiert und abgegrenzt. Anschließend werden der Stand der Forschung zum Thema Softwareevaluation und -auswahl sowie relevante Befunde aus dem Bereich der Entscheidungspsychologie, insbesondere zum Konsumentenverhalten, dargestellt und hinsichtlich ihrer Brauchbarkeit für die Beantwortung der gestellten Fragen diskutiert.

Im zweiten Teil dieser Arbeit (Kapitel 5) wird die Entwicklung des Forschungsansatzes zur empirischen Untersuchung der Fragestellung beschrieben. Dessen explorativer Charakter trägt dem Umstand Rechnung, dass die Literatur zur Softwareauswahl fast ausschließlich präskriptiver Natur ist, während die Befunde auf dem Gebiet Entscheidungspsychologie den im vorigen Abschnitt erläuterten Konsequenzen der Immaterialität von Software nicht gerecht werden. Entsprechend folgt die Datenerhebung und Auswertung keinem standardisierten Design, sondern erfolgt ganzheitlich im Sinne heuristischer Beobachtung und wurde speziell für den zu untersuchenden Sachverhalt entwickelt.

Der dritte Teil (Kapitel 6 und 7) ist der Auswertung der Beobachtungen sowie der Darstellung und Diskussion der Ergebnisse gewidmet. Die Auswertung erfolgt im Wesentlichen mit qualitativen, z. T. selbst entwickelten Methoden, geleitet vom Ziel der Exploration, Hypothesen zu generieren. In der anschließenden Diskussion werden die Ergebnisse zur bestehenden Theorie in Bezug gesetzt, Grenzen aufgezeigt und daraus resultierender, weiterer Forschungsbedarf skizziert.

2 Terminologische Konkretisierung und Abgrenzung

In diesem Kapitel werden die im Titel der vorliegenden Arbeit verwendeten Begriffe erläutert, um das in der Einleitung erst grob umrissene Thema zu konkretisieren und einzugrenzen. Weitere Fachbegriffe werden hier noch nicht eingeführt, sondern jeweils erst im weiteren Verlauf der Arbeit, wenn sie benötigt werden. Damit soll vermieden werden, den Leser mit einer an dieser Stelle unnötigen Fülle von Begriffen und Konzepten zu belasten.

2.1 Desktopsoftware

Der Begriff *Software* bezeichnet alle nichtphysischen Funktionsbestandteile eines Computers. Dies umfasst sowohl Computerprogramme (*aktive Software*) als auch die zur Verwendung mit Computerprogrammen bestimmten Daten sowie die technischen Beschreibungen und Entwicklungsdokumente hierzu (*passive Software*; vgl. z. B. Sommerville, 2007). Sie wird in der Regel in Gegensatz zu Hardware gesetzt, welche die physischen Funktionsbestandteile eines Computers bezeichnet und derer sie für ihre Ausführung bedarf. Im Weiteren wird der Begriff Software ausschließlich in seinem aktiven Sinne, also in der Bedeutung von Computerprogrammen verwendet.

Das Ausführen einer Software ruft stets eine ganz bestimmte Wirkung oder ein ganz bestimmtes Ergebnis hervor. Oft hat der Benutzer dabei die Möglichkeit, diese Wirkung zu steuern, sei es interaktiv während der Ausführung oder über Parameter, die bereits beim Aufruf übergeben werden. Die Gesamtheit aller möglichen erzielbaren Wirkungen einer Software wird als deren *Funktionalität* bezeichnet (Heinrich et al., 2004). Dieser Begriff wird jedoch auch für jeden einzeln abrufbaren Effekt im Sinne des Ergebnisses einer Handlung verwendet (Röck, 2007), sodass die (Gesamt-)Funktionalität einer Software der Menge ihrer Teilfunktionalitäten entspricht. So setzt sich die Gesamtfunktionalität z. B. eines Textverarbeitungsprogrammes, die als „Erzeugen und Bearbeiten von Dokumenten“ beschrieben werden kann, aus Teilfunktionalitäten wie „Anlegen eines neuen Dokumentes“, „Öffnen eines vorhandenen Dokumentes“, „Eingeben von Text“, „Löschen von Text“, „Drucken eines Dokumentes“ sowie vielen weiteren einzeln abrufbaren Wirkungen zusammen.

Software kann nach verschiedenen Kriterien unterteilt werden. Eines davon ist die Frage, ob ein Programm zur Lösung eines spezifischen Realweltproblems¹ dient. Ist ein solcher Anwendungsbezug gegeben, spricht man von *Anwendungssoftware* (Heinrich et al.,

¹ Damit ist ein Problem gemeint, dass nicht dem Computers selbst entspringt bzw. nur in dessen künstlicher Welt existiert, sondern jenseits davon, in der „realen Welt“ besteht.

2004; Stahlknecht & Hasenkamp, 2005). Der Gegenpol ist Software, die problemunabhängig zum Betrieb des Computers benötigt wird. Solche *Systemsoftware* ist allein durch die Existenz des Computers begründet, d. h. ohne Computer gibt es auch die Probleme, die durch Systemsoftware gelöst werden, nicht.

Zur Verdeutlichung: Ein Textverarbeitungsprogramm wie z. B. Microsoft Word ist Anwendungssoftware. Das zu lösende Realweltproblem besteht in der Erstellung von gedruckten oder druckbaren Dokumenten, und zwar unabhängig von der Existenz des Computers. Dieser ist hier nur eine von verschiedenen Möglichkeiten zur Lösung des Problems. Der Windows-Explorer² hingegen ist Systemsoftware, denn seine Aufgabe (im Wesentlichen das Verwalten von Dateien) ist nicht an ein Realweltproblem geknüpft, sondern entsteht allein aus der Existenz des Computers: Ohne diesen gäbe es keine Dateien, die zu verwalten wären. Auch Word manipuliert zwar letztlich Dateien, jedoch ist das hier nicht die Hauptaufgabe, sondern nur Mittel zum eigentlichen Zweck des Programms.

Computersysteme sind üblicherweise mit einer grafischen Benutzerschnittstelle ausgestattet, d. h. die Interaktion mit dem Benutzer erfolgt nicht über den reinen Austausch von Textein- und -ausgaben, sondern über vielfältige Verwendung von grafischen Elementen. Diese sollen über ihren metaphorischen Charakter einen Bezug zur Arbeitsumgebung jenseits des Computers herstellen und eine weitgehend intuitive Bedienung ermöglichen (Heinrich et al., 2004). Die Grund- oder Arbeitsfläche einer solchen Benutzerschnittstelle soll entsprechend den Schreibtisch des Benutzers symbolisieren und wird deshalb als *Desktop* bezeichnet.

Die beiden oben genannten Beispiele für Anwendungs- bzw. Systemsoftware (Word und Explorer) haben gemeinsam, dass ihre Tätigkeit unmittelbar auf dem Desktop für den menschlichen Benutzer sicht- oder erlebbar wird, indem er mit den Programmen interagiert (vgl. Abb. 2.1). Das muss nicht notwendigerweise so sein, denn es gibt auch Software, deren Aktivität für den Benutzer nicht unmittelbar sichtbar ist. Beispiele dafür aus dem Bereich der Anwendungssoftware sind Datenbankserver oder die Verarbeitungslogik eines ERP³-Systems; im Grunde generell die jeweiligen sog. Back-ends einer jeden Client-Server-Architektur⁴. Der Benutzer interagiert jeweils nur mit der Client-Software (dem Front-end), die ihrerseits auf die Dienste des Servers zugreift. Aus dem Bereich der Systemsoftware wären die zahlreichen, im Hintergrund laufenden Dienstprogramme des Betriebssystems zu nennen, z. B. der Spooler (spoolsv.exe), der dafür sorgt, dass Druckaufträge in die Druckerwarteschlange eingereicht und nacheinander an den Drucker gesendet werden.

Unter *Desktopsoftware* sollen alle jene Programme verstanden werden, deren Aktivität durch direkte und regelmäßige, d. h. nicht auf initiale Konfiguration oder außerordentliche Ereignisse beschränkte, Interaktion mit dem Benutzer gekennzeichnet ist. Dazu gehören

² Nicht zu verwechseln mit dem *Internet-Explorer*, einem WWW-Browser.

³ Enterprise Resource Planning

⁴ Eine Erläuterung der Begriffe *Client-Server*, *Back-end* sowie des im Folgenden noch zu benutzenden *Front-end* findet sich z. B. bei Date (2000), dort im Zusammenhang mit Datenbanksystemen.

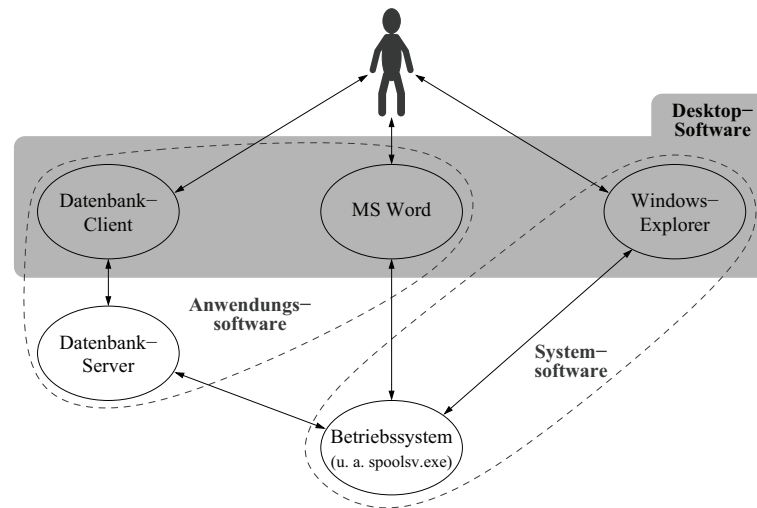


Abbildung 2.1: Desktopsoftware (eigene Darstellung)

nicht nur Stand-alone-Anwendungen⁵, sondern auch die sog. Front-ends (z. B. grafische Benutzerschnittstellen, GUI⁶) von Client-Server-Architekturen. Desktopsoftware ist demnach nicht mit Anwendungssoftware gleichzusetzen und ebenso wenig als das Gegenteil von Systemsoftware zu betrachten (vgl. Abb. 2.1).

Die angegebene Definition mag z. B. durch das Fehlen einer weitergehenden Erklärung, was ein „außerordentliches Ereignis“ ist, unscharf sein. Für die Zwecke der vorliegenden Untersuchung ist sie dennoch ausreichend, denn sie dient nicht dazu, für eine bestimmte Software entscheiden zu können, ob sie Desktopsoftware ist oder nicht. Es geht vielmehr darum, deutlich zu machen, welche prinzipielle Art von Software mit diesem nicht allgemein üblichen Begriff gefasst werden soll.

2.2 Endbenutzer

Ein *Benutzer* ist eine Person, die ein Hilfsmittel, hier konkret Informations- und Kommunikationstechnik, zur Unterstützung bei der Durchführung ihrer Aufgaben verwendet (Heinrich et al., 2004; Stahlknecht & Hasenkamp, 2005; Stickel, 1997). Der Benutzer ist grundsätzlich vom *Anwender* zu unterscheiden.⁷ Letzterer ist in der Regel eine Organisa-

⁵ Mit dem Begriff Stand-alone werden Programme bezeichnet, die ihre Aufgabe eigenständig, d. h. ohne das Vorhandensein weiterer Software, z. B. eines Servers, erfüllen kann.

⁶ Graphical User Interface

⁷ Die beiden Begriffe erscheinen synonym, jedoch ist der zwischen ihnen bestehende konzeptionelle Unterschied für die weiteren Ausführungen wichtig. Da sie in der einschlägigen deutschsprachigen Literatur (z. B. Balzert, 1996; Heinrich, 2002) als Fachbegriffe eingeführt und verwendet werden, werden sie auch in dieser Arbeit so benutzt. In der Praxis sind dagegen zur Bezeichnung des Anwenders stärker abgrenzende Begriffe üblich: „Mir ist die Beschreibung des Anwenders fremd. In Deiner Beschreibung erkenne ich den Kunden, der tatsächlich oft ungleich dem Benutzer ist. Der Kunde kauft die Software, arbeitet

tion (z. B. Unternehmen, Abteilung), die ein Informations- und Kommunikationssystem zur Unterstützung ihrer Aufgaben einsetzt oder einzusetzen beabsichtigt. Der Anwender verantwortet insofern Auswahl, Beschaffung und Einsatz einer Software, der Benutzer interagiert mit ihr. Man könnte auch sagen, dass ein Anwender Software in einem *abstrakten* Sinne einsetzt („Die Universität Rostock verwendet das Prüfungsverwaltungssystem HISPOS“), der Benutzer dagegen *konkret* mit ihr umgeht („Ich muss noch die Klausurergebnisse ins HISPOS eingeben“).

Benutzer werden nach der Art ihrer Tätigkeit häufig in zwei Typen unterteilt (z. B. Stahlknecht & Hasenkamp, 2005; Stickel, 1997)⁸: Zum einen gibt es Spezialisten der Informationsverarbeitung, die mit Aufgaben der Entwicklung, Einführung, Wartung oder des Betriebs von Informationssystemen befasst sind und zu diesem Zweck natürlich auch unmittelbar mit diesen interagieren, also z. B. Softwareentwickler und Administratoren. Der zweite Typ umfasst Personen, die Probleme bearbeiten, zu deren Lösung die von den Spezialisten entwickelte Software dient. Diese Personen werden als *Endbenutzer* bezeichnet (ebd.). Bei ihnen werden im Allgemeinen keine Kenntnisse über die interne Funktion des verwendeten Hilfsmittels vorausgesetzt, wohl aber über die mit dessen Hilfe zu lösende Aufgabe. Das im vorigen Absatz genannte Beispiel des Individuums, welches die Klausurergebnisse einzugeben hat, bezieht sich also auf einen Endbenutzer.

Für den hier zu untersuchenden Sachverhalt kann die Gruppe der Endbenutzer weiter eingegrenzt werden. Es geht um Personen, die selbst die Entscheidung über Anschaffung und Einsatz der von ihr auch selbst zu benutzenden Software treffen, insofern also die Rollen des Benutzers und Anwenders in sich vereinen. In besonderem Maße trifft dies auf sog. Heim- oder Privatanwender zu, die Software nicht im Dienst einer Organisation, sondern nur für ihre persönlichen Belange einsetzen und benutzen. Eine Synonymisierung dieser Bezeichnungen mit dem Begriff des Endbenutzers wäre jedoch für die vorliegende Untersuchung unzutreffend, da Menschen in ihrer Rolle als Mitarbeiter einer Organisation nicht ausgeschlossen werden sollen. Auch für diesen Fall ist es denkbar, dass der Benutzer im Rahmen eines Budgets frei über die zu beschaffende Software entscheiden kann, statt dass ihm die Verwendung bestimmter Software sowohl grundsätzlich als auch hinsichtlich der konkreten Programme „von oben“ diktiert wird.

Der Fokus liegt hier also nicht auf juristischen oder vordergründig organisatorischen Aspekten, also z. B. der Frage, mit wem ein Kaufvertrag zustande kommt, welche Rechte und Pflichten ggf. daraus erwachsen, wie sich deren Wahrnehmung organisiert ist usw. Von Bedeutung ist hier vielmehr der Aspekt, dass der künftige Benutzer (1) selbst den Wunsch hat, Software mit einer bestimmten Funktionalität zu verwenden, und (2) selbst entscheidet, welche von mehreren angebotenen Alternativen er einsetzen will. Der Endbenutzer übernimmt insofern zumindest den inhaltlichen, wenn auch nicht unbedingt den kaufmännischen Teil der Verantwortung des Anwenders im obigen Sinne. Da hier nur der

aber nicht unbedingt selbst damit. Anwender und Benutzer kenne ich nur als synonym“ (Steinmann, 2007).

⁸ Diese Unterscheidung wird jedoch nicht einheitlich vertreten. So führen z. B. Heinrich et al. (2004) den Begriff des Endbenutzers als ein Synonym zum Benutzer auf.

inhaltliche Teil dieser Rolle relevant ist, ist die Verwendung des unter Einschluss weiterer Aspekte ggf. anders zu fassenden Begriffs des Endbenutzers durchaus zu vertreten.

2.3 Interaktive Ad-hoc-Evaluation und -Auswahl

Evaluation gilt als sehr komplexer Begriff, der eine Vielzahl möglicher Verhaltensweisen umfasst und sich somit prinzipiell einer abstrakten, die Wirklichkeit gleichzeitig voll umfassenden Definition entzieht (Wottawa & Thierau, 1998). Im Kontext dieser Arbeit ist Evaluation zu verstehen als die zielbezogene Beurteilung von beliebigen Objekten (z. B. Prozessen, Produkten, Personen) auf der Grundlage eines Systems von Beurteilungskriterien (vgl. z. B. Heinrich et al., 2004). Ein wichtiger Punkt dabei ist der Zielbezug, denn dieser verbindet den Vorgang des Beurteilens mit dessen Anlass sowie seinen Ergebnissen bzw. Konsequenzen. Das bedeutet vor allem, dass eine Evaluation niemals Selbstzweck sein kann, sondern immer einer höheren Aktivität, nämlich irgendeiner Form des Entscheidens untergeordnet ist. Das ist unmittelbar einleuchtend, denn das Ergebnis einer Beurteilung ist selbstverständlich ein Urteil über ein Objekt. Welchen Sinn hat aber ein Urteil, wenn es nicht dazu dient, eine Entscheidung im Zusammenhang mit dem Objekt zu treffen?

Es geht beim Evaluieren also darum, Informationen, die für eine Entscheidung benötigt werden, zu beschaffen und ggf. aufzubereiten, um einem Objekt oder einzelnen seiner Aspekte einen Wert beimessen zu können. Dieser Wert oder diese Werte finden dann Berücksichtigung beim Treffen der Entscheidung. Evaluieren und Entscheiden sind nicht voneinander zu trennende Aktivitäten, es sei denn, die Entscheidung wird impulsiv, gewohnheitsmäßig oder zufällig getroffen. Ebenso ist eine Evaluation eine dem bewussten Entscheiden notwendig vorgelagerte Aktivität: „[E]ssentially, we evaluate in order to decide, that is, to choose between different possibilities“ (Carney & Wallnau, 1998, S. 852).

Das *Auswählen* einer Option aus einer Menge von Alternativen ist eine Form des Entscheidens. Im Grunde kann jedes Entscheidungs- auch als Auswahlproblem aufgefasst werden, auch wenn die Alternativenmenge nur eine Option enthält, denn es gibt stets auch die Möglichkeit, diese eine Option nicht zu wählen. Besteht diese Möglichkeit nicht, gibt es bei nur einer Option auch kein Entscheidungsproblem. Diese abstrakte Sicht soll hier aber in den Hintergrund treten, sodass der Begriff Auswahl stets eine Menge von mindestens zwei Optionen (exklusive Nichtwahl-Option) impliziert.

Damit eine bewusste, d. h. weder impulsive, noch gewohnheitsmäßige oder zufällige Auswahlentscheidung objektiv nachvollziehbar und wiederholbar ist, bedarf es zum einen konkreter Kriterien sowie geeigneter Metriken, d. h. Maßsysteme für die Werte einer Option hinsichtlich dieser Kriterien, zum anderen konkreter Methoden zur Bestimmung und ggf. Aggregation dieser Werte. Um zu einer korrekten Entscheidung zu kommen, ist es darüber hinaus erforderlich, für jede Option aus der Auswahlmenge dieselben Kriterien, Metriken und Methoden anzuwenden. Eine Situation, die die Einhaltung dieser Forderungen weder ermutigt noch erzwingt, lässt dem Individuum die Freiheit, Kriterien, Metriken und Methoden nach Belieben zu variieren und sich von der Situation leiten zu lassen, statt

seinerseits die Situation zu bestimmen. Ein solches Vorgehen kann als *ad hoc* bezeichnet werden.

Ad hoc ist eine lateinische Phrase und bedeutet wörtlich „für dieses“, im übertragenen Sinne „für diesen Augenblick gemacht“ oder „zur Sache passend“ (Duden, 2003, S. 104). In der Umgangssprache bezeichnet sie improvisierte Handlungen (vgl. „aus dem Stegreif“) und Dinge, die speziell für einen Zweck entworfen wurden oder spontan aus einer Situation heraus entstanden sind. Hier ist damit entsprechend ein Vorgehen zur Evaluation von Software gemeint, das nicht zwingend einer bestimmten Methodik folgt und insofern auch nicht wiederholbar ist. Es soll betont werden, dass ein Individuum ohne einen Testplan, ohne konkret spezifizierte Anforderungen, sondern spontan, reaktiv und u. U. mit einer mehr oder weniger diffusen Vorstellung davon, was die Software wie leisten soll, ein Programm auswählt. Gerade bei Endbenutzern, die in der Regel weder über die Fähigkeiten noch die Ressourcen zur Durchführung einer formalen Evaluation verfügen, andererseits aber unmittelbar mit dem durch die Software zu lösenden Problem vertraut sind, weil es ihr eigenes ist, dürfte dies der gängige Modus sein. Voraussetzung dafür ist natürlich, dass das betreffende Individuum überhaupt bereit ist bzw. sich in der Lage sieht, die Prüfung und Auswahl von Software vorzunehmen.

Darüber hinaus erscheint eine Ad-hoc-Evaluation angemessen, wenn die zu beschaffende Software aufgrund ihres Preises oder ihrer strategischen Bedeutung für ein Unternehmen oder eine Privatperson als wenig risikobehaftet angesehen wird. Bei Softwaresystemen, die dagegen eine Investition von mehreren tausend bis Millionen Euro bedeuten, ist dagegen zu erwarten, dass die Anforderungen an die Software genau festgeschrieben sind und die Fähigkeit der angebotenen Produkte, diese Anforderungen zu erfüllen, anhand eines Pflichtenhefts systematisch und mit entsprechendem Aufwand überprüft wird. Zudem zeigen die in Abschnitt 1.1 genannten Download-Portale, dass das Angebot an eher „kleinen“ Softwareprodukten mit überschaubarer Funktionalität sehr groß ist. Dadurch wird es dem Kunden leicht gemacht, auch extreme Ad-hoc-Entscheidungen (z. B. rasches Verwerfen einer Alternative nach nur kurzer Betrachtung) zu fällen.

Die Beschaffung von Informationen über die Alternativen, auf deren Grundlage die Auswahl getroffen werden soll, kann auf verschiedene Weise erfolgen. So könnten z. B. Kollegen oder Freunde nach ihren Erfahrungen und Empfehlungen befragt, Software-Reviews in einschlägigen Zeitschriften nachgelesen oder Produktbeschreibungen beim Hersteller eingeholt werden. Bleibt die Informationsbeschaffung darauf beschränkt, fußt die Entscheidung letztlich auf „Orakelbefragungen“. Der Benutzer *erlebt* die Software dabei aber nicht selbst, sondern unterwirft sich den Wertesystemen Anderer.

Aufgrund des in Abschnitt 1.2 diskutierten besonderen Produktcharakters von Software und der damit verbundenen Möglichkeit des Ausprobierens vor dem Kauf kann die Evaluation von Desktopsoftware mit überschaubarer Funktionalität jedoch auch durch testweise Benutzung erfolgen. Im abstrakten Sinne erfolgt also die Beschaffung der Informationen, auf die sich das Urteil und schließlich die Entscheidung gründet, durch die Interaktion mit den zu evaluierenden Programmen. *Interaktion* beschreibt im Zusammenhang mit Softwarebenutzung einen Betriebsmodus, der durch eine Kommunikation zwischen Mensch und Maschine charakterisiert ist: Der Mensch erteilt der Software Instruktionen, welche darauf mit deren Ausführung und/oder ggf. Hinweisen oder Fehlermeldun-

gen reagiert (vgl. Heinrich et al., 2004). Auf diese Weise kann die Funktionsweise eines Programms erschlossen und der Verlauf bzw. das Ergebnis dieses Erschließungsprozesses zur Grundlage der Bewertung des Programms werden.

Zusammengefasst bedeutet also *interaktive Ad-hoc-Evaluation und -Auswahl*, dass ein Individuum durch spontanes Ausprobieren eine Beurteilung von Software zum Zweck der Auswahl aus mehreren Optionen vornimmt. Dabei wird das Vorgehen nicht durch eine explizit vorgegebene Methode, sondern in erster Linie durch das aktuelle Geschehen, d. h. durch den Verlauf der Interaktion geleitet. Objekt einer solchen Evaluation ist preisgünstige oder sogar kostenlose Software, die dadurch eher den Charakter eines massenhaft vertriebenen Konsum- als eines exklusiven Investitionsgutes hat.

3 Stand der Forschung zur Evaluation und Auswahl von Software

Dieses Kapitel gibt einen Überblick über den Stand der Forschung zur Evaluation und Auswahl von Software. Sowohl die Literatur zum Thema Softwarequalität als auch prä- und deskriptive Studien zur Softwareauswahl werden hinsichtlich ihrer Eignung als Ausgangspunkt zur Klärung der eingangs gestellten Forschungsfragen diskutiert. Besondere Aufmerksamkeit wird dabei dem Technology Acceptance Model und dessen Anwendung auf Auswahlprobleme gewidmet.

3.1 Softwarequalität

Als erster Ansatzpunkt zur Klärung der Frage nach den Gründen für die Auswahl einer Software bietet sich die Beschäftigung mit der Literatur zur Softwarequalität an: Der Begriff der Qualität beschreibt im Allgemeinen die Güte einer Sache. Entsprechend kann Individuen in der Regel eine Qualitätspräferenz unterstellt werden, d. h. dass das Produkt mit der höchsten Qualität favorisiert wird. Wenn also bekannt ist, wonach Individuen die Qualität von Software beurteilen, könnten Erklärungen bzw. Voraussagen möglich sein, warum welches Programm ausgewählt wird.

Dazu ist zunächst zu klären, was Softwarequalität konkret bedeutet bzw. wie sie bestimmt werden kann. Das Schrifttum ist reich an Arbeiten, die sich mit dieser Frage beschäftigen, ohne sie jedoch in allgemein akzeptierter oder gar allgemein gültiger Weise beantworten zu können. Vielmehr zeichnet sich seit einiger Zeit ab, dass es aufgrund der Vielschichtigkeit des Qualitätsbegriffs keine eindeutige Antwort gibt, sondern es immer auf den jeweils vordergründigen Aspekt ankommt: „To determine if one piece of software is of higher quality than another is a difficult task at present; it depends on our idea of quality [...] These views are often conflicting and the term quality cannot be defined in such a way as to encompass these“ (Woodman, 1994, S. 2; vgl. auch Kitchenham & Pfleeger, 1996).

Eine viel beachtete Zerlegung des Qualitätsbegriffs im Allgemeinen wurde von Garvin (1984) vorgenommen, der in der bis dahin existierenden Literatur fünf verschiedene Sichten auf Qualität identifiziert:

- Der *transzendente* Ansatz fasst Qualität als nicht analysierbare, durch Erfahrung aber universell erkennbare Eigenschaft im Sinne inhärenter Exzellenz.

- Der *produktbasierte* Ansatz geht ganz im Gegenteil davon aus, dass Qualität eine präzise messbare Variable ist, die durch die jeweilige Ausprägung von Produktattributen bestimmt wird.
- Auch der *herstellerbasierte* Ansatz ermöglicht die eindeutige Messbarkeit von Qualität, indem er sie als Übereinstimmung mit vorher verabschiedeten Spezifikationen definiert. Jede Abweichung von diesen Anforderungen führt zur Minderung der Qualität.
- Der *nutzerbasierte* Ansatz verortet Qualität am anderen Ende der Wertschöpfung, nämlich im Auge des Betrachters. Die höchste Qualität hat demnach das Produkt, welches am besten die individuellen Bedürfnisse des jeweiligen Benutzers erfüllt. Die Konsequenz daraus ist, dass die Qualität eines Produktes nicht nur von dessen Eigenschaften abhängt, sondern auch davon, wer es wofür benutzen will. Qualität wäre somit ein subjektives und von verschiedenen Faktoren jenseits des Produktes selbst abhängiges Konstrukt.
- Der *wertbasierte* Ansatz schließlich berücksichtigt explizit den Preis, der für ein Produkt mit bestimmten Eigenschaften zu zahlen ist. Je höher der Preis dieses Produktes, desto geringer ist seine wertbasierte Qualität, weil für dieselbe Leistung mehr gezahlt werden muss.

Modelle von Softwarequalität haben die Aufgabe, dieses abstrakte Konstrukt greifbar und damit mess- und vor allem auch kontrollierbar zu machen: „What must be recognized in any attempt to build a quality model is that software does not directly manifest quality attributes. Instead, it exhibits product characteristics that *imply* or contribute to quality attributes“ (Dromey, 1995, S. 146, Hervorhebung im Original). Die ersten Modelle, die zeitlich überdauernd und viel beachteten Eingang in die Literatur gefunden haben, stammen von McCall (McCall, Richards & Walters, 1977) und Boehm (Boehm, Brown, Lipow, MacLeod & Merritt, 1978). Sie folgen einem Faktor-Kriterium-Metrik-Paradigma (Woodman, 1994; vgl. Abb. 3.1), d. h. sie zerlegen das Qualitätskonstrukt (oberste Ebene in Abb. 3.1) zunächst in mehrere *Qualitätsfaktoren* (zweite Ebene in Abb. 3.1), welche wichtige äußere Charakteristika eines Softwaresystems repräsentieren. Beispiele für Qualitätsfaktoren aus dem McCall-Modell (insgesamt 11 Faktoren) sind *Effizienz*, *Zuverlässigkeit* und *Benutzbarkeit*. Jeder dieser Faktoren wird von verschiedenen *Qualitätskriterien* beeinflusst (dritte Ebene in Abb. 3.1), z. B. Zuverlässigkeit durch *Genauigkeit*, *Konsistenz*, *Einfachheit* und *Fehlertoleranz*, die ihrerseits jeweils durch konkrete Metriken (unterste Ebene in Abb. 3.1) messbar sein sollen.

Nach McCall und Boehm gab es weitere ähnliche Modelle (z. B. Deutsch & Willis, 1988; Grady & Caswell, 1987), die sich in ihrer Schwerpunktsetzung und Terminologie unterscheiden, den grundsätzlichen dekompositorischen Ansatz jedoch gemeinsam haben. Eine Vereinheitlichung und Standardisierung der Softwarequalitätsfaktoren erfolgte durch die Norm ISO/IEC 9126-1 (ISO/IEC, 2001), in der sechs *Quality characteristics* (entspricht den Qualitätsfaktoren) mit entsprechenden *Sub-characteristics* festgeschrieben werden.

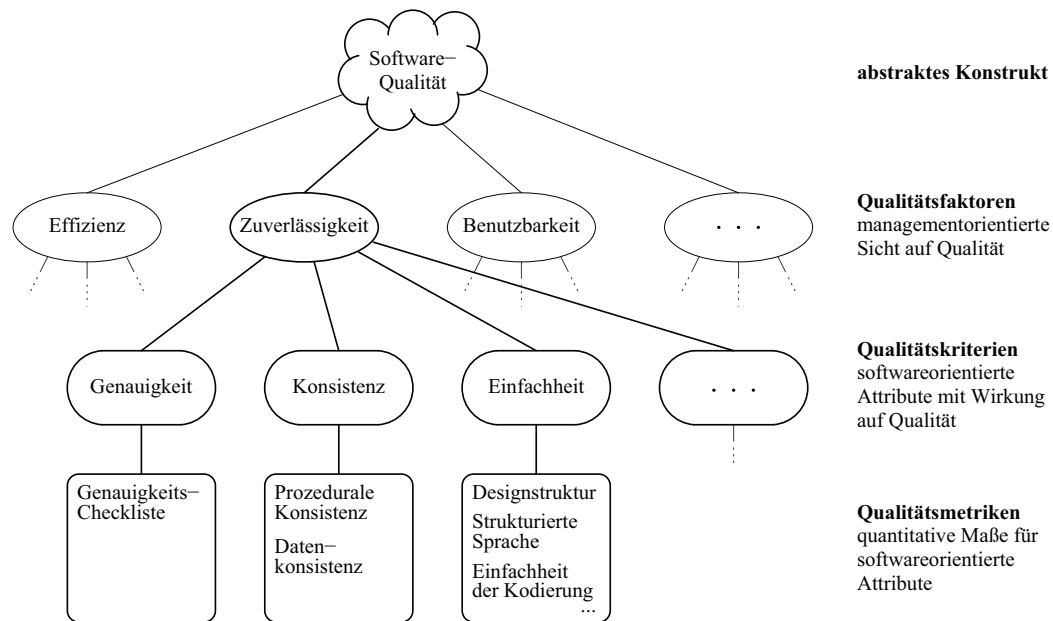


Abbildung 3.1: Qualitätsfaktoren, -kriterien und -metriken im McCall-Modell (eigene Darstellung nach McCall, 1994)

Der Ausgangspunkt der oben beschriebenen Qualitätsmodelle ist stets das *äußere Verhalten*¹ der Software. Die Messung dieses Verhaltens erfolgt dagegen oft über Attribute mit Bezug zu Softwareentwurf und -produktion bzw. erfordert Einsicht in die *innere Struktur* von Prozess und Produkt. Die dabei verwendeten Qualitätsmetriken beziehen sich also auf die *Herstellung*, nicht die *Nutzung* von Software (Kitchenham, Linkman, Pasquini & Nanni, 1997). Insofern ist das in diesen Modellen zum Ausdruck kommende Verständnis von Qualität dem produkt- bzw. herstellerbasierten Ansatz (Garvin, 1984) zuzuordnen. Es geht in erster Linie darum, dem Softwareproduzenten ein Rahmenwerk zu geben, mit dessen Hilfe er die Qualität seiner Produkte überwachen und steuern kann. Das Streben nach Qualität nimmt somit Einfluss auf den Entwicklungsprozess, also die Art und Weise, wie die Softwareherstellung betrieben wird. Softwarequalität wird somit zumindest für den Hersteller vom diffusen Konzept zum überprüfbareren Ziel, auf das konkret hingearbeitet werden kann, um eine vertragliche Verpflichtungen erfüllende bzw. am Markt erfolgreiche Software zu produzieren.

Dieses Verständnis spiegelt sich in der Literatur, d. h. die existierenden Arbeiten befassen sich mit unterschiedlichen Schwerpunkten in der Regel damit, wie Qualität in Software „eingebaut“ (z. B. Dromey, 1995) werden kann. Solche Schwerpunkte sind z. B.:

- die Optimierung des Entwicklungsprozesses hinsichtlich der Berücksichtigung von Qualitätsanforderungen (Beckworth & Altman, 1997);
- der Zusammenhang zwischen den ISO-Qualitätscharakteristika und Konstrukten der Programmiersprache (Dromey, 1995, 1996);

¹ Die dynamischen Aspekte einer Software, d. h. ihre beobacht- bzw. messbaren Aktionen und Reaktionen.

- Integration verschiedener Aspekte bestehender Modelle und Anpassung an ein konkretes Projekt (Kitchenham et al., 1997);
- Zuverlässigkeit und der Zusammenhang zwischen Softwarequalität und Kosten sowie Zeitplan (Musa, 1987);
- die Typisierung von Softwareprojekten zwecks Zuordnung eines geeigneten, standardisierten Qualitätsprofils zur Prozesssteuerung (Musson & Dodman, 1995);
- Effizienz und Effektivität von Prozess und Produkt (Ortega, Pérez & Rojas, 2003);
- Änderbarkeit von Software und ihre Wechselwirkung mit anderen Qualitätsfaktoren (Royce, 1990).

Diese Produktionssicht macht Qualität zu einer inhärenten Eigenschaft von Software und vernachlässigt die Nutzersicht, nach der es einen absoluten, allgemein gültigen Qualitätsbegriff nicht geben kann: „The conventional assumption that quality is an attribute of a product is misleading, as the attributes required for quality will depend on how the product is used“ (Bevan, 1995, S. 115). So lässt sich in der Literatur zur Softwarequalität etwa seit Beginn der Neunzigerjahre ein Trend beobachten, Nutzerzufriedenheit, individuelle Nutzeranforderungen sowie die Abhängigkeit vom Kontext, in dem die Benutzung und Bewertung eines Produkts erfolgt, zu berücksichtigen. Die ISO trägt dem Erfordernis einer Nutzersicht Rechnung, indem sie die ursprüngliche Version der Norm ISO/IEC 9126 um den Begriff der *Quality in Use* erweitert: „Quality in Use is the user’s view of the quality of the software product when it is used in a specific environment and a specific context of use. It measures the extent to which users can achieve their goals in a particular environment, rather than measuring the properties of the software itself“ (ISO/IEC, 2001, S. 5).

Vidgen, Wood-Harper und Wood (1993, S. 102) stellen fest: „Quality is contingent and resides in the user’s perception of the product.“ Die Qualitätswahrnehmung des Nutzers geht jedoch über die technischen Eigenschaften der Software hinaus: „How users *experience* software, affects their opinion about its quality“ (Hellens, 1997, S. 1, Hervorhebung J.Z.). Denning (1992, S. 13) schlug bereits relativ früh vor, „to reframe the question, from ‘What is software quality?’ to ‘How do we satisfy the customers of our software?’ [...] The greater the level of satisfaction, the more likely the customer is to say the software is of good quality.“ Der Nutzer darf nicht zur Abstraktion werden, weil seine Zufriedenheit die Qualität und den Erfolg der Software bestimmt (Denning & Dunham, 2003).

Verbunden damit ist das Verständnis von Softwarequalität im Sinne der Eignung für einen bestimmten Zweck. Dromey und McGettrick (1992, S. 48f) nennen diesen Aspekt „fitness-for-purpose“ und schließen in dessen Definition neben der Erfüllung funktionaler Anforderungen auch ein Maß für die Angemessenheit einer Lösung hinsichtlich des Problemkontextes ein. Das heißt, gute Qualität liegt nicht allein dadurch vor, dass eine Software ihrer Spezifikation entspricht (herstellerbasierte Sicht; vgl. Garvin, 1984), sondern die spezifizierten Anforderungen müssen auch dem entsprechen, was tatsächlich benötigt und gewünscht wird. Dieser Zusatz erscheint trivial, trägt aber dem Umstand Rechnung,

dass der Nutzer oft nicht genau weiß, was er eigentlich will und was überhaupt möglich ist bzw. dass er nicht in der Lage ist, seine Anforderungen in der für den Softwareentwurf benötigten Form auszudrücken (Dahme & Raeithel, 1997). Darüber hinaus können Informationen über Entwurfsziele dadurch verloren gehen, dass sie schlecht oder nicht formalisierbar sind (Rangarajan, Swaminathan, Hedge & Jacob, 2001). Werden solche impliziten Entwurfsziele nicht angemessen berücksichtigt, ist der Nutzer mit der Software unzufrieden. In diesem Sinne erheben Kusters, Solingen und Trienekens (1997) die Nutzererwartungen zur ultimativen Anforderung, stellen aber gleichzeitig fest: „Requirements are not always made explicit by the people involved, but they do expect the system to adhere to them none the less“ (S. 184).

Auch die Literatur zu einem so verstandenen Qualitätsbegriff konzentriert sich jedoch auf die Softwareentwicklung und beschäftigt sich mit der Frage, wie die Erwartungen der Nutzer möglichst genau erfasst und umgesetzt werden können. Das Problem hinsichtlich der Verwendbarkeit dieser Arbeiten für die vorliegende Studie besteht darin, dass es hier gerade *nicht* um maßgeschneiderte, d. h. nach den Spezifikationen des Nutzers hergestellte Software geht. Gegenstand der Untersuchung ist vielmehr die Auswahlentscheidung aus einer Menge generischer Produkte², für die zudem keine weitere Beratung oder Anpassung durch den Hersteller oder ein affiliertes Unternehmen vorgesehen ist. Da die einzelnen konkreten Nutzer solcher Software bei der Entwicklung nicht bekannt, sondern nur als allgemeine Zielgruppe repräsentiert sind, können auch keine individuellen Anforderungen berücksichtigt werden. Die Spezifikation, auf deren Basis eine Standardsoftware entwickelt wird bzw. wurde, spiegelt somit nur die pauschal vermuteten Bedürfnisse dieser Zielgruppe wider und kann insofern nicht Grundlage für das Qualitätsurteil eines konkreten Nutzers sein (Gentleman, 1996).

Zusammenfassend bleibt festzustellen, dass existierende Modelle von Softwarequalität stets zum Ziel haben, diese „irgendwie“ zu messen, sei es zur Steuerung des Entwicklungsprozesses oder zur Ermittlung eines Scores für ein fertiges Produkt. Eine Anwendung für Zwecke der hier interessierenden Evaluation fertiger Produkte aus Kundensicht setzt konkrete und detaillierte, somit genau überprüfbare Anforderungen voraus (z. B. Beus-Dukic & Bøegh, 2003; Carvallo & Franch, 2006; Franch & Carvallo, 2003; Maiden & Ncube, 1998) und ist aufgrund der Komplexität von Software sowie des Qualitätskonstrukts aufwendig und nicht trivial (Blin & Tsoukiàs, 2001; Endres, 2003; Nesbit, 1984). Der Grundgedanke der hier zu untersuchenden Ad-hoc-Evaluation geht aber gerade vom Fehlen einer genauen Spezifikation aus, sodass sie ein Urteil über Softwarequalität allenfalls im transzendenten Sinne (vgl. Garvin, 1984), ansonsten jedoch lediglich *Qualitätserwartungen* im Sinne von Vermutungen hervorbringen kann. Insofern ist Softwarequalität als abstraktes Konstrukt zwar relevant, jedoch können die gängigen Ansätze zu dessen Konkretisierung hier nicht verwendet werden.

² Solche Produkte werden in der Literatur auch als (*Commercial*) *Off-The-Shelf* (COTS) Software bezeichnet. Allerdings fallen darunter nicht nur die hier relevanten, für den Endnutzer gedachten Desktop-Programme, sondern vor allem auch z. B. fremdgefertigte Softwarekomponenten, die in bestehende oder in der Entwicklung begriffene Produkte integriert werden. Eine Diskussion der Verwendung der Terminologie findet sich bei Carney und Long (2000).

3.2 Präskriptive Studien zur Evaluation und Auswahl von Software

Die Literatur zum Thema Softwareauswahl ist umfangreich, hat aber in der Regel präskriptiven Charakter in dem Sinne, dass nicht beschrieben wird, was Individuen bei der Evaluation von Software tatsächlich tun, sondern was sie idealerweise tun *sollten*, um zu einer guten Entscheidung zu kommen. Im Folgenden wird ein Überblick über die Literatur gegeben und jeweils der Bezug zur vorliegenden Untersuchung diskutiert.

Die Evaluation und Auswahl von Software ist eng an den Begriff der Softwarequalität geknüpft: Letztlich kann der Wert, der einem Produkt bei seiner Evaluation zugeschrieben werden soll, als ein Maß für Softwarequalität verstanden werden. Wie oben bereits festgestellt, ist dieses Konstrukt ebenso wie Software selbst jedoch äußerst komplex, so dass dies auch für deren Evaluation gelten dürfte. Der damit verbundene Aufwand scheint gescheut zu werden. Tatsächlich wird z. B. beklagt, dass

- die Softwareauswahl in der Praxis nicht als definierter, wiederholbarer Prozess erfolgt (Kontio, 1996),
- nur oberflächlich bzw. anhand von Produktbeschreibungen und Bestsellerlisten evaluiert wird (Comella-Dorda, Dean, Morris & Oberndorf, 2002; Comella-Dorda et al., 2004),
- Techniken zur Entscheidungsunterstützung naiv und fehlerhaft angewendet werden (Stamelos, Vlahavasa, Refanidisa & Tsoukiàs, 2000).

House (2005, S. 5) berichtet gar von einem Fall, in dem Rivalitäten zwischen verschiedenen Abteilungen eine Rolle bei der Auswahl einer Software spielten: „A surprising number of divisions had an even simpler algorithm – if Division X chose Product A, then we will pick either Product B or Product C, since they’ve never chosen rightly in history.“

Solche Nachlässigkeiten sind umso gravierender, je mehr es nicht um Entscheidungen über individuelle Desktopanwendungen geht, sondern vielmehr ein unternehmensweites Informationssystem angeschafft werden soll, das einen entsprechenden Investitionsumfang, organisatorische Veränderungen und viele Stakeholder³ mit unterschiedlichen Anforderungen bedeutet. Zur Illustration: In einer empirischen Studie über die Entscheidungsfindung österreichischer Unternehmen bei der Auswahl betrieblicher Standardsoftware (z. B. SAP R/3) ermittelten die Autoren eine durchschnittliche Entscheidungsdauer von 26 Wochen und durchschnittliche Kosten der Entscheidungsfindung von fast 64 000 Euro (Bernroider & Koch, 2000). Insofern ist es nicht verwunderlich, dass ein bedeutender Teil der Literatur derartige Beschaffungen mit Investitionscharakter thematisiert.

³ Ein *Stakeholder* ist jede Person oder Gruppe von Personen, die in irgendeiner Weise mit dem zu beschaffenden Informationssystem zu tun hat bzw. haben. Typische Stakeholder im Zusammenhang mit einer betrieblichen Standardsoftware sind z. B. die Endbenutzer, die im Alltag mit dem System zurechtkommen müssen, die Manager, die seine Beschaffung verantworten müssen, die Fachleute der EDV-Abteilung, die es implementieren und warten müssen.

Propagiert werden sehr gründliche und formale, deshalb auch aufwendige präskriptive Vorgehensmodelle, die einerseits dem Gewicht des Problems angemessen sind, andererseits aber auch der Ressourcen einer großen Organisation bedürfen (z. B. Blin & Tsoukiàs, 2001; Boloix & Robillard, 1995; Brenner, 1990; Franch, 2005; Morisio & Tsoukiàs, 1997; Stamelos et al., 2000). Im Vordergrund steht dabei stets, dass Klarheit über die Anforderungen besteht, denn nur dann ist eine formale Evaluation möglich. Berücksichtigt man zudem den präskriptiven Charakter dieser Studien, wird klar, dass sie keinen Beitrag zur Beantwortung der hier zu untersuchenden Fragestellung leisten können.

Weniger formale Ansätze finden sich in der Regel in der älteren Literatur (z. B. Martin & McClure, 1983; Nesbit, 1984; Timmreck, 1973). Sie stammen meist aus der Zeit des (beginnenden) Wechsels vom zentralen Großrechner zu Personal Computern (PC). Damit verbunden war die zunehmende Verfügbarkeit von Standardsoftware für Mikrocomputer und damit das Auftreten des Auswahlproblems. Allerdings scheint es auch zu dieser Zeit weniger von Interesse gewesen zu sein, wie und warum sich Individuen für oder gegen eine Alternative entscheiden, als vielmehr einen Leitfaden für die richtige Auswahl zu liefern, denn die genannten Arbeiten haben den Charakter von Praktikertutorials für Informatiklaien. Der erste Schritt besteht aber, wie bei den oben diskutierten formalen Vorgehensmodellen, auch bei Martin und McClure (1983) sowie Timmreck (1973) in einer detaillierten Anforderungsanalyse. Der Kontext der Arbeiten lässt allerdings schließen, dass sich die Autoren bei ihren Beschreibungen nicht auf Desktopsoftware, sondern auf Beschaffungen mit Investitionscharakter beziehen. So liefern Martin und McClure (1983) ein Muster für einen Softwarenutzungsvertrag, und Timmreck (1973, S. 199) spricht explizit von einem „substantial investment“, was allerdings auch den Verhältnissen zur Zeit der Entstehung seines Artikels geschuldet sein mag.

Nesbit (1984, S. 74) bezieht sich dagegen eindeutig auf „microcomputer software“. Diese ist zwar nicht notwendigerweise mit den hier interessierenden Desktopanwendungen gleichzusetzen, jedoch spricht die Autorin explizit von Produkten mit einem Preis zwischen \$ 30 und \$ 1 000. Darüber hinaus verbindet sie detaillierte Analyse und Vergleich von Anforderungen und Spezifikationen mit einem Aufwand von mehreren Mannmonaten⁴, der eher bei der Auswahl von Software für Großrechner angemessen sei. Ganz im Sinne der Fragestellung der vorliegenden Arbeit stellt sie fest: „The user requirements for microcomputer software are articulated in a very general way. Users are interested in spreadsheet capability or word processing. You may hear a user say he wants a spreadsheet to do departmental budgeting or forecasting, but he’s unlikely to describe his needs precisely“ (ebd., S. 74).

Entsprechend allgemein wird z. B. empfohlen, darauf zu achten, dass „The user interface requires little effort“, „The package must perform useful work“ oder „The software is not frustrating to use“ (ebd., S. 74ff). Timmreck (1973, S. 210) erwähnt, obwohl er eigent-

⁴ Ein Mannmonat, in jüngerer Zeit auch politisch korrekt als Personenmonat bezeichnet, ist ein verbreitetes Maß für den Umfang eines Projektes und beschreibt die Arbeit, die ein Mitarbeiter in einem Monat leisten kann. Ein Projekt im Umfang von 10 Personenmonaten kann demnach, zumindest rein rechnerisch, von einem Mitarbeiter in 10 Monaten, von 10 Mitarbeitern in einem Monat oder von 20 Mitarbeitern in einem halben Monat bearbeitet werden. Warum das in der Realität aber nicht so einfach ist, erklärt Brooks (1995).

lich ein gründliches Vorgehen empfiehlt, dass „the simplest ad hoc approach to computer selection is to simply ‘eyeball’ the proposals, perhaps perform a few simple calculations, and pick the system that looks best or costs least.“ Beide Autoren bleiben aber bei solch generellen Feststellungen und untersuchen nicht weiter, wie es im Zuge einer Ad-hoc-Evaluation tatsächlich zur Entscheidung kommt bzw. welche Faktoren dabei eine Rolle spielen.

In der aktuellen Literatur zur Softwareevaluation und -auswahl sind Arbeiten vorherrschend, die sich mit dem Thema *Softwarekomponenten* befassen. Im Vergleich zu den bereits diskutierten unternehmensweiten Komplettlösungen handelt es sich dabei in der Regel um relativ kleine Softwareprodukte. Insofern kommen solche Studien als Annäherung an die hier interessierenden individuellen Desktopanwendungen infrage. Der Verwendung von Softwarekomponenten liegt der Gedanke zugrunde, einen Mittelweg zwischen der vollständigen Eigenentwicklung einerseits und dem Einkauf eines fertigen Standardproduktes andererseits zu schaffen, indem ein spezifischen Anforderungen genügendes, individuelles Produkt mit geringem Aufwand aus fertig eingekauften Modulen zusammengesetzt wird. Entsprechend ist unter einer Komponente eine unabhängige Softwareeinheit zu verstehen, die über festgelegte Schnittstellen verfügt und ohne Änderungen mit weiteren Komponenten zu einem Softwaresystem zusammengefügt werden kann (für eine tiefere Diskussion des Begriffs, vgl. z. B. Sommerville, 2007).

Es handelt sich dabei also nicht um fertige, für den Endbenutzer gedachte Desktopanwendungen. Softwarekomponenten haben vielmehr den Charakter von Rohstoffen, deren Qualität Einfluss auf das Gesamtprodukt nimmt. Insofern hat eine mangelhafte Auswahlentscheidung hier viel größere Auswirkungen, als es bei einer unabhängigen Desktopanwendung mit ähnlicher Komplexität und Preis der Fall wäre. Dies dürfte der Grund dafür sein, dass auch für die Evaluation und Auswahl relativ kleiner Komponenten aufwendige, formale Modelle entwickelt werden, die nicht nur umfangreiche Analyseaktivitäten vorschreiben, sondern auch grundlegende Expertise auf dem Gebiet des Softwareengineering voraussetzen, welche beim gewöhnlichen Nutzer nicht zu erwarten ist (z. B. Ballurio, Scalzo & Rose, 2002; Beus-Dukic & Bøegh, 2003; Carvallo & Franch, 2006; Comella-Dorda et al., 2002, 2004; Kontio, 1996; Oh, Lee & Rhew, 2003). Insofern helfen auch diese Ansätze bei der Beantwortung der vorliegenden Fragestellung nicht weiter.

Vordergründig mit der Auswahl von Desktopsoftware für oder gar *durch* Endbenutzer befassen sich nur wenige, ebenfalls präskriptiv ausgerichtete Arbeiten, die Softwareevaluation und -auswahl als komplexes sog. Multi-Criteria-Decision-Making (MCDM)-Problem behandeln (z. B. E. E. Anderson, 1989, 1990; E. E. Anderson & Chen, 1997; Le Blanc & Jelassi, 1989, 1994). Ausgangspunkt ist die Feststellung, dass „most users lack the technical expertise or time to perform a requirements analysis of their needs, to translate these ideas into software specifications, to develop criteria, measurements and testing procedures for performance assessment, and to conduct the actual evaluations“ (E. E. Anderson, 1989, S. 707f). Die Folge davon ist, dass sich die Nutzer auf externe Expertenratings verlassen und diese zum Zweck der Entscheidungsfindung auswerten müssen. Das heißt, dass „the decision involves multiple attributes, with end-users often having no explicit formula for maximizing utility. Therefore, an end-user must weigh and evaluate attributes of each software alternative based upon their own implicit value function“ (Le

Blanc & Jelassi, 1994, S. 117). Das Anliegen der verschiedenen Studien besteht in erster Linie darin, Nutzern diese fehlende Formel zur Nutzenmaximierung zur Verfügung zu stellen.

In den Arbeiten von E. E. Anderson (1990) und Le Blanc und Jelassi (1994) werden verschiedene klassische Verfahren⁵ zur Lösung von MCDM-Problemen daraufhin untersucht, welche Option jeweils als Sieger hervorgeht. Dabei zeigt sich, dass die verschiedenen Entscheidungsmodelle zu sehr konsistenten Ergebnissen hinsichtlich der zu favorisierenden Option kommen, wenn durch grobe Vorauswahl ein akzeptables Mindestniveau der Softwareleistung sichergestellt wird. Die Auswahl des Entscheidungsmodells hängt somit in erster Linie von der Entscheidungssituation, insbesondere den verfügbaren Informationen ab. E. E. Anderson (1990) schlägt deshalb Heuristiken vor, unter welchen Umständen oder in welcher Situation welches Verfahren angewendet werden sollte. Berücksichtigt werden dabei Faktoren wie die Fähigkeit des Nutzers, Attribute zu gewichten; seine Bereitschaft, hohe Kosten und Komplexität zu akzeptieren; die Zulässigkeit kompensatorischer Trade-offs⁶; sowie das Vorhandensein von Korrelationen zwischen Attributen. Ein neuer Ansatz zur multiattributiven Quantifizierung des Nutzens eines Produktes findet sich bei E. E. Anderson (1989). Mit Verweis auf Nachteile traditioneller kompensatorischer Verfahren (z. B. Linear Weighted Adding, LWA; s. Abschnitt 4.2.2.2), bei denen Mängel eines Produktes durch starke Leistung bei anderen Kriterien überdeckt werden können, entwickelt er ein Verfahren, das auf paarweisen Vergleichen zwischen jeweils zwei Alternativen basiert sowie das Vorhandensein von Ausreißern berücksichtigt.

Ein bemerkenswerter Aspekt der hier diskutierten Ansätze ist, dass spezifische funktionale Anforderungen anscheinend keine Rolle spielen! Die „uniqueness of each user’s requirements“ (E. E. Anderson, 1989, S. 708) soll durch das Verteilen von Gewichten auf verschiedene Attribute berücksichtigt werden. Das heißt, ein Nutzer kann bestimmen, wie wichtig ihm z. B. die Dokumentation oder grundlegende bzw. fortgeschrittene Funktionalitäten sind. Übereinstimmung darüber, welche das konkret sein sollen sowie deren Vorhandensein wird stillschweigend vorausgesetzt bzw. versteckt sich zusammen mit der Art und Weise der Umsetzung in einer kardinalen Expertenbewertung. Fraglich ist darüber hinaus, ob ein Nutzer in der Lage ist, die verfügbaren Attribute entsprechend zu gewichten, vor allem, nachdem zuvor argumentiert wird, dass ihm die Zeit und Expertise fehlen, seine Anforderungen zu analysieren (E. E. Anderson, 1989). Bereits Timmreck (1973, S. 200) diskutiert in überspitzter Form den grundsätzlichen Sinn, Attribute von Hard- und Software mit Gewichten zu versehen: „There are so many qualitative factors in computer selection that it is a waste of time and money to go through elaborate, rigid procedures. It is absurd to say, for example, that ‘hardware quality’ is twice as important as ‘the availability of sophisticated data management systems’; the two are simply not comparable. Besides, even if scientific procedures could be meaningfully used, they

⁵ Namentlich: Linear Weighted Adding (LWA), Elimination by Aspects (EBA), Linear Assignment Model (LAM), Maximax (MAX), Lexicographic Ordering (LEX), Multi Attribute Utility Theory (MAUT).

⁶ Das bedeutet, dass Schwächen bei einem Attribute durch Stärken bei einem anderen ausgeglichen werden können (vgl. auch Abschnitt 4.2.2.1).

would normally be tools for justification of what the user already knows he wants rather than for true objective selection.“

Ein weiteres Problem ergibt sich aus der Kombination von diffusen und/oder schwer quantifizier- bzw. vergleichbaren Anforderungen einerseits und der leichten Verfügbarkeit bereits vorhandener, von außen vorgegebener Informationen andererseits: „It is much easier to evaluate what you happen to see than taking the effort to determine and search for important characteristics“ (Kontio, 1996, S. 202). Falls oder solange dem Entscheider aufgrund mangelnder Kenntnisse oder Recherchebereitschaft also unklar ist, was wichtig ist, hält er sich an die Kriterien, die ihm von Softwareanbietern bzw. mehr oder weniger unabhängigen Experten als relevant vorgegeben werden. Daraus folgt die Gefahr, bei der Entscheidung vor allem leicht zu messende, bereits fertig quantifizierte oder anderweitig exponierte Aspekte zu berücksichtigen und andere, aber unter Umständen viel wichtigere Attribute zu vernachlässigen.

Als problematisch ist auch die Aggregation der einzelnen Attributwerte zu einem numerischen Gesamturteil mithilfe eines arithmetischen Verfahrens wie LWA anzusehen (Kontio, 1996; Stamelos et al., 2000). Da das Ergebnis dieser Konsolidierung reelle Zahlen sind, werden diese leicht als Werte auf einer Intervall- oder Verhältnisskala interpretiert. Das hieße beispielsweise, dass Produkt A mit einem Score von 5.5 tatsächlich um 10 % besser ist als Produkt B mit einem Score von 5.0. Diese Interpretation ist jedoch nur dann zulässig, wenn jeder der in die Aggregation eingehenden Einzelwerte ebenfalls valide auf einer Intervallskala gemessen wurde. Jedoch dürften in den meisten Fällen schon allein wegen des qualitativen Charakters vieler Attribute lediglich ordinale Likert-Skalen⁷ zur Anwendung kommen. Für die gewichtete Addition sind die verbalen Anker in Zahlen zu übertragen, z. B. „sehr schlecht“ erhält die Wertigkeit 1 (bzw. 1.0), „schlecht“ wird zu 2 (bzw. 2.0) usw. Somit suggeriert die Interpretation der Ergebnisse als Werte auf einer Verhältnisskala eine Urteilsschärfe, die nicht vorhanden ist, und repräsentiert einen verzerrten Sachverhalt, weil tatsächlich lediglich eine Rangfolge wiedergegeben wird (ebd.).

Wie als Reaktion auf solche Kritik beschreiben und testen Le Blanc und Jelassi (1989) am Beispiel der Auswahl von Software zur Entscheidungsunterstützung ein Vorgehen, das die gewichtete Summierung einzelner Attributwerte nur als Zwischenschritt betrachtet und aus drei Phasen besteht: (1) Reduktion des Alternativenpools durch einen Screeningprozess, in dem alle Optionen eliminiert werden, die grobe Mindestanforderungen nicht erfüllen. (2) Detailliertere Evaluation der verbleibenden Kandidaten mittels LWA. (3) Überprüfen des Siegers auf tatsächliche Eignung durch Anwendung auf konkrete Probleme. Dieser letzte Schritt wird wie folgt begründet: „The primary reasons for this stage are to ensure that the DSS⁸ package can be used effectively and to provide one last chance to reconsider the DSS selection. It is often difficult to determine the degree of user satisfaction until [...] utilizing the selected DSS software“ (Le Blanc & Jelassi, 1989, S. 58).

Das Testen von Software ist an sich nichts Besonderes, sondern fester Bestandteil des Softwarelebenszyklus (z. B. Sommerville, 2007). Interessant ist hier, dass kein formales

⁷ Zum Beispiel: „sehr schlecht“ - „schlecht“ - „eher schlecht“ - „mittelmäßig“ - „eher gut“ - „gut“ - „sehr gut“.

⁸ Decision Support System

Testen zum Finden von Fehlern oder zum Prüfen auf Erfüllung vertraglich vereinbarter Anforderungen gemeint ist. Vielmehr soll auf informelle Weise *ausprobiert* werden, ob die theoretisch beste Alternative sich in der praktischen Anwendung auf ein konkretes Problem bewährt. Dieser Gedanke findet sich auch bei McDougall, Squires und Guss (1996), die argumentieren, dass das reine Anwenden von formalen Checklisten zu verzerrten Ergebnissen bei der Softwareevaluation führen kann. Der Grund dafür liege darin, dass auf diese Weise viele subtile Aspekte, die sich nur in der tatsächlichen Benutzung erschließen, nicht erfasst und somit auch nicht berücksichtigt werden. So könnten z. B. verschiedene Textverarbeitungsprogramme nach formaler Evaluation auf dem Papier völlig gleichwertig sein, in der tatsächlichen Eignung für ein bestimmtes Problem aber deutlich divergieren.

Ein weiterer interessanter Aspekt der Probierphase von Le Blanc und Jelassi (1989) ist, dass sie dazu führen kann, ggf. Anforderungen an die Software zu ändern. Anlass dazu könnte sein, dass sich in der Benutzung ein Feature oder eine Funktionalität, an die vorher nicht gedacht wurde, die als wünschenswert herausstellt. Auch der umgekehrte Fall wäre denkbar: Eine bisher geforderte Eigenschaft erweist sich in der Erprobung als nicht so wichtig wie bisher angenommen, sodass eventuell daran gescheiterte Alternativen wieder interessant werden. Franch (2005, S. 563) spricht in diesem Zusammenhang vom „classical SP⁹-based software systems life cycle in which SP selection and requirements elicitation are two complementary activities“. Es ist genau diese Diffusität bzw. Volatilität von Anforderungen, die in der vorliegenden Untersuchung als typisch für die Auswahl von relativ preisgünstiger Desktopsoftware angesehen wird (vgl. auch Nesbit, 1984). Der grundsätzliche Gedanke, dass Softwareevaluation und -auswahl aus verschiedenen Gründen das praktische Ausprobieren zumindest einschließen sollte, ist also in der Literatur durchaus präsent und insofern nicht sonderlich exotisch. Andererseits ist jedoch auch festzustellen, dass der empirischen Beobachtung und Beschreibung dieses Ausprobierens bisher praktisch keine Aufmerksamkeit geschenkt wurde.

3.3 Deskriptive Studien zur Evaluation und Auswahl von Software

Wie oben bereits erwähnt, finden sich in der Literatur kaum Arbeiten, die als deskriptiv in dem Sinne bezeichnet werden können, dass sie beschreiben, was Individuen tatsächlich tun (statt was sie idealerweise tun sollten), um Software zu evaluieren und auszuwählen bzw. welche Faktoren dabei eine Rolle spielen. Zwar wurde und wird Entscheidungsverhalten von Konsumenten an sich ausführlich beforscht (z. B. Kroeber-Riel & Weinberg, 2003, s. auch Kapitel 4), jedoch praktisch nicht speziell für Software. Über die Gründe dafür kann nur spekuliert werden. Jedoch liefern die wenigen existierenden Arbeiten einige interessante Befunde, die für die vorliegende Arbeit, insbesondere den Aufbau der empirischen Untersuchung, von Relevanz sind.

⁹ Software Package

Galletta, King und Rateb (1993) untersuchten den Einfluss von Expertise auf den Verlauf und das Ergebnis der Auswahl eines Datenbankmanagementsystems (DBMS) aus vorgegebenen Alternativen. Im Vordergrund stand dabei die Frage nach der Konsistenz der Evaluation. Damit ist zum einen die Übereinstimmung der Probanden untereinander gemeint, zum anderen der Zusammenhang zwischen den zuvor geäußerten Ansichten eines Probanden hinsichtlich der Wichtigkeit verschiedener Attribute der Software und seinem Auswahlresultat. Die Evaluation lief im Prinzip theoretisch ab, insofern den Probanden nicht die Produkte selbst, sondern Attributlisten mit jeweils extern vorgegebenen Ausprägungen bzw. Bewertungen zur Verfügung standen, auf deren Basis die DBMS in eine Rangfolge gebracht werden sollten. Außerdem wurde für jeden Probanden die individuelle subjektive Wichtigkeit der einzelnen Attribute erhoben, um daraus und aus den Attributausprägungen eine Erwartung hinsichtlich der von ihm bestimmten Rangfolge abzuleiten, d. h. die Rangfolge, die sich bei Konsistenz der Entscheidung zu den selbst festgelegten Attributgewichten ergeben müsste.¹⁰ Die Unterscheidung zwischen Experten und Novizen erfolgte anhand der Erfahrung mit Datenbankprogrammierung sowie mit DBMS-Akquise oder Ausbildung von Anderen in der Nutzung von DBMS.

Die Ergebnisse der Experimente zeigen, dass die Experten zu anderen Auswahlresultaten als die Novizen kommen, untereinander größere Übereinstimmung bei der Vergabe von Attributgewichten erreichen, und doppelt so konsistent wie Novizen bei der Anwendung dieser Gewichte sind. Parallel erhobene Verbalprotokolle lassen den Schluss zu, dass diese Unterschiede dadurch zustande kommen, dass Experten ihre Erfahrung in das Auswahlproblem einfließen lassen und so besser in der Lage sind, Wichtiges von Unwichtigem zu unterscheiden. Ein besonders interessanter Befund ist, dass Markennamen anscheinend keine Rolle bei der Auswahl spielen. Nur ein Proband nahm überhaupt Bezug auf die aufgrund des Produktnamens zu erwartende Leistung.

Kritisch anzumerken ist, dass der Besonderheit von Software als vor dem Kauf uneingeschränkt ausprobierbares Gut keine Rechnung getragen wurde. Das Auswahlproblem wurde im Grunde auf ein theoretisches MCDM-Problem reduziert, da dessen Gegenstand nur durch eine Auflistung von Attributen repräsentiert war. Dennoch kann keine Rede von einem beliebig austauschbaren Gegenstand sein, da bei Software im Gegensatz zu vielen anderen Konsumgütern große Unterschiede in der Expertise der Nutzer bestehen und gezeigt wurde, dass derartige Unterschiede von Bedeutung sind. Insofern sollten die diesbezüglichen Befunde durchaus eine spezielle Relevanz für Software haben. Der Einfluss von Expertise auf eine interaktive Evaluation lässt sich aus der Studie von Galletta et al. (1993) zwar nicht unmittelbar ableiten, jedoch dürften Unterschiede zwischen Experten und Novizen auch hier nicht überraschen.

Das Ausprobieren von Software zum Zweck der Auswahl einer präferierten Alternative steht im Vordergrund einer Untersuchung von Kemp et al. (2002). Gegenstand ist die Entscheidung für eine Software zur Vorbeugung von RSI¹¹. Die Funktionalität solcher

¹⁰ Nach welcher Methode diese geformt werden soll, bleibt jedoch unklar.

¹¹ Repetitive Strain Injury. Bezeichnet „einen pathophysiologischen Mechanismus für tätigkeitsbedingte muskuloskeletale Störungen, die sich v. a. im Unterarm-/Handbereich als chronifizierende Schmerzen, Kraftverlust und Missempfindungen äußern. Der RSI-Mechanismus bezieht sich auf Schädigungen ex-

Programme besteht darin, den Nutzer nach einer gewissen Zeit der Arbeit am Computer daran zu erinnern, eine Pause einzulegen und einige Entspannungsübungen durchzuführen. Solche Unterbrechungen werden leicht als lästig empfunden, weshalb das Hauptziel dieser Studie darin bestand, Nutzerpräferenzen in Situationen zu erforschen, in denen Widerstand gegen die Verwendung bestimmter Software besteht.

Jeder der 24 Probanden (12, die bereits an RSI litten, als Symptomgruppe sowie 12 Gesunde als Kontrollgruppe) wurde vollständig über den Zweck der Untersuchung informiert und benutzte drei von den Autoren ausgewählte Produkte jeweils eine Woche lang. Die Probanden führten ein Tagebuch und bewerteten außerdem jedes Produkt im Anschluss an die jeweilige Testwoche mit einem Fragebogen hinsichtlich der Konstrukte *Affekt* (wie fühlt man sich bei der Benutzung), *Erlernbarkeit* (wie einfach ist es, mit der Software vertraut zu werden), *Hilfe* (wie nützlich sind die Hinweise und Aufforderungen der Software), *Effizienz* (wie effektiv ist die Software hinsichtlich der Erfüllung einer Aufgabe) sowie *Kontrolle* (wie konsistent reagiert die Software auf Kommandos). In einem abschließenden Interview waren die drei Produkte in eine Rangfolge zu bringen.

Ein interessantes Ergebnis ist, dass die Kontrollgruppe zu einer einheitlicheren Präferenz für ein Produkt kommt als die Symptomgruppe. Dieser Effekt könnte einerseits der kleinen Stichprobe geschuldet, andererseits aber Indiz dafür sein, dass der Grad von Involvement¹² Einfluss auf die Entscheidung nimmt. In jedem Fall erreichten die präferierten Produkte signifikant höhere Fragebogenscores hinsichtlich der Konstrukte *Affekt* und *Effizienz*; hinsichtlich der anderen Dimensionen gab es keine Unterschiede zwischen den Produkten. Es scheint also vor allem wichtig zu sein, dass eine Software ihren Zweck erfüllt, ohne den Benutzer zu frustrieren. Auf welche Weise das konkret passiert, scheint nebensächlich zu sein, solange man sich gut dabei fühlt. Das für die Autoren wichtigste Ergebnis war, dass 92 % der Probanden bereit waren, das von ihnen präferierte Programm auch nach der Testphase weiter zu nutzen. Um also Nutzer dazu zu bringen, bestimmte Software zu nutzen, muss man sie selbst herausfinden lassen, welche Alternative am besten ihre Bedürfnisse erfüllt, statt ihnen ein Produkt „von oben“ zu verordnen.

Von der vorliegenden Untersuchung ist diese Studie insofern abzugrenzen, als bei dem hier zu untersuchenden Sachverhalt die Bereitschaft zur Nutzung der evaluierten Software bereits vorhanden ist: Der Benutzer wünscht ausdrücklich eine bestimmte Funktionalität und sucht nach einem Produkt, das diese liefert. Außerdem erhielten die Probanden in der Studie von Kemp et al. (2002) eine Einführung in die verschiedenen Programme. Insofern stellt diese Testphase einen reinen Praxistest dar. Die Notwendigkeit, sich vor der Prüfung der Tauglichkeit einer Software deren Funktionsweise erst einmal zu erschließen, entfällt. Überhaupt spielt in diesem speziellen Fall die Interaktion eine untergeordnete Rolle, denn es geht bei der Evaluation der Programme vor allem darum, wie sie sich in die sonstige Arbeitsumgebung einfügen. Die Interaktion besteht hier in der Kenntnisnahme

trem häufig bewegten Gewebes, die durch ungünstige Kraft- und Haltungsanforderungen verstärkt werden“ (Sorgatz, 2002, S. 1006).

¹² Dieses vor allem in der Konsumentenforschung verwendete hypothetische Konstrukt bezeichnet die „Ich-Beteiligung bzw. gedankliches Engagement und die damit verbundene Aktivierung, mit der sich jemand einem Sachverhalt oder einer Aktivität zuwendet“ (Kroeber-Riel & Weinberg, 2003, S. 345).

und Reaktion auf die Erinnerung, eine Pause einzulegen, und ist insofern, wie bei einem Spiel, bereits das Ziel und nicht, wie z. B. bei einem Zeichenprogramm, der Weg dorthin. Daraus folgt auch, dass sich die Frage nach dem Vorgehen bei der Evaluation weder stellt noch untersucht wurde.

Nochmals aufzugreifen ist der Befund, dass die geäußerten Präferenzen vor allem durch die wahrgenommenen Ausprägungen der Konstrukte Affekt und Effizienz zu erklären sind, weil sich darüber ein Bezug zu Arbeiten von Hassenzahl und Kollegen (Hassenzahl, 2003; Hassenzahl, Beu & Burmester, 2001; Hassenzahl, Burmester & Koller, 2003; Hassenzahl, Platz, Burmester & Lehner, 2000) herstellen lässt, in denen das Zustandekommen von Attraktivitätsurteilen bei der Evaluation interaktiver Software untersucht wird. Postuliert wird, dass dabei über deren Gebrauchstauglichkeit hinaus weitere Aspekte eine Rolle spielen. Diese beruhen auf menschlichen Bedürfnissen nach Stimulation (Streben nach persönlicher Entwicklung) und Identität (Kommunikation des Selbst), während bei Gebrauchstauglichkeit der Bedarf zur kontrollierten Manipulation der Umwelt im Vordergrund steht. Entsprechend wird zwischen aufgabenunabhängiger *hedonischer* und aufgabenbezogener *pragmatischer* Qualität (vgl. Affekt vs. Effizienz) unterschieden.¹³

Aus der Kombination von hedonischen (*HQ*) und pragmatischen Qualitäten (*PQ*) können sich verschiedene Produktcharaktere ergeben. Wünschenswert sind hohe Ausprägungen auf beiden Dimensionen, also ein Produkt, das nicht nur zufrieden stellende Ergebnisse liefert, sondern auch positive Emotionen wie Freude auslöst. Ein schwach hedonisches und stark pragmatisches Produkt kann als handlungsorientiert, ein stark hedonisches und schwach pragmatisches als selbstorientiert bezeichnet werden (Hassenzahl, 2003). Welche Variante letztlich bevorzugt wird, hängt von den individuellen Vorstellungen des Benutzers ab. In verschiedenen Experimenten konnte gezeigt werden, dass die beiden Qualitätsaspekte zu gleichen Teilen das Attraktivitätsurteil hinsichtlich einer Software bestimmen. Die Messung erfolgte über einen Fragebogen in Form eines semantischen Differentials namens *AttrakDiff*, als deren endgültige Items folgende Paare festgelegt wurden (Hassenzahl et al., 2003):

HQ/Stimulation: harmlos – herausfordernd, lahm – fesselnd, phantasielos – kreativ, originell – konventionell, neuartig – herkömmlich, innovativ – konservativ, mutig – vorsichtig;

HQ/Identifikation: ausgrenzend – einbeziehend, bringt mich den Leuten näher – trennt mich von Leuten, isolierend – verbindend, nicht vorzeigbar – vorzeigbar, minderwertig – wertvoll, stilvoll – stillos, fachmännisch – laienhaft;

PQ: praktisch – unpraktisch, widerspenstig – handhabbar, voraussagbar – unberechenbar, verwirrend – übersichtlich, umständlich – direkt, menschlich – technisch, einfach – kompliziert.

¹³ „Hedonisch ist ja eigentlich ein Kunstwort im Deutschen. Es müsste hedonistisch heißen. Ich habe mich ursprünglich für ‘pragmatisch’ und ‘hedonisch’ entschieden, um nicht die ganzen philosophischen Konnotationen dabei zu haben, aber den Konzepten die richtige Ausrichtung zu geben (das hat nicht ganz geklappt). Ja, pragmatisch ist so etwas wie utilitaristisch, nur eben auf interaktive Systeme bezogen. Hedonisch ebenso“ (Hassenzahl, 2004).

In einer früheren Version (Hassenzahl et al., 2000) waren die Items „original – ordinary“ und „innovative – conservative“ (hedonische Qualität) sowie „simple – complex“ (pragmatische Qualität) negativ mit dem jeweils anderen Qualitätsaspekt korreliert. Daraus folgt, dass das Idealbild von hohen Ausprägungen auf beiden Qualitätsdimensionen u. U. nicht erreichbar sein könnte, weil z. B. ein sehr innovatives Produkt (hohe *HQ*) als sehr komplex (niedrige *PQ*), oder ein sehr einfaches (hohe *PQ*) als gewöhnlich und langweilig (niedrige *HQ*) wahrgenommen wird. Dies legt nahe, dass es nicht das eine beste Produkt für alle geben kann, sondern der notwendige Trade-off zwischen *HQ* und *PQ* dazu führt, dass jedes Produkt, das nicht gerade schwach auf beiden Dimensionen ist, stets entweder von mehr handlungsorientierten oder von mehr selbstorientierten Nutzern präferiert wird.

3.4 Das Technology Acceptance Model

3.4.1 Modellentwicklung

Das Technology Acceptance Model (TAM) nimmt in der Literatur zur Softwareevaluation und -auswahl insofern eine Sonderstellung ein, als hier tatsächlich versucht wird zu erklären, warum sich Individuen für oder gegen die Benutzung eines bestimmten Programms entscheiden. Deshalb wird es in einem eigenen Abschnitt behandelt.

Das TAM wurde ursprünglich von Davis (1989) entwickelt und hat seither eine Reihe von Modifikationen und Erweiterungen (z. B. Davis, Bagozzi & Warshaw, 1989; Dishaw & Strong, 1999; Gefen & Straub, 1997; Karahanna & Straub, 1999; Reinicke & Marakas, 2005; Venkatesh, 2000; Venkatesh & Davis, 2000; Venkatesh, Morris, Davis & Davis, 2003) sowie empirischen Überprüfungen (z. B. Adams, Nelson & Todd, 1992; Davis, 1993; Segars & Grover, 1993; Szajna, 1994, 1996) erfahren. Obwohl es bei diesem Modell zunächst um die grundsätzliche Akzeptanz von „Technologie“ im weiteren Sinne statt um die Auswahl von Software aus mehreren Alternativen geht, haben seine Postulate dennoch Relevanz für die vorliegende Untersuchung. Eine prägnante Beschreibung des ursprünglichen Hintergrundes findet sich bei Davis et al. (1989, S. 982): „Computer systems cannot improve organizational performance if they aren't used. Unfortunately, resistance to end-user systems by managers and professionals is a widespread problem. To better predict, explain, and increase user acceptance, we need to better understand why people accept or reject computers.“¹⁴

Es geht also vordergründig um Situationen, in denen über die grundsätzliche Benutzung eines Computers bzw. einer bestimmten Software entschieden wird, welche unverbindlich

¹⁴ Im theoretischen Teil der Arbeiten zum TAM werden Begriffe wie *computer*, *end-user system*, *information system*, *application system*, *computer-based technology* ohne erkennbaren Bedeutungsunterschied verwendet. Die Beschreibungen von Methoden und Designs sowie die Items in den verwendeten Instrumenten beziehen sich dagegen auf *programs* und *software*, z. B. bei Davis et al. (1989) die Textverarbeitungssoftware *WriteOne*. Insofern ist nicht davon auszugehen, dass eine explizite Unterscheidung zwischen Hard- und Software beabsichtigt oder zu berücksichtigen ist. Vielmehr ist stets die notwendige Einheit von Hard- und Software zu betrachten (vgl. Abschnitt 2.1).

angeboten oder „von oben“ gewünscht wird. Die Frage lautet, unter welchen Bedingungen dieses Angebot angenommen wird. Insofern ist die Ausgangslage eine andere als bei der Auswahl einer Software, deren Einsatz grundsätzlich selbst gewollt wird, es also nicht um das *ob*, sondern vielmehr das *was* geht. Wird jedoch berücksichtigt, dass auch die Entscheidung, etwas zu tun oder zu lassen im Grunde eine Auswahl aus Alternativen darstellt (vgl. Abschnitt 2.3), werden die Parallelen offenbar: Die Auswahl eines Programms aus einer Menge von z. B. drei Angeboten kann auch als drei einzelne Entscheidungen über Annahme oder Ablehnung jeweils einer Option aufgefasst werden. Deshalb sind die Determinanten der einfachen Akzeptanz- auch für die multiple Auswahlentscheidung interessant, auch wenn Aspekte wie z. B. direkte Vergleiche zwischen den Alternativen keine Rolle spielen.

Grundlage des TAM ist die Theorie des überlegten Handelns (*Theory of Reasoned Action* (TRA); Ajzen & Fishbein, 1980; Fishbein & Ajzen, 1975). Diese Theorie befasst sich mit der Vorhersage von Handlungen, über deren Ausführung oder Unterlassung eine Person nachdenkt. Die einzige Determinante des Verhaltens ist dabei die Verhaltensabsicht, die Intention (*Behavioral Intention*, *BI*), welche ihrerseits durch die individuelle Einstellung (*Attitude*, *A*) sowie durch *subjektive Normen* (*SN*) hinsichtlich des fraglichen Verhaltens bestimmt wird (vgl. Abb. 3.2). Dabei ist die Wahrscheinlichkeit, dass das Verhalten ausgeübt wird, umso größer, je stärker die Intention ist.

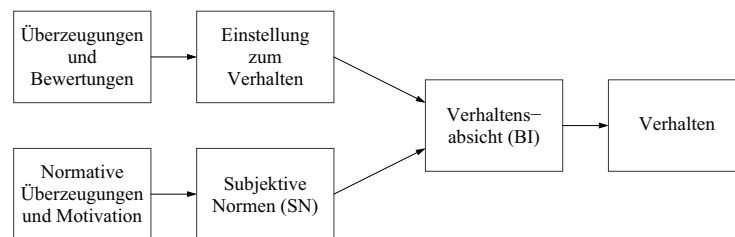


Abbildung 3.2: Theorie des begründeten Handelns (nach Ajzen & Fishbein, 1980, S. 100)

Einstellung wird bei Fishbein und Ajzen (1975, S. 11) als „the amount of affect for or against some object“ verstanden, begleitet von dem Hinweis, dass „affect“ und „evaluation“ synonym verwendet werden. Die aktuelle Forschung definiert Einstellung als eine „psychologische Tendenz, die dadurch zum Ausdruck kommt, dass man einen bestimmten Gegenstand mit einem gewissen Grad an Zustimmung oder Ablehnung bewertet“ (Bohner, 2003, S. 267, nach Eagly & Chaiken, 1998, S. 269). Dieser Gegenstand kann alles sein, was eine Person wahrnehmen oder sich vorstellen kann, z. B. Objekte, Personen (einschließlich sich selbst), Sachverhalte, abstrakte Ideen, soziale Konstrukte usw.

Die Einstellungskomponente der Verhaltensintention beschreibt demnach, ob die Ausführung einer bestimmten Handlung positiv oder negativ bewertet wird. Diese Bewertung hängt von den möglichen Konsequenzen 1..n der fraglichen Handlung ab. So zieht z. B. der Erwerb und Einsatz eines Computers auf jeden Fall einen Geldabgang, hoffentlich eine Arbeiterleichterung, eventuell übermäßigen Aufwand und Ärger bei der Installation und Konfiguration usw. nach sich. Ein Individuum hegt nun für jede dieser Konsequenzen eine subjektive Überzeugung (*belief*, b_i) hinsichtlich ihrer Eintrittswahrscheinlich-

keit, sowie eine positive oder negative Wertschätzung (*evaluation*, e_i). Der Wert einer Konsequenz entspricht im Sinne einer Erwartungs-mal-Wert-Theorie (*Expectancy-Value Models*; z. B. Nerdinger, 1995, S. 87ff) dem Produkt von b_i und e_i . Die endgültige Einstellung gegenüber dem Handeln, hier dem Kauf des Computers, ergibt sich als Summe aller relevanten Überzeugung-mal-Wert-Produkte: $A = f(\sum_{i=1}^n b_i e_i)$. A steht für die Einstellung gegenüber dem fraglichen Verhalten; b_i ist die Überzeugung, d. h. die subjektive Wahrscheinlichkeit, dass das Verhalten die Konsequenz i haben wird; e_i ist die Wertschätzung der Konsequenz i .

Die subjektive Normkomponente der Verhaltensabsicht beschreibt die Wahrnehmung, dass Menschen, die von Bedeutung für eine Person sind, meinen, sie solle das fragliche Verhalten ausführen oder unterlassen. Eine subjektive Norm ergibt sich aus den wahrgenommenen Erwartungen einer Bezugsperson j (*normative belief*, nb_j) multipliziert mit der Motivation, diesen Erwartungen zu entsprechen (*motivation to comply*, mc_j). Die Gesamtheit subjektiver Normen resultiert aus der Summe der Einzelprodukte für jede Bezugsperson: $SN = f(\sum_{i=1}^n nb_i mc_i)$.

Eine Person wird nach der TRA eine Handlung also dann ausführen, wenn sie sie positiv bewertet und wenn sie glaubt, dass für sie bedeutsame Personen dieses Verhalten ebenfalls positiv bewerten würden. Gibt es für die Person keine relevanten Bezugspersonen, so wird die Einstellungs-determinante ein größeres Gewicht bekommen. Andererseits kann eine starke Verankerung der Person in einer Gruppe bewirken, dass der subjektiv erlebte Druck die vorherrschende Verhaltens-determinante darstellt und die Einstellung für die Verhaltensvorhersage irrelevant wird. Einstellungen und subjektive Normen beeinflussen also die Absicht ein bestimmtes Verhalten zu zeigen oder nicht zu zeigen, die Absicht ihrerseits nimmt Einfluss auf die tatsächliche Verhaltensausführung (Frey, Stahl & Gollwitzer, 1993).

Das Technology Acceptance Model (Davis, 1989) übernimmt den grundsätzlichen Wirkungspfad der TRA (Überzeugungen \rightarrow Einstellung \rightarrow Verhaltens-/Benutzungsabsicht \rightarrow Verhalten/Benutzung; vgl. Abb. 3.3), berücksichtigt aber „because of its uncertain theoretical and psychometric status“¹⁵ (Davis et al., 1989, S. 986) keine subjektiven Normen. Außerdem werden explizit zwei ganz konkrete Überzeugungen (Beliefs) festgelegt, nämlich hinsichtlich der wahrgenommenen Nützlichkeit (*Usefulness*, U) sowie der wahrgenommenen Leichtigkeit der Benutzung¹⁶ (*Ease of Use*, EoU). Diese werden als fundamentale und differenzierte¹⁷ Konstrukte behandelt. Darin unterscheidet sich das TAM deutlich von der TRA, wo je nach Gegenstand relevante Attribute bzw. Konsequenzen der Produktbenutzung durch Pretests festgelegt und zu einem Konstrukt aufaddiert werden

¹⁵ Das Problem besteht nach Davis et al. (1989) vor allem darin, dass soziale Normen sowohl direkten (*Compliance*) als auch, wie in der TRA postuliert, indirekten Einfluss (über die Einstellung aufgrund von Identifikations- und Internalisierungsprozessen) auf die Verhaltensabsicht nehmen können (vgl. Aronson, Wilson & Akert, 2004). Diese verschiedenen Einflüsse seien schwierig zu entflechten und würden durch die gängigen Messinstrumente für sozialen Normen nicht unterschieden.

¹⁶ Im Folgenden mit dem Ziel besser lesbarer Formulierungen als „Benutzungsfreundlichkeit“ bezeichnet.

¹⁷ Das heißt, sie sind unabhängig von der konkreten Technologie stets relevant und beschreiben klar voneinander abgegrenzte Sachverhalte.

(Bagozzi, 1990, S. 54f). Wahrgenommene Nützlichkeit ist die subjektive Erwartung des künftigen Benutzers, dass ein bestimmtes Anwendungssystem seine Leistung im Kontext einer Organisation verbessern wird. Wahrgenommene Benutzungsfreundlichkeit ist die Stärke der Erwartung, dass das Anwendungssystem ohne Mühe zu benutzen ist.

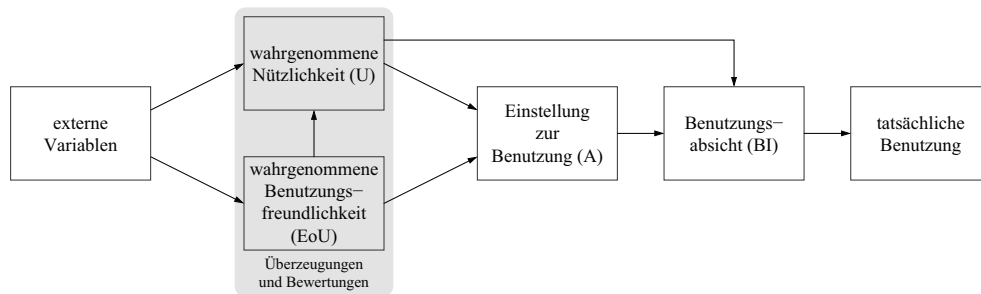


Abbildung 3.3: Das Technology Acceptance Model (nach Davis et al., 1989)

Analog zur TRA postuliert das TAM, dass die Computerbenutzung durch die Benutzungsabsicht (*Behavioral Intention to Use, BI*) determiniert wird, unterscheidet sich aber darin, dass *BI* nicht nur aus der Einstellung gegenüber der Benutzung, sondern auch aus der Nützlichkeit resultiert (vgl. Abb. 3.3). Diesem Zusammenhang liegt der Gedanke zugrunde, dass Menschen in Organisationen Absichten hinsichtlich Verhaltenweisen entwickeln, von denen sie Steigerungen ihrer Arbeitsleistung erwarten, und zwar jenseits ihrer Einstellung gegenüber diesem Verhalten. Über den Einfluss der Nützlichkeit auf die Einstellung hinaus betrachten sie das Handlungsergebnis „gesteigerte Arbeitsleistung“ als instrumentell für gewünschte Ziele, wie z. B. höheres Einkommen, Beförderung, Anerkennung (vgl. Vroom, 1964).

Verbesserungen in der Benutzungsfreundlichkeit wirken nicht nur auf die Einstellung, sondern können durch einen Beitrag zu erhöhter Leistung auch instrumentell wirksam werden: Arbeitersparnis z. B. durch eine geschickt gestaltete Benutzerschnittstelle kann an anderer Stelle investiert werden, sodass das mit demselben Aufwand bewältigte Pensum steigt. In dem Ausmaß, in dem *EoU* zu erhöhter Leistung beiträgt, hat *EoU* einen Effekt auf *U* (vgl. Abb. 3.3). Darüber hinaus können sowohl *U* als auch *EoU* von externen Variablen, z. B. den Charakteristika des fraglichen Anwendungssystems, den Verkaufsbemühungen des Anbieters, durch den Anwender organisierte Schulungen etc. beeinflusst werden¹⁸.

Eine Studie von Davis et al. (1989), die zusammen mit der Arbeit von Davis (1989) als maßgebliches Werk für die Einführung des TAM angesehen wird (Venkatesh & Davis,

¹⁸ Der Einfluss externer Variablen wird auch im Rahmen der TRA diskutiert, findet dort jedoch keine Aufnahme, weil „there is no *necessary* relation between any external variable and a given behavior. Whereas the relations specified in our theory [...] are always assumed to hold so long as appropriate measures are obtained, external variables are not expected to have such consistent effects [...]. If an external variable is found to be related to a given behavior at a given time, it may no longer be related to the behavior at some other time“ (Ajzen & Fishbein, 1980, S. 85, Hervorhebung J.Z.). Deren diesbezüglicher Status im TAM wird nicht expliziert, jedoch erscheint es sinnvoll, auch hier keinen notwendigerweise stabilen und konsistenten Einfluss *bestimmter* externer Variablen anzunehmen.

2000, S. 187), führte zu einer weiteren Vereinfachung, indem das Konstrukt der Einstellung aus dem Modell entfernt wurde. Ursprünglich ging es darum, die Erklärungskraft des TAM für die Benutzung oder Nichtbenutzung einer Software mit der der TRA zu vergleichen. Versuchspersonen waren Studienanfänger an einer amerikanischen Universität, denen auf den Campusrechnern die Textverarbeitungssoftware *WriteOne* zur Verfügung stand.

Zu Beginn des Semesters erhielten die Studenten eine einstündige Einführung in das Programm, an deren Ende (Zeitpunkt t_1) die Konstrukte des TAM (sowie der TRA) außer der tatsächlichen Benutzung gemessen wurden. 14 Wochen später (Zeitpunkt t_2) erfolgte eine zweite Erhebung, diesmal unter Einschluss der tatsächlichen Benutzung, welche als selbst berichtete Benutzung operationalisiert wurde.¹⁹ Wahrgenommene Nützlichkeit und Benutzungsfreundlichkeit wurden über jeweils vier Items wie: „Using WriteOne in the MBA program would increase my productivity“ bzw. „I would find it easy to get WriteOne to do what I want it to do“ gemessen (Davis et al., 1989, S. 991).

Die Ergebnisse zeigten einen starken direkten Einfluss von U auf BI in beiden Zeitpunkten, wogegen der Effekt von A auf BI in t_1 deutlich schwächer und in t_2 nicht signifikant war. Entgegen dem Modell hatte außerdem EoU einen signifikanten direkten Effekt auf BI in t_1 , in t_2 dagegen lediglich auf U . Daraus folgt, dass die Einstellung den Einfluss der Überzeugungen auf die Benutzungsabsicht weitaus weniger und keinesfalls vollständig mediiert, wie im ursprünglichen Modell postuliert. Infolgedessen wird unter Verzicht auf A eine schlankere kausale Struktur eingeführt, die das Benutzungsverhalten mit nur drei theoretischen Konstrukten erklärt und vorhersagt, nämlich BI , U und EoU (vgl. Abb. 3.4).

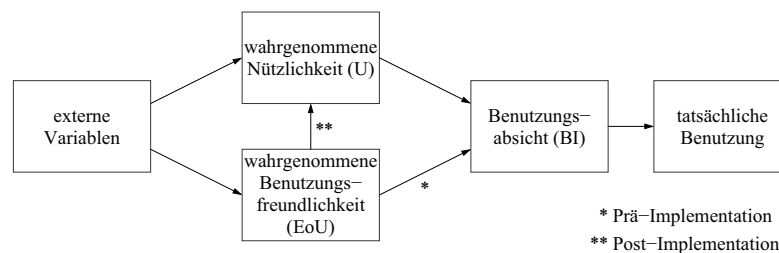


Abbildung 3.4: Das revidierte Technology Acceptance Model (nach Szajna, 1996, S. 87)

Um der veränderten Wirkung von EoU in t_1 und t_2 Rechnung zu tragen, wird darüber hinaus zwischen Prä- bzw. Post-Implementations-Version des TAM unterschieden (vgl. Abb. 3.4). In der Prä-Version, also nach einer kurzen Einführung in die Software ohne eigenständige Benutzung, hat EoU direkten Einfluss auf BI . In der Post-Version dagegen, nachdem der Benutzer Gelegenheit hatte, das Programm im Alltag zu erleben, wird dieser Einfluss durch U mediiert. Davis et al. (1989) erklären dies damit, dass Individuen EoU in t_1 vor dem Hintergrund von Selbstwirksamkeit verarbeiten, d. h. einschätzen, wie wahrscheinlich sie mit der Software zurechtkommen werden. In t_2 , wenn die Bedienung

¹⁹ Zu resultierenden Validitätsproblemen und Konsequenzen s. Szajna (1996).

des Programms durch seine Benutzung erlernt wurde, sind solche Aspekte weniger salient. *EoU* hat nun eine mehr instrumentelle Rolle, d. h. es geht mehr um ihren Beitrag zum effektiven Umgang mit der Software und damit die Frage der Leistungssteigerung, welche ihrerseits, wie oben ausgeführt, den Kernaspekt der Nützlichkeit darstellt.

Als eine der wichtigsten Erkenntnisse für die Frage der Akzeptanz von Softwaresystemen heben Davis et al. (1989, S. 1000) die zentrale Rolle der Nützlichkeit hervor: „Yet our data indicates that, although ease of use is clearly important, the usefulness of the system is even more important and should not be overlooked. Users may be willing to tolerate a difficult interface in order to access functionality that is very important, while no amount of ease of use will be able to compensate for a system that doesn't do a useful task.“

3.4.2 Technology Acceptance Model und Softwareauswahl

Die Aufgabe des Technology Acceptance Model (TAM) ist es, zu erklären bzw. vorherzusagen, unter welchen Umständen eine bestimmte, zur Verfügung stehende Technologie, z. B. eine Software, tatsächlich benutzt oder aber vermieden (werden) wird (Davis et al., 1989). Die Auswahl eines Programms aus mehreren Alternativen kann so interpretiert werden, dass eine Option akzeptiert, die anderen dagegen abgelehnt werden. Insofern ist eine Anwendung des TAM auch auf solche Probleme denkbar.

Eine präskriptive Berücksichtigung der postulierten Zusammenhänge sollte dazu führen, dass das Programm ausgewählt wird, welches eine die Benutzungsabsicht maximierende Kombination von wahrgenommener Nützlichkeit und Benutzungsfreundlichkeit aufweist. Zu diesem Zweck müssten bei den künftigen Benutzern für jede Alternative die Konstrukte Nützlichkeit und Benutzungsfreundlichkeit erhoben und daraus die jeweilige Benutzungsabsicht ermittelt werden. Eine solche Entscheidungshilfe ist allerdings nur in dem hier nicht relevanten Fall sinnvoll, dass Anwender, somit Entscheider und Risikoträger einerseits und Benutzer andererseits nicht dieselbe Person sind (vgl. Abschnitt 2.2). In diesem Fall will der Anwender möglichst sicher gehen, dass sich die Investition auch lohnt, d. h. die von *ihm* ausgewählte und beschaffte Software tatsächlich benutzt wird. Das Bedürfnis nach einer solchen Absicherung entsteht daraus, dass der Anwender die Investition zwar zu verantworten hat, ihr Erfolg aber durch das Verhalten anderer Individuen zumindest mitbestimmt wird.

Im Fall der Auswahlentscheidung durch Endbenutzer (Anwender = Benutzer) dagegen ist die Situation eine andere, denn hier erfolgt die Beschaffung überhaupt erst als Folge der Benutzungsabsicht, welche so stark ist, dass das Individuum bereit ist, dafür Geld auszugeben bzw. dies bei der Organisation, für die es tätig ist, einzufordern. Das zuvor diskutierte Problem der Unsicherheit hinsichtlich der Akzeptanz einer von dritter Seite beschafften Software gibt es insofern nicht: Einerseits ist hier zu erwarten, dass die selbst ausgewählte Software tatsächlich benutzt wird, andererseits ist dies von untergeordneter Bedeutung, da eventuelle Nachteile aus der Nichtbenutzung i. d. R. nur das Individuum selbst zu tragen hat. Dies sollte auch gelten, wenn die Software durch eine Organisation

finanziert wurde, da eine Beschaffung, die auf Betreiben eines Individuums erfolgte und inhaltlich diesem selbst überlassen wurde, wohl kaum strategischen Charakter hat.

In einem deskriptiven Sinne kann das TAM dazu dienen, zu erklären, warum eine bestimmte Software anderen Alternativen vorgezogen wird, nämlich vermutlich genau dann, wenn sie als nützlicher und leichter zu bedienen wahrgenommen wird. In diesem Fall sollte die Benutzungsabsicht für dieses Programm höher als für alle anderen sein, was sich in der Entscheidung für diese Alternative manifestiert. Eine entsprechende Untersuchung wurde von Szajna (1994) vorgenommen. Es ging dabei um die Auswahl eines Datenbankmanagementsystems (DBMS), das im weiteren Verlauf eines Softwareentwicklungsprojektes verwendet werden sollte.

47 Studenten wurden in Gruppen von 4 bis 6 Personen aufgeteilt. Jedem Studenten wurde eines von 9 DBMS zugewiesen, verbunden mit der Aufgabe, sich damit vertraut zu machen²⁰ und in seiner Gruppe vorgegebene, einheitliche Features „seiner“ Software zu demonstrieren. Jedes Individuum hatte somit Kontakt mit 4, 5 oder 6 Programmen und füllte für jedes ein Instrument zur Erhebung von Nützlichkeit (*U*) und Benutzungsfreundlichkeit (*EoU*) (Davis, 1989, angepasst auf DBMS-Kontext) aus. Nach Abschluss der Vorführungen entschied sich jedes Individuum unabhängig von der Gruppe für eine der jeweils präsentierten Alternativen, die es für das Projekt verwenden wollte.

Die Ergebnisse der Untersuchung bestätigen die prädiktive Validität des *U/EoU*-Instrumentes für die Erklärung von Benutzungsverhalten. Bei der Auswertung wurde darüber hinaus auch versucht, mögliche alternative Erklärungen für die Entscheidungen auszuschließen. So konnte gezeigt werden, dass es nicht wahrscheinlicher war, das selbst demonstrierte DBMS zu wählen, sowie dass weder die Reihenfolge der Präsentationen noch die Gruppenzugehörigkeit einen Einfluss auf die Auswahl hatten.

Die Studie von (Szajna, 1994) erlangt besondere Bedeutung für die Erklärung von Technologieakzeptanz aber vor allem dadurch, dass, anders als in früheren Untersuchungen (z. B. Davis et al., 1989), die abhängige Variable hier weder die Benutzungsabsicht noch das selbst berichtete, sondern das *tatsächliche*, objektiv beobachtete Verhalten ist (vgl. auch Szajna, 1996). Dieser Umstand erhöht einerseits die Validität der Untersuchung und trägt andererseits dazu bei, die oben diskutierte Lücke zwischen Anwender und Benutzer zu schließen: Die Auswahl erfolgte durch genau die Individuen, die die Software anschließend auch benutzen sollten. Dies lässt den Schluss zu, dass wahrgenommene Nützlichkeit und Benutzungsfreundlichkeit nicht nur Einfluss darauf haben, ob eine *bestimmte* Software auch benutzt wird, wenn sie ohnehin verfügbar ist. Vielmehr scheinen diese Konstrukte auch zu bestimmen, welche Alternative favorisiert wird, wenn die Möglichkeit der Wahl besteht.

Offen bleibt allerdings der Verlauf und der Einfluss des Erarbeitungsprozesses, d. h. auf welche Weise die Wahrnehmungen von Nützlichkeit und Benutzungsfreundlichkeit zustande kamen, denn die Probanden in der Studie von Szajna (1994) mussten sich die Funktionsweise der Software nicht selbst erarbeiten, sondern bekamen die relevanten Features fertig präsentiert. Dass es einen Unterschied für die Wahrnehmung einer Sache bedeuten kann, ob man erklärt bekommt oder selbst herausfinden muss, wie sie funktioniert,

²⁰ In welchem Umfang und auf welche Weise dies erfolgte, wird nicht berichtet.

ist plausibel. Als Beispiel möge ein Parkscheinautomat dienen: Typischerweise kann dort die gewünschte Parkdauer nicht explizit gewählt werden, sondern ergibt sich automatisch aus dem Betrag des eingeworfenen Geldes. Wenn der Benutzer dies aber nicht weiß, wird er vermutlich analog zu den meisten sonstigen Automaten (z. B. Fahrscheinautomat) erst versuchen, irgendwie einzugeben, was er denn haben möchte, bevor er bezahlt. Da dies nicht gelingt, ist eine negative Wahrnehmung der Benutzungsfreundlichkeit zu erwarten. Diese negative Wahrnehmung resultiert nun aber nicht notwendigerweise aus dem Funktionsprinzip per se, sondern kann auch in seiner mangelhaften Erkennbarkeit begründet sein.

Im Zusammenhang mit Szajnas (1994) Untersuchung könnte der Einwand erhoben werden, dass jeder an dieser Studie teilnehmende Student sich zumindest mit „seinem“ Programm, welches er den Kommilitonen vorstellen sollte, selbst auseinanderzusetzen hatte. Als Gegenargument wäre erstens aufzuführen, dass dies eben immer nur auf *eine* von vier bis sechs Alternativen zutrifft. Zweitens ist nicht bekannt, auf welche Weise, mit welchen Hilfsmitteln oder sonstiger Unterstützung die Erarbeitung erfolgte. So wäre z. B. denkbar, dass ein Student vom Projektleiter eine Kurzreferenz der benötigten Funktionalitäten erhalten hat, also seinerseits mit fertigem Wissen ausgestattet wurde.

Drittens, und das ist der wichtigste Punkt, war jeder Student auf diese eine („seine“) Alternative festgelegt, und hatte nicht die Möglichkeit, bei Problemen auf eine andere Software auszuweichen. Insofern hätte er bei Missfallen nur die Unterlassungsoption gehabt. Diese dürfte jedoch aufgrund der Situation nicht akzeptabel gewesen sein: Es handelte sich um eine verbindliche (im Gegensatz zu der hier zu untersuchenden freiwilligen) Aufgabenstellung in einem Abhängigkeitsverhältnis, d. h. eine Nichterfüllung hätte vermutlich Ernst zu nehmende Konsequenzen, z. B. Verlust des Leistungsnachweises für das Projekt, gehabt. Hinzu kommen weitere Aspekte wie soziale Normen und Ansehen in der Gruppe. Deshalb ist davon auszugehen, dass jene Studenten für die Vorbereitung ihrer Präsentationen mehr Zeit hatten bzw. sich mehr Zeit genommen haben, als sie dies für eine private Ad-hoc-Auswahl getan hätten. Es erscheint jedenfalls plausibel, dass aus den genannten Gründen eine größere Leidens- und Arbeitsbereitschaft vorgelegen hat, als es für den hier zu untersuchenden Sachverhalt zu erwarten ist. Dies könnte dann z. B. dazu führen, dass ein Student nach anfänglichen Problemen die Genialität der Software erkennt; ein Punkt, den er einem unverbindlichen Kontext mit beliebig verfügbaren Alternativen und entsprechend geringerer Bereitschaft, Unzulänglichkeiten hinzunehmen, u. U. nicht erreicht und das Programm vorher abgelehnt hätte.

Dieser Gedanke führt zum Ausgangspunkt dieser Diskussion zurück, denn die jeweils anderen Probanden, denen die Features jenes Programms lediglich demonstriert wurden, sahen nur die Genialität der Software und blieben von den Mühen der Erarbeitung, von denen ein Einfluss auf die Wahrnehmung des Ergebnisses zu erwarten ist, verschont.

3.5 Zusammenfassung und Schlussfolgerung

Die bestehende Literatur zur Evaluation und Auswahl von Software hat überwiegend präskriptiven Charakter, d. h. sie entwickelt und überprüft Methoden, deren Anwendung zur

Feststellung der Softwarequalität bzw. einem optimalen Auswahlergebnis führen soll. Das Spektrum reicht von sehr informellen Leitfäden bis zu sehr elaborierten, mit hohem Bedarf an Ressourcen verbundenen Modellen. Deskriptive Studien beschäftigen sich in erster Linie mit der Frage der Akzeptanz verfügbarer Software. Als wichtige Determinanten werden die Nützlichkeit sowie die Leichtigkeit der Benutzung (Benutzungsfreundlichkeit) identifiziert. Dabei wird jedoch die Kenntnis der Funktionsweise der fraglichen Programme nicht thematisiert oder aber durch Schulungen sichergestellt. Zudem wird generell ignoriert, zumindest aber nicht expliziert, dass Software zunehmend auch als Konsumgut zu verstehen ist. Lediglich einige Forschungen zur Attraktivität von Software kommen zu dem Ergebnis, dass über die Gebrauchseigenschaften hinaus auch sog. hedonische Aspekte (Identifikation und Stimulation) eine Rolle spielen.

Weitgehend offen bleibt die Untersuchung eines *interaktiven* Evaluations- und Auswahlprozesses, z. B. welche Features einer Software in welchem Maße, in welcher Reihenfolge berücksichtigt werden oder ganz allgemein die Frage, was Individuen wie tun, um aus einer Menge verfügbarer Programme durch einfaches Ausprobieren (*hands-on*, im Gegensatz zu *hands-off* auf der Basis von Attributlisten) eine Auswahl zu treffen. Deshalb wird im Folgenden der speziell mit Softwareauswahl befasste Bereich der Literatur verlassen und ein Überblick über relevante Aspekte der Psychologie von Entscheidungen, insbesondere aus dem Bereich der Konsumentenforschung, wo die Frage des Auswahlverhaltens von Individuen eine zentrale Rolle spielt, gegeben.

4 Zur Psychologie der Entscheidung

Dieses Kapitel gibt einen Überblick über ausgewählte Befunde zum Entscheidungsverhalten von Konsumenten. Nach Einführung wichtiger Begriffe und Konzepte, die zum Verständnis des Folgenden nötig sind, werden kognitive und emotionale Prozesse, die für die hier relevante Form des Entscheidens dominant sind, erläutert. Da die Befunde typischerweise für andere Güter als Software erhoben wurden, werden sie jeweils zum hier zu untersuchenden Sachverhalt in Bezug gesetzt.

4.1 Grundlegende Begriffe und Konzepte

4.1.1 Entscheidung, Option, Attribut

Gegenstand der Entscheidungsforschung sind Situationen, in denen aus mindestens zwei Optionen diejenige auszuwählen ist, die einem Individuum den höchsten Beitrag zur Erreichung seiner Ziele verspricht. Jungermann, Pfister und Fischer (2005) sprechen von einer *präferenziellen*¹ Entscheidung, womit gemeint ist, dass eine Option gegenüber der (oder den) anderen präferiert, d. h. vorgezogen wird. Unter *Entscheidung* ist jedoch nicht nur der Moment bzw. das unmittelbare Ergebnis der Präferenzmanifestation zu verstehen, sondern auch der bis dorthin sowie ggf. darüber hinausführende Prozess:

Wir verstehen [...] Entscheidung als einen Prozeß, dessen zentrale Komponenten Beurteilungen (*judgments*) und Wahlen (*choices*) sind. Dieser Prozeß beginnt entweder damit, daß eine Person erkennt, daß es mindestens zwei Optionen gibt; oder er beginnt damit, daß eine Person eine Diskrepanz zwischen dem gegebenen und einem erwünschten Zustand wahrnimmt und dadurch zur Suche nach Optionen veranlaßt wird, die diese Diskrepanz überbrücken könnten. Der Prozeß endet meist, wenn eine Person sich durch die Wahl einer Option festlegt; er endet manchmal aber auch erst mit der Umsetzung der getroffenen Wahl bzw. der retrospektiven Bewertung der Entscheidung. (Jungermann et al., 2005, S. 4, Hervorhebungen im Original)

*Optionen*² können mehr oder weniger materielle Objekte (z. B. Auto, Software, Versicherung), Handlungen (z. B. Kino- vs. Theaterbesuch), Strategien im Sinne allgemeiner

¹ Der Begriff *Präferenz* bezeichnet sowohl den Vorzug oder die Begünstigung einer Alternative als auch die Vorliebe, die ein Individuum für etwas hat.

² Synonym zu *Option* wird sowohl in der Literatur als auch in der Umgangssprache häufig der Begriff *Alternative* verwendet, was jedoch nicht korrekt ist (vgl. Jungermann et al., 2005). „Alternative“ bezeichnet im strengen Wortsinn die Entscheidung zwischen genau zwei sich ausschließenden Optionen, beschreibt

Zielrichtungen, die den Rahmen für konkrete Handlungen festlegen (z. B. konsumieren vs. sparen) oder Entscheidungsregeln (z. B. Münzwurf vs. gründliches Abwägen) sein (vgl. Jungermann et al., 2005). Zu berücksichtigen ist häufig auch die Option, keine Auswahl vorzunehmen, also z. B. weder Auto A noch Auto B, sondern gar keines zu kaufen oder weder ins Kino noch ins Theater zu gehen, sondern zu Hause zu bleiben. In diesem Fall kommt das Beibehalten des Status quo als weitere Option hinzu. Das heißt, selbst wenn scheinbar nur eine Option zur Verfügung steht, sind es in Wirklichkeit oft zwei, nämlich immer dann, wenn das Individuum die Wahl hat, diese zu akzeptieren oder abzulehnen.

Jede Option wird durch eine Menge von *Attributen* beschrieben. Ein Attribut ist als abstrakte Eigenschaft oder Merkmal eines Objektes zu verstehen und nicht mit seiner konkreten Ausprägung zu verwechseln. So kann z. B. ein Auto unter anderem durch das Attribut „Farbe“ beschrieben werden, dessen konkrete Ausprägung z. B. „rot“ ist. Dabei kann jedes Attribut je nach Individuum ein unterschiedliches Gewicht für die Entscheidung haben und in entsprechend unterschiedlichem Maße gegenüber anderen Attributen aufrechenbar sein. Das heißt z. B., dass für Person A der Preis eines Autos wichtiger als die Motorleistung ist, und beides schwerer wiegt als die Farbe, während Person B die Farbe und die Leistung als gleich wichtig und den Preis als unbedeutend ansieht.

Optionen heißen *vergleichbar*, wenn sie durch dieselbe Menge von Attributen beschrieben werden. Die weiteren Ausführungen werden sich auf derartige Entscheidungsprobleme beschränken, denn auch bei dem hier zu untersuchenden Sachverhalt ist die so verstandene Vergleichbarkeit der Optionen gegeben. Dieser Umstand resultiert aus der Annahme, dass bereits die grundsätzliche Entscheidung getroffen wurde, *überhaupt* eine Software zu beschaffen. Es stellt sich somit nur noch die Frage, *welche* es sein soll. *Nicht vergleichbare* Optionen wären beteiligt, wenn zudem weitere, prinzipiell andersartige Möglichkeiten für die Lösung des Problems zur Debatte stünden, wie z. B. die Einstellung eines Mitarbeiters.³

4.1.2 Rationales Entscheiden vs. beschränkte Rationalität

Das im vorigen Abschnitt angeführte Beispiel hinsichtlich der unterschiedlichen Gewichtung von Attributen eines Autos macht bereits deutlich, dass Entscheidungen häufig komplexe Informationsverarbeitungsprozesse zugrunde liegen. So stellt sich für Person A z. B. die Frage, wie viel Geld ihr eine um n Einheiten bessere Motorleistung wert ist. Person B

also nicht die einzelnen Elemente der Auswahlmenge, sondern das Auswahlproblem (vgl. „jemanden vor die Alternative stellen“). Ein vertretbarer Kompromiss mit der Umgangssprache besteht nach Ansicht des Verfassers darin, als Alternative *jede andere Option* außer der gerade betrachteten, die eben alternativ zu jener zur Verfügung steht, zu bezeichnen (vgl. auch Duden, 2003). In Situationen, in denen nur eine Option verfügbar ist, existiert demzufolge keine Alternative und damit auch kein Entscheidungsproblem. Entsprechend wird dieser Begriff hier vor allem verwendet werden, um die neben einer bestimmten Option außerdem verbleibenden zu bezeichnen.

³ Zu Befunden hinsichtlich der Auswahl aus nicht vergleichbaren Optionen (*noncomparable choice*) siehe z. B. Bettman und Sujian (1987); M. D. Johnson (1984).

steht vor dem noch schwierigeren Problem, Motorleistung und Farbe gegeneinander abzuwägen. Die Theorie der rationalen Entscheidung (*Rational Choice Theory*; z. B. Gerrard, 1993b; Hogarth & Reder, 1987; vgl. auch Kroeber-Riel & Weinberg, 2003) geht davon aus, dass der Entscheider ein rational handelndes, nutzenmaximierendes Individuum ist.⁴ Das bedeutet vor allem, dass er über die *Fähigkeit* und *Bereitschaft* verfügt, jeder angebotenen Option einen bestimmten Nutzen beizumessen und jene mit dem höchsten sich so ergebenden Nutzenbeitrag auszuwählen (Bettman, Luce & Payne, 1998). Eine kompakte Zusammenfassung des Grundgedankens der Rational Choice Theory liefert Gerrard (1993a, S. 56f):

The defining characteristic of the logical theory of rational choice is its formality. It reduces choice to a problem of pure logic, a matter of deducing the best means to attain given ends. Furthermore, the theory is structured so as to ensure that the rational choice is conclusive. A well-defined objective function implies the existence of a single optimal choice. The assumption of complete information implies that the optimal choice is identified by the agent. [...] Predicted behavior depends only on the 'objective' properties of the choice situation.

Die im vorigen Absatz genannten Annahmen stehen aber oft im Widerspruch zu den Erfahrungen des Alltags. Weder verfügen Individuen in jedem Fall über wohldefinierte Nutzenfunktionen noch über vollständige Informationen. Insofern ist die Rational Choice Theory ungeeignet, um zu erklären, wie Konsumenten tatsächlich zu ihren Entscheidungen kommen (Bettman, 1979). Ganz im Gegenteil zeigt sich eine *beschränkte* Rationalität, die auf Grenzen hinsichtlich des Arbeitsspeichers und der Verarbeitungsfähigkeiten der Individuen zurückzuführen ist (Selten, 2001; Simon, 1955, 1990): „Because of the limits on their computing speeds and power, intelligent systems must use approximate methods to handle most tasks. Their rationality is bounded“ (Simon, 1990, S. 6).

Hinzu kommt, dass der Mensch besser in der Lage ist, *Unterschiede* zwischen Optionen wahrzunehmen, als einer Option unabhängig von anderen einen bestimmten Wert oder Nutzen zuzuweisen (Tversky & Kahneman, 1991; vgl. auch Abschnitt 4.2.3.3 zu Kontexteffekten), wie es im Sinne rationaler Entscheidung erforderlich wäre. Insofern ist davon auszugehen, dass das Entscheidungsverhalten eines Individuums eben nicht, wenigstens nicht immer, rational ist, sondern maßgeblich sowohl von den Eigenschaften des menschlichen Informationsverarbeitungssystems als auch denen der Entscheidungssituation bestimmt wird (Simon, 1990).

Die Beschränkungen hinsichtlich seiner Informationsverarbeitungskapazität bedeuten vor allem auch, dass der Mensch nicht in der Lage ist, für viele verschiedene Entscheidungsprobleme Präferenzen zu generieren und auf Abruf bereit zu halten (March, 1978).

⁴ Die Rational Choice Theory ist weniger eine konkrete Theorie als vielmehr eine Sammelbezeichnung für verschiedene Ansätze einer Handlungstheorie der Wirtschafts- und Sozialwissenschaften, die den handelnden Subjekten rationales, d. h. nutzenmaximierendes Verhalten zuschreiben. Eine übersichtliche Darstellung des gemeinsamen Grundprinzips als System von Axiomen, die auf der Grundlage einer Nutzenmaximierungsfunktion zu einer entsprechenden Präferenzordnung führen, findet sich z. B. bei Gerrard (1993a).

Daraus folgt, dass Entscheidungsprozesse nicht nur kontextabhängig, d. h. abhängig von den Merkmalen der Entscheidungssituation, sondern vor allem für komplexe Probleme häufig auch *konstruktiv* sind. Das bedeutet, dass Individuen anders als in vertrauten Entscheidungssituationen, mit denen sie Erfahrung haben, nicht über fertige, stabile, wohldefinierte Präferenzsysteme verfügen, sondern diese erst im Zuge der Verarbeitung des Entscheidungsproblems entwickeln oder konstruieren (Bettman et al., 1998).

Aus den in diesem Abschnitt dargelegten Überlegungen folgt, dass das menschliche Entscheidungsverhalten nicht allein durch die Annahme von Rationalität erklärt werden kann. Da das kognitive System des Menschen beschränkt ist bzw. nicht immer voll genutzt wird, kommt es häufig zu Abweichungen des tatsächlichen vom rationalen Verhalten. Die folgenden Abschnitte befassen sich entsprechend mit einigen wichtigen Determinanten menschlichen Entscheidungsverhaltens.

4.1.3 Typologie des Entscheidungsverhaltens

Um die weitere Auseinandersetzung mit psychologischen Aspekten des Entscheidungsverhaltens zu leiten, erscheint die Einordnung des hier zu untersuchenden Problems der Softwareauswahl in eine entsprechende Typologie sinnvoll. Kroeber-Riel und Weinberg (2003, auch zum Folgenden) erklären Entscheidungsverhalten im Rahmen der Konsumentenforschung aus der Verknüpfung insbesondere kognitiver und emotionaler Prozesse. Als potenziell hilfreich können außerdem reaktive Prozesse berücksichtigt werden. *Kognitiv* bedeutet dabei die gedankliche Steuerung der Entscheidung, *emotional* die Aktivierung⁵ und ihre Interpretation und *reaktiv* das automatische Reagieren in der Handlungssituation. Je nach Dominanz eines oder mehrerer dieser Prozesse unterscheiden die Autoren vier Arten von Entscheidungsverhalten (vgl. Abb. 4.1): impulsiv, habitualisiert, limitiert sowie extensiv.

Art der Entscheidung	Dominante Prozesse		
	kognitiv	emotional	reaktiv
extensiv	●	●	
limitiert	●		
habitualisiert			●
impulsiv		●	●

Abbildung 4.1: Dominante psychische Prozesse und Entscheidungsverhalten (nach Kroeber-Riel & Weinberg, 2003, S. 370)

⁵ Aktivierung ist ein zentrales Konstrukt in der Forschungsarbeit von Kroeber-Riel und Weinberg (2003). Darunter wird „ein zentralnervöser Erregungsvorgang verstanden, der mit physiologischen Methoden gemessen werden kann. Die Stärke der in einer Verhaltenssituation vorhandenen Aktivierung bestimmt wesentlich das emotionale Erleben (Aktivierung als Grunddimension von Emotionen [s. Abschnitt 4.3]) sowie die gedanklichen Leistungen der Konsumenten“ (ebd., S. 13).

Extensives Entscheidungsverhalten wird stark kognitiv gesteuert. Diese Steuerung ist umso stärker, je weniger der Konsument über bewährte Entscheidungsmuster verfügt. Eine entsprechend zentrale Rolle spielen Prozesse der Informationsaufnahme und -verarbeitung. Die kognitive Steuerung wird durch emotionale Prozesse angetrieben: Die subjektiv wahrgenommenen Anforderungen an das Entscheidungsverhalten und die Entscheidungsziele, das sog. Anspruchsniveau, aktivieren das Informationsverhalten und werden somit als Regulativ wirksam. Durch kognitive Vereinfachung des Entscheidungsverhaltens kommt es zu *limitierten* Entscheidungen, „die geplant und überlegt gefällt werden und die auf Wissen bzw. Erfahrungen beruhen. Reaktive Prozesse steuern die Informationsverarbeitung also ebenso wenig wie bei extensiven Kaufentscheidungen“ (Kroeber-Riel & Weinberg, 2003, S. 384), bei denen zwischen Reiz und Reaktion eine extensive Informationsverarbeitung stattfindet, die Stimuluswahrnehmung also nicht zu automatischen Handlungen führt. Emotionale Prozesse sind bei limitiertem Verhalten jedoch von untergeordneter Bedeutung. „Das leuchtet sofort ein, wenn man extensive und limitierte Kaufentscheidungen in einer zeitlichen Folge sieht, so dass die Entscheidungssituation weder neuartig noch schwierig ist“ (ebd., S. 384).

Impulsives sowie habitualisiertes Verhalten läuft unter geringer kognitiver Kontrolle ab. Dafür spielen reaktive Prozesse eine zentrale Rolle. Impulsive Entscheidungen sind unmittelbar reizgesteuert (d. h. die Stimuluswahrnehmung führt zu automatischen Handlungen) und werden häufig auch von Emotionen begleitet: „Der Konsument reagiert weitgehend automatisch: Er wählt das Produkt ohne weiteres Nachdenken, einfach deswegen, weil es ihm gefällt, seinen besonderen Vorlieben entspricht“ (ebd., S. 409). *Habitualisiertes* Verhalten dagegen läuft nicht nur gedanken-, sondern auch emotionslos ab. Es basiert auf durch Gewohnheit verfestigten Verhaltensmustern und wird deshalb auch als routinemäßig bezeichnet. Gegenüber limitiertem Verhalten kommt es zu einer weiteren Vereinfachung der Entscheidung, die bei totaler kognitiver Entlastung auch reaktiv gefällt werden kann und dann quasi automatisch abläuft.

Im Zusammenhang mit dem hier zu untersuchenden Problem der Evaluation und Auswahl von Software erscheint eine Beschränkung der weiteren Betrachtung auf extensives und limitiertes Entscheidungsverhalten sinnvoll, da aus verschiedenen Gründen nicht davon auszugehen ist, dass reaktive Prozesse eine bedeutende Rolle dabei spielen: Der wichtigste Grund liegt darin, dass es hier um eine *geplante*, im Gegensatz zu einer spontanen Entscheidung geht, d. h. der Bedarf an einer bestimmten Software ist dem Individuum bereits bewusst, bevor es sich in die Entscheidungssituation begibt. Außerdem wird davon ausgegangen, dass das konkrete Auswahlproblem neuartig ist, d. h. dass keine verfestigten Verhaltensmuster ausgeprägt sind, wie sie für habitualisiertes Entscheiden typisch sind.

Über diese beiden Punkte hinaus sollte ggf. auch die Komplexität der zu evaluierenden Softwareprodukte im Zusammenwirken mit den Konsequenzen des Auswahlergebnisses für die Arbeitsleistung des Individuums dazu führen, dass die Entscheidung unter stärkerer kognitiver Kontrolle abläuft. Entsprechend werden sich die weiteren Ausführungen auf die Diskussion kognitiver und emotionaler Prozesse konzentrieren, welche nach der oben vorgestellten Typologie nach Kroeber-Riel und Weinberg (2003) für extensives und limitiertes Entscheidungsverhalten dominant sind.

4.2 Entscheidungsverhalten und Kognition

4.2.1 Schematheorie

„Der Begriff Kognition wird als Sammelbezeichnung für die geistige Aktivität von Menschen verwendet. In der kognitionspsychologischen Forschung bezeichnet Kognition die Gesamtheit der informationsverarbeitenden Prozesse und Strukturen eines intelligenten Systems“ (Wenninger, 2001, S. 352). Ein grundlegender Baustein für die Erklärung jeder komplexen menschlichen Informationsverarbeitung sind sog. *Schemata* (vgl. zusammenfassend Kroeber-Riel & Weinberg, 2003). Da auch dem Evaluieren von Objekten und dem Treffen von Entscheidungen letztlich Informationsverarbeitungsprozesse zugrunde liegen, soll zunächst dieses zentrale Konzept eingeführt und erläutert werden, um bei der folgenden Würdigung der Literatur zu kognitiven Aspekten menschlichen Entscheidungsverhaltens sowie auch im empirischen Teil der vorliegenden Arbeit auf diesen Abschnitt Bezug nehmen zu können.

Ein *Schema* ist eine Repräsentationsform von vorhandenem Wissen über Objekte oder Handlungen (vgl., auch zum Folgenden, J. R. Anderson, 2007). Genauer gesagt, geht es um *kategoriales* Wissen, das in Form einer Struktur sog. *Slots* (qualifizierte Leerstellen oder Speicherplätze) abgelegt ist. Diese Slots enthalten die typischen Ausprägungen (sog. Default-Werte) verschiedener Attribute für die Mitglieder einer durch ein Schema repräsentierten Kategorie. Hier zeigt sich bereits sehr deutlich die Relevanz des Schemabegriffs für das Analysieren und Bewerten von bzw. Entscheiden zwischen Optionen, da auch diese durch Ausprägungen bestimmter Attribute beschrieben werden können (vgl. Abschnitt 4.1.1).

Jedes Schema verfügt zudem über einen Oberbegriff-Slot, der auf eine übergeordnete Kategorie verweist und so eine hierarchische Verknüpfung der verfügbaren Schemata ermöglicht. Dadurch entsteht eine Generalisierungshierarchie, in der die gespeicherten Konzepte von unten nach oben immer allgemeiner werden. Als Beispiel gibt J. R. Anderson (2007, S. 187) ein mögliches Schema für die Kategorie „Haus“ an:

Haus	
Oberbegriff:	Gebäude
Teile:	Zimmer
Material:	Holz, Ziegel, Stein
Funktion:	Wohnraum des Menschen
Form:	rechteckig, dreieckig
Größe:	10 .. 1 000 m ²

Solange keine widersprüchlichen Informationen vorliegen, übernimmt (erbt) das Haus-Schema die Merkmale seines Oberbegriffs „Gebäude“, z. B. dass es Wände und ein Dach besitzt. Falls ein spezielles Gebäude keine Wände besitzt, z. B. ein offener Carport, muss die vom Oberbegriff abweichende Information im Carport-Schema entsprechend über-

schrieben werden.⁶ Jede Kombination aus Slot und Ausprägung beschreibt dabei ein *typisches* Merkmal. Das heißt z. B., dass das obige Beispiel nicht ausschließt, dass ein Haus auch aus einem anderen als den genannten Materialien gebaut sein kann. Ein Individuum wird sich also beim Anblick eines aus Stahl und Glas erbauten Hauses nicht weigern, dieses als Haus anzuerkennen, denn die gespeicherten Ausprägungen des Material-Slots sind eben nur typisch, aber nicht ausschließlich.

In diesem Zusammenhang ist noch einmal hervorzuheben, dass durch oder in Schemata *keine konkreten* Objekte, sondern nur abstrakte Schablonen gespeichert sind: „Schemata sind Abstraktionen spezifischer Exemplare, die zu Schlussfolgerungen über [konkrete] Exemplare der in den Schemata repräsentierten Konzepte genutzt werden können. Wenn wir wissen, dass etwas ein Haus ist, dann können wir anhand der Schemadefinition erschließen, dass es wahrscheinlich aus Holz oder Stein gemacht ist und dass es Wände, Fenster und Decken besitzt“ (J. R. Anderson, 2007, S. 187f).

Von Bedeutung für die Wechselwirkung zwischen Schemata und der realen Umwelt sind die aus der Entwicklungspsychologie Piagets (1974, 2003) entlehnten Begriffe der Assimilation und Akkommodation. Sofern ausgebildete Schemata zur Verfügung stehen, können bestimmte Erfahrungen und Informationen mit ihrer Hilfe aufgenommen werden. Diese Angleichung der Umwelt an das Individuum heißt *Assimilation*. So wird zum Beispiel ein bestimmtes Objekt, das in die Schablone des Haus-Schemas passt, als Haus eingeordnet. Diese Assimilation bestimmt das Verhalten gegenüber diesem Objekt. So wird ein Individuum z. B. davon ausgehen, dass dieses Objekt durch eine Tür betreten werden kann.

Anders herum kann es erforderlich werden, ungeeignete oder nicht mehr passende Schemata umzuformen, um sie veränderten oder anderweitig neuen Umweltbedingungen anzupassen. Die dabei erfolgende Angleichung des Individuums an die Umwelt heißt *Akkommodation*. So kann z. B. eine Umformung des Haus-Schemas erforderlich werden, um die neue Information abzuspeichern, dass ein Haus auch aus Glas und Stahl errichtet sein kann. Assimilation und Akkommodation bedingen einander, d. h. es gibt keine Assimilation ohne Akkommodation oder umgekehrt. So muss ein Haus überhaupt erst einmal als solches erkannt, d. h. dem Haus-Schema assimiliert werden, bevor dieses z. B. hinsichtlich neuer Materialien akkommodiert werden kann.

Akkommodation und Assimilation sind [...] gegenläufige Prozesse, die einander ergänzen. [...] Bei der Erfassung neuer Zusammenhänge wird erst versucht, die Information mit Hilfe bereits vorhandener Schemata zu assimilieren. Da das Problem auf diese Weise nicht bewältigt werden kann, müssen die Schemata eine Akkommodation an die Umwelt erfahren. Nun kann ein nächster Assimilationsversuch unternommen werden. Unter Umständen ist nun wieder ein neuer Akkommodationsprozeß erforderlich. Dieses Wechselspiel wird so lange fortgesetzt, bis zwischen beiden Prozessen ein Gleichge-

⁶ Dem auf dem Gebiet der Softwareentwicklung erfahrenen Leser wird hier zu Recht die Analogie des Schemabegriffs zum Konzept der Klasse im objektorientierten Paradigma (z. B. B. Meyer, 1990) auffallen: Auch Klassen sind in einer Hierarchie organisiert, sodass untergeordnete Subklassen von ihren übergeordneten Superklassen erben, vorhandene Merkmale ggf. überschreiben und neue hinzufügen.

wicht besteht: das Problem kann assimiliert werden, die nötigen Schemata sind an das Problem akkommodiert. (Oerter, 1968, S. 288; vgl. Piaget, 2003)

Das beschriebene Zusammenwirken von Schemata, Assimilation und Akkommodation ist unmittelbar relevant für das zu untersuchende Problem des Ausprobierens unbekannter Software: Ein, je nach Vorkenntnissen, mehr oder weniger elaboriertes Software-Schema steuert, welche Wirkung von bestimmten Dialogelementen erwartet wird. So dürfte bei den meisten Individuen ein Diskettensymbol mit der Bedeutung „Dokument speichern“ verbunden sein und wird für das aktuell untersuchte Programm entsprechend assimiliert. Bringt die Betätigung dieses Dialogelementes jedoch eine andere Wirkung hervor, muss das (bzw. müssen die) entsprechende(n) Schema(ta) angepasst werden.

4.2.2 Entscheidungsregeln

4.2.2.1 Merkmale von Entscheidungsregeln

Es ist unmittelbar einleuchtend, dass eine Entscheidung umso komplexer und damit umso schwieriger ist, je mehr Attribute zu berücksichtigen sind. Dieser Zusammenhang wird weiter verschärft, wenn diese Attribute hinsichtlich ihrer wünschenswerten Ausprägungen in Konflikt stehen (vgl. Jungermann et al., 2005), z. B. beim Auto die Motorleistung und der Benzinverbrauch. Eines der wichtigsten Ergebnisse der Forschung zum Entscheidungsverhalten von Konsumenten besteht darin, dass diese je nach situativen und individuellen Bedingungen eine Vielzahl verschiedener Regeln (auch Strategien oder Heuristiken genannt) einsetzen, um Entscheidungen zu treffen (z. B. Jacoby, Jaccard, Kuss, Troutman & Mazursky, 1987). In der Konsequenz war eine Vielzahl von Untersuchungen von der daraus erwachsenden Frage geleitet, wodurch diese Regeln charakterisiert werden können, welche Eigenschaften sie haben und durch welche Faktoren ihre Verwendung beeinflusst wird (einen Überblick liefern Bettman et al., 1998). Die in den folgenden Abschnitten diskutierten Befunde sind entsprechend häufig in Bezug zu dieser Frage zu interpretieren. Deshalb soll an dieser Stelle zunächst ein kurzer Überblick über einige wichtige Entscheidungsregeln und die in diesem Zusammenhang verwendeten Begriffe gegeben werden.

Bettman et al. (1998) führen vier Merkmale auf, nach denen sich Entscheidungsregeln hinsichtlich der zugrunde liegenden Informationsverarbeitung charakterisieren bzw. unterscheiden lassen. Ein erstes Kriterium ist die Informationsmenge, auf deren Grundlage die Entscheidung getroffen wird. Dabei ist die Menge der angebotenen oder verfügbaren Informationen von der Menge der tatsächlich verarbeiteten zu unterscheiden, denn das Informationsangebot muss nicht unbedingt vollständig genutzt werden. Unter Umständen kann es dem Entscheider auch reichen, nur einen Teil der verfügbaren Informationen tatsächlich zu verarbeiten. Entsprechend kann die Informationsverarbeitung als *extensiv* bzw. *limitiert* bezeichnet werden. So ist beim Auswählen einer Software mit hinreichend komplexem Funktionalitätsangebot nicht in jedem Falle davon auszugehen, dass eine vollständige (extensive) Evaluation erfolgt. In Abhängigkeit von weiteren Fak-

toren kommt stattdessen eine beschränkte (limitierte) Berücksichtigung lediglich einiger besonders wichtiger Aspekte infrage.

Das zweite Merkmal beschreibt, inwiefern allen angebotenen Optionen die gleiche Aufmerksamkeit hinsichtlich der Berücksichtigung verfügbarer Informationen zukommt. Wird für jedes Attribut bzw. jede Option die gleiche Informationsmenge verarbeitet, heißt die Regel *konsistent*, anderenfalls *selektiv*. Hat ein Individuum z. B. drei Programme zum Zeichnen von Diagrammen zur Auswahl, könnte es für alle drei stets nur genau dieselben Funktionalitäten untersuchen. In diesem Fall geht es konsistent vor. Prüft es dagegen bei Programm A nur unbedingt benötigte Funktionalitäten, und vernachlässigt diesen Teil bei Programm B zugunsten interessanter „Spielereien“, arbeitet es selektiv.

Drittes Unterscheidungskriterium ist, ob die Informationsverarbeitung entlang der einzelnen Optionen oder der einzelnen relevanten Attribute erfolgt: Werden zunächst alle interessierenden Attribute einer einzigen Option untersucht, bevor die nächste an die Reihe kommt, heißt die zugrunde liegende Regel *alternativenbasiert*. Erfolgt die Verarbeitung in der Form, dass die verschiedenen Werte eines einzigen Attributs über alle Optionen hinweg betrachtet werden, heißt sie *attributbasiert*. Ermittelt zum Beispiel der Software-Entscheider zunächst für alle Programme, wie ein Symbol erzeugt wird, dann für alle Programme, wie ein Pfeil erzeugt wird, dann für alle Programme, wie ein Symbol beschriftet wird usw., geht er attributbasiert vor. Bildet er sich über alle benötigten Funktionalitäten ein Gesamturteil über Programm A, dann ebenso ein Gesamturteil für B und schließlich für C, arbeitet er alternativenbasiert. Beide Vorgehensweisen müssen nicht über den gesamten Entscheidungsprozess in Reinform auftreten; auch Mischformen sind denkbar.

Das vierte und letzte Merkmal beschreibt, ob eine Regel Trade-offs zwischen Attributen erlaubt: Kann ein schlechter Wert auf einem Attribut durch einen guten Wert auf einem anderen ausgeglichen werden, heißt die zugrunde liegende Regel *kompensatorisch*. Verbunden damit ist stets die Überlegung, wie viel „schlecht“ des einen Attributs für wie viel „gut“ des anderen in Kauf zu nehmen ist. Erlaubt die verwendete Regel keine solchen Trade-offs, heißt sie *nicht-kompensatorisch*.

Die Anwendung von Entscheidungsregeln erfolgt sowohl interindividuell als auch in Abhängigkeit von der Situation sehr unterschiedlich. Selbst im Verlauf eines einzigen Entscheidungsprozesses kommt es zur Anwendung mehrerer verschiedener Strategien oder Heuristiken (Jacoby et al., 1987). So identifiziert z. B. Payne (1976) im Verlauf eines Entscheidungsprozesses zwei typische Phasen: Zunächst wird das Entscheidungsproblem reduziert, indem meist durch eine Regel mit limitierter Informationsverarbeitung einige Optionen ausgeschlossen werden (vgl. auch Otnes, Lowry & Shrum, 1997). Anschließend werden die verbleibenden Optionen gründlicher untersucht. Konsistent dazu beobachten Bettman und Park (1980), dass Konsumenten in der ersten, der Eliminierungsphase eher attributbasiert, in der zweiten eher alternativenbasiert evaluieren.

Eine wichtige Rolle hinsichtlich der Frage, ob bzw. welche Entscheidungsregeln in einer bestimmten Situation angewendet werden, spielen die jeweils verfolgten Metaziele (Bettman et al., 1998; vgl. auch Jungermann et al., 2005). Die wichtigsten Metaziele sind nach Bettman et al. (1998):

- die Minimierung des kognitiven Aufwands bei der Entscheidung,

- die Maximierung der Genauigkeit der Entscheidung,
- die Minimierung negativer Emotionen während der Entscheidung,
- die Maximierung der Rechtfertigbarkeit der Entscheidung.

Der Zusammenhang zwischen Zielen und Regeln ergibt sich dadurch, dass jedes Individuum je nach Erfahrung über ein spezifisches Repertoire verschiedener Strategien zur Lösung von Entscheidungsproblemen verfügt. Jede dieser Strategien hat in Abhängigkeit von der Situation Vor- und Nachteile im Hinblick auf das Erreichen bestimmter Ziele. Je nach Situation dominieren unterschiedliche Ziele oder auch Kombinationen von Zielen, was zur Anwendung dazu passender Strategien führt. So sollte z. B. bei dominantem Streben nach Minimierung des kognitiven Aufwands eher eine Entscheidungsregel mit limitierter statt extensiver Informationsverarbeitung angewendet werden.

4.2.2.2 Spezifische Entscheidungsregeln

In diesem Abschnitt werden einige wichtige spezifische Entscheidungsregeln vorgestellt. Da im Verlauf der vorliegenden Arbeit an verschiedenen Stellen Bezug auf solche Regeln genommen wird, erscheint es sinnvoll, an dieser Stelle einen kurzen Überblick zu geben, um eine entsprechende Einordnung zu ermöglichen.

Eine Entscheidungsregel, die vor allem in der präskriptiven Entscheidungstheorie eine zentrale Rolle spielt, ist die MAU-Regel (*Multi-Attribute Utility*; z. B. Jungermann et al., 2005). Ihre Bedeutung für die Diskussion psychologischer Modelle besteht vor allem darin, dass sie dem Grundgedanken von Rational Choice (vgl. Abschnitt 4.1.2) folgt und deshalb als Referenz zur Einordnung anderer Strategien dienen kann (z. B. Bettman et al., 1998, dort unter dem Oberbegriff *Weighted Adding*). Diese Regel sieht vor, jeder angebotenen Option einen Gesamtnutzenwert zuzuweisen, der sich aus der Summe der gewichteten Partialnutzenwerte der einzelnen Attribute ergibt. Gewählt wird dann die Option mit dem höchsten Gesamtnutzen. Voraussetzung ist, dass der Entscheider sowohl eine Vorstellung über die Wichtigkeit der einzelnen Attribute hat als auch in der Lage ist, Nutzenwerte für die Ausprägungen dieser Attribute zu ermitteln.⁷ Die MAU-Regel verarbeitet mithin potenziell große Informationsmengen, ist konsistent, alternativenbasiert und kompensatorisch.

Es ist klar, dass diese Strategie bereits bei wenig komplexen Entscheidungsproblemen hohe Anforderungen an die kognitiven Ressourcen stellt. Im Gegenzug werden bei validem Input (Gewichtungssystem und Attributwerte) aber sehr genaue Ergebnisse geliefert. Etwas aufgeweicht werden die hohen Anforderungen durch die *Equal-Weight*-Strategie (Payne, Bettman & Johnson, 1993), bei der auf die Gewichtung verzichtet und entsprechend jedem Attributwert das gleiche Gewicht zugewiesen wird. Trotz der Vereinfachung liefert diese Regel oft noch hinreichend genaue Ergebnisse (Dawes, 1979).

⁷ Zu weiteren Einschränkungen s. Jungermann et al. (2005).

Die Berücksichtigung beschränkter Rationalität dagegen zeigt sich in Regeln mit heuristischem⁸ Charakter, die davon abrücken, dass stets jede verfügbare Information bei der Entscheidungsfindung verwendet wird. Primitive Grundformen sind die *Dominanz-*(DOM), die *Konjunktions-*(CON) und die *Disjunktionsregel* (DIS; vgl., auch zu allen anderen Regeln, Jungermann et al., 2005). Bei DOM wird diejenige Option gewählt, die auf allen Attributen mindestens so gut wie und auf mindestens einem Attribut besser als alle anderen Optionen ist. Ist keine Option in diesem Sinne dominant, liefert die DOM-Regel kein Ergebnis. CON und DIS erfordern die Festlegung eines Schwellenwertes für die Ausprägung eines jeden Attributes, der erreicht oder übertroffen werden muss, damit die betreffende Option infrage kommt. Bei CON wird dann diejenige Option gewählt, die auf allen Attributen den Schwellenwert erfüllt, bei DIS jene, welche den Schwellenwert für mindestens ein Attribut erreicht.

Häufig diskutierte komplexere Entscheidungsregeln sind die *lexikografische Ordnung* (LEX), *Satisficing* (SAT) und *Elimination by Aspects* (EBA). Die LEX-Regel geht von einer ordinalen Gewichtung der Attribute aus. Das Vorgehen ist attributbasiert und wählt diejenige Option, die für das wichtigste Attribut den höchsten Wert erreicht. Kommt es dabei zu einem Unentschieden, entscheidet zwischen den verbleibenden Optionen der höchste Wert des jeweils nächst wichtigen Attributs, bis nur noch eine Option übrig bleibt.

Die SAT-Regel wurde von Simon (1955) beschrieben und basiert auf der Überlegung, dass dem Entscheider oft bereits ein befriedigendes statt eines optimalen Ergebnisses ausreicht. Im Gegensatz zu den bisher betrachteten Regeln wird hier keine Gewichtung der Attribute benötigt, sondern es ist analog zu DIS/CON für jedes Attribut ein Mindestniveau der zu erreichenden Wertausprägung festzulegen. Die Verarbeitung erfolgt alternativenbasiert in beliebiger Reihenfolge. Es wird jeweils geprüft, ob die aktuelle Option bei allen Attributen den Mindestwert erreicht. Falls nein, wird die Option abgelehnt und die nächste untersucht. Falls ja, wird die Option ausgewählt und der Entscheidungsprozess beendet, sodass ggf. einige Optionen unbetrachtet bleiben. Darin besteht auch der wesentliche Unterschied zu CON: Aus dem Entscheidungsprozess geht immer nur maximal eine befriedigende Option hervor, nämlich jene, die als Erste für alle Attribute die gesetzten Schwellenwerte erreicht.

Eine Kombination von Elementen der beiden vorgenannten heuristischen Strategien verwendet die von Tversky (1972) vorgeschlagene EBA-Regel. Jede Option wird als ein Bündel von sog. *Aspekten* betrachtet, die sich in den Attributwerten manifestieren. Benötigt wird wie bei LEX eine ordinale Gewichtung und wie bei SAT für jedes Attribut (bzw. jeden Aspekt) ein Schwellenwert. Beginnend mit dem höchstgewichteten Attribut werden nun alle Optionen eliminiert, die für dieses Attribut das Mindestniveau nicht erreichen. Verbleibt mehr als eine Option in der Auswahlmenge, wird die Prozedur mit dem jeweils nächst wichtigen Attribut solange wiederholt, bis nur noch eine Option übrig bleibt.

Ein Strategieansatz unter der Bezeichnung *Additive Difference* betrachtet Unterschiede zwischen den Attributausprägungen der Optionen und rechnet diese Unterschiede gegen-

⁸ Eine Heuristik ist eine einfache (weil evolvierte und erlernte Fähigkeiten nutzende) und intelligente (weil Umweltstrukturen nutzende) Regel, die den Prozess einer Problemlösung beschreibt (Gigerenzer & Gaissmaier, 2006).

einander auf (Wright & Barbour, 1977; Lussier & Olshavsky, 1979; Bettman & Zins, 1979). Eine Version dieses Ansatzes, *Majority of confirming Dimensions*, führt paarweise Vergleiche von Optionen durch (Russo & Doshier, 1983). Die Option, die für die meisten Attribute besser abschneidet, kommt eine Runde weiter und tritt gegen die nächste bisher unbetrachtete Option an. Eine Verminderung der kognitiven Last ist hier insofern gegeben, als keine Notwendigkeit besteht, Attributausprägungen einen konkreten Zahlenwert zuzuweisen. Die verarbeitete Informationsmenge ist zwar insgesamt extensiv, wird aber durch die Organisation als Sequenz einfacher ordinaler Vergleiche zwischen jeweils nur zwei Attributausprägungen handhabbarer.

Ein wichtiger Aspekt im Zusammenhang mit Entscheidungsstrategien liegt darin, dass die Wahrnehmung des Gesamtwertes einer Option immer auch von den konkurrierenden, ebenfalls in der Auswahlmenge enthaltenen Optionen abhängt. Bereits Tversky (1972) zeigt, dass die Annahme simpler Skalierbarkeit basierend auf dem Prinzip der Unabhängigkeit der Präferenz von irrelevanten Alternativen⁹ inadäquat ist, sondern die Auswahlwahrscheinlichkeit einer bestimmten Option vielmehr auch von der Gesamtheit der verfügbaren Optionen abhängt. Dies ist vor allem dadurch plausibel, dass der in einigen der vorgenannten Ansätze (DIS, CON, SAT, EBA) benötigte Schwellenwert nicht immer per se bekannt sein, sondern sich insbesondere bei neuartigen Entscheidungsproblemen erst durch Inspektion der verfügbaren Optionen bilden wird.

Die explizite Berücksichtigung der Relationen zwischen Optionen bzw. deren Wahrnehmung führt zu Heuristiken mehr perzeptueller Natur (Simonson & Tversky, 1992). So ermittelt zum Beispiel das *Componential Context Model* (Tversky & Simonson, 1993) die Gesamtbewertung einer Option als Summe der gewichteten Nutzenbeiträge der einzelnen Attribute *plus* der Summe des relativen Vorteils gegenüber den konkurrierenden Optionen. Es ist jedoch zu vermuten, dass eine tatsächlich rein perzeptuelle Erfassung der relativen Wertkomponente nur bei kleinen Problemen erfolgen kann. Ein weiteres Entscheidungsmodell unter Berücksichtigung relativer Vorteile, das *Advantage Model*, findet sich bei Shafir, Osherson und Smith (1989, 1993).

4.2.3 Kognitive Determinanten des Entscheidungsverhaltens

4.2.3.1 Schwierigkeit des Entscheidungsproblems

Die Schwierigkeit eines Entscheidungsproblems ist ein abstraktes Konstrukt, das hier in Anlehnung an Bettman et al. (1998) durch dessen Größe sowie die Vollständigkeit verfügbarer Informationen konkretisiert wird. Die Autoren nennen als weitere Konkretisierungen Konflikte durch negativ korrelierte Attributausprägungen sowie Zeitdruck. Diese beiden Aspekte werden jedoch für diese Untersuchung als weniger relevant betrachtet: Dem Verfasser ist keine prinzipiell negative Attributkorrelation von Softwareattributen bewusst, die hier eine wichtige Rolle spielen könnte. Sicherlich wird es Programme geben, die z. B. optisch sehr ansprechend, dafür aber schlecht zu bedienen sind. Es ist jedoch

⁹ Dieses Prinzip besagt, dass die Rangfolge zwischen zwei Optionen unabhängig vom Vorhandensein weiterer Optionen besteht (s. Abschnitt 4.2.3.3).

nicht so, dass Optik und Bedienkomfort als Attribute immer und grundsätzlich im Gegensatz zueinander stehen. Zeitdruck wird nicht berücksichtigt, weil das Entscheidungsproblem auf den freiwilligen Entschluss eines Individuums zurückgeht. Insofern sollten Termine oder Ähnliches keine Rolle spielen, sondern so viel Zeit zur Verfügung stehen, wie das Individuum sich selbst dafür nehmen will.

Die *Größe* eines Entscheidungsproblems wird durch die Anzahl der zu evaluierenden Optionen sowie deren Attribute bestimmt. Bei einer großen Anzahl von Optionen werden vermehrt nicht-kompensatorische statt kompensatorischer Strategien angewendet, um Optionen zu eliminieren und damit das Problem zu verkleinern (E. J. Johnson & Meyer, 1984; Payne, 1976). Viele Optionen bedeuten aber nicht nur ein größeres Entscheidungsproblem, sondern auch mehr Auswahl, was vom Konsumenten in der Regel zunächst positiv bewertet wird (z. B. Broniarczyk, Hoyer & McAlister, 1998; Oppewal & Koelemeijer, 2005). Es zeigt sich jedoch, dass eine große Auswahl nicht notwendigerweise zu besseren Ergebnissen führt, dabei aber die Entscheidung erschwert (Iyengar & Lepper, 2000), u. U. so sehr, dass sie vermieden oder verschoben wird, wenn diese Möglichkeit besteht (Dhar, 1997). Verfügt ein Individuum jedoch über eine konkrete Vorstellung von der idealen Option, kann eine große Auswahlmenge die Entscheidung auch erleichtern, weil dadurch die Wahrscheinlichkeit steigt, dass das Ideal in ihr enthalten ist (Chernev, 2003).

Für das hier zu untersuchende Problem der Auswahl von Software aus einer offenen Menge ergibt sich die interessante Konstellation, dass das Individuum zunächst mit einer potenziell sehr großen Anzahl an Optionen konfrontiert ist, nämlich dann, wenn viele Programme für die relevante Anwendungsdomäne angeboten werden. Jedoch kann zu diesem Zeitpunkt noch keine detaillierte Evaluation erfolgen, da in Downloadportalen typischerweise keine umfangreichen Informationen angeboten werden. Davon abgesehen soll die Bewertung ohnehin auf der Basis praktischen Ausprobierens erfolgen, d. h. die Software muss erst heruntergeladen und installiert werden. Insofern ist das Individuum zu einer Vorauswahl unter limitierter Informationsverarbeitung quasi gezwungen und bestimmt auf diese Weise selbst, wie viele Optionen in die nächste Auswahlphase eintreten. Erst hier spielt dann die Anzahl der zu berücksichtigenden Attribute eine Rolle.

Die Anzahl der Attribute, mit denen die Optionen beschrieben werden, hat keinen unmittelbaren Einfluss auf die verwendete Strategie, kann aber die Selektivität der Informationsverarbeitung erhöhen (Olshavsky, 1979; Payne, 1976). Hinsichtlich des Einflusses der Problemgröße auf die Korrektheit von Entscheidungen ist die Befundlage uneinheitlich. Beobachtet wurden sowohl generell negative Wirkungen (Informationsüberflutung; Jacoby, Speller & Kohn, 1974a, 1974b; Keller & Staelin, 1987; Malhotra, 1982) als auch negative Wirkungen nur, wenn zusätzlich zur Anzahl der Attribute auch die der Optionen erhöht wurde (Russo, 1974; Summers, 1974; Wilkie, 1974). Eine Erklärung dafür könnte in einer fehlerbehafteten Messung der Entscheidungsqualität liegen (R. J. Meyer & Johnson, 1989). Dagegen sieht Lurie (2004) einen Fehler hinsichtlich der Feststellung der Problemgröße: Er argumentiert, dass diese weniger durch die reine *Anzahl* der Attribute bzw. Optionen, sondern vielmehr durch die *Struktur* der Informationen bestimmt wird, z. B. die Anzahl der Attributlevel, d. h. möglicher Ausprägungen von Attributen, und deren Verteilung über die Optionen.

Bei der Softwareauswahl ist die Ansicht plausibel, dass die Anzahl der zu verarbeitenden Attribute nicht nur von den *benötigten*, sondern ebenso von den jeweils *angebotenen* Funktionalitäten bzw. entsprechenden Dialogelementen abhängt. Dadurch ergibt sich der in der Literatur weitgehend unbeachtete Fall, dass die Optionen zwar in dem Sinne vergleichbar sind, dass sie über dieselbe relevante Attributmenge verfügen, d. h. sie haben eine Ausprägung hinsichtlich jeder benötigten Funktionalität. Diese relevanten Attribute sind jedoch erst aus einer *jeweils unterschiedlich strukturierten* Menge weiterer Attribute herauszufiltern. Das heißt, die konkrete Entscheidung, ob ein Attribut relevant ist oder nicht, kann nicht pauschal für alle Optionen erfolgen, weil die Attributmenge offen und für jede Option anders ist.

Unvollständigkeit der Informationen über die verfügbaren Optionen trägt nicht per se zur Schwierigkeit eines Problems bei. Je nach verwendeter Heuristik kann es sein, dass die fehlende Information ohnehin nicht benötigt wird. Anderenfalls muss der Entscheider, falls er die lückenhaft beschriebenen Optionen nicht pauschal eliminieren will, geeignete Annahmen treffen. Dies geschieht zum einen, indem er die fehlenden Attributwerte aus den entsprechenden vorhandenen Ausprägungen bei konkurrierenden Alternativen ableitet und mit einem Abschlag versieht, der der Unsicherheit des so ermittelten Wertes Rechnung tragen soll (R. J. Meyer, 1981; J. Huber & McCann, 1982; kritisch dazu: Simmons & Lynch, 1991). Zum anderen werden aber auch vorhandene Ausprägungen *anderer* Attribute *derselben* Option zum Füllen von Lücken verwendet (Ford & Smith, 1987; R. D. Johnson & Levin, 1985). Als Rahmen zur Erklärung, wann welche Quelle genutzt wird, kann die *Accessibility-Diagnosticity-Theorie* dienen (Feldman & Lynch, 1988). Wenn andere Attributwerte derselben Option zugänglich (*accessible*) sind, werden diese unter Berücksichtigung bekannter Korrelationen zwischen den Attributen benutzt (Dick, Chakravarti & Biehal, 1990). Bekannte Informationen über konkurrierende Optionen werden dagegen genutzt, wenn sie diagnostisch sind (Ross & Creyer, 1992). Beide Informationsquellen können jedoch durch intuitive Ansichten über wahrscheinliche Zusammenhänge zwischen Attributen dominiert werden (Broniarczyk & Alba, 1994).

Das Problem der Softwareauswahl durch direktes Ausprobieren fügt den bisher berücksichtigten einen neuen Aspekt hinzu: Fehlende Informationen fehlen hier nicht endgültig, sondern sind prinzipiell beschaffbar. Das Individuum muss „nur“ fleißig genug suchen und probieren. Es ist also nicht so, dass fehlende Informationen mehr oder weniger erraten werden müssen. Wovon es jedoch abhängt, ob geraten oder weiter nach sicheren Informationen gesucht wird, ist unklar. Ebenfalls offen bleibt, ob bzw. welchen Einfluss es auf die Entscheidung hat, wenn geraten wurde, obwohl sichere Informationen potenziell zur Verfügung standen.

4.2.3.2 Vorkenntnisse des Individuums

4.2.3.2.1 Speicherinteraktion

Im Zuge eines Entscheidungsprozesses werden nicht nur Informationen verarbeitet, die unmittelbar im Zusammenhang mit den Optionen angeboten werden. Von Bedeutung ist auch, ob und welche Informationen der Entscheider bereits vorher hatte, denn Gedächtnisinhalte und Entscheidungsprozess interagieren systematisch miteinander (Lynch & Srull,

1982). Generell erhöhen Vorkenntnisse in der relevanten Produktklasse die Effizienz der Suche nach Informationen und erleichtern deren Aufnahme (Brucks, 1985). Jedoch lernen Individuen mit höherem Vorwissen tendenziell weniger über neue Produkte als solche mit weniger Kenntnissen. Erstere könnten zwar mehr lernen, tun es aber nicht aufgrund motivationaler Defizite, welche sich vor allem in Unaufmerksamkeit beim Enkodieren neuer Informationen zeigen (Wood & Lynch, 2002).

Informationen über Optionen, die vor der Beschäftigung mit einer Auswahl Aufgabe unabhängig von dieser gelernt werden, werden alternativenbasiert gespeichert. Entsprechend erfolgt auch die Bearbeitung eines anschließenden Entscheidungsproblems eher alternativenbasiert. Werden Individuen ohne vorherige Lernphase mit dem Entscheidungsproblem konfrontiert, erfolgt dessen Bearbeitung und auch die Organisation der so akquirierten Gedächtnisinhalte eher attributbasiert (Biehal & Chakravarti, 1982; E. J. Johnson & Russo, 1984). Wird das Problem nach dem Treffen einer Entscheidung erweitert und erneut vorgelegt, besteht bei Personen mit vorheriger Lernphase eine größere Wahrscheinlichkeit, dass eine Neuorientierung unter Berücksichtigung der zusätzlichen Informationen erfolgt (Biehal & Chakravarti, 1983).

Von den im Gedächtnis gespeicherten, für eine Entscheidung relevanten Informationen werden bevorzugt jene benutzt, die sowohl verfügbar als auch in erhöhtem Maße diagnostisch sind (Feldman & Lynch, 1988; Lynch, Marmorstein & Weigold, 1988). Ist eine verfügbare Information wenig diagnostisch, wird jedoch kaum versucht, potenziell diagnostischere, aber weniger leicht zugreifbare Gedächtnisinhalte verfügbar zu machen. Insofern könnten Konsumenten als „kognitive Geizkragen“ (Lynch et al., 1988, S. 171) angesehen werden, die versuchen, Entscheidungen möglichst mit jenen Informationen zu treffen, die in der aktuellen Situation salient sind.

Daraus folgt, dass eine Auswahlentscheidung durch Manipulation der Verfügbarkeit einer gespeicherten Option beeinflusst werden kann, ohne die eigentliche Bewertung dieser Option zu verändern (Nedungadi, 1990). Die Leichtigkeit des Abrufs von Informationen nimmt genau dann Einfluss auf die Bewertung eines Stimulus, wenn die Leichtigkeit selbst als diagnostisch wahrgenommen wird. Ansonsten ist sie uninformativ und rückt zugunsten der abgerufenen Inhalte in den Hintergrund (Schwarz & Vaughn, 2000). Das ist dann der Fall, wenn Informationen nicht oder sehr leicht zugreifbar sind: In diesen Fällen wird das Erleben der Abrufleichtigkeit antizipiert und ist deshalb nicht diagnostisch. Bei moderater Verfügbarkeit dagegen ist das Abruferlebnis nicht vorhersagbar und hat somit Bedeutung für die Bewertung des Stimulus (Tybout, Sternthal, Malaviya, Bakamitsos & Park, 2005).

Die Salienz von Attributen im Verlauf einer Entscheidung kann zu persistenten Präferenzen für diese Attributmenge führen, nämlich dann, wenn sich das Individuum seiner Entscheidung sicher war und sich diese durch anschließende Erfahrungen mit der gewählten Option nicht als falsch herausgestellt hat (Muthukrishnan & Kardes, 2001). Das heißt, dass bei wiederholten Entscheidungen aus derselben Produktklasse bevorzugt jene Attribute herangezogen werden, die beim ersten Mal salient waren und deren Evaluation zu einem sicheren Ergebnis geführt hat. Diese werden also gegenüber anderen, und zwar auch zusätzlich verfügbaren Attributen mit objektiv größerer Relevanz, höher gewichtet.

Dies kann damit erklärt werden, dass der Entscheider seine Theorie über den Zusammenhang zwischen diesen Attributen und dem Nutzen der gewählten Option bestätigt sieht.

Die in diesem Abschnitt zusammengefassten Befunde sind vor allem interessant hinsichtlich der Frage, wie die beim Ausprobieren einer unbekanntem Software erlebten Ereignisse sowie die erzielten Ergebnisse bewertet werden. Demnach sind deutliche Unterschiede zwischen erfahrenen und weniger erfahrenen Individuen zu erwarten, da bei ersteren von einer erhöhten Verfügbarkeit und Diagnostizität relevanter Vorkenntnisse auszugehen ist. Diese Individuen sollten entsprechend effizienter vorgehen und zu besseren Ergebnissen kommen. Dies gilt allerdings zunächst nur unter der Voraussetzung, dass es allen Individuen in gleicher Weise gelingt, überhaupt zu bewertbaren Ereignissen und Ergebnissen zu kommen. Das heißt, dass auch in diesem Punkt der in dieser Untersuchung anscheinend neuartige Aspekt eine besondere Rolle spielt, dass entscheidungsrelevante Informationen nicht präsentiert werden, sondern zunächst erst einmal in unvorhersagbarer Weise zu erarbeiten sind.

4.2.3.2.2 Kategorien

Von unmittelbarem Interesse für die Bedeutung von vorhandenem Wissen ist die Kategorisierungsforschung. Kategorisierung bedeutet, dass Individuen die sie umgebenden Objekte in Kategorien organisieren, was eine effiziente Verarbeitung der Umwelt ermöglicht (Mervis & Rosch, 1981). Kategorien sind somit als spezielle Form von Schemata (vgl. Abschnitt 4.2.1) zu verstehen und werden durch konkrete Exemplare und abstrakte Prototypen mit typischen Attributausprägungen repräsentiert (Elio & Anderson, 1981). Kann nun ein Stimulus in eine existierende Kategorie eingeordnet werden, wird die Evaluation der Kategorie einfach auf den Stimulus übertragen (kategorienbasierte Verarbeitung). Existiert keine passende Kategorie, muss eine Evaluation für den Stimulus dagegen stückweise erarbeitet werden (Fiske, 1982; Fiske & Pavelchak, 1986). Eine den Stimulus begleitende Analogie, die eine Ähnlichkeit mit einer vertrauten Basiskategorie unterstreicht, verleiht der Evaluationsübertragung besonderen Nachdruck, und zwar in dem Maße, in dem der Rezipient über die Fähigkeit und die kognitiven Ressourcen verfügt, den Zusammenhang zwischen Stimulus und Kategorie herzustellen (Roehm & Sternthal, 2001).

Im Konsumentenkontext konnte Sujian (1985) zeigen, dass der Verarbeitungsmodus davon abhängt, ob Produktinformationen zu aus entsprechenden Kategorien abgeleiteten Erwartungen passen: Falls ja, erfolgt die Informationsverarbeitung eher kategorienbasiert, falls nein, dann eher stückweise. Je größer die Expertise eines Individuums, desto stärker wird dieser Zusammenhang. Das heißt zum einen, dass umfangreicheres Vorwissen es ermöglicht, zu einem Stimulus passende Kategorien effizient zur Ableitung einer Bewertung dieses Stimulus einzusetzen. Bei nicht passenden Stimuli scheinen die vorhandenen Kategorien die stückweise Verarbeitung zu leiten, wenn versucht wird, die einzelnen Informationen zu einer Bewertung zu integrieren. Auch Novizen erkennen, ob Stimuli zur Kategorienerwartung passen, verarbeiten aber stärker als Experten auch bei Abweichungen kategorienbasiert (ebd.). Auf eine bestimmte Kategorie verweisende generelle Merkmale eines Stimulus scheinen schwerer zu wiegen als konkrete Attributausprägungen, die gegen die Anwendung dieser Kategorie sprechen.

Dass Kategorien bzw. Schemata eine wichtige Rolle bei der Evaluation von Objekten spielen, ist unmittelbar plausibel: Das in ihnen in Form von typischen Attributausprägungen gespeicherte Wissen sollte als Anker für die Einordnung dieser Objekte dienen. Bei für den Konsumenten neuen Produkten kann moderate Kategorien- oder Schemainkongruenz, d. h. Abweichung der tatsächlichen Attributwerte von den aus dem passenden Schema abgeleiteten Erwartungen, dazu führen, dass das Produkt besser bewertet wird als kongruente oder stark inkongruente Optionen (Meyers-Levy & Tybout, 1989). Erklärt wird dies damit, dass moderat inkongruente Stimuli Erregung und das Bestreben auslösen, die Inkongruenz aufzulösen. Da diese nur moderat ist, wird dies gelingen. Es stellt sich eine deutlich positive Reaktion ob der gelungenen Einordnung des Stimulus und der eigenen kognitiven Leistung ein. Ein von vorn herein kongruenter Stimulus löst nur vergleichsweise schwache positive Resonanz durch eine gewisse Vertrautheit aus und schneidet schlechter ab. Starke Inkongruenz dagegen kann nicht ohne tiefe Eingriffe in vorhandene Schemata bewältigt werden und führt zu negativen Reaktionen durch Unsicherheit und Frustration (Mandler, 1982).

Moderiert wird dieser Effekt durch das Ausmaß an Vorkenntnissen, d. h. den Umfang und die Spezifität vorhandener Schemata, und das im Zusammenhang mit der Entscheidung wahrgenommene Risiko. Besitzen Konsumenten elaboriertes Wissen in der relevanten Kategorie, nimmt der oben beschriebene Einfluss von Schemakongruenz auf die Produktbewertung ab. Von größerer Bedeutung ist dann die Valenz schemaabhängiger Assoziationen hinsichtlich spezifischer Attribute (Peracchio & Tybout, 1996). Voraussetzung für das Auftreten einer Präferenz moderat schema-inkongruenter Optionen ist, dass die Entscheidung als risikofrei wahrgenommen wird. Das heißt, dass bei riskanten Entscheidungen schema-kongruente Produkte präferiert werden (Campbell & Goodstein, 2001).

Sogenannte „wirklich neue Produkte“ (Lehmann, 1994), die in keine vorhandene Kategorie passen, vereinen oft Merkmale, die aus verschiedenen Kategorien abgeleitet werden. Die Konstruktion entsprechender neuer Wissensstrukturen scheint dann unter Nutzung bzw. Transfer von Bekanntem zu erfolgen (Gregan-Paxton & Roedder John, 1997; Yamauchi & Markman, 2000). Dabei nimmt die erste plausible Kategorie, die dem Individuum als Informationsquelle dargeboten wird, einen dominanten Einfluss auf die Kategorisierungen, Erwartungen und Präferenzen hinsichtlich des neuen Produktes (Moreau, Markman & Lehmann, 2001). Die Wirkrichtung neuartiger Attribute auf die Evaluation eines Produktes hängt von dessen Komplexität ab: Ist diese gering, wird ein neuartiges Attribut in der Regel als zusätzlicher Nutzen wahrgenommen und wirkt positiv auf die Einschätzung. Bei hoher Produktkomplexität kann dagegen eine negative Wirkung aufgrund der Erwartung hohen Lernaufwands eintreten (Mukherjee & Hoyer, 2001).

Einige der in diesem Abschnitt referierten Befunde aus der Kategorisierungsforschung sind durch ihre Nähe zur Schematheorie potenziell fruchtbar für das hier zu untersuchende Problem, da durch Schemata auch gesteuert werden sollte, wie ein Individuum versucht, eine unbekannte Software zu bedienen. Insbesondere die Arbeit von Meyers-Levy und Tybout (1989) lässt z. B. den Schluss zu, dass ein nicht den Erwartungen gemäß reagierendes Programm ungeachtet seiner tatsächlichen Qualitäten negativ bewertet wird.

Gemäß Peracchio und Tybout (1996) sollten dann aber vor allem erfahrene Individuen in der Lage sein, diese Qualitäten dennoch zu erkennen und zu würdigen.

Die Studien hinsichtlich „wirklich neuer Produkte“ sollten vor allem für das Verhalten wenig erfahrener Individuen von Belang sein. Demnach müsste das Erleben der als Erstes evaluierten Option Einfluss auf die Bewertung nachfolgender Programme nehmen. Unklar ist jedoch, inwieweit bestimmte Attribute tatsächlich als neuartig wirken können. Es ist sowohl denkbar, dass ein eigentlich nicht neuartiges Attribut einem unerfahrenen Individuum aber neuartig erscheint, als auch dass ein tatsächlich neuartiges Attribut aufgrund der individuellen Unerfahrenheit neben den vielen anderen Eindrücken untergeht. Das Individuum ist durch das selbstständige Ausprobieren darauf angewiesen, selbst einzuschätzen, welchen Nutzen es aus den einzelnen Attributen ziehen kann.

4.2.3.3 Kontext-Effekte

Nach der Theorie von Entscheidungen nach dem Prinzip der Nutzenmaximierung assoziiert der Entscheider mit jeder Option einen bestimmten Nutzen und wählt die Option mit dem höchsten Wert (R. D. Luce, 1959). Die einzelnen Nutzenwerte der verfügbaren Optionen und die sich daraus ergebende Präferenzordnung sind dabei unabhängig von der An- oder Abwesenheit weiterer Optionen. Dieses Prinzip heißt *Unabhängigkeit (der Präferenz) von irrelevanten Alternativen* (Tversky, 1972) und bedeutet, dass eine bestehende Präferenz von A über B unbeeinflusst davon bestehen muss, ob eine weitere (insofern irrelevante) Option C verfügbar ist. Dieses Prinzip erscheint zwar höchst einleuchtend, ist jedoch nicht immer zutreffend. Vielmehr sind systematische Verstöße gegen seine allgemeine Gültigkeit beobachtbar, welche nahe legen, dass die Präferenzordnung durch den Kontext der in der Auswahlmenge enthaltenen Optionen beeinflusst wird (z. B. J. Huber, Payne & Puto, 1982; Simonson, 1989). Es geht bei Kontext-Effekten also um den Einfluss, den die zwischen den Optionen bestehenden Beziehungen auf die Entscheidung entfalten.

Ein grundlegendes Konzept dabei ist der *Kontrast* zwischen einer fokalen Option und dem durch aktuelle Alternativen bzw. frühere Erfahrungen bestimmten Kontext, in dem die Evaluation erfolgt (Lynch, Chakravarti & Mitra, 1991). Aus solchen Kontrasten resultierende Effekte sind auf dem Gebiet der Wahrnehmungspsychologie bekannte Phänomene (vgl. z. B. Goldstein, 2001; Guski, 1991). Ein eindrucksvolles Beispiel dafür ist die in Abbildung 4.2 dargestellte optische Täuschung, die auf einem Effekt basiert, der als *Simultankontrast* bezeichnet wird: Beide inneren Kreise haben exakt dieselbe graue Farbe, erscheinen aber je nach Hintergrund unterschiedlich hell. Auf ähnliche Weise kann im Zusammenhang mit einem Entscheidungsproblem ein und dieselbe Option vor dem Hintergrund schlechterer Alternativen attraktiv, vor dem Hintergrund besserer Alternativen dagegen unattraktiv erscheinen (Tversky & Simonson, 1993).

Eine unmittelbare Manifestation dieses Prinzips ist der *Range-Effekt*. Er kommt nach der Range-Theorie (Parducci, 1968, 1974) dadurch zustande, dass Individuen bei der Evaluation eines Stimulus die beiden Extremwerte identifizieren, die dieser Stimulus annehmen kann. Dadurch legen sie den Kontext fest, in dem die Evaluation erfolgt. Dieser (Werte-)Bereich (*Range*) ergibt sich aus früheren Erfahrungen oder den Attributausprä-

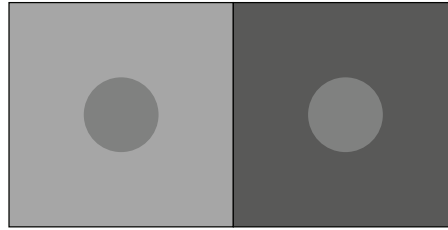


Abbildung 4.2: Simultankontrast (nach Goldstein, 2001, S. 68)

gungen der aktuellen Auswahlmenge. Die Bewertung eines Stimulus ist somit eine Funktion von dessen Position innerhalb dieses Bereiches. Ein Attributwert 9 sollte z. B. in einem Kontext, der Werte von 1 bis 10 enthält, ziemlich gut da stehen. Enthält der Kontext jedoch Optionen, die es auf 100 bringen, wirkt ein Wert von 9 dagegen recht schwach. Der Range-Effekt beschreibt die daraus folgende Tendenz, den Unterschied zwischen zwei Werten um so geringer wahrzunehmen, je breiter der Bereich ist, in dessen Kontext die Evaluation erfolgt (Yeung & Soman, 2005).

Simonson und Tversky (1992) postulieren, dass Kontrasteffekte auch hinsichtlich der Trade-offs zwischen Attributen auftreten und erklären damit den *Attraktions-Effekt* (Chernev, 2005; J. Huber et al., 1982; J. Huber & Puto, 1983; vgl. Abb. 4.3): Angenommen, ein Individuum habe die Wahl zwischen zwei Programmen A und B, wobei A zwar die besseren Ergebnisse liefert, dafür aber deutlich hässlicher ist. Es ist nun zu entscheiden, ob die Hässlichkeit der Oberfläche für die guten Ergebnisse in Kauf genommen werden soll. Das Anbieten eines dritten Programms C sollte bei rationaler Entscheidungsfindung nichts an der relativen Stärke der Präferenz für A oder B ändern. Der Attraktions-Effekt beschreibt die Tendenz, sich verstärkt für A zu entscheiden, wenn C durch A, aber nicht durch B dominiert wird (asymmetrische Dominanz)¹⁰. Dies ist z. B. der Fall, wenn C schlechtere Ergebnisse liefert als A, ohne dabei aber besser auszusehen. Dadurch, dass A eindeutig besser ist als C, erscheint A attraktiver als B, das weder eindeutig besser als A noch C ist. Mit anderen Worten: Die Austauschrate von Ästhetik zu Ergebnisqualität von A zu C ist eindeutig schlechter als die von A zu B. Deshalb erscheint der durch A und B bestimmte Trade-off bei Anwesenheit von C akzeptabler als ohne.

Ein weiteres grundlegendes Konzept, die *Abneigung gegen Extreme*, basiert auf dem Prinzip der *Verlustaversion* (Tversky & Kahneman, 1991), welches besagt, dass Individuen Verluste stärker wahrnehmen als korrespondierende Gewinne. Im Zusammenhang mit einem Entscheidungsproblem sind Gewinne und Verluste als Vor- bzw. Nachteile einer Option gegenüber einer anderen zu verstehen (Simonson & Tversky, 1992). In obigem Beispiel hätte Programm A gegenüber B den Vorteil besserer Ergebnisse und den Nachteil einer weniger ansprechenden Optik. Bei B ist die Situation genau umgekehrt. Das Prinzip der Abneigung gegen Extreme beschreibt nun die Tendenz, eine Option mit moderaten

¹⁰ Dominanz bedeutet, dass eine Alternative A auf mindestens einem Attribut besser als, und auf allen anderen ebenso gut ist, wie eine Alternative C. Asymmetrische Dominanz liegt vor, wenn eine Option C zwar von A, aber nicht von B dominiert wird (Jungermann et al., 2005).

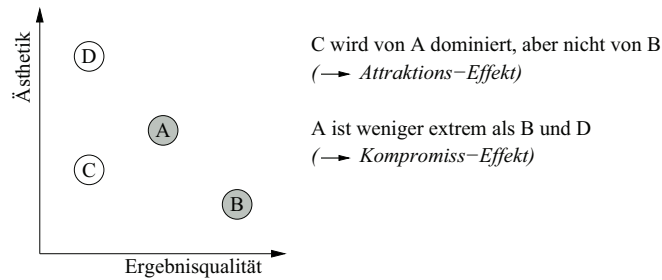


Abbildung 4.3: Dominanzverhältnisse und Attraktions- bzw. Kompromiss-Effekt (nach Chernev, 2005, S. 214)

Attributwerten attraktiver wahrzunehmen als eine ansonsten rational gleichwertige Option mit extremen Attributausprägungen (Tversky & Simonson, 1993).

Im Beispiel (Abb. 4.3) bedarf es dazu mindestens einer weiteren Option, da bei binärer Auswahl keine Option extremer ist als die andere. Wird also ein weiteres, rational ebenfalls gleichwertiges Programm D eingeführt, das noch hässlicher, aber auch noch besser ist als A, liegt A mit seinen Attributwerten zwischen B und D. Die mit Option A assoziierten Vor- und Nachteile gegenüber B und D sind somit kleiner (weniger extrem) als die von B und D untereinander. In dieser Situation manifestiert sich ein *Kompromiss-Effekt* als die Tendenz, die mittlere Option A zu bevorzugen (Simonson, 1989).

4.3 Entscheidungsverhalten und Emotion

4.3.1 Terminologische Abgrenzung

Angesichts der wenig reflektierten Verwendung des Begriffs Emotion als Synonym für Gefühl in der Umgangssprache erscheint zunächst eine Abgrenzung der relevanten Terminologie angebracht. *Emotion* ist ein hypothetisches Konstrukt, ein „qualitativ näher beschreibbarer Zustand, der mit Veränderungen auf einer oder mehreren der folgenden Ebenen einhergeht“ (Schmidt-Atzert, 1996, S. 21) und insofern nur über beobachtbare Phänomene auf diesen Ebenen manifest wird: (1) *Gefühl*, welches das subjektive Erleben beschreibt; (2) *körperlicher Zustand*, also physiologische Merkmale wie z. B. Blutdruck; (3) *Ausdruck*, d. h. Mimik oder sonstige das Erleben reflektierende Verhaltensweisen (ebd.; Izard, 1999). Gefühl ist demnach nur ein Aspekt von Emotion und dieser nicht gleichzusetzen.

Ein weiterer, in diesem Zusammenhang relevanter Begriff, nämlich *Stimmung* ist dagegen bereits intuitiv etwas anderes als Emotion und kann von dieser dadurch unterschieden werden, dass sie schwächer aber stabiler und länger andauernd ist. Darüber hinaus fehlt ein klarer Bezug zu einem Auslöser, d. h. im Gegensatz zu einer Emotion kann für eine Stimmung keine eindeutige Ursache oder ein Objekt benannt werden (Schmidt-Atzert, 1996). Eine Stimmung kann somit als der Hintergrund verstanden werden, vor dem sich das Erleben abspielt und der dessen Wahrnehmung als Dauertönung beeinflusst.

In der relevanten englischsprachigen Literatur unterbleiben derartige terminologische Abgrenzungen in der Regel. Vielmehr werden die Begriffe *affect*, *emotion* und, allerdings seltener, *feeling* zum Teil sogar explizit synonym verwendet (z. B. Isen, 1997). Was im konkreten Fall im Hinblick auf die oben dargelegte Terminologie jeweils gemeint ist, muss aus dem Kontext erschlossen werden. Eine gewisse Differenzierung erfährt der Begriff *mood*, der nicht verwendet wird, wenn eigentlich eine Emotion oder ein Gefühl gemeint ist. Jedoch wird auf Stimmungen häufig auch mit *affect* und *emotion* Bezug genommen (z. B. Lewinson & Mano, 1993). Keinesfalls gleichzusetzen sind das im Englischen universell für alle Formen des Empfindens verwendete *affect* und der deutsche Begriff *Affekt*. Letzterer bezeichnet sehr spezifisch eine kurze, vor allem aber sehr heftige und potenziell zu unkontrolliertem Verhalten führende Emotion (Schmidt-Atzert, 1996) und wird vor allem in juristischen Zusammenhängen verwendet.

Emotion und Stimmung spielen in der Literatur zum Bewerten von Optionen und Treffen einer Entscheidung auf ganz unterschiedliche Weise eine Rolle: Zum einen kann die Stimmung eines Subjekts als problemunabhängiger Faktor, der nicht auf ein bestimmtes Objekt gerichtet ist und dessen Ursprung nicht mit dem Entscheidungsproblem zusammenhängt, Einfluss auf die Entscheidung nehmen (z. B. Isen, 1997; Lewinson & Mano, 1993). Zum zweiten kann eine emotionale Reaktion, die durch eine Option aus der Auswahlmenge hervorgerufen wird bzw. auf diese gerichtet ist, als diagnostische Information in die Evaluation und Entscheidungsfindung einfließen (z. B. Pham, 1998). Drittens kann das Entscheidungsproblem als solches durch seine Schwierigkeit negative Emotionen bewirken, die über das Streben des Individuums, sie zu vermeiden, prozesssteuernd wirken können (z. B. M. F. Luce, 1998).

4.3.2 Einfluss von Stimmung als Hintergrundvariable

4.3.2.1 Stimmung als eindimensionales Konstrukt

Die Stimmung eines Menschen nimmt auf komplexe Weise Einfluss auf die Bearbeitung von Problemen und somit auch auf Entscheidungsprozesse. Ein in diesem Zusammenhang grundlegendes Konzept ist das der Stimmungskongruenz, welches besagt, dass zur Stimmung passende Inhalte leichter erlernt bzw. besser erinnert werden als unpassende (Bower, 1981). Ein Individuum wird also aktuell präsentierte Informationen, die im Zusammenhang mit stimmungsaktivierten Kognitionen stehen, extensiver und selektiver verarbeiten. Eine Folge davon ist, dass Evaluationen von der aktuellen Stimmung beeinflusst werden, d. h. in guter Stimmung abgegebene Urteile besser ausfallen als in schlechter (Bower, 1991; Forgas & Bower, 1988).

Dieser Effekt scheint bei positiver Stimmung deutlicher zu sein als bei negativer (Leventhal & Tomarken, 1986). Das bedeutet, dass insbesondere affektiv positive Reize den Zugriff auf entsprechend positive Gedächtnisinhalte erleichtern bzw. verstärkt positive Kognitionen auslösen (Teasdale & Fogarty, 1979). Wird weiterhin berücksichtigt, dass positive Gedächtnisinhalte extensiver und facettenreicher gespeichert werden als negative (Cramer, 1968), führt dies zu dem Schluss, dass die Bearbeitung desselben Problems bei guter Stimmung in einem komplexeren kognitiven Kontext erfolgt als bei negativer Stim-

mung (Isen, Johnson, Mertz & Robinson, 1985). Dies kann zum einen zu besseren und effizienteren Lösungen führen und zum anderen weitere Effekte aufgrund des erweiterten Kontexts nach sich ziehen (Isen, 1997; vgl. Abschnitt 4.2.3.3 zu Kontexteffekten).

Eine Voraussetzung für das Wirken von Kongruenzeffekten scheint das Fehlen starker Schemata bzw. Kategorien für die Evaluation von Stimulusmaterial zu sein (Fiedler, 1985, S. 127): „Die Stimmungsabhängigkeit des Gedächtnisses verschwindet in dem Maße, wie der Erinnerungsprozeß durch eine systematische Wissensstruktur gesteuert wird.“ Das heißt, wenn ein Stimulus in eine vorhandene Kategorie passt, scheint deren fest verankerte Evaluation unbeeinflusst von aktuellen emotionalen Befindlichkeiten auf diesen übertragen zu werden.

Als Alternative zur speicherbasierten Erklärung der Wirkung von Stimmungskongruenz auf Beurteilungen postuliert Adaval (2001) einen *Affect-Confirmation-Prozess*. Dieses Konzept basiert nicht auf der Verfügbarkeit bereits vorhandener Kognitionen, sondern stellt die affektive Valenz von Informationen, die zum aktuellen Zeitpunkt aufgenommen werden, in den Vordergrund. Demnach nehmen bei der Evaluation von Stimuli stimmungskongruente Informationen stärkeren Einfluss auf das Gesamturteil als inkongruente. Das heißt, gut gestimmte Individuen schenken den positiven, schlecht gestimmte hingegen den negativen Aspekten eines Objektes mehr Beachtung. So könnte z. B. die Reputation eines Softwareherstellers eine Information mit positiven Implikationen sein, die etwas schrille Gestaltung eines konkreten Programms eine mit negativen. Je nach Stimmung sollte nun die Reputation bzw. die Optik schwerer wiegen und ein Gesamturteil über die Software *ceteris paribus* entsprechend positiver bzw. negativer ausfallen.

Adavals (2001) Untersuchung am Beispiel von Jeanshosen, Sweatshirts, Laufschuhen und Rucksäcken liefert Unterstützung für diesen Mechanismus und identifiziert zudem eine Nebenbedingung: Stimmungskongruente Beurteilungen treten typischerweise dann auf, wenn hedonistische Kriterien schwerer wiegen als utilitaristische. Dabei sind unter utilitaristischen Kriterien solche mit Bezug zu einer nützlichen Funktionalität zu verstehen, z. B. die Dämpfungseigenschaften eines Sportschuhs. Hedonistische Kriterien fokussieren dagegen auf die emotionalen Erlebnisse, die beim Konsum bzw. der Benutzung des Produktes erwartet werden, z. B. das Markenimage des Schuhs (vgl. auch Mano & Oliver, 1993; Batra & Ahtola, 1990). Hinsichtlich des seinem Modell zugrunde liegenden Mechanismus' stellt Adaval (2003) fest, dass Stimmungskongruenz nicht die *Gewichtung* des betreffenden Attributs erhöht, sondern dessen *Valenz* verstärkt. Das heißt, dass stimmungskongruente Informationen nicht stärker gewichtet, sondern höher bewertet werden.

4.3.2.2 Stimmung als mehrdimensionales Konstrukt

In den bisher diskutierten Arbeiten wurde der emotionale Zustand eines Individuums stets als eindimensionales Konstrukt mit den Ausprägungen „gut“, „neutral“ und „schlecht“ unterstellt. Jedoch finden sich in der Literatur auch Befunde, die von einer Mehrdimensionalität ausgehen und dadurch ein differenzierteres Verständnis der Zusammenhänge zwischen emotionalen Konstrukten und Entscheidungsprozessen ermöglichen sollen (Levinson & Mano, 1993; Mano, 1990). Als primäre Dimensionen von Emotion werden dabei, basierend auf dem im Folgenden erläuterten *Circumplex-Modell der Emotionen*

(Russell, 1980; Russell & Pratt, 1980; vgl. auch Watson & Tellegen, 1985), *Lust – Unlust* und *Erregung – Ruhe* verwendet.

Russell (1980) ließ Versuchspersonen 28 Adjektive zur Beschreibung emotionaler Zustände in 8 Kategorien einordnen. Diese Kategorien sollten anschließend in eine zirkuläre Ordnung gebracht werden, sodass einander gegenüberliegende Wörter gegensätzliche, bzw. nahe beieinander liegende Wörter ähnliche Gefühle beschreiben. In einer zweiten Studie sollten Versuchspersonen dieselben Adjektive nach Ähnlichkeit in 4, 7, 10 bzw. 13 Gruppen einsortieren. Als Maß der wahrgenommenen Ähnlichkeit zwischen jeweils zwei Wörtern diente die Häufigkeit der Einordnung in dieselbe Gruppe, gewichtet nach der Anzahl der jeweils verfügbaren Optionen. Die Ergebnisse dieser Studien unterstützten die Hypothese, dass eine Emotion bzw. Stimmung durch ihre Position in einem durch zwei Dimensionen aufgespannten Raum beschrieben werden kann. Die idealisierte Form des Circumplex-Modells der Emotionen ist in Abbildung 4.4 dargestellt.

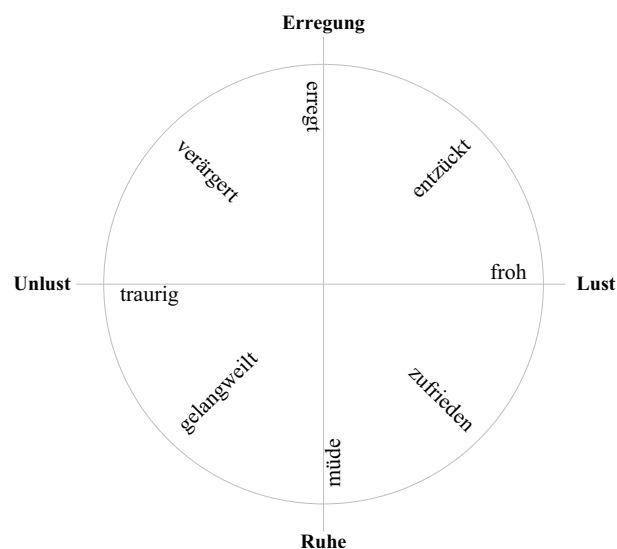


Abbildung 4.4: Circumplex-Modell der Emotion (nach Russell, 1980, S. 1164; deutsche Beschriftung nach Schmidt-Atzert, 1996, S. 93)

Es erscheint intuitiv einleuchtend, dass es sich unterschiedlich auf die Bearbeitung eines Entscheidungsproblems auswirkt, ob sich ein Individuum z. B. positiv/ruhig oder positiv/erregt fühlt, ob es verärgert oder gelangweilt ist. Ausgehend von diesem Ansatz erhoben Lewinson und Mano (1993) die aktuelle emotionale Befindlichkeit ihrer Versuchspersonen und ließen sie anschließend eine Auswahl aus sechs hypothetischen Reisetaschen, beschrieben durch jeweils fünf Attribute treffen. In einem zweiten Experiment, in dem die Stimmung manipuliert wurde, konnten die Ergebnisse der ersten Studie am Beispiel von Managemententscheidungen (Auswahl von Mitarbeitern für Beförderung) repliziert werden. Demnach gehen Individuen, deren Stimmung auf der Lust-Unlust-Dimension näher an Lust verortet ist, gründlicher vor in dem Sinne, dass sie länger überlegen und die angebotenen Informationen in größerem Umfang nutzen. Zudem ist eine eher alternativenbasierte Verarbeitung zu beobachten.

Dieses Ergebnis ist konsistent zu Befunden von Vertretern des eindimensionalen Verständnisses von Emotion (z. B. Bower, 1981; Isen, Daubman & Nowicki, 1987), wonach eine Strategie passend zur emotionalen Befindlichkeit ausgewählt wird: Individuen in guter Stimmung nehmen das Entscheidungsproblem als unterhaltsame Aktivität wahr und verwenden entsprechend elaboriertere Strategien. Individuen in schlechter Stimmung dagegen empfinden die Aufgabe als Belastung und versuchen sich dieser so schnell wie möglich zu entledigen. Hinsichtlich der Erregung-Ruhe-Dimension stellen Lewinson und Mano (1993) fest, dass erregtere Individuen dazu neigen, den Verarbeitungsaufwand zu reduzieren, indem weniger Informationen genutzt und unkritische Dimensionen ignoriert werden, weil aufgrund der höheren Erregung weniger Aufmerksamkeitsressourcen zur Verfügung stehen. Ruhigere Individuen, die sowohl kritische als auch unkritische Produktattribute berücksichtigen, kommen entsprechend zu besseren Ergebnissen.

4.3.3 Emotion als Information

In den bisher dargestellten Befunden, die das Gros der Literatur zum Thema repräsentieren, war die Rolle von Emotionen auf die einer Hintergrundvariable im Sinne der problemunabhängigen aktuellen Stimmung von Individuen beschränkt, die Einfluss auf die Art und Weise der Informationsverarbeitung zur Entscheidungsfindung nimmt. In ihrer mittlerweile klassischen sozialpsychologischen Untersuchung gingen Schwarz und Clore (1983) der Frage nach, in welchem Maße Stimmungskongruenz stets und automatisch oder aber nur dann auftritt, wenn sich das Individuum der Ursache seiner Stimmung nicht bewusst ist. Clore (1992) illustriert dieses Problem sehr anschaulich, indem er fragt, warum nicht jeder in die Person verliebt ist, die einem den Gehaltsscheck überreicht: Schließlich sollte man dabei in sehr guter Stimmung sein, was ein entsprechend hohes Urteil hinsichtlich der Attraktivität dieser Person nach sich zieht.

In zwei Experimenten zeigen Schwarz und Clore (1983), dass Stimmungskongruenz tatsächlich nur dann auftritt, wenn ein Individuum nicht auf einen möglichen problemunabhängigen Grund für seine Stimmung hingewiesen wird. Befragt nach ihrer Lebenszufriedenheit gaben die Versuchspersonen nur dann stimmungskongruente Urteile ab, wenn sie nicht auf eine mögliche externe Ursache (angebliche Effekte der besonderen Gestaltung des Labors bzw. das Wetter) für ihre Stimmung gestoßen wurden. Clore und Parrott (1991, S. 109) erklären den zugrunde liegenden Mechanismus wie folgt:

The basic premise [...] is that just as emotional expressions convey emotional information to other people, emotional experience conveys such information to oneself. Emotional experience serves as feedback from the appraisal system, signifying or confirming that a situation is personally significant. [...]

[T]he output of affective processes includes distinctive experiences that indicate how important an event is and in what way it is important. Such emotional experiences serve as information for judgment and decision making processes.

Die Tendenz, einen Zusammenhang zwischen dem emotionalen Erleben und dem aktuellen Gegenstand der Beschäftigung zu unterstellen, wird das *Unmittelbarkeitsprinzip* genannt (Clore et al., 2001). Dieses Prinzip gilt jedoch nur, wenn die Relevanz eines emotionalen Zustandes für das Zielobjekt nicht durch besondere Umstände infrage gestellt wird: „In short, what comes to mind seems relevant by default. In contrast, assessments that highlight the irrelevance and low diagnosticity of the input have to be triggered by salient features of the situation“ (Schwarz & Clore, 2003, S. 299).

Die Verwendung affektiver Informationen bei Bewertungen verursacht immer dann Fehltritte, wenn Stimmungen als Reaktion auf den Urteilsgegenstand fehlattribuiert werden, was bei guter (schlechter) Stimmung zu positiveren (negativeren) Bewertungen, also Stimmungskongruenz führt (Schwarz & Clore, 1988). Ein Hinweis auf eine konkurrierende Ursache kann eine solche Fehlattription korrigieren, was in der Summe darauf hinweist, dass das aktuelle affektive Erleben grundsätzlich in Bezug zur aktuellen Fragestellung gesetzt und somit als diagnostische Information zur Bewertung von Stimuli herangezogen wird. Der Prozess der Urteilsbildung scheint also einzuschließen, dass sich Individuen (implizit) fragen: „Was für ein Gefühl habe ich dabei?“ Schwarz und Clore (1988) sprechen in diesem Zusammenhang von einer *How-do-I-feel-about-it?*-Heuristik, quasi der Erhebung des Bauchgefühls in den Rang entscheidungsrelevanter Information.

Dieses Konzept ist bei der hier zu untersuchenden Softwareevaluation durch Ausprobieren in besonderem Maße von Interesse, weil die zur Entscheidung führenden Informationen nicht zweifelsfrei verbal präsentiert oder erhoben werden, sondern relevante wie irrelevante Attributausprägungen durch die Interaktion mit der Software unmittelbar erlebt oder erfahren werden. Insofern erscheint es plausibel, dass sich ein Individuum stets auch (implizit) fragt: „*Fühlt* sich das, was ich hier mache, richtig an? *Empfinde* ich diesen Mechanismus als einfach oder umständlich?“, oder feststellt: „So etwas ist *nervig!* Das *ärgert* mich schon hier ...!“

Affektive Reaktionen erfolgen zudem nicht nur auf die ermittelte Attributausprägung, sondern haben ihre Ursache auch in dem kognitiven Aufwand, der dafür betrieben werden musste. Garbarino und Edell (1997) zeigen, dass eine Option, deren Evaluation erhöhten kognitiven Aufwand erfordert, negative Emotionen hervorruft und weniger häufig gewählt wird als eine objektiv gleichwertige, aber leichter zu evaluierende Option. Dieser Zusammenhang könnte sich z. B. dann zeigen, wenn das Dialogmodell einer Software nicht zu den vorhandenen Schemata des Entscheiders passt, sodass das Erschließen der Funktionsweise nur mühsam gelingt. Die Folge davon sind negative Emotionen (z. B. Verwirrung, Unsicherheit, Ärger), die auf die Software attribuiert werden („Das ist alles total blöd gemacht hier!“), was der Bildung einer Präferenz für dieses Produkt nicht förderlich ist.

Die Rolle von Emotion als Information in Entscheidungsprozessen wurde z. B. von Pham (1998) untersucht. Im Gegensatz zu z. B. Schwarz und Clore (1983) geht es hier nicht um die Erklärung von Stimmungskongruenz durch Fehlattription, sondern um die Wirkung von affektiven Reaktionen, deren Repräsentativität für ein Zielobjekt nicht infrage gestellt wird. Es wird also davon ausgegangen, dass das emotionale Erleben in der Wahrnehmung des betroffenen Individuums mit dem Zielobjekt im Zusammenhang steht und insofern den Charakter einer Emotion hat (vgl. Abschnitt 4.3.1). Untersucht wird die Frage, unter welchen Umständen eine Berücksichtigung solcher Emotionen erfolgt, sie

also nicht nur als *repräsentativ*, sondern auch als *relevant* (bzw. diagnostisch) angesehen werden. Ein interessanter Aspekt dieser Studie ist, dass die Entscheidung, die die Versuchspersonen zu treffen hatten (Kinobesuch: ja oder nein), nicht wie sonst üblich als hypothetisch, sondern als real dargestellt wurde.

Pham (1998) kommt zu dem Ergebnis, dass affektive Informationen immer dann diagnostischen Charakter haben, wenn bei der Entscheidung konsumatorische bzw. hedonistische (vs. instrumentelle bzw. utilitaristische) Motive dominieren (vgl. auch Adaval, 2001). Das heißt, wenn bei den Instruktionen der Kinobesuch als Erlebnis im Vordergrund stand, nahmen aktuelle Emotionen Einfluss auf die Entscheidung. Wurde der Kinobesuch dagegen als inhaltliche Vorbereitung auf eine vergütete Folgestudie dargestellt, spielten Emotionen eine deutlich geringere Rolle. Darüber hinaus fallen Entscheidungen unter Berücksichtigung von Emotionen potenziell schneller und sind dabei interindividuell stabiler und konsistenter als solche auf der Basis emotionsloser Bewertungen und Vergleiche (Pham, Cohen, Pracejus & Hughes, 2001).

Nach einer Studie von Yeung und Wyer (2004) scheinen die oben beschriebenen Effekte und Kontingenzen jedoch nur dann aufzutreten, wenn ein Individuum vor der Verarbeitung der Attributinformationen, welche dann ggf. repräsentative und relevante emotionale Reaktionen nach sich zieht, keinen visuellen Kontakt mit dem Zielobjekt und somit keine Möglichkeit hatte, unabhängig von konkreten Informationen eine erste Abschätzung vorzunehmen. Erfolgt dagegen eine solche Abschätzung, kann diese ihrerseits positive oder negative affektive Reaktionen hervorrufen. Wenn das passiert, nimmt diese emotionale Erfahrung Einfluss auf den ersten Eindruck und die folgenden Beurteilungen, und zwar unabhängig davon, ob das Objekt nach hedonistischen oder instrumentellen Kriterien beurteilt wird. Bleiben auf Abschätzungen beruhende affektive Reaktionen aus, nehmen anderweitig verursachte Emotionen keinen Einfluss auf die Bewertung des Objektes. Mit den Worten der Autoren: „In short, under conditions in which consumers have a chance to appraise a product at the outset, the impact of affect on their judgments depends on whether this appraisal evokes affective reactions rather than the judgmental criteria they might otherwise have applied“ (Yeung & Wyer, 2004, S. 412).

Diese Einschränkung hat wichtige Konsequenzen für die praktische Anwendbarkeit sonstiger Befunde, da die Versuchspersonen im Labor in der Regel eben keinen Kontakt zu den Entscheidungsobjekten haben, sondern ihre Urteile auf der Basis ausgewählter, verbal präsentierter Attributwerte treffen müssen. Dies steht im Gegensatz zur wirklichen Welt, wo die fraglichen Objekte typischerweise zuerst *gesehen*, und erst dann weitere Informationen (mit potenziellen emotionalen Konsequenzen) erfragt werden. Andererseits ist zu fragen, ob die erste Abschätzung nicht auch einfach ein Informationslieferant ist, nämlich über die visuelle Erscheinung eines Objektes. Falls das Aussehen eines Produktes ein relevantes Kriterium ist oder andere relevante Aspekte daraus abgeleitet werden können, kann die Ausprägung dieses Attributes eine affektive Reaktion hervorrufen, die ihrerseits als Information verwertet wird.

4.3.4 Emotion als Ergebnis des Entscheidungsproblems

Bisher ging es um affektive Zustände, die entweder qua Stimmung von außen in das Entscheidungsproblem eingebracht werden oder als spontane Reaktion auf eine Option deren Evaluation beeinflussen. Beim Gedanken an die sprichwörtliche Qual der Wahl ist jedoch ohne weiteres plausibel, dass unter bestimmten Umständen auch das Entscheidungsproblem als solches, also allein der Umstand, sich für eine von mehreren Optionen entscheiden zu sollen, Unbehagen oder sonstige, in der Regel negative Emotionen hervorrufen kann. Ebenso plausibel ist, dass der Prozess und das Ergebnis der Beschäftigung mit einem solchen Problem von derartigen Emotionen beeinflusst werden: „[C]onsumer trade-offs can be qualitatively different from one another and [...] these differences have implications for choice patterns“ (M. F. Luce, Payne & Bettman, 1999, S. 157).

Negative Emotionen stellen sich vor allem dann ein, wenn das Abwägen zwischen Optionen Trade-offs notwendig macht, an denen Attribute beteiligt sind, die an hochgeschätzte Werte geknüpft sind (M. F. Luce, 1998). Beispiele dafür sind Sicherheit, Unabhängigkeit oder physische Unversehrtheit, also generell „weiche“ und somit „harten“ ökonomischen Werten entgegen gesetzte Konzepte. Baron und Spranca (1997) führen dafür den Begriff *geschützter Werte* ein, die eine erhöhte Trade-off-Resistenz aufweisen. Damit ist gemeint, dass ein Mangel an derartigen Werten nur in beschränktem Maße durch eine höhere Ausprägung bei anderen Attributen ausgeglichen werden kann. Wird dieses Konzept etwas allgemeiner gefasst, sollten darunter potenziell alle Attribute mit schlecht greifbaren, d. h. schwer quantifizierbaren oder unsicheren Konsequenzen gehören, was besonders beim selbstständigen, interaktiven Erschließen einer Software eine Rolle spielt. Fragen und Überlegungen wie die Folgenden erscheinen nahe liegend und natürlich: „Wie einfach ist ‘einfach’? Welche von zwei alternativen Umsetzungen einer Funktionalität wird sich auf Dauer bewähren? Lieber genial aber exotisch oder etwas umständlich aber vertraut? Das hat gut funktioniert, aber kriege ich das beim nächsten Mal auch so hin? Nicht, dass ich nachher nicht weiterkomme und mich mit meiner eigenen Auswahl blamiere ...“

Der Entscheider muss in solchen Fällen Abschätzungen und Priorisierungen vornehmen, die nur bedingt rein kognitiv möglich sind, weil die verfügbare Datenlage dafür nicht „hart“ genug ist. Das Entscheiden auf der Basis „weicher“ Faktoren ist jedoch mit Unsicherheit und deshalb bei entsprechender Wichtigkeit der Konsequenzen mit einem negativen affektiven Zustand verbunden („Ich fühle mich nicht wohl dabei ...“). In solchen Situationen wird die Vermeidung negativer Emotionen zum salienten Metaziel und interagiert mit Bestrebungen nach Aufwandsminimierung und Genauigkeitsmaximierung (vgl. Abschnitt 4.2.2.1 zur Auswahl von Entscheidungsstrategien). In mehreren Experimenten beobachteten M. F. Luce, Bettman und Payne (1997) bei zunehmender negativer Emotion einen Wechsel zu aufwendigeren Informationsverarbeitungsstrategien. Dies erscheint plausibel, setzt man ein der Wichtigkeit des Problems geschuldetes Bestreben, die *richtige* Entscheidung zu treffen, voraus. Anders als bei *kognitiv* schwierigen Problemen, bei denen extensiveres simultan mit mehr alternativenbasiertem Vorgehen auftritt (Payne, Bettman & Luce, 1996), erfolgt die Verarbeitung hier jedoch mehr attributbasiert. Durch

Fokussierung auf immer nur ein Attribut rücken Trade-offs in den Hintergrund; statt sie aufzulösen werden sie und die mit ihnen verbundenen negativen Emotionen vermieden.

Den Einfluss negativer Emotionsladung auf das *Ergebnis* von Entscheidungen haben M. F. Luce et al. (1999) untersucht. In verschiedenen Aufgaben sollten Versuchspersonen Optionen bewerten, die jeweils durch ein „weiches“, so genanntes Qualitätsattribut (z. B. Sicherheit eines Autos, Zustand eines Appartements) sowie einen Preis beschrieben waren. Dabei zeigte sich jeweils eine Präferenz für die Option mit einer höheren Ausprägung des Qualitätsattributs, wenn dieses emotionsbeladen war. Das heißt, dass sich die Versuchspersonen hinsichtlich wichtiger Werte repräsentierender Attribute eher für die bessere (aber teurere) als die billigere (aber schlechtere) Option entschieden. Darüber hinaus konnte gezeigt werden, dass eine solche Qualitätspräferenz vor allem dann auftritt, wenn die Attributwerte als Verluste wahrgenommen werden: Hat Person A bisher eine besonders komfortable Software benutzt, so werden zwei Optionen, die beide weniger komfortabel sind, als Verlust erscheinen. Ergo sollte A sich mit höherer Wahrscheinlichkeit für die bessere (und teurere) von diesen beiden entscheiden. Person B, für die beide Optionen eine Verbesserung darstellen und somit als Gewinn wahrgenommen werden, sollte diese Tendenz nicht zeigen.

Durch schwierige Trade-offs induzierte negative Emotionen können nicht nur Verlauf und Ergebnis des Entscheidungsprozesses beeinflussen, sondern auch dazu führen, dass die Entscheidung aufgeschoben oder ganz vermieden wird, wenn diese Option zur Verfügung steht (M. F. Luce, 1998). Diese Tendenz, emotionsbeladene Trade-offs zu vermeiden, kann jedoch gemindert werden, indem die kognitive Last des Individuums erhöht wird. In einer Untersuchung von Drolet und Luce (2004) wurde dies erreicht, indem man die Testgruppe vor der Bearbeitung eines Entscheidungsproblems eine Liste von Wörtern auswendig lernen ließ, die später abgefragt werden sollten. Diese Teilnehmer gaben mehr als doppelt so häufig wie die Kontrollgruppe Trade-offs, also das bewusste Abwägen von Vor- und Nachteilen, als Gründe für ihre Entscheidung an, zeigten also eine deutlich verminderte Trade-off-Aversion. Kognitive Last stört anscheinend die Fähigkeit, relevante Selbstziele bzw. die emotional negativen Konsequenzen von schwierigen Trade-offs zu berücksichtigen und reduziert dadurch das Bedürfnis nach Vermeidung.

4.4 Zusammenfassung und Schlussfolgerung

Die Literatur zur Psychologie von Konsumentenentscheidungen befasst sich im Wesentlichen mit der Erklärung *tatsächlichen* Verhaltens und hat insofern deskriptiven Charakter. Von Bedeutung für den hier zu untersuchenden Sachverhalt sind insbesondere kognitive und emotionale Prozesse, welche für die Steuerung bewusster Entscheidungen verantwortlich sind. Wichtige kognitive Determinanten sowohl des Entscheidungsverhaltens als auch des Ergebnisses sind die Vorkenntnisse des Individuums, die Schwierigkeit des Entscheidungsproblems sowie der Kontext, in den es eingebettet ist. Emotionen spielen in der Form von Stimmungen als „Dauertönung des Erlebens“ ebenso eine Rolle wie als Informationsquelle sowie als ggf. zu vermeidende Begleiterscheinung bzw. Resultat von Entscheidungen.

Wie die deskriptive Forschung zur Softwareauswahl im Besonderen (vgl. Kapitel 3) ist jedoch auch die zur Entscheidungsfindung von Konsumenten im Allgemeinen durch die fehlende Berücksichtigung von Entscheidungen auf der Basis nicht angeleiteter Erprobungen gekennzeichnet. Damit sind hier Untersuchungen gemeint, in der die Auswahl aus einer Menge *nicht* lediglich auf der Basis abstrakter Informationen über die angebotenen Alternativen erfolgt, sondern vielmehr aufgrund von konkreten Erfahrungen bei deren Erprobung in realen Anwendungszusammenhängen. „Nicht angeleitet“ verweist auf den Umstand, dass das auswählende Individuum nicht in die korrekte bzw. intendierte Benutzung der zu erprobenden Objekte eingewiesen wird.

Stattdessen werden im Bereich der Konsumentenforschung die für die Bewertung bzw. Entscheidung verfügbaren Informations-Items in aufbereiteter Form (typischerweise als Option×Attribut-Matrizen) bereitgestellt. Für die jeweils untersuchte Fragestellung (z. B. Studien zu *Verarbeitungsprozessen* oder zum Umgang mit Informationslücken) interessante Einschränkungen des Informationszugriffs bestehen üblicherweise darin, dass z. B. immer nur ein Item angeschaut werden kann, die Items unsortiert sind oder bestimmte Items gar nicht zur Verfügung stehen. Die untersuchten Konsumentenentscheidungen basieren somit nicht auf der Erprobung der zur Auswahl stehenden Produkte, sondern auf der Evaluation abstrakt dargebotener Informationen.

Im Bereich der Forschung zur Softwareauswahl gibt es neben Studien der oben skizzierten Form zwar auch einige, in denen die Bewertung durch bzw. Auswahl nach Interaktion mit den Alternativen erfolgt. Allerdings sind dort die Individuen zunächst stets in der Funktionsweise der Programme unterwiesen worden. Insofern handelt es sich um angeleitete Erprobungen. Es gab keinen Zweifel über die korrekte Benutzung; entscheidungsrelevant war in dieser Hinsicht lediglich die Kongruenz zu den Nutzungsgewohnheiten des jeweiligen Individuums.

Es gibt jedoch keine Untersuchungen, in denen eine für die Entscheidung benötigte Information aus der konkreten Funktionsweise eines Entscheidungsgegenstandes (z. B. einer Software) besteht *und* diese Information durch direkte Auseinandersetzung mit diesem Gegenstand extrahiert werden muss. Die Besonderheit der hier zu untersuchenden Fragestellung besteht also darin, dass sich die dem Entscheider zur Verfügung gestellten Informationen in der Überlassung der Programme erschöpfen. Eine Einweisung in die Funktion oder eine Bereitstellung von Metamaterial wie Option×Attribut-Matrizen (z. B. Expertenratings) sind nicht vorgesehen.

5 Empirische Untersuchung

Dieses Kapitel dient der Beschreibung der durchgeführten empirischen Untersuchung zur Klärung der Forschungsfragen. Zunächst werden der explorative Ansatz dieser Studie sowie das Laute Denken als Methode zur Datenerhebung erläutert. Da die Exploration keinem standardisierten Vorgehen folgt, werden anschließend ausführlich die Entwicklung der konkreten Untersuchungsanlage sowie die Aufbereitung der erhobenen Daten für die Auswertung beschrieben. Da sich die Auswertungsmethoden selbst erst durch die Auseinandersetzung mit den erhobenen Daten ergeben, werden sie erst im nächsten Kapitel in unmittelbarem Zusammenhang mit den Ergebnissen dargestellt.

5.1 Forschungsfragen

Die im Folgenden beschriebene empirische Untersuchung wird durch das Ziel geleitet, einen ersten Zugang zu den in Kapitel 1 aufgeworfenen Fragen zu liefern und dadurch zu ihrer Klärung beizutragen. Diese Fragen beziehen sich auf das Verhalten von Individuen bei der Ad-hoc-Evaluation von Software sowie ihre Gründe für die Auswahl eines Programms aus einer Menge und können als Ergebnis des Literaturstudiums weiter konkretisiert werden.

Durch die Literatur zur Psychologie der Entscheidung zieht sich wie ein roter Faden der Befund, dass das Verhalten von Konsumenten durch die konkreten Eigenschaften von Entscheidungssituation und -gegenstand sowie der Person des Entscheiders selbst abhängt (z. B. Bettman et al., 1998). Deshalb erscheint es sinnvoll, kein interindividuell einheitliches Vorgehen zu erwarten, sondern vielmehr nach Unterschieden bzw. typischen Charakteristika im Vorgehen der Individuen sowie Hinweisen auf deren Einfluss auf das Auswahlergebnis zu suchen.

Darüber hinaus legen einige Befunde nahe, dass insbesondere die einschlägige Erfahrung eines Individuums Einfluss auf den Prozess und das Ergebnis der Entscheidung nehmen könnte. Zum einen haben Galletta et al. (1993) am Beispiel von Datenbankmanagementsystemen gezeigt, dass zwischen Experten und Novizen Unterschiede im Vorgehen bei der Evaluation von Software bestehen (vgl. Abschnitt 3.3). Zum anderen sind solche Unterschiede auch aufgrund der Befundlage hinsichtlich individueller Vorkenntnisse beim Treffen von Konsumentenentscheidungen zu erwarten (z. B. Brucks, 1985; Peracchio & Tybout, 1996; vgl. Abschnitt 4.2.3.2).

Die Forschung zur Attraktivität und Akzeptanz von Software (z. B. Hassenzahl et al., 2003; Kemp et al., 2002; vgl. Abschnitt 3.3) sowie zum Technology Acceptance Model (TAM; Davis, 1989; Davis et al., 1989; vgl. Abschnitt 3.4) zeigt, dass die Bewertung von Software bzw. die Absicht, sie zu benutzen, wesentlich durch einige wenige, allgemein-

gültige abstrakte Faktoren beeinflusst wird. Deshalb erscheint es Erfolg versprechend, auch bei der hier zu untersuchenden Ad-hoc-Evaluation und -auswahl von Software nach solchen Aspekten zu suchen. Interessant erscheint zudem die Frage, ob diese Aspekte analog zum TAM (Davis et al., 1989) bzw. der Attraktivitätsforschung (Hassenzahl et al., 2000) in einem Zusammenhang stehen.

Entsprechend dieser Überlegungen soll die empirische Untersuchung durch folgende Forschungsfragen geleitet werden:

1. Auf welche Weise erfolgt die Evaluation, d. h. Erforschung und Prüfung, sowie der Vergleich der zur Auswahl stehenden Programme?
 - a) Lassen sich verschiedene Herangehensweisen der einzelnen Individuen im Sinne von Bearbeitungsstrategien oder Arbeitsstilen unterscheiden? Falls ja, worin bestehen die typischen Charakteristika der verschiedenen Herangehensweisen?
 - b) Gibt es Zusammenhänge zwischen einschlägiger Erfahrung, Bearbeitungsstil und Auswahlresultat?
2. Wovon hängt es ab, welche Software ausgewählt wird?
 - a) Gibt es allgemeingültige Aspekte (z. B. Attribute, kritische Ereignisse), die zur Präferenz bzw. zur Ablehnung einer Software führen?
 - b) In welchem Zusammenhang stehen diese Aspekte?

5.2 Forschungsmethode

5.2.1 Exploration und heuristische Beobachtung

Hinsichtlich der Frage, wie Individuen durch ungeplantes Ausprobieren ein Produkt aus einer Menge auswählen und welche Aspekte für das Ergebnis ausschlaggebend sind, können aus der bestehenden Theorie nur sehr spekulative Hypothesen abgeleitet werden. Der Grund dafür liegt darin, dass hier ein bisher unberücksichtigter Modus der Informationsbeschaffung zur Anwendung kommt (vgl. Abschnitt 4.4), von dem intuitiv weitere, jedoch unklare Wirkungen auf Verlauf und Ergebnis des Entscheidungsprozesses zu erwarten sind. Insofern wird mit der vorliegenden Untersuchung Neuland betreten, weshalb es angebracht erscheint, sich explorativ und Hypothesen generierend (statt testend) mit der Fragestellung auseinanderzusetzen: „In hypothesis generating research, the researcher explores a set of data searching for relationships and patterns, and then proposes hypotheses which may then be tested in some subsequent study“ (Hartwick & Barki, 1994, S. 447).

Exploratives Vorgehen bedeutet, dass nicht von schon Bekanntem, aus dem Hypothesen abgeleitet wurden, ausgegangen wird. Vielmehr zielt es auf Neues, auf Entdeckungen (z. B. Bortz & Döring, 2006). Zu diesem Zweck erscheint es sinnvoll, den zu explorierenden Sachverhalt natürlich zu halten und so wenig wie möglich zu manipulieren. Das

bedeutet in diesem Fall, dass verschiedene Versuchspersonen mit einem möglichst realistischen Softwareevaluations- und -auswahlproblem konfrontiert und bei dessen Bearbeitung beobachtet werden. Das Ziel der Beobachtung besteht in der vorliegenden Untersuchung darin, zu einem ersten, allgemeinen, ganzheitlichen Verständnis von Ad-hoc-Evaluation und -auswahl von Software zu kommen, das als Ausgangspunkt für die Beantwortung weiterer, detaillierterer Fragen dienen kann. Beobachtung ist hier vor dem Hintergrund der Exploration als *heuristische* Beobachtung zu verstehen. Die Erläuterung dieses Begriffs durch Greve und Wentura (1997) trifft genau den methodischen Ansatz dieser Untersuchung:

Der Forscher beteiligt sich nicht aktiv, er mischt sich nicht ins Geschehen ein, er kontrolliert und manipuliert nicht, er „lässt sich auf den Gegenstand ein“, ohne bereits konkrete explizite Vermutungen oder gar Theorien zu haben, d. h. ohne von vorneherein auf bestimmte Aspekte besonders oder ausschließlich zu achten und dabei andere zu ignorieren oder zu vernachlässigen. Er betrachtet seinen Gegenstand vielmehr so unvoreingenommen und so vollständig wie irgend möglich. [...]

Der heuristische Beobachter hofft, daß sich das, was er da betrachtet, zu einer Struktur, zu einer Systematik ordnet, daß er das Muster, das sich hinter dem individuellen Durcheinander verbirgt, nach und nach erkennt, daß er interessante Zusammenhänge *entdeckt*. Darin liegt die besondere Stärke der heuristischen Beobachtung: Mit einer sozusagen „gleichschwebenden“ Aufmerksamkeit nach und nach die Kennerschaft erwerben, aus der die guten Ideen erwachsen, weil einem Dinge auffallen, die durch ihre Ungewöhnlichkeit aus den unmerklich ausgebildeten Strukturen herausragen oder so prägnante Muster bilden, daß sie zu einer expliziten Struktur werden. Diese Beobachtung ist nicht (explizit) theoriegeleitet, kann aber insofern systematisch sein, als sie um Erfassung und *Ordnung* bestimmter (weiter) Ausschnitte des Geschehens systematisch bemüht ist. (S. 21, Hervorhebungen im Original)

Damit wird auch deutlich, dass das hier gewählte Vorgehen der nicht-experimentellen Forschung zuzuordnen ist: Diese unterscheidet sich von der experimentellen im Wesentlichen dadurch, dass der Forscher beim Experiment systematisch eine oder mehrere unabhängige Variablen manipuliert (O. Huber, 2005; Hussy & Jain, 2002), was hier nicht der Fall ist. Ziel eines Experimentes ist die Prüfung von Kausalhypothesen, d. h. Ursache-Wirkungs-Beziehungen zwischen unabhängigen und abhängigen Variablen. Auch dies trifft hier nicht zu, da Hypothesen aus den Untersuchungsergebnissen überhaupt erst abgeleitet werden sollen: „Der heuristische Beobachter beobachtet, um Hypothesen, Zusammenhänge, vielleicht sogar Theorien zu *generieren*, der Experimentator experimentiert, um Hypothesen, Zusammenhänge oder Theorien zu *überprüfen*“ (Greve & Wentura, 1997, S. 22, Hervorhebungen im Original).

5.2.2 Lautes Denken

Die folgenden Ausführungen sind gliederungslogisch eigentlich der Datenerhebung zuzuordnen. Da aber auf die verwendete Methode, das Laute Denken, bei den Ausführungen zur Entwicklung der Untersuchungsanlage an verschiedenen Stellen Bezug genommen wird, erfolgt bereits an dieser Stelle eine Einführung. Spezielle Aspekte werden dann jeweils im Zusammenhang mit der Anwendung der Methode, insbesondere in Abschnitt 5.3.6, diskutiert.

Die Beobachtung der Versuchspersonen bei der Bearbeitung eines Auswahlproblems soll unter anderem Vermutungen darüber ermöglichen, welche Erlebnisse im Umgang mit einer Software Einfluss auf deren Beurteilung bzw. Akzeptanz oder Ablehnung haben. Dafür ist es erforderlich, Zugang zu den mit einzelnen Interaktionen sowie deren Ergebnissen verbundenen Kognitionen zu erlangen. Zu diesem Zweck bietet sich die Methode des Lauten Denkens an (Ericsson & Simon, 1980; Newell & Simon, 1972; Someren, Barnard & Sandberg, 1994; Wagner, 1994): Die Versuchsperson wird aufgefordert, parallel zur Bearbeitung eines Problems alle Gedanken auszusprechen, die ihr dabei durch den Kopf gehen. Diese Verbalisierungen werden aufgezeichnet und können anschließend analysiert werden. Dabei lassen sich im Allgemeinen zwei Vorgehensweisen unterscheiden (Deffner, Heydemann & Borstel, 1984): Zum einen die Protokollanalyse zur Entwicklung eines Modells des Problemlöseprozesses (Newell & Simon, 1972), zum anderen inhaltsanalytische Ansätze, bei denen sprachliche Äußerungen zunächst kategorisiert und dann mit verschiedenen, in der Regel quantitativen Methoden weiter ausgewertet werden (z. B. G. L. Huber & Mandl, 1994; Mayring, 2007).

Das von Ericsson und Simon (1980) entwickelte Modell des Lauten Denkens basiert auf der Vorstellung menschlicher Informationsverarbeitung als ein System von Speichern mit unterschiedlicher Kapazität, Zugriffszeit und Persistenz. Die Autoren gehen davon aus, es gäbe: „several sensory stores of short duration, a short-term memory (STM) with limited capacity and/or intermediate duration, and a long-term memory (LTM) with large capacity and relatively permanent storage, but with slow fixation and access times“ (ebd., S. 223). Außerdem wird ein Zentralprozessor angenommen, der kontrolliert, welche Informationen ins STM gelangen, um dort für die aktuellen kognitiven Prozesse verwendet zu werden. Die Steuerung des Informationsflusses zum STM wird als Prozess der Aufmerksamkeit verstanden, d. h. die betreffenden Informationseinheiten werden sozusagen beachtet. Hinzu kommt die Annahme, dass die kürzlich beachteten Informationen im STM gespeichert sind und dadurch *unmittelbar* für die weitere Verarbeitung, z. B. für die Produktion von Verbalisierungen, zur Verfügung steht. Inhalte des LTM dagegen müssten für solche Zwecke erst aufgespürt und ins STM transferiert werden.¹

Aus dem so skizzierten Modell menschlicher Informationsverarbeitung folgt, dass Verbalisierungsprozesse stets jene Informationen produzieren (externalisieren), die sich im

¹ Die Autoren merken an, dass es für den vorliegenden Zusammenhang jedoch keinen wirklichen Unterschied mache, ob es tatsächlich *verschiedene* Speicher oder aber *einen einzigen* homogenen Speicher mit verschiedenen Zugriffsmodi gibt. Wichtig sei lediglich die Annahme, dass nur die Inhalte des STM (oder des jeweiligen Äquivalents) *direkt* zugreifbar sind (Ericsson & Simon, 1980; vgl. zur Struktur des Gedächtnisses auch J. R. Anderson, 2007).

STM befinden. „In the case of thinking-aloud instructions, the information verbalized will then be some portion of the information *currently being attended to*“ (ebd., S. 225, Hervorhebung J.Z.). Dieser Verbalisierungsmodus wird *concurrent* (simultan, nebenläufig) genannt. Im Fall einer Instruktion zum *retrospektiven* (rückblickenden) Verbalisieren dagegen, die nicht parallel, sondern erst nach Ausführung des interessierenden Prozesses erteilt wird, kommt das Modell zu anderen Vorhersagen: „some previously heeded information will still be in STM, permitting direct reporting [...] and facilitating retrieval of *additional information stored in LTM* in episodic associations that were formed when the information was heeded. The control process in retrieving previously heeded information from LTM, however, may be rather variable“ (ebd., S. 226, Hervorhebung J.Z.).

Deshalb hat nebenläufiges Lautes Denken gegenüber retrospektiven Methoden der Datenerhebung vor allem Vorteile hinsichtlich der Validität (vgl. auch Someren et al., 1994), die daraus resultieren, dass die Verbalisierung parallel zur Handlungsausführung erfolgt und deshalb unmittelbar die dabei auftretenden Kognitionen externalisiert. Wird ein Subjekt dagegen erst nach der Bearbeitung eines Problems zu den dabei aufgetretenen Kognitionen befragt (z. B. strukturierte Techniken wie Fragebogenerhebung), kann es leicht zu Fehlern aufgrund fehlender, falscher und/oder durch Interpretationen verzerrter Erinnerungen kommen. Dabei handelt es sich um ein „common problem with paper-and-pencil measures, which require the participant to reflect back over a host of situations and then draw a general conclusion about his or her characteristic ways of thinking“ (Davison, Vogel & Coffman, 1997, S. 952). Solche Fehler können zwar gemildert werden, indem das Individuum nicht erst nach, sondern regelmäßig oder anlassbezogen während der Problembearbeitung nach seinen aktuellen Gedanken befragt wird („Prompting“; Someren et al., 1994, S. 23). Dieses Vorgehen hat jedoch den Nachteil, dass es die Versuchsperson in ihrem natürlichen Handlungsablauf unterbricht und diesen dadurch beeinflusst. Beim nebenläufigen Lauten Denken kann dagegen von einer wesentlichen Übereinstimmung der Verbalisierungen mit den tatsächlichen Gedanken sowie davon ausgegangen werden, dass die Aktivität des Verbalisierens keinen störenden Einfluss auf den Handlungsablauf nimmt (Deffner, 1984; Ericsson & Simon, 1980).

Für die vorliegende Studie erscheint die Methode des Lauten Denkens insbesondere geeignet, weil der zu untersuchende Sachverhalt wegen des Fehlens von Hypothesen zunächst möglichst vollständig zu erfassen ist, ohne dass ausgewählte Aspekte besonders im Fokus stehen. Darüber hinaus geht es darum, wie einzelne Interaktionen erlebt werden. Interessant sind also die Kognitionen, die jeweils im Zusammenhang damit auftreten, und nicht, wie dieses Erleben retrospektiv gewertet wird.

5.3 Entwicklung und Aufbau der Untersuchung

5.3.1 Pilotphase

Die Entwicklung der konkreten Untersuchungsanlage erfolgte in einer Pilotphase, in deren Verlauf vor allem verschiedene Formulierungen für die Aufgabenstellung an die Pro-

banden getestet wurden. Außerdem ging es darum, den Versuchsleiter² mit der Methode des Lauten Denkens vertraut zu machen sowie ihn in diesem Zusammenhang auf mögliche Probleme im Versuchsablauf vorzubereiten. Zudem mussten verschiedene Softwarealternativen getestet werden. Die Pilotphase umfasste neun Sitzungen mit jeweils anderen Versuchspersonen und ging gleitend in die Hauptphase über, indem in den letzten beiden Pilotsitzungen mit unwesentlichen Abweichungen bereits die finale Aufgabenstellung und Ablauf verwendet wurden. Die Aufzeichnungen aus diesen beiden letzten Sitzungen wurden in die Auswertung einbezogen, die aus den ersten sieben Sitzungen wurden nicht weiter verwendet.

5.3.2 Bereitstellung der zu erprobenden Software

5.3.2.1 Softwarekategorie

Bei der Auswahl der durch die Versuchspersonen jeweils zu evaluierenden Programme musste zunächst eine Softwarekategorie gefunden werden, die folgenden Anforderungen genügt:

(1) Die Nützlichkeit der Software bzw. der Sinn der von ihr zu leistenden Funktionalität darf nicht erklärungsbedürftig sein.

In dieser Studie soll explorativ untersucht werden, wie Individuen, die eine Software zur Erfüllung einer benötigten Funktionalität suchen, ein Programm aus einer Menge von Alternativen auswählen. Der Forschungsansatz besteht entsprechend darin, ein möglichst realistisches Auswahlproblem zu kreieren. Dies kann nur gelingen, wenn der Versuchsperson der grundsätzliche Sinn oder der Nutzen einer Software selbstverständlich ist, nach Möglichkeit gar aus eigener Erfahrung. Aus diesem Selbstverständnis heraus müssen sich die Anforderungen an die Software ohne externe Vorgabe implizit ergeben. Dies ist unbedingt notwendig, um das Wesen einer Ad-hoc-Evaluation abzubilden (vgl. Abschnitt 2.3).

Wird dagegen die Aufgabe gestellt, ein Programm zur Lösung eines Problems auszuwählen, das der Versuchsperson bisher nicht bewusst war oder nicht einsichtig ist, wird das Auswahlproblem von vornherein künstlich sein. Zudem wäre davon auszugehen, dass der Auswahlprozess von Unsicherheit über das Verständnis dessen, was die Software leisten soll, bestimmt ist. Um dies auszuschalten, müsste der Versuchsleiter eine Liste der zu erfüllenden Anforderungen bereitstellen (externe Vorgabe), was einer expliziten Spezifikation entspricht. Dies steht jedoch im Widerspruch zu der hier zu untersuchenden Fragestellung. Die Evaluation durch den Probanden würde dann nicht ad hoc, sondern höchstwahrscheinlich durch ein Abarbeiten der vorgegebenen Liste erfolgen.

(2) Die von der Software zu leistende Funktionalität muss in Teilfunktionalitäten bzw. das zu lösende Problem in Bearbeitungsschritte zerlegbar sein.

Diese Forderung ergibt sich aus dem Umstand, dass unter anderem das Vorgehen verschiedener Individuen bei der Begutachtung von Software untersucht werden soll. Inter-

² Der Versuchsleiter ist mit dem Verfasser dieser Arbeit identisch.

essante Beobachtungen sind diesbezüglich aber nur dann zu erwarten, wenn dieses Vorgehen nicht bereits durch die Struktur des zu lösenden Problems determiniert ist. Dieser Fall wäre z. B. dann gegeben, wenn es zur Bearbeitung des Problems nur eines oder sehr weniger Mausklicks in vorgegebener Abfolge bedürfte. Jedoch ist nur, wenn mehrere Schritte in nicht zwingender Reihenfolge und Anzahl kombiniert werden müssen und somit individuelle Pfade möglich sind, zu erwarten, dass Unterschiede in der Herangehensweise erkennbar werden.

Ein weiterer Effekt einer nicht zu geringen Software- bzw. Problemkomplexität besteht in der resultierenden Notwendigkeit zur Benutzung mehrerer, verschiedener Teilfunktionalitäten. Dadurch müssen sich die Versuchspersonen in größerer Breite und Tiefe mit der Software auseinandersetzen und ggf. verschiedene Aspekte gegeneinander abwägen. Da diese Prozesse verbalisiert werden, sind reichhaltigere Aufschlüsse darüber zu erwarten, warum bestimmte Programme gewählt oder abgelehnt wurden.

(3) Die Software sollte mehr Funktionalitäten anbieten, als für die Bearbeitung des aktuellen Problems unbedingt benötigt werden.

Zur Erläuterung dieser Forderung sind zwei Punkte von Bedeutung. Zum einen ist es unrealistisch, dass eine Software nur und ausschließlich das „kann“, was zur Bearbeitung eines konkreten Problems gerade benötigt wird. Im Gegenteil sind vor allem kommerzielle, generische Computerprogramme für den einzelnen Anwender meist mit einem Übermaß an Funktionalität ausgestattet, um sie für eine möglichst große, im Detail unbekannte Zielgruppe attraktiv zu machen (vgl. Abschnitt 3.1). Deshalb ist es von erheblicher Relevanz sowohl für die Validität dieser Studie als auch die angestrebte Natürlichkeit der Laborsituation, dass die Versuchspersonen aus der Fülle der angebotenen Funktionalitäten die im Augenblick relevanten herausfinden müssen.

Zum anderen hat diese Forderung Einfluss auf die Ergiebigkeit des zu erhebenden Datenmaterials. Die Versuchspersonen müssen Gelegenheit haben, von dem durch Beschränkung auf das absolut Notwendige quasi vorgezeichneten Pfad abzuweichen und mit einzubeziehen, was eine Software sonst noch zu bieten hat. Ein vielleicht nicht unmittelbar benötigter Zusatznutzen könnte u. U. den Ausschlag für oder gegen eine Option geben, Mängel bei der eigentlichen Funktionalität ausgleichen, oder aber auch völlig unbeachtet bleiben. Zudem eröffnet ein Zusatznutzen die Möglichkeit, dass unterschiedliche Anspruchsniveaus Einfluss auf die Evaluationsprozesse nehmen: Wovon der eine nicht einmal etwas ahnt und entzückt über dessen Entdeckung ist, könnte für den anderen ein von spezialisierter Software unbedingt zu erwartendes Feature sein, dessen Fehlen zur Abwertung führt.³

(4) Die Interaktionen müssen hinsichtlich ihrer Intention und ihres Ablaufs gut nachvollziehbar sein.

Um als Beobachter vor allem nach dem Ende einer Sitzung den Ablauf der Evaluation nachvollziehen zu können, müssen die von der Versuchsperson unternommenen Interaktionen sowie deren Wirkungen gut erkennbar und somit die jeweilige Absicht, die damit

³ Im Sinne eines Motivators vs. Hygienefaktors der Zwei-Faktoren-Theorie (Herzberg, Mausner & Snyderman, 1959).

verfolgt wurde, möglichst transparent sein. Deshalb ist eine Softwarekategorie zu wählen, in der die Funktionalitäten vor allem mit der Maus, d. h. auf dem Bildschirm sichtbar und damit für den Beobachter nachvollziehbar abgerufen werden. Zudem müssen die Interaktionen in der Regel mit grafischen Manifestationen verbunden sein, sodass deren Erfolg auch ohne Kommentar durch die Versuchsperson möglichst zuverlässig abschätzbar ist. Für die Beobachtung ungeeignet sind dagegen Programme, die vor allem über Tastatureingaben gesteuert werden und keine gut erkennbaren Ergebnisse liefern. Solche Programme dürften zwar in Zeiten der grafischen Benutzeroberflächen selten sein, dennoch darf dieser Punkt nicht als selbstverständlich angesehen und unerwähnt oder unberücksichtigt bleiben.⁴

Die Berücksichtigung dieser Forderungen führte zur Entscheidung, Programme zur Erstellung von Flussdiagrammen zu verwenden. Flussdiagramme sind ein einfaches und intuitiv verständliches Notationsmittel zur grafischen Darstellung von Abläufen, wie z. B. Geschäftsprozessen. Sie bestehen aus miteinander verbundenen geometrischen Figuren mit festgelegter Bedeutung. Bei der Verwendung werden die Symbole mit Texten beschriftet, die dem jeweiligen Anwendungsfall entsprechen. Durch Pfeile werden sie zu einem Ablauf verbunden (vgl. Abb. 5.1).

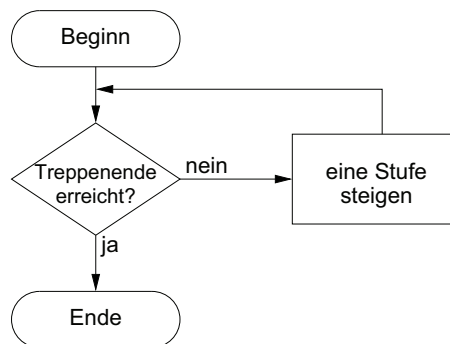


Abbildung 5.1: Beispiel eines einfachen Flussdiagramms (eigene Darstellung)

Die Nützlichkeit entsprechender Software, nämlich die Erstellung von Abbildungen aus einfachsten Elementen, ist unmittelbar einsichtig. Die zu leistenden Funktionalitäten sind überschaubar, bedürfen keiner besonderen Erklärungen und sind somit von allen Probanden leicht zu verstehen. Darüber hinaus ist davon auszugehen, dass die meisten Computerbenutzer in irgendeiner Form schon einmal mit dem Problem der Erstellung einer Abbildung konfrontiert waren (Forderung 1).

Der Umstand, dass Flussdiagramme aus mehreren unterschiedlichen Elementen (Symbole, Pfeile, Texte) bestehen, die alle einzeln erzeugt, beschriftet, miteinander verbunden und ggf. formatiert werden müssen oder können, stellt sicher, dass mehrere verschiedene Teilfunktionalitäten erschlossen und angewendet werden müssen, sodass verschiede-

⁴ Vgl. Gilb (1988, S. 39): „The principle of the obvious: ‘Obvious’ things, which ‘everybody knows’ cannot be left to take care of themselves.“

ne Bearbeitungspfade möglich sind. Je nach Herangehensweise kann jede Funktionalität auch mehrfach zum Einsatz kommen (Forderung 2).

Flussdiagramme bieten, trotz ihrer Einfachheit, einige Besonderheiten, denen durch eine Software in unterschiedlicher Weise Rechnung getragen werden kann. Als Beispiel wären die optisch ansprechende Positionierung der Texte in den geometrischen Symbolen (automatisch vs. manuell), deren Verknüpfung mit Pfeilen sowie die Ausrichtung der Elemente zueinander (automatisch vs. manuell, statisch vs. dynamisch) zu nennen. Zudem bieten die meisten Programme in verschiedenem Umfang Zusatznutzen an, wie z. B. Exportfilter, verschiedene Formatierungs- und Gestaltungsmöglichkeiten, Symbolbibliotheken für weitere Anwendungsgebiete usw. (Forderung 3).

Dadurch, dass das Ergebnis der Softwarebenutzung eine grafische Abbildung sein soll, ist zwangsläufig nahezu jeder Arbeitsschritt darauf gerichtet, einen visuell wahrnehmbaren Effekt hervorzurufen, z. B. das Erscheinen oder Verschieben eines Symbols, das Verkleinern einer Beschriftung usw. Darüber hinaus werden diese Funktionalitäten typischerweise durch gut verfolgbare Mausektionen abgerufen. Der Grund dafür liegt darin, dass mit Desktopsoftware in der Regel versucht wird, „normale“ Arbeitsgewohnheiten abzubilden (vgl. Abschnitt 2.1). Der Mauszeiger stellt in diesem Sinne den abstrakten Zeichenstift des Benutzers dar und sollte somit in alle relevanten Interaktionen involviert sein. Aus dem jeweiligen Bearbeitungsstand einer Abbildung kann zudem, jenseits einer eventuell ausbleibenden Verbalisierung, darauf geschlossen werden, was die Versuchsperson gerade vorhat (Forderung 4).

5.3.2.2 Präsentation der Alternativen

Wie die Simulation eines „möglichst realistischen Softwareevaluations- und -auswahlproblems“ (vgl. Abschnitt 5.2.1) konkret auszusehen hat, ist sicher nicht eindeutig zu beantworten. In der Realität sind im hier zu untersuchenden Zusammenhang prinzipiell zwei Abläufe denkbar:

Variante 1 beginnt mit dem Entschluss eines Individuums, zur Bearbeitung eines Problems eine Software zu beschaffen. Es lädt sich die Demoversion der ersten, aufgrund der Beschreibung geeignet erscheinenden Software, die es findet. Es probiert diese aus und kommt zu einem von zwei Ergebnissen: (a) Die Software wird für gut befunden und weiter verwendet. (b) Die Software wird für schlecht befunden und abgelehnt. Das Individuum lädt sich eine andere Demoversion und evaluiert diese. Das geht solange weiter, bis ein zufrieden stellendes Programm gefunden oder die Suche aufgegeben wird (entspricht im Wesentlichen der SAT-Heuristik; vgl. Abschnitt 4.2.2.2).

In Variante 2 lädt das Individuum z. B. aus demselben Internetportal, in dem es die erste Software gefunden hat, gleich noch weitere, ebenfalls geeignet erscheinende Programme. Dies ist ein sinnvolles Vorgehen, wenn ein schneller Internetanschluss zur Verfügung steht⁵ und es als größerer Aufwand empfunden wird, später erneut zu suchen. Alle Alternativen werden evaluiert, denn egal, wie das zuerst begutachtete Programm abschneidet:

⁵ Dies dürfte wegen der Größe der zu ladenden Dateien regelmäßig der Fall sein, wenn Softwarebeschaffung auf diesem Weg erfolgt.

Das nächste könnte ja noch besser sein und wurde ohnehin schon heruntergeladen. Wird keine der Alternativen für gut genug befunden, wird der nächste Satz Demoversionen beschafft und ebenso evaluiert, bis ein zufrieden stellendes Programm gefunden oder die Suche aufgegeben wird.

Folgende Überlegung spricht dafür, Variante 1 auszuschließen, also den erfolgreichen Abbruch der Bearbeitung des Auswahlproblems nach dem ersten Programm nicht zuzulassen: Ein Ziel der Untersuchung ist es, Aspekte zu identifizieren, die zur Akzeptanz oder Ablehnung einer Option führen. Für diesen Zweck verwendbare Anhaltspunkte sind in höherer Zahl zu erwarten, wenn eine Person mehr als eine Software betrachtet: Angenommen, ein Programm schneidet bei Aspekt A gut, bei Aspekt B schlecht ab und wird insgesamt für gut befunden. Wird die Evaluation jetzt abgebrochen, lässt sich die Hypothese aufstellen, dass A schwerer wiegt als B. Wird dagegen ein weiteres Programm betrachtet, das bei A schlecht und bei B gut abschneidet, und insgesamt besser bewertet wird als das Erste, ist diese Hypothese nicht mehr haltbar. Vielmehr ist jetzt anzunehmen, dass B wichtiger ist als A, oder dass A und B gleich schwer wiegen und ein weiterer Aspekt C den Ausschlag gegeben hat.

Aus diesen Überlegungen heraus wird bei der vorliegenden Untersuchung ein Durchlauf der oben skizzierten Variante 2 zugrunde gelegt, d. h. die Versuchspersonen wählen jeweils aus einer festen Anzahl von Optionen. Für die Simulation im Labor sollten die Suche und das Herunterladen der Software jedoch entfallen, da dies nicht Gegenstand der Untersuchung ist und somit lediglich die Zeit und Aufmerksamkeit der Versuchspersonen verschwenden würde. Zudem würde es durch die unkontrollierte Vorauswahl sehr leicht zu einer nicht mehr handhabbaren Heterogenisierung des auszuwertenden Materials kommen. Deshalb muss der Versuchsleiter eine Vorauswahl treffen und diese Programme lokal zur Verfügung stellen.

Zu klären bleibt noch die Frage nach der Anzahl der zur Auswahl zu stellenden Optionen. Eine obere Grenze ist sinnvoll im Interesse der Handhabbarkeit des zu erhebenden Datenmaterials, eine untere für dessen Ergiebigkeit. Ein Angebot von zwei Programmen erfüllt bereits die Anforderungen an ein Auswahlproblem im engeren Sinne (vgl. Abschnitt 2.3) und verspricht übersichtliches Material, in dem die interessierenden Prozesse repräsentiert sind. Die Hinzunahme einer dritten Option erscheint jedoch sinnvoll: Zum einen, um bei besonders wortkargen Probanden trotzdem genügend Material zu erhalten, zum anderen, um für den nicht auszuschließenden Fall von Softwarefehlern gewappnet zu sein.

5.3.2.3 Verwendete Programme

Für die Vorauswahl wurden zunächst acht Programme von verschiedenen Softwareportalen zur Verwendung in der Pilotphase geladen. In deren Verlauf zeigte sich, dass die für die Benutzeroberfläche in der Regel verwendete englische Sprache zu unerwünschten Verzerrungen der Auswahlentscheidung führen kann. Deshalb wurde nochmals gezielt nach Alternativen mit deutschen Versionen gesucht. In der vorliegenden Untersuchung wurden schließlich folgende Programme verwendet:

SmartDraw 6 von SmartDraw.com⁶ war zum Zeitpunkt der Untersuchung eines von zwei Programmen, von dem eine deutschsprachige 30-Tage-Testversion angeboten wurde. Die Testperiode wurde auf Anfrage durch den Hersteller verlängert. Als einzige Einschränkung verblieb ein Wasserzeichen des Herstellers auf den Ausdrucken der erzeugten Diagramme. Die Software wurde als günstige und leistungsfähige Alternative zu deutlich teureren, etablierten Produkten z. B. aus dem Hause Microsoft vermarktet.

Flowcharter 2003 von iGrafx⁷ konnte ebenfalls als deutschsprachige Testversion ohne Einschränkung der Funktionalität geladen werden. Tatsächlich verwendet wurde jedoch eine funktionsgleiche Vollversion, die dem Versuchsleiter auf Anfrage nach Verlängerung der Testperiode vom Hersteller unentgeltlich für den Einsatz in der Studie zur Verfügung gestellt wurde.

Microsoft⁸ *Visio 2003* konnte als uneingeschränkte Vollversion genutzt werden, da dem Versuchsleiter eine Lizenz des Lehrstuhls für Wirtschaftsinformatik an der Universität Rostock zur Verfügung gestellt wurde.

Diagram Studio 3.0 von Gadwin⁹ wurde als Testversion genutzt, deren einzige und für die Zwecke dieser Untersuchung unerhebliche funktionale Einschränkung darin bestand, dass die erstellten Abbildungen nicht gespeichert, gedruckt oder exportiert, d. h. im Prinzip nicht weiter verwendet werden konnten. Bei dieser Software handelt es sich um einen so genannten Klon von Microsoft Visio, d. h. sie kopiert dessen Funktionalität, Erscheinungsbild und Bedienlogik mit nur geringen Abweichungen. Obwohl im Prinzip mit Visio austauschbar, wurde Diagram Studio nur als englischsprachiger Ersatz für Visio verwendet, wenn eine Versuchsperson bereits vorher Kontakt mit Visio hatte. Diese Situation war aufgrund der hohen Verbreitung von Microsoft-Produkten nicht unwahrscheinlich und ist tatsächlich auch mehrfach eingetreten.

Mögliche Verzerrungen aufgrund der Englischsprachigkeit von Diagram Studio wurden als unwahrscheinlich in Kauf genommen, weil davon ausgegangen wurde, dass die Sprache der Benutzeroberfläche gegenüber deren struktureller Vertrautheit eine untergeordnete Rolle spielen würde. Wichtiger erschien es, neben SmartDraw und Flowcharter eine formal unbekannte dritte Option anzubieten, die aber in der Auswertung mit Visio zusammengefasst werden konnte.¹⁰ Tatsächlich könnte die Englischsprachigkeit sogar dazu beitragen, dass Diagram Studio als neues, unbekanntes Programm wahrgenommen und behandelt wird, zusätzlich zu dem Umstand, dass sich die Versuchsperson, trotz der offensichtlichen Entsprechung zu Visio, nicht vollständig sicher sein konnte, dass Diagram Studio wirklich auch genauso funktioniert.

Die genannten Programme wurden installiert und konnten von den Probanden direkt vom Windows-Desktop über entsprechend beschriftete Icons gestartet werden. Die Über-

⁶ www.smartdraw.de

⁷ www.igrafx.de

⁸ www.microsoft.de

⁹ www.gadwin.com

¹⁰ Bei den folgenden Ausführungen, insbesondere bei der Vorstellung der Ergebnisse in Kapitel 6, wird die je nach Versuchsperson mit Visio oder Diagramm Studio besetzte Option aus Gründen der besseren Lesbarkeit sowie der harmonischeren Formatierung von Tabellen mit „Visio/DS“ bezeichnet.

legung, die Icons neben dem Namen auch mit dem jeweiligen Preis der Software zu versehen, wurde im Verlauf der Pilotphase verworfen: Zum einen geht es hier nicht um eine Kaufentscheidung. Diese steht erst am Ende einer üblicherweise 30tägigen Testperiode an. Zum anderen weiß der Proband, dass er den Preis ohnehin nicht zahlen muss, sodass diese Information also nicht unbedingt die normalerweise zu erwartende Wirkung entfalten wird. Stattdessen könnte es eventuell sogar zu Verzerrungen kommen, indem der Preis als Indiz für die Qualität eines Programms gewertet wird, ein hoher Preis also die Präferenz verstärkt, statt sie zu mindern, und umgekehrt (Cialdini, 2002).

5.3.3 Aufgabenstellung

5.3.3.1 Vorüberlegungen

Bei der Aufgabenstellung oder Instruktion an den Probanden geht es darum, ihn dazu zu bringen, dass er ihm geeignet erscheinende Handlungen ausführt, um ein bestimmtes, vorgegebenes Ziel zu erreichen, hier: eine Auswahl aus drei zur Verfügung stehenden Programmen zu treffen. Es kommt also zum einen darauf an, dass der Proband grundsätzlich bereit ist, die Instruktion des Versuchsleiters zu befolgen, zum anderen darauf, dass ihm klar wird, was dieser von ihm erwartet. Die Bereitschaft des Probanden entspricht seiner Motivation und entsteht aus der Wechselwirkung zwischen seinen Motiven und situativen Anreizen, die deren Befriedigung versprechen (z. B. J. Heckhausen & Heckhausen, 2006). Das Vorliegen einer Zielvorstellung führt nach Ach (1935, zit. nach Kiesel, 2003, S. 26f) im Zusammenwirken mit distalen Reizen zur Auslösung einer sog. *determinierenden Tendenz*, die ihrerseits Handlungen zu deren Erreichung bewirkt. Diese Tendenz ist umso stärker, d. h. desto rascher und vor allem sicherer wird das Ziel erreicht, je spezifischer die Zielvorstellung ausfällt.

Der Verlauf menschlichen Handelns wird durch H. Heckhausens (1987) Rubikon-Modell beschrieben. Nach diesem Modell beginnt dieser Verlauf mit einer Phase des Abwägens verschiedener Wünsche und Handlungsoptionen (prädezyonale Motivation), gefolgt von einer Phase des Planens konkreter Handlungsstrategien (präaktionale Volition¹¹). Anschließend kommt es zur Durchführung dieser Strategien (aktionale Volition). Der Handlungsverlauf wird abgeschlossen durch eine Phase der Bewertung des Handlungsergebnisses (postaktionale Motivation).

Von besonderem Interesse für die Formulierung einer Aufgabenstellung sind die Volitionsphasen, denn hier geht es darum, dass der Proband auf der Basis dessen, was von ihm verlangt wird, konkrete Handlungspläne entwickelt und durchführt. Das motivationale Problem der grundsätzlichen Handlungsbereitschaft eines Probanden dagegen stellt sich bereits bei dessen Rekrutierung. Mit seinem freiwilligen Erscheinen zur Teilnahme an der Untersuchung kann es als gelöst betrachtet werden und bedarf im Zusammenhang mit der

¹¹ „Der Begriff *Volition* verdeutlicht den Unterschied zwischen den beiden grundlegenden Aufgaben der Erklärung des Handelns: Nachdem eine Entscheidung für eine Handlungsalternative gefallen ist, wird ihre Realisierung *gewollt*, d. h. die Handlungsvorbereitung und das Handeln werden durch den Willen gesteuert“ (Nerdinger, 1995, S. 78, Hervorhebungen im Original).

Formulierung der Aufgabenstellung keiner vordergründigen Berücksichtigung mehr. Die Phase der postaktionalen Motivation liegt indessen jenseits der Untersuchung, denn wie der Proband sein Handeln nachträglich bewertet und welche Auswirkungen dies auf künftiges Handeln hat ist hier nicht von Interesse und muss deshalb nicht beeinflusst werden.

Ein Hauptanliegen der Instruktion muss es zunächst also sein, dass der Proband erfährt, „was seine Aufgaben sind und v. a., was er bei der Erfüllung der Aufgaben erreichen soll. Mit anderen Worten, es müssen *Ziele* [...] gesetzt werden, damit der Neuling [hier: der Proband] weiß, was von ihm erwartet wird und daran sein Arbeitshandeln ausrichten kann“ (Nerdinger, 1995, S. 78, Hervorhebung im Original). Das Setzen von Zielen ist von zentraler Bedeutung für die Steuerung menschlichen Handelns: „Ziele veranlassen Handlungen, sie organisieren die Handlungen und lenken sie auf die angestrebten Ergebnisse“ (Kleinbeck, 2006, S. 256).

In der präaktionalen Volitionsphase geht es für den Probanden darum, „sich Gedanken darüber zu machen, auf welche Weise er das [...] gesetzte Ziel auch wirklich realisieren will“ (Achtziger & Gollwitzer, 2006, S. 280). Maßnahmen zur Zielerreichung können routinierte, bereits eingeübte Verhaltensweisen sein, die mehr oder weniger automatisch ablaufen, aber auch solche, die noch nicht etabliert sind und weiterer Elaboration bedürfen (ebd.). Auf die dabei ablaufenden Prozesse muss die Aufgabenstellung einwirken, um zu erreichen, dass das vom Probanden im Labor ausgeführte Handeln möglichst dem beim Bearbeiten eines entsprechenden realen Problems entspricht, wie es im Rahmen der vorliegenden explorativen Untersuchung wünschenswert ist.

Auf dem Feld der Arbeitspsychologie gibt es zahlreiche Befunde zur Wechselwirkung zwischen der Art der Zielsetzung und dem zur Erreichung der Ziele gezeigten Verhalten bzw. der resultierenden Leistung, die auf der Zielsetzungstheorie von Locke und Latham (1990) aufbauen (einen Überblick liefert z. B. Kleinbeck, 2006). Die vor diesem Hintergrund für die Formulierung am wichtigsten erscheinenden Dimensionen sind die Spezifität von Zielen sowie die Zielbindung.

Hinsichtlich der *Spezifität* von Zielen gibt es eine eindeutige Befundlage: Spezifische bzw. konkrete Ziele führen zu besseren Leistungen als vage (Tubbs, 1986). Die zugrunde liegenden Untersuchungen variieren in der Regel die Spezifität der Quantität eines Zieles (im Sinne von „Erhöhen Sie den Umsatz“ vs. „Erhöhen Sie den Umsatz um 10%“). Jedoch ist es unmittelbar plausibel, die Spezifität eines Zieles auch im Sinne von „Art und Umfang zur Verfügung gestellter Informationen“ zu interpretieren, die dabei helfen zu verstehen, worin das Ziel inhaltlich bzw. qualitativ überhaupt besteht. In der vorliegenden Untersuchung ist eine Leistung umso besser, je authentischer sie ist. Deshalb sollte auch die Aufgabenstellung möglichst authentisch in dem Sinne sein, dass die Probanden eine klare und einheitliche Vorstellung davon erhalten, worauf es ankommt.

Tatsächlich zeigte sich in der Pilotphase sehr schnell, dass relativ vage Anweisungen wie „Überprüfe und entscheide, ob du eine bestimmte Software zur Erstellung von Flussdiagrammen verwenden würdest“ oder „Überprüfe und entscheide, welches der drei Programme du am besten findest“ ungeeignet sind, weil eben die Frage, „worauf es ankommt“, offen bleibt: Auch wenn einer Person der Aufbau und die Bedeutung eines Flussdiagramms völlig klar sind, fällt es schwer, sich ohne ein konkretes Anwendungsziel für eine solche Software in die Rolle des potenziellen Kunden zu versetzen. Es fehlt die

Orientierung, in welchem Maße man sich mit der Software auseinanderzusetzen hat und welche Aspekte wichtig sind, bzw. ein Ziel, auf das hin die Evaluation erfolgen soll.¹²

Deshalb bleibt es dem individuellen Verständnis (*Aufgabenschema*; Hackman, 1969; vgl. auch Rosenstiel, 2007) des Probanden überlassen, welche bzw. in welchem Umfang eine Software Funktionalitäten zur Erstellung von Flussdiagrammen bieten sollte. Individuelle Unterschiede bei den Probanden sind zwar gewünscht und interessant, dürfen aber nicht auf der Interpretation des durch die Software zu lösenden Problems beruhen. Um solche Unterschiede würdigen zu können, muss es deshalb gelingen, eine einheitliche, klare Aufgabenstellung zu schaffen, die andererseits aber auch den Spielraum zu deren Entfaltung offen lässt. Aus diesem Grund sollte es jedoch möglichst vermieden werden, zum Zweck der Evaluation das Erzeugen einer konkreten, vorgegebenen Abbildung zu verlangen. Dies würde die angestrebte Authentizität der Zielsetzung beeinträchtigen und erscheint als Eingriff in das freie Evaluationsverhalten, weil dadurch das Handeln in konkrete, vorbestimmte Bahnen gelenkt wird.

Zielbindung, die zweite hier als besonders relevant erachtete Dimension, beschreibt das Gefühl der Verpflichtung gegenüber einem Ziel (Nerdinger, 1995), und wird auch interpretiert als der Grad der zur Zielerreichung investierten Anstrengung (Kleinbeck, 1996). Obwohl Locke und Latham (1990) sowohl einen direkten als auch einen moderierenden Effekt der Zielbindung auf die Leistung eines Individuums postulieren, wird in der anwendungsbezogenen Literatur vorrangig die Moderatorwirkung diskutiert (z. B. Kleinbeck, 2006; Nerdinger, 2001). Diese besteht darin, dass bei hoher Zielbindung ein positiver Zusammenhang zwischen der Schwierigkeit eines Ziels und der zu seiner Erreichung gezeigten Leistung besteht. Praktisch bedeutet dies, dass Zielbindung (unter anderem) dafür verantwortlich ist, in welchem Maße Ziele auch bei auftretenden Hindernissen und/oder Ablenkungen weiter verfolgt werden (ebda.).

In der vorliegenden Untersuchung ist das Erzeugen von Zielbindung wichtig, um, analog zur Spezifität, trotz der Künstlichkeit der Laborsituation eine größtmögliche Authentizität des Handelns der Versuchspersonen zu erreichen. In der zu simulierenden Realweltsituation hat das Ziel – das Ergebnis der Softwareauswahl – eine bestimmte Bedeutung oder Wichtigkeit für das handelnde Individuum, welche im Labor per se nicht gegeben und deshalb über die Aufgabenstellung zu erzeugen ist. Dies kann durch Einflussnahme auf die Bedingungsfaktoren der Zielbindung, insbesondere die Erwartungen hinsichtlich der Instrumentalität der Handlungsergebnisse, erfolgen (Kleinbeck, 1996, 2006).

Mit Instrumentalität ist ein Mittel-Zweck-Zusammenhang gemeint (vgl. VIE-Theorie; Vroom, 1964): Das Ergebnis einer Handlung kann in bestimmter Beziehung zu weiteren Konsequenzen¹³ stehen, diesen z. B. förderlich oder hinderlich sein. Die Stärke und Richtung dieser Beziehung entspricht der Instrumentalität des Handlungsergebnisses für die Handlungsergebnisse. Ein Individuum fühlt sich enger an ein Ziel gebunden, wenn es dieses Ziel als instrumentell für angestrebte Folgen wahrnimmt. Um beim Probanden eine möglichst realistische Zielbindung zu erreichen, muss das Ergebnis der Softwareauswahl

¹² Vgl. Gilb (1988, S. 28): „The principle of fuzzy targets: Projects without clear goals will not achieve their goals clearly. (You can't hit the bullseye, if you don't know where the target is!)“

¹³ Folgen des Ergebnisses der Handlung oder „Handlungsergebnisse“ (Nerdinger, 1995, S. 95f).

(seines Handelns) für ihn in einer ähnlichen Weise instrumentell sein, wie es in der zu simulierenden realen Situation der Fall wäre.

Im Zusammenhang mit den Überlegungen, wie die Versuchspersonen zu einer möglichst realistischen Bearbeitung des Auswahlproblems zu bewegen sind, ist auch der Begriff des Involvement zu diskutieren. Dieses vor allem in der Konsumentenforschung verwendete hypothetische Konstrukt bezeichnet die „Ich-Beteiligung bzw. gedankliches Engagement und die damit verbundene Aktivierung, mit der sich jemand einem Sachverhalt oder einer Aktivität zuwendet“ (Kroeber-Riel & Weinberg, 2003, S. 345). Die Stärke des Involvement wirkt sich auf das Informations- und Entscheidungsverhalten eines Individuums aus. Hohes Involvement führt zu aktiver Informationssuche und -verarbeitung. Dagegen werden Entscheidungen mit niedrigem Involvement als persönlich unbedeutend empfunden, was in einer eingeschränkten Informationsaufnahme resultiert (ebd.).

Im Labor sind die Versuchspersonen per se in der Regel nur wenig in den Gegenstand ihres Handelns involviert, da sie in einer künstlichen Situation etwas fremd Angewiesenes tun, das keine weiteren Konsequenzen für sie hat. Der Bezug zum „Ich“ fehlt. Simuliert werden soll jedoch eine Situation, in der ein Individuum aus einer konkreten, persönlichen Bedürfnislage (unzureichende Softwareausstattung) heraus Handlungen zur Verbesserung der eigenen Situation (Evaluation und Auswahl besser geeigneter Software) unternimmt, und insofern ein erhöhtes Maß an Involvement aufweisen sollte. Die Aufgabenstellung muss also eine dieser Situation möglichst nahe kommende Ich-Beteiligung des Probanden herstellen, um ein annähernd realistisches Verhalten zu bewirken. Hinweise darauf, dass das persönliche Involvement bei der Auswahl von Software eine Rolle spielt, finden sich z. B. bei Kemp et al. (2002; vgl. Abschnitt 3.3).

5.3.3.2 Ableitung der Aufgabenstellung

Die Ausführungen des vorigen Abschnitts können in zwei Anforderungen an die Form und den Inhalt der Aufgabenstellung zusammengefasst werden:

1. Es muss klar werden, was zu tun ist, d. h. worin der Sinn, das Ziel, der situative Kontext des geforderten Handelns besteht. Es soll aber kein bestimmtes Vorgehen impliziert oder gar vorgeschrieben werden (Zielspezifität und Involvement).
2. Die Versuchsperson muss motiviert werden, sich mit einer gewissen Ernsthaftigkeit mit dem gestellten Problem zu beschäftigen und dabei die eigenen Präferenzen zu berücksichtigen, statt vor allem das Künstliche der Situation wahrzunehmen und zu versuchen, den vermeintlichen Erwartungen des Versuchsleiters zu entsprechen (Zielbindung und Involvement).

Um diese Anforderungen zu erfüllen, wurde versucht, einen situativen Kontext zu konstruieren, der der abzubildenden realen Situation möglichst nahe kommt. Diese ist unter anderem dadurch gekennzeichnet, dass das Individuum die aktuelle Softwareausstattung für unzureichend hält und sich deshalb nach Alternativen umsieht. Es ist ihm genau bekannt, welches Problem durch die neue Software gelöst werden soll. Deshalb besteht auf

jeden Fall eine, wenn auch vielleicht noch diffuse Vorstellung davon, was das gesuchte Programm leisten soll, die aber doch ausreicht, um sich an erlebten Optionen in Zustimmung oder Ablehnung zu konkretisieren.

Diese Vorstellung könnte etwa so aussehen: „Ich brauche ein Werkzeug, mit dem ich Flussdiagramme erzeugen kann. Wie das genau aussehen und funktionieren soll, weiß ich nicht und ist mir im Grunde auch egal, solange ich gut damit zurechtkomme. Wenn ich es sehe und ausprobiere, werde ich erkennen, ob es das ist, was ich brauche.“ Daraus folgt, dass es gelingen muss, die Versuchsperson in die Lage eines „Bedürftigen“ mit implizit erlebten statt explizit vorgegebenen Anforderungen zu versetzen, um ein realistisches Maß an Zielspezifität und Involvement zu erreichen (vgl. Abschnitt 5.3.3.1).

Zu diesem Zweck werden die Probanden vor der eigentlichen Auswahlaufgabe gebeten, ein vorgegebenes Flussdiagramm mit Hilfe der Zeichenfunktion des Textverarbeitungsprogramms Microsoft Word zu erzeugen und zu pflegen, d. h. nachträglich zu verändern. Dadurch erleben sie selbst die Situation eines Menschen, dessen Aufgabe es ist, derartige Abbildungen unter Verwendung einer dafür unzureichenden Software zu zeichnen und zu pflegen, statt sich dies nur vorzustellen. Sie erleben ganz unmittelbar, worin die damit verbundenen Probleme bestehen und erhalten dadurch die angestrebte Vorstellung davon, was ein spezialisiertes Programm leisten sollte. Da Microsoft Word als Textverarbeitungsprogramm nur bedingt dafür geeignet¹⁴ ist, Grafiken irgendeiner Art, geschweige denn unter Berücksichtigung semantischer Besonderheiten zu erzeugen und zu pflegen, wird u. U. auch ein gewisser „Leidensdruck“ erzeugt, welcher ein plausibler Aspekt der Situation des realen Evaluators ist und deshalb dazu beiträgt, diese zu simulieren. Nicht zuletzt ergibt sich aus dieser vorbereitenden Betätigung eine gute Gelegenheit, vor der eigentlichen Untersuchung mit dem Lauten Denken vertraut zu werden.

Ein weiteres Charakteristikum der zu simulierenden Situation besteht darin, dass die Auswahl einer neuen Software *für sich selbst* erfolgt, d. h. das Individuum selbst wird auch mit den Konsequenzen der Entscheidung konfrontiert. Diese bestehen vor allem darin, dass diese Software nunmehr auch benutzt wird oder aber als Fehlkauf anzusehen ist, ggf. verbunden damit, die Auswahlentscheidung sowie die mit dem neuen Programm erzielten Ergebnisse vor Kollegen oder Vorgesetzten rechtfertigen zu müssen. Derartigen Konsequenzen entspringt ein bestimmtes Involvement sowie Zielbindung (vgl. Abschnitt 5.3.3.1), welche dafür sorgen, dass die Evaluation und Entscheidung mit entsprechender Ausdauer und Sorgfalt betrieben werden. Im Rahmen der Teilnahme an einer Studie sind die Probanden aber per se weder involviert (und zwar weder persönlich noch über das Produkt oder die Situation) noch liegt eine adäquate Zielbindung vor.

Vor diesem Hintergrund muss die Anleitung der Probanden so erfolgen, dass ihre Aufgabe nicht mit der Auswahlentscheidung erledigt ist. Vielmehr müssen sie glauben, dass das Ergebnis der Auswahl Einfluss auf ihre weitere Performance in der Sitzung hat. Dadurch wird das Auswahlergebnis instrumentell für die möglichst erfolgreiche Teilnahme an der Untersuchung, zu der die Probanden annahmegemäß motiviert sind (vgl. Abschnitt 5.3.3.1). Diese Instrumentalität erhöht die Bindung an das Ziel, die bestgeeignete

¹⁴ So ist z. B. keine Möglichkeit vorgesehen, zusammenhängende abgewinkelte Linien zu erzeugen, so dass diese aufwendig aus einzelnen geraden Fragmenten zusammengesetzt werden müssen.

Software auszuwählen. Außerdem wird die Versuchsperson weiter in den zu bearbeitenden Sachverhalt involviert, wodurch der Aspekt, dem Versuchsleiter durch Teilnahme an seiner Studie einen Gefallen zu tun, zugunsten einer verstärkten Ich-Beteiligung in den Hintergrund tritt.

Den Probanden wird deshalb erklärt, es ginge darum, zu erforschen, wie sie dabei zurechtkommen, mit einer bis dahin unbekanntem Software ein vorgegebenes Flussdiagramm möglichst schnell und optisch ansprechend zu erzeugen. Die zu verwendende Software dürften bzw. sollten sie sich jedoch vorher selbst aussuchen. Dadurch zerfällt die Aufgabe aus Sicht der Versuchsperson in zwei Teile: erstens das Auswählen, zweitens das Verwenden einer Software. Dabei wird davon ausgegangen, dass die Probanden prinzipiell motiviert sind, beim zweiten Teil möglichst gut abzuschneiden, woraus die gewünschte Zielbindung zur Steuerung des Verhaltens im ersten, eigentlich interessierenden Teil entspringen sollte.

Als weiterer Bestandteil der Aufgabenstellung wird ein Dummy-Fragebogen¹⁵ eingeführt, der zur Bewertung eines jeden Produktes ausgefüllt werden soll. Damit werden zwei Ziele verfolgt:

1. in der Wahrnehmung des Probanden den Aspekt verstärken, dass weniger er und sein Verhalten als vielmehr die Software im Mittelpunkt des Interesses steht, und damit Hemmungen verringern („evaluation apprehension“; O. Huber, 2005, S. 194f; vgl. auch Abschnitt 5.3.3.3);
2. aufbauend auf der vorausgesetzten Motivation des Probanden, einen brauchbaren Beitrag zur Forschung zu leisten (vgl. Abschnitt 5.3.3.1) – und dazu gehört auch das korrekte Ausfüllen des Fragebogens – einer nur flüchtigen Evaluation der Alternativen vorbeugen. Dazu könnte es eventuell kommen, weil der Proband der Auswahl-aufgabe nur untergeordnete Wichtigkeit beimisst und sie so schnell wie möglich erledigen will, um den Ablauf nicht unnötig zu bremsen. Schließlich handelt es sich dabei angeblich „nur“ um eine Vorbereitung der „eigentlichen“ Aufgabe.

Beim verwendeten Fragebogen handelt es sich um ein Derivat, d. h. eine modifizierte Version des AttrakDiff-Instrumentes (Hassenzahl et al., 2003, 2000; vgl. Abschnitt 3.3). AttrakDiff misst in Form eines semantischen Differenzials die Attraktivität von interaktiven Produkten, insbesondere auch Software, und erschien insofern inhaltlich für die beabsichtigten Zwecke geeignet. Selbstversuch und Konsultation bei insgesamt neun anderen Personen zeigten jedoch, dass die Items teilweise nicht verstanden bzw. Itempaare nicht als entgegengesetzte Pole einer Dimension angesehen wurden. Deshalb wurden, wieder in Konsultation mit Unbeteiligten, unklare Itempaare eliminiert und solche, die ähnliche Aspekte beschreiben, entsprechend zusammengefasst. Dies war außerdem geboten, da das Original-Instrument 21 Itempaare umfasst, was für die hier vorgesehene Verwendung zu umfangreich erschien.

Folgende Itempaare wurden verwendet: verwirrend – übersichtlich, ermüdend/ lahm – fetzig/fesselnd, fremdartig – vertraut, widerspenstig – kontrollierbar, unattraktiv – attraktiv, konservativ – innovativ, hausbacken – professionell, kompliziert – einfach.

¹⁵ Damit ist ein Fragebogen gemeint, der nicht tatsächlich irgendein Konstrukt messen soll.

5.3.3.3 Wortlaut und Kommentar

Das Abfassen der Arbeitsanweisung erfolgte unter größtmöglicher Beachtung der in der einschlägigen Literatur hinreichend diskutierten Regeln, wie: so kurz wie möglich und so lang wie nötig, Aufklären über Inhalt und Zweck der Untersuchung, kurze Sätze, Vermeiden von Fachsprache usw. (z. B. O. Huber, 2005; Hussy & Jain, 2002). Im Folgenden werden die Formulierungen für die beiden zentralen Aufgaben wiedergegeben und kommentiert. Für sonstige Bestandteile (z. B. Einstimmung und Aufwärmübungen) wird auf Abschnitt 5.3.6 sowie Anhang A verwiesen.

Die Instruktion, die den Probanden für den ersten Teil (Zeichnen mit Microsoft Word; s. Abschnitt 5.3.3.2) vorgelegt wurde, lautete wie folgt. Dabei entsprechen die Hervorhebungen denen im Original. Die Zahlen im Subskript dienen der Identifikation der einzelnen Sätze für die anschließende Kommentierung.

¹Um das Zeichnen von Flussdiagrammen mit einer nicht spezialisierten Software auszuprobieren und sich weiter ans Laute Denken zu gewöhnen, erstellen Sie bitte die unten stehende Abbildung mithilfe der Ihnen bekannten Zeichenfunktionen von Microsoft Word. ²Sie haben dafür **max. 30 min Zeit**. ³Achten Sie darauf, dass Ihre Abbildung **optisch ansprechend** ausfällt.

⁴Versuchen Sie wieder, alles zu erzählen, was Ihnen bei der Arbeit einfällt oder durch den Kopf geht: ⁵Was funktioniert gut, was ist lästig, was begeistert Sie, was treibt Sie in den Wahn usw. ⁶Beobachten und beschreiben Sie dabei auch, **wie Sie mit der Aufgabe an sich zurecht kommen und welche besonderen Unterstützungen** Sie sich bei einer Software, die auf das Erstellen von Flussdiagrammen spezialisiert ist, wünschen würden.

Satz₁ enthält die engere Aufgabenstellung. Satz_{2,3} dient dazu, das zu lösende Problem zu verschärfen, um den Probanden die Unzulänglichkeiten einer nur bedingt geeigneten Software erleben zu lassen. Satz_{4,5} enthält die Anweisung zum Lauten Denken. Der Proband soll ermutigt werden, mit größtmöglicher Selbstverständlichkeit wirklich alles auszusprechen, woran er bei der Aufgabenbearbeitung denkt. Satz₆ steht in Zusammenhang mit der Intention von Satz_{2,3} und soll dazu anregen, aus dem erlebten „Leidensdruck“ eine Vorstellung hinsichtlich dessen, was eine spezialisierte Software besser machen soll, zu entwickeln.

Die finale Arbeitsanweisung, mit der die Probanden nach Erledigung der vorbereiteten Aktivitäten konfrontiert wurden, lautete wie folgt:

⁷Diese Aufgabe besteht aus zwei Teilen. ⁸Wie Sie bereits wissen, sollen Sie mit einem darauf spezialisierten Programm ein Ihnen noch **unbekanntes** Flussdiagramm zeichnen, das etwa den **Umfang** des Beispiels „Kleiderauswahl für den Mann“ hat. ⁹Dabei geht es darum, möglichst **schnell und präzise** zu einem ansprechenden Ergebnis zu kommen.

¹⁰Zunächst aber sollen Sie sich aus verschiedenen Alternativen **in aller Ruhe** ein Programm **aussuchen**, mit dem Sie diese Aufgabe bearbeiten wollen. ¹¹Auf dem Computer sind verschiedene Programme zum Zeichnen von Flussdiagrammen installiert. ¹²Finden Sie heraus, welches dieser Programme Ihnen am besten gefällt. ¹³Es gibt jetzt noch **keine zeitliche Beschränkung**. ¹⁴Gehen Sie also sorgfältig und gründlich vor. ¹⁵Wenn Sie sich nicht entscheiden können, berücksichtigen Sie ruhig auch, welche zusätzlichen nützlichen Funktionalitäten die Produkte anzubieten haben. ¹⁶Wer weiß, wofür Sie das nachher brauchen können . . .

¹⁷Stellen Sie aber auf jeden Fall sicher, dass Sie **alle benötigten Funktionen** abrufen können und gut mit der **Bedienlogik** zurechtkommen.

¹⁸Sie werden gebeten werden, Ihre Entscheidung zu **begründen** bzw. zu erklären, warum Sie die anderen Kandidaten weniger gut finden. ¹⁹Außerdem sollen Sie für jedes der Programme das umseitige **Bewertungsprofil** ausfüllen. ²⁰**Wie** Sie jedoch bei Ihrer Untersuchung der Alternativen im Einzelnen vorgehen, **ist völlig Ihnen selbst überlassen**; ebenso, ob Sie sich nacheinander oder parallel mit den einzelnen Programmen beschäftigen. ²¹Wenn Sie möchten, dürfen Sie dabei auch Notizen machen. ²²Spielen Sie ungehemmt mit der Software herum; Sie können nichts kaputt und keine Fehler machen: ²³Auf dem Prüfstand steht hier die Software, nicht Sie!

²⁴Der Versuchsleiter soll jetzt nicht mehr in Ihre Handlungen eingreifen, sondern passiver Beobachter sein. ²⁵Vergessen Sie aber bitte nicht, wieder die ganze Zeit **laut mitzudenken**. ²⁶Lassen Sie den Versuchsleiter an Ihren Befindlichkeiten teilhaben und sprechen Sie laut aus, was Ihnen gefällt, was Ihnen missfällt, was Sie überrascht usw.

Satz₇₋₉ soll den Probanden weiter in die durch die Software zu lösende Problematik und damit in die Auswahl eines Programms involvieren und qua Herstellen einer Instrumentalität des Auswahlresultates Zielbindung erzeugen. Dass das angeblich zu erzeugende Diagramm noch unbekannt ist, entspricht der zu simulierenden Realweltsituation, in der auch noch nicht alle zu bearbeitenden Probleme bereits genau bekannt sind. Das Beispiel „Kleiderauswahl für den Mann“ ist den Probanden aus den ihnen zur Vorbereitung übergebenen Materialien vertraut (vgl. Abschnitt 5.3.4.2). Satz₉ steht in besonderem Zusammenhang zu der vorbereitenden Übung mit Microsoft Word, in der gerade die Punkte Schnelligkeit und Präzision problematisch gewesen sein sollten. Es wird erwartet, dass der Proband unter diesem Eindruck in erhöhtem Maße bestrebt ist, das für ihn beste Programm zu finden.

Satz₁₀₋₁₂ enthält die eigentliche Aufgabenstellung. Ein besonderer Zweck wird lediglich mit dem Hinweis „in aller Ruhe“ verfolgt. Er soll im Zusammenwirken mit Satz_{13,14} deutlich machen, dass, obwohl die Auswahl angeblich nur Mittel zum eigentlichen Zweck

der Studie ist, diese aber trotzdem nicht flüchtig erfolgen soll, weil davon die weitere Performance abhängt. Satz_{15,16} ermutigt dazu, schreibt aber nicht vor, bei Bedarf auch über das unbedingt Notwendige hinauszugehen. Der Proband soll nicht das Gefühl haben, dass er damit den Fortschritt der Sitzung aufhalten würde. Satz₁₆ unterstreicht darüber hinaus die Ungewissheit über die angeblich folgende Aufgabe und soll dadurch die Instrumentalität des Auswahlresultates für weitere Folgen verstärken.

Satz₁₇ ist besonders hervorgehoben und steht im Zusammenhang mit der Spezifität des zu erreichenden Ziels, das im Wesentlichen aus den beiden genannten Punkten besteht: Es ist herauszufinden, wie ein Programm funktioniert, und einzuschätzen, ob man glaubt, gut damit arbeiten zu können. Die starke Hervorhebung entspringt den Erfahrungen der Pilotphase und soll die zentrale Bedeutung dieses Satzes für das zu erreichende Ziel unterstreichen.

Satz₁₈ nimmt Bezug auf den in der Realität, zumindest in einem betrieblichen Umfeld, nicht unwahrscheinlichen Rechtfertigungszwang, wofür das zur Verfügung gestellte Geld ausgegeben wurde oder werden soll (vgl. Abschnitt 5.3.3.2). Satz₁₉ verweist auf den Dummy-Fragebogen und soll ebenfalls zur Sorgfalt ermahnen bzw. einer unrealistisch flüchtigen Begutachtung vorbeugen.

Satz_{20,21} gibt technische Hinweise und versichert dem Probanden, dass es nicht *das eine* richtige Vorgehen bei der Auswahl gibt, und soll dazu beitragen, dass er sich möglichst unbefangen verhält. Satz_{22,23} soll diesen Aspekt weiter verstärken. Damit soll vermieden werden, dass der Proband die Labor- als Prüfungssituation wahrnimmt und Angst hat, seine Gedanken offenzulegen oder sich vor dem Versuchsleiter durch seine Ungeschicklichkeit zu blamieren („evaluation apprehension“; O. Huber, 2005, S. 194f; vgl. auch Deffner, 1984; Someren et al., 1994). Durch Satz₂₃ werden eventuell auftretende Probleme deshalb vorbeugend auf die Software attribuiert¹⁶.

Satz₂₄ schließt den Versuchsleiter als Hilfequelle bei Problemen aus und soll deutlich machen, dass z. B. die Entscheidung über einen eventuellen Abbruch der Begutachtung eines Produktes allein beim Probanden liegt. Satz_{25,26} erinnert noch einmal an das laute Mitdenken.

5.3.4 Versuchspersonen

5.3.4.1 Auswahl und Rekrutierung

An der Hauptstudie nahmen zwanzig Personen teil, darunter zwölf Männer und acht Frauen. Deren Alter lag zwischen 23 und 38 Jahren, bei einem Mittelwert von 27,9 Jahren und einem Median von 27,5 Jahren. Alle Teilnehmer hatten einen akademischen Hintergrund, d. h. sie befanden sich zum Zeitpunkt ihrer Teilnahme in einer Hochschulausbildung (10) oder hatten diese bereits abgeschlossen (10). Die Studenten waren in Betriebswirtschaftslehre (4), Business Informatics (3), Wirtschaftsinformatik (2) sowie Sonderpädagogik (1) eingeschrieben. Unter den Absolventen befanden sich sieben Diplom-Kaufleute, zwei

¹⁶ Kausalattribution beschreibt den Schlussfolgerungsprozess, durch den Beobachter einen Effekt auf eine oder mehrere Ursachen zurückführen (z. B. Fincham & Hewstone, 2003).

Diplom-Wirtschaftsinformatiker sowie ein Diplom-Volkswirt. Sie waren zum Zeitpunkt ihrer Teilnahme als wissenschaftliche Mitarbeiter der Universität (6), in der Softwareentwicklung (2), der Unternehmensberatung (1) sowie dem Personalwesen (1) tätig.

Die Rekrutierung erfolgte über persönliche Ansprache durch den Versuchsleiter sowie durch eine Ausschreibung auf den WWW-Seiten der Wirtschafts- und Sozialwissenschaftlichen Fakultät der Universität Rostock. Extrinsische Anreize für die Teilnahme, wie z. B. finanzielle oder sonstige Kompensation, wurden nicht geboten. Die potenziellen Teilnehmer wurden bereits bei der Ansprache/Ausschreibung in groben Zügen darüber informiert, was sie zu tun hätten.

Auf diese Weise wurden bereits in dieser Phase Personen ausgeschlossen, die bei der Arbeit mit dem Computer unsicher sind und sich die selbstständige Begutachtung unbekannter Software nicht zutrauen. Diese Einschränkung war nötig, um eine Verzerrung der Ergebnisse zu vermeiden (O. Huber, 2005). Nur Individuen, die auch im Alltag selbstständig neue Software ausprobieren (würden), können im Labor Aufschlüsse zu den hier interessierenden Fragestellungen liefern. Ergänzend wurden die Kandidaten zu ihrer Einstellung zum Umgang mit dem Computer befragt. Der dazu verwendete Fragebogen (s. Anhang A) enthielt Items wie „Ich benutze den Computer sehr gern, weil es mir Spaß macht“, „Ich habe keine besondere Zu- oder Abneigung gegenüber dem Computer. Er ist ein Werkzeug, und wenn es nötig ist, benutze ich ihn eben“, „Ich benutze Computer ungern, weil ich Angst habe, Fehler zu machen“.¹⁷ Keiner der Probanden ließ erkennen, sich im Umgang mit dem Computer unsicher zu fühlen.

Ein Nebeneffekt der Einschränkung auf Personen, die sich die Bearbeitung der zu erwartenden Aufgaben zutrauen, liegt darin, dass dieses Vertrauen in die eigene Tüchtigkeit (*Selbstwirksamkeit*; Bandura, 1982) ein weiterer Bestimmungsfaktor von Zielbindung ist (Kleinbeck, 1996, 2006; s. Abschnitt 5.3.3.1). Über diesen Zusammenhang wird ein weiterer Beitrag zur Erzielung realitätsnahen Verhaltens im Labor bzw. zur Validität der Ergebnisse geleistet.

Bei der Auswahl der Teilnehmer wurde außerdem darauf geachtet, dass sie in unterschiedlichem Maß Erfahrung¹⁸ im Hinblick auf die Computerbenutzung im Allgemeinen und die Verwendung von Grafikprogrammen im Besonderen aufweisen. Dadurch sollte zum einen die Gültigkeit der Ergebnisse nicht noch mehr, als es durch die kleine Stichprobe ohnehin der Fall ist, eingeschränkt sowie ggf. eine differenzierte Hypothesenbildung ermöglicht werden.

Zur Einschätzung ihres fachlichen Hintergrundes füllten die Probanden einen weiteren Fragebogen aus, in dem nach ihren Erfahrungen im Umgang mit dem Computer und Software allgemein sowie mit der Erstellung von Grafiken gefragt wurde. Die Antworten auf

¹⁷ Der Fragebogen sollte nicht dafür eingesetzt werden, ein klar definiertes Konstrukt zu messen, sondern diene lediglich dazu, dem Versuchsleiter einen Eindruck vom Verhältnis des potenziellen Probanden zum Umgang mit dem Computer zu vermitteln. Deshalb wurde der Aufwand für eine Validierung des Fragebogens als unangemessen erachtet.

¹⁸ Hier ist Erfahrung von der zuvor diskutierten Unsicherheit zu unterscheiden. Ein Individuum, das nur in geringer Breite oder Tiefe mit dem Computer arbeitet, hat deshalb nicht notwendigerweise auch Berührungängste.

Fragen nach der Erfahrung im Umgang mit dem Computer und Software allgemein sollten Grenzfälle entscheiden bzw. feinere Unterteilungen ermöglichen, wurden aber nicht weiter verwendet. Zu deren Gestaltung wird deshalb an dieser Stelle auf Anhang A verwiesen. Zur Einordnung in eine der Erfahrungsklassen „gering“, „unspezifisch“ bzw. „spezifisch“ wurden lediglich die Antworten auf folgende Fragen herangezogen:

Haben Sie zuvor schon einmal Flussdiagramme gezeichnet?

nein ja, aber nur fremde abgezeichnet ja, auch selbst entworfen

Falls ja, welche Software haben Sie dafür verwendet? (*offene Frage*)

Welche Ihnen bekannte Software würden Sie heute am ehesten zum Zeichnen eines Flussdiagramms verwenden? (*offene Frage*)

In die Gruppe mit *geringer* Erfahrung wurden Personen eingeordnet, die weder jemals selbst Flussdiagramme gezeichnet hatten noch angeben konnten, welche Software sie heute dafür verwenden würden. Als *unspezifisch* erfahren wurden Personen eingestuft, die zwar schon selbst Flussdiagramme gezeichnet haben, dafür aber keine spezialisierte Software benutzt hatten und heute ein allgemeines Werkzeug wie Microsoft PowerPoint wählen würden. *Spezifische* Erfahrung wurde Personen zugeschrieben, die bereits früher einschlägige Grafiksoftware benutzt hatten oder heute benutzen würden. Aus dieser Einteilung gingen vier gering, acht unspezifisch und acht spezifisch erfahrene Probanden hervor.

5.3.4.2 Vorbereitung

Bei der Rekrutierung wurde den Probanden ein Faltblatt mit Informationen über den Zweck, die Bestandteile und den Aufbau von Flussdiagrammen übergeben (s. Anhang A). Die verbalen Erläuterungen waren äußerst knapp gehalten, um die Teilnehmer nicht durch eine große Menge Text davon abzuhalten, das Material wirklich zu lesen. Illustriert wurden die Erläuterungen durch drei Beispiele für Flussdiagramme unterschiedlichen Umfangs.

Das erste entspricht dem in Abb. 5.1 auf S. 72 und hatte den Zweck, absoluten Neulingen das Grundprinzip zu verdeutlichen, das zweite (12 Symbole) diente später in der Sitzung als Vorlage für die Übung „Zeichnen mit Word“ (vgl. Abschnitt 5.3.3.2), das dritte (20 Symbole) war seitenfüllend und wurde in der finalen Aufgabenstellung als Referenz für den Umfang eines angeblich selbst zu erstellenden Diagramms verwendet („Kleiderauswahl für den Mann“; vgl. Abschnitt 5.3.3.2 und 5.3.3.3). Vor allem jedoch sollte es durch seinen humorvollen Inhalt dazu anregen, einmal ein größeres Diagramm vollständig zu studieren und sich dabei der Einfachheit der Notation bewusst zu werden. Dies erschien wichtig, um eventuell vorhandene Hemmungen oder Unsicherheiten hinsichtlich der Anwendungsdomäne abzubauen. Schließlich sollte eine Situation simuliert werden, in der sich das Individuum in ebendieser Domäne grundsätzlich zurechtfindet. Inhaltliche Befindlichkeiten würden einen weiteren zu berücksichtigenden Aspekt einbringen und die

Auswertung dadurch weiter komplizieren. Auch deshalb erschien es geboten, die Probanden mit Flussdiagrammen an sich so weit wie möglich auf vertrauten Boden zu stellen.

Außerdem erhielten die Versuchspersonen ein Faltblatt, das den Inhalt der Studie zusammenfassen und vor allem auf die Erhebungsmethode, das Laute Denken, vorbereiten sollte (s. Anhang A). Weitere Bestandteile dieses Faltblattes waren die im vorigen Abschnitt beschriebene Erhebung des fachlichen Hintergrundes sowie eine Präferenzabfrage hinsichtlich der Wichtigkeit verschiedener Aspekte von Anwendungssoftware. Die vorgegebenen Aspekte entsprachen den Dimensionen, die auch für den in der Sitzung auszufüllenden Dummy-Fragebogen verwendet wurden (vgl. Abschnitt 5.3.3.2). Die Probanden hatten die Aufgabe, je nach Wichtigkeit insgesamt zwölf Punkte auf diese acht Dimensionen so zu verteilen, dass die persönliche Präferenzstruktur abgebildet wird. Da eine Gleichgewichtung aller Dimensionen unter dieser Vorgabe nicht möglich ist, waren die Probanden gezwungen, sich wenigstens etwas mehr als nur oberflächlich mit deren Bedeutungen auseinanderzusetzen, falls sie die Punkte nicht völlig willkürlich verteilen wollten.

Der Sinn dieser Übung bestand darin, die Versuchspersonen durch die semantische und gewichtende Beschäftigung mit Aspekten, die Relevanz für die Attraktivität von Software haben, in die sie erwartenden Aufgaben zu involvieren. Schließlich gilt es, sich in die Lage eines Individuums zu versetzen, dessen Bedürfnis, einer mangelhaften Softwareausstattung abzuweichen, nicht plötzlich entstanden, sondern über einen gewissen Zeitraum gewachsen ist. Insofern erscheint es sinnvoll, die Probanden zu veranlassen, sich bereits im Vorfeld ihrer eigenen Präferenzen bewusst zu werden.

Darüber hinaus sollten die Versuchspersonen mit den im Dummy-Fragebogen verwendeten Dimensionen und deren Semantik vertraut gemacht werden, damit das Ausfüllen im Verlauf der Sitzung zügig vonstatten geht. Es wäre unerwünscht, wenn ein Proband erst dann anfinge, über semantische Fragen nachzudenken und dadurch der Fragebogen eventuell zum dominierenden Problem würde. Angesichts seiner Aufgabe als Mittel zum Zweck (vgl. Abschnitt 5.3.3.2) sollte dessen Bearbeitung aber nicht in den Vordergrund rücken oder als kompliziert wahrgenommen werden.

5.3.5 Datenerfassung

Die zu erfassenden Daten bestanden in den Interaktionen der Probanden mit der Software bei deren Begutachtung, den damit verbundenen Kognitionen sowie dem Ergebnis der Auswahl einschließlich ex-post vorgebrachter Gründe für die Entscheidung. Die Interaktionen und das Ergebnis sind unmittelbar beobachtbar, der Zugang zu den begleitenden Kognitionen erfolgte mit der Methode des Lauten Denkens (vgl. Abschnitt 5.2.2). Zur Aufzeichnung dieser Daten wurde die Software *Camtasia Recorder* von TechSmith¹⁹ verwendet, welche aus einem früheren Projekt am Lehrstuhl für Wirtschaftsinformatik an der Universität Rostock zur Verfügung stand. Diese Software läuft im Hintergrund und zeichnet das Geschehen auf dem Monitor als Videofilm auf, der anschließend als Datei im

¹⁹ www.techsmith.de

AVI-Format vorliegt. Parallel wird über ein externes Mikrofon eine Audioaufzeichnung ermöglicht, die in den Videofilm eingebunden wird.

Auf diese Weise wird der Proband nicht durch eine Videokamera oder Ähnliches abgelenkt oder beeinflusst. Dass dies ein wichtiger Aspekt ist, zeigt der Umstand, dass sich die meisten Probanden vor ihrer Zustimmung zur Teilnahme versichern ließen, dass keine Videoaufzeichnung erfolgen würde. Bei Verwendung der Camtasia-Software wird lediglich durch die Anwesenheit eines miniaturisierten Ansteckmikrofons offensichtlich, dass überhaupt irgendetwas aufgezeichnet wird. Ein weiterer Vorteil gegenüber der Verwendung einer Videokamera besteht darin, dass das Bild gestochen scharf ist, weil es in Originalauflösung direkt aus der Grafikausgabe des Computers erzeugt wird. Darüber hinaus können die Position des Mauszeigers sowie Mausklicks farblich hervorgehoben werden, sodass die erfolgten Interaktionen sehr gut nachvollziehbar sind.

5.3.6 Geplanter Ablauf einer Sitzung

Zu Beginn der Sitzung wird die Versuchsperson in ein lockeres Gespräch verwickelt, um sie zu entspannen und ihr Gelegenheit zu geben, sich mit der Umgebung vertraut zu machen (Adaptionsphase; O. Huber, 2005). Außerdem soll der Versuchsleiter einen Eindruck vom Naturell des Probanden, insbesondere von seiner bevorzugten Umgangssprache gewinnen, um den Ablauf möglichst effektiv steuern zu können. Als Einstieg in dieses Gespräch bietet sich das an die Versuchsperson ausgegebene und von ihr bearbeitete Material an (vgl. Abschnitt 5.3.4.2). Dessen Diskussion trägt außerdem dazu bei, die zu erhebenden Informationen über den individuellen fachlichen Hintergrund zu vertiefen oder zu konkretisieren sowie letzte inhaltliche bzw. semantische Unklarheiten, die den reibungslosen Ablauf der Sitzung stören könnten, auszuräumen.

Bereits in dieser frühen Phase der Sitzung wird die Versuchsperson mit dem Mikrofon verkabelt. Würde dies erst kurz vor der eigentlichen Erhebung erfolgen, würde der bis dahin etablierte Aktivitäts- und vor allem Redefluss unterbrochen und der Proband unmissverständlich darauf aufmerksam gemacht, dass es ab jetzt „ernst wird“. Das dadurch aufgefrischte Gefühl, beobachtet zu werden, könnte über die in jedem Fall störende Unterbrechung hinaus zu reaktiven Effekten führen, die den weiteren Verlauf verzerren und damit die Validität der Beobachtung beeinträchtigen (Bortz & Döring, 2006). Zu erwarten wäre je nach Naturell des Probanden insbesondere gehemmtes oder übertriebenes Verhalten. Die frühe Erledigung der Verkabelung birgt dagegen die Chance, dass die Versuchsperson vergisst, dass sie ein Mikrofon trägt, sodass aus diesem Umstand resultierende Effekte gemindert werden.

Im Zuge der Unterhaltung wird nochmals der Zweck der Untersuchung umrissen, dabei aber der Umstand, dass es vordergründig um die Auswahl geht, verschleiert. Vielmehr soll allgemein vom Umgehen mit unbekannter Software die Rede sein. In diesem Zusammenhang wird auf die Erhebungsmethode, das Laute Denken, hingeleitet und nochmals erläutert, was darunter zu verstehen ist. Besonders zu betonen ist dabei, dass hemmungslos wirklich alles ausgesprochen werden soll, egal wie unpassend es erscheint.

In der Pilotstudie hatte sich gezeigt, dass die Versuchspersonen trotz aller Erläuterungen keine rechte Vorstellung davon haben, was diesbezüglich von ihnen erwartet wird. Deshalb wird ihnen zur Illustration ein Video gezeigt, in dem laut denkend eine Software zur Bearbeitung eines zur vorliegenden Studie sachfremden Problems benutzt wird. Alternativ wäre eine direkte Demonstration durch den Versuchsleiter möglich, jedoch wäre dies mit dem Nachteil behaftet, dass eine solche Vorführung von dessen Tagesform abhängig wäre (O. Huber, 2005). Über die Konstanz der Instruktion hinaus hat ein Video den Vorteil, dass zu seiner Erstellung beliebig viele Versuche unternommen werden können, bis ein zufrieden stellendes Ergebnis vorliegt.

Anschließend erhält die Versuchsperson Gelegenheit, selbst Lautes Denken zu üben, indem ihr folgende, aus Someren et al. (1994, S. 4) entlehnte Aufgabe zur Lösung übergeben wird:

Vater, Mutter und Kind sind zusammen 80 Jahre alt. Der Vater ist doppelt so alt wie das Kind, und die Mutter ist genauso alt wie der Vater. Wie alt ist das Kind?

Um die Versuchsperson zu enthemmen, wird ihr erklärt, dass es viele verschiedene Lösungswege gibt und bisher noch keiner im ersten Anlauf die richtige Lösung gefunden hat. Bei Bedarf wird immer wieder an das Laute Denken erinnert.

Im nächsten Schritt soll sich die Versuchsperson an Lautes Denken parallel zu einer manuellen Verrichtung am Computer gewöhnen. Zu diesem Zweck erhält sie die Aufgabe, ein vorgegebenes Flussdiagramm mit Hilfe der Zeichenfunktion von Microsoft Word zu erzeugen (vgl. Abschnitt 5.3.3.2). Aus den Erfahrungen der Pilotphase unterstützt der Versuchsleiter das Laute Denken, indem er sich der Versuchsperson über das Geschehen auf dem Computermonitor unterhält. Dadurch wird eine gegenüber dem Monolog als natürlicher empfundene Gesprächssituation geschaffen, welche sich als sehr gut geeignet erwiesen hat, den Probanden zum Sprechen zu bringen. Gegen Ende der Aufgabe muss sich der Versuchsleiter jedoch allmählich aus dem Dialog zurückziehen, da er bei der Bearbeitung der folgenden, finalen Aufgabenstellung möglichst passiv sein soll.

Nach Fertigstellung der Zeichnung bzw. nach Ablauf der vorgegebenen Zeit erhält die Versuchsperson die Aufgabe, sich aus drei Alternativen eine Software auszusuchen, angeblich um damit anschließend unter Zeitdruck nach einer Vorlage ein optisch ansprechendes Flussdiagramm zu erzeugen (vgl. Abschnitt 5.3.3.3).

Der Versuchsleiter muss jetzt möglichst passiv bleiben und sitzt schräg hinter dem Probanden außerhalb dessen unmittelbaren Sichtfeldes (Frommann, 2002). Dadurch wird Letzterem signalisiert, dass er ab jetzt auf sich allein gestellt ist. Trotzdem bleibt der Versuchsleiter präsent und vermeidet dadurch, dass der Proband das Gefühl hat, ins Nichts zu sprechen und seinen Verbalisierungsstrom versiegen lässt. Ein weiterer Grund für die Sitzposition liegt in der Vermeidung von Versuchsleiter-Erwartungseffekten (O. Huber, 2005). Unkontrollierte Mimik und Gestik in Reaktion auf das Verhalten des Probanden können von diesem nicht so leicht wahrgenommen werden und ihn somit weniger beeinflussen. Zudem hat die Pilotphase gezeigt, dass es dem Versuchsleiter an seiner Position im Rücken des Probanden leichter fällt, sich als passiver Beobachter zu verhalten. Dies gilt insbesondere, wenn es darum geht, den Probanden „einfach machen zu lassen“, d. h.

Geduld zu bewahren und nicht verfrüht in einen stockenden Ablauf einzugreifen (vgl. Deffner, 1984).

Notwendig werdende Erinnerungen an das Laute Denken und Antworten auf eventuelle Fragen der Versuchspersonen müssen non-suggestiv bzw. „nicht-direktiv“ (Deffner, 1984, S. 14) sein. Dieser Punkt wird in der Literatur immer wieder als besonders wichtig herausgestellt, da anderenfalls die Gefahr besteht, die Kognitionen der Versuchsperson zu beeinflussen (vgl. z. B. Someren et al., 1994). Nachfragen wie: „Warum hast du jetzt nicht auf *xy* geklickt?“ sind deshalb unbedingt zu vermeiden. Andererseits hat die Pilotphase gezeigt, dass Erinnerungen an das Laute Denken über eine bloße Aufforderung zum Weitersprechen hinausgehen müssen, da diese meist dazu führen, dass der Proband, der u. U. gerade der Meinung war, nichts zu sagen zu haben, einfach nur verbalisiert, was er gerade tut. Dies ist aber ohnehin bereits durch die Videoaufzeichnung zu verfolgen.

Vielmehr sollte versucht werden, einen offenen Bezug zur Handlung herzustellen, z. B.: „Was hast du gerade vor?“, „Warum hast du *das* gemacht?“ oder „Wie fandest du *das*?“ Dabei bleibt es dem Probanden überlassen zu entscheiden, was denn *das* wohl wäre. Eine eventuelle Rückfrage, was der Versuchsleiter meine oder ob Handlung *xy* gemeint wäre, kann pauschal mit „Was Du gerade getan hast“ oder „Hmhm“ beantwortet werden. Dadurch wird wieder eine Art Dialog etabliert, statt den ohnehin gehemmten Probanden durch eine bloße Sprechaufforderung erneut auf seine ungewohnte Rolle als Kommentator der eigenen Handlungen hinzuweisen.

Sobald die Versuchsperson die Begutachtung eines Programms beendet hat, wird ihr der Dummy-Fragebogen vorgelegt. Auch während dessen Bearbeitung soll das Laute Denken fortgesetzt werden. Dies könnte zum einen weitere interessante Kognitionen liefern und soll zum anderen eine längere Phase des Schweigens vermeiden, nach der es u. U. schwer fällt, wieder in den Redefluss zu finden.

Nachdem der Proband seine Entscheidung getroffen hat, wird er gebeten, diese noch einmal zu begründen. Anschließend wird ihm erklärt, dass das bisher gesammelte Material ausreiche und deshalb auf die angekündigte letzte Aufgabe verzichtet werden kann.

5.3.7 Durchführung

Die Durchführung der Sitzungen der Hauptphase erfolgte kontinuierlich über einen Zeitraum von mehreren Monaten im Büro des Versuchsleiters. Obwohl der geplante Versuchsablauf in der Regel eingehalten werden konnte, kam es zu den im Folgenden beschriebenen Abweichungen. Die Notation „VpNummer“ dient dazu, auf bestimmte Versuchspersonen Bezug zu nehmen.

Durch Fehlkonfiguration der Aufzeichnungssoftware liegt für Vp19 nur der Videostrom ohne Audioaufzeichnung vor, d. h. die zugehörigen Kommentare wurden nicht gespeichert. Dies hat Auswirkungen vor allem für den zweiten Teil der qualitativen Analyse (Abschnitt 6.3), in der die Daten dieses Probanden nicht verwendet werden konnten. Die quantitative Auswertung sowie der erste Teil der qualitativen Analyse konnten jedoch ohne Einschränkung durchgeführt werden.

Vp1, Vp2 und Vp5 evaluierten jeweils nur zwei statt drei Programme. Der Grund liegt bei Vp1 und Vp2 darin, dass ihre Sitzungen in den fließenden Übergang von der Pilot- zur Hauptphase einzuordnen sind. Das heißt, sie haben bereits den beschriebenen endgültigen Verlauf einschließlich der finalen Formulierung der Aufgabenstellung durchlaufen. Die Beschränkung auf zwei Programme rührt daher, dass Flowcharter zu diesem Zeitpunkt noch nicht verfügbar war. Die Entscheidung, mit Ausnahme von Diagram Studio nur deutschsprachige Programme zu verwenden (vgl. Abschnitt 5.3.2.3), war aber bereits gefallen und sollte eingehalten werden. Vp5 offenbarte erst im Verlauf der Sitzung, dass sie mit Visio als regelmäßig genutztem Werkzeug bereits vertraut sei. Dies war insofern ein Problem, als Vp5 nur kurzfristig zur Verfügung stand, als am Computer, der für die Studie genutzt wurde, Wartungsarbeiten durchgeführt wurden. Deshalb war Diagram Studio nicht installiert. Dies nachzuholen erschien als zu schwerwiegende Unterbrechung des Arbeitsflusses, sodass auf die Begutachtung eines dritten Programms verzichtet wurde.

Diese drei Probanden wurden trotzdem in die Auswertung einbezogen, weil die genaue Anzahl der zu evaluierenden Programme im Hinblick auf die Beantwortung der Fragestellung dieser Studie inhaltlich keine kritische Variable darstellt (vgl. Abschnitt 5.3.2.2). Auch bei nur zwei Alternativen finden die interessierenden Evaluations- und Auswahlprozesse statt, wenn auch in tendenziell geringerem Umfang. Gerade Vp1 und Vp2 haben jedoch vergleichsweise umfangreiches, und Vp5 sehr deutliches und inhaltlich reichhaltiges Material geliefert, sodass nichts gegen die Verwendung der erhobenen Daten spricht. Die abweichende Anzahl evaluierter Optionen ist insofern lediglich bei der Ermittlung einiger Kennzahlen entsprechend zu berücksichtigen.

Im Verlauf der Studie wurde der Modus der Vermittlung der Aufgabenstellung verändert. Nachdem im Zuge der Pilotphase die Formulierung zur finalen Version gereift war, wurde sie den Probanden zum Lesen übergeben. Um flüchtiges Überfliegen zu vermeiden (O. Huber, 2005), sollte laut vorgelesen werden. Dies schien jedoch einigen Versuchspersonen unangenehm zu sein und dadurch die bis dahin gelockerte Situation wieder zu spannen. Anscheinend wurde dies als Signal gewertet, dass „es jetzt ernst wird“. Deshalb wurde ab Vp5 dazu übergegangen, die Probanden mündlich zu instruieren. Dabei hielt sich der Versuchsleiter eng an die geschriebene Version, ohne diese jedoch wörtlich vorzulesen.

Dadurch eventuell auftretende Unterschiede in der Rezeption der Aufgabenstellung wurden als unwesentlich in Kauf genommen, zumal „eine Festlegung der äußeren Form noch längst keine Gewähr für eine in allen Fällen gleiche Vermittlung von Anweisungen, Erwartungen, Information über das Experiment usw. [ist]“ (Deffner, 1984, S. 65; vgl. Friedman, 1967). Zudem handelt es sich hier auch gar nicht um ein Experiment, bei dem die möglichst identische Instruktion von kritischer Bedeutung ist (O. Huber, 2005). Die Instruktion nimmt hier nicht die Rolle einer unabhängigen Variablen an. Es geht vielmehr darum, den Probanden in eine bestimmte Situation zu versetzen, um sein Verhalten darin zu explorieren. Leichte, in Abhängigkeit vom Rezipienten vorgenommene Abweichungen in der Formulierung könnten deshalb die Effektivität der Aufgabenstellung eventuell sogar verbessern.

Zur Vermeidung von Reihenfolgeeffekten wurde die relative Anordnung der Icons zum Starten der Programme zufällig variiert. Grundsätzlich wurden diese untereinander in der

rechten oberen Bildschirmcke platziert. Tabelle 5.1 zeigt, wie oft jedes Programm an erster (oben), zweiter (Mitte) bzw. dritter (unten) Stelle positioniert war. Die abweichenden Zeilen- und Spaltensummen kommen dadurch zustande, dass drei Probanden nur zwei Programme zur Auswahl hatten (zweimal fehlte Flowcharter, einmal Visio). Ergänzt wurde die Anordnung jeweils durch ein nach unten deutlich abgesetztes Icon zum Starten von Microsoft Word.

Position	SmartDraw	Flowcharter	Visio/DS
oben	10	8	2
Mitte	3	10	7
unten	7	0	10

Tabelle 5.1: Häufigkeiten der Anordnung der Programm-Icons auf dem Windows-Desktop

5.4 Auswertung

5.4.1 Beschreibung der Daten

Die auszuwertenden Daten bestehen aus den AVI-Videos, die während der Sitzungen mithilfe der Software Camtasia Recorder aufgezeichnet wurden (vgl. Abschnitt 5.3.5). Diese Videos können prinzipiell mit jedem Programm oder Gerät, das zur Wiedergabe von AVI-Dateien fähig ist, abgespielt werden, sofern der von Camtasia verwendete Codec der Firma TechSmith verfügbar ist.

Der visuelle Teil der Videos besteht aus der Bildschirmausgabe während der Sitzungen, sodass die erfolgten Interaktionen beliebig oft „eins zu eins“ nachverfolgt werden können. Außerdem enthalten die Videos die parallel zu den Interaktionen ausgesprochenen Verbalisierungen der Probanden. Zum Zweck des bequemeren und übersichtlicheren Zugriffs sowie Verweisens auf die Verbalisierungen wurden darüber hinaus Transkripte angefertigt. Dabei, und bei deren Verwendung in der vorliegenden Arbeit, wurden folgende Konventionen eingehalten:

- Verbalisierungen der Versuchsperson erscheinen in normaler Schrift, z. B.: „Dann fang ich einfach mal hier mit dem obersten an.“
- Verbalisierungen, die durch Ablesen von Bildschirminhalten (z. B. Menüeinträge oder Hilfetexte) oder Mitsprechen selbst geschriebener Wörter zustande kommen, werden in den Transkripten in normale „Gänsefüßchen“ eingefasst. Beim Zitieren werden zwecks besserer Unterscheidung von der Einfassung des Gesamtzitats stattdessen «französische Anführungszeichen» verwendet, z. B.: „Geh ich mal auf «Drehen» und dreh es um.“

- Beschreibungen des Verhaltens der Versuchsperson bzw. der Software werden kursiv gesetzt, z. B.: „*startet Flowcharter* → *Begrüßungsbildschirm erscheint*“. Der Rechtspfeil (→) kennzeichnet aus einer Handlung unmittelbar folgende Effekte.
- (Verhaltens)beschreibungen, die zur näheren Erläuterung unmittelbar in den Verbalisierungsfluss eingebettet sind, werden kursiv gesetzt und in kursive eckige Klammern eingefasst, z. B.: „Und von da oben kann man da wohl nicht dran [*an den Kreis*], wieso kann ich da nicht ...“, oder: „Das ist ja faszinierend [*ironisch*]“.
- Bereits im Transkript vorgenommene Auslassungen, z. B. wenn der Proband einen ausführlichen Hilfetext laut liest, erscheinen kursiv, z. B.: „Im Tutorial lernen Sie [...]. Na gut, dann machen wir das mal“.
- Auslassungen und Hinzufügungen, die erst beim Zitieren von Transkriptpassagen in der vorliegenden Arbeit vorgenommen werden, erfolgen gemäß den üblichen Regeln, z. B.: „Das [...] ist hier [SmartDraw] alles total schlecht gemacht.“
- Unverständliche Verbalisierungen werden durch „[?]“ gekennzeichnet, z. B.: „Das ist ja [?], das ist nicht schlecht“.

Die Transkripte zu den Aufzeichnungen sind auf Anfrage vom Verfasser erhältlich.

5.4.2 Aufbereitung des Materials für die Analyse

5.4.2.1 Funktionalitätskategorien

Der Benutzer erlebt eine Software, indem er mit ihr interagiert, also ihre Funktionalität abrufen mit dem Zweck, bestimmte Effekte oder Ergebnisse zu erzielen, bestimmte Wirkungen hervorzurufen. Dabei ist es normalerweise nicht der Fall, dass die Gesamtfunktionalität einer Software nicht weiter zerlegbar ist. Ein extremes und in seiner Funktionalität nicht besonders sinnvolles Beispiel dafür wäre ein Programm, das genau eine Schaltfläche anbietet und bei deren Anklicken über den PC-Lautsprecher ein Piepsen ausgibt. Stattdessen wird in der Regel eine Sammlung von Teilfunktionalitäten angeboten, die einzeln abgerufen und ggf. miteinander kombiniert werden können.

Das Zerlegen der gewünschten Gesamtfunktionalität eines Softwaresystems in Teilfunktionalitäten ist ein wichtiger Schritt in der Spezifikationsphase des Entwicklungsprozesses (z. B. Sommerville, 2007). Der Benutzer einer Software muss jedoch den umgekehrten Weg gehen, d. h. er sucht sich je nach aktueller Aufgabenstellung die passenden Bausteine zusammen, aus denen er die gewünschte Funktionalität konstruiert. Für den Fall von Programmen zum Erzeugen von Flussdiagrammen wären dies Funktionalitäten wie: Anlegen einer neuen Datei, Erzeugen und Positionieren eines Symbols, Erzeugen einer Beschriftung für das Symbol, ggf. Einpassen der Schrift in das Symbol, Erzeugen eines Verbindungspfeils, ggf. Verändern des Pfeilverlaufs (z. B. weil neue Symbole hinzugekommen sind), Einfärben von Symbolen (z. B. weil es sonst langweilig aussieht), Speichern oder Drucken des Diagramms usw.

Um die Fülle von angebotenen und deshalb von den Probanden in dieser Untersuchung auch frei abrufbaren Funktionalitäten für die Auswertung handhabbar zu machen, erscheint es sinnvoll, sie in einige wenige Kategorien zu gliedern. Das konzeptionelle Gruppieren von Funktionalitäten ist nicht neu und findet in der Regel auf der „hohen“ Ebene der Systemarchitektur statt. Allerdings handelt es sich dabei in der Regel um eine technisch bzw. organisatorisch motivierte Gliederung. So finden sich z. B. bei Denert (1990) die nach Aufgabenbereichen gegliederten Module Dialogführung, Anwendungskern, Datenverwaltung, Anwendungsdienste und Fehlerbehandlung. Alpar, Grob, Weimann und Winter (2005) unterscheiden Datenhaltungs-, Ablage- und Archivierungs-, Datensicherungs-, Büro-, Telekommunikations- und Sicherheitsdienste. Dern (2003), als letztes Beispiel, konzipiert Anwendungstypen zur Klassifikation von Informationssystemen, die durch Anwendung bestimmter Kriterien zustande kommen, z. B. unternehmenskritische/operative, analytische und mobile Anwendungen.

Für die Zwecke dieser Untersuchung wird jedoch eine Kategorisierung mit Bezug zum Einsatz der Software für die Bearbeitung eines ganz bestimmten Problems benötigt und muss insofern auf viel „tieferer“ und vor allem konkreter Ebene stattfinden. Deshalb müssen entsprechende Kategorien neu und speziell entwickelt werden. Der Ansatzpunkt liegt in der Überlegung, dass jede der von einer Software angebotenen Funktionalitäten eine bestimmte Relevanz oder Wichtigkeit für die Bearbeitung eines bestimmten Problems hat. Dieses besteht im vorliegenden Fall im möglichst effizienten Erstellen eines optisch ansprechenden, möglicherweise noch zu ändernden Flussdiagramms. Wenn z. B. das Fehlen einer Funktionalität die Bearbeitung dieser Aufgabe praktisch unmöglich macht, ist ihre Wichtigkeit sehr hoch. Wird eine Funktionalität für die zu lösende Aufgabe gar nicht benötigt, ist dieses Maß sehr niedrig. Um nachvollziehbar beschreiben zu können, in welchem Umfang ein Individuum sich mit Funktionalitäten von großer Wichtigkeit beschäftigt bzw. darüber hinaus auch solche von geringer Wichtigkeit evaluiert, ist jede Interaktion einer bestimmten Kategorie zuzuordnen. Um ein entsprechendes Kategorisierungsschema einfach zu halten, werden drei Funktionalitätsklassen unterschieden.

Als *Kernfunktionalitäten* werden hier solche bezeichnet, die zwingend in irgendeiner Form abrufbar sein müssen, um ein bestimmtes Problem inhaltlich bearbeiten zu können. Der Zusatz „inhaltlich“ ist als Gegenteil von „technisch“ zu verstehen und dient der Abgrenzung von den nachfolgend beschriebenen Dienstfunktionalitäten. Aus dem hier zugrunde liegenden Verständnis von Kernfunktionalitäten folgt, dass es möglich sein muss, diese abschließend aufzuzählen. Im vorliegenden Fall umfasst diese Aufzählung das Erzeugen und Beschriften von Symbolen und Pfeilen sowie die Manipulation der verschiedenen Elemente, d. h. sie wieder zu entfernen sowie ihre Größe und Position zu verändern, z. B. um sie aneinander anzupassen, zueinander auszurichten oder Platz für neu hinzukommende Elemente zu schaffen.

Als *Dienstfunktionalitäten* werden hier solche bezeichnet, die quasi standardmäßig von jeder Software, für die sie sinnvoll sind, angeboten werden. Solche Funktionalitäten können als anwendungsneutral oder generisch beschrieben werden und sind typischerweise im „Datei“- bzw. im „Bearbeiten“-Menü eines Programms zu finden. Sie sind entweder aus technischen (vs. inhaltlichen, s. o.) Gründen notwendig (z. B. „Datei öffnen“, „Datei speichern“) oder unterstützen den technischen Vorgang der Problembearbeitung

durch Bereitstellung von elementaren oder Basishandlungen (z. B. „Kopieren“, „Einfügen“, „Rückgängig“).

Als *Extrafunktionalitäten* werden hier solche bezeichnet, die weder Kern- noch Dienstfunktionalitäten sind, also solche, deren Fehlen die Zielerreichung nicht verhindert, die aber auch nicht als generische Basishandlungen einzuordnen sind. Darunter fallen vor allem Funktionalitäten, die der Bequemlichkeit dienen (z. B. automatisches Ausrichten von Elementen), individuellen gestalterischen Spielraum eröffnen (z. B. Einfärben von Symbolen, Schatteneffekte und dergleichen), oder gar nichts mit der Aufgabe zu tun haben (z. B. eine E-Mail-Schaltfläche).

Es sei an dieser Stelle noch einmal explizit darauf hingewiesen, dass sich insbesondere die Zuordnung zu Kern- vs. Extrafunktionalitäten aus dem oder den zu bearbeitenden Problem(en) ergibt. In der hier verwendeten Bedeutung ist keine Funktionalität per se das eine oder andere.²⁰ Was für eine Aufgabenstellung eine Kern-, kann für eine andere eine verzichtbare Extrafunktionalität sein.

Um die Einordnung einer konkreten Funktionalität in eine der hier eingeführten Kategorien transparenter zu machen, wurde vom Verfasser eine entsprechende Vorschrift entwickelt. Diese basiert auf der Verortung in einem durch die Dimensionen Notwendigkeit, Problemspezifität und Ziel(erreichungs)beitrag aufgespannten Raum (vgl. Abb. 5.2). Das Kriterium der *Notwendigkeit* einer Funktionalität F drückt aus, ob es grundsätzlich *sinnvoll möglich* ist, die Aufgabe auch ohne das Vorhandensein einer Implementation von F zu lösen. Der *Zielbeitrag* von F drückt aus, ob F bei der Bearbeitung der Aufgabe sinnvoll einsetzbar ist, also zur effizienten Lösung des Problems beiträgt, unabhängig davon, ob prinzipiell auch auf F verzichtet werden kann. Bei der *Problemspezifität*, im Folgenden auch etwas weniger sperrig als *Aufgabenbezug* bezeichnet geht es um die Frage: Hat F einen konkreten inhaltlichen Bezug zur Aufgabe, oder leistet F einen allgemeinen Dienst, der problemübergreifend genutzt werden kann? Jedes dieser Kriterien dient als eine Dimension mit dichotomer Struktur, auf der eine Funktionalität mit ausreichender Sicherheit einer der beiden Ausprägungen „hoch“ bzw. „niedrig“ zugeordnet werden kann.

Als Kernfunktionalitäten (in Abb. 5.2 hellgrau dargestellt) werden im Kontext dieser Untersuchung entsprechend solche eingeordnet, die sowohl notwendig sind als auch dazu beitragen, das Ziel zu erreichen. Sobald diese beiden Kriterien erfüllt sind, ist die Problemspezifität nebensächlich und teilweise schwer zu bestimmen. Der Übergang zwischen hoher und niedriger Ausprägung ist je nach Sichtweise fließend. So ist z. B. intuitiv klar, dass so etwas wie „Schriftgröße verändern“ Kernfunktionalität sein muss, um auch längere Texte in Symbole einpassen zu können. Besonders problemspezifisch ist dies aber nicht, sondern vielmehr regelmäßig in fast allen Anwendungen, die in der einen oder anderen Form mit Text umgehen, implementiert.

²⁰ Im Zusammenhang z. B. mit der Architektur eines Softwaresystems wird der Begriff Kernfunktionalität ganz anders verwendet, nämlich um diese von Dialogfunktionalitäten abzugrenzen. Erstere erledigen die „eigentlichen“ Aufgaben im Inneren, dem Kern des Systems, Letztere führen den Dialog mit dem Benutzer, sind also für die interaktive Beschaffung von Daten und die Ausgabe von Informationen zuständig (Röck, 2001; vgl. auch Denert, 1990).

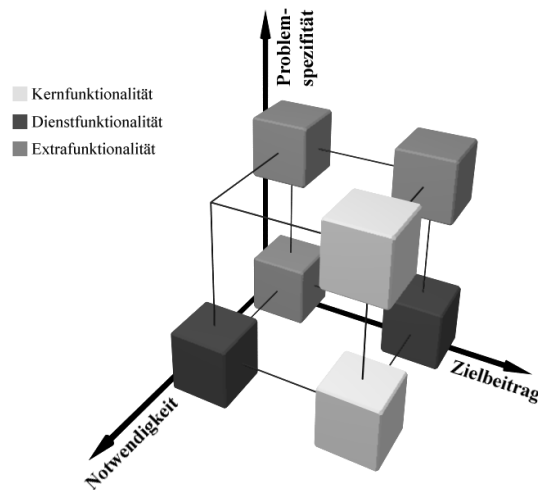


Abbildung 5.2: Funktionalitätskategorien (eigene Darstellung)

Das Kriterium der Problemspezifität dient deshalb in erster Linie zur Abgrenzung der Dienstfunktionalitäten (dunkelgrau), welche sich aufgrund ihres generischen Charakters in den Bereichen mit niedrigem/fehlendem Aufgabenbezug befinden. Eine Dienstfunktionalität, die zwar notwendig ist, aber keinen Zielbeitrag liefert (in Abb. 5.2 links unten), ist z. B. „Datei speichern“; eine, die hilft, das Ziel zu erreichen, aber nicht unbedingt nötig ist (rechts unten), ist z. B. „Kopieren & Einfügen“.

Als Extrafunktionalitäten (grau) gelten hier zum einen solche, die zwar verzichtbar sind, aber einen problemspezifischen Zielbeitrag leisten (rechts oben), wie z. B. „automatisches Ausrichten“, zum anderen solche, die unabhängig von ihrem Aufgabenbezug weder notwendig sind noch zur effizienten Zielerreichung beitragen (hinten), z. B. „Symbole einfärben“ oder eine E-Mail-Schaltfläche. Das Cluster „notwendig und aufgabenbezogen ohne Zielbeitrag“ bleibt unbesetzt, weil die Kombination dieser Ausprägungen nicht sinnvoll erscheint.

Bei der weiteren Analyse werden nur Kern- und Extrafunktionalitäten berücksichtigt. Der Grund für die Nichtberücksichtigung von Dienstfunktionalitäten ist, dass sie die nachfolgende Auswertung unnötig komplizieren würden. Unnötig deshalb, weil sie einem Quasistandard entsprechen, d. h. sowohl bei im Alltag als auch in dieser Untersuchung verwendeten Programmen immer vorhanden sind und immer gleich funktionieren, sodass kein Einfluss auf den Verlauf oder das Ergebnis einer Evaluation zu erwarten ist.

Tatsächlich wurden Dienstfunktionalitäten von den Probanden rein mechanisch verwendet. Die wenigen Verbalisierungen, die im Zusammenhang mit deren Benutzung überhaupt aufgetreten sind, lassen nicht erkennen, dass es sich dabei um etwas zu Erprobendes handelt. Vielmehr unterstreichen sie die hier vertretene Auffassung, dass sowohl das Vorhandensein als auch eine bestimmte Funktionsweise von Dienstfunktionalitäten vorausgesetzt werden, und haben insofern lediglich den Charakter einer Beschreibung des

aktuellen Tuns (z. B. „So, hier noch mal eben kopieren“, Vp1.232²¹) oder einer neutralen Kenntnisnahme einer Selbstverständlichkeit (z. B. „Gut, machen wir . . . kopieren wir den «Hörer abnehmen», Steuerung-Copy, Steuerung-V, geht auch“, Vp12.118).

5.4.2.2 Rahmenmerkmale und Orientierungsphasen

Eine Software ist nicht nur eine Ansammlung von Funktionalitäten, die bei Bedarf zum Zweck der Zielerreichung abgerufen werden können. Vielmehr müssen diese Funktionalitäten auch irgendwie angeboten oder präsentiert werden bzw. ganz allgemein in einem bestimmten Rahmen „stattfinden“. Die Eigenschaften einer Software, die diesen Rahmen bestimmen, werden hier entsprechend als *Rahmenmerkmale* bezeichnet. Damit ist z. B. gemeint, wie die Dialogelemente, über die Funktionalitäten abgerufen oder in ihrem konkreten Wirken gesteuert werden können, gestaltet sind, wie die Arbeitsfläche des Programms gegliedert ist, ob und in welcher Form dem Benutzer aktiv Unterstützung angeboten wird, ob die Zeichenfläche mit einem Raster hinterlegt ist usw.

Rahmenmerkmale sind anders als Funktionalitäten keine Angebote, einen bestimmten, mehr oder weniger zielführenden Effekt zu produzieren, und können insofern nicht abgerufen werden, sondern sind „einfach da“. Sie stehen im Gegensatz zu den angestrebten Effekten von Funktionalitätsabrufen nicht im Mittelpunkt bzw. sind nicht Gegenstand von Interaktionen, sondern gestalten den Rahmen, in dem diese erfolgen. Dennoch können Rahmenmerkmale nicht nur passiv wahrgenommen (*Rahmenwahrnehmung*), sondern auch aktiv erforscht werden (*Rahmenexploration*): Wenn z. B. auf der Zeichenfläche kein Raster angezeigt wird, kann nach einer Möglichkeit gesucht werden, dies zu konfigurieren. Streng genommen handelt es sich bei einer solchen Möglichkeit um eine Funktionalität. Da sich deren Ergebnis aber nur auf die Software selbst bezieht, also den Rahmen, in dem die Benutzung der „eigentlichen“ Funktionalitäten erfolgt, wird auch eine derartige Suche nach Konfigurationsmöglichkeiten als Exploration eines Rahmenmerkmals interpretiert.

Als *Orientierungsphasen* werden hier Abschnitte während des Evaluationsprozesses bezeichnet, in denen das Individuum offensichtlich nicht auf der Suche nach dem Auslöser für eine bestimmte Funktionalität ist, sondern sich einen allgemeinen Überblick über die Benutzungsoberfläche eines Programms verschafft. Darunter fällt auch das in diesem Sinne ziellose Browsen über Schaltflächen bzw. durch Menüs, um zu sehen, welche interessanten Möglichkeiten die Software über die bisher gezielt untersuchten hinaus zu bieten hat.

²¹ Diese Notation wird verwendet, um auf eine bestimmte Zeile im Transkript einer Versuchsperson zu verweisen, in diesem Fall auf Vp1, Zeile 232. Die Transkripte sind dieser Arbeit auf CD-ROM beigelegt.

5.4.2.3 Abstraktion des Evaluationsverlaufes

Um zu ermitteln, in welchem Umfang jeweils die verschiedenen Funktionalitätskategorien exploriert²², Rahmenmerkmale wahrgenommen und Orientierungsphasen absolviert wurden, musste für jeden Probanden ein Modell des Evaluationsverlaufes erstellt werden, welches es erlaubt, die entsprechenden Kennzahlen zu ermitteln. Es war also eine Methode zu entwickeln, nach der die Rohdaten in Gestalt der AVI-Videos in eine abstrakte, d. h. auf das Wesentliche reduzierte Form transformiert werden.

Diese Methode wurde inspiriert durch Arbeiten zur sog. *Behavioral-Process Analysis* auf dem Gebiet der Entscheidungsforschung (z. B. Jacoby, Chestnut, Weigl & Fisher, 1976; Jacoby, Chestnut & Fisher, 1978; Jacoby et al., 1994, 1987; Payne, Braunstein & Carroll, 1978). In diesen Arbeiten steht im Gegensatz zum „üblichen“ oder traditionellen Vorgehen²³ der *Prozess*, in dessen Verlauf ein Individuum zu einer Bewertung oder Entscheidung kommt, im Vordergrund. Entsprechend wird versucht, z. B. die Abfolge kognitiver Zustände, die ein Individuum durchläuft, sichtbar zu machen (Payne et al., 1978) oder aufzuzeichnen, welche für eine Entscheidung zur Verfügung stehenden Informations-Items in welcher Reihenfolge und Häufigkeit zugegriffen werden (Jacoby et al., 1987).

Analog zum Vorgehen von Jacoby et al. (1987) wurde hier der Ansatz gewählt, das Evaluationsverhalten bzw. das Ausprobieren im Wesentlichen als Informationssuche zu modellieren. Dies erscheint unmittelbar plausibel, geht es doch beim Ausprobieren im Wesentlichen darum, Informationen darüber zu beschaffen, wie die Software funktioniert bzw. welche Ergebnisse sie liefert, um auf dieser Basis eine Entscheidung zu treffen. Das resultierende Modell soll es erlauben, möglichst unmittelbar und einfach verschiedene, das Evaluationsverhalten beschreibende Kennzahlen abzuleiten. Dazu ist es erforderlich, für jeden Probanden in übersichtlicher Form zu notieren, welche Funktionalitäten (eingesortiert in die jeweilige Kategorie) mit welchem Erfolg in welcher Reihenfolge ausprobiert wurden, in welchem Maße Rahmenmerkmale gewürdigt wurden sowie Orientierungsphasen stattfanden.

Zu diesem Zweck wurde vom Verfasser ein Formblatt entwickelt und im Verlauf der Auswertung sukzessive verfeinert, sodass in der finalen Version möglichst viele Aspekte des Evaluationsverhaltens abgebildet werden konnten. Dieses Formblatt wird im Folgenden als *Interaktionsprotokollbogen (IPB)*, die darin enthaltenen Aufzeichnungen als *Interaktionsprotokoll* bezeichnet. Als zu erfassende Einheiten werden *Suchepisoden* zugrunde gelegt. Darunter ist jeweils die Menge aller zusammenhängenden Interaktionen zu verstehen, die mit dem Ziel, einen bestimmten Effekt hervorzurufen, erfolgen. Eine

²² Synonym zu „Erschließen der Funktionsweise und Begutachten der konkreten Ergebnisse eines Funktionalitätsabrufs“ wird im Folgenden in der Regel die weniger sperrige Formulierung „Explorieren einer Funktionalität“ verwendet.

²³ Dieses besteht darin, dass Versuchspersonen einem oder mehreren Stimuli ausgesetzt werden. Anschließend werden die interessierenden Variablen erhoben, ohne den Prozess, durch den sich diese Variablen einstellen, explizit zu beobachten. Payne et al. (1978, S. 17) sprechen von „input-output analyses“ und führen weiter aus: „Typically, research has been focused on data which reflect only the end product of the decision processes, for example, choice proportions, rankings, or ratings.“

Suchepisode beginnt im Moment der erkennbaren Absicht, eine bestimmte Funktionalität abzurufen, und endet, wenn der Abruf gelungen ist oder die Suche erfolglos abgebrochen wird. Konkret werden für jeden Probanden und jede Software folgende Informationen im Zeitablauf erfasst:

- Funktionalitätskategorie einer Suchepisode: Wird eine Kern- oder eine Extrafunktionalität exploriert?
- Anlass einer Suchepisode: Exploriert der Proband eine Funktionalität von sich aus oder wurde er durch die Software darauf gebracht, z. B. durch einen zufällig entdeckten, interessant klingenden Menüeintrag?
- Dauer einer Suchepisode: Wie lange dauert es ab dem Beginn der Suche bis zum Abbruch oder erfolgreichen Abschluss?
- Bewertung des Ergebnisses einer Suchepisode: Bewertet der Proband die in der Suchepisode ermittelte Information positiv, negativ oder neutral bzw. gar nicht?
- Abbruch einer Suchepisode: Hat der Proband gefunden, was er gesucht hat, oder hat er die Suche aufgegeben? Hier liegt ein anderer Gedanke zugrunde als beim vorigen Punkt: Die negative Bewertung einer Suchepisode muss nicht auf dem erfolglosen Abbruch basieren, sondern kann als Ursache auch z. B. die schlechte Implementation einer Funktionalität haben. Andererseits muss auch ein erfolgreicher Abbruch nicht unbedingt negativ bewertet werden, z. B. wenn es sich um eine verzichtbare Extrafunktionalität handelt.
- Wiederholung einer Suchepisode: Handelt es sich um die erste Exploration der Funktionalität oder hat der Proband die Funktionsweise vergessen bzw. muss sich vergewissern und sucht deshalb noch einmal?
- Wahrgenommene sowie explorierte Rahmenmerkmale.
- Stattgefundene Orientierungsphasen.

Ein IPB besteht aus drei Tabellen – eine für jedes untersuchte Programm – des in Abb. 5.3 dargestellten Formats (für ein ausgefülltes Beispiel s. Abschnitt 5.4.2.4). Ganz links werden Metainformationen (Programmname, Komplexität der erzeugten Abbildung, aufgewendete Zeit) eingetragen. Jede Spalte des Gitters ist für die Erfassung eines Evaluationsschritts (Suchepisode, Orientierungsphase oder Rahmenexploration) vorgesehen. Rahmenwahrnehmungen werden in derselben Spalte aufgenommen wie der Evaluationsschritt, in dessen Zusammenhang die Wahrnehmung erfolgte. Werden im Zuge eines Schrittes mehrere Rahmenmerkmale wahrgenommen, erstreckt sich dieser Schritt entsprechend über mehrere Spalten. Über jeder Spalte wird jeweils ein Verweis auf die entsprechenden Zeilen des Transkriptes eingetragen.

Die Zeilen des IPB repräsentieren die möglichen Inhalte eines Evaluationsschritts: Orientierung (*O*), Kernfunktionalität (*S_E* bis *M*), Extrafunktionalität (*X*) oder Rahmenmerkmal (*R*). Der Bereich für die Erfassung von Funktionalitätsexplorationen ist noch in Kern-

Prog.name (Kplx. / Zeit)	O										
	S _E										
	S _B										
	V _E										
	V _B										
	M										
	X										
	R										

Abbildung 5.3: Format des Bogens zur abstrakten Erfassung des Evaluationsverhaltens (IPB; eigene Darstellung)

und Extrafunktionalitäten unterteilt. Die Kernfunktionalitäten sind zwecks besserer Übersichtlichkeit einzeln aufgezählt, sodass leichter erkennbar ist, ob a), angesichts ihrer Notwendigkeit, sie vollständig abgearbeitet wurden, und b) eine bestimmte Funktionalität bzw. ein Aspekt derselben wiederholt kommentiert oder exploriert wird. Als Kernfunktionalitäten gelten konkret (vgl. Abschnitt 5.4.2.1):

S_E – Erzeugen eines Symbols (*nicht* Verschieben/Größe ändern $\rightarrow M$);

S_B – Beschriften eines Symbols (*nicht* Ändern von Schriftart $\rightarrow X$ /-größe $\rightarrow M$);

V_E – Erzeugen einer Verbindungslinie mit Pfeilspitze (*nicht* Ändern der Pfeilform $\rightarrow X$);

V_B – Beschriften einer Verbindung (*nicht* Ändern von Schriftart/-größe, s. S_B);

M – Manipulieren eines Objektes (Position, Größe, Linienverlauf, Entfernen).

Diese Aufzählung ergibt sich aus der Aufgabe, die die Versuchspersonen mit der gewählten Software angeblich bearbeiten sollen (vgl. Abschnitt 5.3.3.2): dem Zeichnen und ggf. nachträglichen Verändern eines Flussdiagramms auf der Basis der Elemente sowie deren zugrunde liegender Semantik, wie sie im Rahmen der Vorbereitung der Probanden vorgestellt wurden (vgl. Abschnitt 5.3.4.2). Demnach werden außer Symbolen, Verbindungen und Beschriftungen keine weiteren grafischen Objekte, wie z. B. Hintergründe oder Annotationen benötigt. Über das Erzeugen und Kombinieren dieser Elemente hinaus wurden lediglich Manipulationen hinsichtlich Position und Größe als benötigte Funktionalität eingeführt (durch Ankündigung eventueller Änderungen sowie den generellen Hinweis, dass die Symbole gleich groß und zueinander ausgerichtet sein sollen). Weder explizit noch implizit wurde dagegen gefordert, z. B. das standardmäßige Aussehen dieser Elemente (Farbe, Strichstärke, Pfeilform usw.) frei gestalten zu können.

In das entsprechende Feld für einen Evaluationsschritt wird, soweit anwendbar, dessen Dauer eingetragen. Keine Dauer haben Rahmenwahrnehmungen sowie wertende Kommentare bezüglich neu entdeckter Aspekte von Funktionalitäten, die bereits früher untersucht wurden, also nicht Gegenstand der aktuellen Suchepisode sind, falls denn überhaupt

eine stattgefunden hat. Beispiel: Der Proband sucht, findet und wertet den Mechanismus zum Erzeugen eines Symbols. Einige Bearbeitungsschritte später, beim Betrachten des bisher erzeugten Diagramms, fällt ihm auf, dass die Größe der unterschiedlichen Symbole aufeinander abgestimmt ist und gibt hierzu einen positiven Kommentar ab. Dieser Kommentar sollte in seiner Eigenschaft als Wertung prinzipiell Eingang in den IPB finden. Er kann zwar einer Funktionalität, aber keiner Suchepisode zugeordnet werden. Deshalb kann auch nicht angegeben werden, ob und wie lange der Proband nach dieser Information (Symbole haben automatisch die gleiche Größe) gesucht hat.

Weiteren Aufschluss über Verlauf und Ergebnis eines Evaluationsschritts gibt die Formatierung des jeweiligen Feldes im IPB-Gitter.²⁴ Um auszudrücken, dass die entsprechende Suchepisode nicht vom Probanden, sondern von der Software initiiert wurde, wird die zugehörige Zeitangabe fett geschrieben, z. B. **13**. „Von der Software initiiert“ bedeutet, dass der Proband auf ein zufällig entdecktes Angebot reagiert: Im Verlauf einer Orientierungsphase oder auf der Suche nach einer Funktionalität wird ein interessant klingendes oder die Neugier weckendes Dialogelement entdeckt, z. B.: „Sekundäre Textbereiche? Oops. Was ist das?“ (Vp6.22). Die Suchepisode wird also nicht durch eine bereits vorher bestehende Absicht des Individuums ausgelöst, sondern dieses wird vielmehr durch die Software überhaupt erst darauf gebracht oder dazu verleitet, eine bestimmte, in der Regel eine Extra-Funktionalität zu explorieren.

Der erfolglose Abbruch einer Suchepisode, d. h. das Individuum gibt die Suche auf, ohne das Ziel erreicht zu haben, wird markiert, indem die zugehörige Zeitangabe mit einem Rahmen versehen wird, z. B. 13. Besteht das Ziel einer Suchepisode in einer Information, die bereits Gegenstand eines früheren Evaluationsschritts war, wird die zugehörige Zeitangabe unterstrichen, z. B. 13. Anlass für eine wiederholte Exploration muss nicht unbedingt ein vorheriger erfolgloser Abbruch sein. Stattdessen könnte das Individuum einfach vergessen haben oder unsicher geworden sein, wie eine bestimmte Funktionalität abgerufen wird. Sämtliche Markierungen können ggf. auch miteinander kombiniert werden, auch wenn nicht alle Kombinationen sinnvoll sind.²⁵ Eine wiederholte Suche, die von der Software angeregt wurde und erfolglos abgebrochen wird, würde dann z. B. so aussehen: 13.

Die Bewertung eines Evaluationsschrittes, genauer: dessen, was ein Individuum in dessen Verlauf über die Software erfahren hat, erfolgt durch entsprechende Einfärbung des jeweiligen Feldes im IPB-Gitter. Rot steht für eine explizit negative Wertung, grün für eine explizit positive. Wird ein neutraler (Kenntnisnahme), ein ambivalenter (sowohl positive als auch negative Aspekte) oder gar kein Kommentar abgegeben, wird das Feld gelb gefärbt.

Um die naturgemäß große Heterogenität der Daten zu bewältigen und eine einheitliche Kodierung der Videoprotokolle zu gewährleisten, wurden im Verlauf der Auswertung sukzessiv Regeln hinsichtlich der Berücksichtigung von Interaktionen und Verbalisierungen entwickelt. Revisionen des Regelwerks erfolgten anlässlich des Auftretens von Spezial-

²⁴ Ein Beispiel, in dem alle bisher und nachfolgend beschriebenen Fälle dargestellt und nochmals erläutert werden, findet sich weiter unten in Abschnitt 5.4.2.4.

²⁵ Eine wiederholte Suche wird normalerweise nicht von der Software angeregt sein.

fällen, die mit den bis dahin etablierten Regeln nicht entschieden werden konnten. Die bis dahin aufbereiteten Daten wurden unter Berücksichtigung der Modifikationen überarbeitet, sodass alle finalen, der weiteren Auswertung zugeführten Abstraktionen den Vorgaben der letzten, im Folgenden erläuterten Version genügen.

Für wertende Kommentare gilt, dass diese möglichst explizit und möglichst wenig der Interpretation überlassen sind. Das heißt insbesondere für als positive Wertung einzustufende Kommentare, dass ein einfaches „gut“ nicht ausreicht. Dieses wird vielmehr als neutrale Bemerkung im Sinne eines Füllwortes oder einer Kenntnisaufnahme behandelt (z. B. nach Abschluss einer Interaktion: „Gut, dann wollen wir mal noch . . .“), es sei denn, der Tonfall lässt absolut keinen Zweifel an einer anderen Deutung. Ansonsten bedarf es einer adverbialen Beifügung (z. B. „Sehr gut!“) oder eines ganzen Satzes (z. B. „Das ist gut.“). Für negative Bemerkungen gilt dies nicht, da entsprechende Wörter normalerweise nicht analog zu besagtem „gut“ verwendet werden. Abgesehen davon tauchten in dieser Studie Wörter wie z. B. „schlecht“ oder „blöd“ stets im Zusammenhang mit mehr oder weniger vollständigen Sätzen auf.

Hinsichtlich der Handlungen der Versuchspersonen gilt, dass grundsätzlich nur die erste Nutzung einer Funktionalität berücksichtigt wird, es sei denn, weitere Nutzungen erfolgen explizit zu dem Zweck, neue oder weitere Informationen bezüglich ihres Verhaltens zu erarbeiten (*Instrumentalitätsregel*). Sonstige wiederholte Nutzungen werden als rein instrumentell angesehen, d. h. sie erfolgen nicht mehr um des Ausprobierens ihrer selbst willen. Zum Beispiel kann das Erzeugen eines zweiten Symbols rein instrumentell zu dem Zweck erfolgen, das Explorieren der Funktionalität zum Verbinden zweier Symbole zu ermöglichen. Aufgezeichnet wird insofern also nicht, wie oft eine Funktionalität genutzt wird, sondern wie oft und mit welchem Erfolg versucht wird, jeweils neue Informationen über sie zu erlangen.

Den Sinn dieser Regel möge folgendes Beispiel verdeutlichen: Proband A erzeugt vier in einer Flucht liegende Symbole und verbindet jeweils zwei davon mit einer geraden Linie. Proband B erzeugt auch vier Symbole, verbindet zwei davon mit einer geraden Linie, zwei weitere mit einer abgewinkelten Linie und probiert anschließend aus, ob er auch eine Verbindung zwischen einem Symbol und einer Linie zeichnen kann. Beide Probanden haben im Endeffekt jeweils dreimal die Kernfunktionalität „Verbindung erzeugen (V_E)“ genutzt. Es ist aber intuitiv klar, dass beide etwas qualitativ anderes gemacht haben.

Dieser Unterschied wird durch Anwendung der oben beschriebenen Instrumentalitätsregel deutlich gemacht: Für Proband A wird notiert, dass er *eine* Information über V_E erhoben hat, denn die letzten beiden Nutzungen werden als rein instrumentell nicht berücksichtigt. Proband B kommt dagegen auf *drei* Informationen über V_E , denn seine erneuten Nutzungen waren keine „platten“ Wiederholungen der ersten, sondern hatten stets den Zweck, das Wissen über V_E zu vertiefen.

Zu unterscheiden sind die „platten“ Wiederholungen des Probanden A jedoch von den weiter oben diskutierten wiederholten Explorationen, die durch Unterstreichen der Zeitangabe markiert werden. Hier geht es zwar auch nochmals um dieselbe Funktionalität, jedoch vor anderem Hintergrund, denn anders als bei Proband A wird nicht etwas Bekanntes abgerufen, sondern erneut versucht, überhaupt erst einmal die Funktionsweise zu erschließen.

Eine weitere Regel befasst sich mit der Behandlung einer Suchepisode, die durch eine Ablenkung unterbrochen wird (*Unterbrechungsregel*). Das ist zum Beispiel dann der Fall, wenn ein Proband während der Suche nach dem Mechanismus zum Erzeugen von Verbindungen auf die Möglichkeit, die Symbole farbig zu gestalten, stößt und dies sofort ausprobieren will. Es wäre in diesem Zusammenhang inhaltlich falsch, dies als einen erfolglosen Abbruch im Sinne eines Aufgebens, wie es oben diskutiert wurde, zu behandeln. Konsequenterweise müsste dann die Wiederaufnahme der ursprünglichen Suche als zweiter Anlauf notiert werden, was ebenso falsch wäre, denn weder ist der Proband im ersten Anlauf gescheitert, noch hat er das dort Gelernte wieder vergessen.

Es wird deshalb festgelegt, dass eine solche Suchepisode ignoriert wird, wenn die ursprüngliche Suche nicht wieder aufgenommen wird. Einem so „leichtfertig“ aufgegebenen Aspekt kann nicht unterstellt werden, bestimmend für das Evaluationsverhalten zu sein. Bei Wiederaufnahme gilt die Suchepisode als unterbrochen, d. h. sie besteht aus zwei Fragmenten, deren Zeiten und ggf. Wertungen zusammengefasst werden. Die grafische Aufnahme in den IPB lässt die Unterbrechung zugunsten der Ablenkung deutlich werden (vgl. Zeile S_B Abb. 5.4).

Die nach dem in diesem Abschnitt erläuterten Vorgehen erarbeiteten Interaktionsprotokolle sind auf Anfrage vom Verfasser erhältlich.

5.4.2.4 Abstraktionsbeispiel

Die folgende Abbildung zeigt einen Ausschnitt aus dem Interaktionsprotokoll eines fiktiven Probanden. Die senkrecht geschriebenen grauen Zahlen dienen hier nicht, wie sonst, zur Bezugnahme auf das Transkript, sondern zur Identifikation der einzelnen Spalten für die nachfolgende Erläuterung.

	0	1	2	3	4	5	6	7	8
Visio (6 – 555'')	O	30							
	S_E		25						
	S_B					–	–	70	
	V_E				40				35
	V_B								
	M								
	X						30		
	R								

Abbildung 5.4: Beispiel für ein Interaktionsprotokoll (eigene Darstellung)

Spalte 0 gibt Auskunft darüber, dass der Proband Visio ausprobiert und dabei eine Abbildung mit der Komplexität²⁶ 6 gezeichnet hat. Insgesamt hat er sich 9 Minuten und 55 Sekunden (555'') mit der Software beschäftigt. Der erste Evaluationsschritt besteht in

²⁶ Komplexität ist eine Kennzahl, mit der die Realitätsnähe eines im Zuge des Ausprobierens erstellten Flussdiagramms beschrieben wird. Zur Ermittlung s. Abschnitt 6.2.3.1.

einer Orientierungsphase von 30 s, in deren Verlauf der Proband das Fehlen eines Zeichenrasters bemängelt. Dies wird in derselben Spalte wie die Orientierungsphase durch rotes²⁷ Einfärben des Rahmenmerkmal-Feldes notiert (Spalte 2). Anschließend versucht er, ein Symbol zu erzeugen und braucht 25 s, bis er herausgefunden hat, wie es funktioniert. Dieser Erfolg wird aber nicht weiter kommentiert, also erscheint die Suchepisode vor gelbem Hintergrund in Spalte 3. Anschließend wird gleich noch ein Symbol erzeugt, damit etwas zum Verbinden da ist. Gemäß der Instrumentalitätsregel erscheint dieser Evaluationsschritt nicht im Interaktionsprotokoll. Die Suche nach dem Mechanismus zum Erzeugen einer Verbindung wird nach 40 s mit einem negativen Kommentar vorerst aufgegeben. Im Interaktionsprotokoll erscheint dies als rotes Feld mit eingerahmter Zeitangabe (Spalte 4).

Der Proband wendet sich als nächstes dem Beschriften seiner Symbole zu. Nachdem er die Schaltflächen abgesucht hat, versucht er es über das Kontextmenü. Dort fällt ihm jedoch als Erstes der Eintrag „Symbolfarbe“ auf. Diese Funktionalität wird zwar nicht benötigt, interessiert ihn aber trotzdem. Er unterbricht die Suche nach dem Textwerkzeug, um zunächst 30 s mit den Farben herumzuexperimentieren und dabei seine Begeisterung über die gebotenen Möglichkeiten auszudrücken. Dieser von der Software initiierte Evaluationsschritt erscheint entsprechend mit fett gedruckter Zeitangabe und grünem Hintergrund in der Zeile für Extrafunktionalitäten (Spalte 6). Anschließend nimmt der Proband die eigentliche Suche wieder auf. Das Beschriften gelingt schließlich und wird mit einem neutralen „Okay“ quittiert. Entsprechend der Unterbrechungsregel erstreckt sich dieser Evaluationsschritt über drei Spalten (Spalte 5 bis 7), um die Ablenkung auch grafisch einzuschließen. Vermerkt wird die Summe der Zeiten vor und nach der Unterbrechung, angesichts der abschließenden neutralen Kenntnisnahme auf gelbem Grund.

Schließlich versucht sich der Proband noch einmal am Verbinden seiner beiden Symbole. Diesmal klappt es und er findet, dass diese Funktionalität trotz anfänglicher Schwierigkeiten „eigentlich ziemlich clever“ umgesetzt ist. Die positive Bewertung führt zur grünen Färbung des Feldes. Die Zeitangabe wird unterstrichen, weil es sich um die erneute Exploration einer bereits vorher schon einmal untersuchten Funktionalität handelt (Spalte 8).

²⁷ In der dieser Veröffentlichung ist rot in Abb. 5.4 als dunkelgrau dargestellt, grün als mittleres grau und gelb als hellgrau.

6 Ergebnisse der empirischen Untersuchung

In diesem Kapitel werden die in der Auseinandersetzung mit den erhobenen Daten entwickelten Auswertungsmethoden beschrieben und die entsprechenden Ergebnisse dargestellt. Als erster Schritt werden die Daten zunächst nach Hinweisen auf Zusammenhänge untersucht, die unmittelbar, d. h. ohne tiefer gehende inhaltliche Beschäftigung mit dem Datenmaterial beobachtbar sind. Deshalb wird dieser Teil zur Abgrenzung von den anschließenden qualitativen Analysen als quantitative Auswertung bezeichnet. Die beiden qualitativen Schritte dienen unmittelbar der Beantwortung der Fragen nach dem Evaluationsverhalten der Individuen sowie den Gründen für ihre Entscheidung. Die Diskussion der Ergebnisse erfolgt dann zusammenhängend im nächsten Kapitel.

6.1 Quantitative Auswertung

6.1.1 Auswahlhäufigkeit

Von den 20 Versuchspersonen entschieden sich 6 für SmartDraw, 4 für Flowcharter und 10 für Visio/DS. Dabei ist jedoch zu berücksichtigen, dass nicht jedes Programm auch jedem Probanden zur Auswahl stand (vgl. Abschnitt 5.3.7). Jeweils ins Verhältnis dazu gesetzt, ob die jeweilige Option überhaupt zur Verfügung stand, ergibt sich folgendes Bild: SmartDraw 6-mal von 20 (30 %), Flowcharter 4-mal von 18 (22 %), Visio/DS 10-mal von 19 (53 %). Von den Probanden, denen alle drei Programme zur Wahl standen, wurde SmartDraw 5-mal ausgewählt, Flowcharter 3-mal, Visio/DS 9-mal.

6.1.2 Bearbeitungsreihenfolge

Mit Ausnahme von Vp2, Vp4, Vp8 und Vp18 haben alle Probanden die Alternativen in der durch ihre Anordnung auf dem Bildschirm suggerierten Reihenfolge (von oben nach unten) untersucht. Vp2 und Vp4 arbeiteten das Angebot von unten nach oben ab, Vp8 wählte ohne erkennbaren Grund die Reihenfolge oben – unten – Mitte, und Vp18 begann wegen des zur Aufgabe passenden Namens mit dem in der Mitte positionierten Programm (Flowcharter). Daraus ergibt sich die in Tabelle 6.1 dargestellte Verteilung der Bearbeitungspositionen. Berücksichtigt werden hier nur die 17 Probanden, denen drei (statt nur zwei) Optionen zur Auswahl standen.

Bearbeitungs- position	SmartDraw	Flowcharter	Visio/DS
1.	8	8	1
2.	3	8	6
3.	6	1	10

Tabelle 6.1: Verteilung der Bearbeitungspositionen für die angebotenen Programme (nur Probanden mit drei Optionen)

Um zu prüfen, ob es Hinweise auf einen Einfluss der Bearbeitungsreihenfolge auf das Auswahlresultat gibt¹, wurde ermittelt, in wie vielen Fällen das ausgewählte Programm (im Folgenden auch als der Sieger bezeichnet) als Erstes (5-mal), Zweites (8-mal) bzw. Drittes (7-mal) begutachtet wurde. Eine Verzerrung kommt dabei zustande, weil drei Versuchspersonen nur zwei Optionen hatten. Werden die Siegerpositionen für beide Konstellationen getrennt ausgezählt, ergibt sich die in Tabelle 6.2 dargestellte Verteilung.

Probanden	Siegerposition		
	1.	2.	3.
mit zwei Optionen	2	1	—
mit drei Optionen	3	7	7

Tabelle 6.2: Verteilung der Siegerpositionen

Für den Drei-Optionen-Fall ist festzustellen, dass am häufigsten das als Zweites bzw. Drittes untersuchte Programm ausgewählt wurde, nämlich in jeweils 41 % der Fälle. Bezogen auf die einzelnen Optionen, d. h. wie oft eine *bestimmte* Software gewählt wurde, wenn sie als Erste, Zweite bzw. Dritte untersucht wurde, ergibt sich die in Tabelle 6.3 dargestellte Verteilung (nur Probanden mit drei Optionen). Auffällig ist dabei vor allem, dass Flowcharter ausschließlich nach Bearbeitung an zweiter Position gewählt wurde, obwohl es ebenso häufig als Erstes untersucht wurde.

6.1.3 Bearbeitungsaufwand

Der Aufwand, den ein Individuum für die Evaluation betrieben hat bzw. betreiben musste, wurde über die Bearbeitungszeit operationalisiert. Die Messung der Bearbeitungszeit für ein Programm begann, sobald es nach dem Starten betriebsbereit war bzw., bei bereits geöffnetem Programm, sobald der Proband dieses in den Vordergrund holte. Sie wurde beendet, sobald das Programm geschlossen, minimiert oder in den Hintergrund gelegt wurde

¹ Mehr als Hinweise können hier nicht abgeleitet werden, da die Anordnung der Programm-Icons auf dem Desktop (und damit die Bearbeitungsreihenfolge, s. voriger Absatz) nicht systematisch, sondern nur zufällig variiert wurde.

Bearbeitungs- position	SmartDraw		Flowcharter		Visio/DS	
1.	3 / 8	(0.38)	0 / 8	(0.00)	0 / 1	(0.00)
2.	0 / 3	(0.00)	3 / 8	(0.38)	4 / 6	(0.67)
3.	2 / 6	(0.33)	0 / 1	(0.00)	5 / 10	(0.50)
alle	5 / 17	(0.29)	3 / 17	(0.18)	9 / 17	(0.53)

Tabelle 6.3: Absolute und relative Auswahlhäufigkeiten nach Bearbeitungsposition (nur Probanden mit drei Optionen)

bzw. der Proband seine Aufmerksamkeit erkennbar abwandte, z. B. um den Fragebogen auszufüllen. Tabelle 6.4 gibt einen Überblick über durchschnittliche Bearbeitungszeiten nach verschiedenen Kriterien.

Bearbeitungszeit in s für ...	MW	Med	Min	Max	SD
... alle Programme*	1489	1410	260	3260	835
... ein einzelnes Programm	521	420	110	1980	355
... das gewählte Programm	479	468	180	1010	228
... das erste Programm*	618	460	110	1980	485
... das zweite Programm*	386	360	125	930	229
... das dritte Programm*	485	405	230	1090	237
... SmartDraw	515	428	160	930	244
... Flowcharter	546	395	110	1980	505
... Visio/DS	505	430	170	1180	388

Tabelle 6.4: Statistik der Bearbeitungszeiten nach verschiedenen Kriterien (* nur Probanden mit drei Optionen; MW = Mittelwert, Med = Median, Min = Minimum, Max = Maximum, SD = Standardabweichung)

Die Daten zeigen eine enorme Spannweite hinsichtlich des betriebenen Aufwandes. So beträgt z. B. die maximale Bearbeitungszeit für ein einzelnes Programm mit 1980 s das 18-fache des entsprechenden Minimums. Die durchschnittlichen Zeiten für die einzelnen Optionen sind relativ ausgeglichen. Hinsichtlich der Bearbeitungsreihenfolge ist festzustellen, dass die durchschnittliche Bearbeitungszeit für das erste Programm mit Abstand am größten ist, gefolgt von der für das dritte bzw. zweite Programm. Der für die letztlich ausgewählte Option betriebene Aufwand liegt etwa 8 % und damit nicht auffällig unter dem Gesamtdurchschnitt für ein einzelnes Programm.

6.1.4 Erfahrung

Von den zwanzig Versuchspersonen wurden vier als gering (Versuchspersonen 2, 6, 14, 15), acht als unspezifisch (Versuchspersonen 3, 8, 11, 13, 16, 17, 19, 20) und acht als spezifisch (Versuchspersonen 1, 4, 5, 7, 9, 10, 12, 18) erfahren eingestuft. Abbildung 6.1 zeigt das Auswahlresultat im Zusammenhang mit der Erfahrung der Versuchspersonen (nur Probanden mit drei Optionen). Die Präferenzen der einzelnen Erfahrungsklassen entsprechen mit einer Ausnahme im Wesentlichen der Präferenzverteilung in der Gesamtstichprobe: Lediglich die gering Erfahrenen weichen davon ab, indem keiner dieser Probanden Flowcharter gewählt hat.

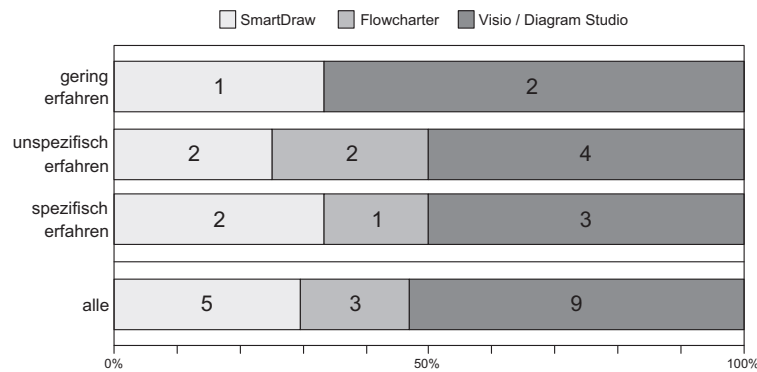


Abbildung 6.1: Absolute und relative Auswahlhäufigkeiten nach Erfahrung (nur Probanden mit drei Optionen)

Die Auswahlhäufigkeiten in Abhängigkeit von der Bearbeitungsreihenfolge und der Erfahrung, d. h. wie oft gering, unspezifisch bzw. spezifisch Erfahrene das als Erstes, Zweites bzw. Drittes ausprobierte Programm wählten, sind in Tabelle 6.5 dargestellt (nur Probanden mit drei Optionen). Dabei zeigt sich eine gegenüber der Gesamtstichprobe erhöhte Tendenz der spezifisch und unspezifisch Erfahrenen, nicht das als Erstes untersuchte Programm zu wählen.

Bearbeitungsposition	Erfahrung		
	gering	unspezifisch	spezifisch
1.	1	1	1
2.	1	3	3
3.	1	4	2

Tabelle 6.5: Auswahlhäufigkeiten nach Erfahrung und Bearbeitungsposition (nur Probanden mit drei Optionen)

Die auf die drei Erfahrungsgruppen aufgeschlüsselten durchschnittlichen Bearbeitungszeiten sind in Tabelle 6.6 dargestellt. Entgegen der intuitiven Erwartung weisen jedoch nicht die spezifisch, sondern die unspezifisch Erfahrenen die mit Abstand kürzeste durchschnittliche Bearbeitungszeit für ein einzelnes Programm auf. Dagegen fällt bei Ersteren

der Anstieg vom zweiten zum dritten Programm weniger deutlich als bei den anderen Gruppen aus.

Mittelwert der Bearbeitungszeit in s für ...	Erfahrung			
	gering	unspezif.	spezifisch	alle
... alle Programme*	1620	1298	1680	1489
... ein einzelnes Programm	611	433	574	521
... das gewählte Programm	695	346	504	479
... das erste Programm*	738	533	671	618
... das zweite Programm*	353	321	490	386
... das dritte Programm*	528	443	519	485
... SmartDraw	724	429	496	515
... Flowcharter	392	512	651	546
... Visio/DS	662	357	585	505

Tabelle 6.6: Durchschnittliche Bearbeitungszeiten nach Erfahrung und verschiedenen Kriterien (* nur Probanden mit drei Optionen)

6.2 Qualitative Analyse I: Evaluationsstrategie

6.2.1 Ansatz: Empirisch begründete Typenbildung

Bei der Frage nach der Evaluationsstrategie (Herangehensweise/Bearbeitungsstil) geht es darum, was die Versuchspersonen getan haben, d. h. welche Interaktionen in welchem Zusammenhang und Umfang ausgeführt wurden, um sich ein Urteil über die angebotenen Programme zu bilden. Von besonderem Interesse ist dabei zunächst weniger die isolierte Analyse von Einzelfällen, als vielmehr der Versuch, zu einer Typologie von Evaluationsstrategien zu gelangen (vgl. Abschnitt 5.1). Sinn und Ziel einer solchen Typologie liegen vor allem darin, mögliche Zusammenhänge zwischen dem Prozess und dem Ergebnis der Softwareevaluation und -auswahl zu untersuchen, also die Frage, ob Individuen, die sich ähnlich verhalten, auch dieselbe Entscheidung treffen. Wenn es gelingt, in der Realität vorkommendes Verhalten durch Charakterisierung einiger weniger Typen zu beschreiben, wäre es in einem nächsten Schritt besser als auf der Basis vieler einzelner Fälle möglich, einerseits überprüfbare Hypothesen, andererseits Empfehlungen für die Gestaltung von Software abzuleiten, um unter Ausnutzung des Wissens darüber, wie sie evaluiert wird, deren wahrgenommene Attraktivität und damit letztlich ihre Akzeptanz zu erhöhen.

Eine *Typologie* ist nach Kluge (1999) „das Ergebnis eines Gruppierungsprozesses, bei dem ein Objektbereich anhand eines oder mehrerer Merkmale in Gruppen bzw. Typen eingeteilt wird, so daß sich die Elemente innerhalb eines Typus möglichst ähnlich sind (*interne Homogenität*) und sich die Typen voneinander möglichst stark unterscheiden (*externe*

Heterogenität). Mit dem Begriff *Typus* werden die gebildeten Teil- oder Untergruppen bezeichnet, die gemeinsame Eigenschaften aufweisen und anhand der spezifischen Konstellation dieser Eigenschaften beschrieben und charakterisiert werden können“ (S. 26f, Hervorhebungen im Original). Durch einen Typus soll also das Charakteristische (Typische) einer Menge von Objekten zum Ausdruck gebracht werden.

Dabei sind Typen jedoch von Klassen zu unterscheiden, welche klar definierter und fester Grenzen bedürfen, wohingegen Typen durch unscharfe Grenzen und fließende Übergänge gekennzeichnet sind (Kluge, 1999). Entsprechend geht es bei einer Typologie nicht darum, identische, sondern vielmehr „nur“ ähnliche Elemente zusammenzufassen, mit der Konsequenz, dass Typen die Realität nie exakt abbilden, sondern immer mehr oder weniger große Abweichungen bestehen. Es ist wichtig, sich dessen bei der weiteren Verwendung einer Typologie bewusst zu sein, weil sonst die Gefahr besteht, die herausragenden Charakteristika eines Typs unzulässig zu verallgemeinern, sodass das Augenmerk nur noch auf diese Aspekte gerichtet wird und die empirische Vielfalt in den Hintergrund gerät (ebd.).

In der Literatur werden vor allem empirische (*Realtypen*) und heuristische Typologien (*Idealtypen*) unterschieden (vgl., auch zum Folgenden, Kluge, 1999). Erstere orientieren sich an der empirischen Verteilung der Untersuchungselemente und werden induktiv aus den vorliegenden Daten ermittelt, wogegen Letztere vorrangig aus theoretischen Überlegungen abgeleitet werden und Resultate von Isolierung und Überspitzung bestimmter Aspekte konkreter Phänomene, insofern Abstraktionen des real Existierenden sind. Da die hier zu erarbeitende Typologie auf dem empirischen Datenmaterial basieren und das tatsächliche Verhalten von Individuen widerspiegeln soll, beschränken sich die weiteren Ausführungen auf Realtypen.

Für das Vorgehen bei der empirisch begründeten Typenbildung schlägt Kluge (1999) ein Stufenmodell vor, an dem sich die hier vorgenommene Auswertung grob orientiert. Demnach sind als Erstes relevante Merkmalsdimensionen zu erarbeiten, die den sog. *Merkmalsraum* bilden und anhand derer charakteristische Ähnlichkeiten und Unterschiede zwischen den zu gruppierenden Elementen erfasst werden können.

Im zweiten Schritt werden dann die Elemente anhand ihrer entsprechenden Merkmalsausprägungen miteinander verglichen und gruppiert. Die Gruppierung kann *agglomerativ* erfolgen, indem ausgehend von den Einzelfällen ähnliche Elemente zusammengefasst werden, oder *divisiv*, indem die Gesamtgruppe schrittweise jeweils anhand eines Merkmals immer weiter in Teilgruppen zerlegt wird. Zum zweiten Schritt gehört außerdem die Analyse der so ermittelten Gruppen auf empirische Regelmäßigkeiten um zu ermitteln, wie stark die einzelnen Felder des Merkmalsraumes besetzt sind. Bei Zellen, die schwach oder gar nicht besetzt sind, ist zu klären, ob sie in der Realität nicht vorkommen oder ob andere Gründe vorliegen. Gegebenenfalls kann der Merkmalsraum entsprechend reduziert werden.

Der dritte Schritt sieht eine Analyse der Sinnzusammenhänge vor. Dabei wird unterstellt, dass die angetroffenen Merkmalskombinationen nicht zufällig, sondern Ausdruck tiefer liegender Strukturen sind. Ziel dieses Schrittes soll sein, sowohl die einzelnen Typen als auch die gesamte Typologie besser zu verstehen. Falls nötig, nämlich wenn die ver-

wendeten Dimensionen nicht für einen sinnvollen Vergleich ausreichen, kann zum ersten Schritt zurückgekehrt werden, um den Merkmalsraum zu erweitern.

Als vierter und letzter Schritt sollen die einzelnen Typen anhand ihrer Merkmalskombinationen und der gefundenen Sinnzusammenhänge umfassend charakterisiert werden. Dabei ist insbesondere darauf zu achten, dass zwar einerseits das Gemeinsame der Elemente eines Typus hervorgehoben wird, andererseits aber nicht der Aspekt verloren geht, dass diese sich zwar ähneln, aber nicht identisch sind. Besondere Sorgfalt wird auch bei der Vergabe von Kurzbezeichnungen für die einzelnen Typen angemahnt, „da es hierbei schnell zu Verkürzungen oder gar Verzerrungen kommen kann, die der Komplexität der untersuchten Sachverhalte nicht gerecht werden“ (Kluge, 1999, S. 280).

6.2.2 Vorläufige Typen

Die Konstruktion des Merkmalsraums erfolgte in der vorliegenden Untersuchung mittels einer Vorgehensweise, die *Substruktion* genannt wird (Lazarsfeld & Barton, 1951). Die Grundidee dieses Verfahrens liegt darin, dass die Merkmale, auf denen eine Typologie beruht, auch ausgehend von bereits gegebenen Typen ermittelbar sein müssen. Den Ausgangspunkt der Analyse bilden dabei vorläufige, intuitiv gebildete Typen als Ergebnis pragmatischer Reduktionen, d. h. Zusammenfassungen von Merkmalskombinationen, die die Übersicht über das Datenmaterial erleichtern. Zu ermitteln ist dann ein Merkmalsraum, aus dem diese intuitive Typologie hergeleitet werden kann. Lazarsfeld und Barton (1951) schreiben dazu:

Often it might be much better for the student to become deeply acquainted with his material [...], and then bring order into it by first blocking out a few main types on a completely impressionistic basis. After doing so, he can reconsider the matter and substruct to his own typological intuitions an adequate attribute space [...] The best results, probably will be gained by using this combination of an initial general survey and a subsequent systematic analysis. (S. 179)

Das Verfahren der Substruktion erscheint hier besonders passend und angemessen, weil das zu untersuchende Verhalten der Individuen noch völlig offen ist und sich deshalb zu dessen Beschreibung geeignete Merkmale nahezu zwangsläufig nur aus einer ersten Inspektion des Datenmaterials ergeben können. Obwohl es mehrerer Versuche mit verschiedenen Dimensionen bedurfte, bis ein akzeptables Ergebnis vorlag, wird die weitere Erarbeitung der Typologie aus Gründen der besseren Strukturierung als weitgehend zyklischer Prozess beschrieben und auf fruchtlose Varianten nicht weiter eingegangen.

Hauptbestandteil einer jeden Evaluation war aufgabengemäß das Erschließen der Funktionsweise sowie das Begutachten der konkreten Ergebnisse der zur Erstellung von Flussdiagrammen grundlegenden Funktionalitäten. Bereits während der Datenerhebung kristallisierten sich jedoch einige deutliche Unterschiede hinsichtlich der Herangehensweise heraus. Ein erstes, sehr offensichtliches Unterscheidungsmerkmal war der Umfang bzw.

die Realitätsnähe der im Zuge des Ausprobierens erzeugten Abbildungen. Dieses Merkmal ist sehr direkt beobachtbar und wird aufgrund seiner daraus resultierenden Validität als Hauptunterscheidungskriterium verwendet.

Einige Versuchspersonen zeichneten zum Teil recht umfangreiche und in sich komplexe Diagramme und machten dabei einen sehr zielstrebigem Eindruck, d. h. sie konzentrierten sich auf die absolut notwendigen Funktionalitäten und arbeiteten diese ohne große Umschweife ab (vTyp² I). Bedingt durch die größeren Diagramme schien die Exploration der Funktionalitäten in größerer Tiefe zu erfolgen, d. h. das Verhalten oder der Effekt z. B. eines Dialogelementes wurde nicht nur in einem, sondern in mehreren Zusammenhängen, unter verschiedenen Aspekten erprobt.

Andere Versuchspersonen hingegen begnügten sich damit, jedes benötigte Element nur einmal zu erzeugen (wenn überhaupt; oft wurde z. B. die Linienbeschriftung vergessen). Ein Teil auch dieser Gruppe wirkte im oben beschriebenen Sinne sehr zielstrebig, hinterließ dabei aber den Eindruck, die Evaluation mit möglichst wenig Aufwand hinter sich bringen zu wollen (vTyp II). Ihr Ansatz schien zu sein, dass ein komplettes Diagramm nichts anderes ist, als eine Zusammensetzung mehrerer Einzelelemente. Darin unterschieden sie sich deutlich von den Vertretern des vTyp I, die anscheinend gerade dieses Zusammensetzen als kritisch empfanden und prüfen wollten, wie sich die Software dabei verhält.

Ein weiterer Teil der Versuchspersonen mit wenig komplexen Abbildungen dagegen ließ diese Zielgerichtetheit vermissen und schien sich eher von der Software leiten zu lassen. Zwar trieben auch hier die zum Erzeugen eines Flussdiagramms benötigten Funktionalitäten das Geschehen an, jedoch schienen sich diese Probanden wenig um die semantische Struktur ihres Diagramms zu kümmern, solange sie nur möglichst viel herumprobieren konnten. Statt sich gezielt auf das zu beschränken, was für den aktuellen Zweck unmittelbar benötigt wird, neigten sie dazu, auch darüber hinausgehende Funktionalitäten zu explorieren (vTyp III).

Schließlich gab es einige Probanden, die vergleichsweise große Probleme hatten, die nötigen Funktionalitäten zu erschließen. Sie erzeugten ebenfalls nur rudimentäre Diagramme und neigten dazu, bereits entdeckte Funktionsweisen wieder zu vergessen. Diese Versuchspersonen waren bereit, ggf. umständlich zu improvisieren, wenn es auch nach wiederholten Anläufen nicht gelang, den gewünschten Effekt direkt zu erzeugen. In der Konsequenz waren sie auch mit optisch weniger ansprechenden Ergebnissen zufrieden, anscheinend froh, überhaupt etwas hinbekommen zu haben (vTyp IV).

Mit wenigen Ausnahmen (Vp3 und Vp13) ließen sich alle Versuchspersonen mit einiger Klarheit einem dieser vorläufigen Typen zuordnen, nämlich:

- vTyp I: Versuchspersonen 1, 4, 5, 12, 16, 20
- vTyp II: Versuchspersonen 8, 9, 10, 11, 19
- vTyp III: Versuchspersonen 6, 7, 17, 18
- vTyp IV: Versuchspersonen 2, 14, 15

² vTyp = vorläufiger Typus

Ausgehend von dieser intuitiven Typologie ergibt sich ein Merkmalsraum, in den die Versuchspersonen hinsichtlich der Komplexität ihrer Abbildungen sowie Art und Umfang der unternommenen Evaluationsschritte, insbesondere Exploration von Kern- und Extrafunktionalitäten, eingeordnet werden können. Dafür bedarf es jedoch einer Operationalisierung entsprechender Dimensionen, um die Verortung der einzelnen Individuen transparent und nachvollziehbar zu machen.

6.2.3 Verwendete Merkmale und Merkmalsausprägungen

6.2.3.1 Komplexität von Abbildungen

Der Begriff *Komplexität* wurde wegen seiner qualitativen Konnotationen bewusst gewählt, um das zu beschreibende Konzept vom eher quantitativ zu verstehenden *Umfang* abzugrenzen.³ Es ist sehr plausibel, anzunehmen, dass z. B. zehn zusammenhanglos erzeugten und anscheinend planlos auf einer Zeichenfläche verteilten Symbolen eine andere Evaluationsstrategie zugrunde liegt, als wenn ebendiese zehn Symbole sinnvoll zu einem Ablauf verbunden werden. Es geht beim Konstrukt der Komplexität einer Abbildung also darum, zu erfassen, in welchem Maße ein Proband bestrebt war, zum Zwecke der Evaluation ein konkretes, tendenziell realistisches Problem zu bearbeiten vs. sich weitgehend problemneutral oder abstrakt die Funktionsweise einer Software zu erschließen.

Die Ermittlung der Komplexität einer Abbildung erfolgte durch Vergabe von Punkten auf der Basis ihres komplexesten Zustandes, d. h. zu dem Zeitpunkt, zu dem die höchste Punktzahl erzielt wurde. Das bedeutet vor allem, dass ein späteres Reduzieren des Diagramms, z. B. um Funktionalitäten zum Entfernen von Elementen auszuprobieren, ohne mindernden Einfluss auf den Komplexitätswert bleibt. Da dem Verständnis von Komplexität hier der *Zusammenhang* der einzelnen Elemente (Symbole, Pfeile, Text) bzw. das Bestreben, ein realistisches Diagramm zu erzeugen, zugrunde liegt, wurden folgende Regeln für die Punktvergabe etabliert, deren Anwendung in Abbildung 6.2 anhand einiger Beispiele illustriert wird:

1. Für eine gerade Verbindung zwischen zwei Symbolen gibt es einen Punkt.
2. Für eine abgewinkelte Verbindung zwischen zwei Symbolen gibt es zwei Punkte.
3. Sind dieselben zwei Symbole mit mehreren Pfeilen verbunden, wird jeweils nur eine gerade sowie eine abgewinkelte Verbindung berücksichtigt, da das Anlegen von mehr als zwei Verbindungen (streng genommen sogar nur einer) zwischen denselben zwei Symbolen nicht sinnvoll und somit kein Ausdruck von Komplexität im hier zugrunde liegenden Sinne ist.
4. Für jeden verwendeten Beschriftungstyp (d. h. an einem Symbol bzw. einer Linie) gibt es jeweils einen Punkt, d. h. für das Beschriften eines Diagramms können insgesamt maximal zwei Punkte vergeben werden, egal wie viele Beschriftungen tatsächlich angebracht wurden.

³ „Komplexität [...]: Vielschichtigkeit; das Ineinander vieler Merkmale“ (Duden, 2003, S. 931)

5. Wurde ein fragmentiertes Diagramm erzeugt, d. h. bestehend aus mehreren, nicht miteinander verbundenen Teilen, wird nur das Fragment mit der höchsten Punktzahl berücksichtigt.

Die minimale Komplexität, die erreicht werden muss, um alle relevanten Aspekte der benötigten Kernfunktionalitäten an einem Diagramm auszuprobieren, beträgt entsprechend 5 (vgl. vorletztes Beispiel in Abb. 6.2: jeweils eine gerade (1) und eine abgewinkelte Verbindung (2) sowie jeweils mindestens eine Symbol- (1) und eine Linienbeschriftung (1)).

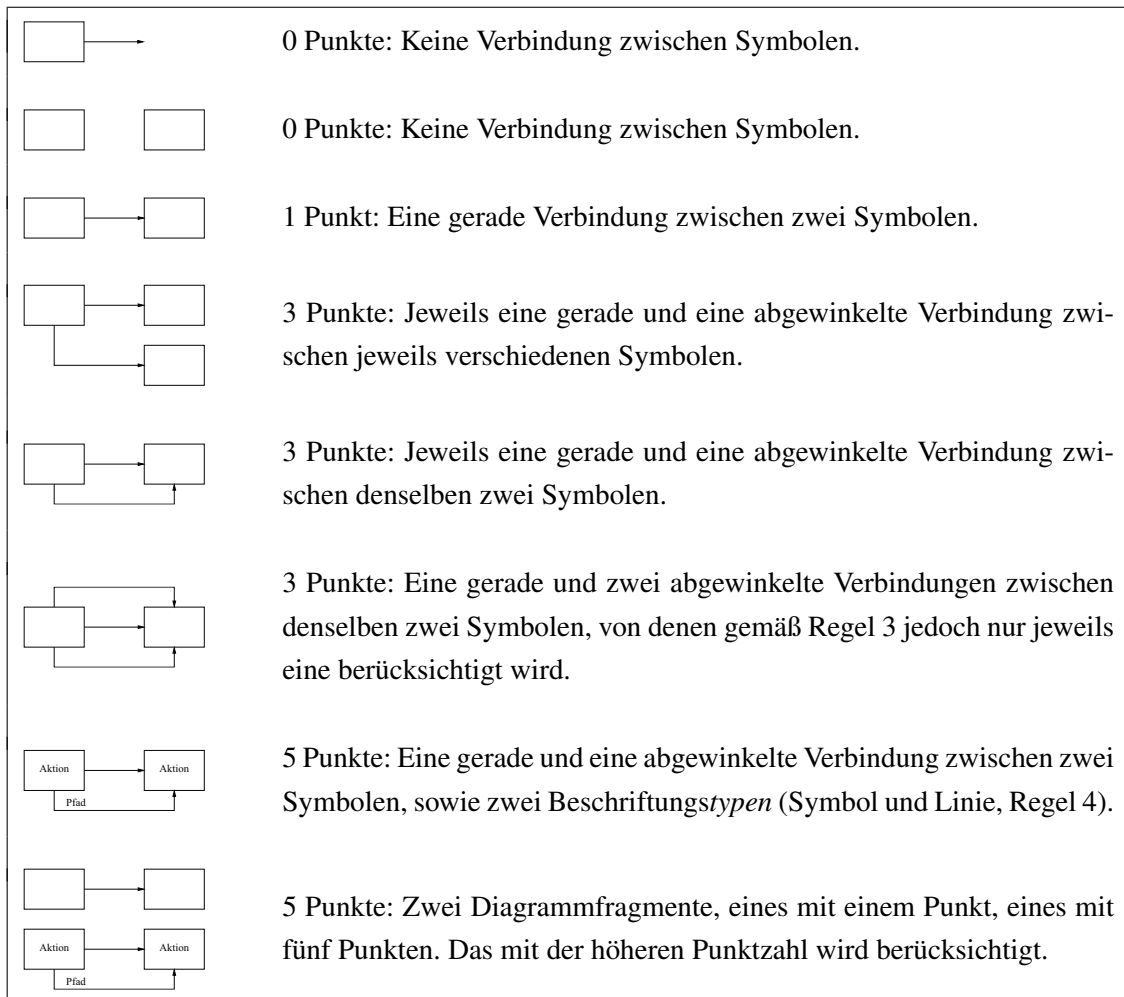


Abbildung 6.2: Bestimmung der Komplexität eines Diagramms (eigene Darstellung)

6.2.3.2 Kennzahlen

Die Merkmale, Aspekte, Auffälligkeiten, die zu der in Abschnitt 6.2.2 entwickelten vorläufigen Typologie geführt haben, dienen als Wegweiser für die Suche nach entsprechenden Operationalisierungen anhand der aufbereiteten Daten der Interaktionsprotokolle der

einzelnen Probanden (vgl. Abschnitt 5.4.2.3).⁴ Bei der Ermittlung einer jeden Kennzahl⁵ wird grundsätzlich so verfahren, dass diese zunächst für jedes evaluierte Programm einzeln bestimmt und dann deren Durchschnitt über alle evaluierten Programme berechnet wird. Bei rein additiven Kennzahlen (ϕK_{px} , ϕK) erfolgt die Durchschnittsbildung vor allem, um trotz unterschiedlicher Anzahl evaluierter Programme (zwei vs. drei) eine Vergleichbarkeit der Werte zu ermöglichen. Ein weiterer Grund besteht darin, dass es mit dem auf ein Programm normalisierten Wert leichter fällt, eine Vorstellung von der erzeugten Abbildung zu entwickeln. Für die relativen Kennzahlen ($\phi K'_{rel}$, X_{rel}) wäre eine allgemeine Begründung dieses Vorgehens sehr umständlich und erfolgt deshalb weiter unten am Beispiel von $\phi K'_{rel}$.

Durchschnittliche Komplexität eines Flussdiagramms ϕK_{px}

$$\phi K_{px} = \frac{\sum_{i=1}^j K_{px_i}}{j} \quad \text{mit } j = \text{Anzahl evaluierter Programme}$$

Die bei der Ermittlung dieser Kennzahl verwendete Größe K_{px_i} steht für die Komplexität der mit dem i -ten Programm erzeugten Abbildung und ergibt sich unmittelbar auf die in Abschnitt 6.2.3.1 dargelegte Weise. Deshalb bedarf die Bestimmung von ϕK_{px} über die Angabe der Berechnungsvorschrift hinaus keiner weiteren Erläuterung. ϕK_{px} kann als Maß dafür interpretiert werden, inwieweit ein Individuum die Erprobung der Software als Simulation eines realistischen Problems angeht. Sie dient vor allem der Abgrenzung von vTyp I.

Durchschnittliche Anzahl der eingeholten Informationen über Kernfunktionalitäten ϕK

$$\phi K = \frac{\sum_{i=1}^j K_i}{j} \quad \text{mit } j = \text{Anzahl evaluierter Programme}$$

Die Größe K_i steht für die Anzahl der über die Kernfunktionalitäten des i -ten Programms eingeholten Informationen. Zur Ermittlung dieses Wertes werden in den IPB-Zeilen S_E bis M jeweils alle Felder mit einer nicht unterstrichenen Zeitangabe ausgezählt, d. h. jene, die sich auf Erstexplorationen beziehen. Farblich markierte Felder *ohne* Zeitangabe bedeuten lediglich nachträgliche Wertungen zu Aspekten, die bereits untersucht oder nur passiv wahrgenommen wurden, d. h. ohne die jeweilige Information aktiv erheben zu wollen. Für das *Verhalten* eines Individuums erscheint aber nur die Berücksichtigung dessen sinnvoll, was aktiv nachgefragt wurde. Felder mit einer *unterstrichenen* Zeitangabe beziehen sich auf wiederholte Explorationen, d. h. die in solchen Evaluationsschritten gesuchte

⁴ Grundsätzlich kann aus diesen Daten eine nahezu unüberschaubare Anzahl unterschiedlichster Werte abgeleitet werden. Hier werden nur die erläutert, welche gemäß der vorläufigen Typologie für die weitere Auswertung relevant sind. Jene hingegen, welche im Zuge einer Sondierung der Daten außerdem ermittelt wurden, sich aber als fruchtlos erwiesen haben (z. B. Rahmenexplorationen und Orientierungsphasen; vgl. Abschnitt 5.4.2.2), werden hier nicht diskutiert, um die Ausführungen nicht unnötig aufzublähen.

⁵ Im Folgenden wird, solange noch die reine Ermittlung numerischer Werte im Vordergrund steht, bevorzugt von Kennzahlen statt von Merkmalen gesprochen.

Information wird ebenfalls in einem früheren Feld repräsentiert und ist somit in der Auszählung bereits berücksichtigt.

ϕK kann als Maß dafür interpretiert werden, wie gründlich ein Individuum vorgeht bzw. inwieweit es versucht, möglichst viele Eventualitäten durch Ermittlung entsprechender Informationen abzusichern. Die Rolle dieser Kennzahl bei der Typenbildung ist noch unklar; sie wird wegen ihrer aus dem Sachzusammenhang vermuteten Bedeutung eingeführt, die darin begründet ist, dass die Kernfunktionalitäten im Mittelpunkt der Evaluation stehen oder zumindest stehen sollten.

Durchschnittliches Verhältnis von wiederholt zu insgesamt eingeholten Informationen über Kernfunktionalitäten $\phi K'_{rel}$

$$\phi K'_{rel} = \frac{\sum_{i=1}^j \frac{K'_i}{K_i}}{j} \quad \text{mit } j = \text{Anzahl evaluierter Programme}$$

K'_i steht für die Anzahl der über die Kernfunktionalitäten des i -ten Programms *wiederholt* nachgefragten Informationen. Zur Ermittlung dieses Wertes sind im Interaktionsprotokoll jeweils die Felder mit unterstrichenen, d. h. wiederholte Explorations repräsentierenden Zeitangaben auszuzählen. Die K'_i eines jeden Programms werden dann mit dem entsprechenden Wert für K_i ins Verhältnis gesetzt. Die Modellierung als Verhältniszahl ist nötig, weil nur so eine Vergleichbarkeit möglich wird: Zwei Wiederholungen bei auch insgesamt nur zwei ermittelten Informationen haben eine ganz andere Bedeutung als gegenüber 15. Anschließend wird der Durchschnitt dieses Verhältnisses für alle Programme eines Probanden berechnet.

Statt der einzelnen Verhältnisbildungen mit anschließender Durchschnittsermittlung wäre es auch möglich, die Summe der wiederholt ermittelten Informationen über alle Programme durch die Summe der insgesamt ermittelten Informationen über alle Programme zu dividieren. Dies würde jedoch dem Sachverhalt nicht gerecht, da die bei einem Programm nötig gewordenen Wiederholungen nur mit der Implementation der entsprechenden Funktionalitäten bei genau *diesem* Programm, nicht aber bei den anderen, im Zusammenhang stehen. Deshalb wäre es nicht korrekt, die Wiederholungen über alle Programme zu addieren.

Ein einzelnes Verhältnis von K'_i zu K_i ist insofern nicht nur durch den Probanden, sondern auch durch das Programm bestimmt. Erst die Menge aller Einzelverhältnisse, ausgedrückt z. B. durch deren Durchschnitt, kann als Merkmal des Evaluationsstils eines Individuums interpretiert werden, nämlich als Maß dafür, wie sicher ein Individuum beim Explorieren einer Software ist. Je größer dieses Verhältnis, desto häufiger mussten neue Anläufe unternommen werden, um zu ermitteln, wie bestimmte Funktionalitäten umgesetzt sind. $\phi K'_{rel}$ dient in erster Linie zur Unterscheidung zwischen vTyp II und IV.

Durchschnittliches Verhältnis von Informationen über Extrafunktionalitäten zu jenen über Kernfunktionalitäten ϕX_{rel}

$$\phi X_{rel} = \frac{\sum_{i=1}^j \frac{X_i}{K_i}}{j} \quad \text{mit } j = \text{Anzahl evaluierter Programme}$$

X_i steht für die Anzahl der über die Extrafunktionalitäten des i -ten Programms eingeholten Informationen. Die Kennzahl ϕX_{rel} wird analog zu dem Verfahren für $\phi K'_{rel}$ ermittelt. Für X_i werden wie bei K_i nur Erstexplorationen berücksichtigt. Auch hier ist die Verhältnisbildung der Verwendung der absoluten Anzahl der Informationen über Extrafunktionalitäten vorzuziehen, um eine Vergleichbarkeit der erhaltenen Werte zu ermöglichen. ϕX_{rel} kann interpretiert werden als das Maß, in dem ein Individuum auch nicht benötigte Funktionalitäten exploriert, also z. B. auf Angebote der Software eingeht oder Möglichkeiten zur individuellen Gestaltung sucht. Diese Kennzahl wird zur Abgrenzung des vTyp III verwendet.

Die für die einzelnen Versuchspersonen ermittelten Werte für die genannten Kennzahlen sind in Abbildung 6.3 als Diagramm dargestellt. Zwecks besserer Lesbarkeit und wegen abweichender Skalierung werden die additiven (ϕK_{px} , ϕK) bzw. relativen Kennzahlen (ϕX_{rel} , $\phi K'_{rel}$) auf jeweils unterschiedlichen Achsen abgetragen.

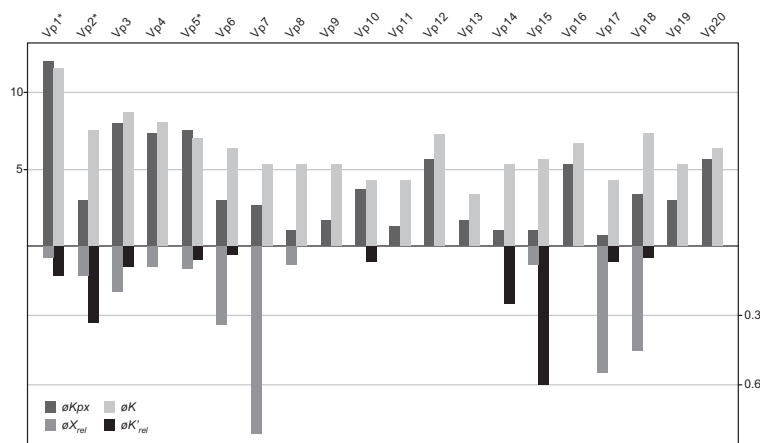


Abbildung 6.3: Diagramm der Kennzahlenwerte für alle Probanden (* Versuchsperson mit nur zwei Optionen)

6.2.3.3 Dichotomisierung der Merkmale

Um den Prozess der Typenbildung hinsichtlich der möglichen Merkmalskombinationen handhabbar zu gestalten, wurden die Dimensionen dichotomisiert, d. h. auf die Ausprägungen „hoch“ und „niedrig“ beschränkt. Somit war für jede Dimension zu entscheiden, an welchem Wert eine entsprechende Grenze gezogen werden sollte. Für ϕK_{px} wurde der Wert 5.0 gewählt, weil dies der Mindestkomplexität eines Diagramms, das alle laut

Problemstellung benötigten Elemente enthält, entspricht (vgl. Abschnitt 6.2.3.1). Entsprechend gilt ein Diagramm mit einer Komplexität von ≥ 5.0 als hoch komplex, eines mit einer Komplexität von < 5.0 als niedrig komplex. Dabei mag es als unpassend erscheinen, bereits eine Zeichnung mit der für eine vollständige Evaluation erforderlichen Mindestkomplexität als hoch komplex zu bezeichnen. Es ist jedoch zu bedenken, dass zum einen der Mittelwert aller Probanden sogar nur 3.93 beträgt (Median = 3.0), und es zum anderen nicht vordergründig um die Beurteilung von Diagrammen geht, sondern um die Typisierung des dazu führenden Verhaltens. Im Zusammenhang mit dieser Kennzahl geht es dabei vor allem um das Bestreben, alle relevanten Elemente eines Flussdiagramms im Zusammenhang auszuprobieren.

Da mit der durchschnittlichen Komplexität aller drei im Zuge der Evaluation erstellten Diagramme gearbeitet wird, kann zwar einerseits ein einzelnes Diagramm auch noch weniger komplex sein, dafür müssen aber die anderen umso höhere Werte aufweisen, um dennoch auf den geforderten Durchschnittswert zu kommen. Bei einer solchen Konstellation ist es plausibel, anzunehmen, dass das Individuum grundsätzlich vorhatte, mehr als nur rudimentäre Diagramme zu zeichnen, bei dem einen „Ausrutscher“ nach unten jedoch durch irgendetwas davon abgehalten wurde. Auch wenn konstant immer „nur“ ein Diagramm mit genau der Mindestkomplexität gezeichnet wurde, kann eine gewisse Sorgfalt unterstellt werden, die eine Einordnung als hoch komplex rechtfertigt.

Auch für ϕK kann eine Grenze theoretisch abgeleitet werden. Analog zu ϕK_{px} wird das Kriterium der Vollständigkeit zugrunde gelegt. Demnach müssen mindestens sechs Informationen über Kernfunktionalitäten eingeholt werden, nämlich: Erzeugen eines Symbols, Beschriften eines Symbols, Erzeugen einer geraden Verbindung, Erzeugen einer abgewinkelten Verbindung, Beschriften einer Verbindung, Manipulieren eines Elementes. Entsprechend gilt eine durchschnittliche Anzahl explorierter Aspekte von Kernfunktionalitäten von ≥ 6.0 als „hoch“, ein Wert von < 6.0 als „niedrig“. Die bereits für ϕK_{px} geführte Argumentation hinsichtlich der Einordnung bereits des Mindestwertes als „hoch“ gilt hier analog.

Für die relativen Merkmale ($\phi K'_{rel}$, ϕX_{rel}) muss die Festlegung der Grenze zwischen „hoher“ und „niedriger“ Ausprägung auf der Basis des Datenmaterials erfolgen: Es gilt, z. B. für den Fall von ϕX_{rel} zu entscheiden, ab welchem Verhältnis von Extra- zu Kernfunktionalitäten ein Individuum als nicht besonders auf die eigentliche Aufgabe fokussiert, sondern mehr softwaregeleitet im Sinne des vTyp III gilt. Ein willkürlich festgelegter Wert ist dabei kaum zu begründen. Eine Ausnahme würde eventuell die Wahl eines Grenzwertes von 0.0 darstellen, jedoch ist dieser für den vorgesehenen Zweck nicht sinnvoll, da bereits die Exploration einer einzigen Extrafunktionalität im Verlauf der gesamten Evaluation zu einer „hohen“ Merkmalsausprägung führen würde.

Zur Lösung dieses Problems wurde die Verteilung der Kennzahl ϕX_{rel} über die Versuchspersonen hinsichtlich auffälliger Sprünge untersucht. Abbildung 6.4 zeigt ein Diagramm der aufsteigend sortierten Werte von ϕX_{rel} , zusammen mit der jeweiligen Änderung zum Vorgänger ($\Delta \phi X_{rel}$) sowie den entsprechenden Ausprägungen von ϕK_{px} . Es ist gut zu erkennen, dass die Werte für ϕX_{rel} (im Diagramm mittelgrau) zunächst auf niedrigem Niveau nahezu konstant sind und dann recht gleichmäßig ansteigen. Den ersten als leicht erhöht zu bezeichnenden Wert weist Vp3 auf. Allerdings fällt die Differenz

zwischen Vp3 und ihrem Vorgänger deutlich geringer aus als die zu ihrem Nachfolger. Darüber hinaus ist zu berücksichtigen, dass Vp3 bei ϕK_{px} (dunkelgrau), der zentralen Dimension (vgl. Abschnitt 6.2.2), einen mehr als doppelt so hohen Wert aufweist als alle anderen Versuchspersonen mit erhöhtem ϕX_{rel} . Insofern erscheint es sinnvoll und plausibel, den Grenzwert für die Unterscheidung zwischen „hohem“ und „niedrigem“ ϕX_{rel} zwischen den Werten von Vp3 und Vp6 festzulegen.

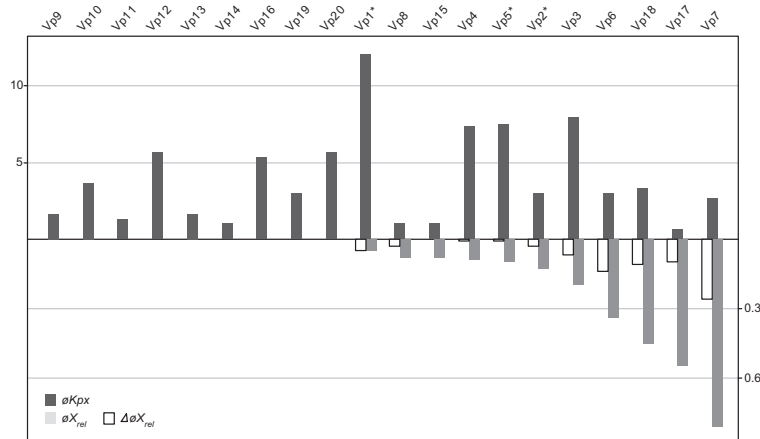


Abbildung 6.4: Diagramm der Werte für ϕX_{rel} und ϕK_{px} (* Versuchsperson mit nur zwei Optionen)

Zur Festlegung des Grenzwertes für $\phi K'_{rel}$ wurde aus den prinzipiell gleichen Gründen analog vorgegangen. Das entsprechende Diagramm ist in Abbildung 6.5 dargestellt. Wie ϕX_{rel} , zeigt auch $\phi K'_{rel}$ (im Diagramm schwarz) über den größten Teil der Versuchspersonen hinweg einen zunächst sehr flachen Verlauf, um dann zügig anzusteigen. Der erste größere Sprung tritt zwischen Vp1 und Vp14 auf. Auch hier können die korrespondierenden Ausprägungen von ϕK_{px} (dunkelgrau) zur Entscheidungsunterstützung herangezogen werden: Alle Probanden ab Vp14 weisen zum Teil extrem niedrige Komplexitätswerte auf, die darüber hinaus geradezu dramatisch kleiner sind als die der beiden unmittelbaren Vorgänger, Vp1 und Vp3. Aus diesen Gründen wird die Grenze für die Unterscheidung zwischen „hohem“ und „niedrigem“ $\phi K'_{rel}$ zwischen den Werten von Vp1 und Vp14 festgelegt.

6.2.4 Typenbildung

6.2.4.1 Besetzte Merkmalskombinationen

Nach Ersetzung der stetigen Kennzahlenwerte durch die entsprechenden dichotomen Ausprägungen „niedrig“ und „hoch“ spannen die vier Dimensionen ϕK_{px} , ϕK , $\phi K'_{rel}$ und ϕX_{rel} einen Merkmalsraum mit $2 \times 2 \times 2 \times 2 = 16$ Zellen auf. Kluge (1999) schlägt vor, die Verteilung der Versuchspersonen auf diese Zellen entsprechend ihrer jeweiligen Ausprägungen durch eine verschachtelte Tabellenstruktur darzustellen. Allerdings ist zu berücksichtigen, dass hier nur 6 der 16 Zellen des Merkmalsraums tatsächlich besetzt sind,

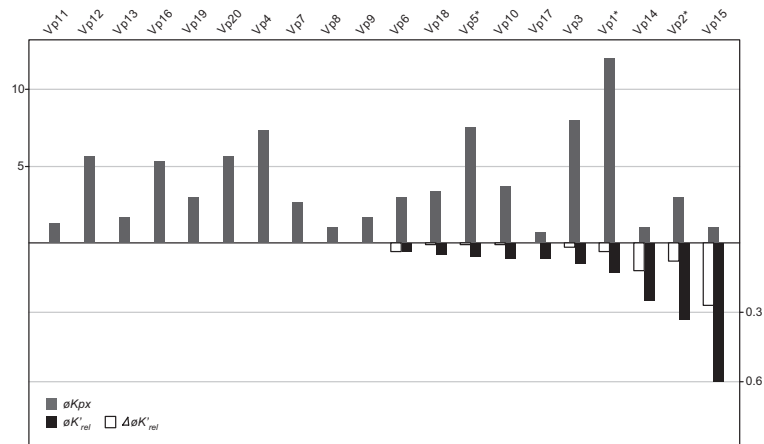


Abbildung 6.5: Diagramm der Werte für $\phi K'_{rel}$ und ϕK_{px} (* Versuchsperson mit nur zwei Optionen)

d. h. dass in der Stichprobe nur die in Tabelle 6.7 angegebenen Kombinationen von hohen und niedrigen Merkmalsausprägungen auftreten. Weil also die meisten Zellen ohnehin unbesetzt bleiben (z. B. die Zelle für den Fall, dass alle Merkmale hoch ausgeprägt sind), erscheint eine Darstellung der Verteilung als Diagramm (Abb. 6.6) geeigneter.

besetzte Zelle	Ausprägung von				Anzahl Probanden
	ϕK_{px}	ϕK	$\phi K'_{rel}$	ϕX_{rel}	
1	hoch	hoch	niedrig	niedrig	7
2	niedrig	niedrig	niedrig	niedrig	6
3	niedrig	niedrig	niedrig	hoch	2
4	niedrig	hoch	niedrig	hoch	2
5	niedrig	niedrig	hoch	niedrig	2
6	niedrig	hoch	hoch	niedrig	1

Tabelle 6.7: Besetzte Merkmalskombinationen bzw. Zellen des Merkmalsraums

Es ist gut zu erkennen, dass ϕK wider Erwarten nur eine untergeordnete Rolle bei der Unterscheidung der Typen zu spielen scheint. Diese Kennzahl ist hoch mit ϕK_{px} korreliert ($r = 0.86$)⁶ und weicht in seiner dichotomen Ausprägung nur in drei Fällen vom entsprechenden Wert für ϕK_{px} ab. Insofern leistet es kaum einen darüber hinausgehenden Beitrag zur Gruppierung der Fälle. Die hohe Korrelation erscheint durchaus plausibel: Je komplexer das gezeichnete Diagramm, umso häufiger und tendenziell variabler müssen die einzelnen Funktionalitäten genutzt werden. Dies führt dazu, dass mehr Informa-

⁶ Pearson-Korrelation, signifikant bei $p = 0.001$ (t-Test). Hinsichtlich der Korrelationen der stetigen Ausprägungen der anderen Merkmale täuscht das Diagramm in Abb. 6.6: Diese liegen zwischen -0.18 und 0.10 und sind nicht signifikant bei $p = 0.05$.

tionen über die entsprechenden Funktionalitäten erhoben werden. Alternativ kann auch argumentiert werden, dass das Bestreben, die benötigten Funktionalitäten möglichst umfassend zu explorieren, zu einem immer weiteren Anwachsen der Zeichnung führt, weil viele verschiedene Diagrammkonstellationen ausprobiert werden sollen.

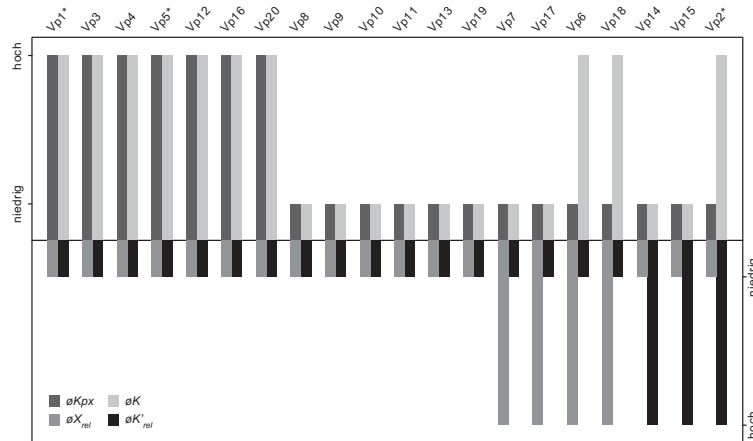


Abbildung 6.6: Diagramm der Werte für alle Merkmale in dichotomer Ausprägung (* Versuchsperson mit nur zwei Optionen). Hervorgehoben ist die typbezogene Zusammengehörigkeit der Versuchspersonen.

Außerdem kennzeichnend für die untergeordnete Rolle und entscheidend für die weitere Behandlung von ϕK ist die relativ geringe Streuung der Kennzahlenwerte. Der Variationskoeffizient⁷ beträgt hier nur 29 %, im Gegensatz zu 75 % für ϕK_{px} , 151 % für ϕX_{rel} sowie 179 % für $\phi K'_{rel}$. Für die Typenbildung sollten jedoch vor allem Merkmale mit möglichst großer Varianz verwendet werden (Kluge, 1999). Um die Typologie nicht mit (Unter-)Typen, für die nur jeweils ein oder zwei Vertreter vorhanden sind, aufzublähen⁸, werden unter Vernachlässigung der Ausprägungen von ϕK die Zellen der Fälle 3 und 4 sowie 5 und 6 zu jeweils einer zusammengefasst.

Infolgedessen verbleiben vier (Haupt-)Typen, die der vorläufigen Typologie (vgl. Abschnitt 6.2.2) entsprechen und durch ihre Ausprägungen auf drei Merkmalen (ϕK_{px} , ϕX_{rel} , $\phi K'_{rel}$) voneinander abgegrenzt werden können. Sie werden in den folgenden Abschnitten näher charakterisiert und bezeichnet als

- Typ I (die *Aufgabenorientierten*),
- Typ II (die *Minimalisten*),
- Typ III (die *Verspielten*),

⁷ Der Variationskoeffizient stellt eine Normierung der Varianz dar. Er ist das Verhältnis von Standardabweichung zu Mittelwert und wird in der Regel in Prozent angegeben.

⁸ „Da mit Hilfe einer Typologie ein u. U. recht komplexer Sachverhalt strukturiert werden soll, ist es wichtig, daß man zu einer überschaubaren Anzahl von Gruppen gelangt. Unterschiede sollten also ignoriert werden, wenn sie nicht relevant sind“ (Kluge, 1999, S. 102). Eine untergeordnete Relevanz von ϕK wird hier durch die relativ geringe Varianz der Werte nahe gelegt.

- Typ IV (die *Unsicheren*).

6.2.4.2 Typ I – Die Aufgabenorientierten

Typ I (Versuchspersonen 1, 3, 4, 5, 12, 16 und 20) weist hohe Ausprägungen bei ϕK_{px} , und niedrige bei $\phi K'_{rel}$ sowie ϕX_{rel} auf. Das Verhalten der Individuen dieses Typs kann als Ausdruck eines hohen Aufgaben- und Zielbewusstseins interpretiert werden, d. h. sie wollen sichergehen, dass sie die mit der ausgewählten Software zu bearbeitenden Probleme tatsächlich werden lösen können („Gut, und ich will noch so'n viereckiges Ding machen, damit ich wieder dieses ... mit den ... wie man so'ne Rückkopplung eingeben kann, üben kann.“, Vp20.96). Deshalb investieren sie entsprechenden Aufwand in die Evaluation: Typ I-Probanden erproben die Kombination mehrerer Basishandlungen zur Bearbeitung eines größeren Problems und recherchieren das Verhalten der Kernfunktionalitäten hinsichtlich des möglichen Eintritts von Spezialfällen:

- v_{p1.61} Gut, dann werd ich jetzt noch eine Verbindung machen, und dann werd ich mal probier'n, wie man die, die verwinkelten Pfeile, ob man das ganz gut hinbekommt.
[...]
- 153 Gut, ich probier's jetzt noch mal anders rum.
[...]
- 226 Hm, gut also, jetzt scheint's vielleicht doch zu funktionieren, also probier ich's hier noch mal, ehm ...
[...]
- 327 Ja, das funktioniert auch,
328 also probier ich das hier noch mal.
[...]
- 376 Obwohl, ich probier jetzt einfach mal, was da Probleme gemacht hatte, einfach diesen Pfeil, der mehrere Verzweigungen hat hier [...]

Je nachdem, wie leicht oder mühsam die nötigen Explorationen von der Hand gehen, nimmt die Evaluation unterschiedlich viel Zeit in Anspruch. Sie wird mit der nötigen Gründlichkeit und Ausdauer betrieben und auch bei andauernden Schwierigkeiten erst spät aufgegeben (besonders deutlich z. B. bei den langwierigen Flowcharter-Evaluationen von Vp3 und Vp4). Das Treffen einer „guten“ Entscheidung hat Priorität.

Die Tiefe der Exploration, verbunden mit dem Bemühen um einen harmonischen Gesamteindruck der Probezeichnung, führt zu einer überdurchschnittlichen Realitätsnähe und Komplexität der erzeugten Diagramme (vgl. Abb. 6.7). Typ I-Probanden achten darauf, dass Beschriftungen in die Symbole hineinpassen, dass die Symbole korrekt zueinander ausgerichtet sind, auf saubere Verbindungen und Linienverläufe usw.:

- v_{p16.36} Stift ansetzen ... na, das ist ein bisschen komisch. Der macht diese Linie nicht gerade.
Die ist immer ein bisschen verzerrt.

37 *beendet Linie* → *Hinweis; liest*

38 Da frag ich mich, warum er das jetzt nicht gerade macht. Hm, die Linie wird schief.
[...]

63 Nee, den krieg ich nicht gerade! Hm, na gut. Das ist natürlich dumm, dann wird das alles ein bisschen schief und krumm.

64 Den hier oben könnte ich bestimmt noch ein bisschen neu machen.

65 *löscht den ersten Pfeil und zieht neu*

66 Nee, wird nicht gerade. Ich denk mal, dass das vielleicht an der kurzen Strecke liegt. Nochmal ... ich krieg ihn nicht gerade.

67 Nee, keine Ahnung. Mach ich zu. «Änderungen speichern ... Nein.»

Vp16.200 Man muss auch achten darauf, dass man die Elemente untereinander setzt, damit die Pfeile auch gerade werden. Wenn die beiden Punkte nicht genau gegen ... untereinander sind, dann wird der Pfeil nicht gerade.

Vp5.142 Ist natürlich sehr schön, dass er die Schrift automatisch verkleinert, aber nicht genug, es nutzt nichts. Muss das Ganze dann trotzdem noch größer ziehen.

Vp5.156 Hier muss ich die dann vielleicht auch noch verknüpfen, oder ist das jetzt schon verknüpft gewesen? Hier ist verknüpft, es ist verknüpft. Es ist verknüpft, es liegt aber irgendwie nicht auf der Mitte. Und ich werde es hier wahrscheinlich nie hinkriegen, dass dieser Pfeil jetzt gerade verläuft.

157 Also das ist einfach schlecht. Das ist [?] einfach schlecht.

Das Vorgehen dieser Individuen ist zielstrebig und fokussiert, d. h. sie lassen sich nicht von unnötigen Spielereien (sog. Gimmicks) ablenken, die eine Software neben den Kernfunktionalitäten anbietet. Durch diese fehlende Beschäftigung mit Unwesentlichem haben sie es auch kaum nötig, bereits schon einmal Erschlossenes nochmals explizit zu verifizieren. Da sie praktisch nichts anderes tun, als dieselben Kernfunktionalitäten in kurzen Abständen immer wieder zu benutzen, haben sie praktisch gar keine Chance, deren prinzipielle Funktionsweise wieder zu vergessen. Da bei den Vertretern des Typs I die Bewältigung des aktuell anstehenden Problems im Vordergrund steht, werden sie die *Aufgabenorientierten* genannt.

6.2.4.3 Typ II – Die Minimalisten

Typ II (Versuchspersonen 8, 9, 10, 11, 13 und 19) hat auf allen drei Merkmalen niedrige Ausprägungen. Hinsichtlich ihrer Zielstrebigkeit und Fokussiertheit ähneln Individuen dieses Typs den Aufgabenorientierten, jedoch opfern sie die Tiefe bzw. Gründlichkeit der Evaluation dem Bestreben, den Aufwand dafür möglichst gering zu halten: Alle sechs

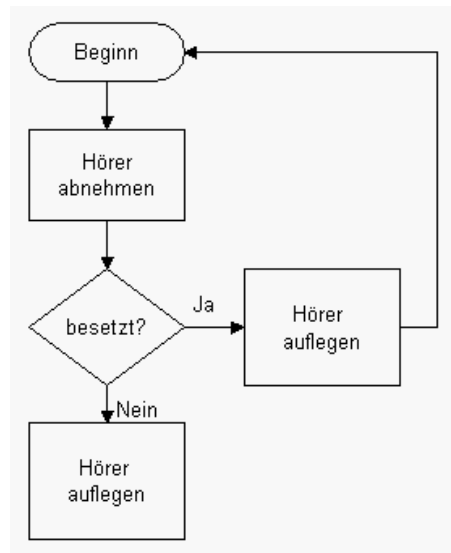


Abbildung 6.7: Typische Probezeichnung eines Typ I-Probanden (Vp12 mit Flowcharter)

Vertreter dieses Typs befinden sich unter den acht Probanden mit der kürzesten durchschnittlichen Bearbeitungszeit für ein Programm.

Sie explorieren zwar, wie die nötigen Funktionalitäten prinzipiell abgerufen werden, verzichten aber darauf, sie in einem größeren Zusammenhang zu erproben. Entsprechend erzeugen sie nur minimale Abbildungen (vgl. Abb. 6.8), ggf. auch nur ein einzelnes Symbol mit einem einzigen, ins Leere führenden Pfeil. Sie verschaffen sich insofern lediglich einen flüchtigen Eindruck davon, wie es sein wird, die Software auf ein realistisches Problem anzuwenden und verlassen sich darauf, dass es dann genauso klappen wird. In diesem Punkt unterscheiden sie sich deutlich von den Aufgabenorientierten, welche bestrebt sind, auf „Nummer Sicher“ zu gehen:

v_{p9.17} VL: *Ja, na ja, wie gesagt, Du musst ja nachher damit klarkommen ...*

18 Ja, es sieht erst mal nicht verkehrt aus. Also hier rechts ...

19 VL: ... weil, wenn Du nachher was nicht hinkriegst, ist das doof.

20 Ooch, da bin ich zuversichtlich. Ich glaub das krieg ich ... also das sieht erst mal nicht schlecht aus [...]

v_{p11.14} [kurz nach dem Start von Flowcharter, bis dahin ohne Interaktion] Okay. Also, das ist ja schon mal so ähnlich, wie, wie wir das bei Word kennen.

15 Unten sind jetzt aber nicht diese ... Hier kann man auch diese Verbindungen ... «Weitere Symbole – Symbolbibliothek – Symbole hinzufügen/entfernen» ... naa, das bringt jetzt nichts, hier.

16 Ahh, hier haben wir die Pfeile ... die Pfeile ...

- 17 Hier, das heißt bestimmt Schreiben, ja. Das geht bestimmt genauso ... wie geht das?
[probiert erfolgreich Text im freien Raum aus] Ja, okay.
- 18 Und wenn ich das jetzt nicht mehr will? *[markiert und entfernt Text]* Ah, okay.
- 19 Hier kann man bestimmt Zahlen ... Was brauchen wir noch? Wir brauchen die Symbole, wir brauchen die Zahlen, brauchen die Pfeile, hmm, ja, das ist ja schon mal ganz geschmeidig. Ich glaube, hiermit würde ich das hinkriegen.
- 20 Dann kuck ich mir mal das andere an.
- 21 *minimiert Flowcharter, startet SmartDraw*

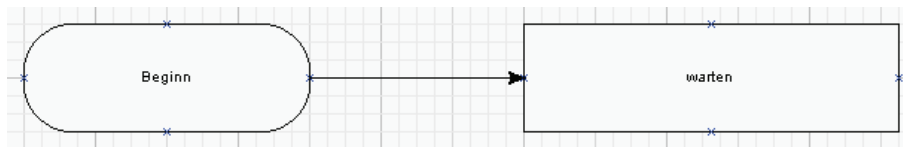


Abbildung 6.8: Typische Probezeichnung eines Typ II-Probanden (Vp9 mit Diagram Studio)

Bei den Vertretern des Typs II steht die Effizienz der Evaluation im Vordergrund: Sie entscheiden sich schnell, ob sie ein Programm mögen, und neigen bei Missfallen oder Schwierigkeiten dazu, noch weniger als das absolut Notwendige zu explorieren und einfach abzubrechen. Auch sonst sichern sie im Wesentlichen nur ab, dass sie mit einer Software nicht völlig scheitern. Das Aussehen der Probeabbildung ist nebensächlich. Deshalb werden sie die *Minimalisten* genannt. Im Folgenden ist exemplarisch für dieses minimalistische Vorgehen die komplette Evaluation von Diagram Studio durch Vp9 wiedergegeben (zugehörige Zeichnung in Abb. 6.8):

Vp9.49 *beendet Flowcharter, startet Diagram Studio*

- 50 «Gadwin». Das kenn ich nicht. Oha! «View Tutorial». Nee!
- 51 Ehm, «Block Diagram» ... «Flowchart». Flowchart woll'n wir machen, ehm, und wir, wir nehmen mal ... «Top-Down Diagrams, Information-Tracking Diagrams» ... nehmen wir mal ein «Basic Flowchart», genau.
- 52 *wählt dies aus → Arbeitsfläche*
- 53 Oha, hier ist ja einiges an Optionen ... «Pan and Zoom», «Get Started» ...
- 54 «Basic Flowchart Symbols» ... na, das ist abgekuckt von Visio, das sieht man ja gleich! Auch alles da, mhm, fangen wir mal an mit den Formen.
- 55 *klickt Symbol und versucht erfolglos, es frei zu zeichnen*
- 56 Huh? Jetzt gib mir doch mal 'nen Prozess! Muss ich den hier rüberziehen? Ah, rüberziehen, sehr schön.
- 57 Das ist ja'n dreister Visio-Klon, muss ich ja mal sagen!

- 58 So ... ach, nee, das war was anderes. Wo ist jetzt der «Terminator»? Den ziehen wir jetzt rüber. Das ist «Beginn» ...
- 59 *klickt und beschriftet Symbol*
- 60 Das ist ja schön, hier ist ja gleich das Raster mit drin, das find ich ja viel besser!
- 61 So, hin und her ziehen kann man das auch ... die Box ist vielleicht ein bisschen groß geraten, aber wir testen ja bloß. «Warten» schreiben wir jetzt hier rein [*beschriftet Symbol*]
- 62 So, jetzt müssen wir das verbinden ... «Arrows and Connectors» ... eh, «Line Connector», muss ich den jetzt hier rüber ziehen? Schwupp. Blopp.
- 63 *tut dies und verbindet mit Symbolen*
- 64 Ey, mach den mal hier rein. Genau.
- 65 Ja, das ist ja einfach. Dit is einfach!
- 66 So, jetzt wollen wir noch Text so drinne haben. «Backgrounds» ... «Text-Tool», das ist das passende, denk ich mal. Mm, schreiben wir mal «ja» ... Mensch! Passt.
- 67 Brauchen wir sonst noch was? Von den Formen her, denke ich mal, ist auch alles da, was wir brauchen. «Decision» ... Was is'n [?] komisch hier? [*Maus rollt nicht gleichmäßig*] «Curve Connector», Trapez haben wir, «Decision» hatten wir auch irgendwo? Genau. «Process» haben wir ... ja, haben wir auch alles. Der kann auch alles. Jo, können wir den nächsten probieren.
- 68 *beendet Diagram Studio, startet SmartDraw → Demo-Dialog*

6.2.4.4 Typ III – Die Verspielten

Typ III (Versuchspersonen 6, 7, 17 und 18) hat niedrige Ausprägungen bei ϕK_{px} und $\phi K'_{rel}$ sowie hohe bei ϕX_{rel} . Individuen dieses Typs teilen die Zuversicht der Minimalisten, dass im großen Maßstab alles genauso funktionieren wird wie im kleinen, d. h. sie halten es nicht für nötig, die Software an einem etwas größeren, zusammenhängenden Diagramm zu erproben. Ihr Verhalten wird allerdings im Gegensatz zu diesen nicht durch ein Streben nach Effizienz getrieben: Nur ein Proband aus dieser Gruppe hat eine sehr kurze Bearbeitungszeit (Vp17), zwei liegen nahe beim Mittelwert aller Probanden und einer stellt mit 1087 s sogar die längste durchschnittliche Bearbeitungszeit für ein Programm (Vp18).

Was die Vertreter des Typs III auszeichnet, ist das Explorieren von Extrafunktionalitäten. Sie „spielen“ mit der Software und erfreuen sich an Gimmicks und individuellen Gestaltungsmöglichkeiten. Dies tun sie sowohl aus eigenem Antrieb, z. B. weil sie bestimmte optische Effekte erzielen oder sich die Arbeit erleichtern wollen, als auch „fremdmotiviert“, wenn sie beim ziellosen Browsen oder auf der Suche nach etwas ganz anderen zufällig darauf stoßen:

Vp7.136 zieht weiteren, sehr breiten umrandeten Pfeil auf die Zeichenfläche

137 Bumm. Mit Schatteneffekt. Gibt's hier eigentlich 'ne Vorschau? Die sehen hier immer so cool aus, aber bei mir nicht.

[...]

150 Aber, was's denn hier mit der Startseite, wo ist die denn? Das sah doch viel besser aus als das, was ich hier hab. Das will ich auch so bunt haben.

151 *sucht in der Menüleiste*

152 Ich werd mal noch so Farben reinkippen hier, das mit dem Eimer machen, blupp.

[...]

182 So, jetzt wird hier mal eingefärbt. Schön mit Pastell, mmm. Und drum herum die Linie, kann ich die dicker machen? So, und, ja, so dick. Und in welcher Farbe? «Linienfarbe», orange, dunkelgelb ... okay. Noch'n dicken Schatteneffekt, und jetzt wird das ja richtig stylermäßig. Ja, so könnte mir das gefallen. Hier gibt's auch noch'n Schatten ... Schatten, kein Schatten ... so so.

183 Gib mir hier mal so'n Bezier ... Baiser ... das muss ja keinen Sinn machen hier ... ich steck dich hier rein ... oh, das sieht ja hässlich aus. Mhm, so, flex ... schön dick ... und in welcher Farbe wirst du sein? Lila ... ho ho!

184 Ja, das ist ja cool, damit kann man ja was Schickes machen, das gefällt mir gut.

Vp6.21 *öffnet Kontextmenü am Symbol und klickt «Text Layout» → Dialog*

22 Wieso? Hier ist doch gar kein Text. «Sekundäre Textbereiche»? Oops. Was ist das?

23 *klickt «Hinzufügen – links» und schließt Dialog; klickt ins Symbol, links im Prozesssymbol wird ein Teilbereich sichtbar*

24 Ahh, kuck mal! Schreibt er jetzt senkrecht?

25 *gibt Text ein, dieser erscheint waagrecht*

26 Kann man dich auch senkrecht schreiben lassen?

Hinsichtlich der Tiefe des Erschließens werden Extra- jedoch ähnlich oberflächlich wie Kernfunktionalitäten behandelt, d. h. es wird lediglich versucht, punktuell herauszufinden, was sie auf welche Weise tun, statt sie zum Erzeugen eines Gesamtwerkes einzusetzen. Entsprechend rudimentär und ungeordnet, durch viele ausprobierte Effekte teilweise geradezu chaotisch sehen ihre Testabbildungen aus (vgl. Abb. 6.9).

Obwohl es auch Typ III-Probanden aufgabengemäß darum geht, wie man mit einem bestimmten Programm Flussdiagramme erzeugt, wollen sie außerdem herausfinden, was man noch alles damit tun kann. Insofern beschränken sie sich nicht in dem Maße wie Aufgabenorientierte und Minimalisten auf den einen, engen, vom aktuellen Problem vorgegebenen Funktionalitätsbereich, sondern lassen sich in gewisser Weise treiben bzw. den Evaluationspfad durch das Angebot der Software bestimmen. Sie lassen sich von Detailproblemen ablenken, für die sie Lösungen suchen, auch wenn sie nicht unmittelbar relevant für die Aufgabe sind. Ob das sichtbare Ergebnis ihres Tuns sinnvoll ist, ist ne-

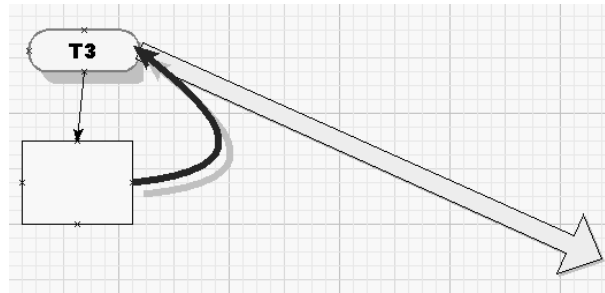


Abbildung 6.9: Typische Probezeichnung eines Typ III-Probanden (Vp7 mit Diagram Studio)

bensächlich („... das muss ja keinen Sinn machen hier ...“, Vp7.183). Deshalb werden die Vertreter dieses Typs die *Verspielten* genannt.

6.2.4.5 Typ IV – Die Unsicheren

Typ IV (Versuchspersonen 2, 14 und 15) hat niedrige Ausprägungen bei ϕK_{px} und ϕX_{rel} sowie hohe bei $\phi K'_{rel}$. Individuen dieses Typs erscheinen bei flüchtiger Betrachtung wie Minimalisten, weil sie nur rudimentäre Abbildungen erzeugen und sich inhaltlich auf das Notwendige beschränken (vgl. Abb. 6.9 und 6.10). Anders als diese tun sie dies aber nicht aus einem Streben nach Effizienz (alle drei Vertreter dieses Typs befinden sich in der langsameren Hälfte der Stichprobe), sondern weil sie sich schwer tun, mit der Software zurechtzukommen. Sie sind sehr wohl bereit, die Evaluation mit erhöhtem Aufwand zu betreiben, welcher sich in langem Suchen und ggf. Entwickeln umständlicher Behelfslösungen (sog. Workarounds) entfaltet:

v_{p14.52} Ehm ... «Freie Linie oder Figur ohne Unterbrechung mit dem Stift zeichnen» ... Wie krieg ich denn da'n Pfeil ran? Na ja gut, ist auch egal. Ehm ... «Bogen» ... wer braucht'n Bogen? «Mehrere Objekte ... Ein Objekt» ... ehm, tja ... da!

53 *hat ein Dreieck-Symbol gefunden, klickt → Hinweisfenster*

54 «Mit SmartDraw ...» aha, aha, aha.

55 *schließt Hinweis und erzeugt Dreieck am Linienende*

56 Hm, na toll. Und jetzt können wir das bestimmt noch schwarz machen ... irgendwo ... hier irgendwo, bestimmt. «Tabellen» nicht ... Ausfüllen, Füllfarbe ... na, muss auch nicht. Der Pfeil muss ja nicht schwarz sein.

57 *VL: Na, eigentlich schon.*

58 Ich hab mich da nicht so.

59 *Schaltflächen*

60 Ehm, das ist, wenn das so wie Word ist, müsste das ja hier oben sein, ne? «Fett ... kursiv ... unterstrichen» ... ist'n komisches Kursiv-Zeichen hier ... «unterstrichen ...

subskript ... plus – minus ... Tabelle ... Zeichnen ... Grafik ... Wordart ... Clipart ... Gleichungen ... Worddokument».

61 *öffnet Kontextmenü am Dreieck*

62 «Farbe einfügen»! Ja! [*wählt schwarze Füllung*] Toll! Na, das ist doch super.

Vp15.43 Also, ich würd jetzt hier improvisieren, ne. Ich würd jetzt so'n Dreieck nehmen und das jetzt hier anfügen, ne, wenn ich jetzt unbedingt 'nen Pfeil malen muss oder so. Ne.

44 *tut dies*

45 Ja. Würd ich jetzt so machen. So. Und dann vielleicht irgendwie noch mal raus und dann, [*Kontextmenü*] eh, «Farbe einfügen» und dann schwarz oder so. Also, jetzt, so. Ne. Das wär jetzt meine Idee. So, und dann würd ich weiterkommen.

46 Dann würde man ... das ist, das geht hier, also das ist schon okay.

Eine Folge des langen, weitschweifigen, nicht unbedingt zielgerichteten Suchens ist, dass Typ IV-Individuen häufig nicht mehr nachvollziehen können oder wieder vergessen, wie sie einen bestimmten Effekt erreicht haben, und erneut auf die Suche gehen. Auf diese Weise, und weil sie öfter als andere Probanden Suchepisoden abbrechen und später wieder aufnehmen, kommen die vergleichsweise hohen Werte für $\phi K'_{rel}$ zustande:

Vp2.302 So, warte mal, wenn ich jetzt wieder was rein schreiben möchte,

303 würde ich, würde ich, würde ich ...

304 *sucht in den Schaltflächen*

305 Wie hab ich'n das gemacht?

306 Jetzt hab ich das schon wieder vergessen.

Vp14.132 *kann jetzt ohne weiteres Text auf der Linie eingeben*

133 Hahh, das ist auch nicht schlecht, ne? Wenn er die Linie unterbricht, das gefällt mir sogar noch viel besser.

134 Wie hab ich das jetzt gemacht? Hm! Jetzt hat er den sogar gedreht?! Hab ich irgendwas gemacht eben?

Vp14.194 *zieht zufällig eine Verbindung von Symbol zu Symbol*

195 Häh? Was hab ich denn jetzt anderes gemacht? Was hab ich denn eben gemacht? Hm, ist ja auch egal, ne? Das ist auf jeden Fall Schwachsinn.

[...]

207 «Einfügen». Ich weiß es nicht. Also, ich weiß auch nicht, was ich hier die ganze Zeit mache. «Rückgängig».

- v_{p15.189} Aber ich will ja jetzt eigentlich nur mal einen Text einfügen, ne, und und ... «Format ... Bearbeiten ... Einfügen»? Von was? Nee. «Bearbeiten ... Seite einrichten ... Seitenansicht ... Drucken ... Bearbeiten ... Rückgängig ... Ansicht» ... nee ... «Einfügen ... Feld ... Kommentar»? [*klickt*] Ist das immer ein Kommentar? Nee, nee, nee, nee. Kann mir nicht vorstellen, dass es das ist.
- 190 Wie war das denn noch, wie hab ich das denn gemacht? Jörg? [*VL*] Hm? «Hintergrund» ...
- 191 *doppelklickt Symbol*
- 192 Aha, Doppelklick, hm. Und dann? Ach, so war das, genau. Doppelklick, rechte Maus, und dann kann ich alles verändern. Gut.
- v_{p15.194} *versucht Linie zu ziehen → verschiebt Symbol*
- 195 Häh? Nee, was mach ich jetzt hier für Faxen? Hm? Noch mal. [*erfolglos*] Verwirrt mich ja jetzt hier schon wieder so. «Keine Linienenden»? Den anklicken ... «Einfügen»? Nee. Was ist das denn?
- 196 Wie hab ich denn diesen Pfeil hier gemalt? Ich meine, ich könnte auch den kopieren und hierhin setzen, und dann ... Das ist aber ganz schön umständlich, weil ich dann immer drehen müsste und so. Irgendwie ist das ...

Da diese Probanden nur bedingt die Kontrolle über die Software erlangen und somit der Abruf jeder einzelnen Funktionalität ein mühsames Unterfangen ist, bleiben ihre Probenbezeichnungen klein (vgl. Abb. 6.10). Die Frage nach komplexeren Diagrammen und dem Erproben der Funktionalitäten im größeren Zusammenhang stellt sich ihnen insofern nicht. Sie haben wichtigere Probleme, nämlich die gewünschten Wirkungen *überhaupt* hervorrufen zu können:

- v_{p2.148} Kann ich das auch verschieben?
- 149 *versucht erfolglos, Textfeld zu verschieben*
- 150 Nee.
- 151 Was ist'n das jetzt?
- 152 Kann ich das noch?
- 153 Nee, warte, da war's doch jetzt, da war's eben, wo ist es?
- 154 Jetzt ist es weg.
- 155 *klickt erfolglos auf der Suche nach Verschiebegriffen*
- 156 Jetzt ist es wieder weg.
- 157 Da!
- 158 Immer, wenn ich das mache ...
- 159 Nee.

160 Wie kann ich denn das jetzt verschieben?

161 Gar nicht. Gar nicht, wahrscheinlich.

162 Das ist doof, das gefällt mir auch nicht

163 Ich weiß nicht, wie ich das jetzt verschieben kann.

Vp14.64 *zieht weitere Linie*

65 So, ehm, Pfeil ab. Er zeigt mir das nur beim ersten Mal, wenn ich's das erste Mal benutze, wahrscheinlich. Weil ich ja dann schon gelernt hab.

66 Ich hätte ja auch den ganzen Pfeil kopieren können, ne.

67 *versucht per Drag&Drop zu kopieren*

68 Nee, o Gott, bloß nicht. Nichts ändern!

[...] [erzeugt einen zweiten Pfeil auf dieselbe umständliche Weise wie oben]

99 Okay, das hat tierisch lange gedauert.

Vp14.267 *erzeugt anscheinend unbeabsichtigt einige nur aus den Pfeilspitzen bestehende Linien*

268 Nee. Hähh? Ich hab hier lauter Pfeile. Wo kommen die denn alle her?

Vp15.200 *versucht erfolglos, Symbol frei zu zeichnen, dann per Einfügen → zuletzt kopierter Pfeil*

201 Ey, was ist das denn hier? Was ...? Hab ich jetzt hier immer einen Dreher drin mit?

Jetzt macht er mir'n Pfeil, wo ich keinen brauche ... «Zeichenblatt» ... Weiß ich nicht.

Das ist schon ein bisschen ...

Im Gegensatz zu den Typen I bis III entspringt das Charakteristische des Verhaltens von Typ IV-Individuen also nicht direkt ihrer Intention, sondern wird anscheinend durch das ungünstige Zusammentreffen ihrer individuellen Fähigkeiten und Kenntnisse einerseits sowie der spezifischen Gegebenheiten von Problem und Software andererseits determiniert. Insofern befinden sie sich sozusagen „auf fremdem Terrain“ und müssen sich eher vorsichtig tastend als selbstbewusst fordernd bewegen. Deshalb werden die Vertreter dieses Typs die *Unsicheren* genannt.

6.2.5 Evaluationstyp und Auswahlergebnis sowie Erfahrung

Über das Wissen um die reine Existenz verschiedener Evaluationstypen hinaus ist die Untersuchung eines Zusammenhangs zwischen Typus und Auswahlergebnis von Interesse: Falls ein systematischer Einfluss des Evaluationsverhaltens auf das Auswahlergebnis existiert, wäre dies ein wichtiger Faktor bei der Gestaltung und zielgruppenorientierten Diversifikation von Software. Darüber hinaus soll nach Hinweisen auf Zusammenhänge zwischen Typus und einschlägiger Erfahrung eines Individuums gesucht werden, weil darin ein Ansatz für die theoretische Erklärung und damit ein tieferes Verständnis der Typologie liegen könnte.

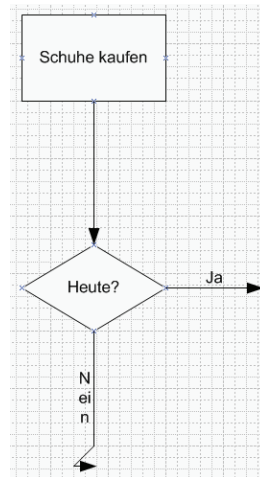


Abbildung 6.10: Typische Probezeichnung eines Typ IV-Probanden (Vp14 mit Visio)

Hinsichtlich des Auswahlresultates ergibt sich die in Abbildung 6.11 dargestellte Verteilung. Im Gegensatz zu den Auswahlhäufigkeiten nach der einschlägigen Erfahrung der Probanden (vgl. Abschnitt 6.1.4) weichen die Verteilungen für die einzelnen Evaluationstypen deutlich von der Gesamtverteilung ab. Auffällig ist insbesondere, dass lediglich bei den Aufgabenorientierten (Typ I) alle Programme vertreten sind. Dagegen hat keiner der Minimalisten (Typ II) bzw. Unsicheren (Typ IV) Flowcharter gewählt, und keiner der Verspielten (Typ III) entschied sich für SmartDraw. Bemerkenswert ist darüber hinaus, dass es Visio/DS bei den Aufgabenorientierten nicht gelingt, seine sonstige Vormachtstellung einzunehmen. Lediglich einer von fünf Probanden in dieser Gruppe entschied sich für dieses Programm.

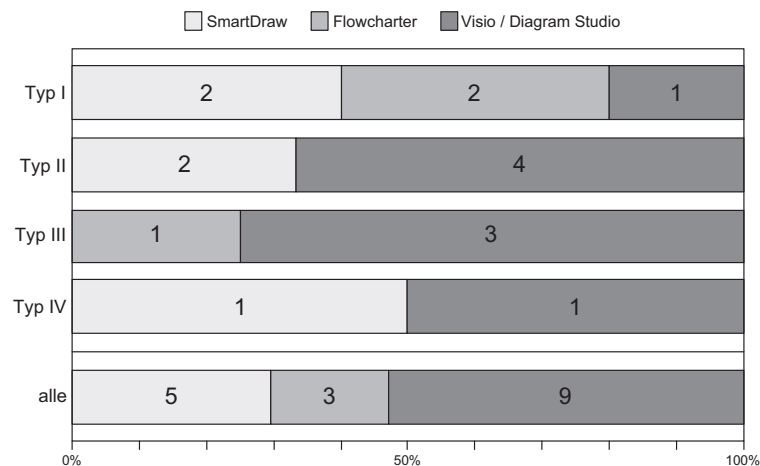


Abbildung 6.11: Absolute und relative Auswahlhäufigkeiten nach Evaluationstyp (nur Probanden mit drei Optionen)

Die Verteilung der Versuchspersonen hinsichtlich ihrer einschlägigen Erfahrung und ihres Evaluationstypus ist in Abbildung 6.12 dargestellt. Das auffälligste, wenn auch nicht besonders überraschende Ergebnis dieser Analyse ist, dass sich die Unsicheren ausschließlich aus Individuen mit geringer Erfahrung rekrutieren bzw. diese sich mit nur einer Ausnahme entsprechend Typ IV verhalten. Die Gesamtverteilung der Probanden auf die Erfahrungsklassen wird am ehesten durch Typ III repräsentiert, der als einziger Individuen aus jeder Gruppe enthält. Gemäß Typ I und II verhalten sich dagegen ausschließlich spezifisch und unspezifisch Erfahrene.

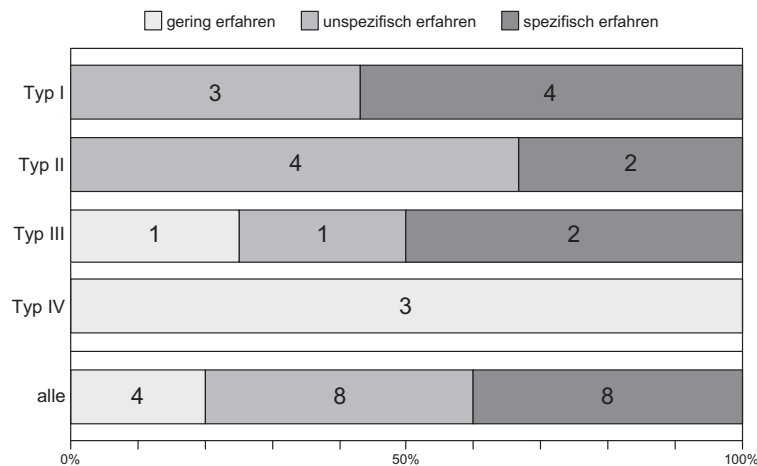


Abbildung 6.12: Absolute und relative Verteilung der Probanden nach Erfahrung und Evaluationstyp

Versuchspersonen mit spezifischer Erfahrung evaluieren vorwiegend aufgabenorientiert, nämlich in 50 % der Fälle; die übrigen Fälle verteilen sich gleichmäßig auf Typ II und III. Unspezifisch Erfahrene verhalten sich in 50 % der Fälle minimalistisch, im Übrigen eher aufgabenorientiert als verspielt (3 vs. 1 Probanden). Aus umgekehrter Sicht hat mit zwei Dritteln der überwiegende Teil der Minimalisten unspezifische Erfahrung, wogegen bei den Aufgabenorientierten das Verhältnis zwischen spezifisch und unspezifisch Erfahrenen ausgewogen ist (4 vs. 3 Probanden).

6.3 Qualitative Analyse II: Gründe für die Entscheidung

6.3.1 Ansatz: Heuristische Beobachtung

In diesem Teil der Analyse geht es darum, aus der Beobachtung der Versuchspersonen (genauer: ihrer Interaktionen mit und ihrer Kommentare zu den Programmen) während der Softwareevaluation auf Gründe für ihre jeweiligen Entscheidungen zu schließen. Da nicht von konkreten Hypothesen hinsichtlich dieser Gründe ausgegangen wird, erscheint es angebracht, im Sinne *heuristischer Beobachtung* vorzugehen. Dieser Begriff wurde bereits in Abschnitt 5.2.1 eingeführt, um den dieser Studie insgesamt zugrunde liegenden

Forschungsansatz zu charakterisieren, und wird an dieser Stelle wieder aufgegriffen und vertieft. Heuristische Beobachtung ist als *Haltung* gegenüber dem Untersuchungsgegenstand zu verstehen und folgt dem Gedanken, diesen möglichst unvoreingenommen und ganzheitlich zu erfassen und zu versuchen, aus dem beobachteten Ganzen Erkenntnisse zu gewinnen (vgl., auch zum Folgenden, Bortz & Döring, 2006; Greve & Wentura, 1997). Es geht darum, Strukturen und interessante Zusammenhänge zu entdecken und nach und nach „das Muster, das sich hinter dem individuellen Durcheinander verbirgt“ (Greve & Wentura, 1997, S. 21), zu erkennen.

Bortz und Döring (2006) nennen dieses Herangehen *freie* Beobachtung und erläutern: „Sie kommt vor allem für Untersuchungen in Betracht, mit denen ein bisher weitgehend unerforschtes Gebiet erkundet werden soll“ (S. 269). Heuristisch oder frei heißt diese Beobachtungsform deshalb, weil sie sich nicht auf einen einzigen, aus einer Theorie hergeleiteten Aspekt eines Geschehens oder Sachverhaltes beschränkt: „Das Beobachtungsprotokoll sollte eine möglichst umfassende Dokumentation von ganzen Ereignisabläufen und von interessant erscheinenden Einzelheiten [...] enthalten“ (ebd., S. 269).

Konzentration auf bestimmte Details dagegen ist das Merkmal der *deduktiven* Beobachtung als Methode der Datenerhebung. Der deduktive Beobachter ist ausdrücklich gehalten, nur auf einen oder wenige bestimmte, absichtlich und systematisch ausgewählte Aspekte der Wirklichkeit zu achten und alles andere zu ignorieren. Das Ziel besteht dabei darin, Hypothesen durch ein Experiment zu überprüfen, indem unabhängige Variablen manipuliert und abhängige Variablen gemessen werden, in diesem Fall durch die Methode der Beobachtung. Dafür muss aber klar sein, welche Variablen von Bedeutung sind oder sein sollen.

Diese Klarheit besteht in der vorliegenden Untersuchung nicht. Bekannt ist lediglich die abhängige Variable, nämlich das Auswahlergebnis. Das Ziel dieses Teils der Analyse besteht entsprechend darin, Kandidaten für unabhängige Variablen überhaupt erst zu entdecken. Die für jeden Probanden zu beantwortende Frage lautet demnach: Warum hat sich dieses Individuum genau für das gewählte und gegen die anderen Programme entschieden? Es geht also um die Identifikation der für die Entscheidung ausschlaggebenden Aspekte der Softwareevaluation. Ob bzw. in welchem Maße ein Aspekt jedoch ausschlaggebend war, kann nur durch heuristische Beobachtung, d. h. durch Berücksichtigung des Gesamtzusammenhanges entschieden werden, nicht aber durch punktuelle Erhebung eines einzelnen, ex-ante geeignet erscheinenden (sich ex post aber eventuell als nicht ausreichend oder gar irrelevant herausstellenden) Konstruktes.

Ein Beispiel soll dies verdeutlichen: Ein Proband äußert sich bei der Begutachtung eines der drei Programme mehrfach negativ, z. B. über die irreführenden Bezeichnungen von irgendwelchen Dialogelementen, und gibt insgesamt vielleicht sogar mehr negative als positive Kommentare. Trotzdem entscheidet er sich am Ende für genau dieses Programm, obwohl er vielleicht bei den anderen sogar weniger kritische Äußerungen abgegeben hat. Eine isolierte Betrachtung oder gar lediglich Auszählung positiver und negativer Kommentare kann in diesem Fall keine Erklärung für die Entscheidung liefern. Nur die Berücksichtigung des Gesamtzusammenhanges bringt z. B. zutage, dass sich die negativen Bemerkungen zum gewählten Programm auf tolerierbare Kleinigkeiten bezogen, welche durch positives Erleben anderer, als zentral und wichtig empfundener Aspekte

mehr als aufgewogen wurden. Oder der Gesamtverlauf der Evaluation zeigt, dass der kritisierte Punkt nur bei der einmalig zu leistenden Erschließung einer Software Schwierigkeiten machte und deshalb als gelöst und für weitere Benutzungen unproblematisch betrachtet wurde, wohingegen andere Nachteile aufgrund wenig durchdachter Implementation wichtiger Funktionalitäten immer wieder auftreten würden.

Hinweise auf solche Relativierungen müssen nicht unmittelbar mit wertenden Kommentaren verbunden sein, sondern können auch erst viel später auftreten. Umso wichtiger ist es, beobachtetes positives oder negatives Erleben nicht isoliert zu betrachten, sondern unter Berücksichtigung des gesamten Evaluationsprozesses hinsichtlich seiner Bedeutung für die Entscheidung zu *interpretieren*. Nur auf diese Weise kann es über das Identifizieren von Schlüsselerlebnissen auch gelingen, vorhandene Unterschiede im Gewicht oder gar eine Hierarchie von Aspekten hinsichtlich ihres Einflusses auf die Auswahlentscheidung zu erkennen (vgl. auch Abschnitt 5.3.2.2).

Die Umsetzung des skizzierten Ansatzes ist sehr einfach gehalten. Dessen Prinzip besteht darin, den zu untersuchenden Sachverhalt, hier repräsentiert durch die aufgezeichneten Interaktionen der Versuchspersonen mit der Software einschließlich begleitender Verbalisierungen, nach Mustern oder „interessant erscheinenden Einzelheiten“ (Bortz & Döring, 2006, S. 269) zu durchsuchen und zu einer Systematik zu ordnen (vgl. Greve & Wentura, 1997). Dementsprechend wird der Evaluationsprozess jedes einzelnen Probanden auf konkrete potenzielle Gründe für dessen Entscheidung analysiert. Dabei wird davon ausgegangen, dass derartige Gründe sowohl den Programmen selbst als auch dem individuellen Evaluationsverlauf entspringen können. Besonderes Augenmerk liegt somit zum einen auf Eigenschaften der Software (z. B. die spezifische Umsetzung einer Funktionalität), die, dokumentiert durch die Kommentare der Versuchspersonen, besonderen Eindruck hinterlassen haben. Zum anderen ist auf verlaufsbezogene Phänomene (z. B. langes Suchen oder gar Scheitern bei der Funktionalitätsexploration) zu achten, für die aufgrund ihrer Bewertung oder unterschiedlicher Ausprägung bei den einzelnen Optionen ein Einfluss auf die Auswahlentscheidung vermutet werden kann.

Allein die Sammlung von auffälligen Ereignissen liefert aber noch keine Struktur, denn zum einen fehlt deren Zusammenhang zum Auswahlergebnis, zum anderen bezieht sich das Beobachtete immer nur auf die individuellen Gegebenheiten des gerade untersuchten Falls. Deshalb sind die jeweils notierten Ereignisse zunächst in Bezug zum Auswahlergebnis zu interpretieren. Durch diese Interpretation werden sie zu potenziellen Gründen, indem ihnen eine bestimmte Wichtigkeit oder Unwichtigkeit für die Entscheidung zugeschrieben wird. Zur Entdeckung fallübergreifender Strukturen bedarf es nun noch einer Verallgemeinerung dieser Interpretation, d. h. der konkret beobachtete Sachverhalt bzw. die in ihn hinein interpretierten Zusammenhänge sind unter Verwendung abstrakter Konstrukte auszudrücken, um eine Vergleichbarkeit der einzelnen Fälle zu ermöglichen. Das Ergebnis dieses Schritts sind Einzelfallhypothesen, die aufgrund ihrer gemeinsamen, abstrakten Terminologie ggf. zusammengefasst werden können.

6.3.2 Analyse der Einzelfälle

Aus den im vorigen Abschnitt dargelegten Überlegungen ergibt sich eine Struktur für die Analyse des Materials zu jedem Einzelfall: Es sind die Interaktionen mit jedem evaluierten Programm nach interessant bzw. für die Entscheidung relevant erscheinenden Ereignissen (im weitesten Sinne dieses Begriffs) zu durchsuchen. Darüber hinaus werden Informationen die Entscheidung selbst betreffend benötigt: Vor allem natürlich, welche Option gewählt wurde, aber auch Randerscheinungen, wie z. B. vom Probanden laut geäußerte Abwägungen von Aspekten, wenn die Entscheidung nicht ganz einfach war. Basierend auf diesen Informationen kann dann die Interpretation des Beobachteten erfolgen, also Überlegungen hinsichtlich des Zusammenhanges von Ereignissen und Auswahlergebnis. Um diese Interpretation zu unterstützen, erscheint außerdem eine kurze Charakterisierung des prinzipiellen Vorgehens der Versuchsperson sinnvoll, weil auf diese Weise einzelne Ereignisse ggf. in einen Kontext gebettet werden können. Der letzte Schritt besteht in der Formulierung abstrakter Hypothesen als eigentliches Ziel der Analyse.

Aus Gründen der besseren Übersichtlichkeit und Transparenz erfolgt die Darstellung der Analyse sowie ihrer Ergebnisse für jeden Probanden entsprechend der im vorigen Absatz entwickelten Struktur jeweils in derselben, im Folgenden erläuterten Form:

Versuchsperson – Erfahrungsklasse – Evaluationstyp – Auswahlergebnis

Vorgehen Kurze Charakterisierung des allgemeinen Vorgehens, ggf. ergänzt um weitere, optionenübergreifende Punkte, wie z. B. die generelle Beachtung ästhetischer Aspekte. An dieser Stelle wird sich im Wesentlichen die zuvor erarbeitete Typologie widerspiegeln (vgl. Abschnitt 6.2).

SmartDraw / Flowcharter / Visio/DS Für jedes untersuchte Programm werden in der Reihenfolge der Bearbeitung Softwaremerkmale und Ereignisse von vermuteter Wichtigkeit aufgelistet. Grundlage für diese Vermutungen sind die jeweiligen Verbalisierungen. Für die Einschätzung der Wertigkeit von Kommentaren gelten die in Abschnitt 5.4.2.3 erläuterten Regeln.

Darüber hinaus wird jeweils notiert, ob das Programm in den wichtigen Teilen unter Kontrolle gebracht werden konnte, d. h. ob es der Versuchsperson gelungen ist, die explorierten Kernfunktionalitäten bewusst und mit dem gewünschten Ergebnis abzurufen. In diesem Fall wird das jeweilige Programmkürzel durch ein * markiert, z. B. *Visio/DS**. Diese Information erscheint wichtig, denn nur wenn der Proband die Kontrolle über ein Programm erlangen konnte, kann davon ausgegangen werden, dass diese Option überhaupt als Sieger infrage kam.

Entscheidung Charakterisierung der unmittelbaren Entscheidung, z. B. ob sie sehr klar war oder dem Probanden schwer gefallen ist und ggf. welche diesbezüglichen Überlegungen geäußert wurden.

Interpretation Die Entscheidung sowie ggf. deren Art und Weise wird mit den vorausgegangenen Ereignissen zu erklären versucht. Dabei wird grundsätzlich davon ausgegangen, dass positiv kommentierte Aspekte die Präferenz für das jeweilige Pro-

gramm verstärken, negative sie abschwächen. Kommt es dabei zu Widersprüchen, muss eine Hierarchie dieser Aspekte unterstellt werden.

Hypothese(n) Die in der Interpretation unterstellten Zusammenhänge zwischen den konkreten Eigenschaften der Software und der geäußerten Präferenz werden unter Verwendung abstrakter Begriffe als allgemeingültige Hypothese(n) formuliert. Deren Kennzeichnung erfolgt in der Form „HNummerDesProbanden.LaufendeNummer“. Sofern der konkrete Sachverhalt dem nicht entgegensteht, werden einheitliche Formulierungen verwendet, um die weitere Auswertung, insbesondere die Aggregation der Einzelfallhypothesen, zu erleichtern.

Da es bei dieser Form der Darstellung vor allem um Übersichtlichkeit geht, sind sperrige Formulierungen fehl am Platze. Deshalb werden zum Ausdrücken häufig auftretender Sachverhalte kurze Bezeichnungen verwendet, die im Folgenden erläutert werden. Einige dieser Bezeichnungen sind in anderen Zusammenhängen bzw. auf bestimmten Gebieten als Fachbegriffe etabliert. Sie werden insofern redefiniert und sind ausschließlich in der hier festgelegten Bedeutung zu verstehen.

Advisor – SmartDraw bietet nach dem Start den sog. *SmartDraw Advisor* an, ein Fenster mit einer Kurzreferenz der wichtigsten Funktionalitäten zum Erstellen von Diagrammen. Dieses Angebot wurde von den meisten Probanden nicht angenommen. Deshalb ist der gegenteilige Fall besonders vermerkt.

Grundnutzen – die Menge der jeweils explorierten Kernfunktionalitäten.

erschlossener Grundnutzen – es ist gelungen, alle explorierten Kernfunktionalitäten auf wiederholbare Weise abzurufen. „Wiederholbar“ bedeutet, dass dem Individuum bewusst ist, auf welche Weise es den gewünschten Effekt hervorgerufen hat.

Zusatznutzen – die Teilmenge der explorierten und als sinnvoll oder wünschenswert erachteten Extrafunktionalitäten.

Ergebnisqualität – die spezifische Art und Weise der Implementation insbesondere der Kernfunktionalitäten hinsichtlich des konkreten grafischen Ergebnisses.

Handhabbarkeit – die spezifische Art und Weise der Implementation der Funktionalitäten, insbesondere hinsichtlich des Abrufmechanismus' sowie des Interaktionsverhaltens.

Optik – Erscheinungsbild einer Software, insbesondere ästhetische Aspekte.

Das beschriebene Format und Vorgehen sollen durch das folgende, sehr kurze und überspitzte Beispiel illustriert und verdeutlicht werden:

Vp0 – unspezifisch erfahren – aufgabenorientiert – Flowcharter

Vorgehen Erzeugt jeweils ähnliches, realistisches Diagramm. Prüft keine Extrafunktionalitäten. Kämpft lange um Problemlösungen.

*SmartDraw** Nutzt den Advisor; findet Erzeugen von Pfeilen umständlich; pixeliges grafisches Ergebnis.

*Flowcharter** Leichte Anfangsschwierigkeiten; lobt das halbautomatische Verbinden von Symbolen.

Visio/DS Positiver Ersteindruck; sucht lange nach dem Mechanismus zum Erzeugen von Pfeilen; bricht schließlich erfolglos ab.

Entscheidung Unproblematisch.

Interpretation Visio/DS kommt als Sieger nicht infrage, weil der Grundnutzen nicht vollständig erschlossen werden konnte. Flowcharter wird gegenüber SmartDraw bevorzugt, weil es trotz Anfangsschwierigkeiten insgesamt besser zu bedienen ist und das bessere Ergebnis liefert.

Hypothese(n)

- H0.1** Ein Programm wird abgelehnt, wenn sein Grundnutzen nicht erschlossen werden kann.
- H0.2** Bei erschlossenem Grundnutzen wird das Programm mit der besseren Ergebnisqualität präferiert.
- H0.3** Bei erschlossenem Grundnutzen wird das Programm mit der besseren Handhabbarkeit präferiert.

Hinsichtlich der Formulierung der Hypothesen ist darauf hinzuweisen, dass sich die darin thematisierten Sachverhalte stets auf die *Wahrnehmung* des jeweiligen Individuums beziehen. Dies ist bei Konstrukten wie Ergebnisqualität oder Handhabbarkeit unmittelbar einleuchtend. Aber auch z. B. „bei erschlossenem Grundnutzen“ bedeutet nicht zwingend, dass tatsächlich alle objektiv erforderlichen Kernfunktionalitäten erfolgreich und in hinreichendem Maße exploriert wurden. Dies ist z. B. bei den Minimalisten kaum der Fall. Es kommt vielmehr darauf an, dass das Individuum der Meinung ist, sich die Funktionsweise der Software so weit erschlossen zu haben, dass eine anschließende Benutzung möglich ist.

Eine ausführliche Formulierung z. B. von H0.2 müsste also konsequenterweise etwa wie folgt lauten: „Bei als hinreichend erschlossen wahrgenommenem Grundnutzen wird das Programm, dessen Ergebnisqualität als besser wahrgenommen wird, präferiert.“ Da diese Variante aber deutlich schwerer lesbar bzw. verständlich ist und es hier ohnehin auf wahrgenommene (statt, soweit überhaupt möglich, objektiv gemessene) Sachverhalte ankommt, wird die einfache Formulierung verwendet.

Die nach dem in diesem Abschnitt erläuterten Vorgehen erarbeiteten Einzelfallanalysen sind auf Anfrage vom Verfasser erhältlich.

6.3.3 Aggregation der Einzelfallhypothesen

Das Ziel dieses Analyseschritts besteht darin, die große Anzahl (38) der jeweils auf einen Einzelfall bezogenen Hypothesen unter Wahrung ihres Sinngehaltes zu einer überschau-

baren Menge zusammenzufassen sowie zu allgemeinen Aussagen zu kommen, die potenzielle Gültigkeit nicht nur für ein Individuum, sondern für eine größere Stichprobe haben können. Um dies zu erleichtern, wurden bei der Ableitung der Einzelfallhypothesen für bestimmte Sachverhalte standardisierte Formulierungen verwendet. Eine erste Reduktion kann deshalb einfach durch Entfernung aller Wiederholungen erfolgen, sodass jeweils nur das erste Exemplar einer Hypothese in der Analyse verbleibt.

Parallel dazu wird eine Sortierung nach zwar nicht mehr identischen, aber ähnlichen Inhalten vorgenommen. Für jede Gruppe, die dadurch entsteht, werden weitere Zusammenfassungen diskutiert. Zu diesem Zweck ist, ggf. unter Rückgriff auf die Einzelfallanalyse, vor allem zu prüfen, ob die Aussage einer Hypothese logisch bereits in einer anderen Formulierung enthalten ist oder aber eine Zusammenfassung der ursprünglichen Bedeutung widersprüche. Eine weitere Methode besteht darin, abweichende Formulierungen oder Begriffe auf gemeinsame Bedeutungen oder zugrunde liegende theoretische Konstrukte zu analysieren, um so eine entsprechende Verallgemeinerung zu rechtfertigen.

Nach dem Entfernen von Wiederholungen verbleiben als allein stehende Hypothesen, d. h. solche, die nicht für eine Gruppierung nach ähnlichen Inhalten infrage kommen:

- H1.1** Bei erschlossenem Grundnutzen wird das Programm mit der besseren Ergebnisqualität präferiert.
- H3.1** Ein Programm wird abgelehnt, wenn sein Grundnutzen nicht erschlossen werden kann.
- H17.3** Je größer die Vertrautheit, desto besser ist die Handhabbarkeit bzw. desto geringer der Erschließungsaufwand.
- H18.3** Aufgabenbezogener Zusatznutzen wiegt schwerer als aufgabenfremder Zusatznutzen.

Dagegen beziehen sich H1.2, H3.2 und H17.2; H4.2 und H6.4; H7.4 und H9.2; sowie H11.2, H15.2 und H18.1 jeweils auf ähnliche Sachverhalte. Die weitere Zusammenfassung bzw. Verallgemeinerung dieser Hypothesen wird im Folgenden diskutiert.

- H1.2** Bei erschlossenem Grundnutzen wird das Programm mit der besseren Handhabbarkeit präferiert.
- H3.2** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität wird das Programm mit der besseren Handhabbarkeit präferiert.
- H17.2** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität wird das Programm mit der besseren Handhabbarkeit bzw. dem geringeren Erschließungsaufwand präferiert.

H3.2 und H17.2 unterscheiden sich darin, dass in Letzterer zusätzlich zur Handhabbarkeit explizit vom Erschließungsaufwand die Rede ist. Diese beiden Konzepte sind eng miteinander verwandt. Es kann davon ausgegangen werden, dass beim Erschließen oder Explorieren einer unbekannteren Software die Wissensstrukturen (Schemata) des Individuums eine Rolle spielen (zur Schematheorie s. J. R. Anderson, 2007 sowie Abschnitt 4.2.1).

Das mehr oder weniger spezifische gespeicherte Wissen über die Benutzung von Software bestimmt, wie versucht wird, die benötigten Funktionalitäten abzurufen. Der Erschließungsaufwand sollte insofern davon abhängen, wie gut die verfügbaren Schemata und die tatsächlichen Gegebenheiten zusammenpassen: Je größer die Übereinstimmung, desto schneller und leichter kann die Funktionsweise erschlossen werden.

Auch die Bewertung der Handhabbarkeit sollte von dieser Übereinstimmung bzw. davon, wie leicht oder schwer sich Unbekanntes assimilieren und vor allem auch akkommodieren lässt, abhängen.⁹ Falls die Funktionsweise den Erwartungen entspricht oder neue Konzepte mit geringem kognitivem Aufwand in die Schemastruktur integrierbar sind, sollte die Handhabbarkeit positiv oder neutral, anderenfalls bzw. wenn dem Individuum einfachere Interaktionsmuster bekannt sind, eher negativ bewertet werden (Mandler, 1982; Meyers-Levy & Tybout, 1989; vgl. Abschnitt 4.2.3.2.2). Vor diesem Hintergrund erscheint es vertretbar, den Erschließungsaufwand als einen Aspekt der Handhabbarkeit zu betrachten, sodass H17.2 nicht weiter gesondert aufrechterhalten, sondern unter H3.2 subsumiert wird.

H1.2 und H3.2 unterscheiden sich darin, dass Letztere als weitere Vorbedingung (zusätzlich zum erschlossenen Grundnutzen) gleichwertige Ergebnisqualität aufführt. H1.2 lässt damit die Möglichkeit offen, dass eine Software bei besserer Handhabbarkeit auch dann präferiert wird, wenn die Ergebnisqualität schlechter ist. Eine Revision der betreffenden Einzelfallanalysen zeigt jedoch, dass eine solche Konstellation vom Datenmaterial nicht unterstützt wird. Vielmehr tritt die Formulierung von H1.2 in jedem der sechs Fälle (Versuchspersonen 1, 2, 5, 6, 7, 16) zusammen mit einer Hypothese im Sinne von H1.1 auf. Das heißt, dass eine Handhabbarkeits-Präferenz stets zusammen mit einer Ergebnisqualitäts-Präferenz oder, gemäß H3.2, einer Ergebnisqualitäts-Gleichheit auftritt. Insofern erscheint es vertretbar, H1.2 unter H3.2 zu subsumieren.

H4.2 Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit dem größeren Zusatznutzen präferiert.

H6.4 Bei erschlossenem Grundnutzen wird das Programm mit dem größeren Zusatznutzen präferiert.

H4.2 und H6.4 unterscheiden sich darin, dass Erstere neben dem gesicherten Zusatznutzen noch weitere Vorbedingungen stellt, nämlich akzeptable Ergebnisqualität sowie Handhabbarkeit. Diese wurde jedoch bei H6.4 nicht unterlassen, weil die Zusatznutzen-Präferenz *trotz* schlechterer Ergebnisqualität und Handhabbarkeit aufgetreten ist, sondern weil die präferierte Option auch in diesen beiden Aspekten am besten abgeschnitten hat. Deshalb ergeben sich aus diesem Einzelfall keine Vorbedingungen für das Wirken eines Zusatznutzens. Die Entscheidungen aller anderen Probanden, bei denen der Zusatznutzen

⁹ Assimilation bedeutet die Integration eines neuen Reizes in ein existierendes Schema, d. h. der Reiz passt zur vorhandenen Wissensstruktur und wird entsprechend abgelegt. Akkommodation erfolgt, wenn ein neuer Reiz nicht in die existierenden Schemata passt und diese entsprechend erweitert oder modifiziert werden müssen, um den Reiz unterbringen zu können (Piaget, 1974; vgl. Abschnitt 4.2.1 zur Schematheorie).

eine Rolle spielte, legen jedoch nahe, dass zunächst die genannten Vorbedingungen erfüllt sein müssen. Da der bei Vp6 beobachtete Sachverhalt nicht dagegen spricht, erscheint es vertretbar, H6.4 unter H4.2 zu subsumieren.

H7.4 Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit der besseren Optik präferiert.

H9.2 Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit professionellerer Optik präferiert.

H7.4 und H9.2 unterscheiden sich nur im Adjektiv, das dem entscheidenden Aspekt der Optik beigelegt wurde. „Professionell“ kann dabei ohne weiteres als eine Konkretisierung von „besser“ und daraus folgend H9.2 als Konkretisierung von H7.4 aufgefasst werden. Noch passender erscheint es jedoch, H9.2 als Spezialfall eines der in H11.2, H15.2 und H18.2 behandelten, „Sicherheit vermittelnden weichen Faktoren“ zu interpretieren, weil es Vp7 in erster Linie um den ästhetischen Anspruch, Vp9 dagegen vor allem um die Implikationen der Optik ging: „Ja, und hier, das wirkt einfach ein bisschen noch’n Tick professioneller. Und wenn man [...] so’n professionelles Programm hat, da kann man eigentlich von ... von ausgehen, dass man da alles mit machen kann. Bei, sonst, bei so’n anderen halb gewalkten Programmen da hat man immer das Risiko [...] dass das da vielleicht nicht drin ist, und hier, das hier sieht mir so aus, als ob das schon nicht irgendwie in so ’nem halben Jahr programmiert ist, sondern irgendwie, eh, ja da’n bisschen Aufwand rein geflossen ist.“, Vp9.128). Da somit H9.2 bereits in anderen Formulierungen enthalten ist, wird sie nicht weiter gesondert behandelt.

H11.2 Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit den mehr Sicherheit vermittelnden weichen Faktoren (Vertrautheit) präferiert.

H15.2 Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit den mehr Sicherheit vermittelnden weichen Faktoren (Vertrautheit, Übersichtlichkeit, berechenbares Verhalten) präferiert.

H18.1 Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit den mehr Sicherheit vermittelnden weichen Faktoren (Vertrautheit, Übersichtlichkeit) präferiert.

H11.2, H15.2 und H18.1 unterscheiden sich lediglich im Umfang der Aufzählung konkreter „Sicherheit vermittelnder weicher Faktoren“. Da es bei der Formulierung dieser Hypothesen nicht um jeweils unterschiedliche Konzepte ging, sondern lediglich um eine Illustration, was mit der abstrakten Verallgemeinerung „Sicherheit vermittelnder weicher Faktor“ jeweils gemeint ist, können diese Konkretisierungen problemlos zusammengefasst werden. Unter Berücksichtigung der Diskussion von H7.4 und H9.2 im vorigen Absatz ist diese Aufzählung noch um den Aspekt „professionelle Optik“ zu erweitern.

Nach Berücksichtigung der diskutierten Zusammenfassungen verbleiben folgende aggregierte Hypothesen:

- H1.1** Bei erschlossenem Grundnutzen wird das Programm mit der besseren Ergebnisqualität präferiert.
- H3.1** Ein Programm wird abgelehnt, wenn sein Grundnutzen nicht erschlossen werden kann.
- H3.2** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität wird das Programm mit der besseren Handhabbarkeit präferiert.
- H4.2** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit dem größeren Zusatznutzen präferiert.
- H7.4** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit der besseren Optik präferiert.
- H15.2** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit den mehr Sicherheit vermittelnden weichen Faktoren (Vertrautheit, Übersichtlichkeit, berechenbares Verhalten, professionelle Optik) präferiert.
- H17.3** Je größer die Vertrautheit, desto besser ist die Handhabbarkeit bzw. desto geringer der Erschließungsaufwand.
- H18.3** Aufgabenbezogener Zusatznutzen wiegt schwerer als aufgabenfremder Zusatznutzen.

Die in H1.1 bis H15.2 enthaltenen, schrittweise umfangreicher werdenden Vorbedingungen legen eine Hierarchie der für die Auswahl von Software relevanten Aspekte nahe, insbesondere da sich die Hypothesen in eine Reihenfolge bringen lassen, in der lediglich weitere Vorbedingungen hinzu kommen, aber keine wieder fallen gelassen werden. Das Entscheidungskriterium einer Position in dieser Reihenfolge wird dabei zur weiteren Vorbedingung für die nächste (vgl. Tab. 6.8).

Hypothese	Vorbedingung	Entscheidungskriterium
H3.1	—	erschlossener Grundnutzen
H1.1	erschlossener Grundnutzen	Ergebnisqualität
H3.2	+ gleichw. Ergebnisqualität	Handhabbarkeit
H4.2, H7.4, H15.2	+ gleichw. Handhabbarkeit	Zusatznutzen/Optik/Sicherheit

Tabelle 6.8: Hierarchie der Entscheidungskriterien

Demnach muss zunächst der Grundnutzen gesichert werden können, damit eine Option in der Auswahlmenge verbleibt. Wenn dies für mehrere gelingt, wird als nächstes die Ergebnisqualität berücksichtigt, d. h. die Option mit dem besten „Output“ gewählt. Gibt es hier keinen klaren Favoriten, entscheidet die Handhabbarkeit. Erst wenn auch nach Hinzunahme dieses Kriteriums keine eindeutige Entscheidung möglich ist, kommen Aspekte

zum Zuge, die für das aktuell zu bearbeitende Problem nicht unmittelbar relevant sind. Dies sind zum einen eventuell vorhandener Zusatznutzen, der die Einsatzmöglichkeiten der Software für weitere Aufgaben erweitert, sowie zum anderen „weiche“ Faktoren zur Vermittlung von Sicherheit sowie zur Befriedigung ästhetischer Ansprüche (Optik).

Nach entsprechender Neunummerierung gehen aus der Analyse der Einzelfälle folgende, finale Hypothesen hinsichtlich der Entscheidungsgründe bei der Ad-hoc-Auswahl von Desktop-Software aus einer Menge von Optionen hervor:

- H1** Ein Programm wird abgelehnt, wenn sein Grundnutzen nicht erschlossen werden kann.
- H2** Bei erschlossenem Grundnutzen wird das Programm mit der besseren Ergebnisqualität präferiert.
- H3** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität wird das Programm mit der besseren Handhabbarkeit präferiert.
- H4a** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit dem größeren Zusatznutzen präferiert.
- H4b** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit der besseren Optik präferiert.
- H4c** Bei erschlossenem Grundnutzen und gleichwertiger Ergebnisqualität sowie Handhabbarkeit wird das Programm mit den mehr Sicherheit vermittelnden weichen Faktoren (Vertrautheit, Übersichtlichkeit, berechenbares Verhalten, professionelle Optik) präferiert.
- H5** Je größer die Vertrautheit, desto besser ist die Handhabbarkeit bzw. desto geringer der Erschließungsaufwand.
- H6** Aufgabenbezogener Zusatznutzen wiegt schwerer als aufgabenfremder Zusatznutzen.

7 Diskussion der Ergebnisse

In diesem Kapitel werden die Ergebnisse der einzelnen Teile der Auswertung diskutiert, d. h. es wird versucht, sie mit Bedeutungen zu versehen bzw. zur bestehenden Theorie in Bezug zu setzen. Außerdem werden die Grenzen der vorliegenden Untersuchung umrissen sowie Implikationen für Forschung und Praxis abgeleitet.

7.1 Ergebnisse der Quantitativen Auswertung

In Bezug auf die *Systematik von Entscheidungsstrategien* (Bettman et al., 1998; vgl. Abschnitt 4.2.2.1) ist generell eine im Wesentlichen alternativenbasierte Bearbeitung festzustellen, d. h. die Alternativen wurden jeweils als geschlossene Einheit und in der Regel abschließend evaluiert, bevor zur nächsten übergegangen wurde. „Fliegende Wechsel“ zwischen den Programmen fanden, wenn überhaupt, nicht im Sinne eines grundsätzlich attributbasierten Vorgehens statt, sondern entweder

- a) um Aspekte, die beim gerade bearbeiteten Programm aufgefallen sind, bei dem oder den bereits untersuchten im Sinne einer Gedächtnisauffrischung nachzuprüfen, oder
- b) weil nach einem Durchlauf durch alle Programme noch kein Sieger feststand und erneut versucht werden sollte, offene Probleme bei bereits ausprobierten Programmen doch noch zu lösen.

Ein typisches Beispiel für Fall a) ist Vp7, der bei der Begutachtung von Visio/DS das Vorhandensein eines Rasters auffiel:

v_{p7.111} Hatte der andere [Flowcharter] auch'n Raster?

112 *wechselt zu Flowcharter und zurück*

113 Nee! Ah, das ist ja'n Pluspunkt für Diagram Studio, das bietet'n Raster an.

114 Und ... zeigt sofort die Ankerpunkte, was ...

115 *wechselt zu Flowcharter und zurück*

116 ... der Flowcharter auch nicht macht.

Ein Beispiel für Fall b) liefert Vp16, die nach Unklarheiten beim dritten Programm (Visio), zum ersten (SmartDraw) zurückkehrte, um dort nochmals zu versuchen, die bisher nur als umständlichen Workaround realisierte Funktionalität zum Zeichnen von Verbindungen korrekt zu erschließen:

- Vp16.117 [in Visio] Was ist denn das? «Einrasten am Verbindungspunkt». Was ist das jetzt hier?
 118 Ja, weiß nicht. Das ist ja auch wieder ein bisschen . . . weiß ich nicht . . .
 119 *wechselt zu SmartDraw*
 120 Also, hier noch mal hingehen. Hier, jetzt mal vom Pfeil abgesehen, würd ich ja sagen, komm ich hiermit am besten klar.
 121 *zieht Linie* [um Anbringen einer Pfeilspitze zu erschließen]

Attributbasiertes Vorgehen im eigentlichen Sinne (vgl. Abschnitt 4.2.2.1) hätte bedeutet, alle Programme parallel zu öffnen, und dann z. B. für alle zunächst zu untersuchen, wie Symbole erstellt werden, dann für alle, wie Verbindungen zu erzeugen sind usw. Ein solches Vorgehen erscheint jedoch umständlich, was durch das Verhalten der Versuchspersonen bestätigt wird: „Ich denke mal, ich sollte mich erst mit dem einen Programm durcharbeiten, weil ansonsten bin ich nachher völlig verwirrt“ (Vp4.1). Der Grund dafür liegt vermutlich darin, dass die einzelnen Attributwerte, in diesem Fall die verschiedenen Implementationen der benötigten Funktionalitäten, nicht leicht und isoliert vergleichbar sind, sondern durch Ausprobieren erst erarbeitet werden müssen und im Zusammenhang mit anderen Attributwerten stehen. Damit ist gemeint, dass es bei der Evaluation einer Software auf das *Zusammenwirken* aller benötigten Funktionalitäten ankommt, um zum gewünschten Ergebnis zu kommen. Demzufolge müssen die Softwareattribute auch in ihrem Zusammenwirken, statt isoliert, begutachtet werden.

Hinsichtlich des Zusammenhangs zwischen *Bearbeitungsreihenfolge und Auswahlhäufigkeit* ist festzustellen, dass am häufigsten das als Zweites oder Drittes untersuchte Programm ausgewählt wurde, nämlich in jeweils 41 % der Fälle (nur Probanden mit drei Optionen). Ein Grund dafür könnte darin liegen, dass beim jeweils ersten Programm Probleme auftreten, die der Unerfahrenheit der Probanden zuzuschreiben sind. Bei den anschließend evaluierten Programmen kann dann beim Ausprobieren des ersten Gelernten zur Anwendung kommen. Dies hätte z. B. zur Folge, dass Mechanismen, die beim ersten Programm schwer erarbeitet werden mussten, bei den anderen Alternativen leichter von der Hand gehen. Als Ergebnis dessen würden sich gegenüber der ersten Option verstärkt negative Emotionen einstellen (Garbarino & Edell, 1997), was das Auswahlergebnis erklären könnte.¹

In diesem Fall wäre es allerdings plausibel, wenn sich ein solcher Effekt besonders deutlich bei gering, und weniger deutlich bei spezifisch erfahrenen Individuen zeigt. Tatsächlich ist die Verteilung bei den Unerfahrenen jedoch ausgeglichen und hat nur einer der spezifisch Erfahrenen das als Erstes evaluierte Programm gewählt (vgl. Tab. 6.5 auf S. 108). Eine andere, wesentlich einfachere Erklärung für die geringe Berücksichtigung der ersten Alternative könnte deshalb in Umkehrung der Kausalität darin liegen, dass das am häufigsten gewählte Programm, Visio/DS (53 % der Probanden mit drei Optionen), nur ein einziges Mal als Erstes ausprobiert wurde. Der beschriebene Effekt wird mit den hier erhobenen Daten nicht weiter zu klären sein, unter anderem weil die Bearbeitungsreihenfolge nicht systematisch, sondern nur zufällig manipuliert wurde. Die weitere Aus-

¹ Zur Rolle von Emotionen im Entscheidungsprozess vgl. Abschnitt 4.3.

wertung hinsichtlich der Gründe für die Auswahlentscheidung (Abschnitt 6.3) legt jedoch nahe, dass diese Variable keinen wirksamen Einfluss auf das Ergebnis entfaltet.

Unklar ist auch der Zusammenhang zur *durchschnittlichen Bearbeitungszeit*, welche vom ersten zum zweiten Programm deutlich abnimmt und zum dritten wieder ansteigt (vgl. Tab. 6.4 auf S. 107). Es erscheint plausibel, dass für die auffällig lange durchschnittliche Bearbeitungszeit für das erste Programm (618 s vs. 386 s und 485 s für das zweite bzw. dritte) tatsächlich Lern- und Gewöhnungseffekte, die den Umgang mit den nachgelagerten Programmen beschleunigen, verantwortlich sind. Warum jedoch für das jeweils letzte Programm wieder mehr Zeit aufgewendet wurde oder werden musste, kann nicht erklärt werden. Auch Hinweise für eine generelle Verortung der Ursache bei spezifischen Eigenschaften der Programme an sich fehlen, da deren durchschnittliche Bearbeitungszeiten relativ ausgeglichen sind (SmartDraw 515 s, Flowcharter 546 s, Visio/DS 505 s).

Zu bedenken bleibt jedoch auch hier, dass mangels systematischer Manipulation eine Vielzahl von Erklärungen infrage kommt. Insbesondere sei darauf hingewiesen, dass eine lange Bearbeitungszeit ihre Ursache nicht notwendigerweise in Problemen mit der Software hat (z. B. Vp3). Vielmehr ist es ebenso plausibel, dass gerade gutes Gelingen, Spaß bei der Benutzung und Begeisterung über die Möglichkeiten zu ausgedehntem Herumprobieren führen kann (z. B. Vp7). Analog muss eine kurze Bearbeitungszeit nicht unbedingt durch rasche Erfolge begründet sein (z. B. Vp17), sondern kann ihre Ursache in einer schnellen Entscheidung über die Ablehnung des Programms haben, weil ein Individuum gar keinen Zugang dazu findet (z. B. Vp13). Für alle Konstellationen finden sich Belege in den Daten. Entsprechend kann auch nicht davon ausgegangen werden, dass stets die Software mit der kürzesten, weil vermeintlich problemlosesten Bearbeitung gewählt wird. Tatsächlich trifft dies nur auf sieben und damit etwa ein Drittel der zwanzig Probanden zu.

Die *einschlägige Erfahrung* der Probanden steht in einem zum Teil etwas überraschenden Zusammenhang zur Bearbeitungszeit (vgl. Tab. 6.6 auf S. 109). Erwartungsgemäß brauchten die gering Erfahrenen im Durchschnitt am längsten für das Ausprobieren eines einzelnen Programms (611 s). Am schnellsten waren aber nicht die spezifisch (547 s), sondern die unspezifisch Erfahrenen (433 s). Tatsächlich verzeichnen Erstere sogar die längste durchschnittliche Gesamtbearbeitungszeit. Allerdings nimmt bei den spezifisch Erfahrenen die Einzelbearbeitungszeit vom zweiten zum dritten Programm nicht so deutlich wie in den anderen Gruppen zu. Deren Entwicklung könnte hier also vollständig mit Lern- und Gewöhnungseffekten erklärbar sein.

Die im Vergleich zum Gesamtdurchschnitt auffällig kurze Bearbeitungszeit der gering Erfahrenen für Flowcharter (392 s vs. 546 s) ist leicht durch das schnelle Aufgeben zweier der drei in diese Gruppe eingeordneten Probanden (Vp14 und Vp15) zu erklären. Auch für die relativ lange Beschäftigung der spezifisch Erfahrenen mit diesem Programm (651 s) sind offensichtlich zwei Ausreißer verantwortlich, nämlich Vp4 (1090 s), die große Probleme hatte, Flowcharter unter Kontrolle zu bekommen, sowie Vp18 (1610 s), die lange mit aufgabenfremden, sie aber anderweitig interessierenden Diagrammtypen herumprobierte.

Die besonders kurze Beschäftigung der unspezifisch Erfahrenen mit Visio/DS (357 s vs. 505 s im Gesamtdurchschnitt) dürfte ihre Ursache im PowerPoint-Hintergrund haben,

den die meisten der in diese Gruppe fallenden Versuchspersonen besitzen. Da sowohl Visio als auch PowerPoint vom selben Hersteller (Microsoft) stammen, ist eine gewisse Vertrautheit mit der Bedienoberfläche und den Funktionsweisen verschiedener Dialogelemente nicht verwunderlich.

Hinsichtlich eines Zusammenhangs zwischen der *einschlägigen Erfahrung* eines Individuums und dem *Auswahlergebnis* besteht die einzige zu diskutierende Auffälligkeit darin, dass keiner der gering Erfahrenen Flowcharter gewählt hat (vgl. Abb. 6.1 auf S.108). Die im Vergleich zur Gesamtverteilung darauf entfallenden Anteile werden von Visio/DS übernommen. Es ist jedoch zu berücksichtigen, dass diese Gruppe nur aus drei Probanden besteht. Eine über diesen Umstand hinausgehende Erklärung könnte darin liegen, dass Flowcharter in der Standardkonfiguration mit einer vergleichsweise knappen und farblosen, dadurch nüchtern und technisch wirkenden Bedienoberfläche aufwartet (vgl. Abb. 7.1), durch die wenig erfahrene Benutzer u. U. abgeschreckt werden. Belege dafür finden sich besonders deutlich bei Vp15:

Vp15.69 *Hauptmenü*

- 70 «Datei ... Neu». Machen wir mal ... das ist ... «Prozess», ja, was ist das jetzt?
- 71 wählt «Prozess» → *Arbeitsfläche mit Abteilung*
- 72 Ach du Schande, was ist das denn? Ahh nee!
- 73 *schließt Dokument und beendet Flowcharter*
- 74 Ich muss das noch mal zumachen.
- 75 Also das war ... das ist Flowcharter jetzt hier, ne?
- 76 VL: *Mhm.*
- 77 *startet Flowcharter*
- 78 «Neues Dokument ... Prozess», ja, was ist das? «Prozess ... Ursache-Wirkungs-Diagramm ... Basisdiagramm». Also das find ich schon ...
- 79 «Prozess» ... kann ich jetzt erst mal so gar nichts mit anfangen, ne, «Ursache-Wirkungs-Diagramm» ... «Basisdiagramm» würde ich jetzt mal gehen.
- 80 *klickt dies*
- 81 *erzeugt Symbol*
- 82 Aha. Ach so.
- 83 *klickt Stift-Button und versucht Linie zu ziehen*
- 84 Würde ich jetzt ... Also das ist sehr ... Ja, also hier fehlt mir ja wirklich da so'n bisschen der Bezug, ne. Also hier ... das wäre jetzt, würde mich jetzt nicht so ansprechen hier. Das finde ich so'n bisschen ...
- 85 Was ist das hier eigentlich alles? Alles sehr unübersichtlich hier, von den, von den einzelnen ... Das wird sicherlich hier Schrift, Textfeld sein ... ja.
- 86 Was ist das hier? «Weitere Symbole». Ach nee, das ...

87 Nee, also hier müsste ich mich zu lange erst mal reinkucken.

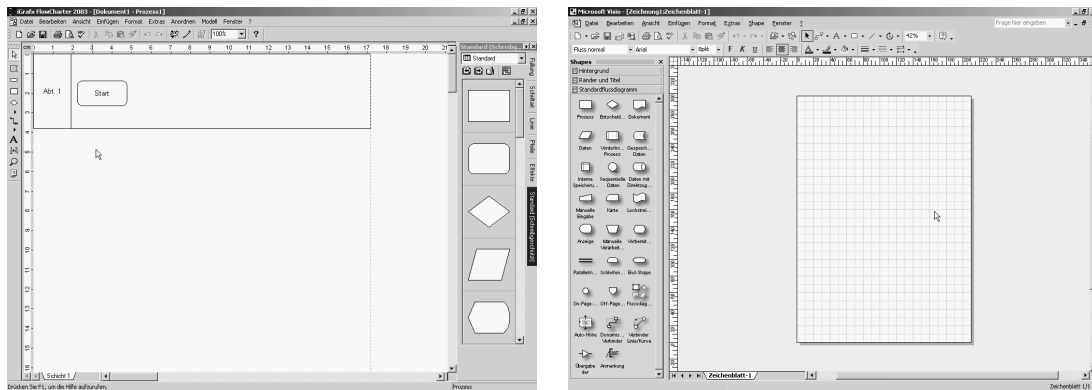


Abbildung 7.1: Die Benutzeroberflächen von Flowcharter (links) und Visio (rechts) im Vergleich

Erschwerend kommt hinzu, dass die Anordnung der Bedienelemente nicht ganz konsistent ist, welche sich insbesondere beim Erzeugen von Verbindungen bei fast allen Probanden irreführend auswirkte.² Tatsächlich brechen alle drei als gering erfahren eingestufte Probanden die Evaluation von Flowcharter mit deutlich abwertenden Kommentaren ab, ohne die benötigten Funktionalitäten vollständig erschlossen zu haben (vgl. auch obiges Blockzitat von Vp15):

Vp6.96 Komisch, das Programm! Kucken wir mal'n anderes an. [nach langem, erfolglosem Kampf mit V_E].

Vp6.211 [rückblickend] Ach, das war der blöde Flowcharter.

Vp14.219 Na super! Echt Klasse. Nee, das reicht. Das ist doof. Darf ich das wegmachen? Das nervt. Wahrscheinlich bin ich zu doof für dieses Programm.

Vp15.222 [...] gut, weiß nicht, hier ist es mir zu unübersichtlich. Also, hier würd ich schnell die Lust dran verlieren. Nee, komm ich nicht so richtig ... bin ich nicht so richtig vertraut.

Ansonsten liefern die Daten keine Hinweise auf einen Zusammenhang zwischen der einschlägigen Erfahrung eines Individuums und dem Auswahlergebnis (vgl. Abb. 6.1 auf S. 108), d. h. es scheint nicht so zu sein, dass ein vergleichbares Erfahrungsniveau, zumindest in der hier operationalisierten Form, zu übereinstimmenden Präferenzen führt. Insbesondere in den Gruppen der spezifisch sowie unspezifisch Erfahrenen entsprechen die Anteile der einzelnen Programme (unter Berücksichtigung der aufgrund der Stichprobengröße begrenzten mathematischen Möglichkeiten) etwa denen der Gesamtstichprobe,

²Die Anordnung der Bedienelemente lässt den von fast allen Probanden zunächst gezogenen Schluss zu, dass Symbole und Verbindungen nach demselben Dialogprinzip erzeugt werden können, was aber nicht der Fall ist und zu entsprechenden Fehlbedienungen mit „unerklärlichen“ Misserfolgen führte.

mit einer optisch leicht verstärkten Tendenz der unspezifisch Erfahrenen zugunsten von Flowcharter (vgl. Abb. 6.1 auf S. 108). Allerdings wäre es aufgrund der Gruppengrößen auch gar nicht möglich gewesen, genau die Verteilung der Gesamtstichprobe zu reproduzieren.

Eine Analyse der Verteilung der Erfahrungsgruppen auf die einzelnen Programme verstärkt den ausgeglichenen Gesamteindruck. Lediglich Flowcharter weicht etwas ab, indem dessen Anteil an gering durch unspezifisch Erfahrene übernommen wird, entspricht aber ansonsten den beiden anderen Optionen, die im Wesentlichen die Gesamtverteilung der Probanden auf die Erfahrungsklassen repräsentieren und deren Wähler sich jeweils zu etwa einem Drittel aus spezifisch Erfahrenen rekrutieren.

Zusammenfassend bleibt festzustellen, dass sich allein aus den in diesem Teil der Analyse ausgewerteten, unmittelbar beobachtbaren Variablen, d. h. ohne tiefer gehende, qualitative Analyse des individuellen Geschehens im Verlauf der einzelnen Evaluationen keine zufrieden stellenden Ansätze zur typisierten Beschreibung des Verhaltens der Individuen bzw. zur Erklärung ihrer Auswahlentscheidungen ergeben.

7.2 Ergebnisse der Qualitativen Analyse I

7.2.1 Typologie

Die qualitative Untersuchung des Verhaltens der Individuen bei der Evaluation und Auswahl von Software zur Bearbeitung einer konkreten Problemkategorie hat deutliche Unterschiede zutage gebracht. Als wichtige Gruppierungskriterien stellten sich die Komplexität der zu Testzwecken erzeugten Diagramme sowie das Ausmaß der Berücksichtigung nicht zwingend erforderlicher Funktionalitäten heraus. Darin manifestierte sich eine interindividuell mehr oder weniger stark ausgeprägte Neigung, die angebotenen Programme nicht nur flüchtig, sondern mehr in die Tiefe bzw. die Breite gehend auf die Art und Weise sowie den Umfang der Implementation sowohl benötigter als auch weiterer Funktionalitäten zu testen. Diese Neigung wurde als Ausdruck des unterschiedlich gelagerten Bestrebens interpretiert, die „richtige“ Entscheidung zu treffen bzw. sich gegen Überraschungen abzusichern, indem bereits vorher ein realistisches Problem bearbeitet wird.

Darüber hinaus zeigten sich Unterschiede hinsichtlich des Eindrucks von der Sicherheit und Selbstbestimmtheit des Vorgehens, den die Probanden hinterließen. Es wurde argumentiert, dass das Verhalten der Unsicheren im Gegensatz zu dem der anderen nur teilweise ihrer unmittelbaren Intention entspringt, sondern erheblich durch das ungünstige Zusammentreffen ihrer individuellen Fähigkeiten und Kenntnisse einerseits sowie der spezifischen Gegebenheiten von Problem und Software andererseits determiniert wird. Dies unterscheidet sie von den Aufgabenorientierten, Minimalisten und Verspielten als Gruppe, sodass die Struktur der in Abschnitt 6.2 erarbeiteten Typologie als Baum dargestellt werden kann (vgl. Abb. 7.2), woran sich die folgende Diskussion orientieren wird.

Einen Ansatz zur Erklärung der beobachteten Unterschiede zwischen Aufgabenorientierten, Minimalisten und Verspielten untereinander liefert die Theorie eines Trade-off

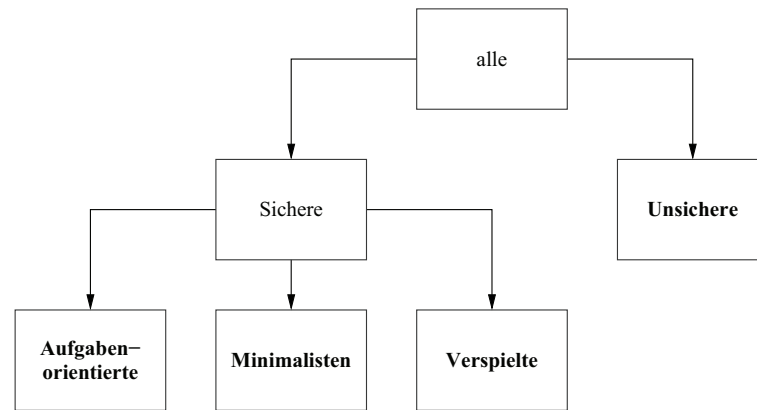


Abbildung 7.2: Struktur der Typologie des Evaluationsverhaltens (eigene Darstellung)

zwischen Aufwand und Nutzen einer Entscheidungsstrategie (vgl. *Effort-Accuracy Framework*, Payne, 1982; *Choice-Goals Framework*, Bettman et al., 1998). Demnach kann eine Entscheidungsstrategie durch den Grad ihrer Präzision und den dafür erforderlichen Aufwand charakterisiert werden. Das Entscheidungsverhalten eines Individuums basiert entsprechend auf einem Kompromiss zwischen dem Bestreben, die beste Entscheidung zu treffen, und dem Bestreben, den dafür zu investierenden kognitiven Aufwand zu minimieren.

Allerdings geht die hier untersuchte Situation über die reine Entscheidungsfindung und die damit verbundene Beschränkung auf die Verarbeitung von extern bereit gestellten Informationen hinaus. Deshalb muss das Konzept der Aufwandsbereitschaft neben der kognitiven um eine motivationale sowie eine manuelle Komponente erweitert werden: Zum einen müssen die Informationen, auf denen eine Entscheidung basiert, nicht nur verarbeitet, sondern zunächst erst einmal durch zum Teil mühsames Ausprobieren (Trial & Error) überhaupt erst einmal beschafft werden. Zum anderen stellt das Erzeugen weiterer Flussdiagrammelemente, nachdem die dafür nötigen Mechanismen bereits exploriert wurden, keine besondere kognitive Herausforderung mehr dar, sondern ist vor allem als rein manuelle Verrichtung anzusehen.

Das Verhalten aufgabenorientierter Individuen wäre im Rahmen dieses Ansatzes vom Wunsch danach determiniert, eine korrekte Entscheidung zu treffen. Deshalb sind sie bereit, den mit einer gründlichen Evaluation verbundenen erhöhten Aufwand zu treiben. Minimalisten verkörpern das genaue Gegenteil dazu: Sie wollen so wenig Aufwand wie möglich investieren und nehmen dabei mehr oder weniger bewusst das Risiko in Kauf, dass sich beim späteren Einsatz der gewählten Software herausstellt, dass bestimmte Funktionalitäten nicht im benötigten Umfang oder auf die gewünschte Weise implementiert sind. Indiz für die geringe Aufwandsbereitschaft ist der Umstand, dass sich alle sechs Vertreter dieses Typs unter den acht Probanden mit den kürzesten durchschnittlichen Bearbeitungszeiten für ein einzelnes Programm befinden.

Dass die Aufgabenorientierten nicht analog dazu die längsten Bearbeitungszeiten aufweisen, spricht nicht gegen den hier diskutierten Erklärungsansatz. Die oben referenzierte Theorie (Payne, 1982; Bettman et al., 1998) postuliert lediglich *hohe* Präzision bzw.

niedrigen Aufwand als wünschenswerte Ziele. Dagegen ist die Annahme eines Wunsches nach *niedriger* Präzision oder *hohem* Aufwand in der Regel nicht sinnvoll. Das heißt, dass diese beiden Größen nur in der jeweils wünschenswerten Ausprägung Determinanten des Verhaltens sein können. Bei vorgegebenem Niveau der einen Größe stellt sich das der jeweils anderen je nach Situation entsprechend ein und muss hingenommen werden.

Es ist also nicht davon auszugehen, dass Aufgabenorientierte unbedingt hohen Aufwand treiben *wollen*; sie akzeptieren ihn vielmehr als Preis der tatsächlich gewünschten Präzision. Der Aufwand kann aber unter günstigen Umständen auch niedrig bleiben. Analog *wollen* die Minimalisten keine unzuverlässige Entscheidung. Sie akzeptieren sie jedoch als Preis eines geringen Evaluationsaufwands. Ergo liefern Minimalisten geradezu notwendigerweise kurze Bearbeitungszeiten im Sinne einer unabhängigen Variablen. Die Bearbeitungszeiten der Aufgabenorientierten dagegen sind abhängige Variablen. Sie tendieren zwar dazu, länger zu sein, streuen aber in Abhängigkeit von den Umständen des Evaluationsverlaufs.

Bei den Verspielten führt, wie bei den Minimalisten, ebenfalls der Drang nach Aufwandsminimierung zu nur oberflächlicher Exploration der Kernfunktionalitäten, welche sich in wenig komplexen Zeichnungen manifestiert. Das Extrem liefert hier Vp17 mit einer durchschnittlichen Komplexität von nur 0.67! Das heißt, dass bei mindestens einem Programm nur „sinnlos herumgeklickt“ wurde, ohne zu einem zählbaren Ergebnis zu kommen. Obwohl es sich eigentlich nur aus dem gesamten Evaluationszusammenhang vollständig erschließt, der hier unmöglich wiedergegeben werden kann, wird im Folgenden versucht, einen Eindruck vom sprunghaften Vorgehen dieses Probanden zu geben:

- Vp17.58 So. Und wie krieg ich da jetzt ne Pfeilspitze ran? Na, hervorragend. Lassen wir das.
 59 Jetzt frage ich mich, wie ich hier Text einfügen soll.
 [...]

80 «Text an Linie ausrichten ... horizontal». Das ist natürlich schön.
 81 *klickt mehrfach erfolglos*
 82 Macht er da irgendwas? Oder hab ich das falsch verstanden? Hab ich wohl falsch verstanden.
 83 Alles klar. So, jetzt würde ich wahrscheinlich schon das nächste Programm ausprobieren.
 [...]

110 «Effekte ... Pfeile» noch mal da, ist auch gut. «Linie ... Schriftart ... Füllung». Ja, nicht schlecht.
 111 *Hauptmenü «Ansicht – Symbolleisten» → Symbolleisten-Dialog*
 112 Das kann man wahrscheinlich irgendwo einfügen, die unterschiedlichen Symbolleisten. «Objektarten ... Zoomsteuerung» ... Mal sehen, was da passiert.
 113 *schaltet alle verfügbaren Symbolleisten ein*

- 114 Alles klar. Noch allen möglichen Scheiß, den ich mir hierhin ziehen kann. Na Klasse.
Ob ich das nun brauche oder nicht, wer weiß; gut zu wissen, dass es da ist.
- 115 Joa, nö, weiß ich im Prinzip alles, was ich wissen muss.
- 116 Werd ich noch mal hier reinkucken [*Hauptmenü*] ... «Bearbeiten» ... «Galerie ... Ausgabe ... Lineale» ... Lineale sind schon drin, die werden dann weggehen, alles klar.
- 117 «Objekte ... Effekte» ... So auf den ersten Blick seh ich das natürlich nicht, aber es scheint mir hier tatsächlich noch ein paar mehr Funktionalitäten zu haben. Wirkt auf mich jetzt so. Natürlich kenn ich die auch alle schon, weil die so ähnlich sind ...
- 118 «Konvertieren in ... Grafik», auch 'ne feine Sache.
- 119 Okay, das sind normale Fenster, gut. Das war's.
[...]
- 178 Aber das scheint hier auch einigermaßen zu funktionieren, also ...
- 179 *VL: Ja nun, aber dass du damit klar kommst nachher ...*
- 180 Jaa ... so auf die Schnelle nicht.

Das geradezu konfus erscheinende Verhalten von Vp17 führt zu der Vermutung, dass verspielte Individuen vor allem den kognitiven Aufwand einer konsequenten und fokussierten Erarbeitung des Notwendigen scheuen. Die Folge davon ist wenig zielgerichtetes, leicht ablenkbares Explorieren mit dem Ergebnis, dass auch auf viele Extrafunktionalitäten getroffen wird. Anders als Minimalisten sind Verspielte jedoch zu ausgiebigen manuellen Verrichtungen bereit, sodass interessante Extras gern ausprobiert werden. Weil dies aus eigenem Interesse und Antrieb, quasi als „Kür“ erfolgt, wird der dafür erforderliche Aufwand nicht als Last, sondern eher als Spaß wahrgenommen. Dieser Aspekt zeigt sich sehr deutlich bei Vp7:

- ^{Vp7.182} So, jetzt wird hier mal eingefärbt. Schön mit Pastell, mmm. Und drum herum die Linie, kann ich die dicker machen? So, und, ja, so dick. Und ich welcher Farbe? «Linienfarbe», orange, dunkelgelb ... okay. Noch'n dicken Schatteneffekt, und jetzt wird das ja richtig stylermäßig. Ja, so könnte mir das gefallen. Hier gibt's auch noch'n Schatten ... Schatten, kein Schatten ... so so.

Verspielte haben durchaus ein Interesse an der Entscheidungsqualität, jedoch ist dieses anders gelagert als bei den Aufgabenorientierten. Eine korrekte Entscheidung hinsichtlich des aktuell zu lösenden Problems ist zwar nicht wichtig genug, um eine tiefe Exploration der Kernfunktionalitäten zu bewirken (z. B. Vp17.178–180, s. Blockzitat oben), jedoch scheint es nicht unwichtig zu sein, welche darüber hinausgehenden Fähigkeiten eine Software hat, z. B. (vgl. auch Vp7 oben):

- ^{Vp18.45} [...] Ursache-Wirkungs-Diagramm kann ich nämlich sehr gut gebrauchen für meine Doktorarbeit; das würde mir also schon mal gefallen hinsichtlich der Erstellung solcher

Diagramme. Und von daher wollen wir jetzt mal hier weiter kucken, wie das ... da bin ich da drin.

Insofern verkörpern Verspielte ein Art Kompromiss zwischen Minimalisten und Aufgabenorientierten: Für Minimalisten steht die Aufwandsminimierung im Vordergrund, deshalb explorieren sie weder in die Breite noch in die Tiefe. Bei den Aufgabenorientierten dominiert das Streben nach der richtigen Entscheidung für das aktuelle Problem, deshalb explorieren sie tief, aber nicht breit. Verspielte hingegen wollen auch über das aktuelle Problem hinausgehende Funktionalitäten untersuchen, sich dabei aber nicht besonders anstrengen, deshalb explorieren sie breit, aber nicht tief.

Die Unsicheren nehmen in dieser Argumentation eine Sonderstellung ein, da bei ihnen davon ausgegangen wird, dass ihr Verhalten nicht nur ihrer Intention entspringt, also nicht in erster Linie eine Folge von Abwägungen hinsichtlich Aufwand und Präzision der Entscheidung ist, sondern maßgeblich durch ihre eingeschränkten Kenntnisse und Fähigkeiten hinsichtlich der zu bewältigenden Aufgabe diktiert wird. In der Komplexität ihrer Abbildungen erinnern sie zwar an Minimalisten, unterscheiden sich von diesen jedoch darin, dass sie sich zum Teil sehr lange mit einer Software auseinandersetzen, statt die Evaluation bei hartnäckigen Problemen einfach abubrechen und sich dem nächsten Programm zuzuwenden. Diese Aufwandsbereitschaft erinnert an die Aufgabenorientierten.

Es ist jedoch zu berücksichtigen, dass die Unsicheren bei *allen* Programmen Schwierigkeiten haben und es sich insofern nicht leisten können, eine Option vorschnell aufzugeben: Bei der Nächsten ist mit denselben Problemen zu rechnen, weil diese vor allem in der mangelnden Erfahrung der Person begründet sind. Dieser Überlegung folgend, weisen Unsichere ein ganz eigenes Muster hinsichtlich Aufwand und Präzision der Entscheidung auf: Die Entscheidungsqualität ist wichtig, aber auf einem anderen Niveau als bei den Aufgabenorientierten, denn es gilt, ein Programm zu finden, mit dem das Problem *überhaupt* gelöst werden kann. Sie geben sich entsprechend mit einer minimalistischen Tiefe der Evaluation zufrieden, weil durch ihre Unerfahrenheit damit bereits hohe Ansprüche an ihre kognitiven Ressourcen gestellt werden.

Diese Besonderheit der Unsicheren gegenüber den anderen Typen kann durch die Schematheorie erklärt werden. Im Allgemeinen ist unter einem Schema eine Form der Repräsentation von Wissen über Klassen von Sachverhalten zu verstehen. Schemata sind in einer Generalisierungshierarchie strukturiert, d. h. dass die Konzepte von unten nach oben immer allgemeiner werden. Sie sind so beschaffen, dass sie Schlussfolgerungen über konkrete Exemplare der in den Schemata repräsentierten Konzepte erleichtern (z. B. J. R. Anderson, 2007; vgl. Abschnitt 4.2.1).

Demnach müssten die Schemata, über die ein Individuum verfügt, auch Einfluss darauf haben, auf welche Weise versucht wird, eine Software zu bedienen. Je nachdem, in welchem Maße die Schemahierarchie hinsichtlich der aktuell zu bedienenden Softwarekategorie über spezifische oder nur allgemeine Konzepte verfügt, sollte die Bedienung leichter oder schwerer fallen. Jemand, der schon Erfahrung mit Diagrammsoftware hat, besitzt eine Vorstellung oder ein Konzept (ein Schema) davon, wie z. B. ein Verbindungspfeil typischerweise erzeugt wird. Ein Bestandteil dieses Schemas könnte z. B. sein, dass beim Fehlen eines direkten Pfeil-Buttons erst eine einfache Linie gezogen und dann in

einem Formatierungsdialog die Pfeilspitze eingestellt werden muss. Insofern kann dieser erfahrene Jemand zielgerichtet nach einer entsprechenden Implementation suchen.

Das weniger elaborierte Softwarebedienungs-Schema eines Individuums ohne solche Erfahrung dagegen mag sich auf das allgemeine Wissen beschränken, dass alle Effekte immer mit irgendwelchen Buttons erzeugt werden. Dieses Individuum wird sich entsprechend schwerer tun, über die Suche nach dem Pfeil-Button hinauszugehen. Wird dann bei mehr oder weniger planlosem Herumklicken zufällig eine Lösung gefunden, kann sie u. U. nicht mehr nachvollzogen werden, weil das eher unbewusst erworbene neue Wissen noch nicht in die Schemastruktur aufgenommen wurde.

Diese Beschreibung eines solchen Individuums passt auf den Typus der Unsicheren, der ausschließlich durch gering erfahrene Probanden besetzt ist (vgl. Abb. 6.12 auf S. 133). Zu der Argumentation kompatibel ist auch der Umstand, dass es lediglich in der Gruppe der spezifisch Erfahrenen jeder einzelnen Versuchsperson gelungen ist, jedes Programm unter Kontrolle bringen, d. h. an keiner Funktionalität endgültig zu scheitern. In der Gruppe der unspezifisch Erfahrenen trifft dies nur auf drei von acht, bei den gering Erfahrenen auf eine von vier Versuchspersonen zu.

7.2.2 Zusammenhang von Typologie und Auswahlergebnis sowie Erfahrung

Der Zusammenhang zwischen einschlägiger Erfahrung und Evaluationstyp ist intuitiv plausibel und insofern nicht überraschend (vgl. Abb. 6.12 auf S. 133). Insbesondere der Umstand, dass sämtliche Unsicheren (Typ IV) nur geringe Erfahrung haben, passt ins Bild und ist bereits im vorigen Abschnitt diskutiert worden. Das von dem der drei anderen Probanden in dieser Erfahrungsgruppe abweichende Verhalten von Vp6 (verspielt statt unsicher) kann inhaltlich evtl. dadurch erklärt werden, dass sie, obgleich im Hinblick auf Flussdiagrammsoftware gering erfahren, als Wirtschaftsinformatik-Studentin einen etwas vertrauteren Umgang mit dem Computer pflegt als BWL- und Lehramt-Studenten (Vp14 und Vp15 bzw. Vp2).

Bei direktem Vergleich der unspezifisch mit den spezifisch Erfahrenen ist festzustellen, dass bei Ersteren Typ II (minimalistisch), bei Letzteren dagegen Typ I (aufgabenorientiert) und III (verspielt) etwas stärker vertreten sind. Obwohl dies einfach der Varianz geschuldet sein könnte, lassen sich auch inhaltliche Erklärungsansätze finden: Spezifische Erfahrung mit dem Zeichnen von Diagrammen hält die entsprechenden Probanden von minimalistischer Evaluation (bzw. davon, dem Streben nach Aufwandsminimierung nachzugeben, vgl. Abschnitt 7.2.1) eher ab: Sie wissen aus ebendieser Erfahrung, was dabei alles Probleme machen bzw. welche mehr oder weniger hilfreichen oder interessanten Extrafunktionalitäten es geben kann. Entsprechend neigen sie dazu, ihre Begutachtungen mehr in die Tiefe (Typ I) oder in die Breite (Typ III) auszudehnen.

Diese Erklärung ist kompatibel zu einem Befund von Galletta et al. (1993), die Unterschiede zwischen Experten und Novizen bei der „Hands-off“-Auswahl³ von Datenbanksystemen untersuchten. Die Autoren stellten fest, dass Experten viermal so viele sog. „Elaboration Statements“ (ebd., S. 15) abgaben wie Novizen, d.h. Bemerkungen machten, die die Anwendung externer, nicht aus den vorgelegten Materialien abgeleiteter Erfahrungen erkennen ließen. Die Schlussfolgerung daraus entspricht dem hier zugrunde gelegten Erklärungsansatz für das Verhalten der spezifisch Erfahrenen, nämlich dass die Erfahrung Einfluss auf den Auswahlprozess nimmt. Allerdings stellten Galletta et al. (1993) auch einen deutlichen Einfluss auf das *Auswahlergebnis* fest, der in dieser Studie nicht erkennbar ist (vgl. Abschnitt 6.1.4).

Auch die Befundlage hinsichtlich eines Zusammenhangs zwischen Auswahlergebnis und Evaluationstyp ist in der vorliegenden Studie nicht eindeutig. Vielmehr erscheint die Zuordnung zufällig und nur indirekt erklärbar. Die mangelnde Berücksichtigung von Flowcharter durch Unsichere könnte mit dessen technisch-nüchtern und inkonsistent wirkender und insofern abschreckender bzw. verwirrender Benutzeroberfläche zu tun haben und wurde bereits in Abschnitt 7.1 diskutiert.

Auch Flowcharters äußere Erscheinung könnte dazu beigetragen haben, dass sich keiner der Minimalisten für dieses Programm entschieden hat. Ein entsprechender Erklärungsansatz basiert darauf, dass diesen Probanden in Abschnitt 7.2.1 als verhaltensbestimmendes Motiv ein Streben nach Aufwandsminimierung unterstellt wurde. Der kognitive und zeitliche Aufwand, sich die wenig vertraute, kompliziert erscheinende Bedienung von Flowcharter beim Auftreten von Problemen doch noch zu erschließen, könnte als zu hoch empfunden worden sein, was angesichts des Vorhandenseins weiterer Optionen zu dessen Ablehnung führte:

Vp8.35 So, das will ich jetzt auch noch mal verbinden, von hier nach da, an die Linie ... ist schon wieder schwierig. [?] Hmm. Wie krieg ich einen Pfeil ... vielleicht von der Linie anfangen ... nein, nur von da ... auch nicht von da. Nicht mal von da! Hm, gut.

36 Dann probiern wir erstmal die anderen aus.

Vp13.45 Ehm, also Beschriftungen ... «Abteilung einfügen», nee, Abteilung will ich eigentlich auch nicht einfügen.

46 *rechte Reiterleiste*

47 «Schriftart»! Dann muss man doch auch irgendwie Beschriftungen einfügen können, wenn man die Schriftart festlegen kann.

48 Also Pfeile beschriften scheint mir hier nicht so gut zu klappen.

[...]

50 *minimiert Flowcharter, startet Visio*

³ Im Gegensatz zum hier untersuchten interaktiven „Hands-on“-Vorgehen kamen die Versuchspersonen bei Galletta et al. (1993) nicht in direkten Kontakt mit den Alternativen, sondern wählten auf der Basis von bereitgestellten Attributlisten aus.

Andere Minimalisten hatten zwar keine solche Schwierigkeiten und fanden sogar anerkennende Worte für Flowcharter⁴, beschäftigten sich aber nicht eingehend genug mit diesem Programm, um dessen besondere Stärken zu erkennen. Aufgabenorientierte hingegen, denen statt des Evaluationsaufwandes das Risiko, die beste Option durch eine nur oberflächliche Begutachtung zu verpassen, zu hoch war (vgl. Abschnitt 7.2.1), gaben sich nicht so schnell zufrieden. Dadurch blickten sie sozusagen hinter die nüchterne Fassade und konnten sich die „inneren Werte“ von Flowcharter erschließen. Das heißt zwar nicht, dass notwendigerweise jeder Aufgabenorientierte von diesem Programm begeistert war, könnte aber dessen relativ starke Stellung bei diesen Probanden erklären. Tatsächlich wurden, passend zum Charakter des aufgabenorientierten Typus, anscheinend vor allem spezifische Merkmale der Umsetzung von Kernfunktionalitäten als Stärke wahrgenommen, wenn die Mechanismen zu deren Abruf erst einmal durchschaut und beherrscht werden:

Vp5.49 *wählt den Reiter «Pfeile» und zieht eine Verbindung von Symbol zu Symbol*

- 50 Dann drück ich jetzt mal auf Pfeile und kuck mal, ob ich das hinkriege ... faszinierend.
51 Und das «Nein» taucht auch sofort auf! Und wenn ich hier ein «Ja» haben will? Na egal.
52 *zieht weitere Verbindung zu anderem Symbol*
53 Und da runter ist «Ja», das ist ja [?], das ist nicht schlecht.

Vp12.79 So, noch ein Pfeil. Pfeil anklicken, Seite ... Oh! Jetzt macht er automatisch ein «Nein». Und wenn ich da ein «Ja» hin haben will?

- 80 *öffnet Kontextmenü an der Linie, ändert Label*
81 Aha, rechte Maustaste, «Ja». Das ist ja schon mal nicht schlecht.
82 Das ist ... und wenn ich jetzt hier unten ... Steuerung-Copy, Steuerung-V ... das Ding, na! Ich will das Ding verschieben ... wenn ich jetzt hier wieder so'n Pfeil ran mache, macht er dann automatisch ein «Nein» ran. Ist ja geil! Das ist schon mal cool.
83 Und wenn ich jetzt hier wieder ne Verbindung nach oben haben will, malt er auch schön um die Ecke. Das ist gut.

Vp16.132 Also ich hatte leichte Anfangsprobleme, aber ich muss sagen, es ist auch gut zu handhaben jetzt, wo ich mich 'nen Augenblick eingefuchst hab. Mit dem anderen konnte man gleich drauf loslegen, aber dieses ist wahrscheinlich dafür ein bisschen umfangreicher. Kann man auch für Schaltungen und sonstwas anwenden.

- 133 Okay. Also von den Pfeilen bin ich ja ein bisschen begeistert ...
[...]

⁴ Z. B.: „Formen sind da, Pfeile sind da, kann man verbinden, Text passt rein. Ja, brauchbar, würd' ich sagen“ (Vp9.47), oder: „Was brauchen wir noch? Wir brauchen die Symbole, wir brauchen die Zahlen, brauchen die Pfeile, hmm, ja, das ist ja schon mal ganz geschmeidig. Ich glaube, hiermit würde ich das hinkriegen“ (Vp11.19).

- 137 Aha, jetzt hat er hier gleich was drauf geschrieben. Das ist ja interessant. Da ist schon «Nein» drüber geschrieben.
- 138 Nein, Moment. Über dem Pfeil ist gleich schon ein Feld, wo «Nein» steht, und ich wollte da ein «Ja» rein haben. Kann ich das verändern? Ja, ich klick rechts und kann aus dem «Nein» ein «Ja» machen. Das fetzt ja. Das ist ja so'n bisschen wie Mitdenken hier.
[...]
- 142 Also wähl ich noch mal das Rechteck aus, vergrößere das, platziere das ordentlich da unter. So, da möchte ich auch einen Pfeil runter haben, ja, und da schreibt er auch gleich was ran.
- 143 Aha, er scheint also zu erkennen, dass dieses Feld hier eine Auswahlmöglichkeit ist. Das begeistert mich ja schon mal. Rechts, und ... Ach nee, «Nein» war ja richtig.
- 144 Super. Das finde ich ja ein bisschen gut.

Minimalisten dagegen, nicht unbedingt auf der Suche nach dem besten Programm, sondern nach dem, welches sie sich mit dem geringsten Aufwand erschließen können, haben eher eine Präferenz für Visio/DS. Da dieses Programm mit Microsoft denselben Hersteller hat, wie die von vielen Individuen alltäglich benutzten Büroprogramme, z. B. Word oder PowerPoint, sind zumindest Teile von dessen Oberflächengestaltung und Bedienmechanismen geläufiger als bei den anderen Alternativen. Das führt einerseits zu einer leichter von der Hand gehenden Evaluation, andererseits zu dem Vertrauen, auch bei nur oberflächlicher Begutachtung einschätzen zu können, ob die spätere Benutzung problemlos gelingen wird.

Diese Erklärungsansätze sind recht spekulativ und unvollständig, denn die Wahl von SmartDraw sowie die Auswahlresultate der Verspielten können damit nicht erklärt werden. Deshalb muss festgestellt werden, dass die Ergebnisse keinen direkten Zusammenhang des Evaluationstyps auf das Auswahlresultat nahe legen. Plausibel erscheint allenfalls ein moderierender Einfluss, der über den Kompromiss zwischen geringem Evaluationsaufwand und der Korrektheit der Entscheidung wirksam wird (*Effort-Accuracy Trade-off*, vgl. Abschnitt 7.2.1). Zur Illustration werden im Folgenden zwei Fälle konstruiert und am Beispiel eines Aufgabenorientierten (A) und eines Minimalisten (M) als den beiden Gegenpolen des Aufwand-Genauigkeit-Konstruktes erläutert. Dabei wird unterstellt, dass A und M inhaltlich, also sozusagen „von der Sache her“, die gleichen Präferenzen haben.

Im ersten Fall werde angenommen, dass es beim oberflächlichen Kontakt mit einer Software keine Probleme gibt. M wird die Evaluation dann mit einem positiven Urteil beenden. A hingegen, getrieben von dem Wunsch, sich gegen weitere Eventualitäten abzusichern, wird mit der Begutachtung der Software fortfahren und dabei u. U. auf Aspekte stoßen, die gegen eine Entscheidung für das Programm sprechen und somit zu einem anderen Urteil als M kommen. Wären M diese Aspekte bekannt gewesen, wäre er aufgrund der unterstellten identischen Präferenzstruktur zum selben Urteil wie A gekommen. Das abweichende Evaluationsverhalten hat dies aber verhindert und somit moderierenden Einfluss auf den Zusammenhang zwischen den Merkmalen der Software und ihrer Bewertung bzw. der Auswahlentscheidung genommen.

Im zweiten Fall seien Probleme beim Explorieren der Software angenommen, d. h. das Individuum komme aus irgendwelchen Gründen mit dem aktuellen Programm nicht zurecht. M wird die Evaluation mit einem negativen Urteil abbrechen, weil ihm der Aufwand zur genaueren Untersuchung und Lösung der Probleme zu hoch ist. A hingegen, einer vorschnellen Entscheidung abgeneigt, wird mehr Aufwand investieren und versuchen, den Problemen auf den Grund zu gehen. Vielleicht ist die Software ja sehr gut und muss nur anders bedient werden, als er es gewohnt ist. Falls dies zum Erfolg führt, wird A trotz gleicher Präferenzstruktur wie M, moderiert durch abweichendes Evaluationsverhalten, zu einem anderen Urteil über dieselbe Software kommen.

Zur Einordnung des beschriebenen Moderatoreffekts in die bestehende Theorie bietet sich das Technology Acceptance Model (TAM, Davis et al., 1989; Davis, 1989; vgl. Abschnitt 3.4.1) an. In diesem Modell wird davon ausgegangen, dass die Absicht, eine Technologie, z. B. eine Software, zu benutzen, von deren wahrgenommener Nützlichkeit (U) sowie Benutzungsfreundlichkeit (EoU) abhängt, welche ihrerseits durch externe Variablen, z. B. die Eigenschaften der fragliche Software, bestimmt werden. Ein moderierender Einfluss des Evaluationstyps müsste im TAM auf den Prozess wirken, durch den die externen Variablen zu den Wahrnehmungen bzw. Überzeugungen hinsichtlich U und EoU verarbeitet werden. Dieser Zusammenhang ist in Abbildung 7.3 für die Prä-Implementations-Version⁵ des revidierten TAM (Davis et al., 1989) dargestellt.

Der im Evaluationstypus eines Individuums repräsentierte Trade-off zwischen Aufwand und Genauigkeit der Evaluation steuert, wie gründlich die Eigenschaften einer Software (die externen Variablen) exploriert werden, z. B. ob ihre tatsächliche Funktionsweise und ihr ganzes Potenzial erkannt werden. Dies wiederum hat Einfluss auf deren Wahrnehmung als nützlich bzw. benutzungsfreundlich. Wenn z. B. das vorgesehene Dialogmodell durch zu oberflächliche Betrachtung nicht erkannt wird und die benötigten Funktionalitäten deshalb nur auf Umwegen abrufbar sind, sollte die Benutzungsfreundlichkeit als schlechter wahrgenommen werden, als sie es tatsächlich ist. Andererseits könnte z. B. eine eher breite als tiefe Begutachtung, wie sie von verspielten Individuen vorgenommen wird, zu einem ganz anderen Nützlichkeitsurteil führen als das fokussierte, dafür aber gründlichere Vorgehen der Aufgabenorientierten.

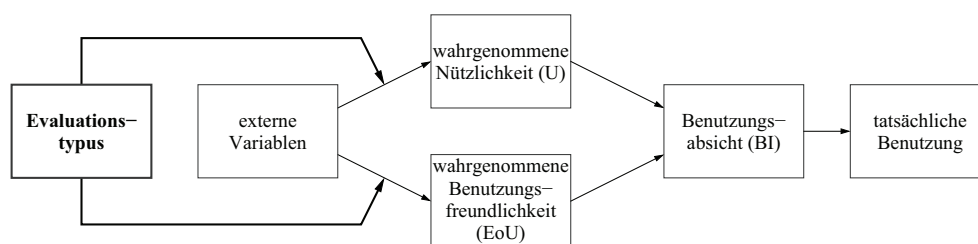


Abbildung 7.3: Der moderierende Einfluss des Evaluationstyps im revidierten Technology Acceptance Model (Prä-Implementations-Version, eigene Darstellung nach Szajna, 1996, S. 87)

⁵ Die Prä-Implementations-Version bezieht sich auf die Zusammenhänge vor der tatsächlichen Benutzung der Software (vgl. Abschnitt 3.4.1). Deshalb ist nur diese hier relevant.

Zusammenfassend ist festzustellen, dass der Evaluationstypus zwar den Prozess der Ad-hoc-Softwareauswahl bestimmt und auf diese Weise auch Einfluss auf dessen Ergebnis nimmt. Jedoch ist dieser Einfluss nur indirekt und kann deshalb nicht als der eigentliche Grund (bzw. einer der Gründe) für die Entscheidung angesehen werden. Die Auseinandersetzung mit dem Datenmaterial hinterlässt vielmehr den Eindruck, dass die Gründe für die Entscheidung (neben den Eigenschaften der Software selbst) eher im individuellen als im typischen Verlauf des Evaluationsprozesses zu suchen sind. Deshalb erfolgte im zweiten Teil der qualitativen Analyse eine einzelfallweise Untersuchung auf hervorstechende Merkmale und potenzielle Schlüsselereignisse im Umgang mit den Alternativen, welche zur Erklärung der jeweiligen Auswahlentscheidung herangezogen werden können.

7.3 Ergebnisse der Qualitativen Analyse II

Das interessanteste Ergebnis dieses Teils der Analyse sind weniger die Hypothesen jeweils für sich genommen, sondern vor allem das durch H1 bis H4a-c konstruierte hierarchische System von Abhängigkeiten. Das in H1 postulierte Kriterium ist dabei unmittelbar plausibel und bedarf keiner weiteren Diskussion: Wenn es nicht gelingt, in hinreichendem Maße herauszufinden, wie ein Programm funktioniert, kommt es nicht für die Auswahl infrage, weil seine Nützlichkeit ein Mindestniveau nicht übersteigt.

Vor allem hinsichtlich des Zusammenhangs von H2 und H3 ist zu berücksichtigen, dass die in den Hypothesen verwendeten Konstrukte nicht dichotom, sondern stetig sind. Das heißt z. B., dass ein Programm nicht einfach entweder gut oder schlecht handhabbar ist. Vielmehr gibt es dabei Abstufungen, sodass es beim Vergleich zweier Optionen u. U. um Nuancen geht, die je nach Ausprägung anderer Aspekte vermutlich mehr oder weniger ausschlaggebend sind. Damit ist gemeint, dass z. B. ein kleiner Vorteil in der Ergebnisqualität, der *ceteris paribus* gemäß H2 die Entscheidung herbeiführen müsste, angesichts gewaltiger Unterschiede in der Handhabbarkeit unwesentlich wird, sodass die Ergebnisqualität als gleichwertig betrachtet und das Problem gemäß H3 entschieden wird (vgl. dazu Abschnitt 4.2.3.3 über Kontexteffekte). Insofern wird klar, dass H2 und H3 letztlich einen Trade-off zwischen Handhabbarkeit und Ergebnisqualität postulieren, in den jedoch Letztere mit wesentlich größerem Gewicht eingeht.

Obwohl H2 und H3 nahe legen, dass Handhabbarkeit irrelevant ist, solange sich eine Präferenzordnung allein aus der Ergebnisqualität ergibt, ist zudem davon auszugehen, dass die Handhabbarkeit nicht beliebig schlecht sein darf, sondern sich über einem Mindestniveau bewegen muss. Andererseits kann jedoch auch vermutet werden, dass die Konstellation „außerordentlich gute Ergebnisqualität bei außerordentlich schlechter Handhabbarkeit“ für den hier untersuchten Sachverhalt der nicht angeleiteten Erprobung (vgl. Abschnitt 4.4) nicht besonders relevant ist. Anders als bei einer Unterweisung in der Benutzung der Software, bei der einem Individuum gezeigt wird, wie es mit Schwierigkeiten zurechtkommt, ist hier nämlich davon auszugehen, dass durch schlechte Handhabbarkeit unterhalb eines Mindestniveaus das theoretisch mögliche Erzielen „außerordentlich

guter“ Ergebnisse, vielleicht sogar die bloße Erschließung des Grundnutzens praktisch verhindert wird.

Der Einschluss des Grundnutzens in die Diskussion liefert einen weiteren Anhaltspunkt dafür, dass die Hypothesen in ihrer Struktur sinnvoll sind. Bereits in Abschnitt 6.3.3 wurden Erschließungsaufwand und Handhabbarkeit auf die Verfügbarkeit passender Schemata zurückgeführt (zur Schematheorie s. J. R. Anderson, 2007 sowie Abschnitt 4.2.1). Eine „außerordentlich schlechte“ Handhabbarkeit sollte demnach bedeuten, dass die Bedienung der betreffenden Software so sehr von den Schemata des Individuums abweicht, dass dieses überhaupt nicht damit zurechtkommt und bereits am Erschließen des Grundnutzens scheitert. Das würde aber bedeuten, dass H2 und H3 überhaupt nur dann relevant werden, wenn das oben bereits angesprochene Mindestniveau von Handhabbarkeit gegeben ist.

Mit diesem Hintergrund kann auch H5 theoretisch unterfüttert werden, da Vertrautheit hier letztlich nichts anderes bedeutet, als dass durch den Reiz, den die Konfrontation mit einem Programm darstellt, ein passend erscheinendes Schema aktiviert wird. Erweist sich dieses dann als tatsächlich anwendbar, d. h. funktioniert die Software mit nur unwesentlichen Abweichungen entsprechend der Vorgaben des Schemas, stellt sich gemäß der Argumentation in Abschnitt 6.3.3 eine neutrale bis positive Wahrnehmung hinsichtlich des Erschließungsaufwandes bzw. der Handhabbarkeit ein (Mandler, 1982; Meyers-Levy & Tybout, 1989).

Die Gruppe der H4-Hypothesen postuliert, dass über das Notwendige hinausgehende Aspekte erst dann relevant werden, wenn die angemessene Lösbarkeit der eigentlichen Aufgabe sichergestellt ist. Dieser Zusammenhang ist unmittelbar plausibel, soll aber dennoch mit einer Anmerkung ergänzt werden. Wie schon oben bei der Diskussion von H2 und H3 erläutert, ist auch hier die Stetigkeit der Konstrukte zu berücksichtigen und insofern davon auszugehen, dass die Wahrnehmung von Unterschieden zwischen den Optionen von den Ausprägungen der Attribute abhängen. Das heißt, dass große Unterschiede z. B. beim Zusatznutzen kleine Unterschiede beim Grundnutzen verschwinden lassen, so dass die betreffenden Optionen diesbezüglich als gleichwertig wahrgenommen werden. In der Konsequenz können also kleine Nachteile bei einem übergeordneten Kriterium durch große Vorteile bei einem untergeordneten ausgeglichen werden und umgekehrt.

Das in den Hypothesen H1 bis H4 postulierte „Primat des Notwendigen“ spiegelt sich auch in H6, wonach aufgabenbezogener Zusatznutzen schwerer wiegt als aufgabenfremder. Hier besteht jedoch kein Diskussionsbedarf, da es ohne weiteres plausibel ist, dass ein für die aktuelle Aufgabe relevanter und insofern sicherer Zusatznutzen gegenüber einem ungewissen bevorzugt wird. Ungewiss deshalb, weil er für das aktuelle Problem nicht benötigt wird, sondern lediglich die Bereitschaft für die Bewältigung zukünftiger Aufgaben erhöht.

Ein Bezug der zentralen, in H1 bis H3 ausgedrückten Analyseergebnisse zur bestehenden Theorie lässt sich, wie schon für die Evaluationstypen (vgl. Abschnitt 7.2.2), über das Technology Acceptance Model (TAM; Davis, 1989; Davis et al., 1989; vgl. Abschnitt 3.4.1) herstellen. Das TAM soll zwar ursprünglich nur die Akzeptanz einer einzigen, bestimmten Technologie erklären, jedoch konnte Szajna (1994) am Beispiel von Datenbanksoftware zeigen, dass mit diesem Modell auch die Auswahl aus einer Menge

erklärt werden kann (vgl. Abschnitt 3.4.2), indem jeder Option ein Score für die zentralen Konstrukte Nützlichkeit und Benutzungsfreundlichkeit zugeordnet wird. Gewählt wird die Option mit dem höchsten Score.

Ins Auge fällt dabei die semantische Ähnlichkeit der verwendeten Konstrukte, deren Ausprägungen Einfluss auf die Akzeptanz einer Technologie nehmen: Grundnutzen und Ergebnisqualität scheinen der Nützlichkeit im TAM zu entsprechen, welche definiert ist als das Ausmaß, in dem eine Person glaubt, die Verwendung einer bestimmten Software seine Leistungen verbessern wird (Davis, 1989). Dies kann aber nur dann der Fall sein, wenn der Grundnutzen erschlossen wurde und die Ergebnisqualität zufrieden stellend ist. Ein Programm, dessen Funktionsweise unklar bleibt, ist ebenso wenig nützlich wie eines, das unbrauchbare Ergebnisse liefert. Es ist allerdings zu beachten, dass hier weniger der objektiv angebotene Grundnutzen gemeint ist, sondern vielmehr jener, der praktisch abgerufen werden konnte. Dies ist ein wichtiger Unterschied zwischen der durch das TAM behandelten angeleiteten und der hier untersuchten, nicht angeleiteten Erprobungssituation (vgl. Abschnitt 4.4). Dieser Punkt wird weiter unten nochmals aufgegriffen und mit seinen Konsequenzen diskutiert.

Die Handhabbarkeit hat einen Bezug zur Benutzungsfreundlichkeit im TAM, welche definiert ist als Ausmaß, in dem eine Person glaubt, dass die Verwendung eines bestimmten Systems frei von Anstrengung ist (ebd.). Diese Definition entspricht recht genau dem, was hier als Handhabbarkeit bezeichnet wurde, nämlich Aspekte wie z. B. Kontrollierbarkeit, Unterlassen störender Hinweise, „Ein-Klick-Lösungen“⁶, sinnvolle Automatismen usw.

Auch die hier postulierte Hierarchie „Grundnutzen vor Ergebnisqualität vor Handhabbarkeit“ findet sich im TAM wieder, da dessen Autoren aus ihren Daten schlussfolgern, dass der Nützlichkeit einer Technologie ein höherer Stellenwert zukommt als der Benutzungsfreundlichkeit: „Yet our data indicates that, although ease of use is clearly important, the usefulness of the system is even more important and should not be overlooked. Users may be willing to tolerate a difficult interface in order to access functionality that is very important, while no amount of ease of use will be able to compensate for a system that doesn't do a useful task“ (Davis et al., 1989, S. 1000). Insofern sind die Ergebnisse der vorliegenden Studie in den bis hierher diskutierten Aspekten mit dem TAM kompatibel und legen nahe, dass dieses Modell auch für die hier untersuchte nicht angeleitete Erprobungssituation Gültigkeit hat.

Einer weiter gehenden Diskussion bedarf die Rolle des erschlossenen Grundnutzens, da hier bei näherer Betrachtung zwei Aspekte von Bedeutung sind. Der Ex-post-Umstand, ob dessen Erschließung gelungen ist oder nicht, hat ganz klar mit der Nützlichkeit der Software (*U*) zu tun: Falls ja, ist das Programm zumindest prinzipiell nützlich, falls nein, dann ist es ebenso prinzipiell nicht nützlich. Der Weg dorthin sollte dagegen zunächst erst einmal durch die Benutzungsfreundlichkeit (*EoU*) bzw. das, was die Person in dieser Hinsicht wahrnimmt, determiniert sein: Bekommt sie die Software nicht unter Kontrol-

⁶ Damit ist gemeint, dass eine Funktionalität mit nur einem Mausklick abgerufen werden kann. Gegenbeispiel ist das Erzeugen eines Verbindungspfeils, indem zunächst eine Linie gezeichnet wird, die in einem zweiten Schritt mit einer Pfeilspitze zu versehen ist.

le, wird sicherlich die Benutzungsfreundlichkeit als nicht besonders gut wahrgenommen („Die Benutzerführung ist wirklich, ich möchte jetzt keinen Kraftausdruck verwenden, aber völlig für’n Arsch.“, Vp4.144). Parallel dazu kann es aufgrund derselben Umstände auch passieren, dass die Exploration des Grundnutzens scheitert, was zu negativer Wahrnehmung der Nützlichkeit führt.

Basierend auf diesen Überlegungen kann für den Fall der nicht angeleiteten Erprobung ein bisher unberücksichtigter Zusammenhang zwischen U und EoU diskutiert werden. Eine negative Wahrnehmung von EoU müsste demnach tendenziell mit negativer Wahrnehmung von U einhergehen. Eine positive Wahrnehmung von EoU muss dagegen nicht ebenso zu positiver Wahrnehmung von U führen, weil U außerdem noch den Aspekt Ergebnisqualität umfasst, die trotz aller Benutzungsfreundlichkeit schlecht sein kann. Insofern scheidet ein direkter Zusammenhang aus.

Vielmehr scheint es so, dass es durch die Benutzungsfreundlichkeit ermöglicht oder eben verhindert wird, dass die externen Variablen (die Eigenschaften der Software) ihren Einfluss auf die Wahrnehmung der Nützlichkeit entfalten können: Ist die Benutzungsfreundlichkeit so schlecht, dass es dem Individuum nicht gelingt, die Kernfunktionalitäten abzurufen, können deren Ergebnisse aussehen wie sie wollen; das Individuum wird sie nie zu Gesicht bekommen. Erst das Mindestmaß an EoU , das die Erschließung des Grundnutzens ermöglicht, lässt es zu, dass darüber hinausgehende Determinanten von U wirksam werden.⁷ Dieser Argumentation folgend, ist ein moderierender Einfluss von EoU auf den Zusammenhang zwischen externen Variablen und U zu vermuten. In Abbildung 7.4 ist dieser Einfluss in der Prä-Implementations-Version des revidierten TAM (Davis et al., 1989) dargestellt.

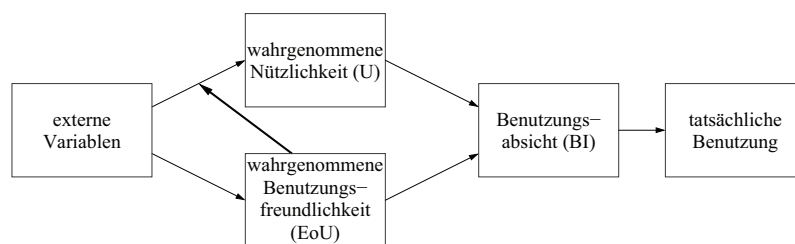


Abbildung 7.4: Der moderierende Einfluss der Benutzungsfreundlichkeit auf die Nützlichkeit im revidierten Technology Acceptance Model (Prä-Implementations-Version, eigene Darstellung nach Szajna, 1996, S. 87)

7.4 Grenzen der Untersuchung

Bei der Diskussion der Grenzen einer Untersuchung geht es um zu berücksichtigende Einschränkungen der Validität der erzielten Befunde (Hager & Spies, 1991). Das bedeutet in der gängigen Forschungspraxis häufig vor allem Überlegungen hinsichtlich der Verallgemeinerbarkeit festgestellter Hypothesenbewährungen (ebd.). Bei der vorliegenden Studie

⁷ Eine ähnliche Argumentation wurde bereits weiter oben für den Zusammenhang von H2 und H3 geführt.

stellt sich die Frage in dieser Form jedoch nicht, da ihr Ziel darin besteht, Hypothesen überhaupt erst zu entwickeln. Stattdessen ist zu diskutieren, für welche Sachverhalte diese Hypothesen gültig sein können, nachdem sie auf der Basis von Beobachtungen in einer sehr speziellen Situation zustande gekommen sind.

Wie in jeder nicht verdeckten Laboruntersuchung ist zunächst zu berücksichtigen, dass den Versuchspersonen voll bewusst war, dass sie unter Beobachtung stehen. Deshalb ist nicht auszuschließen, dass sie sich in einer privaten Situation anders verhalten hätten (O. Huber, 2005; Bortz & Döring, 2006). Dies betrifft sowohl das unmittelbar problembezogene als auch das problemunabhängige Verhalten. Mit Ersterem sind Aspekte gemeint, die dem Wunsch nach Erwartungskonformität entspringen und sich entsprechend im Streben nach Erfüllung der vermuteten inhaltlichen Erwartungen des Versuchsleiters manifestieren, wie z. B. die Breite und Tiefe der Softwareevaluation, die Gewichtung verschiedener Kriterien sowie schließlich auch die resultierende Entscheidung. Zwar wurde durch sorgfältige Entwicklung der Aufgabenstellung versucht, solche Effekte zu vermeiden bzw. zu minimieren (vgl. Abschnitt 5.3.3), jedoch kann nicht ausgeschlossen werden, dass sie in der einen oder anderen Form dennoch aufgetreten sind.

Problemunabhängiges Verhalten dagegen bezieht sich darauf, dass die bewusste Beobachtung, insbesondere in Verbindung mit dem Lauten Denken, u. U. zu Hemmungen oder aber im Gegenteil zu einem Hang zur Übertreibung führt. Dies sind allerdings Effekte, die bei einer Laboruntersuchung stets in Kauf zu nehmen sind, weil sie in der individuellen Persönlichkeit der Probanden begründet sind. So schien z. B. Vp7 recht extrovertiert zu sein und regelrecht in der geforderten Kommentierung ihres Tuns zu schwelgen. Vp10 dagegen war zwar vor und nach der Aufzeichnung sehr gesprächig, während der Aufgabenbearbeitung aber eher schweigsam. Vp5 ließ deutliche Neigungen zur Ironie erkennen.

In Verbindung mit der künstlichen Situation steht die Unsicherheit, in welchem Maße es gelungen ist, den Versuchspersonen den Eindruck zu vermitteln, dass ihre Entscheidung Konsequenzen für weitere von ihnen zu bearbeitende Aufgaben hat. So könnte es sein, dass eine Person mit wenig ausgeprägtem Leistungsmotiv entsprechend wenig motiviert war, die für sie beste Option zu identifizieren, weil die künstliche Laborsituation als Anreiz nicht ausreichte. Unabhängig davon ist es möglich, dass Probanden ihre Teilnahme an der Studie als reinen Spaß betrachteten und die gestellten Aufgaben deshalb nicht ernst nahmen. Allerdings finden sich in den Daten keine konkreten Anhaltspunkte, die vermuten lassen, dass eine der Versuchspersonen nicht ernsthaft versucht hat, das bestgeeignete Programm herauszufinden. Dennoch wäre es natürlich von Vorteil, wenn die Konsequenzen der Auswahlentscheidung einen Bezug zur realen Situation der Probanden hätten, z. B. wenn eine von Studenten auszuwählende Software für ein aktuell zu bearbeitendes Projekt eingesetzt werden soll.

Hinsichtlich der zur Auswertung verwendeten Methoden ist anzumerken, dass diese extra zum Zweck dieser Untersuchung entwickelt wurden und insofern nicht als erprobtes Instrumentarium anzusehen sind. Dies ist jedoch vor allem dem Charakter der Studie geschuldet: Wenn bisher nicht erforschte Sachverhalte qualitativ untersucht werden, erscheint es zur Auswertung des Beobachteten unvermeidbar, eigene Ansätze zu entwickeln oder vorhandene so zu erweitern und zu modifizieren, dass sie auf das neuartige Daten-

material anwendbar sind. Viel wichtiger erscheint der Umstand, dass die Auswertung zu einem großen Teil durch Interpretation von Sachverhalten erfolgte. Diese Interpretation ist notwendigerweise subjektiv und deshalb anfällig für Effekte, die in der Person des Verfassers begründet und trotz größter Sorgfalt und Gewissenhaftigkeit nicht auszuschließen sind. Eine Minderung solcher Effekte kann einfach dadurch erreicht werden, dass die Auswertung statt durch eine einzelne Person durch ein Team erfolgt. Dadurch werden persönliche Einflüsse in Gestalt abweichender Interpretationen desselben Sachverhalts offenbar und können entsprechend hinterfragt werden. Für die vorliegende Studie standen die dafür erforderlichen, angesichts des hohen Aufwands nicht unerheblichen Ressourcen jedoch nicht zur Verfügung.

Hinsichtlich der Zusammensetzung der Stichprobe ist es als Vorteil zu werten, dass es gelungen ist, Versuchspersonen unterschiedlichen Alters und mit unterschiedlichem beruflichen Hintergrund zu rekrutieren. Dennoch entspricht sie sicher nicht der relevanten Grundgesamtheit, was aber andererseits wegen des aufwendigen qualitativen Ansatzes praktisch auch nicht möglich gewesen wäre. In weiteren Studien kann die Auswertung auf der Basis der hier geleisteten Vorarbeit sowie umfangreicherer personeller Ausstattung u. U. effizienter erfolgen, was eine größere und repräsentativ zusammengesetzte Stichprobe ermöglicht. Insbesondere wäre darauf zu achten, auch Personen ohne akademischen Hintergrund zu rekrutieren.

Als letzter, aber inhaltlich sehr wichtiger Punkt ist die Anwendungsdomäne der zur Auswahl gestellten Programme zu diskutieren. Die hier verwendete Erstellung von Flussdiagrammen ist durch ein besonders hohes Maß an Interaktivität gekennzeichnet, welches der Ergiebigkeit des zu erhebenden Datenmaterials entgegen kam, aber natürlich nicht bei allen Desktop-Programmen so auftritt. Darüber hinaus liefert nicht jede Software ein so konkretes Ergebnis wie ein sicht-, speicher- und druckbares Flussdiagramm. Ein Beispiel wäre ein Programm, das die Funktionalität eines Weckers implementiert: Hier gibt es kein bleibendes Ergebnis und die Interaktion ist mit dem Einstellen der Weckzeit im Wesentlichen erledigt. Es kann (muss natürlich aber nicht) sein, dass deshalb z. B. andere Aspekte als die in Abschnitt 6.3 identifizierten eine Rolle für die Entscheidung spielen oder anders voneinander abhängen. Insofern erscheint es angebracht, die Gültigkeit der erarbeiteten Hypothesen zunächst auf solche Software zu beschränken, die ein annähernd vergleichbares Maß an Interaktivität und Ergebnispräsenz aufweisen.

7.5 Implikationen für Forschung und Praxis

Da als Ergebnis dieser Untersuchung nur Hypothesen vorliegen, die noch einer Überprüfung bedürfen, lassen sich Implikationen für die Praxis lediglich skizzieren. Die Analyse des Evaluationsverhaltens führte zu der Vermutung, dass die Art und Weise der Informationsbeschaffung zur Bearbeitung des Auswahlproblems Einfluss darauf hat, wie ein und dieselbe Software rezipiert wird. Das heißt, dass dasselbe Produkt für verschiedene Evaluationstypen an der Oberfläche u. U. unterschiedlich gestaltet sein muss, damit seine Vorteile entsprechend erkannt werden. Diese Folgerung unterscheidet sich substantiell von der Feststellung, dass Geschmäcker unterschiedlich sind, denn hier handelt es sich

nicht um eine Frage des Geschmacks, z. B. ob ein Individuum einen bestimmten Effekt lieber per Maus oder per Tastatur erzeugt. Vielmehr geht es darum, dass oder ob dieses Individuum die Evaluation nicht schon vorher abbricht, denn erst, wenn der Punkt, an dem besagter Effekt relevant wird, überhaupt erreicht wird, stellt sich die Frage, ob ihm der implementierte Mechanismus zusagt oder nicht.

Eng damit verbunden sind die zentralen Befunde aus der Analyse der Auswahlgründe. Demnach, so trivial dies auch erscheint, ist es am wichtigsten, dass es überhaupt erst einmal gelingt, den Grundnutzen zu erschließen. Dies ist ein Aspekt, der in der bisherigen Forschung zu Auswahlentscheidungen jedoch keine Beachtung gefunden hat. Stattdessen wird stets davon ausgegangen, dass einem Individuum die Art und Weise der Umsetzung der relevanten Funktionalitäten klar ist oder zweifelsfrei klar gemacht wird und es nur noch entscheiden muss, ob es mit dieser Variante zurechtkommt. Wie wichtig dieser Aspekt aber ist, zeigt sich daran, dass es den Versuchspersonen in dieser Studie nur in 39 von 57 Fällen gelungen ist, die Software im jeweils gewünschten Umfang unter Kontrolle zu bekommen. Das heißt, dass fast ein Drittel der untersuchten Programme nicht unbedingt deshalb abgelehnt wurde, weil es dem Individuum nicht zusagte, sondern weil es gar nicht erst gelungen ist, die Interaktionsregeln zu erkennen.

Verschärft wird dieses Problem dadurch, dass nur in zwei Fällen ernsthaft, wenn auch erfolglos, versucht wurde, derartige Probleme durch Nutzung der Hilfe-Funktion des jeweiligen Programms zu lösen. In der Regel wurde die Benutzung der Online-Hilfe pauschal abgelehnt (im Sinne von: „Wenn die Hilfe nötig ist, um das zu lösen, kann das Programm nichts taugen“) oder nach wenigen Sekunden als wenig Erfolg versprechend wieder abgebrochen. Automatische Hilfsangebote in Form von Popups wurden als lästig empfunden. Gelegentlich genutzt wurde dagegen das von SmartDraw angebotene Kurztutorial.

In diesen Beobachtungen wird eine Diskrepanz zwischen Art, wahrgenommenem Umfang und Qualität der Online-Hilfe einerseits und den Erwartungen der Benutzer hinsichtlich Unterstützung bzw. der Bereitschaft, diese überhaupt anzunehmen, andererseits, offenbar. Diese Diskrepanz deutet im Zusammenhang mit der nicht angeleiteten Erprobung einer Software auf ein grundlegendes Problem hin, für das neue Lösungen gefunden werden müssen. Im hier untersuchten Sachverhalt spielt das selbstständige Erkennen der Funktionsweise eine zentrale Rolle. Nur wenn dies gelingt, kommen die sonstigen Eigenschaften des Produktes, z. B. Aspekte wie Ergebnisqualität oder Benutzungsfreundlichkeit, überhaupt zum Tragen.

Um das so wichtige erfolgreiche Erschließen der Funktionsweise sicherzustellen, muss der Benutzer also unterstützt werden, aber eben in einer Weise, die er auch akzeptiert. Ein Ansatzpunkt könnte darin liegen, dass viele Benutzer das Lesen von Hilfetexten als zu aufwendig zu empfinden scheinen. Insofern könnten Videos oder ähnlich umgesetzte visuelle Demonstrationen auf bessere Akzeptanz stoßen. Eine genauere Untersuchung der konkreten Gründe für das Scheitern beim Erschließen einer Software sowie der Erwartungen an eine effektive und effiziente Unterstützung sind in jedem Fall interessante Fragen, die weiterer Forschung bedürfen.

Weitere Implikationen für die Forschung ergeben sich unmittelbar aus dem explorativen, Hypothesen generierenden Ansatz dieser Studie. Der sich hier nur vage abzeichnen-

de Zusammenhang zwischen den Eigenschaften der Software, dem Evaluationstypus und dem Auswahlresultat bedarf zunächst einer Konkretisierung auf der Basis gezielt erhobener Datenmaterialien. In einem nächsten Schritt sind die hier aus der Beobachtung eines speziellen Sachverhalts induktiv abgeleiteten Hypothesen zu überprüfen.

Der besondere Anspruch entsprechender Untersuchungen dürfte in der hohen Komplexität von Software im Allgemeinen sowie von Standardsoftware im Besonderen liegen. Letztere ist häufig durch eine Vielzahl verschiedenster Funktionalitäten geradezu überladen (vgl. Abschnitt 3.1), was neben der Unübersichtlichkeit dieses Angebots an sich zu entsprechend komplexen Benutzerschnittstellen führt. Diese Komplexität gilt es einerseits zu reduzieren, um den zu untersuchenden Sachverhalt greifbar zu machen, andererseits darf diese Reduktion jedoch nicht zu weit getrieben werden, um zu praktisch verwertbaren Ergebnissen zu kommen.

Insbesondere sind konkret messbare Operationalisierungen für hier verwendete Konstrukte wie Ergebnisqualität und Handhabbarkeit zu erarbeiten, die im Rahmen dieser qualitativen Studie zunächst nur mit intuitiven Bedeutungen versehen wurden. Ein Ansatzpunkt sind die Vorgaben des Technology Acceptance Models, nachdem bereits gezeigt wurde, dass die Hypothesen in dessen Rahmen eingeordnet werden können. Über den inhaltlichen Anspruch hinaus besteht zudem ein hoher Bedarf an personellen und materiellen Ressourcen, die in einem so komplexen Umfeld nötig sind, um diese Konstrukte systematisch zu manipulieren sowie eine größere Stichprobe zu erheben und auszuwerten.

Als besonders interessant und wichtig für den Aufbau weiterer Studien sowie die gezielte Verwertbarkeit ihrer Ergebnisse erscheint auch die Untersuchung der potenziellen Auswirkungen, die sich aus Unterschieden hinsichtlich der im vorigen Abschnitt angesprochenen Aspekte „Ausmaß der Interaktion“ und „Sichtbarkeit des Ergebnisses“ ergeben. Über inhaltliche Fragestellungen hinaus ist auch nach effizienten Möglichkeiten zu suchen, wie die bei der Ad-hoc-Evaluation und -Auswahl interessierenden Sachverhalte erhoben und ausgewertet werden können. Die aufwendigen qualitativen Auswertungen, die im Rahmen dieser explorativen Studie praktisch unvermeidbar waren, erscheinen wenig geeignet, größere Stichproben zu bearbeiten und dürften bei ihrer Anwendung zur Hypothesenprüfung zu erheblichen Validitätsproblemen führen.

Ein mittelbar mit dieser Studie verbundenes Forschungsfeld ergibt sich aus der Frage, was im Zuge der Softwarebeschaffung im Anschluss an die Ad-hoc-Auswahl geschieht. Deren Ergebnis ist schließlich noch keine finale Kaufentscheidung, schon allein deshalb nicht, weil in einigen Fällen, z. B. wenn es sich bei dem ausgewählten Produkt um Freeware handelt, gar kein Kauf stattfindet. Selbst wenn ein Preis zu entrichten ist, verbleiben nach der ersten Auswahl in der Regel 30 Tage Testzeit, bis die Entscheidung für oder gegen den Kauf fallen muss. Eine interessante Frage ist z. B., ob bzw. wie stark und auf welche Weise sich ein Individuum an das zunächst ausgewählte Produkt gebunden fühlt, d. h. ob es ein Programm auch nach dreiwöchiger Benutzung beim Auftreten eines Problems ebenso schnell verwirft, wie es das beim Erstkontakt im Rahmen der Ad-hoc-Evaluation getan hätte.

Über die hier konkret angesprochenen Punkte hinaus lässt die zunehmende Verbreitung von Hochgeschwindigkeits-Internetanschlüssen in privaten Haushalten erwarten, dass die Beschaffung von Software in noch größerem Umfang als bisher durch Endbenutzer per

Download aus dem Internet erfolgen wird. Das bedeutet, dass die Relevanz des mit dieser Studie betretenen Forschungsfeldes generell weiter steigt und auch das Thema „Software als Konsumgut“ weitere Aufmerksamkeit verdient.

8 Zusammenfassung

Durch die weite Verbreitung von Mikrocomputern (PCs) in Unternehmen, Institutionen und weiter zunehmend auch in privaten Haushalten besteht ein entsprechender Bedarf an Software für zahlreiche Zwecke, die für den Einsatz von PCs in verschiedensten Bereichen erforderlich ist. Mit der Öffnung des Internets für kommerzielle Zwecke, der Entwicklung des World Wide Web sowie der allgemeinen Verfügbarkeit und Nutzung von Breitbandanschlüssen sind die Voraussetzungen für die Distribution von Software in einer Form gegeben, die ihrem immateriellen Charakter entspricht. Insbesondere besteht die Möglichkeit, aus der Vielzahl von Lösungen, die über Downloadportale oder direkt bei den Entwicklern angeboten werden, geeignet erscheinende Programme als Testversion herunterzuladen. Diese Testversionen können, ggf. an realen Aufgaben unter Alltagsbedingungen, durch spontanes Ausprobieren auf ihre Tauglichkeit untersucht werden.

Besonderes Kennzeichen dieses Evaluationsmodus' ist das Fehlen einer klaren Spezifikation, also einer Auflistung von konkreten Anforderungen, deren Erfüllung durch die Software schrittweise abgeprüft werden kann. Vielmehr wird hier von Individuen ausgegangen, die nicht über das Fachwissen verfügen, welches für das Erstellen und Überprüfen einer Spezifikation nötig ist, und/oder angesichts des geringen Risikos der fraglichen Softwarebeschaffung nicht bereit sind, den damit verbundenen Aufwand auf sich zu nehmen. Entsprechend wird nur eine ungefähre Vorstellung davon angenommen, was das gewünschte Programm auf welche Weise leisten soll. Insofern kann die (fehlende) Spezifikation auch nicht als Leitfaden für die Evaluation dienen. Deren Verlauf wird vielmehr durch die konkrete Interaktion mit der Software bestimmt.

Das Ziel der vorliegenden Studie bestand darin, zu untersuchen, *wie* Individuen bei einer derartigen Ad-Hoc-Evaluation von Software vorgehen und *warum* sie sich für eine bestimmte Option entscheiden. In der Literatur zur Evaluation und Auswahl von Software wird das oben skizzierte Vorgehen praktisch nicht thematisiert. Vorherrschend sind präskriptive Arbeiten, die sich mit der Entwicklung von Modellen befassen, wie vor dem Hintergrund riskanter Beschaffungen (z. B. teuer oder unternehmenskritisch) zu verfahren ist, um zu der „richtigen“ Entscheidung zu kommen. Die relativ wenigen deskriptiven Studien auf diesem Gebiet beschäftigen sich in der Regel mit Analysen hinsichtlich des Einsatzes solcher Modelle sowie der Anwendung von Heuristiken aus dem Bereich der Entscheidungstheorie. Der direkte Kontakt bzw. Umgang mit einer Software, wie er beim Ausprobieren stattfindet, spielt dabei aber keine Rolle.

Eine Ausnahme bildet das Technology Acceptance Model (TAM), das basierend auf der Theorie des begründeten Verhaltens zwei Faktoren identifiziert, welche eine zentrale Rolle für die Akzeptanz oder Ablehnung einer Technologie (z. B. eines Softwaresystems) spielen: Nützlichkeit und Benutzungsfreundlichkeit. Die Wahrnehmung eines Individuums hinsichtlich dieser Faktoren kommt hier zwar u. U. durch direkte Interaktion mit

einer Software zustande, jedoch fehlt hier der Aspekt des selbstständigen, nicht angeleiteten Ausprobierens, da in allen Studien zum TAM eine Einweisung in das zu akzeptierende oder abzulehnende Programm erfolgte. Dennoch kann das TAM als theoretischer Rahmen zur Einordnung der vorliegenden Untersuchung bzw. ihrer Ergebnisse dienen.

Die Literatur zur Psychologie von Konsumentenentscheidungen befasst sich im Wesentlichen mit der Erklärung *tatsächlichen* Verhaltens und hat insofern deskriptiven Charakter. Von Bedeutung für den hier zu untersuchenden Sachverhalt sind insbesondere kognitive und emotionale Prozesse, welche für die Steuerung bewusster Entscheidungen verantwortlich sind. Wichtige kognitive Determinanten sowohl des Entscheidungsverhaltens als auch des -ergebnisses sind die Vorkenntnisse des Individuums, die Schwierigkeit des Entscheidungsproblems sowie der Kontext, in den es eingebettet ist. Emotionen spielen in der Form von Stimmungen als „Dauertönung des Erlebens“ ebenso eine Rolle, wie als Informationsquelle sowie als ggf. zu vermeidende Begleiterscheinung von schwierigen Entscheidungen.

Allerdings findet auch in den Untersuchungen zum Entscheidungsverhalten von Konsumenten der Aspekt des praktisch uneingeschränkten Ausprobierens keine Berücksichtigung, vermutlich weil er für klassische Produkte und Dienstleistungen nicht besonders relevant ist. Stattdessen werden die für die Bewertung bzw. Entscheidung verfügbaren Informations-Items in abstrakt aufbereiteter Form (typischerweise als Option×Attribut-Matrizen) bereitgestellt. Die untersuchten Konsumentenentscheidungen basieren somit nicht auf der praktischen Erprobung der zur Auswahl stehenden Produkte, sondern auf der Verarbeitung abstrakt dargebotener Informationen.

Für die vorliegende Studie wurde deshalb ein explorativer Ansatz gewählt, um einen ersten Zugang zu dem zu untersuchenden Sachverhalt zu erarbeiten. Ziel dieses Vorgehens war die Ableitung von Hypothesen auf der Basis heuristischer Beobachtungen. Das wichtigste Merkmal dieses Ansatzes ist die möglichst unvoreingenommene, ganzheitliche Erfassung eines Sachverhalts ohne Konzentration auf bestimmte, sich aus der bestehenden Theorie als relevant ergebende Aspekte. Um im Rahmen solcher Beobachtung möglichst valide Daten zu erheben, wurde versucht, im Labor eine realistische Auswahl-situation zu kreieren. Als Versuchspersonen dienten zwanzig Individuen mit unterschiedlicher Erfahrung hinsichtlich des Umgangs mit einschlägiger Software. Ihnen wurde ein zu bearbeitendes Problem vorgestellt, verbunden mit der Aufgabe, drei verfügbare Programme auszuprobieren, um anschließend zu entscheiden, welches sie dafür benutzen wollen. Als Problembereich wurde das Erzeugen von Flussdiagrammen gewählt, weil die dafür benötigten Funktionalitäten keiner Erklärung oder besonderer Vorkenntnisse bedürfen und die zum Ausprobieren der Software unternommenen Schritte gut beobachtbar sind. Zur Auswahl standen die Programme SmartDraw 6, Flowcharter 2003 und Visio.

Um die beobachteten Sachverhalte entsprechend den Fragestellungen auswerten zu können, wurden die Versuchspersonen außerdem angewiesen, während des Erprobens der Software laut zu denken. Die so geäußerten Verbalisierungen, die den Zugang zu den jeweils aktuellen und somit relevanten Kognitionen ermöglichen sollten, wurden zusammen mit den parallel dazu ablaufenden Interaktionen mit der Software als Video aufgezeichnet und in Transkripten verschriftet. Die Beobachtungen wurden in Interaktionsprotokollen formalisiert, um das Verhalten der einzelnen Versuchspersonen abstrakt fassen und somit

vergleichen zu können. Eine zentrale Rolle spielte dabei die Kategorisierung explorierter Funktionalitäten in Kern- bzw. Extrafunktionalitäten anhand ihrer Verortung in einem dreidimensionalen Raum, aufgespannt durch die Dimensionen Notwendigkeit, Zielbeitrag und Problemspezifität. Außerdem wurden Orientierungsphasen und die Würdigung von Features (Merkmale ohne Funktionalitätscharakter) erfasst.

In einem ersten Auswertungsschritt wurde nach Hinweisen auf Zusammenhänge zwischen rein quantitativ, d. h. unmittelbar und ohne tiefere inhaltliche Analyse beobachtbaren Variablen gesucht. Im Mittelpunkt der Aufmerksamkeit standen die einschlägige Erfahrung der Individuen, die Reihenfolge, in der die Programme ausprobiert wurden, der betriebene Aufwand für einzelne Programme (operationalisiert durch die Bearbeitungszeit) sowie das Auswahlresultat. Es konnten jedoch keine Auffälligkeiten festgestellt werden, die es rechtfertigen würden, hier Zusammenhänge zu vermuten.

In einem weiteren, qualitativen Auswertungsschritt ging es um die Untersuchung der von den Versuchspersonen angewendeten Evaluationsstrategien, d. h. die Klärung der Frage, *wie* Individuen beim Ausprobieren unbekannter Software vorgehen bzw. *was* sie dabei tun. Das Ziel bestand hier vor allem darin, nach typischen Verhaltensmustern zu suchen und diese zu beschreiben. Zu diesem Zweck wurden die Probanden zunächst intuitiv in vorläufige Typen eingeteilt und ausgehend davon in den Daten nach Merkmalen gesucht, durch die sich diese Typologie begründen lässt. Aus den Interaktionsprotokollen konnten folgende Kennzahlen zur Operationalisierung diskriminierender Merkmale abgeleitet werden: die Komplexität der zu Testzwecken erzeugten Abbildungen, die Anzahl der erarbeiteten Informationen über Kernfunktionalitäten, die relative Häufigkeit der wiederholten Erarbeitung von Informationen über Kernfunktionalitäten sowie die relative Anzahl erarbeiteter Informationen über Extrafunktionalitäten.

Es konnten anhand dieser Kennzahlen vier Evaluationstypen unterschieden werden: Die *Aufgabenorientierten* erzeugen komplexe Abbildungen und legen Wert darauf, die Lösbarkeit der anstehenden Aufgabe mit einer Software abzusichern. Die *Minimalisten* betreiben nur geringen Aufwand, d. h. sie testen nur das Nötigste und dies nur in geringer Tiefe. Sie vertrauen darauf, dass sich alles Andere ergeben wird, sie z. B. auch mit den nicht untersuchten Funktionalitäten zurecht kommen werden. Die *Verspielten* testen in die Breite, d. h. sie probieren gern auch Extrafunktionalitäten aus, die für die aktuelle Aufgabe gar nicht benötigt werden, und legen Wert auf individuelle Gestaltungsmöglichkeiten. Die *Unsicheren* wirken überfordert und finden nur schwer Zugang zur korrekten Benutzung der Software. Ihr Tun ist vom Zufall geprägt; häufig können sie nicht nachvollziehen oder sie vergessen, wie sie bestimmte Effekte erzeugt haben und müssen erneut explorieren.

Ein Ansatz zur Erklärung der Typologie liegt im Trade-off hinsichtlich der Erreichung der Metaziele Minimierung des Aufwands und Maximierung der Entscheidungsgenauigkeit. Demnach hätte bei den Aufgabenorientierten das Ziel, die „richtige“ Entscheidung zu treffen, größeres Gewicht, wogegen Minimalisten dem Ziel, die Entscheidung mit möglichst geringem Aufwand herbeizuführen, größere Bedeutung beimessen. Die Verspielten besetzen eine Position zwischen diesen Extremen, während die Unsicheren eine Sonderstellung einnehmen, da davon auszugehen ist, dass ihr Verhalten außer durch ihre Intention in hohem Maße durch Diskrepanzen zwischen zu bewältigender Aufgabe und individuellen Kenntnissen und Fähigkeiten bestimmt wird.

Zusammenhänge zwischen Evaluationstyp und den bereits im ersten Schritt der Auswertung erhobenen Variablen deuten sich mangels gezielter Manipulationen nur vage an. So rekrutieren sich die Unsicheren ausschließlich aus gering erfahrenen Probanden, während sich die spezifisch erfahrenen überwiegend aufgabenorientiert verhalten. Flowcharter wird überwiegend von Aufgabenorientierten gewählt, während sich keiner der Verspielten für SmartDraw entschied. Erklärungsansätze lassen sich in den Konstellationen von Aufwandsbereitschaft und spezifischen Eigenschaften der jeweiligen Programme finden, bleiben aber spekulativ und bedürfen auf der Basis der hier ermittelten Ergebnisse genauerer Untersuchung.

Eine Einordnung in die bestehende Theorie gelingt über das Technology Acceptance Model (TAM), für das in bisherigen Studien stets vollständige Information vorausgesetzt wurde und der Fall des nicht angeleiteten Ausprobierens nicht vorgesehen ist. Die Ergebnisse der hier vorgenommenen qualitativen Untersuchung erlauben die Hypothese, dass der Evaluationstyp Einfluss darauf nimmt, welche Eigenschaften einer Software in welcher Weise zur Wahrnehmung von Nützlichkeit und Benutzungsfreundlichkeit beitragen, und somit als Moderator dieses Zusammenhangs wirkt.

Der letzte, ebenfalls qualitative Auswertungsschritt diente der Untersuchung der Gründe für die Auswahl, also der Klärung der Frage, *warum* sich ein Individuum für ein bestimmtes Programm entschieden hat. Zu diesem Zweck wurde das Evaluationsgeschehen einer jeden Versuchsperson auf kritische Ereignisse hin analysiert und zur Auswahlentscheidung in Bezug gesetzt. Aus der Interpretation dessen wurden Hypothesen hinsichtlich der individuellen Gründe für die konkrete Entscheidung abgeleitet. Die Einzelfallhypothesen wurden anschließend auf Gemeinsamkeiten hin analysiert und zu einem System allgemeiner Hypothesen aggregiert.

Ergebnis dieser Analyse ist ein im Wesentlichen hierarchisch strukturiertes Hypothesensystem, demzufolge die interaktive Ad-hoc-Evaluation und -Auswahl von Software nach dem Prinzip einer lexikografischen Ordnung erfolgt. Als wichtigstes Entscheidungskriterium bzw. abstraktes Attribut wird postuliert, dass es gelingen muss, die Funktionsweise der benötigten Kernfunktionalitäten (hier als Grundnutzen bezeichnet) in wiederholbarer Weise zu erschließen. Nur, wenn dies für mehrere Optionen erfüllt ist, werden als nächste Kriterien die Ergebnisqualität und die Handhabbarkeit (in dieser Reihenfolge) relevant. Untereinander gleichwertig folgen schließlich Optik, Zusatznutzen sowie Sicherheit vermittelnde „weiche“ Faktoren (z. B. Vertrautheit).

Einen theoretischen Rahmen findet auch dieses Hypothesensystem im TAM. Das hier ermittelte Kriterium der Ergebnisqualität und des erschlossenen Grundnutzens haben einen Bezug zum Konstrukt der Nützlichkeit im TAM, die Handhabbarkeit zur Benutzungsfreundlichkeit. Auch die hier postulierte Hierarchie dieser Kriterien findet sich im TAM, was dessen diesbezügliche Gültigkeit auch für den Fall des nicht angeleiteten Ausprobierens nahe legt.

Darüber hinaus scheint jedoch ein weiterer, bisher nicht modellierter Zusammenhang zu existieren. Es wird argumentiert, dass die Erschließbarkeit des Grundnutzens eines Programms durch dessen Handhabbarkeit/Benutzungsfreundlichkeit beeinflusst wird. Die Handhabbarkeit wird dadurch auf zwei Ebenen relevant: Zum einen muss sie so beschaffen sein, dass das selbstständige Erschließen der Funktionsweise der Software gelingt,

zum anderen so, dass diese Funktionsweise als einfach, effizient usw. empfunden wird. Nur dieser zweite Aspekt ist bisher im TAM berücksichtigt. Im Sinne des ersten Aspekts ermöglicht oder verhindert die Benutzungsfreundlichkeit die Erarbeitung eines „wahren“ Urteils über die Nützlichkeit. Es wird deshalb für den hier untersuchten Sachverhalt ein moderierender Einfluss der Benutzungsfreundlichkeit auf die Nützlichkeit vermutet.

Aus den Ergebnissen dieser Studie ergeben sich aufgrund ihres explorativen Charakters zunächst vor allem Implikationen für die weitere Forschung. Die sich hier andeutenden Zusammenhänge sind zum einen gezielt zu konkretisieren, zum anderen auf ihre Generalisierbarkeit zu überprüfen. Dies gilt insbesondere hinsichtlich des Anwendungsgebietes der Software, das hier speziell für die Zwecke dieser Untersuchung ausgewählt wurde. Als interessantes Nebenfeld hat sich aus den Beobachtungen die Frage ergeben, welche Form der Unterstützung Individuen bei der selbstständigen Exploration unbekannter Software anzunehmen bereit sind.

A Materialien

Faltblatt „Flussdiagramme“

Dieses Faltblatt wurde den Versuchspersonen bei Ihrer Rekrutierung zur inhaltlichen Vorbereitung auf ihre Teilnahme am Experiment ausgehändigt. Es informiert über die wichtigsten Syntaxelemente, deren korrekte Verbindung zu einem Flussdiagramm, verweist auf ästhetische Aspekte und gibt mehrere Beispiele. Das Faltblatt besteht aus vier A5-Seiten, deren Inhalt auf den Seiten 174 bis 177 reproduziert ist.

Faltblatt „Über das Experiment“

Dieses Faltblatt wurde den Versuchspersonen bei Ihrer Rekrutierung zur Vorbereitung auf ihre Teilnahme am Experiment sowie zur Erhebung ihres fachlichen Hintergrunds ausgehändigt. Es informiert über den Hintergrund des Experimentes sowie die Methode des Lauten Denkens. Erfragt werden Informationen über das allgemeine Verhältnis der Versuchsperson zum Computer, ihre Erfahrung mit dem Erstellen von Flussdiagrammen sowie mit der Entwicklung und Benutzung von Software. Außerdem soll eine Gewichtung von Kriterien zur Beurteilung von Software vorgenommen werden. Das Faltblatt besteht aus vier A5-Seiten, deren Inhalt auf den Seiten 178 bis 181 reproduziert ist.

Faltblatt „Arbeitsanweisung“

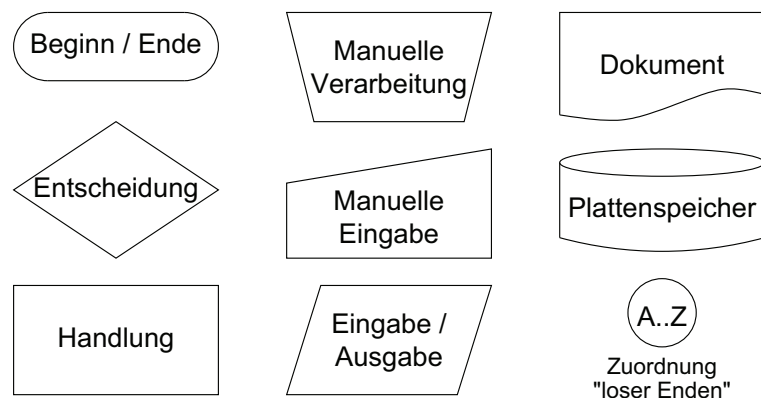
Dieses Faltblatt diente als Leitfaden durch den Ablauf des Experimentes und stand den Versuchspersonen ständig als Vorlage und Arbeitsblatt zur Verfügung. Es enthält die Motivation sowie drei konkrete Aufgabenstellungen: Aufgabe 1 zum Aufwärmen und Üben des Lauten Denkens, Aufgabe 2 zum Zeichnen mit Microsoft Word sowie Aufgabe 3 zum Auswählen eines geeignet erscheinenden Programms. Außerdem ist ein Dummy-Fragebogen in Form eines semantischen Differenzials mit Items zur Softwarebewertung enthalten. Das Faltblatt besteht aus vier A5-Seiten, deren Inhalt auf den Seiten 182 bis 185 reproduziert ist.

Flussdiagramme

Vielen Dank, dass Sie bereit sind, an dieser Untersuchung teilzunehmen. Neben der Ihnen bereits bekannten Voraussetzung, grundsätzlich einen Windows-PC bedienen zu können, benötigen Sie ein wenig Grundwissen über sogenannte Flussdiagramme.

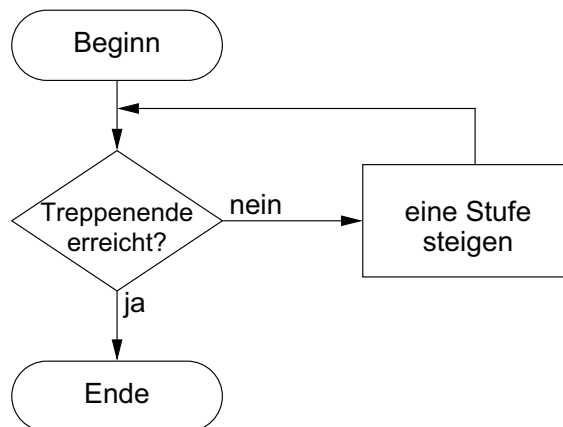
Das ist nicht weiter kompliziert und wahrscheinlich kennen Sie solche Diagramme bereits. Aber wir wollen sicher gehen, dass alle Beteiligten das gleiche Verständnis bezüglich der Thematik haben. Bitte nehmen Sie sich deshalb kurz die Zeit, dieses Faltblatt zu studieren.

Flussdiagramme sind ein einfaches und intuitiv verständliches Notationsmittel zur grafischen Darstellung von Abläufen oder (Geschäfts-) Prozessen. Sie bestehen aus miteinander verbundenen geometrischen Figuren mit festgelegter Bedeutung. Die wichtigsten dieser Symbole sind:



Bei der Verwendung werden die Symbole mit dem jeweiligen Anwendungsfall entsprechenden Texten beschriftet. Durch Pfeile werden sie zu einem Ablauf verbunden. Jeder Pfeil muss sowohl an seinem Anfang als auch an seinem Ende mit einem Symbol oder einem anderen Pfeil verbunden sein. Es darf keine isolierten Symbole oder Pfeile geben.

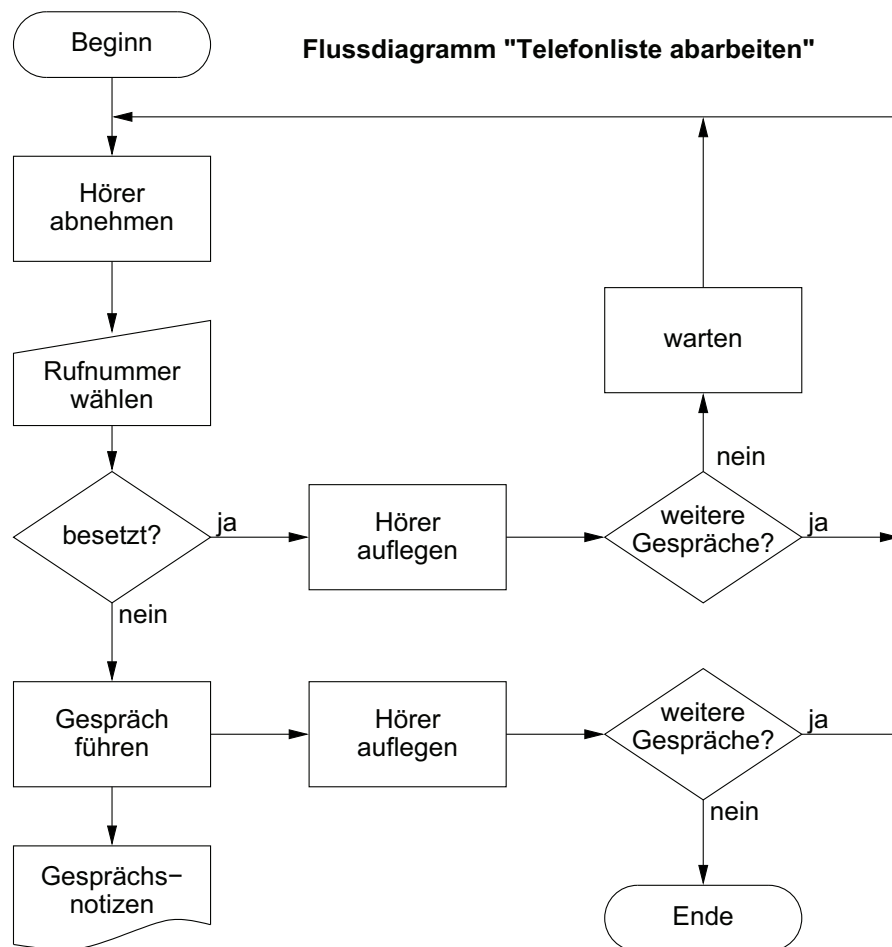
Ein ganz einfaches Beispiel ist ein Flussdiagramm für den Prozess „Treppe steigen“. Nach dem Eintritt in den Ablauf am „Beginn“ ist zu entscheiden, ob man bereits das Ende der Treppe erreicht hat. Falls dem nicht so ist (Abzweig „nein“), führt der weitere Ablauf zur Handlung „eine Stufe steigen“. Nachdem diese Handlung ausgeführt ist, kommt man erneut zu der Entscheidungsstelle. Sobald die Frage nach dem Treppenende mit „ja“ beantwortet werden kann, führt der Fluss zum „Ende“ des Ablaufs.

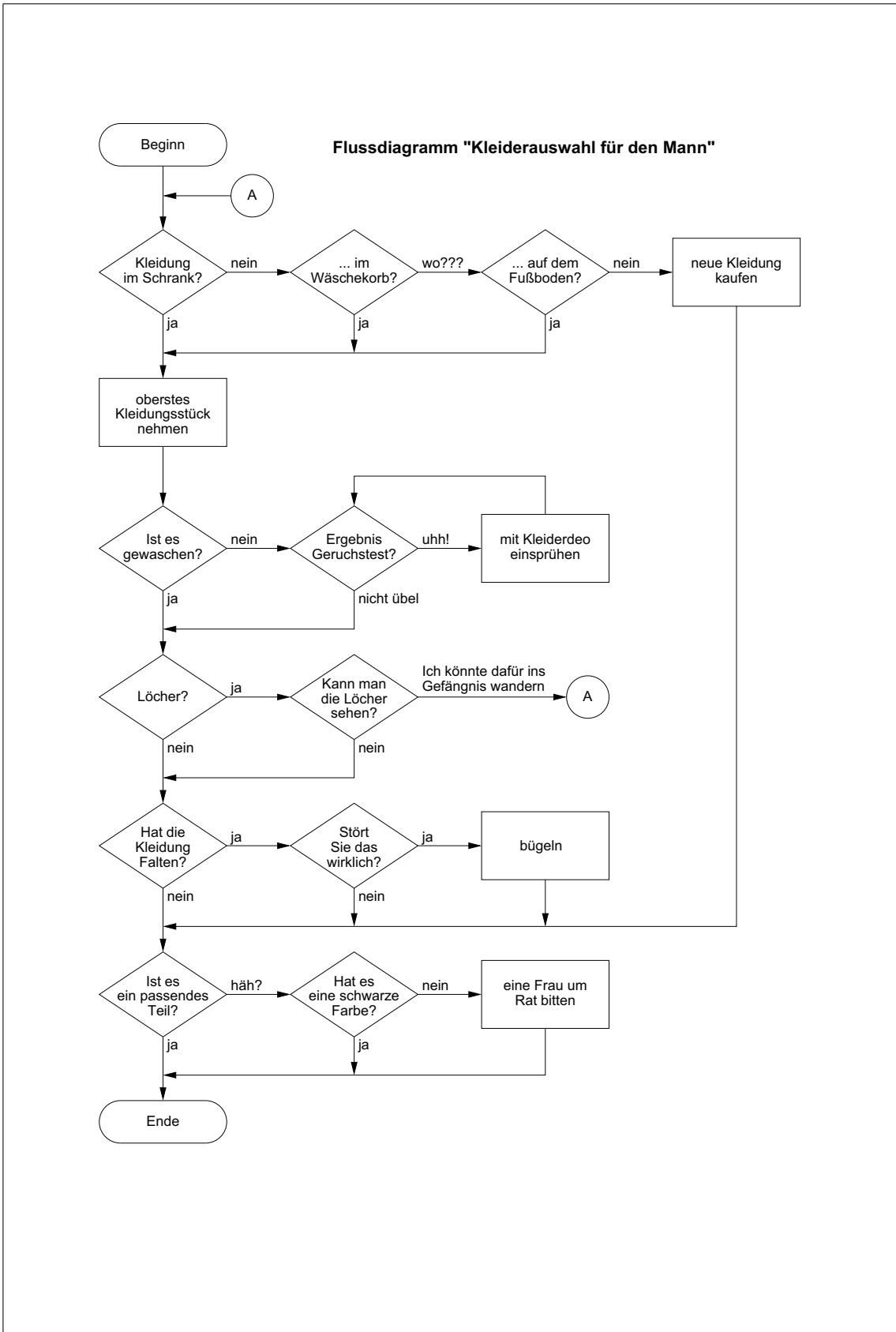


Natürlich sind Flussdiagramme nicht immer so klein und übersichtlich. „Echte“ Exemplare können mitunter ganze Poster oder „Tapetenrollen“ füllen bzw. müssen auf mehrere Blätter aufgeteilt werden. Auf den folgenden Seiten finden Sie noch zwei etwas umfangreichere, aber immer noch recht übersichtliche Exemplare.

Versuchen Sie zur Vorbereitung, ein eigenes Flussdiagramm für eine Ihrer täglichen Verrichtungen (z. B. Einkaufen, im Internet surfen, Auto fahren) zu entwickeln und per Hand oder mit dem Computer zu zeichnen. Ihrer Kreativität sind, insbesondere in Bezug auf die Detailliertheit, kaum Grenzen gesetzt.

Um ein auch optisch ansprechendes Ergebnis zu erzielen, sollte immer darauf geachtet werden, dass die Symbole gleich groß und zueinander ausgerichtet sind, also in einer Flucht stehen, und dass Verbindungslinien präzise mit den Symbolen verbunden sind.





Über das Experiment

Zunächst noch einmal vielen Dank für Ihre Bereitschaft, an diesem Experiment teilzunehmen. Forschung bedeutet Fortschritt, und Sie tragen heute einen Teil dazu bei. Das Ziel dieser Untersuchung besteht darin, herauszufinden, wie verschiedene Menschen mit ihnen unbekannter Software zurechtkommen und sich ein Urteil darüber bilden. Die Ergebnisse sollen dazu beitragen, Software so zu bauen, dass sich verschiedene Benutzer möglichst leicht und schnell den Funktionalitätsumfang und die Bedienung eines Produktes erschließen können. Damit wären sie in der Lage, mit wenig Aufwand die Eignung der Software für bestimmte Probleme zu bewerten. Sowohl Softwarehersteller als auch -benutzer würden davon profitieren.

Um verfolgen zu können, was Ihnen bei der Benutzung verschiedener Programme durch den Kopf geht, sollen Sie all das laut aussprechen. Das ist gar nicht so unnatürlich: Sie haben sicher schon einmal an sich selbst beobachtet, dass Sie bewusst „vor sich hin denken“ oder gar murmeln, wenn Sie z. B. etwas suchen oder eine Denkaufgabe lösen. Genau das ist es, was Lautes Denken ausmacht. Sprechen Sie einfach alles aus, was Ihnen so durch den Kopf geht, so abwegig es Ihnen vielleicht auch erscheint.

Dabei ist es egal, ob das nun „normales“ Nachdenken, wachgerufene Erinnerungen, aktuelle Empfindungen, gedankliche Vergleiche oder sonst etwas sind. Es ist grundsätzlich alles relevant und interessant, denn es tritt während des aktuellen Handelns auf und steht deshalb im Zusammenhang damit.

Versuchen Sie einfach einmal bei einer Tätigkeit im Alltag, für jede Ihrer Handlungen zu kommentieren,

- was Sie gerade **vorhaben**,
- welches Verhalten Sie dabei von Ihrem Handlungsobjekt **erwarten**, und
- wie Sie das tatsächliche Geschehen **werten**.

Der nun folgende Fragebogen soll dazu dienen, einen Überblick über Ihre Erfahrungen mit dem Computer zu gewinnen. Das ist für diese Untersuchung wichtig, um Ihr Vorgehen bei der Lösung verschiedener Probleme besser verstehen zu können.

Bitte kreuzen Sie an, welche der folgenden Aussagen am ehesten auf Sie zutrifft bzw. zutreffen. Mehrfachnennungen sind möglich.

- Ich benutze den Computer sehr gern, weil ich den Umgang damit spannend und faszinierend finde.
- Ich benutze den Computer sehr gern, weil es mir Spaß macht.
- Ich benutze den Computer so oft wie möglich, weil ich glaube, dass er meine Arbeit erleichtert.
- Ich benutze den Computer so oft wie möglich, weil ich damit in der Regel bessere Ergebnisse erziele.
- Ich habe keine besondere Zu- oder Abneigung gegenüber dem Computer. Er ist ein Werkzeug, und wenn es nötig ist, benutze ich ihn eben.
- Ich benutze Computer nur, wenn es unbedingt nötig ist. Ich finde es lästig, dass nichts mehr ohne Computer zu funktionieren scheint.
- Ich benutze Computer ungern, weil ich bei seiner Benutzung unsicher und langsam bin.
- Ich benutze Computer ungern, weil ich Angst habe, Fehler zu machen.
- Ich benutze Computer nur, wenn es sich nicht vermeiden lässt, weil er mir ein bisschen unheimlich vorkommt.
- Ich finde meine Einstellung gegenüber dem Computer in den vorgenannten Punkten nicht oder nicht vollständig wieder und würde sie eher bzw. ergänzend wie folgt beschreiben:

Bitte beantworten Sie folgende Fragen:

Haben Sie zuvor schon einmal Flussdiagramme gezeichnet?

- nein ja, aber nur fremde abgezeichnet ja, auch selbst entworfen

Falls ja, welche Software haben Sie dafür verwendet?

Welche **Ihnen bekannte** Software würden Sie heute am ehesten zum Zeichnen eines Flussdiagramms verwenden?

Bitte geben Sie einen kurzen inhaltlichen Überblick, in welcher Form und welchem Umfang Sie bisher beruflich und privat Erfahrungen mit der Entwicklung bzw. Benutzung von Software gemacht haben. Verfolgen Sie dazu am besten Ihren Lebenslauf und vergessen Sie dabei nicht die Einschätzung Ihres aktuellen Standes. Wenn Sie möchten, können Sie für Ihre Beschreibung folgende Kürzel verwenden, die Sie aber bitte durch kurze **Kommentare inhaltlich konkretisieren**:

Software-Entwicklung	Software-Benutzung
E0 keine Erfahrung	B0 keine Erfahrung
E1 mal davon gehört	B1 mal eine Einführung erhalten
E2 selbst programmiert	B2 gelegentl. Nutzung weniger Programme
E3 Projekterfahrung als Kunde	B3 regelm. Nutzung weniger Programme
E4 Projekterfahrung als Entwickler	B4 regelm. Nutzung vieler Programme

Bitte vergeben Sie für jede der unten stehenden Dimensionen zur Beschreibung wahrgenommener Eigenschaften von Software Punkte, je nachdem wie wichtig Ihnen deren Ausprägung jeweils ist. Sie können auch 0 Punkte vergeben. Verschiedene Eigenschaften dürfen die gleiche Punktzahl erhalten, wenn Sie Ihnen gleich wichtig erscheinen.

Beachten Sie aber, dass Ihnen insgesamt **nur 12 Punkte** zur Verfügung stehen, die Sie entsprechend Ihrer Präferenzen verteilen können. Es geht vor allem um das **Verhältnis der Punktzahlen** zueinander. Sie müssen also abwägen und Entscheidungen treffen.

Die folgenden zwölf Kreise können Sie als **Ausfüllhilfe** benutzen, indem Sie die bereits vergebenen Punkte abstreichen, sodass Sie leicht den Überblick behalten, wie viele Punkte noch zur Verfügung stehen.



Wie wichtig ist es Ihnen, ob eine betriebliche Anwendungssoftware ...

	Punkte
... übersichtlich oder verwirrend ist?	<input type="text"/>
... ermüdend oder fesselnd auf Sie wirkt?	<input type="text"/>
... Ihnen vertraut vorkommt?	<input type="text"/>
... sich gut kontrollierbar oder widerspenstig anfühlt?	<input type="text"/>
... insgesamt (inhaltlich und optisch) attraktiv ist?	<input type="text"/>
... innovativ oder gewöhnlich erscheint?	<input type="text"/>
... einen professionellen Eindruck macht?	<input type="text"/>
... einfach oder kompliziert wirkt?	<input type="text"/>

Arbeitsanweisung

Ihre finale Aufgabe wird es sein, aus mehreren Programmen zum Zeichnen von sog. Flussdiagrammen zunächst eines **auszuwählen**, um damit anschließend **möglichst effizient** eine vorgegebene Abbildung zu erstellen. Durch die Beschäftigung mit den Programmen werden Sie in der Lage sein, auf der Basis Ihres persönlichen Wertesystems jeweils eine Einschätzung der Software abzugeben.

Um verfolgen zu können, wie Sie die Benutzung der Software erleben, welche Befindlichkeiten sich bei Ihnen einstellen und wie Sie schließlich zu Ihren Einschätzungen kommen, sollen Sie kontinuierlich laut mitdenken. Sprechen Sie laut aus, was Ihnen gefällt, was Ihnen missfällt, was Sie überrascht usw.

Es wird Ihnen jetzt ein **Video** gezeigt, das Ihnen einen Eindruck vermitteln soll, was bzgl. des lauten Denkens von Ihnen erwartet wird.

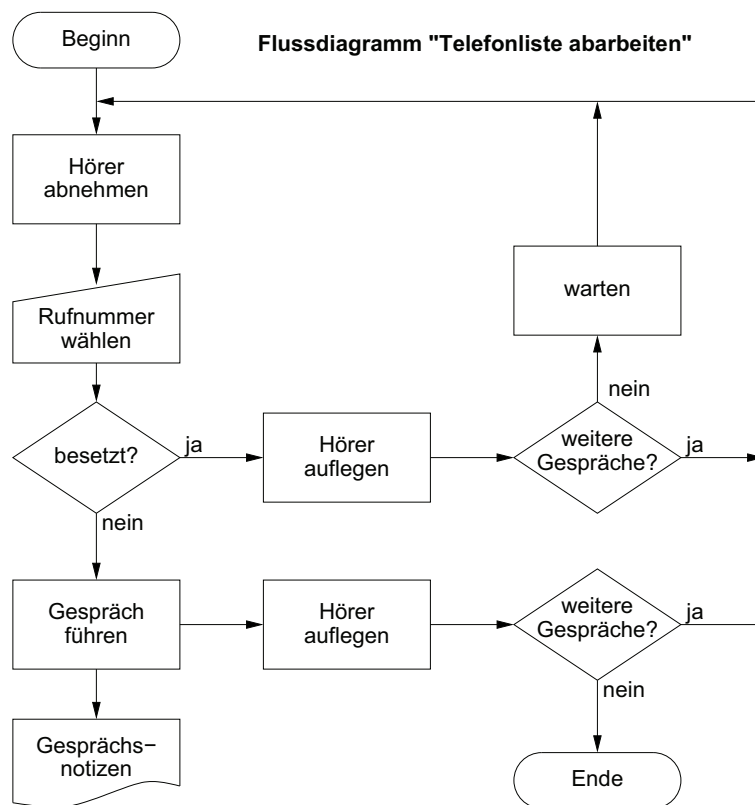
Aufgabe 1

Zum Aufwärmen und Ausprobieren versuchen Sie bitte, laut denkend folgende Aufgabe zu lösen: *Vater, Mutter und Kind sind zusammen 80 Jahre alt. Der Vater ist doppelt so alt wie das Kind, und die Mutter ist genauso alt wie der Vater. Wie alt ist das Kind?* Sie dürfen dabei gern auch etwas aufschreiben; Hauptsache, Sie sprechen alles aus, was Ihnen durch den Kopf geht und lassen den Versuchsleiter an Ihren Überlegungen teilhaben.

Aufgabe 2

Um das Zeichnen von Flussdiagrammen mit einer nicht spezialisierten Software auszuprobieren und sich weiter ans Laute Denken zu gewöhnen, erstellen Sie bitte die unten stehende Abbildung mithilfe der Ihnen bekannten Zeichenfunktionen von Microsoft Word. Sie haben dafür **max. 30 min Zeit**. Achten Sie darauf, dass Ihre Abbildung **optisch ansprechend** ausfällt.

Versuchen Sie wieder, alles zu erzählen, was Ihnen bei der Arbeit einfällt oder durch den Kopf geht: Was funktioniert gut, was ist lästig, was begeistert Sie, was treibt Sie in den Wahn usw. Beobachten und beschreiben Sie dabei auch, **wie Sie mit der Aufgabe an sich zurecht kommen** und **welche besonderen Unterstützungen** Sie sich bei einer Software, die auf das Erstellen von Flussdiagrammen spezialisiert ist, wünschen würden.



Aufgabe 3

Diese Aufgabe besteht aus zwei Teilen. Wie Sie bereits wissen, sollen Sie mit einem darauf spezialisierten Programm ein Ihnen noch **unbekanntes** Flussdiagramm zeichnen, das etwa den **Umfang** des Beispiels „Kleiderauswahl für den Mann“ hat. Dabei geht es darum, möglichst **schnell und präzise** zu einem ansprechenden Ergebnis zu kommen.

Zunächst aber sollen Sie sich aus verschiedenen Alternativen **in aller Ruhe** ein Programm **aussuchen**, mit dem Sie diese Aufgabe bearbeiten wollen. Auf dem Computer sind verschiedene Programme zum Zeichnen von Flussdiagrammen installiert. Finden Sie heraus, welches dieser Programme Ihnen am besten gefällt. Es gibt jetzt noch **keine zeitliche Beschränkung**. Gehen Sie also sorgfältig und gründlich vor. Wenn Sie sich nicht entscheiden können, berücksichtigen Sie ruhig auch, welche zusätzlichen nützlichen Funktionalitäten die Produkte anzubieten haben. Wer weiß, wofür Sie das nachher brauchen können . . .

*Stellen Sie aber auf jeden Fall sicher, dass Sie **alle benötigten Funktionen** abrufen können und gut mit der **Bedienlogik** zurechtkommen.*

Sie werden gebeten werden, Ihre Entscheidung zu **begründen** bzw. zu erklären, warum Sie die anderen Kandidaten weniger gut finden. Außerdem sollen Sie für jedes der Programme das umseitige **Bewertungsprofil** ausfüllen. **Wie** Sie jedoch bei Ihrer Untersuchung der Alternativen im Einzelnen vorgehen, **ist völlig Ihnen selbst überlassen**; ebenso, ob Sie sich nacheinander oder parallel mit den einzelnen Programmen beschäftigen. Wenn Sie möchten, dürfen Sie dabei auch Notizen machen. Spielen Sie ungehemmt mit der Software herum; Sie können nichts kaputt und keine Fehler machen: Auf dem Prüfstand steht hier die Software, nicht Sie!

Der Versuchsleiter soll jetzt nicht mehr in Ihre Handlungen eingreifen, sondern passiver Beobachter sein. Vergessen Sie aber bitte nicht, wieder die ganze Zeit **laut mitzudenken**. Lassen Sie den Versuchsleiter an Ihren Befindlichkeiten teilhaben und sprechen Sie laut aus, was Ihnen gefällt, was Ihnen missfällt, was Sie überrascht usw.

B Versuchspersonen

Vp	Geschlecht	Alter	Beruf	Fach/Branche	Erfahrung
Vp1	m	25	Student	Business Informatics	spezifisch
Vp2	w	25	Studentin	Sonderpädagogik	gering
Vp3	m	28	Dipl.-Kfm.	Unternehmensberatung	unspezifisch
Vp4	m	28	Dipl.-Wi.-Inf.	Softwareentwicklung	spezifisch
Vp5	m	38	Dipl.-Vw.	Softwareentwicklung	spezifisch
Vp6	w	26	Studentin	Wirtschaftsinformatik	gering
Vp7	m	26	Student	Business Informatics	spezifisch
Vp8	w	26	Dipl.-Kffr.	Personalwesen	unspezifisch
Vp9	m	28	Student	Business Informatics	spezifisch
Vp10	m	26	Student	Wirtschaftsinformatik	spezifisch
Vp11	m	23	Student	Betriebswirtschaft	unspezifisch
Vp12	w	27	Dipl.-Wi.-Inf.	Wiss. Mitarbeiterin	spezifisch
Vp13	w	31	Dipl.-Kffr.	Wiss. Mitarbeiterin	unspezifisch
Vp14	w	24	Studentin	Betriebswirtschaft	gering
Vp15	m	29	Student	Betriebswirtschaft	gering
Vp16	m	27	Student	Betriebswirtschaft	unspezifisch
Vp17	m	28	Dipl.-Kfm.	Wiss. Mitarbeiter	unspezifisch
Vp18	w	31	Dipl.-Kffr.	Wiss. Mitarbeiterin	spezifisch
Vp19	m	33	Dipl.-Kfm.	Wiss. Mitarbeiter	unspezifisch
Vp20	w	28	Dipl.-Kffr.	Wiss. Mitarbeiterin	unspezifisch

Tabelle B.1: Hintergrundinformationen zu den Probanden

C Quantifizierung der Interaktionsprotokolle

K_{px}	Komplexität des mit einem Programm erzeugten Flussdiagramms
K	Anzahl der eingeholten Informationen über Kernfunktionalitäten eines Programms
K'	Anzahl der wiederholt eingeholten Informationen über Kernfunktionalitäten eines Programms
t_K	für die Exploration von Kernfunktionalitäten eines Programms aufgewendete Zeit in s
X	Anzahl der eingeholten Informationen über Extrafunktionalitäten eines Programms
t_X	für die Exploration von Extrafunktionalitäten eines Programms aufgewendete Zeit in s
R	Anzahl der Rahmenwahrnehmungen sowie -explorationen bei der Evaluation eines Programms
t_R	für Rahmenexplorationen bei der Evaluation eines Programms aufgewendete Zeit in s
O	Anzahl der Orientierungsphasen bei der Evaluation eines Programms
t_O	für Orientierungsphasen bei der Evaluation eines Programms aufgewendete Zeit in s
t_{ges}	für die Evaluation eines Programms insgesamt aufgewendete Zeit

SmartDraw											
Vp	Kpx	K	K'	t _K	X	t _X	R	t _R	O	t _O	t _{ges}
Vp1*	12	11	0	208	1	12	0	0	0	0	540
Vp2*	3	8	3	410	1	13	2	57	0	0	850
Vp3	7	7	0	306	0	0	2	60	2	102	900
Vp4	5	11	0	423	1	7	1	37	1	54	835
Vp5*	7	5	0	143	1	23	0	0	0	0	340
Vp6	3	8	1	162	1	93	2	57	1	17	420
Vp7	0	4	0	131	3	95	1	84	1	14	380
Vp8	0	5	0	295	0	0	0	0	0	0	410
Vp9	1	4	0	179	0	0	0	0	0	0	300
Vp10	4	5	0	59	0	0	2	173	1	35	390
Vp11	0	4	0	78	0	0	0	0	0	0	160
Vp12	6	8	0	188	0	0	0	0	0	0	255
Vp13	0	2	0	210	0	0	1	21	0	0	320
Vp14	1	6	3	852	0	0	0	0	0	0	930
Vp15	2	8	3	312	2	7	2	45	3	68	695
Vp16	4	6	0	230	0	0	0	0	0	0	510
Vp17	1	5	1	260	2	49	1	51	1	11	435
Vp18	3	7	1	250	4	168	1	231	2	68	930
Vp19	3	4	0	–	0	–	0	–	0	–	235
Vp20	7	8	0	255	0	0	0	0	0	0	460

Tabelle C.1: Quantifizierung der Interaktionsprotokolle für SmartDraw (* Versuchsperson mit nur zwei Optionen)

Flowcharter											
Vp	Kpx	K	K'	t_K	X	t_X	R	t_R	O	t_O	t_{ges}
Vp1*	–	–	–	–	–	–	–	–	–	–	–
Vp2*	–	–	–	–	–	–	–	–	–	–	–
Vp3	9	11	3	835	4	283	3	257	2	141	1980
Vp4	10	6	0	619	1	31	3	120	1	30	1090
Vp5*	8	9	1	261	0	0	1	35	1	30	480
Vp6	4	6	0	334	3	132	2	57	0	0	590
Vp7	7	9	0	176	3	104	1	15	2	84	580
Vp8	1	4	0	282	1	52	0	0	0	0	410
Vp9	2	6	0	77	0	0	1	45	1	72	345
Vp10	4	3	0	19	0	0	3	135	0	0	260
Vp11	0	3	0	23	0	0	2	25	0	0	110
Vp12	8	7	0	75	0	0	0	0	0	0	190
Vp13	3	4	0	78	0	0	1	11	0	0	125
Vp14	1	4	1	250	0	0	0	0	0	0	380
Vp15	0	2	2	71	0	0	2	28	1	42	205
Vp16	7	7	0	297	0	0	1	28	0	0	645
Vp17	1	4	0	16	2	17	1	29	1	20	180
Vp18	4	8	0	239	4	522	5	302	4	314	1610
Vp19	3	7	0	–	0	–	0	–	0	–	125
Vp20	5	4	0	466	0	0	1	16	0	0	520

Tabelle C.2: Quantifizierung der Interaktionsprotokolle für Flowcharter (* Versuchsperson mit nur zwei Optionen)

Visio/DS											
Vp	Kpx	K	K'	t_K	X	t_X	R	t_R	O	t_O	t_{ges}
Vp1*	12	12	3	757	0	0	0	0	0	0	1180
Vp2*	3	7	2	518	1	177	0	0	0	0	1010
Vp3	8	8	0	310	2	23	1	24	3	137	360
Vp4	7	7	0	276	0	0	0	0	1	28	680
Vp5*	–	–	–	–	–	–	–	–	–	–	–
Vp6	2	5	0	122	2	137	0	0	0	0	475
Vp7	1	3	0	110	4	228	0	0	0	0	680
Vp8	2	7	0	192	0	0	0	0	0	0	280
Vp9	2	6	0	77	0	0	0	0	1	10	200
Vp10	3	5	1	174	0	0	1	120	0	0	400
Vp11	4	6	0	137	0	0	0	0	0	0	230
Vp12	3	7	0	120	0	0	0	0	0	0	235
Vp13	2	4	0	113	0	0	0	0	0	0	170
Vp14	1	6	0	451	0	0	0	0	0	0	600
Vp15	1	7	3	344	0	0	0	0	1	51	565
Vp16	5	7	0	378	0	0	0	0	0	0	690
Vp17	0	4	0	259	3	37	0	0	1	15	405
Vp18	3	7	0	236	2	167	2	65	3	104	720
Vp19	3	5	0	–	0	–	0	–	0	–	290
Vp20	5	7	0	289	0	0	0	0	0	0	430

Tabelle C.3: Quantifizierung der Interaktionsprotokolle für Visio/DS (* Versuchsperson mit nur zwei Optionen)

D Kennzahlen der Qualitativen Analyse I

Versuchsperson	ϕK_{px}	ϕK	$\phi K'_{rel}$	ϕX_{rel}
Vp1*	12.00	11.50	0.13	0.05
Vp2*	3.00	7.50	0.33	0.13
Vp3	8.00	8.67	0.09	0.20
Vp4	7.33	8.00	0.00	0.09
Vp5*	7.50	7.00	0.06	0.10
Vp6	3.00	6.33	0.04	0.34
Vp7	2.67	5.33	0.00	0.81
Vp8	1.00	5.33	0.00	0.08
Vp9	1.67	5.33	0.00	0.00
Vp10	3.67	4.33	0.07	0.00
Vp11	1.33	4.33	0.00	0.00
Vp12	5.67	7.33	0.00	0.00
Vp13	1.67	3.33	0.00	0.00
Vp14	1.00	5.33	0.25	0.00
Vp15	1.00	5.67	0.60	0.08
Vp16	5.33	6.67	0.00	0.00
Vp17	0.67	4.33	0.07	0.55
Vp18	3.33	7.33	0.05	0.45
Vp19	3.00	5.33	0.00	0.00
Vp20	5.67	6.33	0.00	0.00

Tabelle D.1: Werte der Kennzahlen der Qualitativen Analyse I (* Versuchsperson mit nur zwei Optionen)

Literaturverzeichnis

- Ach, N. (1935). *Analyse des Willens*. Berlin: Urban & Schwarzenberg.
- Achtziger, A. & Gollwitzer, P. M. (2006). Motivation und Volition im Handlungsverlauf. In J. Heckhausen & H. Heckhausen (Hg.), *Motivation und Handeln* (3., überarb. u. akt. Aufl., S. 277-302). Heidelberg: Springer.
- Adams, D. A., Nelson, R. R. & Todd, P. A. (1992). Perceived usefulness, ease of use, and usage of information technology: A replication. *MIS Quarterly*, 16(2), 227-247.
- Adaval, R. (2001). Sometimes it just feels right: The differential weighting of affect-consistent and affect-inconsistent product information. *Journal of Consumer Research*, 28(1), 1-17.
- Adaval, R. (2003). How good gets better and bad gets worse: Understanding the impact of affect on evaluations of known brands. *Journal of Consumer Research*, 30(3), 352-367.
- Ajzen, I. & Fishbein, M. (1980). *Understanding attitudes and predicting social behavior*. Englewood Cliffs: Prentice-Hall.
- Alpar, P., Grob, H. L., Weimann, P. & Winter, R. (2005). *Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen* (4., überarb. Aufl.). Wiesbaden: Vieweg.
- Anderson, E. E. (1989). A heuristic for software evaluation and selection. *Software: Practice and Experience*, 19(8), 707-717.
- Anderson, E. E. (1990). Choice models for the evaluation and selection of software packages. *Journal of Management Information Systems*, 6(4), 123-138.
- Anderson, E. E. & Chen, Y.-M. (1997). Microcomputer software evaluation: An econometric model. *Decision Support Systems*, 19(2), 75-92.
- Anderson, J. R. (2007). *Kognitive Psychologie* (6. Aufl.). Berlin, Heidelberg: Spektrum.
- Aronson, E., Wilson, T. D. & Akert, R. M. (2004). *Sozialpsychologie* (4., akt. Aufl.). München u. a.: Pearson.
- Bagozzi, R. P. (1990). Buyer behavior models for technological products and services: A critique and proposal. In W. Johnston (Hg.), *Advances in telecommunication management* (Bd. 2, S. 43-69). Greenwich: JAI Press.
- Ballurio, K., Scalzo, B. & Rose, L. (2002). Risk reduction in cots software selection with BASIS. In *ICCBSS '02: Proceedings of the First International Conference on COTS-Based Software Systems* (S. 31-43). London: Springer.
- Balzert, H. (1996). *Lehrbuch der Software-Technik: Software-Entwicklung*. Heidelberg u. a.: Spektrum.
- Bandura, A. (1982). Self-efficacy mechanisms in human agency. *American Psychologist*, 37(2), 122-147.

- Baron, J. & Spranca, M. (1997). Protected values. *Organizational Behavior and Human Decision Processes*, 70(1), 1-16.
- Batra, R. & Ahtola, O. T. (1990). Measuring the hedonic and utilitarian sources of consumer attitudes. *Marketing Letters*, 2(2), 159-170.
- Beckworth, G. & Altman, G. (1997). Defining strategies – measuring quality. *Software Quality Journal*, 6(2), 171-178.
- Bernroider, E. & Koch, S. (2000). Entscheidungsfindung bei der Auswahl betriebswirtschaftlicher Standardsoftware – Ergebnisse einer empirischen Untersuchung in österreichischen Unternehmen. *Wirtschaftsinformatik*, 42(4), 329-338.
- Bettman, J. R. (1979). *An information processing theory of choice*. Reading: Addison-Wesley.
- Bettman, J. R., Luce, M. F. & Payne, J. W. (1998). Constructive consumer choice processes. *Journal of Consumer Research*, 25(3), 187-217.
- Bettman, J. R. & Park, C. W. (1980). Effects of prior knowledge and experience and phase of the choice process on consumer decision processes: A protocol analysis. *Journal of Consumer Research*, 7(3), 234-248.
- Bettman, J. R. & Sujan, M. (1987). Effects of framing on evaluation of comparable and noncomparable alternatives by expert and novice consumers. *Journal of Consumer Research*, 14(2), 141-154.
- Bettman, J. R. & Zins, M. A. (1979). Information format and choice task effects in decision making. *Journal of Consumer Research*, 6(2), 141-153.
- Beus-Dukic, L. & Bøegh, J. (2003). COTS software quality evaluation. In *ICCBSS '03: Proceedings of the Second International Conference on COTS-Based Software Systems* (S. 72-80). London: Springer.
- Bevan, N. (1995). Measuring usability as quality of use. *Software Quality Journal*, 4(2), 115-130.
- Biehal, G. & Chakravarti, D. (1982). Information presentation format and learning goals as determinants of consumers' memory retrieval and choice processes. *Journal of Consumer Research*, 8(4), 431-441.
- Biehal, G. & Chakravarti, D. (1983). Information accessibility as a moderator of consumer choice. *Journal of Consumer Research*, 10(1), 1-14.
- Blin, M.-J. & Tsoukiàs, A. (2001). Multi-criteria methodology contribution to the software quality evaluation. *Software Quality Journal*, 9(2), 113-132.
- Boehm, B. W., Brown, J. R., Lipow, M., MacLeod, G. J. & Merritt, M. J. (1978). *Characteristics of software quality*. New York: Elsevier North-Holland.
- Bohner, G. (2003). Einstellungen. In W. Stroebe, K. Jonas & M. Hewstone (Hg.), *Sozialpsychologie: Eine Einführung* (4. Aufl., S. 265-315). Heidelberg: Springer.
- Boloix, G. & Robillard, P. N. (1995). A software system evaluation framework. *Computer*, 28(12), 17-26.
- Bortz, J. & Döring, N. (2006). *Forschungsmethoden und Evaluation* (4., überarb. Aufl.). Heidelberg: Springer.
- Bower, G. H. (1981). Mood and memory. *American Psychologist*, 36(2), 129-148.
- Bower, G. H. (1991). Mood congruity of social judgments. In J. P. Forgas (Hg.), *Emotion and social judgments* (S. 31-53). Oxford: Pergamon Press.

- Brenner, W. (1990). Auswahl von Standardsoftware. In H. Österle (Hg.), *Integrierte Standardsoftware: Entscheidungshilfen für den Einsatz von Softwarepaketen, Band 2: Auswahl, Einführung und Betrieb von Standardsoftware* (S. 9-24). Hallbergmoos: AIT.
- Broniarczyk, S. M. & Alba, J. W. (1994). The role of consumers' intuitions in inference making. *Journal of Consumer Research*, 21(3), 393-407.
- Broniarczyk, S. M., Hoyer, W. D. & McAlister, L. (1998). Consumers' perceptions of the assortment offered in a grocery category: The impact of item reduction. *Journal of Marketing Research*, 35(2), 166-176.
- Brooks, F. P., Jr. (1995). *The mythical man-month: Essays on software engineering. Anniversary edition*. Reading u. a.: Addison-Wesley.
- Brucks, M. (1985). The effects of product class knowledge on information search behavior. *Journal of Consumer Research*, 12(1), 1-16.
- Campbell, M. C. & Goodstein, R. C. (2001). The moderating effect of perceived risk on consumers' evaluations of product incongruity: Preference for the norm. *Journal of Consumer Research*, 28(3), 439-449.
- Carney, D. J. & Long, F. (2000). What do you mean by cots? finally, a useful answer. *IEEE Software*, 17(2), 83-86.
- Carney, D. J. & Wallnau, K. C. (1998). A basis for evaluation of commercial software. *Information and Software Technology*, 40(14), 851-860.
- Carvalho, J. P. & Franch, X. (2006). Extending the ISO/IEC 9126-1 quality model with non-technical factors for COTS components selection. In *WoSQ '06: Proceedings of the 2006 international workshop on Software quality* (S. 9-14). New York: ACM Press.
- Chernev, A. (2003). When more is less and less is more: The role of ideal point availability and assortment in consumer choice. *Journal of Consumer Research*, 30(2), 170-183.
- Chernev, A. (2005). Context effects without a context: Attribute balance as a reason for choice. *Journal of Consumer Research*, 32(2), 213-223.
- Cialdini, R. B. (2002). *Die Psychologie des Überzeugens: Ein Lehrbuch für alle, die ihren Mitmenschen und sich selbst auf die Schliche kommen wollen* (2., überarb. u. erw. Aufl.). Bern u. a.: Huber.
- Clore, G. L. (1992). Cognitive phenomenology: Feelings and the construction of judgment. In L. L. Martin & A. Tesser (Hg.), *The construction of social judgment* (S. 133-164). Hillsdale: Lawrence Erlbaum Associates.
- Clore, G. L. & Parrott, W. G. (1991). Moods and their vicissitudes: Thoughts and feelings as information. In J. P. Forgas (Hg.), *Emotion and social judgments* (S. 107-123). Oxford: Pergamon Press.
- Clore, G. L., Wyer, R. S., Dienes, B., Gasper, K., Gohm, C. L. & Isbell, L. (2001). Affective feelings as feedback: Some cognitive consequences. In L. L. Martin & G. L. Clore (Hg.), *Theories of mood and cognition: A user's handbook* (S. 27-62). Mahwah: Lawrence Erlbaum Associates.
- Comella-Dorda, S., Dean, J. C., Lewis, G., Morris, E. J., Oberndorf, P. A. & Harper, E. (2004). *A process for COTS software product evaluation* (Technical Report

- CMU/SEI-2003-TR-017 ESC-TR-2003-017). Pittsburgh: Carnegie Mellon University, Software Engineering Institute.
- Comella-Dorda, S., Dean, J. C., Morris, E. J. & Oberndorf, P. A. (2002). A process for COTS software product evaluation. In *ICCBSS '02: Proceedings of the First International Conference on COTS-Based Software Systems* (S. 86-96). London: Springer.
- Cramer, P. (1968). *Word association*. New York: Academic Press.
- Dahme, C. & Raeithel, A. (1997). Ein tätigkeitstheoretischer Ansatz zur Entwicklung von brauchbarer Software. *Informatik-Spektrum*, 20(1), 5-12.
- Date, C. J. (2000). *An introduction to database systems* (7. Aufl.). Reading u. a.: Addison Wesley Longman.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319-340.
- Davis, F. D. (1993). User acceptance of information technology: System characteristics, user perceptions and behavioral impacts. *International Journal of Man-Machine Studies*, 38(3), 475-487.
- Davis, F. D., Bagozzi, R. P. & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, 35(8), 982-1003.
- Davison, G. C., Vogel, R. S. & Coffman, S. G. (1997). Think-aloud approaches to cognitive assessment and the articulated thoughts in simulated situations paradigm. *Journal of Consulting and Clinical Psychology*, 65(6), 950-958.
- Dawes, R. M. (1979). The robust beauty of improper linear models in decision making. *American Psychologist*, 34(7), 571-582.
- Deffner, G. (1984). *Lautes Denken – Untersuchung zur Qualität eines Datenerhebungsverfahrens*. Frankfurt a. M.: Peter Lang.
- Deffner, G., Heydemann, M. & Borstel, G. von. (1984). Ein Kategoriensystem und interaktives Ratingverfahren für die Vorverarbeitung von Protokollen des Lauten Denkens. *Archiv für Psychologie*, 136, 147-162.
- Denert, E. (1990). *Software-Engineering*. Berlin u. a.: Springer.
- Denning, P. J. (1992). What is software quality? *Communications of the ACM*, 35(1), 13-15.
- Denning, P. J. & Dunham, R. (2003). The missing customer. *Communications of the ACM*, 46(3), 19-23.
- Dern, G. (2003). *Management von IT-Architekturen: Informationssysteme im Fokus von Architekturplanung und -entwicklung*. Wiesbaden: Vieweg.
- Deutsch, M. S. & Willis, R. R. (1988). *Software quality engineering*. Englewood Cliffs: Prentice-Hall.
- Dhar, R. (1997). Context and task effects on choice deferral. *Marketing Letters*, 8(1), 119-130.
- Dick, A., Chakravarti, D. & Biehal, G. (1990). Memory-based inferences during consumer choice. *Journal of Consumer Research*, 17(1), 82-93.
- Dishaw, M. T. & Strong, D. M. (1999). Extending the technology acceptance model with task-technology fit constructs. *Information & Management*, 36(1), 9-21.

- Drolet, A. & Luce, M. F. (2004). The rationalizing effects of cognitive load on emotion-based trade-off avoidance. *Journal of Consumer Research*, 31(1), 63-77.
- Dromey, R. G. (1995). A model for software product quality. *IEEE Transactions on Software Engineering*, 21(2), 146-162.
- Dromey, R. G. (1996). Cornering the chimera. *IEEE Software*, 13(1), 33-43.
- Dromey, R. G. & McGettrick, A. D. (1992). On specifying software quality. *Software Quality Journal*, 1(1), 45-74.
- Duden. (2003). *Deutsches Universalwörterbuch* (5. Aufl.). Mannheim u. a.: Dudenverlag.
- Eagly, A. H. & Chaiken, S. (1998). Attitude structure and function. In D. Gilbert, S. T. Fiske & G. Lindzey (Hg.), *Handbook of social psychology* (4. Aufl., S. 269-322). New York: McGraw-Hill.
- Elio, R. & Anderson, J. R. (1981). The effects of category generalizations and instance similarity on schema abstraction. *Journal of Experimental Psychology: Human Learning & Memory*, 7(6), 397-417.
- Endres, A. (2003). Softwarequalität aus Nutzersicht und ihre wirtschaftliche Bedeutung. *Informatik-Spektrum*, 26(1), 20-25.
- Ericsson, A. K. & Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, 87(3), 215-251.
- Feldman, J. M. & Lynch, J. G., Jr. (1988). Self-generated validity and other effects of measurement on belief, attitude, intention, and behavior. *Journal of Applied Psychology*, 73(3), 421-435.
- Fiedler, K. (1985). Zur Stimmungsabhängigkeit kognitiver Funktionen. *Psychologische Rundschau*, 36, 125-134.
- Fincham, F. & Hewstone, M. (2003). Attributionstheorie und -forschung – Von den Grundlagen zur Anwendung. In W. Stroebe, K. Jonas & M. Hewstone (Hg.), *Sozialpsychologie: Eine Einführung* (4., überarb. u. erw. Aufl., S. 215-263). Heidelberg: Springer.
- Fishbein, M. & Ajzen, I. (1975). *Belief, attitude, intention and behavior: An introduction to theory and research*. Reading u. a.: Addison-Wesley.
- Fiske, S. T. (1982). Schema-triggered affect: Applications to social perception. In M. S. Clark & S. T. Fiske (Hg.), *Affect and cognition: The seventeenth annual carnegie symposium on cognition* (S. 55-78). Hillsdale: Lawrence Erlbaum Associates.
- Fiske, S. T. & Pavelchak, M. A. (1986). Category-based versus piecemeal-based affective responses: Developments in schema-triggered affect. In R. M. Sorrentino & E. T. Higgins (Hg.), *The handbook of motivation and cognition: Foundations on social behavior* (S. 167-203). New York: Guilford.
- Ford, G. T. & Smith, R. A. (1987). Inferential beliefs in consumer evaluations: An assessment of alternative processing strategies. *Journal of Consumer Research*, 14(3), 363-371.
- Forgas, J. P. & Bower, G. H. (1988). Affect in social and personal judgments. In K. Fiedler & J. P. Forgas (Hg.), *Affect, cognition and social behavior* (S. 183-208). Toronto: Hogrefe.
- Franch, X. (2005). On the lightweight use of goal-oriented models for software package selection. *Lecture Notes in Computer Science*, 3520/2005, 551-566.

- Franch, X. & Carvalho, J. P. (2003). Using quality models in software package selection. *IEEE Software*, 20(1), 34-41.
- Frey, D., Stahl, D. & Gollwitzer, P. M. (1993). Einstellung und Verhalten: Die Theorie des überlegten Handelns und die Theorie des geplanten Verhaltens. In D. Frey & M. Irle (Hg.), *Theorien der Sozialpsychologie, Band I: Kognitive Theorien* (2. Aufl., S. 361-398). Bern: Huber.
- Friedman, N. (1967). *The social nature of psychological research*. New York: Basic.
- Frommann, U. (2002). *Leitfaden „Lautes Denken“* (). Braunschweig: Technische Universität, Arbeitsstelle für Hochschuldidaktik.
- Galletta, D., King, R. C. & Rateb, D. (1993). The effect of expertise on software selection. *ACM SIGMIS Database*, 24(2), 7-20.
- Garbarino, E. C. & Edell, J. A. (1997). Cognitive effort, affect, and choice. *Journal of Consumer Research*, 24(2), 147-158.
- Garvin, D. A. (1984). What does “product quality” really mean? *Sloan Management Review*, 26(1), 25-43.
- Gefen, D. & Straub, D. W. (1997). Gender differences in the perception and use of e-mail: An extension to the technology acceptance model. *MIS Quarterly*, 21(4), 389-400.
- Gentleman, W. M. (1996). *If software quality is a perception, how do we measure it?* (NRC Report No. 40149). Ottawa: National Research Council of Canada.
- Gerrard, B. (1993a). Beyond the logical theory of rational choice. In B. Gerrard (Hg.), *The economics of rationality* (S. 52-67). London, New York: Routledge.
- Gerrard, B. (Hg.). (1993b). *The economics of rationality*. London, New York: Routledge.
- Gigerenzer, G. & Gaissmaier, W. (2006). Denken und Urteilen unter Unsicherheit: Kognitive Heuristiken. In J. Funke (Hg.), *Denken und Problemlösen (Enzyklopädie der Psychologie: Themenbereich C, Theorie und Forschung: Ser. 2, Kognition; Bd. 8* (S. 329-374). Göttingen u. a.: Hogrefe.
- Gilb, T. (1988). *Principles of software engineering management*. Harlow u. a.: Addison-Wesley.
- Goldstein, E. B. (2001). *Wahrnehmungspsychologie: Eine Einführung*. Heidelberg u. a.: Spektrum.
- Grady, R. B. & Caswell, D. L. (1987). *Software metrics: Establishing a company-wide program*. Englewood Cliffs: Prentice-Hall.
- Gregan-Paxton, J. & Roedder John, D. (1997). Consumer learning by analogy: A model of internal knowledge transfer. *Journal of Consumer Research*, 24(3), 266-284.
- Greve, W. & Wentura, D. (1997). *Wissenschaftliche Beobachtung: Eine Einführung* (2. Aufl.). Weinheim: Beltz PVU.
- Guski, R. (1991). *Wahrnehmen – Ein Lehrbuch*. Stuttgart u. a.: Kohlhammer.
- Hackman, J. R. (1969). Toward understanding the role of tasks in behavioral research. *Acta Psychologica*, 31(2), 97-128.
- Hager, W. & Spies, K. (1991). *Versuchsdurchführung und Versuchsbericht: Ein Leitfaden*. Göttingen u. a.: Hogrefe.
- Hahn, H. (1994). *Das große CD-ROM Buch*. Düsseldorf: Data Becker.
- Hartwick, J. & Barki, H. (1994). Hypothesis testing and hypothesis generating research: An example from the user participation literature. *Information Systems Research*,

- 5(4), 446-449.
- Hassenzahl, M. (2003). The thing and I: Understanding the relationships between user and product. In M. Blythe, C. Overbeeke, A. F. Monk & P. C. Wright (Hg.), *Funology: From usability to enjoyment* (S. 31-42). Dordrecht: Kluwer Academic Publishers.
- Hassenzahl, M. (2004). E-Mail-Korrespondenz, 08.06.2004.
- Hassenzahl, M., Beu, A. & Burmester, M. (2001). Engineering joy. *IEEE Software*, 18(1), 70-76.
- Hassenzahl, M., Burmester, M. & Koller, F. (2003). AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In J. Ziegler & G. Szwillus (Hg.), *Mensch & Computer 2003: Interaktion in Bewegung* (S. 187-196). Stuttgart, Leipzig: B. G. Teubner.
- Hassenzahl, M., Platz, A., Burmester, M. & Lehner, K. (2000). Hedonic and ergonomic quality aspects determine a software's appeal. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems* (S. 201-208). New York: ACM Press.
- Heckhausen, H. (1987). Wünschen – Wählen – Wollen. In H. Heckhausen, P. M. Gollwitzer & F. E. Weinert (Hg.), *Jenseits des Rubikon: Der Wille in den Humanwissenschaften* (S. 3-9). Berlin: Springer.
- Heckhausen, J. & Heckhausen, H. (2006). Motivation und Handeln: Einführung und Überblick. In J. Heckhausen & H. Heckhausen (Hg.), *Motivation und Handeln* (3., überarb. u. akt. Aufl., S. 1-9). Heidelberg: Springer.
- Heinrich, L. J. (2002). *Informationsmanagement: Planung, Überwachung und Steuerung der Informationsinfrastruktur* (7. Aufl.). München, Wien: Oldenbourg.
- Heinrich, L. J., Heinzl, A. & Roithmayr, F. (2004). *Wirtschaftsinformatik-Lexikon* (7., überarb. u. erw. Aufl.). München, Wien: Oldenbourg.
- Hellens, L. A. von. (1997). *Information systems quality versus software quality* (Research Report CIT-96-08). Brisbane: Griffith University, School of Computing and Information Technology.
- Herzberg, F., Mausner, B. & Snyderman, B. (1959). *The motivation to work*. New York: Wiley.
- Hogarth, R. M. & Reder, M. W. (Hg.). (1987). *Rational choice: The contrast between economics and psychology*. Chicago, London: University of Chicago Press.
- House, C. H. (2005). Information worker tools selection, adoption and evaluation: Lessons from software development history. In *HICSS '05: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9* (S. 315.2). Washington: IEEE Computer Society.
- Huber, G. L. & Mandl, H. (Hg.). (1994). *Verbale Daten: Eine Einführung in die Grundlagen und Methoden der Erhebung und Auswertung* (2. Aufl.). Weinheim u. a.: Beltz PVU.
- Huber, J. & McCann, J. M. (1982). The impact of inferential beliefs on product evaluations. *Journal of Marketing Research*, 19(3), 324-333.
- Huber, J., Payne, J. W. & Puto, C. (1982). Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of Consumer Research*, 9(1), 90-98.

- Huber, J. & Puto, C. (1983). Market boundaries and product choice: Illustrating attraction and substitution effects. *Journal of Consumer Research*, 10(1), 31-44.
- Huber, O. (2005). *Das psychologische Experiment: Eine Einführung* (2., überarb. Aufl.). Bern: Huber.
- Hussy, W. & Jain, A. (2002). *Experimentelle Hypothesenprüfung in der Psychologie*. Göttingen u. a.: Hogrefe.
- Isen, A. M. (1997). Positive affect and decision making. In W. M. Goldstein & R. M. Hogarth (Hg.), *Research on judgment and decision making: Currents, connections, and controversies* (S. 509-534). Cambridge: Cambridge University Press.
- Isen, A. M., Daubman, K. A. & Nowicki, G. P. (1987). Positive affect facilitates creative problem solving. *Journal of Personality and Social Psychology*, 52(6), 1122-1131.
- Isen, A. M., Johnson, M. M. S., Mertz, E. & Robinson, G. F. (1985). The influence of positive affect on the unusualness of word associations. *Journal of Personality and Social Psychology*, 48(6), 1413-1426.
- ISO/IEC. (2001). *International Standard ISO/IEC 9126-1: Software engineering – Product quality – Part 1: Quality model*. Genf: International Organization for Standardization, International Electrotechnical Commission.
- Iyengar, S. S. & Lepper, M. R. (2000). When choice is demotivating: Can one desire too much of a good thing? *Journal of Personality and Social Psychology*, 79(6), 995-1006.
- Izard, C. E. (1999). *Die Emotionen des Menschen: Eine Einführung in die Grundlagen der Emotionspsychologie*. Weinheim: Beltz PVU.
- Jacoby, J., Chestnut, R. W. & Fisher, W. A. (1978). A behavioral process approach to information acquisition in nondurable purchasing. *Journal of Marketing Research*, 15(4), 532-544.
- Jacoby, J., Chestnut, R. W., Weigl, K. C. & Fisher, W. A. (1976). Pre-purchase information acquisition: Description of a process methodology, research paradigm, and pilot investigation. In B. B. Anderson (Hg.), *Advances in consumer research* (Bd. 3, S. 306-314). Valdosta: Association for Consumer Research.
- Jacoby, J., Jaccard, J. J., Currim, I., Kuss, A., Ansari, A. & Troutman, T. (1994). Tracing the impact of item-by-item information accessing on uncertainty reduction. *Journal of Consumer Research*, 21(2), 291-303.
- Jacoby, J., Jaccard, J. J., Kuss, A., Troutman, T. & Mazursky, D. (1987). New directions in behavioral process research: Implications for social psychology. *Journal of Experimental Social Psychology*, 23(2), 146-175.
- Jacoby, J., Speller, D. E. & Kohn, C. A. (1974a). Brand choice behavior as a function of information load. *Journal of Marketing Research*, 11(1), 63-69.
- Jacoby, J., Speller, D. E. & Kohn, C. A. (1974b). Brand choice behavior as a function of information load: Replication and extension. *Journal of Consumer Research*, 1(1), 33-42.
- Johnson, E. J. & Meyer, R. J. (1984). Compensatory choice models of noncompensatory processes: The effect of varying context. *Journal of Consumer Research*, 12(4), 406-417.

- Johnson, E. J. & Russo, J. E. (1984). Product familiarity and learning new information. *Journal of Consumer Research*, 11(1), 542-550.
- Johnson, M. D. (1984). Consumer choice strategy for comparing noncomparable alternatives. *Journal of Consumer Research*, 11(3), 741-753.
- Johnson, R. D. & Levin, I. P. (1985). More than meets the eye: The effect of missing information on purchase evaluations. *Journal of Consumer Research*, 12(2), 169-177.
- Jungermann, H., Pfister, H.-R. & Fischer, K. (2005). *Die Psychologie der Entscheidung: Eine Einführung* (2. Aufl.). München: Elsevier / Spektrum.
- Karahanna, E. & Straub, D. W. (1999). The psychological origins of perceived usefulness and ease-of-use. *Information & Management*, 35(4), 237-250.
- Keller, K. L. & Staelin, R. (1987). Effects of quality and quantity of information on decision effectiveness. *Journal of Consumer Research*, 14(2), 200-213.
- Kemp, E. E., Phillips, C. H. E., Pringle, D., Hedderley, D., Dickson, B. & Chan, M. L. K. (2002). Software selection for the management and prevention of RSI in a diverse user community. *International Journal of Industrial Ergonomics*, 29(1), 1-14.
- Kiesel, A. (2003). *Handlungsdeterminierende Prozesse beim Aufgabenwechsel und die Notwendigkeit der Dekomposition von Wechselkosten*. Unveröffentlichte Dissertation, Julius-Maximilians-Universität Würzburg.
- Kitchenham, B., Linkman, S., Pasquini, A. & Nanni, V. (1997). The SQUID approach to defining a quality model. *Software Quality Journal*, 6(3), 211-233.
- Kitchenham, B. & Pfleeger, S. H. (1996). Software quality: The elusive target. *IEEE Software*, 13(1), 12-21.
- Kleinbeck, U. (1996). *Arbeitsmotivation: Entstehung, Wirkung und Förderung*. Weinheim, München: Juventa.
- Kleinbeck, U. (2006). Handlungsziele. In J. Heckhausen & H. Heckhausen (Hg.), *Motivation und Handeln* (3., überarb. u. akt. Aufl., S. 255-276). Heidelberg: Springer.
- Kluge, S. (1999). *Empirisch begründete Typenbildung: Zur Konstruktion von Typen und Typologien in der qualitativen Sozialforschung*. Opladen: Leske + Budrich.
- Klute, R. (1996). *Das World Wide Web: Web-Server und -Clients, HTML 2.0/3.0, HTTP*. Bonn u. a.: Addison-Wesley.
- Kontio, J. (1996). A case study in applying a systematic method for COTS selection. In *ICSE-18: Proceedings of the 18th International Conference on Software Engineering* (S. 201-209). Berlin, Heidelberg, New York: Springer.
- Kroeber-Riel, W. & Weinberg, P. (2003). *Konsumentenverhalten* (8. Aufl.). München: Vahlen.
- Kusters, R. J., Solingen, R. van & Trienekens, J. J. M. (1997). User-perceptions of embedded software quality. In *STEP '97: Proceedings of the 8th International Workshop on Software Technology and Engineering Practice (including CASE '97)* (S. 184-197). Washington: IEEE Computer Society.
- Kyas, O. (1994). *Internet: Zugang, Utilities, Nutzung*. Bergheim: Datacom.
- Lazarsfeld, P. F. & Barton, A. H. (1951). Qualitative measurement in the social sciences: Classification, typologies, and indices. In D. Lerner & H. D. Lasswell (Hg.), *The*

- policy sciences: Recent developments in scope and method* (S. 155-192). Stanford: Stanford University Press.
- Le Blanc, L. & Jelassi, T. (1989). DSS software selection: A multiple criteria decision methodology. *Information & Management*, 17(1), 49-65.
- Le Blanc, L. & Jelassi, T. (1994). An empirical assessment of choice models for software selection: A comparison of the LWA and MAUT techniques. *Revue des Systèmes de Décision*, 3(2), 115-126.
- Lehmann, D. R. (1994). Characteristics of 'really' new products. In M. Adams & J. LaCugna (Hg.), *And now for something completely different: Really new products. Marketing science report no. 94-124* (S. 1-2). Cambridge: Marketing Science Institute.
- Leventhal, M. & Tomarken, A. J. (1986). Emotion: Today's problems. *Annual Review of Psychology*, 37, 565-610.
- Lewinson, S. & Mano, H. (1993). Multi-attribute choice and affect: The influence of naturally occurring and manipulated moods on choice processes. *Journal of Behavioral Decision Making*, 6(1), 33-51.
- Locke, E. A. & Latham, G. P. (1990). *A theory of goal setting and task performance*. Englewood Cliffs: Prentice Hall.
- Luce, M. F. (1998). Choosing to avoid: Coping with negatively emotion-laden consumer decisions. *Journal of Consumer Research*, 24(4), 409-433.
- Luce, M. F., Bettman, J. R. & Payne, J. W. (1997). Choice processing in emotionally difficult decisions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(2), 384-405.
- Luce, M. F., Payne, J. W. & Bettman, J. R. (1999). Emotional trade-off difficulty and choice. *Journal of Marketing Research*, 36(2), 143-159.
- Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis*. New York: Wiley.
- Lurie, N. H. (2004). Decision making in information-rich environments: The role of information structure. *Journal of Consumer Research*, 30(4), 473-486.
- Lussier, D. A. & Olshavsky, R. W. (1979). Task complexity and contingent processing in brand choice. *Journal of Consumer Research*, 6(2), 154-165.
- Lynch, J. G., Jr., Chakravarti, D. & Mitra, A. (1991). Contrast effects in consumer judgments: Changes in mental representations or in the anchoring of rating scales? *Journal of Consumer Research*, 18(3), 284-297.
- Lynch, J. G., Jr., Marmorstein, H. & Weigold, M. F. (1988). Choices from sets including remembered brands: Use of recalled attributes and prior overall evaluations. *Journal of Consumer Research*, 15(2), 169-184.
- Lynch, J. G., Jr. & Srull, T. K. (1982). Memory and attentional factors in consumer choice: Concepts and research methods. *Journal of Consumer Research*, 9(1), 18-37.
- Maiden, N. A. & Ncube, C. (1998). Acquiring COTS software selection requirements. *IEEE Software*, 15(2), 46-56.
- Malhotra, N. K. (1982). Information load and consumer decision making. *Journal of Consumer Research*, 8(4), 419-430.
- Mandler, G. (1982). The structure of value: Accounting for taste. In M. S. Clark & S. T. Fiske (Hg.), *Affect and cognition: The 17th annual Carnegie symposium* (S. 3-36).

- Hillsdale: Lawrence Erlbaum Associates.
- Mano, H. (1990). Emotional states and decision making. In M. Goldberg, G. Gorn & R. Pollay (Hg.), *Advances in consumer research* (Bd. 17, S. 577-584). Provo: Association for Consumer Research.
- Mano, H. & Oliver, R. L. (1993). Assessing the dimensionality and structure of the consumption experience: Evaluation, feeling and satisfaction. *Journal of Consumer Research*, 20(3), 451-466.
- March, J. G. (1978). Bounded rationality, ambiguity, and the engineering of choice. *Bell Journal of Economics*, 9(2), 587-608.
- Martin, J. & McClure, C. (1983). Buying software off the rack. *Harvard Business Review*, 61(6), 32-52.
- Mayring, P. (2007). *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (9., überarb. Aufl.). Weinheim: Beltz UTB.
- McCall, J. A. (1994). Quality factors. In J. J. Marciniak (Hg.), *Encyclopedia of software engineering, Volume 2: O – Z* (S. 958-969). New York u. a.: Wiley.
- McCall, J. A., Richards, P. K. & Walters, G. F. (1977). *Factors in software quality, Vols. I-III* (NTIS AD-A049-014/015/055). Springfield: National Technical Information Service.
- McDougall, A., Squires, D. & Guss, S. (1996). Emphasising use over attributes in selection of educational software. *Education and Information Technologies*, 1(2), 151-164.
- Mervis, C. B. & Rosch, E. (1981). Categorization of natural objects. *Annual Review of Psychology*, 32, 89-115.
- Meyer, B. (1990). *Objektorientierte Softwareentwicklung*. München u. a.: Hanser.
- Meyer, R. J. (1981). A model of multiattribute judgments under attribute uncertainty and information constraint. *Journal of Marketing Research*, 18(4), 428-441.
- Meyer, R. J. & Johnson, E. J. (1989). Information overload and the nonrobustness of linear models: A comment on Keller and Staelin. *Journal of Consumer Research*, 15(4), 498-503.
- Meyers-Levy, J. & Tybout, A. M. (1989). Schema congruity as a basis for product evaluation. *Journal of Consumer Research*, 16(1), 39-54.
- Moreau, C. P., Markman, A. B. & Lehmann, D. R. (2001). 'what is it?' categorization flexibility and consumers' responses to really new products. *Journal of Consumer Research*, 27(4), 289-498.
- Morisio, M. & Tsoukiàs, A. (1997). IusWare: A methodology for the evaluation and selection of software products. *IEE Proceedings - Software Engineering*, 144(3), 162-174.
- Mukherjee, A. & Hoyer, W. D. (2001). The effect of novel attributes on product evaluation. *Journal of Consumer Research*, 28(3), 462-472.
- Musa, J. D. (1987). Software quality and reliability basics. In *ACM '87: Proceedings of the 1987 Fall Joint Computer Conference on Exploring Technology: Today and Tomorrow* (S. 114-115). Los Alamitos: IEEE Computer Society Press.
- Musson, T. & Dodman, E. (1995). An investigation of quality profiles for different types of software. In M. Ross, C. A. Brebbia, G. Staples & J. Stapleton (Hg.), *Software*

- quality management III, Vol. 2: Measuring and maintaining quality* (S. 119-126). Southampton, Boston: Computational Mechanics Publications.
- Muthukrishnan, A. V. & Kardes, F. R. (2001). Persistent preferences for product attributes: The effects of the initial choice context and uninformative experience. *Journal of Consumer Research*, 28(1), 89-104.
- Nedungadi, P. (1990). Recall and consumer consideration sets: Influencing choice without altering brand evaluations. *Journal of Consumer Research*, 17(3), 263-276.
- Nerdinger, F. W. (1995). *Motivation und Handeln in Organisationen: Eine Einführung*. Stuttgart: Kohlhammer.
- Nerdinger, F. W. (2001). Motivierung. In H. Schuler (Hg.), *Lehrbuch der Personalpsychologie* (S. 349-371). Göttingen u. a.: Hogrefe.
- Nesbit, I. S. (1984). Evaluating micro software. *Datamation*, 30(15), 74-78.
- Newell, A. & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs: Prentice Hall.
- OECD. (2007). *OECD science, technology and industry scoreboard 2007: Innovation and performance in the global economy*. Paris: OECD Publishing.
- Oerter, R. (1968). *Moderne Entwicklungspsychologie*. Donauwörth: Ludwig Auer.
- Oh, K. S., Lee, N. Y. & Rhew, S. Y. (2003). A selection process of COTS components based on the quality of software in a special attention to internet. *Lecture Notes in Computer Science*, 2713/2003, 626-631.
- Olshavsky, R. W. (1979). Task complexity and contingent processing in decision making: A replication and extension. *Organizational Behavior and Human Performance*, 24(3), 300-316.
- Oppewal, H. & Koelemeijer, K. (2005). More choice is better: Effects of assortment size and composition on assortment evaluation. *International Journal of Research in Marketing*, 22(1), 45-60.
- Ortega, M., Pérez, M. & Rojas, T. (2003). Construction of a systemic quality model for evaluating a software product. *Software Quality Journal*, 11(3), 219-242.
- Otnes, C., Lowry, T. M. & Shrum, L. J. (1997). Towards an understanding of consumer ambivalence. *Journal of Consumer Research*, 24(1), 80-93.
- Parducci, A. (1968). The relativism of absolute judgment. *Scientific American*, 219(6), 84-90.
- Parducci, A. (1974). Contextual effects: A range-frequency analysis. In E. C. Carterette & M. P. Friedman (Hg.), *Handbook of perception, Vol. 2* (S. 127-141). New York: Academic Press.
- Payne, J. W. (1976). Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organizational Behavior and Human Performance*, 16(2), 366-387.
- Payne, J. W. (1982). Contingent decision behavior. *Psychological Bulletin*, 92(2), 382-402.
- Payne, J. W., Bettman, J. R. & Johnson, E. J. (1993). *The adaptive decision maker*. Cambridge: Cambridge University Press.
- Payne, J. W., Bettman, J. R. & Luce, M. F. (1996). When time is money: Decision behavior under opportunity-cost time pressure. *Organizational Behavior and Human*

- Decision Processes*, 66(2), 131-152.
- Payne, J. W., Braunstein, M. L. & Carroll, J. S. (1978). Exploring predecisional behavior: An alternative approach to decision research. *Organizational Behavior and Human Performance*, 22(1), 17-44.
- Peracchio, L. A. & Tybout, A. M. (1996). The moderating role of prior knowledge in schema-based product evaluation. *Journal of Consumer Research*, 23(3), 177-192.
- Pham, M. T. (1998). Representativeness, relevance, and the use of feelings in decision making. *Journal of Consumer Research*, 25(2), 144-159.
- Pham, M. T., Cohen, J. B., Pracejus, J. W. & Hughes, G. D. (2001). Affect monitoring and the primacy of feelings in judgment. *Journal of Consumer Research*, 28(2), 167-188.
- Piaget, J. (1974). *Die Entwicklung der Wirklichkeit beim Kinde*. Stuttgart: Klett.
- Piaget, J. (2003). *Meine Theorie der geistigen Entwicklung*. Weinheim u. a.: Beltz.
- Rangarajan, K., Swaminathan, N., Hedge, V. & Jacob, J. (2001). Product quality framework: A vehicle for focusing on product quality goals. *Software Engineering Notes*, 26(4), 77-82.
- Röck, H. (2001). *Grundlagen der Wirtschaftsinformatik (Studienjahr 2001/2002)*. Vorlesungsskript. Universität Rostock, Wirtschafts- und Sozialwissenschaftliche Fakultät, Lehrstuhl für Wirtschaftsinformatik.
- Röck, H. (2007). *Grundlagen der Wirtschaftsinformatik (Studienjahr 2007/2008)*. Vorlesungsskript. Universität Rostock, Fakultät für Informatik und Elektrotechnik, Lehrstuhl für Wirtschaftsinformatik.
- Reinicke, B. A. & Marakas, G. M. (2005). Exploring the psychological determinants of perceived ease of use and usefulness. In *HICSS '05: Proceedings of the 38th annual Hawaii International Conference on System Sciences (HICSS'05) - Volume 6* (S. 149.2). Washington: IEEE Computer Society.
- Roehm, M. L. & Sternthal, B. (2001). The moderating effect of knowledge and resources on the persuasive impact of analogies. *Journal of Consumer Research*, 28(2), 257-272.
- Rosenstiel, L. von. (2007). *Grundlagen der Organisationspsychologie* (6. Aufl.). Stuttgart: Schäffer Poeschel.
- Ross, W. T., Jr. & Creyer, E. H. (1992). Making inferences about missing information: The effects of existing information. *Journal of Consumer Research*, 19(1), 14-25.
- Royce, W. (1990). Pragmatic quality metrics for evolutionary software development models. In *TRI-Ada '90: Proceedings of the conference on TRI-ADA '90* (S. 551-565). New York: ACM Press.
- Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6), 1161-1178.
- Russell, J. A. & Pratt, G. (1980). A description of the affective quality attributed to environments. *Journal of Personality and Social Psychology*, 38(2), 311-322.
- Russo, J. E. (1974). More information is better: A re-evaluation of Jacoby, Speller, and Kohn. *Journal of Consumer Research*, 1(3), 68-72.
- Russo, J. E. & Doshier, B. A. (1983). Strategies for multiattribute binary choice. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 9(4), 676-696.

- Schmidt-Atzert, L. (1996). *Lehrbuch der Emotionspsychologie*. Stuttgart u. a.: Kohlhammer.
- Schwarz, N. & Clore, G. L. (1983). Mood, misattribution, and judgments of well-being: Informative and directive functions of affective states. *Journal of Personality and Social Psychology*, 45(3), 513-523.
- Schwarz, N. & Clore, G. L. (1988). How do i feel about it? The informative function of affective states. In K. Fiedler & J. P. Forgas (Hg.), *Affect, cognition, and social behavior* (S. 44-62). Toronto: Hogrefe.
- Schwarz, N. & Clore, G. L. (2003). Mood as information: 20 years later. *Psychological Inquiry*, 14(3&4), 296-303.
- Schwarz, N. & Vaughn, L. A. (2000). The availability heuristic revisited: Ease of recall and content of recall as distinct sources of information. In T. Gilovich, D. Griffin & D. Kahneman (Hg.), *Heuristics and biases: The psychology of intuitive judgment* (S. 103-119). Cambridge: Cambridge University Press.
- Segars, A. H. & Grover, V. (1993). Re-examining perceived ease of use and usefulness: A confirmatory factor analysis. *MIS Quarterly*, 17(4), 517-525.
- Selten, R. (2001). What is bounded rationality? In G. Gigerenzer & R. Selten (Hg.), *Bounded rationality: The adaptive toolbox* (S. 13-36). Cambridge, London: MIT Press.
- Shafir, E. B., Osherson, D. N. & Smith, E. E. (1989). An advantage model of choice. *Journal of Behavioral Decision Making*, 2(1), 1-23.
- Shafir, E. B., Osherson, D. N. & Smith, E. E. (1993). The advantage model: A comparative theory of evaluation and choice under risk. *Organizational Behavior and Human Decision Processes*, 55(3), 325-378.
- Simmons, C. J. & Lynch, J. G., Jr. (1991). Inference effects without inference making? effects of missing information on discounting and use of presented information. *Journal of Consumer Research*, 17(4), 477-491.
- Simon, H. A. (1955). A behavioral model of rational choice. *Quarterly Journal of Economics*, 69(1), 99-118.
- Simon, H. A. (1990). Invariants of human behaviour. *Annual Review of Psychology*, 41, 1-19.
- Simonson, I. (1989). Choice based on reasons: The case of attraction and compromise effects. *Journal of Consumer Research*, 16(2), 158-174.
- Simonson, I. & Tversky, A. (1992). Choice in context: Tradeoff contrast and extremeness aversion. *Journal of Marketing Research*, 29(3), 281-295.
- Someren, M. W. van, Barnard, Y. F. & Sandberg, J. A. C. (1994). *The think aloud method: A practical guide to modelling cognitive processes*. London: Academic Press.
- Sommerville, I. (2007). *Software Engineering* (8. Aufl.). München: Pearson Studium.
- Sorgatz, H. (2002). Repetitive strain injuries: Unterarm-/Handbeschwerden aufgrund repetitiver Belastungsreaktionen des Gewebes. *Der Orthopäde*, 31(10), 1006-1014.
- Stahlknecht, P. & Hasenkamp, U. (2005). *Einführung in die Wirtschaftsinformatik* (11., überarb. Aufl.). Berlin u. a.: Springer.
- Stamelos, I., Vlahavasa, I., Refanidisa, I. & Tsoukiàs, A. (2000). Knowledge based evaluation of software systems: a case study. *Information and Software Technology*,

- 42(5), 333-345.
- Steinmann, C. (2007). E-Mail-Korrespondenz, 21.10.2007.
- Stickel, E. (Hg.). (1997). *Gabler Wirtschaftsinformatik-Lexikon*. Wiesbaden: Gabler.
- Sujan, M. (1985). Consumer knowledge: Effects on evaluation strategies mediating consumer judgments. *Journal of Consumer Research*, 12(1), 31-46.
- Summers, J. O. (1974). Less information is better? *Journal of Marketing Research*, 11(4), 467-468.
- Szajna, B. (1994). Software evaluation and choice: Predictive validation of the technology acceptance instrument. *MIS Quarterly*, 18(3), 319-324.
- Szajna, B. (1996). Empirical evaluation of the revised technology acceptance model. *Management Science*, 42(1), 85-92.
- Teasdale, J. D. & Fogarty, S. J. (1979). Differential effects of induced mood on retrieval of pleasant and unpleasant events from episodic memory. *Journal of Abnormal Psychology*, 88(3), 248-257.
- Timmreck, E. M. (1973). Computer selection methodology. *Computing Surveys*, 5(4), 199-222.
- Tubbs, M. E. (1986). Goal-setting: A meta-analytic examination of the empirical evidence. *Journal of Applied Psychology*, 71(3), 474-483.
- Tversky, A. (1972). Elimination by aspects: A theory of choice. *Psychological Review*, 79(4), 281-299.
- Tversky, A. & Kahneman, D. (1991). Loss aversion in riskless choice: A reference dependent model. *Quarterly Journal of Economics*, 106(4), 1039-1062.
- Tversky, A. & Simonson, I. (1993). Context-dependent preferences. *Management Science*, 39(10), 1179-1189.
- Tybout, A., Sternthal, B., Malaviya, P., Bakamitsos, G. A. & Park, S. (2005). Information accessibility as a moderator of judgments: The role of content versus retrieval ease. *Journal of Consumer Research*, 32(1), 76-85.
- Venkatesh, V. (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. *Information Systems Research*, 11(4), 342-365.
- Venkatesh, V. & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, 46(2), 186-204.
- Venkatesh, V., Morris, M. G., Davis, G. B. & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3), 425-478.
- Vidgen, R., Wood-Harper, A. T. & Wood, R. (1993). A soft systems approach to information systems quality. *Scandinavian Journal of Information Systems*, 5, 97-112.
- Vroom, V. H. (1964). *Work and motivation*. New York: Wiley.
- Wagner, R. W. & A. C. (1994). Die Methode des Lauten Denkens. In G. L. Huber & H. Mandl (Hg.), *Verbale Daten: Eine Einführung in die Grundlagen und Methoden der Erhebung und Auswertung* (2. Aufl., S. 81-103). Weinheim u. a.: Beltz PVU.
- Watson, D. & Tellegen, A. (1985). Toward a consensual structure of mood. *Psychological Bulletin*, 98(2), 219-235.
- Wenninger, G. (Hg.). (2001). *Lexikon der Psychologie, Band 2: F bis L*. Heidelberg: Spektrum.

- Wilkie, W. L. (1974). Analysis of effects of information load. *Journal of Marketing Research*, 11(4), 462-466.
- Wood, S. L. & Lynch, J. G., Jr. (2002). Prior knowledge and complacency in new product learning. *Journal of Consumer Research*, 29(3), 416-426.
- Woodman, I. H. G. (1994). *Relationship between the activities of the software process and the quality of the software product* (Research Report/94/171). Glasgow: University Of Strathclyde, Department of Computer Science, Software Quality Research Group.
- Wottawa, H. & Thierau, H. (1998). *Lehrbuch Evaluation* (2. Aufl.). Bern: Huber.
- Wright, P. L. & Barbour, F. (1977). Phased decision strategies: Sequels to an initial screening. In M. K. Starr & M. Zeleny (Hg.), *North Holland/TIMS studies in the management sciences, Vol. 6: Multiple criteria decision making* (S. 91-109). Amsterdam: North Holland.
- Yamauchi, T. & Markman, A. B. (2000). Inference using categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(3), 776-795.
- Yeung, C. W. M. & Soman, D. (2005). Attribute evaluability and the range effect. *Journal of Consumer Research*, 32(3), 363-369.
- Yeung, C. W. M. & Wyer, R. S., Jr. (2004). Affect, appraisal, and consumer judgment. *Journal of Consumer Research*, 31(2), 412-424.