GESIS

InformationsZentrum
Sozialwissenschaften

IZ-Arbeitsbericht Nr. 29

Methoden zur Evaluation von Software

Marcus Hegner

Mai 2003

InformationsZentrum
Sozialwissenschaften

Lennéstraße 30
D-53113 Bonn
Tel.: 0228/2281-0
Fax.: 0228/2281-120
email: iz@bonn.iz-soz.de
Internet: http://www.gesis.org

ISSN: 1431-6943

Herausgeber: Informationszentrum Sozialwissenschaften der Arbeitsgemeinschaft Sozialwissenschaftlicher Institute e.V. (ASI)

Druck u. Vertrieb: Informationszentrum Sozialwissenschaften, Bonn
Printed in Germany

Inhalt

1 Einleitung

Die Software-Ergonomie ist aus der Einsicht entstanden, dass aus den vielfältigen Gestaltungsmöglichkeiten, die moderne Technologien bereithalten, stets Systeme entwickelt werden müssen, die menschlichen Bedürfnissen und Anforderungen entsprechen. Neben Fragen der technischen Leistungsfähigkeit und Perfektion stellt die Anpassung von Computersystemen an den sie benutzenden Menschen eine entscheidende Gestaltungsaufgabe für alle an einer Entwicklung beteiligten Parteien dar. Die Umsetzung des Gestaltungsziels „Benutzerfreundlichkeit“ bzw. „Usability“ von Softwaresystemen wird einerseits durch ein Angebot an Richtlinienkatalogen sowie nationalen und internationalen Standards unterstützt, andererseits zeigt die Praxis, dass die Entwicklung einer hochwertigen Benutzungsoberfläche einen so komplexen Prozess darstellt, dass Evaluationsmaßnahmen in verschiedenen Entwicklungsstadien unverzichtbar sind. Die ergonomische Bewertung von Software kann dabei mit vielen Methoden erfolgen. Das Testen der Usability eines Produktes ist eine zentrale Methode im Rahmen software-ergonomischer Qualitätssicherung bzw. benutzerorientierter Produktgestaltung. Hierbei werden repräsentative Benutzer mit dem zu testenden Softwaresystem konfrontiert und gebeten, realistische Testaufgaben zu lösen. Sind beispielsweise noch keine Erfahrungswerte über die Benutzerzielgruppe eines neuartigen Softwareprodukts vorhanden, können hier Usability-Tests wertvolle Hinweise über Akzeptanz und Anwendungsmöglichkeiten aus Sicht der Benutzer liefern (s. Nielsen 1993:165). Die Evaluation wird heute hauptsächlich als Mittel der Informationssammlung mit gestaltungsunterstützender Rolle innerhalb eines iterativen Software-Entwicklungsprozesses gesehen. Insbesondere der Evaluation von Prototypen kommt hierbei eine bedeutende Rolle zu, wobei durch Evaluationsmethoden Fehler und Schwachstellen des zu fertigenden Systems in einer frühen Entwicklungsphase aufgedeckt werden sollen (s. Nielsen 1993). Dabei wird das Ziel verfolgt, die Software an die Bedürfnisse der Nutzer anzupassen. Für die software-ergonomische Evaluation steht heute eine große Bandbreite von Methoden und Werkzeugen zur Auswahl. Sie reichen von einfachen Checklisten über Inspektionsverfahren bis hin zu aufwendigen Usability-Tests. Im Mittelpunkt der vorliegenden Arbeit steht die Auseinandersetzung mit software-ergonomischen Evaluationsmethoden. Besondere Erwähnung finden in diesem Rahmen Inspektionsmethoden und Usability-Tests.

2 Evaluation

Wottawa (2001) definiert Evaluation „als das Sammeln und Kombinieren von Daten mit einem gewichteten Satz von Skalen mit denen entweder vergleichende oder numerische Beurteilungen erlangt werden sollen“. Auch für Görner und Iig (1993) bedeutet die Evaluation ein systematisches Sammeln, Auswerten und Interpretieren von Daten, um eine reliable und valide Bewertung der Benutzungsschnittstelle zu ermöglichen. Dabei wird aus den Ergebnissen der Evaluation abgeleitet, ob ein vorab definiertes Designziel erreicht ist bzw. ob und wo weitere Verbesserungsmöglichkeiten ausgeschöpft werden können. Für die Deutsche Gesellschaft für Evaluation (DeGEval 2002) hingegen ist Evaluation die systematische Untersuchung des Nutzens oder Wertes eines Gegenstandes. Derartige Evaluationsgegenstände können beispielsweise Programme, Projekte, Produkte, Maßnahmen, Leistungen, Organisationen, Politik, Technologien oder Forschung sein. Die erzielten Ergebnisse, Schlussfolgerungen oder Empfehlungen müssen nachvollziehbar auf empirisch gewonnenen qualitativen und/oder quantitativen Daten sein. Die Evaluation von Software dient dazu, die Usability einer Benutzungsschnittstelle zu testen und zu verbessern. Bei der Beurteilung von Software kann zwischen zwei Arten von Evaluationen differenziert werden:

Formative Evaluation
Die formative Evaluation ist ein wichtiger Bestandteil iterativer Softwareentwicklung. Sie bewertet Software im Laufe ihres Entwicklungsprozesses. Holz auf der Heide (1993) sieht im Entwicklungsprozess den wichtigsten Ansatz um die software-ergonomische Qualität von Dialogsystemen zu verbessern. Die frühzeitige Benutzerbeteiligung und das Prototyping sind wesentliche Maßnahmen um mit Hilfe des Zyklus aus (Re-) Design und Evaluation befriedigende Ergebnisse zu erzielen. Das Ziel der formativen Evaluation ist eine Optimierung der Nutzungsqualität vor Abschluss der Entwicklungsarbeiten. Erhoben werden hier erstens quantitative Daten, um die Realisierung von Benutzbarkeitszielen im SWE-Prozess zu überprüfen und zweitens qualitative Daten über Schwächen eine Produktprototyps, aus denen Maßnahmen zur Verbesserung der Benutzbarkeit abgeleitet werden sollten. Überwiegend bedient sich die formative Evaluation qualitativer Methoden. Zu den am häufigsten eingesetzten Vorgehensweisen im Bereich der formativen Evaluation zählen die Usability-Tests, eine typische Methode ist hierbei die Thinking Aloud Methode (s. Nielsen 1993:170). Diese haben sich zwar als wirkungsvolle Methodik etabliert, ihre geringe Durchführungsökonomie wird jedoch bemängelt (s. Karat 1997).

Summative Evaluation
Ihr Hauptaspekt liegt in der Analyse und Bewertung des Softwareentwicklungsprozesses hinsichtlich der vorher formulierten Evaluationskriterien und der Überprüfung ihrer Einhaltung. Diese führt zu einer abschließenden Bewertung um beispielsweise die Qualität mehrerer Software-Applikationen miteinander zu vergleichen. Die Erkenntnisse können in Nachfolgeprodukte oder späteren Versionen Verwendung finden. „Die Summative Evaluation überprüft die Hypothese, ob die Maßnahme wirksam ist, bzw. genauso wirkt, wie man es theoretisch erwartet hat“ (s. Bortz & Döring 2002:116). Dabei bedient sie sich überwiegend quantitativer Verfahren (z.B. Fragebögen).

2.1 Der Evaluationsprozess

Die meisten Evaluationen weisen bestimmte Gemeinsamkeiten auf:

- Ausgangspunkt einer Evaluation ist der Untersuchungsgegenstand, das Objekt, das untersucht werden soll. Das Objekt muss nicht unbedingt gegenständlich sein: In der Pädagogik wird Evaluation als „Bewerten von Handlungsalternativen“ verstanden (s. Wottawa 2001).
- Das Objekt soll bestimmte Eigenschaften (Attribute) aufweisen. Die zukünftigen Eigenschaften müssen ermittelt und formuliert werden.
- In einem Prozess werden tatsächliche mit den vorab formulierten Eigenschaften des Objektes verglichen.

2.2 Die Ablauflogik von Evaluationen

Baumgartner (1999) schlägt zum Ablauf einer Evaluation einen vierstufigen Prozess vor (s. Scriven 1980):

Formulierung von Wertekriterien
Kriterien werden ausgewählt und definiert, die das Produkt erfüllen muss, um als gut, wertvoll etc. gelten zu können.

Formulierung von Leistungsstandards
Für die oben eingeführten Kriterien wird eine Norm definiert, die das Produkt erreichen muss, damit das Kriterium als erfüllt angesehen werden kann (Operationalisierung).

Messung und Vergleich (Analyse)
Das Produkt wird jetzt unter der Anwendung der Kriterien untersucht, gemessen und mit den vorgegebenen Leistungsstandards verglichen.

Werturteil (Synthese)
In dieser letzten Phase werden die verschiedenen Ergebnisse zu einem einheitlichen Werturteil verknüpft.

Im Prinzip geht es bei Evaluationen um die Erstellung und Zuweisung eines Werturteils (Produkt = gut/schlecht, wertvoll/wertlos). Dabei ist das Ziel immer eine Bewertung, unabhängig davon, ob dies explizit oder implizit benannt wird.

2.3 Evaluationsziele

Jede Form der Evaluation bemüht sich, ein gewähltes Ziel zu verfolgen (s. Wottawa & Thierau 1998). Nach Holz auf der Heide (1993) werden die Ziele, die durch Evaluationen verfolgt werden, folgendermaßen klassifiziert:

Vergleichende Evaluation (Which is better?):
Dazu werden mindestens zwei unterschiedliche Systeme miteinander verglichen. Es dient dazu, das für den Anwendungsfall bessere Produkt oder aus einer Reihe von Prototypen den Besseren zu bestimmen. Dabei können die subjektive Zufriedenheit der Benutzer oder die objektive Leistungsmessung bei der Durchführung einer Arbeitsaufgabe als Kriterien herangezogen werden. Bei der Bearbeitung werden Fehler und Zeiten der Benutzer protokolliert und anschließend ausgewertet (s. Nielsen 1993:78 f.).

Bewertende Evaluation (How good?):
Hier wird eine bestimmte gewünschte oder geforderte Systemeigenschaft geprüft.

Analysierende Evaluation (Why bad?):
Ziel dieser Evaluation ist es, Hinweise auf Schwachstellen zu erhalten, um direkte Gestaltungsvorschläge zu liefern. Dazu müssen potentielle Benutzer die Möglichkeit haben, das System in einem potentiellen Einsatzfall anzuwenden. Voraussetzung hierfür ist, dass sich die Benutzer erst einmal in das System einarbeitet und anschließend eine Arbeitsaufgabe durchführen. Die Ergebnisse können durch Interviews oder Videoaufzeichnung festgehalten werden.

Die ersten beiden Ziele sind der summativen Evaluation zuzuordnen, das dritte Ziel kann hingegen zur Gruppe der formativen Evaluation gezählt werden. Natürlich können Evaluationen auch für bloße „go/stopp“- Entscheidungen sinnvoll durchgeführt werden: Soll z.B. eine bestimmte Maßnahme fortgeführt oder abgebrochen werden?

2.4 Kriterien für die Wahl einer spezifischen Evaluationsmethode

Für die Wahl einer Evaluationsmethode müssen folgende Fragen beantwortet sein (s. Karat 1997):

Was ist das Ziel der Evaluation?
Geht es um einen abschließenden Test des Gesamtsystems, oder sollen bestimmte Aspekte überprüft werden?

Wer soll die Evaluation durchführen?
Ist es besser, dass ein Experte das System inspiziert, oder sollten auch Benutzer die Evaluation durchführen?

Wo soll die Evaluation durchgeführt werden?
Soll sie in einem Labor oder einem Arbeitsplatz durchgeführt werden?

Welche Informationen werden gesammelt?
Was soll gemessen werden, welche Daten sind von Bedeutung?

Wie groß sind die Ressourcen?
Wie viel Geld und Zeit stehen für die Evaluation zur Verfügung?

2.5 Elemente und Kriterien der Evaluation

Im Bereich der Software-Ergonomie wird nach Oppermann und Reiterer (1994) davon ausgegangen, dass bei der Gestaltung und Bewertung von EDV-Systemen die in Abbildung 1 gezeigten Elemente sowie deren Beziehungen zueinander zu berücksichtigen sind.

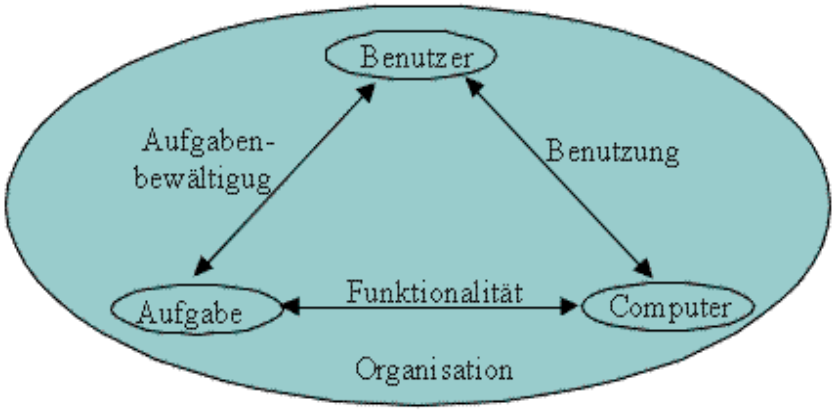

Abbildung 1: Elemente und Beziehungen (Oppermann & Reiterer 1994:337)

Die Maßnahmen des Software-Engineering und der Evaluation, die all diese Elemente und Beziehungen berücksichtigen, werden auch als ganzheitlich bezeichnet. Auf die Ziele einer ganzheitlichen Evaluation wird nachfolgend näher eingegangen (s. Oppermann & Reiterer 1994).

Aufgabenbewältigung
Laut Oppermann und Reiterer (1994) wird die Gestaltung der Beziehung zwischen Benutzer und Aufgabe dadurch bestimmt, inwieweit der Benutzer in der Lage ist, die ihm anvertrauten Aufgaben zu erfüllen und ob er diese als „menschengerecht“ empfindet. Eine menschengerechte Gestaltung der Aufgaben wird als eine essentielle Voraussetzung für eine ganzheitliche Bewertung der ergonomischen Qualität von EDV-Systemen angesehen. In der Arbeitswissenschaft wurden eine Reihe von Kriterien definiert, die die ergonomische Qualität der Aufgabenbewältigung zu erfassen versuchen. Bei der software-ergonomischen Bewertung des EDV-Systems interessiert vor allem, inwieweit die Aufgabenbewältigung durch das EDV-System unterstützt oder behindert wird, ob also die Primäraufgabe „Aufgabenbewältigung“ durch die Sekundäraufgabe „Benutzung des EDV-Systems“ überlagert bzw. in den Hintergrund gedrängt wird. Durch die Berücksichtigung der ergonomischen Qualität der Aufgabenbewältigung soll sichergestellt werden, dass bei der software-ergonomischen Bewertung keine Verengung des Blicks nur auf Schnittstelleneigenschaften des EDV-Systems erfolgt („Schnittstellenkosmetik“).

Funktionalität
Unter Funktionalität eines EDV-Systems wird verstanden, ob das EDV-System aufgabenrelevant und aufgabenangemessen ist. So wird das Ausmaß der Unterstützung der Aufgaben durch das EDV-System bestimmt. Diese ist davon abhängig, ob das EDV-System bestehende Arbeitsaufgaben tatsächlich hinreichend genau abbilden kann oder ob es diese entstellt und verkompliziert. Weiter wird die Funktionalität davon bestimmt, inwieweit der Benutzer das EDV-System, im Hinblick auf bestimmte Aspekte seiner Arbeitsaufgabe, umgestalten kann (z.B. Erweitern durch Hinzufügen neuer Funktionalität). Bei der software-ergonomischen Evaluation ist darauf Rücksicht zu nehmen, inwieweit die vorhandene bzw. nicht vorhandene Funktionalität die Qualität der Benutzung beeinflusst (s. Oppermann & Reiterer 1994).

Benutzung
Anhand der Interaktion zwischen Benutzer und EDV-System wird bestimmt, mit welchem Interaktionsaufwand die Bedienung des EDV-Systems für den Benutzer verbunden ist. Der Interaktionsaufwand wird durch eine Reihe von Kriterien zu bestimmen versucht, z.B. welchen Aufwand muss der Benutzer zum Erlernen betreiben oder welche Möglichkeiten der individuellen Anpas-

sung des EDV-Systems an seinen Arbeitsstil und seine Persönlichkeit werden ihm geboten. Die ergonomische Qualität der Benutzung ist zentraler Bewertungsgegenstand der software-ergonomischen Evaluation (s. Oppermann & Reiterer 1994).

Im Folgenden ist die eigentliche Software zu gestalten beziehungsweise zu bewerten. Aus ergonomischer Sicht wird der Benutzungsschnittstelle eine besondere Rolle zugeschrieben, da sie das Fenster des Benutzers zur eigentlichen Anwendung darstellt (s. Oppermann & Reiterer 1994:340). In der ISO 9241 (Teil 10[1]), die sich speziell auf die Gestaltung und Bewertung von Dialogsystemen bezieht, finden sich einige Grundsätze, deren Umsetzung eine benutzerfreundliche Mensch-Maschine-Interaktion ermöglichen soll.

Zu den sieben Grundsätzen der ISO 9241 Teil 10[2] wird nachfolgend ein Überblick gegeben (s. EN ISO 1995; Oppermann & Reiterer 1994).

Aufgabenangemessenheit (suitability for the task)
Software muss den Benutzer so unterstützen, dass er seine Aufgaben effizient und effektiv erledigen kann, ohne ihn unnötig zu beanspruchen. Das besagt, dass der Benutzer sich auf seine Aufgabenlösung konzentrieren können muss. Die Software darf ihn bei seiner Arbeit nicht belasten. Da die Anzahl der Sinneinheiten, die das Kurzzeitgedächtnis auf einmal wahrnehmen kann, auf 7 ± 2 begrenzt ist, kann die Wahrnehmung durch eine Gliederung der angebotenen Informationen in sinnvolle Gruppen erhöht werden. Die Software sollte zudem so gestaltet sein, dass sie die realen Arbeitsabläufe eines Benutzers nachempfindet. Sie muss sich entsprechend den Kenntnissen und Gewohnheiten des Benutzers verhalten. Das heißt z.B., dass die Positionsmarke automatisch da positioniert wird, wo es vom Arbeitsablauf sinnvoll erscheint.

Selbstbeschreibungsfähigkeit (self-descriptiveness)
Ein Dialog ist dann selbstbeschreibungsfähig, wenn jeder einzelne Schritt dem Benutzer verständlich ist, oder dem Benutzer erklärt wird, wenn er die entsprechende Information verlangt. Selbstbeschreibend bedeutet zum Beispiel, dass ein situationsabhängiges Hilfesystem mit Bezug zur Arbeitsaufgabe existiert. Systemmeldungen (Anweisungen, Fehlermeldungen etc.) müssen präzise, einfach und unmissverständlich sein. Der Benutzer muss den Funkti-

---

[1] Siehe URL: http://www.sozialnetz-hessen.de/ergo-online/Software/S_Ergo-grundsaetze.htm?csok=1

[2] Siehe URL: http://www1.informatik.uni-jena.de/Lehre/SoftErg/vor_r100.htm

onsumfang einer Anwendung schnell und einfach erfassen können. Der Benutzer sollte aus visuell angebotenen Eingabemöglichkeiten auswählen können, und nicht aus dem Gedächtnis eingeben müssen. Nur wenn der Benutzer die sogenannten 5 W beantworten kann - woher komme ich, was ist bis jetzt gemacht worden, wo bin ich, was kann ich als nächstes tun und was kann das System - wird die Software dem Grundsatz der Selbstbeschreibungsfähigkeit gerecht.

Erwartungskonformität (conformity with user expectations)
Software verhält sich dann erwartungskonform, wenn der Dialogablauf den Erwartungen der Benutzer entspricht, die sich aus Erfahrungen mit bisherigen Arbeitsabläufen und der bisherigen Benutzung des Systems bzw. anderer Systeme ergeben. Der zentrale Punkt der Erwartungskonformität ist die Konsistenz der Benutzungsschnittstelle (anwendungsintern und anwendungsübergreifend). Dazu gehört z.B. konsistente Präsentation (z.B. Platzierung der Informationen), konsistente Interaktion (z.B. wiederkehrende Ablaufschemata, einheitliche Funktionstastenbelegung), konsistente Sprache (z.B. einheitlicher Dialogstil, durchgängige Kommandosyntax), Konsistenz zur Arbeitsumgebung (wenn eine Papiervorlage als Quelle existiert, sollte das Bildschirmformular möglichst ähnlich aussehen, gleicher Wortschatz wie in der Arbeitsumgebung), Konsistenz zwischen Bildschirminhalt und Ausdruck („WYSIWYG“-Prinzip, What You See Is What You Get).

Steuerbarkeit (controllability)
Je mehr der Benutzer in der Lage ist, den gesamten Dialogablauf zu beeinflussen, umso steuerbarer ist die Software für ihn. Der Benutzer kontrolliert die Software und nicht die Software ihn. Die Initiative für einen weiteren Arbeitsschritt verbleibt immer beim Benutzer. Der Benutzer muss z.B. die Möglichkeit haben den Dialog zu unterbrechen bzw. fortzusetzen sowie innerhalb eines Dialoges vor- und zurückzugehen, ohne dass er auf eine fixe Bearbeitungssequenz festgelegt wird. Nur so kann er die Geschwindigkeit und die Reihenfolge eines Dialoges nach seinen Bedürfnissen steuern. Die Geschwindigkeit des Dialogssystems sollte immer unter der Kontrolle des Benutzers bleiben. Sie darf nicht die Arbeitsgeschwindigkeit des Benutzers bestimmen.

Fehlertoleranz (error tolerance)
Ein Dialog sollte sich dadurch auszeichnen, dass Eingabefehler zum einen weitgehend verhindert, zum anderen fehlerhafte Eingaben bzw. zu einem falschen Zeitpunkt ausgeführte Funktionen leicht zu korrigieren sind. Das heißt zum Beispiel, dass die Software Eingaben auf das korrekte Format überprüft. Wenn das System Fehler automatisch korrigieren kann, sollte es dem Benutzer über die Ausführung der Korrektur informieren und ihm Gelegenheit ge-

ben, die Korrektur zu überschreiben. Die Software sollte eine Funktion enthalten, die dem Benutzer erlaubt, Dialogschritte rückgängig zu machen. Dadurch wird der Benutzer u. a. ermutigt, auch neue, ihm noch unbekannte Teile der Software zu erkunden. Fehlermeldungen sollten verständlich, sachlich, konstruktiv und einheitlich strukturiert, formuliert und angezeigt werden. Fehlermeldungen sollten keine Werturteile enthalten (z.B. „Unsinnige Eingabe“). Folgenschwere und irreversible Anweisungen (z.B. Löschen von Daten in der Datenbank) müssen in jedem Fall vor der Ausführung noch einmal bestätigt werden, um deren versehentliche Durchführung zu vermeiden.

Individualisierbarkeit (suitability for individualization)
Individualisierbare Dialogsysteme erlauben es dem Benutzer, flexible Anpassungen an die Erfordernisse der Arbeitsaufgabe und/oder an seine Vorlieben und seine speziellen Fähigkeiten vorzunehmen. Dies kommt im Rahmen der betrieblichen Softwareentwicklung selten vor, da die Software speziell für bestimmte Aufgabenstellungen entwickelt wird. Anpassbarkeit ist aber dann sinnvoll, wenn zum einen die Aufgabenstellung eine gewisse Variabilität oder Dynamik aufweist, zum anderen Bildschirmarbeitsplätze abwechselnd von mehreren Benutzern benutzt werden. Um einen optimalen Zugriff auf die jeweils benötigten Aufgabenbereiche zu gewähren, sollte dann der Einstiegsdialog individuell konfigurierbar sein. Der Profi bevorzugt oft andere Dialogtechniken, wie z.B. das Arbeiten mit der Tastatur anstatt der Maus. Das Dialogsystem sollte daher alternative Eingabetechniken unterstützen. Je besser die Erfordernisse, die aus der Arbeitsaufgabe, der Arbeitsumgebung und den Benutzermerkmalen resultieren, bekannt sind, um so geringer sollte das Ausmaß der Möglichkeiten für eine individuelle Anpassung des Dialogsystems sein.

Lernförderlichkeit (suitability for learning)
Die Zeit, die ein Benutzer benötigt, um den Umgang mit einer Anwendung zu erlernen, wird maßgeblich von der sprachlichen und konzeptionellen Ausgestaltung der Oberfläche beeinflusst. Dieser Grundsatz ist wichtig, um dem Benutzer das Gesamtverständnis des Dialogsystems zu erleichtern. Im Gegensatz zur Selbsterklärungsfähigkeit wird mit Lernförderlichkeit das Langzeitgedächtnis des Benutzers angesprochen. Als förderlich zur Verringerung der Einarbeitungszeit erweist es sich, folgendes zu berücksichtigen: Abkürzungen und Kurzbefehle sind sinnfällig und griffig zu wählen. Dinge, die gleich aussehen, sollten immer das gleiche tun. Dialogabläufe sollten in ihrer Grundstruktur immer gleich aufgebaut sein. Wichtige Lernstrategien, wie z.B. Learning by Doing sollten unterstützt werden.

Die Aufgabenangemessenheit sollte laut Ulich (2001:374) in einem präziseren Sinne als die ISO 9241-10-Definition verstanden werden, nämlich als angebrachte Unterstützung von Aufgaben, die den arbeitspsychologischen Kriterien der Aufgabengestaltung erfüllen (Ulich 2001:Abschnitt 4.3.1.). Da Software-Gestaltung ja auch weitgehend Aufgabengestaltung ist, sollten diese Kriterien den Merkmalen der Dialoggestaltung vorgeordnet werden (s. Abbildung: Merkmale benutzerorientierter Dialoggestaltung; Ulich 2001:369, Ulich 1986: 105 ff.).

Die EN ISO 9241-10 bietet Hilfestellungen bei der Konzeption, Gestaltung und Evaluation von Bildschirmarbeitsplätzen und definiert Mindestforderungen für die ergonomische Gestaltung von Software. Die Art und Weise, in der jeder einzelne Grundsatz der Dialoggestaltung umgesetzt werden kann, hängt von den Merkmalen des Benutzers, für den das Dialogsystem gedacht ist, den Arbeitsaufgaben, der Arbeitsumgebung und der jeweils eingesetzten Dialogtechnik ab. Dabei ist zu beachten, dass es Eigenschaften der menschlichen Informationsverarbeitung gibt, die für alle Personen weitestgehend gleich sind, weil sie sich aus der menschlichen Physiologie und Wahrnehmungspsychologie herleiten. Die software-ergonomischen Normen weisen aber meist Richtliniencharakter auf. Denn es fehlen hier die präzise Festlegung auf quantitative Werte von Produkteigenschaften (s. EN ISO 1995).

2.6 Klassifikation von Evaluationsmethoden

Im Bereich der software-ergonomischen Evaluation wird man häufig mit den nachstehenden Begriffen konfrontiert: harte, weiche, formative, summative, quantitative, qualitative, subjektive, objektive, analytische, heuristische, empirische, formale, informelle, experimentelle, leitfadenorientierte, theory-based, user-based Evaluation (s Görner & Ilg 1993). Dabei lässt sich die Art der Datenerhebung auf einem Kontinuum zwischen **subjektiven** und **objektiven** Verfahren darstellen (siehe Abbildung 2). Subjektive Evaluationsmethoden knüpfen unmittelbar an die Beurteilung durch den Benutzer an. Bei subjektiven Evaluationsmethoden werden eher „weiche“ Daten gewonnen, ob die Benutzung des Systems bequem, angenehm, klar, einsichtig ist . Bei objektiven Methoden versucht man, subjektive Einflüsse weitgehend auszuschalten. Für die Systemevaluation bedeutet die Verwendung harter Methoden die Erhebung quantitativer, statisch abgesicherter Daten, wie beispielsweise Ausführungs- und Lernzeiten und Fehlerraten. Zwischen den subjektiven und objektiven liegen die **analytische** (**leitfadenorientierte**) Evaluation durch Experten und die **empirische** (usability assessed by testing the interface with real users) Evaluation. In Abbildung 2 sind die Datenerhebungsverfahren in Abhängig-

keit von der Objektivität der Datenerhebung und dem Grad der Benutzerbeteiligung abgebildet.

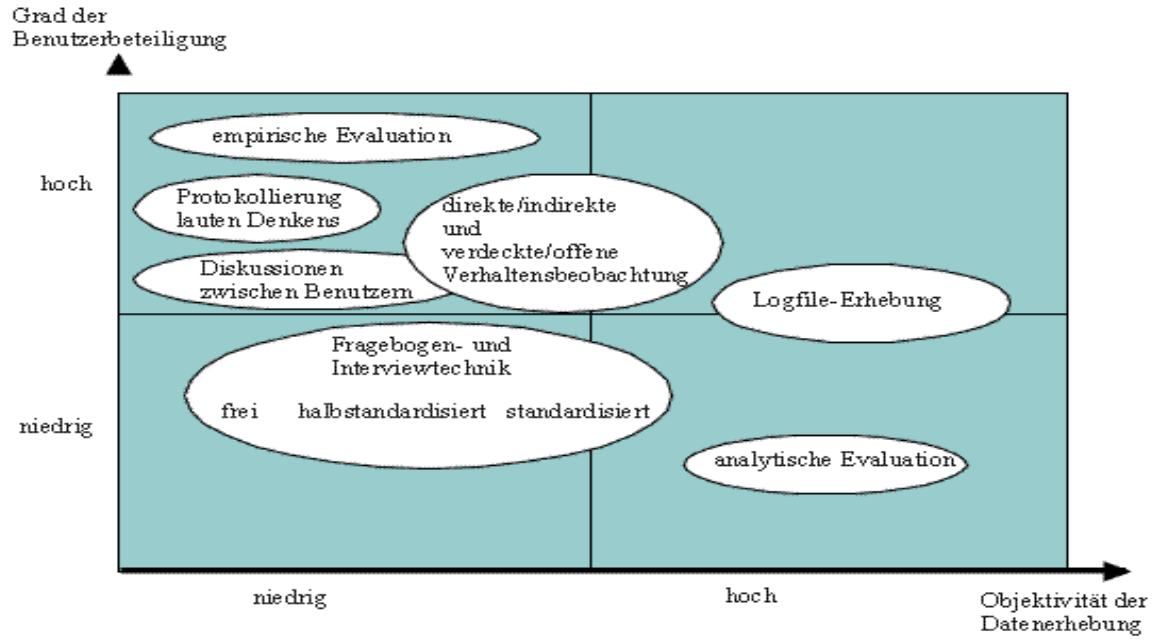

Abbildung 2: Datenerhebungsverfahren für die Evaluation (s. Görner & Ilg 1993)

Für die Evaluation von Benutzungsschnittstellen finden die nachfolgenden Methoden ihren Einsatz, die die Evaluationsbereiche unterschiedlich abdecken.

2.6.1 Objektive Methoden

Hier wird versucht, die subjektiven Einflüsse weitgehend auszuschalten. Objektive Methoden ermitteln harte Daten, beispielsweise präzise Angaben über Bearbeitungszeiten und Fehlerzahlen. Ziel der objektiven Methoden ist die Beobachtung der tatsächlichen Benutzung der Anwendung. Die Beobachtung ohne technische Unterstützung erfolgt mit Hilfe von Beobachtungsprotokollen, die in der Regel in strukturierter Form vorliegen und die zu beobachteten Sachverhalte vorgeben. Der Beobachter kann sich beim Ausfüllen der Protokolle passiv verhalten oder er kann an die beobachtete Person Fragen zur näheren Abklärung bestimmter Sachverhalte stellen (Beobachtungsinterview). Bei der Evaluation von software-ergonomischen Fragestellungen spielt die technische Beobachtung eine wichtige Rolle. Die Beobachtung wird vom System übernommen, indem es alle Interaktionsschritte der Benutzer aufzeichnet. Dieses Vorgehen wird als Logfilerecording bezeichnet. Es erfasst alle Bedienhandlungen des Benutzers an den vorhandenen Eingabegeräten. Der Log-

file ist eine ASCII-Datei, die durch das Statistikpaket (SPSS) ausgewertet werden kann (s. Frieling & Sonntag 1999:137). Die Schwachstelle vom Logfilerecording ist, dass sie das Nutzen von Handbüchern ebenso wenig erfassen kann wie Gesten oder Mimik. Ferner lassen sich die Überlegungen des Benutzers, die zu dem aufgezeichneten Verhalten geführt haben und welche Vorstellungen der Benutzer von der jeweiligen Anwendung hat, nicht mit diesen Verfahren ermitteln. Daher werden als Ergänzung zu den automatischen Protokollierungsmethoden Videoaufzeichnungen eingesetzt. Für die Auswertung der Aufzeichnungen ist es wichtig, dass die Videoaufnahmen den im Protokoll aufgezeichneten Interaktionen zeitlich genau zugeordnet werden können. Exemplarisch wird in Abbildung 3 ein Logfile aus 10 Komponenten dargestellt.

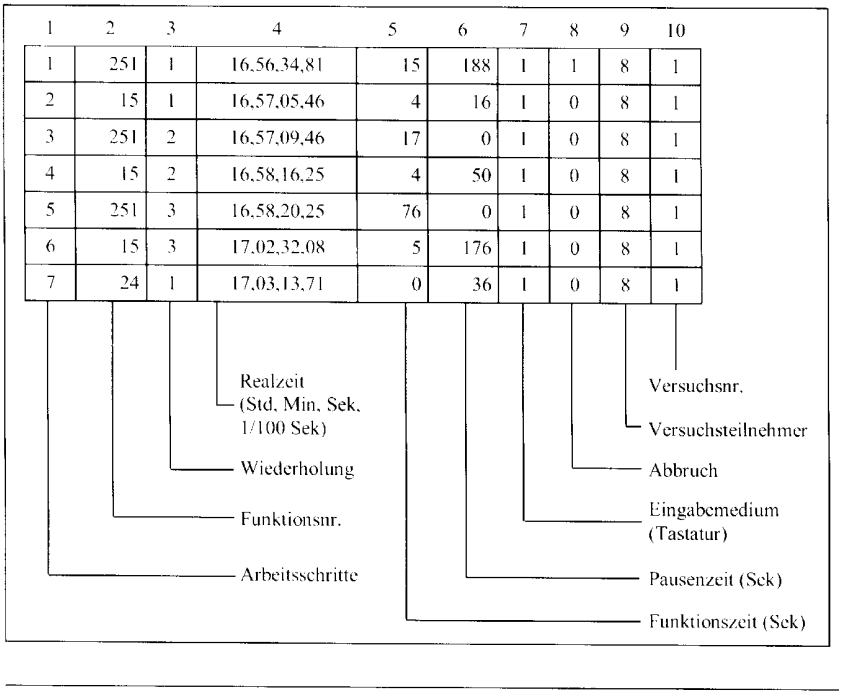

Abbildung 3: Ausschnitt eines Logfileprotokolls aus dem System APC (Frieling & Sonntag 1999:137)

Beispielsweise werden bei der Keystroke Level Methode die Anzahl der Tastatureingaben und die Anzahl der Mausklicks gezählt. Für jeden Klick und jeden Anschlag wird die Zeit gemessen. Am Ende steht für eine zu bearbeitende Aufgabe ein Zeitwert als Ergebnis. Das System kann nun verändert werden und im zweiten Durchlauf überprüft werden, ob eine Verbesserung eingetreten ist. Man kann auch auf diese Weise zwei konkurrierende Systeme miteinander vergleichen. Leider erhält man auch hier keine Kontextinformationen und Verbesserungsvorschläge (s. Bevan & Macleod 1994). Ebenfalls

erhält man keine Angaben über besonders gut gestaltete Komponenten. Es wird kommentarlos gearbeitet. Durch einen zusätzlich ausgestellten Fragebogen kann dieser Nachteil aber ausgeglichen werden.

Vor- und Nachteile
Der bedeutende Vorteil objektiver Methoden besteht darin, dass subjektive Einflüsse der Bewertung ausgeschaltet werden, was bei der Generalisierung der Ergebnisse eine wichtige Rolle spielt. Ein wesentlicher Nachteil besteht darin, dass objektive Evaluationsmethoden i.d.R. aufwendig durchzuführen sind und daher auf den Einsatz in Untersuchungslabor (s. Abbildung 4) beschränkt sind.

2.6.2 Subjektive Methoden

Bei subjektiven Methoden steht die Bewertung der Benutzungsschnittstelle durch den Benutzer im Vordergrund. Es werden hier sogenannte weiche Daten ermittelt, beispielsweise ob die Benutzung des Systems einfach, angenehm und verständlich ist. Vorwiegend werden hier Befragungen der Benutzer eingesetzt. Dabei können die Befragungen per Fragebogen oder als Interview erfolgen. Es ist darauf zu achten, dass durch die Art der Fragestellung die Bewertung nicht beeinflusst wird. Subjektive Methoden lassen sich mit verhältnismäßig geringen Aufwand durchführen. Ein erheblicher Arbeitsaufwand entsteht bei der Auswertung, wenn viele Fragen gestellt werden, und wenn die Fragen eine frei formulierte Beantwortung zulassen. Mit subjektiven Methoden lässt sich die Akzeptanz des Systems feststellen.

Vor- und Nachteile
Die Vorteile der subjektiven Methoden liegen darin, dass man Rückschlüsse bezüglich der Akzeptanz ziehen kann, eine wenig aufwendige und leichte Durchführung möglich ist und sie zur Eingrenzung unstrukturierter Probleme eingesetzt werden kann. Die Nachteile bestehen darin, dass subjektive Methoden anfällig sind für Übertreibungen, es zu hohen Ausfallquoten beim Rücklauf von schriftlichen Antworten kommen kann, Suggestionen durch die Untersuchungsfragestellung gefördert werden, eine Vielzahl an Daten produziert werden und sich damit die Auswertung aufwendig gestalten kann und dass von den Befragten i.d.R. nur einfache Fragen beantwortet werden können (s. Oppermann & Reiterer 1994).

2.6.3 Leitfadenorientierte Evaluationsmethoden

Bei den leitfadenorientierten Evaluationsmethoden wird das Expertenurteil am häufigsten eingesetzt. Dabei wird das Computersystem durch einen Exper-

ten geprüft, der sich an software-ergonomischen Fragestellungen orientiert. Diese Verfahren sind demzufolge subjektiv, da ein Subjekt aufgrund seiner Einschätzung eine software-ergonomische Fragestellung selbst prüft und beantwortet. Diese Verfahren sind dementsprechend auch objektiv, da die software-ergonomischen Prüfkriterien soweit sie operationalisiert und präzisiert sind, dass der Prüfer seine Bewertung aufgrund unmissverständlicher Testvorschriften gibt (s. Oppermann & Reiterer 1994). Werden indessen dem Evaluator die Prüfkriterien und einzusetzenden Methoden vorgegeben, so sprechen Oppermann und Reiterer (1994) von einem methodegeleiteten Expertenurteil. In einem Prüfleitfaden sind die anzuwendenden Methoden und Prüfkriterien festgehalten. Das EVADIS II-Verfahren zählt zu den Prüfverfahren für Expertenurteile, die für die Prüfung von software-ergonomischer Qualität eingesetzt wird. EVADIS II ist ein Verfahren zur Bewertung der software-ergonomischen Qualität von Benutzungsschnittstellen im Bürobereich. Das Verfahren besteht aus fünf Verfahrensschritten, die im Detail im Handbuch zum Verfahren beschrieben sind. Die Evaluierung wird ohne Benutzer durchgeführt, es wird in erster Linie also nicht nach Nutzungsproblemen oder Mängeln aus Sicht der Benutzer gesucht. Die Bewertung einer Software mit EVADIS muss ein Experte mit grundlegenden softwaretechnischen Wissen durchführen. Eine Einarbeitung in das zu evaluierende Softwareprodukt ist nötig. Das Kernstück von EVADIS II ist ein Leitfaden mit 150 detaillierten Prüffragen, die als methodengeleitetes Expertenurteil gedacht sind. Der Leitfaden wird ergänzt durch einen Fragebogen zur Erfassung von Benutzereigenschaften (12 Fragen) und Fragen zur Aufgabengestaltung. Die Befragung der Benutzer dient dazu, sie in die drei verschiedenen Benutzergruppen einzuordnen: „erfahrene und regelmäßige Benutzer“, „unerfahrene und regelmäßige Benutzer“ und „unerfahrene und gelegentliche Benutzer“. Begründet auf dieser Einteilung können die software-ergonomischen Kriterien in ihrer Bedeutung für die aktuelle Bewertung gewichtet werden. EVADIS II enthält eine Anleitung zur Erstellung des Prüfberichts, um die Prüfergebnisse transparent und vergleichbar zu dokumentieren.

2.6.4 Experimentelle Methoden

Sie werden zur Überprüfung theoretischer Annahmen und zum Vergleich verschiedener Systeme eingesetzt. Der Einsatz experimenteller Methoden zur Überprüfung und Weiterentwicklung theoretischer Annahmen über die Mensch-Computer-Interaktion ist problematisch, weil es immer eine große Anzahl von unabhängigen Variablen gibt, von denen bei einem Experiment nur ganz wenige gesondert betrachtet und variiert werden können. Der Einfluss der übrigen ist häufig schwer einzuschätzen. Auch aus der großen Anzahl der abhängigen Variablen lassen sich nur wenige untersuchen, so dass

nicht immer klar ist, wie weit sich das Ergebnis eines Experimentes verallgemeinern lässt. Bei experimentellen Untersuchungen spielen Benchmark Tests eine wichtige Rolle. Hier werden beispielsweise Systeme anhand von standardisierten Aufgaben im Vergleich untersucht. Bis auf die verwendeten Systeme werden hier möglichst alle unabhängigen Variablen konstant gehalten und als abhängige Variable z.B. Ausführungszeit, Fehlerhäufigkeit und Belastung der Benutzer gemessen. Damit lassen sich relative Aussagen über die ergonomische Gestaltung von Anwendungen erhalten. Wenn bei einem Benchmark-Test ein Anwendungssystem besser als ein anderes ist, heißt dies noch nicht, dass dies auch bei einer Veränderung der unabhängigen Variablen also z.B. bei einer anderen Aufgabe oder bei Benutzern mit anderen Vorerfahrungen gilt. Experimentelle Methoden sind sehr aufwendig und können meist nur in speziellen Untersuchungslabors (s. Abbildung 4) durchgeführt werden. Ihre Anwendung ist daher weitgehend auf die Forschung begrenzt.

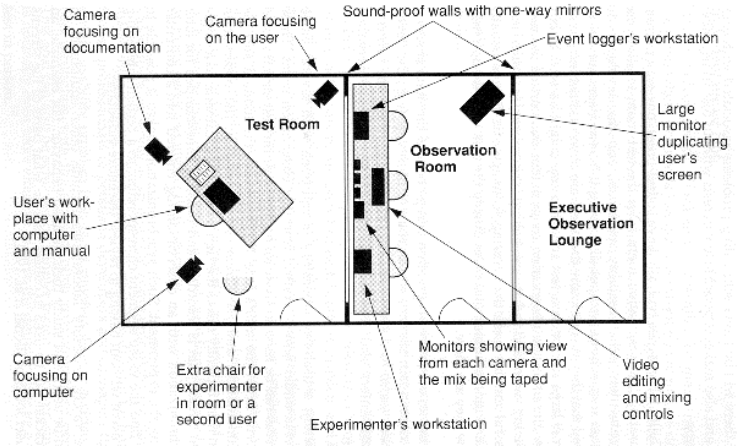

Abbildung 4: Usability-Labor (Nielsen 1993:201)

Vor- und Nachteile

Die Vorteile eines experimentellen Verfahrens liegen darin, dass sie bei gründlicher Durchführung Kausalschlüsse über die Art der Mensch-Rechner-Interaktion ermöglicht und wesentliche Beiträge zur Theorienbildung liefert. Der Nachteil bei der Planung von Experimenten liegt in der angemessenen Bestimmung der abhängigen und unabhängigen Variablen sowie in einer angemessenen Wahl der Untersuchungsumgebung. Oppermann und Reiterer (1994) betonen, dass jede der oben genannten Methoden ihre spezifischen Vorteile und Evaluationsschwerpunkte aufweisen und empfehlen daher eine Kombination der verschiedenen Evaluationsmethoden. Um auf spezifische Bedürfnisse aus der Praxis Rücksicht zu nehmen, wie einfache Handhabung, werden in einer Übersicht (s. Tabelle 1) die Kombination von Evaluationsme-

thoden dargestellt (s. Oppermann & Reiterer 1994: 348). Demzufolge finden die experimentellen Evaluationsmethoden keine Berücksichtigung.

Tabelle 1: Kombination von Evaluationsmethoden (Oppermann & Reiterer 1994:348)

Art der Methode	**Gegenstand der Evaluation**	**Ziel, Methode, Werkzeug**
subjektiv	Benutzer	**Ziel:** Erfassen von Benutzereigenschaften **Methode:** Befragung (schriftlich oder mündlich) **Werkzeug:** Fragebogen, Interviewleitfaden
	Aufgabe und Organisation	**Ziel:** Erfassen der Arbeitszufriedenheit **Methode:** Befragung (schriftlich oder mündlich) **Werkzeug:** Fragebogen, Interviewleitfaden
	Benutzungsschnittstelle der Software (Evaluationsschwerpunkt)	**Ziel:** Erfassen Benutzerakzeptanz **Methode:** Befragung (schriftlich oder mündlich) **Werkzeug:** Fragebogen, Interviewleitfaden
objektiv	Aufgabe und Organisation	**Ziel:** Erfassen der ergonomischen Qualität der Aufgabe **Methode:** Beobachtung **Werkzeug:** Arbeits- und Aufgabenanalyse.
	Benutzungsschnittstelle der Software (Evaluationsschwerpunkt)	**Ziel:** Erfassen der ergonomischen Qualität der Software (qualitativ) **Methode:** Beobachtung **Werkzeug:** Logfilerecording, Videoaufzeichnung
leitfadenorientiert (Methodenschwerpunkt)	Benutzungsschnittstelle der Software (Evaluationsschwerpunkt)	**Ziel:** Erfassen der ergonomischen Qualität der Software (qualitativ) **Methode:** Expertenurteil **Werkzeug:** Leitfaden, Evaluations-Software

2.7 Klassifikation der Prüfverfahren

Die Ergonomieprüfungen bedienen sich nicht nur naturwissenschaftlicher Methoden, sondern beziehen sozialwissenschaftliche Ansätze und Verfahren

mit ein. Die unterschiedlichen Prüfverfahren lassen sich wie folgt klassifizieren:

Nach den Akteuren
Expertengestützte Verfahren: Diese umfassen Inspektionen, Begutachtungen, Walkthroughs. Die Prüfung findet vorzugsweise gegen Checklisten und Prüfaufgaben statt.
Nutzerbezogene Prüfungen: Hierzu gehören Usability-Tests, Befragungen und teilnehmende Beobachtungen. Die Prüfung findet anhand von Prüfaufgaben mit Probanden oder „echten" Nutzern statt.

Nach dem Zeitpunkt
Formative Evaluation, d.h. entwicklungsbegleitende Prüfung: Formative Evaluationen haben eine in der Zukunft gerichtete Perspektive und trachten danach, die verbesserungsbedürftigen Bestandteile eines Programms zu identifizieren. Die Zielsetzung der formativen Evaluation ist also die unmittelbare Anwendung ihrer Ergebnisse in die Praxis. (Erkennung des Ist-Zustandes zum Soll-Zustand und Behebung der „Differenz").
Summative Evaluation, d.h. Prüfung des fertigen Produkts: Summative Evaluationen werden geplant, um den vollen Umfang der Effekte zu erfassen, die zu einem Zeitpunkt in der Vergangenheit durch das Objekt der Evaluation hervorgerufen wurden. Die Zielsetzung der summativen Evaluation ist es also, die Effekte der Evaluation zu messen, und zu betrachten, ob wirklich eine Verbesserung stattgefunden hat.

Nach dem Prüfungsort
Prüfungen unabhängig vom Einsatzort/Arbeitsplatz
Prüfungen am Einsatzort/Arbeitsplatz

Nach dem Umfang der Prüfung
Teilprüfungen: Sie betreffen die Modul-, Abschnitts- und Komponentenebene.
Vollständige Prüfungen

Nach dem angewendeten Beweisprinzip
Verifikation, d.h. Nachweis der Fehlerfreiheit.
Falsifikation, d.h. Unterstellung der Fehlerfreiheit bis zum Gegenergebnis.

Für das Testen der Benutzerfreundlichkeit der Benutzungsschnittstellen sind Testmethoden erforderlich. Solche Methoden sollten sowohl robust als auch relativ einfach sein. In den folgenden Abschnitten wird näher auf die komplementären Prüfverfahren - Inspektionsmethoden und Usability-Tests - eingegangen.

3 Inspektionsmethoden

Der finanzielle Aufwand ist trotz aller Nutzen-Kosten-Vorteile der wichtigste Grund, der Unternehmen davon abhält, die Usability ihrer Produkte zu überprüfen. Um dieses Hürde zu überwinden, wurden als Alternative bzw. Ergänzung zu klassischen Usability-Tests die Inspektionsmethoden entwickelt. Zu den Inspektionsmethoden zählen eine Reihe von Methoden, bei denen Gutachter usability-relevante Gesichtspunkte eines Produktes („Software“) überprüfen. Die Gutachter können dabei Endanwender oder Usability-Experten sein. Dieser Ansatz baut auf der Fähigkeit der Gutachter auf, Probleme der Endanwender vorherzusagen. Grundsätzlich werden für die Inspektionsmethoden 3-12 Gutachter bzw. Prüfer benötigt. Innerhalb dieser Bandbreite ist die Anzahl abhängig von der Expertise der Prüfer und der Komplexität des Produktes. Die Überprüfung der Software durch Experten muss strukturiert und im Vorfeld geplant werden. Sie sind im allgemeinen leitfadenorientiert. Die Ergebnisse beim Einsatz mehrere Experten müssen miteinander vergleichbar bleiben.

3.1 Unterschiedliche Ausprägungen der Inspektionsmethoden

Die verschiedenen Inspektionsmethoden weisen Berührungspunkte und Unterschiede auf, die hier kurz skizziert werden:

Die Expertise der Experten
Allen Methoden ist gemeinsam, dass sie von speziellen Experten geplant und durchgeführt werden. Dabei weisen die Experten Kenntnisse in Software-Ergonomie und Erfahrung im Umgang mit dem Anwendungsprogramm auf. Reale Benutzer des Systems finden bei diesen Methoden keine Berücksichtigung.

Die Anzahl der Evaluatoren
Einige Inspektionsmethoden werden in Gruppen durchgeführt, andere zwar mit mehreren Experten, jedoch individuell und nacheinander.

Ganzheitlichkeit der Inspektion
Grundsätzlich sind diese Methoden keine ganzheitlichen Evaluationsmethoden. Die Methode des Cognitive Walkthrough untersucht beispielsweise die Benutzeroberfläche auf deren Explorationsfreudigkeit. Andere Methoden gehen auf ganz andere Aspekte der Software-Ergonomie ein.

3.2 Die Usability-Inspektionsmethoden

Folgende Usability-Inspektionsmethoden werden beschrieben:

- Cognitive Walkthrough
- Heuristische Evaluation
- Standard Inspection
- Feature Inspection
- Consistency Inspection
- Focus Group

3.2.1 Cognitive Walkthrough

Ein Cognitive Walkthrough (CW) ist eine aufgabenorientierte Inspektionsmethode, d.h. ohne Benutzer. Der Usability-Experte erkundet die Funktionalitäten im Interesse eines imaginären Benutzers. Dabei geht er davon aus, dass der Benutzer die Software erkunden und den Weg des geringsten kognitiven Aufwands gehen wird. Der Usability-Experte legt deshalb Wert auf gute Erlernbarkeit der Software und ermittelt für jede mögliche Aktion den voraussichtlichen kognitiven Aufwand. Der CW stützt sich auf Theorien zum explorierenden Lernen und Problemlösen. Dabei fördern Problemlösestrategien die Entdeckung der korrekten Aktionen. Ein CW sollte einem Usability-Test vorausgehen, damit die offensichtlichen Usability-Probleme vor dem Test behoben werden können (s. Jeffries, Miller, Wharton & Uyeba 1991; Lewis & Wharton 1997).
Die Durchführung eines CW umfasst folgende Schritte (s. Lewis & Wharton, 1997): Vorbereitung, Analyse und Follow-up.

In der **Vorbereitungsphase** müssen folgende Schritte erfüllt sein:

- Benutzercharakteristiken: Eine allgemeine Beschreibung der möglichen Benutzer des Systems, deren Kenntnisstand und Erfahrung mit Computersystemen.
- Beispielaufgaben (Szenarien): eine präzise Beschreibung einer oder mehrerer typischer Arbeitsaufgaben, die mit dem System erledigt werden sollen.
- Beschreibung des Interfaces: Es ist genau zu beschreiben, was der Benutzer während der einzelnen Bedienschritte zu sehen bekommt.
- Handlungssequenzen: Eine Liste aller Möglichkeiten, wie die entsprechende Arbeitsaufgabe gelöst werden kann.
- Welche Folge von Aktionen ist für jede Aufgabe auszuführen?
- Gibt es für eine Aufgabe mehrere korrekte Lösungswege, so wird meist der übliche oder auch der problematischste ausgewählt.

Analyse

Hier beginnt der eigentliche CW. Vom Grundgedanken her sollte der CW von einzelnen Gutachtern oder einer Gruppe von Entwicklern des Systems durchgeführt werden. Für jede Aktivität wird ein Protokoll angefertigt, das die nachfolgenden Fragen an jeden Arbeitsschritt beinhaltet. Bei jeder diese Fragen müssen die Evaluatoren entscheiden, ob keine, einige, mehr als die Hälfte oder die meisten Benutzer Probleme haben werden (s. Lewis & Wharton 1997).

- Beschreibung des Ziels der Benutzer
- Nächste atomare Aktion, die die Benutzer ausführen können
  - Ist diese Aktion vorhanden?
  - Führt diese Aktion zum Ziel?
- Wie bekommt der Benutzer eine Beschreibung für diese Aktion?
  - Gibt es Probleme beim Zugriff ?
- Wie verbindet der Benutzer die Beschreibung mit der Aktion ?
  - Gibt es Probleme beim Verbinden?
- Sind alle anderen Aktionen weniger geeignet?
- Wie führt der Benutzer die Aktion aus?
  - Gibt es Probleme?
- Falls die Aktion länger dauert, wird der Benutzer vor der Aktion davon in Kenntnis gesetzt ?
- Die Aktion wird durchgeführt. Beschreibe die Antwort des Programms
  - Ist es offensichtlich, dass die Aktion zum Ziel geführt hat?
  - Kann der Benutzer auf wichtige Informationen zugreifen?
- Antwort des Systems?
- Beschreibung des veränderten Ziels, falls nötig.
  - Ist es offensichtlich, dass das Ziel geändert werden muss?
  - Falls der Arbeitsschritt beendet ist, ist dies offensichtlich?

Follow-Up

Die durch die Evaluation erkannten Nutzungsprobleme und Ursachen werden aufgezeichnet. Nachfolgend werden Überlegungen zu Designalternativen angestellt und auch protokolliert. Damit wird versucht, die Ergebnisse der Analyse auf die kognitiven Fähigkeiten potentieller Benutzer anzupassen (s. Lewis & Wharton 1997).

Vor- und Nachteile

Die Vorteile des CW sind, das diese Methode im frühen Stadium der Software-Entwicklung eingesetzt werden kann. Denn es werden nicht nur Nutzungsprobleme aufgezeigt, sondern auch deren Ursachen erläutert und somit

begründete Systemveränderungen vorgenommen.
Nachteilig für die Technik des CW ist die Frage, ob die simulierte Sequenz angemessen ist und ob die Einschätzung des Evaluators hinsichtlich der kognitiven Fähigkeiten des späteren Benutzers korrekt ist.

3.2.2 Heuristische Evaluation

Die Heuristische Evaluation nach Nielsen (1993) ist ein pragmatisches und kostenorientiertes Verfahren, bei dem sich mehrere Ergonomieexperten mit der Benutzungsschnittstelle auseinandersetzen. Zur Begutachtung werden die folgenden zehn Heuristiken verwendet: „1) simple and natural dialogue; 2) speak the user's language; 3) minimize user memory load; 4) consistency; 5) provide feedback; 6) clearly marked exits; 7) shortcuts; 8) good error messages; 9) prevent errors, and 10) help and documentation" (Nielsen 1993:115ff.; Lin et al. 1997:269).

Die Heuristische Evaluation verläuft in folgenden Phasen:

Trainingssitzung
Die Gutachter erhalten (wenn nötig) eine detaillierte Einführung in die heuristische Evaluation.

Evaluation
Jeder Gutachter geht mit Hilfe einer Liste von standardisierten Heuristiken mehrmals durch die Benutzungsschnittstelle. Dabei muss er die Heuristiken interpretieren und auf ihre Einhaltung in der Software achten. Die Wiederholung dient dabei einer möglichst vollständigen Erfassung der Probleme, nachdem man sich mit den grundlegenden Funktionsweisen der Software vertraut gemacht hat. Grundsätzlich sollten zwei bis vier Durchgänge dafür ausreichend sein. Ein erster Durchlauf soll ein Gefühl für den Informationsablauf und die Funktionalität des Produktes vermitteln. Im zweiten Schritt konzentriert sich die Prüfung auf einzelne Bedienelemente, ohne ihre Stellung im Gesamtbild aus den Augen zu verlieren. Den Gutachtern werden Beobachter zur Seite gestellt, die die erkannten Probleme und die betroffene Heuristik protokollieren. Die Prüfungen dauern meistens eins bis zwei Stunden. Dies ist aber von der Komplexität und der Funktionalität des Produktes abhängig. Eine Kommunikation der Evaluatoren über ihre Ergebnisse wird erst nach Abschluss der Durchgänge gestattet.

Ergebnisse und Abschlusssitzung
Das Ergebnis der Heuristischen Evaluation ist eine Liste mit einzelnen Usability-Problemen mit Referenzeierungen zu den verletzten Prinzipien und Be-

gründungen, inwieweit diese Prinzipien verletzt wurden. Jedes Problem sollte daher genau beschrieben werden und auf die Heuristik, die sie verletzt, Bezug nehmen. Jedes Problem wird einzeln dokumentiert, damit später möglichst viele kritische Stellen behoben werden können. Hauptzweck der Abschlusssitzung ist es, die von den einzelnen Gutachtern gefundenen Usability-Probleme sowie Verbesserungsvorschläge zu diskutieren. Dies kann in einer Art Brainstorming vonstatten gehen.

Problembewertung („Serverity Ratings“)
Jeder Gutachter erhält eine Liste seiner eingereichten Probleme wieder und gibt eine Problembewertung („Severity“) ab, in wie weit gegen eine Heuristik verstoßen wurde. Die Problembewertung erfolgt gewöhnlich entlang dreier Dimensionen:

- **Problemhäufigkeit**
  Tritt das Problem während vieler oder weniger Interaktionssituationen auf?
- **Problemeinfluss**
  In welchem Ausmaß wird die Aufgabenbewältigung beeinträchtigt?
- **Persistenz**
  Ist das Problem leicht zu umgehen, sobald es bekannt ist?

Diese Dimensionen werden der spezifischen Situation angepasst und gewichtet. Das Ergebnis ist eine Maßzahl, die die eigentliche Problembewertung[3] darstellt.

- Ich stimme nicht zu, dass das überhaupt ein Usability-Problem ist.
- Nur ein kosmetisches Problem – braucht nicht behoben zu werden, solange keine zusätzliche Zeit zur Verfügung steht.
- Kleines Usability-Problem – Behebung erhält geringe Priorität.
- Großes Usability-Problem – sollte behoben werden, hohe Priorität.
- Usability-Katastrophe – sollte unbedingt behoben werden, bevor Produkt eingeführt wird.

Die Übereinstimmung der einzelnen Gutachter wird nach Kendall´s Tau berechnet, ein Koeffizient, der bei ordinalskalierten Daten verwendet werden darf (vgl. Bühl & Zöfel 2000:322f.; Bortz & Lienert 1998:247ff ; Benninghaus 2002:145f, 149ff). Eine andere, adäquates Bewertungsmethode ist die Überprüfung der Intercoderreliabilität, d.h. unterschiedliche Gutachter müssen bei der Analyse desselben Testmaterials mit denselben Methoden zu ver-

[3] Siehe URL: http://www.useit.com/papers/heuristic/severityrating.html

gleichbaren Resultaten kommen (vgl. dazu auch Lisch & Kriz 1978; Schnell, Hill & Esser 1989). Zur Bestimmung der Intercoderreliabilität kann der von Holsti (1969:140; zit nach Mayring 1996) vorgeschlagene Koeffizient verwendet werden, der die Quote der übereinstimmenden Einschätzungen verschiedener Kodierer ermittelt.

$$R = \frac{(Zahl\ der\ Kodierer)x(Zahl\ der\ übereinstimmenden\ Urteile)}{(Zahl\ der\ Kodierurteile)}$$

Ebenfalls empfiehlt es sich hier, die Problembewertungen von mehreren Gutachtern zu erheben, da die Problembewertungen eines Gutachters meist nicht aussagekräftig ist.

Anzahl der Gutachter

Durch Erkenntnisse mehrerer Untersuchungen schlägt Nielsen (1993) vor, mehrere Evaluatoren einzusetzen, da sich die Anzahl der gefundenen Fehler mit der Anzahl der eingesetzten Evaluatoren erhöht. Dies geschieht auch nur bis zu einem bestimmten Grad (s. Abbildung 5.). Ein einzelner Gutachter erkennt ca. 35% der Usability-Probleme. Nielsen (1993) empfiehlt, dass zwischen drei bis fünf Gutachter für eine Evaluation einzusetzen sind, da sie ca. 60-70% der Usability-Probleme finden. Ab zehn Evaluatoren verbessert sich das Ergebnis praktisch nicht mehr, obwohl noch nicht alle Fehler gefunden worden sind.

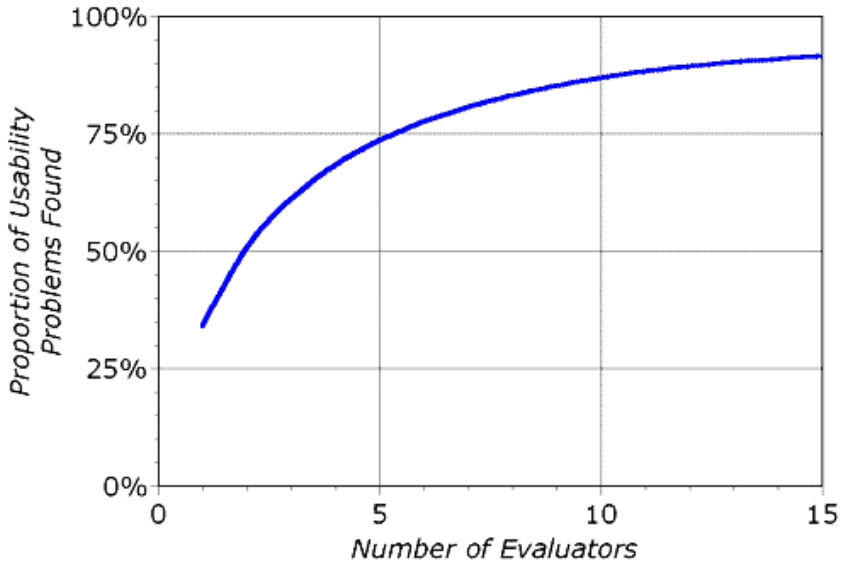

Abbildung 5: Verhältnis von Anzahl der Evaluatoren zu gefundenen Fehlern (Nielsen 1993:156)

Vor- und Nachteile
Die Vorteile der Heuristischen Evaluation liegen darin, dass sie gut einsetzbar ist, wenn Zeit und Geld knapp sind und qualitativ hochwertige Ergebnisse in kurzer Zeit liefert. Externen Mitarbeitern, die nicht an der Entwicklung beteiligt sind, fällt eine kritische Evaluation naturgemäß leichter (s. Nielsen 1993; Lin et al. 1997). Die Methode ist eher eine Inspektion zur Erfassung des „ersten Eindrucks“, den man von einem System gewinnen möchte. Sie wird meist beim Prototyping angewendet.
Die Nachteile der Heuristischen Evaluation bestehen nach Lin et al. (1997) darin, dass die Gutachter nicht die wirklichen Benutzer sind. Diese können zwar versuchen, sich in die Rolle der Benutzer zu versetzen, aber es ist wahrscheinlich, dass die echten Benutzer auch andere als die erkannten Probleme haben.

3.2.3 Standard Inspection

Nach Nielsen (1993) wird bei der standard inspection betrachtet, ob Vorschriften (z.B. Styleguides) korrekt angewendet wurden.

3.2.4 Feature Inspection

Hier werden typische Arbeitsaufgaben verwendet, um bestimmte Fehler zu finden: zu lange Sequenzen, unhandliche Abfolgen von Dialogschritten, Schritte, die normale Benutzer nie benutzen würden und Schritte, die ein erhebliches Wissen voraussetzen, um überhaupt erfolgreich zu sein (Nielsen 1993).

3.2.5 Consistency Inspection

Diese Methode wird überwiegend für große Projekte durchgeführt, um zu untersuchen, ob verschiedene Funktionen in allen Kontexten gleich ausgeführt werden können.

3.2.6 Focus Group

Focus Groups zählen zu den qualitativen Verfahren und orientieren sich dementsprechend an den Methoden der qualitativen Auswertung. Sie sind moderierte Gruppendiskussionen mit ausgewählten Endbenutzern. Ziel ist es, mit 5-8 Teilnehmern (s. Tabelle 2) in ca. 2-3 Stunden ein vorgestelltes Produktkonzept oder bestehendes Produkt zu bewerten. Hierbei ist darauf zu achten, dass die Gruppe sozial möglichst homogen zusammengesetzt ist, damit die

Kommunikationsprozesse nicht durch Statusunterschiede erschwert werden. Auch der Grad der Expertise sollte vergleichbar sein, um sicherzustellen, dass alle Personen „die gleiche Sprache sprechen“. Der Vorteil von Focus Groups ist, dass in kurzer Zeit und mit vertretbarem Aufwand Kundenmeinungen und Anregungen zu sammeln und damit eine bessere Grundlage für Entwicklungsentscheidungen zu gewinnen. Die folgenden Charakteristika weist eine Focus Group auf (s. Honold, 2000:72f.):

- Gruppeninteraktion: Zwischen den Teilnehmern findet eine Interaktion statt. Dies unterscheidet Focus Groups von normierten Gruppen oder Delphi-Gruppen, in denen keine echte Interaktionen zwischen den einzelnen Teilnehmern erfolgt.
- Die Bestimmung des Themas durch den Forscher: Dadurch unterscheiden sich Focus Groups von alltäglichen Gruppendiskussionen.
- Fokussierung: In der Diskussion wird der Focus durch den Forscher auf eine ausgewählte, geringe Anzahl von Fragestellungen gerichtet.

Die Sitzung wird als Video- oder Tonbandaufzeichnung dokumentiert. Damit sind spätere Detailanalysen möglich. Zusätzlich kann damit die Stimme der Endbenutzer firmenintern einem weiteren Personenkreis zugänglich gemacht werden (s. Nielsen 1993:214ff.)

Tabelle 2: Beispiel für Teilnehmerdaten der Focus Groups

Nr. (Vp)	Geschlecht	Alter	Beruf	Ausbildung	Nutzung seit
1	m	22	Designer	Fachhochschule	Feb 96
2	w	25	Student	Abitur	Mai 92
3	w	37	Angestellte	Abitur	März 94
...	...	...	...	...	...

Eine gute theoretische Grundlage für die Auswertung liefert hier die „Grounded Theory“ nach Glaser & Strauss (1967, 1998; Strauss & Corbin 1996). Das Ziel einer solchen Auswertung ist eine Konzeption neuer Modelle oder Theorien auf der Grundlage des zugrundeliegenden Datenmodells und weniger die Auswertung bereits bestehender Hypothesen.

4 Usability-Test (Einbeziehung der Benutzer)

Für den Bereich des Software-Testens wird von der folgenden Definition ausgegangen: „Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden“ (Myers 1999:4) Der Usability-Test simuliert den Praxisfall: Dabei soll sichergestellt werden, dass der Benutzer seinen Anforderungen gerecht mit dem System umgehen kann. Während des Usability-Tests lösen Versuchspersonen[4] Aufgaben mit dem zu testenden Programm, wobei ein Versuchsleiter anwesend ist, der im Notfall eingreifen kann. In der Regel lässt der Versuchsleiter die Versuchsperson alleine arbeiten und gibt gegebenenfalls eine kurze Einführung in das System. Mit Log-Dateien und Video werden die Benutzeraktivitäten aufgezeichnet und später ausgewertet. Dabei geben die Log-Files Aufschluss über die Interaktionszeiten (deskriptive Statistik[5]). An Hand dessen kann man berechnen (erschließende Statistik), an welchen Stellen die Versuchsperson(en) auf Schwierigkeiten stieß. In der Regel werden Usability-Tests im Labor durchgeführt. Sofern die Arbeitsumgebung eine wichtige Rolle spielt, ist ein Usability-Test am Arbeitsplatz dringend zu empfehlen. Schon kleine Stichproben von 3-6 Probanden reichen teilweise aus, um viele wesentliche Fehler zu entdecken. Voraussetzung für die Durchführung von Usability-Tests ist eine ergonomische Mindestqualität der Software. Nur wenn offensichtliche ergonomische Fehler, wie z.B. unverständliche Bezeichnungen beseitigt sind, kann ein Test wirkungsvoll eingesetzt werden. Ansonsten werden Benutzer möglicherweise von den ohnehin offensichtlichen Fehlern so beeinträchtigt, dass gravierende aufgabenabhängige Fehler verloren gehen. Ferner eignen sich Usability-Tests zur Bewertung unterschiedlicher Alternativlösungen und können so eine langwierige Diskussion um eine bessere Lösung erheblich abkürzen. Rubin (1994) definiert Usability Testing als ein Verfahren, die Software sowie je nach Fall auch ihre Schulungsmittel und Dokumentationen systematisch nach Usability-Kriterien hin zu überprüfen, indem ausgewählte Benutzer definierte Testaufgaben lösen müssen. Usability-Tests konzentrieren sich auf die Ermittlung ob ein Produkt leicht erlernbar und zufriedenstellend bei der Benutzung ist und die gesamte Funktionalität enthält, die die Benutzer wünschen.

---

[4] Versuchsperson gilt als Synonym für Testperson

[5] Siehe hierzu Bortz (1993) Kapitel 1

4.1 Maße für Usability

Für die Bewertung von Software ist die Festlegung von Evaluationskriterien, mit Hilfe derer die Bewertung vorgenommen werden kann, eine Voraussetzung. Als allgemeiner Gestaltungsgrundsatz ergonomischer Softwaregestaltung hat sich das Konzept der „Usability“ durchgesetzt. Die Benutzbarkeit eines Software-Produkts ist eine hypothetische Eigenschaft, die Software zugeschrieben wird, wenn sie beispielsweise „benutzerfreundlich“, „angenehm zu bedienen“, „geeignet zur Erfüllung einer bestimmten Aufgabe“ ist. Die Benutzbarkeit wird in mehrere Komponenten bzw. Attribute zerlegt. Sie bilden den Ausgangspunkt für die Definition von Testkriterien, die beim Usability-Testing herangezogen werden, um die Benutzerfreundlichkeit eines Softwaresystems zu bestimmen. Bei der Entwicklung und Evaluation von Benutzungsschnittstellen sollte ein benutzerzentrierter Ansatz verfolgt werden, um die Benutzbarkeit (Usability) des Dialoges hinsichtlich Effektivität (Effectiveness), Effizienz (Efficiency) und Zufriedenheit (Satisfaction) zu verbessern. Der Benutzer soll seine Aufgaben unter ergonomisch günstigen Bedingungen („Ergonomically favourable conditions“) erledigen können. Ebenfalls wird nach der ISO 9241-11[6] Definition Usability erfasst als: „the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use“ (ISO DIN 9241-11)

Die Usability-Komponenten der ISO 9241-11 werden wie folgt definiert:

- **Effectiveness**: „The accuracy and completeness with which specified users can achieve specified goals in particular environments“.
- **Efficiency**: „The resources expended in relation to the accuracy and completeness of goals achieved.“
- **Satisfaction**: „The comfort and acceptability of the work system to its users and other people affected by its use“ (ISO DIN 9241-11).

Nielsen (1993:26) hingegen definiert Usability anhand der folgenden fünf Usability-Attribute:

> „Learnability : The system shoud be easy to learn so that the user can rapidly start getting some work done whith the system.
> Efficeincy: The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.

[6] s. URL: http://www.cse.dcu.ie/essiscope/sm3/product/9241.html#ABSTRACT

> Memorability: The system should be easy to remember, so that the casual user is able to return to the system after some period not having used it, without having to learn everything all over again.
> Errors: The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.
> Satisfaction: The system should be pleasant to use, so that users are subjectively satisfied when using it; they like it.“

Um das zu untersuchende System quantitativ hinsichtlich der Benutzerfreundlichkeit bewerten und mit anderen Testergebnissen vergleichen zu können, muss die Usability gemessen werden. Die zur Verfügung stehenden Daten können für verschiedene Zwecke genutzt werden. Bei der iterativen Testdurchführung können Daten der vergangenen Testobjekte herangezogen werden, so dass die Verbesserung der Benutzerfreundlichkeit im Verlauf der Systementwicklung von den Entwicklern verfolgt werden kann. Weiterhin können Daten von Konkurrenzprodukten mit den eigenen zwecks Positionierung verglichen werden. Letztlich kann die Entscheidung zum Release eines Produktes von der Erfüllung zuvor festgelegter quantitativer Usability-Vorgaben abhängig gemacht werden. Nachfolgend wird näher auf Performance- und subjektive Daten eingegangen:

a) Performance-Daten, die sich direkt quantitativ messen lassen:

Zeit
Die am häufigsten gemessene Performancegröße ist die Zeit, die ein Benutzer benötigt, um eine gegebene Aufgabe ihm Rahmen des Tests zu bewältigen. Außerdem ist es die am einfachsten zu erfassende Größe. Die Aussagekraft ist in den meisten Fällen jedoch begrenzt, da hiermit die Frage verbunden ist, ob das Arbeitsergebnis auch von der gewünschten Qualität ist.

Effektivität
Es muss sichergestellt sein, dass die benötigten Ergebnisse einwandfrei erreicht werden. Um die Effektivität erfassen zu können, müssen zwei Faktoren in Verbindung gesetzt werden. Zum einen muss bestimmt werden, zu wie viel Prozent die gestellte Aufgabe erfüllt wurde („quantity“), zum anderen welche Qualität dieses Ergebnis aufweist („quality“). Das Resultat ist die Aufgabeneffektivität:

$$\text{Task effectiveness} = \frac{(\text{quantity} * \text{quality})\,\%}{100}$$

Die erforderlich Zeit zum Erreichen des Ergebnisses fließt in die Berechnung nicht mit ein.

Effizienz
Ein Werkzeug macht nur dann Sinn, wenn es die Arbeit erleichtert. Software muss also nicht nur benutzbar sondern auch nützlich sein. Die Arbeitserleichterung hängt besonders von der Senkung des notwendigen Bedienaufwandes, insbesondere der aufzuwendenden Denkarbeit („Wo verbirgt sich die Funktion ...?“) ab. Mit Hilfe der Effizienz kann bestimmt werden, ob das System für die Zielbenutzergruppe eine effiziente Unterstützung bei der Erfüllung ihrer Aufgaben ist. Hier werden die beiden Größen Zeit und Effektivität miteinander in Beziehung gesetzt. Das Resultat ist die Aufgaben-Effizienz:

$$\text{Efficiency} = \frac{effectiveness}{time}$$

Als Co-Faktor muss hier die Zeit, die für eine Aufgabe benötigt wird, berücksichtigt werden. Dies hängt direkt vom Wissens- und Erfahrungsstand des Benutzers ab. Daher muss definiert sein, auf welchen Benutzertyp diese Aussage bezogen ist, um verlässliche Aussagen über die Effizienz der Nutzung treffen zu können. In der Regel wird hier auf den erfahrenen Benutzer zurückgegriffen (s. Nielsen 1993).

Fehlerquote
Es handelt sich hier um die Anzahl der Fehler, die ein Benutzer beim Lösen der gestellten Aufgabe begeht. Hierbei wird eine detaillierte Kategorisierung der Fehlerarten vorgenommen, um die spätere Analyse des Tests zu erleichtern. Welche Einordnung sinnvoll ist, hängt vom konkreten Testobjekt und dem Untersuchungsziel ab (s. Nielsen 1993:S.32f.).

Success Rate
Sie ist definiert als Prozentsatz der Aufgaben, die der Benutzer richtig erfüllt. Die Größen Zeit und Qualität bleiben dabei unberücksichtigt.

b) Subjektive Daten, die individuell erfasst werden müssen:

Zufriedenheit
Sie wird definiert als die subjektive Einstellung der Benutzer gegenüber dem System oder dessen Teilaspekten, die im Rahmen des Usability-Tests erfasst werden. Diese Größe ist sehr bedeutungsvoll, da das subjektive Empfinden des Benutzers oft entscheidend für die Akzeptanz eines Systems ist. Erfasst wird die Zufriedenheit in der Regel durch Interviews und Fragebögen. Der Benutzer wird hier zu einzelnen Themen befragt und gibt den Grad seiner Zu-

stimmung bzw. Ablehnung an. Bei Fragebögen wird der Grad der Zustimmung oder Ablehnung gewöhnlich durch Antwortvorgaben ausgedrückt (s. Tabelle 3). Um diese Angabe quantifizieren zu können, wird häufig eine 5-stufige Likert-Skala verwendet (s. Nielsen 1993:33ff.). Dabei müssen die 5 Rating-Kategorien dem Themenkontext angepasst werden. Die Likert-Skala hat den Nachteil, dass der mittlere Skalenwert nicht eindeutig zu interpretieren ist (s. Bortz & Döring 2002:222f.; Schnell et al. 1989:187ff.).

Tabelle 3: Antwortvorgaben bei einer Likert-Skala (Schnell et al. 1989:188)

Stimme stark zu	Stimme zu	teils, teils	Lehne ab	Lehne stark ab
1	2	3	4	5

Natürlich können auch offene Fragen ohne feste Antwortvorgaben verwendet werden, um die subjektive Zufriedenheit zu ermitteln, wobei die Angaben für quantitative Analysen bedingt geeignet sind. Um die Zufriedenheit der Benutzer vollständig zu erfassen, sollten zwei Methoden kombiniert angewendet werden:

- Ein subjektiver Zufriedenheitsfragebogen sollte am Ende einer Studie, der eine einfache, gesamthafte Beurteilung des Systems liefert, eingesetzt werden.
- Beobachtung der Körpersprache des Benutzers zur Feststellung von Zufriedenheit oder Missfallen (Lächeln oder Stirnrunzeln), aber auch Lachen, Brummen oder Bemerkungen wie „cool“ oder „langweilig“. Ein geübter Beobachter kann mit Hilfe der zweiten Methode viel in Erfahrung bringen. Für weniger geübte Usability-Experten ist sie eine schwache und möglicherweise irreführende Datenquelle.

Zur ersten Methode ist zu erwähnen, dass subjektive Zufriedenheitsfragebogen allgemein unter dem Problem leiden, dass sie außerhalb des Kontexts angewendet werden: Sie stützen sich häufig auf die Erinnerung des Benutzers an eine unterhaltsame Erfahrung und nicht auf das tatsächlich Erlebte in dem Augenblick. Dieses kann verringert (wenn auch nicht ganz vermieden) werden, indem mehrere kleine Fragebögen bereits während des Tests ausgefüllt werden und nicht alle Fragen für einen größeren Fragebogen am Ende des Test aufgespart werden. Eine mangelnde Zufriedenheit bei den Benutzern ist oft (aber nicht immer) ein Indiz für Probleme in der Bedienbarkeit der Software. Mögliche andere Auslöser sind Probleme bei der Aufgabengestaltung (Unterforderung oder Überforderung) und der Umgebungsbedingungen (Raumgestaltung, Beleuchtung, Möbel etc.). Des weiteren gibt es noch den Aspekt Benutzerverhalten. Hier wird die Häufigkeit positiver oder negativer

Kommentare sowie Gestiken der Benutzer vermerkt. Laut Nielsen (1993) äußert sich dies in Verwunderung, Begeisterung, Überraschung, Verzweifelung und Frustration. Zur besseren Interpretation könnte nach einem Test durch ein Interview geklärt werden, was den Benutzer zu den Äußerungen verleitet hat.

Einprägsamkeit
Sie wird folgendermaßen untersucht: Es werden Testreihen durchgeführt, bei denen Benutzer nach einen bestimmten Zeitpunkt erneut mit dem System konfrontiert werden. Erfasst wird hier die Zeit zur Bewältigung bereits bekannter Aufgaben sowie die Anzahl der Fehler (Nielsen 1993:31f.). Diese Werte werden mit denen der ersten Testreihen in Beziehung gesetzt. So wird bestimmt, in welchem Maße sich die Benutzer an die Arbeitsfolgen und Aufgaben des Systems erinnern können.

4.2 Basiselemente eines Usability-Tests

Bei einem Usability-Test werden quasi-experimentell konkrete Nutzungssituationen durch repräsentative Endnutzer simuliert, um die Bedienbarkeit eines Produkts oder Prototypen zu überprüfen. Ein Usability-Test ist ein mehrstufiger Prozess, der Testvorbereitungen, den tatsächlichen Test, die Analyse der Testresultate, die Modifikation des Softwaresystems und des wiederholten Testens des Systems umfasst. Im folgenden werden relevante Basiselemente vorgestellt, die bei der Durchführung eines Usability-Tests Verwendung finden sollten

4.2.1 Definition eine Untersuchungsziels

Durch die Zielsetzung eines Tests wird bestimmt, was Gegenstand eines Usability-Tests ist und zu welchem Zeitpunkt der Test durchgeführt wird. Die Ziele des Tests werden in Form eines Fragenkatalogs aufgestellt, der mittels des Tests beantwortet werden soll. Die Fragen müssten möglichst präzise und einfach formuliert sein. Weiterhin wird die Richtung des Usability-Tests festgelegt (s. Mekelburg 2002):

Exploratives Testen
Dieser erste Typ findet seine Hauptanwendung in einer frühen Phase des Software-Entwicklungs-Prozesses (SWE-Prozesses). Im Zentrum der Untersuchung stehen die Denkprozesse der Benutzer, also die Frage, warum geht der Benutzer in dieser Art und Weise vor, bzw. warum bekommt der Benutzer einen bestimmten Eindruck vom System. Ziel ist es, zahlreiche Informationen über das mentale Modell des Benutzers zu sammeln, welches er vom vorliegenden Systementwurf hat, und in welchem Maße beides übereinstimmt.

Vergleichstest
Dieser Testtyp kann in jeder Phase des SWE-Prozesses eingesetzt werden. Das Produkt kann am Ende des SWE-Prozesses mit anderen Konkurrenzprodukten verglichen werden, um Stärken und Schwächen auf beiden Seiten zu identifizieren.

Bewertender Test
Wurde ein neues Produkt entworfen, so soll dieser Entwurf vor seiner weiteren Einführung auf Usability bewertet werden.
Hingegen unterscheidet Honold (2000: 91f.) bei der Zielsetzung des Usability-Testing zwischen summativem und formativem Usability-Testing. Bei einem summativen Usability-Testing evaluiert der Forscher die Güte eines Systems anhand zuvor festgelegter messbarer Kriterien der Performanz (Zeit, Fehlerhäufigkeit). Intention ist eine abschließende Bewertung eines Produktes, beispielsweise mit Produkt A ließ sich die Aufgabe mit 30% weniger Fehlern durchführen als mit Produkt B, oder: 80% aller Nutzer konnten 80% der Aufgaben ohne gravierende Usability-Probleme lösen. Das summative Usability-Testing ist dann einsetzbar, wenn mehrere technische Systeme miteinander verglichen oder wenn im Sinne einer Zertifizierung ein abschließendes Qualitätsurteil über ein Produkt gegeben werden soll. Um beim summativen Usability-Testing zu einer hohen Reliabilität und Validität (s. Kapitel 7.5) zu gelangen, lehnt sich dieses Verfahren stark an den Aufbau klassischer psychologischer Experimente an. Eine Interaktion zwischen Versuchsleiter und Versuchsperson soll möglichst vermieden werden. Die Anzahl der Versuchspersonen sollte relativ groß sein, um eine inferenzstatistische Aussage treffen zu können. Demgegenüber wird beim formativen Usability-Testing schon während des Prozesses des Usability-Engineerings iterativ eingesetzt. Dabei sollten möglichst umfangreiche Kenntnisse über die Art und Ursachen ermittelt werden, mit dem Ziel, auf dieser Grundlage die Gestaltung eines Produktes verbessern zu können. Im Mittelpunkt stehen sowohl quantitative und qualitative Daten. Häufig wird beim formativen Usabilty-Testing auf eine inferenzstatistische Auswertung verzichtet. Beim formativen Usability-Testing geht es darum, die Denk- und Handlungsstrukturen repräsentativer Benutzer zu erfassen, mangelnde Passungen zwischen dieser Struktur und dem technischen System zu erkennen und mögliche Ursachen dafür zu ergründen. Hierbei spielt die Methode des Lauten Denkens (s. Kapitel 4.4.5) eine herausragende Rolle.

4.2.2 Bestimmung der Stichprobe

Die Versuchspersonen sollten möglichst repräsentativ für die Population der Endbenutzer sein (s. Nielsen 1993:177). Die Auswahl der Versuchspersonen

muss daher zufällig aus dem potentiellen oder aktuellen Benutzerkreis erfolgen. Die Gruppengröße sollte aus statistischen Gründen mindestens sechs und aus ökonomischen Gründen maximal zwanzig bis dreißig Personen (pro Gruppe) getragen[7].

Die ausgewählten Versuchspersonen sollten das zu testende System nicht kennen. Da sich diese Bedingung aus diversen Gründen oftmals nicht einhalten lässt, muss die Vorerfahrung der Versuchspersonen mit dem System, bzw. ähnlichen Systemen kontrolliert werden (s. Rautenberg 1991). Zur Erfassung der Benutzereigenschaften kann hierbei die Methode zur Ermittelung der Benutzercharaktere von Urbanek (1991) eingesetzt werden. Natürlich können auch Usabilty-Experten oder die Entwickler selbst für den Test herangezogen werden. Dabei ist jedoch immer zu beachten, dass keine Versuchsperson ein sichereres Testergebnis liefert als der tatsächliche Endbenutzer. Als verlässlich erweist sich ein Testergebnis, wenn die Versuchspersonen kein Vorwissen über das zu untersuchende System/Produkt haben, oder das Vorwissen zumindest systematisch kontrolliert werden kann.

Bestimmung des Benutzerprofils

In Übereinstimmung mit der Zielsetzung wird ein Benutzerprofil erstellt, anhand dessen Versuchspersonen (Vpn) akquiriert werden. Die Personen, die während des Tests die Rolle der Benutzer übernehmen, müssen die tatsächlichen, späteren Benutzer des Systems repräsentieren. Das Ziel des Tests kann durch die falsche Auswahl der Vpn verfälscht bzw. verfehlt werden. Das Profil der Vpn besteht aus einer Beschreibung der Charakteristiken der Benutzer, die das Produkt später benutzen werden. Solche Charakteristiken können Bildungsniveau, Ausdrucksfähigkeit (Muttersprache), Geschlecht, Alter, Beruf, Benutzungshäufigkeit oder Erfahrung im Umgang mit ähnlicher Software sein. Es ist daher sinnvoll die Zielgruppe in Anfänger und Experten zu unterteilen und für beide Gruppen verschiedene Maßstäbe und eventuell auch verschiedene Tests zu verwenden. Hat man keinen direkten Zugriff auf die Endbenutzer, so muss ein möglichst präzises Benutzerprofil erarbeitet werden. Hierbei ist man aber gezwungen, auf Marktstudien oder auf Studien von Konkurrenzprodukten zurückzugreifen.

---

[7] Je größer die Stichprobe ist, desto sensibler ist der statistische Test; man kann somit kleine Effekte messen. Rautenberg (1991) verweist zur Berechnung der genauen Stichprobengröße unter Annahme einer zu erwartenden Effektstörung auf Bortz (s. Bortz 1993).

Logistische Probleme bei der Auswahl der Versuchspersonen
Haben die Versuchspersonen eine längere Anreise zum Testort? Sind die Versuchspersonen Experten in ihrem Gebiet? Experten haben die negative Eigenschaft, dass sie selten und teuer sind. Lässt das Budget finanzielle Anreize zur Teilnahme zu?

4.2.3 Testaufgaben

Bezeichnend für den Usability-Test ist die korrekte Auswahl und Präsentation der im Test zu lösenden repräsentativen Aufgaben (s. Nielsen 1993:185). Die Auswahl sollte mit entsprechender Sorgfalt geschehen, da die Formulierung der Testaufgaben einen erheblichen Einfluss auf die Testresultate hat. Es lassen sich zwei Grundtypen von Aufgaben differenzieren:

- **Handlungsorientierte Aufgabe („process-based task“)**
  Diese beschreiben detailliert, welche Arbeitsschritte der Benutzer vollziehen muss, um das gewünschte Ergebnis zu erhalten. Bei diesem Aufgabentyp wird der Benutzer gezwungen, sich einer vorab definierten und strukturierten Vorgehensweise an die Lösung der Aufgaben heranzugehen. Dies ermöglicht erst den Erhalt vergleichbarer Daten verschiedener Benutzer. Regelrecht ungeeignet ist dieser Aufgabentyp um Informationen über Gedankengänge und mögliche, von den Entwicklern unbeachtete Lösungswege, zu sammeln.
  Beispiel für eine Aufgabenbeschreibung:
  „Laden Sie bitte das Textdokument mit folgendem Namen: ... und ergänzen Sie es um den folgenden Inhalt ...Versehen Sie dieses Textdokument mit allen für die Serienbrieferstellung notwendigen Steueranweisungen.“ (CI 1994)
- **Problemorientierte Aufgabe („results-bases task“)**
  Bei diesem Aufgabentyp wird dem Benutzer ein freies Ziel vorgegeben und er kann selbst entscheiden, welchen Weg er zum Erreichen des Ziels wählen will. Dies ist aber abhängig von Erfahrungen des Benutzers, da die Struktur zur Lösung der Aufgabe von ihm selbst erarbeitet werden muss. Je nach fachspezifischen Vorwissen der Versuchsperson über den Aufgabenkontext ist es sinnvoll, die Aufgabenbeschreibung unterschiedlich abzufassen. Bei hohem fachspezifischen Vorwissen ist die Beschreibung möglichst problemorientiert abzufassen, bei geringem bzw. keinem fachspezifischen Vorwissen ist die Beschreibung handlungsorientiert zu halten. Die handlungsorientierte Aufgabenbeschreibung soll verhindern, dass die beobachteten Benutzungsprobleme überwiegend aufgrund fehlenden Fachwissens zustande gekommen sind.
  Beispiel für eine Aufgabenbeschreibung:

> „Bitte erstellen Sie einen Brief mit folgendem Inhalt ... und bereiten sie ihn zum Eintüten und Versenden an folgenden Adressen ... vor. Bitte benutzen Sie als Briefvorlage das Dokument mit dem Namen ...“ (CI 1994)

Identifikation der Aufgaben

Die Erstellung der Arbeitsaufgaben erfolgt in vier Schritten:
Phase 1 (Aufgabenbeschreibung): Die Aufgabe muss kurz beschrieben werden, da sie den späteren Einsatz des Systems repräsentieren muss.
Phase 2 (Materialbeschreibung): Nachdem die Aufgabe beschrieben wurden ist, müssen die Voraussetzungen (z.B. bestimmte Konfigurationen) und benötigte Materialien für den Test aufgelistet werden.
Phase 3 (Erfolgsbeschreibung): Jede Aufgabe endet mit dem Erreichen eines bestimmten Zustandes, der in dieser Phase beschrieben werden muss. Anhand dieser Beschreibung kann der Erfolg bzw. Misserfolg der Arbeitsaufgabe bestimmt werden.
Phase 4 (Zeitbestimmungen): Jede Arbeitsaufgabe muss innerhalb eines definierten Zeitrahmens bearbeitet und beendet werden.

Erhebung von Szenarien

Ein Szenario ist eine episodische Beschreibung von Aufgaben und Tätigkeiten in ihrem Kontext (s. DATech 2001).

Aufgabenszenario

Der Versuchsleiter legt hier fest, welche Aspekte des technischen Systems beim Usability-Testing untersucht werden sollen. Diese Fragestellung endet in sogenannte Testszenarien. Jedes Szenario definiert die Aufgaben, die von der Versuchsperson gelöst werden sollen.
Dabei ist zu beachten, dass jede Aufgabenstellung für den Endnutzer sinnvoll und bedeutungsvoll ist. Außerdem sollten die Testszenarien in einer für die normale Arbeitsaufgabe logischen Reihenfolge dargeboten werden. Dabei steht die Erfüllung der beschriebenen Aufgabe im Vordergrund. In einem Vortest werden Testszenarien daraufhin überprüft, ob sie für die Versuchspersonen eindeutig und unmissverständlich sind.

Kontext-Szenario

Zur objektiven und validen Erhebung des Nutzungskontextes hat sich das Verfahren der Erhebung und Auswertung von Kontext-Szenarien bewährt, das in Anhang C.1 des DATech Prüfhandbuchs Gebrauchstauglichkeit (s. DATech 2001) näher beschrieben ist. Unter einem Kontext-Szenario verstehen DATech (2001) eine episodische Beschreibung von Aufgaben und Tätigkeiten in ihrem Kontext ohne Bezug zu konkreten Merkmalen eines Softwareprodukts. Damit enthält das Kontext-Szenario keine Information über die

tatsächliche Interaktion mit dem System. Hierbei ist die Darstellung der Aufgabenbearbeitung und nicht die Softwarenutzung bedeutsam. Bei einem Kontext-Szenario wird mittels 22 Fragen ein Überblick über die Arbeitstätigkeit, deren Voraussetzungen, normale Durchführung, eventuelle Besonderheiten bei der Durchführung sowie die organisatorischen Rahmenbedingungen erhoben. Die Anzahl der Benutzer, bei denen ein Kontextszenario zu erheben ist, hängt von der Homogenität der Arbeitsaufgabe und der Benutzergruppe ab. In der Praxis empfiehlt es sich, mit 2-3 Erhebungen zu beginnen und darauf zu achten, wann ein Sättigungseffekt entsteht, d.h. keine relevanten neuen Informationen mehr auftauchen. Häufig gibt es zwei Gründe zur Erhebung von Kontext-Szenarien:

- in der Vorbereitung eines Entwicklungsprojektes
- anlässlich einer Softwareprüfung.

Use-Szenarien
Die Use-Szenarien dienen der Erfassung der Interaktion des Benutzers mit der Software in der gegebenen Nutzungssituation am Bildschirmarbeitsplatz. Das Use-Szenario ist eine episodische Beschreibung der Interaktion mit der Software ohne dass für eine zu beschreibende Kernaufgabe eine vollständige Abdeckung aller möglichen Interaktionen angestrebt wird. Es sollte bereits ein Kontext-Szenario erhoben worden sein, damit die Kernaufgaben eines Arbeitsplatzes bekannt sind. Wurde kein Kontext-Szenario erhoben, so sollten die Kernaufgaben vorab mit dem Benutzer festegestellt werden (s. DATech, 2001:84). Use Szenarien können eventuell durch „screen-shots“ erläutert werden oder komplett aus einem Videofilm bestehen, um die Objektivität der Beschreibung kritischer Nutzungssituationen zu sichern. Es gibt zwei Situationen, in denen die Erfassung der Interaktion unterschiedlich sein kann (s. DATech 2001:77):

- in einem Softwareentwicklungsprozess (z.B. beim Prototyping)
- bei der Softwareprüfung.

4.2.4 Bestimmung der Leistungs- und Zufriedenheitsmetriken

In Abhängigkeit von den Arbeitsaufgaben müssen die Metriken der gesammelten Daten klassifiziert und quantifiziert werden. Da der Schwerpunkt auf Benutzerleistung und -zufriedenheit liegt, wird folgende Einteilung vorgenommen:

- Leistung (quantitative Daten): Die Zeit, die benötigt wurde, um die Aufgabe durchzuführen,
- Zufriedenheit (qualitative Daten):
  Die Benutzerzufriedenheit bei der Durchführung der Aufgabe

Diese Metriken können als Grundlage für die spätere Auswahl und den Einsatz der konkreten Testmethoden dienen.

4.2.5 Spezifikation der Testteilnehmer

Die folgenden Personen sind üblicherweise bei einem Usability-Test anwesend:

Versuchspersonen
Anhand der aufgestellten Benutzerprofile repräsentieren diese Personen die späteren Benutzer des Systems.

Testbeobachter
Neben den Versuchspersonen, gibt es noch Beobachter, die bei der Testdurchführung anwesend sind. Sie greifen in den Test nicht ein und je nach Testraum sind sie für die Versuchsperson nicht sichtbar.

Versuchsleiter
Laut Nielsen (1993:179ff.) sollten Versuchsleiter folgende Qualifikationen aufweisen:

- Erfahrung mit den eingesetzten Methoden
- Umfassende Kenntnisse des Produkts und seiner Oberfläche.

Der Versuchsleiter ist verantwortlich für die gesamte Durchführung des Tests. Dies reicht von der Vorbereitung der Testmaterialien, der Begrüßung und Einweisung der Versuchspersonen bis hin zur Auswertung der Testdaten und der Koordination anderer Beteiligter. Der Versuchsleiter verhält sich während der Aufgabenbearbeitung ruhig. Für ihn ist es wichtig, seinen natürlichen Impuls, der Versuchsperson in problematischen Situationen sofort zu helfen, „im Zaume zu halten“. Hier kann eine klare Absprache zwischen Versuchsleiter und Versuchsperson hilfreich sein. Nur wenn sich die Versuchsperson explizit an den Versuchsleiter wendet und um Hilfe nachfragt, wird der Versuchsleiter aktiv. Ein zu frühes Eingreifen des Versuchsleiters hindert die Versuchsperson, eine eigene Lösung zu finden.
Bei der Durchführung von Benchmark-Tests mit weitgehend EDV-Unerfahrenen ist es empfehlenswert, ihnen in scheinbar ausweglosen Situationen mit Hilfestellungen seitens des Versuchsleiters zur Seite zu stehen. Dies ist besonders dann wichtig, wenn man eine möglichst vollständige Aufgabenbearbeitung erreichen will. Bewährt hat sich hier, dass der Versuchsleiter gemäß dem fünffach gestuften Schema der Versuchsperson Hilfestellungen zukommen lässt. Rautenberg (1991) bezeichnet dieses Schema als „sokratischer Dialog“, weil der Versuchsleiter auf ein Hilfegesuch seitens der Versuchsperson nach dem „Frage Prinzip mit minimaler Information vorgeht“.

1. Stufe: Der Versuchsleiter weist die Versuchsperson auf das Handbuch und/oder das Hilfesystem hin. „Bitte sehen sie im Handbuch (Hilfesystem) nach!“.
2. Stufe: Der Versuchsleiter versucht die Aufmerksamkeit der Versuchsperson auf die Einweisung-/Instruktionsphase zu lenken. „Können Sie sich noch erinnern, was sie in der Einweisung/Instruktion an dieser Stelle getan haben?“.
3. Stufe: Der Versuchsleiter lenkt die Aufmerksamkeit auf den relevanten Suchbereich, indem er den Suchraum einschränkt. „Können Sie sich noch erinnern welche Funktionstaste (Menü, Icon, etc.) in Frage kommt?“.
4. Stufe: Der Versuchsleiter schränkt den Suchraum auf die konkrete Dialogoperation weiter ein. „Können Sie sich noch erinnern, was passieren würde, wenn Sie F... betätigen würden?“.
5. Stufe: Der Versuchsleiter weist die Versuchsperson direktiv an, die entsprechende Dialogoperation durchzuführen. „Bitte denken Sie jetzt an die Funktionstaste F...“.

Oft reicht es aus, wenn der Versuchsleiter bis zur dritten Stufe Hilfestellung gibt, weil dann viele Versuchspersonen ausreichend Informationen haben, um von selbst auf die Lösung des Problems zu kommen (erkennbar an Ausrufen wie „Ach ja“, etc.).

Entwickler
Es sei hier noch explizit darauf hingewiesen, dass Entwickler nicht Bestandteil des Testteams sind. Hierfür sprechen mehrere Gründe (s. Nielsen 1993:180f.):

- Sie sind zu sehr mit dem Design des Produktes beschäftigt, so dass sie nicht objektiv urteilen können.
- Sie sind versucht direkt einzugreifen, wenn Benutzer Probleme bei der Bearbeitung ihrer Aufgabe haben.

4.2.6 Bestimmung einer Testumgebung

Usability-Tests lassen sich sowohl als Feldversuche am Arbeitsplatz als auch im Labor (s. Abbildung 4) unter kontrollierten Bedingungen durchführen.

Usability-Labor
Durch die Kontrolle und Eliminierung von Störvariablen wird ein Vergleich zwischen den Testergebnissen der einzelnen Versuchspersonen erst möglich. Nachteilig wirken sich die sehr technischen Laborräume auf die Versuchsperson aus, die dadurch eingeschüchtert und in ihrem Verhalten beeinflusst werden kann. Durch eine Erklärung der Räumlichkeiten kann vorgebeugt werden,

dass die Versuchsperson sich im negativen Sinne beobachtet fühlt.
Unentbehrliche Komponenten des Usability-Labors sind:

- Testarbeitsplatz (entsprechend der Aufgabenstellung ausgerüstet) in einem (schallgeschützten) Versuchsraum.
- Kontrollraum für die Beobachtung der Versuchspersonen und Bedienung der Aufzeichnungsgeräte.
- Videokameras für die Aufzeichnung der zu testenden Abläufe und der Reaktionen der Versuchspersonen.

Felduntersuchungen
Die Untersuchung findet am Arbeitsplatz der Zielbenutzergruppe statt. Der Vorteil liegt in einer authentischen Arbeitssituation und -umgebung, in der positiv wirkende Einflüsse, wie Arbeitskollegen, vorhanden sind. Ebenfalls vorhandene Störvariablen bewirken, dass die Testergebnisse oft mehrere Erklärungsversuche für Usability-Probleme zulassen. Ein Feld-Usability-Test gibt auch wertvolle Informationen über das Arbeitsumfeld (z.B. Unterbrechungen, Umgebungsbedingungen).

Verhaltensbeobachtung
Die Verfahren der systematischen Verhaltensbeobachtung der Versuchspersonen während des Arbeitens mit dem System lassen sich untergliedern in abwesende und anwesende Beobachtung. Dabei wird versucht, möglichst alle subjektiven Einflüsse der Benutzer, wie z.B. Emotionen, Vorlieben und Stereotype weitgehend auszuschließen. Unterschieden wird hier zwischen:

- **Teilnehmende (anwesende) Beobachtung**
  Der Beobachter sitzt hinter oder neben der Versuchsperson und versucht, die beobachteten Handlungen, Fehler, Ausführungszeiten und andere wahrgenommene Attribute den Evaluationskriterien entsprechend zu beurteilen. Für die Beobachtung von Arbeitssituationen durch direkte Beobachtung/Wahrnehmung spielt die Habituation und Adaptation (Habituation als langfristige Gewöhnung an Bedingungen und Adaptation als kurzfristige Anpassung) an Umweltreize eine ausschlaggebende Rolle. Je mehr man mit bestimmten Arbeitsbedingungen vertraut ist, desto eher nimmt man diese als vertraut wahr und stellt sein Bezugssystem auf den Durchschnitt eigener Erfahrungen ein. Dabei kann die Beurteilung einer Arbeitssituation in Abhängigkeit von der Vorerfahrung der Experten unterschiedlich ausfallen. Die teilnehmende Beobachtung findet ihre Verwendung in einem Usability-Test, wenn der Beobachtungsablauf vom Beobachter nicht unterbrochen wird und die Beobachtung bei allen beobachteten Benutzern unter den gleichen Beobachtungsbedingungen stattfindet. In der Regel sind es äußere An-

lässe, die eine Untersuchung dieser Art nahe legen, z.B. aufgetretene Benutzungsprobleme, oder mangelnde Zufriedenstellungen der Benutzer. Manchmal wird die Methode der teilnehmenden Beobachtung als Beobachtungsinterview bezeichnet, wenn der Schwerpunkt der Untersuchung auf der Befragung des Benutzers liegt. Die methodische Vorgehensweise wechselt oft zwischen Beobachtung und Interview.

- **Nicht-teilnehmende oder indirekte Beobachtung**
  Bei dieser Technik wird der Benutzer *indirekt* beobachtet. Dies kann durch eine Videoaufzeichnung oder logfile recording geschehen. Bei der Videoaufzeichnung werden oft mehrere zeitlich parallele Kameraeinstellungen (Hand, Bildschirm, Gesicht, Körper) zusammen mit synchronisierten logfiles für die Beobachtung eingesetzt (s. Frieling & Sonntag 1999:91ff.).

Beobachtungsfehler
Dorsch (1994) versteht hierunter Fehler, die bei der Beobachtung auf Grund der Leistungsgrenzen des Beobachters vorkommen (Aufmerksamkeitsschwankungen, Ermüdung). Wie aber die Beurteilung kaum von der Beobachtung zu trennen ist, werden auch meist die Beurteilungsfehler zu den Beobachtungsfehlern gerechnet. Wichtige Beurteilungsfehler sind (s. Bortz & Döring 2002:182 ff.):

- Halo-Effekt (s. Zimbardo 1995:527).
- Zentrale Tendenz, d.h. die Tendenz des Beobachters, extreme Urteile zu vermeiden.
- zu frühe Interaktion und Wertung.

Die Häufigkeit von Fehlern kann durch Beobachterschulung und Verwendung von technischen Hilfsmitteln wie z.B. Filmkamera vermindert werden. Zur Konstruktion von Beobachtungsinstrumenten sei auf Schnell et al. (1989:359ff.) verwiesen. Weiterführende Informationen zur Durchführung einer Beobachtungsstudie geben Bortz und Döring (2002:269ff.).

4.2.7 Aufbereitung der Daten

Die Beobachtungsdaten (Video, Tonband, logfile) aller Versuchspersonen müssen hinsichtlich der design-relevanten Information ausgewertet werden, Dazu empfiehlt es sich, auf der Grundlage der durchgeführten Beobachtungen ein Auswertungsraster aufzustellen und dieses Raster systematisch auf die Beobachtungsdaten aller Versuchspersonen anzuwenden. Beobachtungsverfahren werden bei der Evaluation von Mensch-Maschine-Schnittstellen überwiegend eingesetzt, um aus dem beobachteten Verhalten Rückschlüsse auf kognitive Abläufe beim Benutzer zu ziehen. Speziell durch die Beobachtung

der Benutzerreaktionen in Fehlersituationen und bei explorativem Verhalten gibt Aufschluss darüber, wie kompetenzförderlich und flexibel ein Produkt gestaltet ist. Bei der Planung einer Benutzerobservation ist es wichtig, bereits vor der Untersuchung ein Klassifikationsschema zu entwickeln, welches die interessanten Verhaltenskategorien festlegt. Zu Testzwecken empfiehlt es sich, vorab einige Personen zu beobachten und aus diesen Erfahrungen ein Klassifikationsschema zu entwickeln.

4.2.8 Durchführung eines Pilot-Tests

Laut Nielsen (1993:174f.) sollte kein Usability-Test durchgeführt werden, ohne den Testablauf an verschiedenen Testdurchläufen zu prüfen. Hierbei könnten die folgenden Faktoren des Tests überprüft werden:

- Funktionieren der Testrequirements,
- Verständlichkeit der Instruktion und Aufgabenbeschreibungen,
- Verhältnis von Aufgaben und Zeitbeschreibungen,
- Aussagekraft der aufgestellten Metriken.

4.2.9 Durchführung des Tests

Im Umgang mit den Versuchspersonen sind die nachfolgenden Punkte einzuhalten:

- Begrüßung der Versuchsperson und mitteilen von Informationen über die Umgebung (Toiletten, Ausgänge, etc).
- Veranschaulichung der Testumgebung.
- Mit möglichst allgemein verständlichen Worten wird der Versuchsperson Ziel und Zweck des Tests erklärt. Es ist besonders wichtig, der Versuchsperson verständlich zu machen, dass das System und nicht die Versuchsperson getestet werden soll. Erläuterung der eingesetzten Methoden (z.B. Thinking Aloud, Videoaufzeichnung).
- Kurze Einführung in die eigentliche Aufgabe.
- Rückfragen der Benutzer ermitteln und beantworten.
- Beginn der eigentlichen Aufgabe.
- Nachfragen nach jeder bearbeiteten Aufgabe, ob spezielle Probleme aufgetreten sind.
- Nach Beendigung aller Aufgaben Ermittlung der Benutzerkommentare.
- Dank und Überreichung eines kleinen Geschenks.

4.2.10 Ethische Richtlinien und Informationen zum Test

Die Versuchsperson sollte unbedingt über Dauer und Zweck des Tests in Kenntnis gesetzt werden. Sie sollte durch die Einführung in der Lage sein, die allgemeine Struktur und Ablauf des Tests zu verstehen. Ein wichtiger und entspannender Hinweis ist, dass ihr Verhalten nur hinsichtlich der Produkteigenschaften interpretiert wird und nicht zu Rückschlüssen über ihre Fähigkeiten dient. Sinn und Zweck der Untersuchung ist es, die Software zu evaluieren und nicht den Nutzer. Die Teilnahme an dem Test ist freiwillig, und es ist für die Versuchsperson möglich, die Untersuchung zu jedem Zeitpunkt zu beenden (s. Zimbardo 1995:21f.). Als Faustregel gilt: *Der Versuchsperson sollte es nach dem Test nicht schlechter gehen als zuvor*.

Rechtliche Gesichtspunkte
Die Versuchsperson sollte ein Formular unterschreiben, dass er an der Untersuchung freiwillig teilnimmt. Falls Video- oder Audioaufzeichnungen gemacht werden, muss eine Berechtigung zur Weiterverwendung unterschrieben werden. Falls Informationen über das zu entwickelnde Produkt aus marktstrategischen Gründen nicht nach außen gelangen dürfen, sollte die Versuchsperson zu Stillschweigen verpflichtet werden.

Training
Je nach Zielsetzung des Usability-Tests kann es hilfreich sein, die Versuchspersonen im Umgang mit dem System oder Systemkomponenten zu schulen. Die Einführung kann von einer kurzen Erläuterung bis hin zu einem mehrtägigen Seminar reichen, je nach geforderter Qualifikation, die die Versuchsperson bei Beginn des Tests vorweisen sollen. Die Gründe für eine Schulungsmaßnahme könnten sein:

- Der Benutzer hat ein Mindestmaß an systemspezifischen Wissen vorzuweisen, um überhaupt mit dem zu testenden System arbeiten zu können.
- Der Usability-Test soll nur mit erfahrenen Benutzern durchgeführt werden, die aber durch die Neuartigkeit des Produktes nicht vorhanden sind.
- Es sollen ausschließlich komplexe Funktionen des Systems durch erfahrene Benutzer getestet werden.

Testmaterial und Einrichtung
Es ist eine Selbstverständlichkeit, dass zu Testbeginn alle Daten vorliegen müssen. Nichts ist unangenehmer als *missing data* zu riskieren, weil ein Testexperiment nicht sorgfältig vorbereitet worden ist. Daher sollte für jeden Test

eine Liste mit allen nötigen Vorarbeiten, Fragebögen, Formularen usw. angefertigt werden.

4.2.11 Abschlussbefragung

Die Versuchsperson wird häufig nach dem Test angehalten, Fragebögen in Bezug auf subjektive Messgrößen (Zufriedenheit, erster Eindruck etc.) auszufüllen. Es sollte der Versuchsperson auch immer die Möglichkeit gewährt werden, eigene Kommentare und Verbesserungsvorschläge zu geben. Der Versuchsleiter kann die Besprechung auch dazu nutzen, sich von der Versuchsperson gewisse Situationen, die während des Tests aufgetreten sind, erläutern zu lassen (s. Nielsen 1993:191).

4.2.12 Auswertung der Testdaten

Die im Test ermittelten und dokumentierten Daten werden nach quantitativen und qualitativen Fragestellungen analysiert, wie etwa:

- Wie lange dauerte die Erfüllung einer Aufgabe?
- Bei welchen Aufgaben traten Schwierigkeiten auf?
- Welche Schwierigkeiten gab es?
- Auf welche Fehlkonzeptionen (des Benutzers vom System bzw. des Systems vom Benutzer) lassen sich diese Probleme zurückführen?
- Wie häufig treten Fehler auf?
- Wurden die festgelegten Usability-Ziele erreicht?
- Welche Konsequenzen hatten manche Fehler für die Aufgabenerfüllung? (s. Honold, 2000:96).

Eine Beschreibung der aufgetretenen Usability-Probleme, ihre Gewichtung, und ihre Ursachen sind die Grundlage für die Überarbeitung des getesteten Systems.

Nach Beendigung eines Usability-Tests werden die Testdaten sowie die Eindrücke des Versuchsleiters und sonstigen Beobachter des Tests miteinander verglichen und in Beziehung gesetzt, um Ergebnisse formulieren zu können. Die Testauswertung erfolgt hierbei in zwei Schritten:

- Zuerst werden die sogenannten „hotspots“, d.h. die offensichtlichen Usability-Schwächen analysiert, so dass die Entwickler hier die bereits Schwachstellen entfernen können.
- Im zweiten Schritt werden die bisher gewonnenen Erkenntnisse eingehender analysiert, sowie alle restlichen Usability-Schwachpunkte untersucht, die nach vollständiger Auswertung der Testdaten erfasst werden

konnten. Dieser Auswertungsprozess unterteilt sich weiterhin in vier Teilschritte:

- Zusammenfassung der Testdaten,
- Analyse der Testdaten,
- Formulierung von Empfehlungen zur Verbesserung der Benutzerfreundlichkeit,
- Erstellung des abschließenden Testberichts.

4.2.13 Fehler und Fallen beim Testen („pitfalls“)

Die Nichtbeachtung folgender Punkte kann beim Usability-Testing zu Verfälschungen der Ergebnisse führen:

- Die Testaufgabe ist dem Aufgabenkontext des potentiellen Benutzerkreises zu entnehmen.
- Der Test sollte im natürlichen Arbeitskontext der Benutzer erfolgen.
- Die Auswahl der Benutzer sollte repräsentativ sein.
- Der Test ist mit mindestens sechs verschiedenen Benutzern durchzuführen, um die Auswertung mit inferenzstatistischen Methoden zu ermöglichen, damit eine möglichst gute Generalisierbarkeit gewährleistet wird (s. Bortz 1993).

4.3 Auswahl der Testmethoden

Bei der Auswahl sind verschiedene Aspekte zu bedenken:

- Welches *Ziel* verfolgt die Untersuchung?
- Erfahrung der Prüfer.
- Zeitlicher, finanzieller und personeller Aufwand der Methoden.
- Ist es wichtig, dass die Testumgebung der Arbeitsumgebung des Benutzers entspricht?

4.4 Usability-Test-Methoden

Im folgenden Abschnitt werden die Methoden beschrieben, die bei Usability-Tests häufig Anwendung finden. Usability-Tests können mit sehr unterschiedlichen, durchaus dem jeweiligen Entwicklungsstand angepassten Methoden durchgeführt werden. Die folgende Liste erhebt nicht den Anspruch der Vollständigkeit, bietet aber Orientierung schon bei ganz frühen Phasen bis hin zur Einführung des Produkts.

Eine Software-Entwicklungs-Strategie ist das Prototyping, dessen wesentliche Bestimmungsstücke eine frühzeitige Benutzerbeteiligung, die empirische Be-

wertung der Bedienungsfreundlichkeit des Programms durch die Benutzer sowie ein iteratives Design sind (s. Frieling & Sonntag 1999:513).

4.4.1 Rapid Prototyping

Entwickler und Benutzer skizzieren auf Papier (nicht auf Rechnern, das begünstigt die Entwickler) Bildschirmformulare für typische Arbeitsaufgaben.

4.4.2 Prototypenentwicklung

Entwickler führen den Benutzern Bildschirmentwürfe vor. Benutzer kommentieren die Entwürfe direkt. Die Ausführungen werden aufgezeichnet oder protokolliert.

4.4.3 Papier und Bleistift Simulation

Entwickler legen Benutzern Sätze von ausgedruckten Bildschirmformularen vor. Benutzer arbeiten an einer typischen Sachaufgabe, tragen Ergebnisse in Felder ein und skizzieren durch Linien ein typischerweise mit der Maus zurückzulegenden Strecke. Viele Zacken in den Linienzügen lassen auf eine nichtangepasste Feldaufteilung schließen, häufige Übergänge zu anderen Formularen zeigen Probleme des Maskenzuschnittes auf.

4.4.4 Benutzungsorientierte Benchmark-Tests (Leistungsmessung)

Diese Methode umfasst induktive und deduktive benutzungsorientierte Benchmark-Tests. Induktive benutzungsorientierte Benchmark-Tests sind bei der Evaluation eines (z.B. vertikalen) Prototypen, oder einer (Vor-) Version zur Gewinnung von Gestaltungs- und Verbesserungsvorschlägen, bzw. zur Analyse von Schwachstellen in der Benutzbarkeit einsetzbar. Diese Verfahren kommen dann zum Einsatz, wenn nur *ein* Prototyp bzw. nur eine Version der zu testenden Software vorliegt. Dagegen haben deduktive Benchmark-Tests das Ziel, zwischen mehreren Alternativen (mindestens zwei Prototypen bzw. Versionen) zu entscheiden. Außerdem lassen sich hier noch Gestaltungs- und Verbesserungsvorschläge gewinnen. Bei Benchmark-Tests wird eine Programmentwicklung dadurch überprüft, dass mehrere Benutzer unter reproduzierbaren Umfeldbedingungen typische Aufgaben erledigen. Mit objektivierbaren Methoden wie Zeitmessungen, Interaktionsverfolgung und Fehleranalysen lassen sich Hinweise auf Ergonomiedefizite gewinnen. Dabei ist die Überprüfung unterschiedlicher Programmvariationen sehr sinnvoll („Welches

Modul zeigt die größten Fehlerhäufigkeiten?“). Die Benutzerbeteiligung ist sinnvoll, weil bestimmte Eigenschaften interaktiver Systeme nur in der konkreten Interaktion messbar sind.

Es existieren verschiedene Methoden, um die Ziele des Usability-Tests zu überprüfen:

- Thinking Aloud
- Gruppendiskussion und -gespräch
- Constructive Interaction
- Retrospective Testing

4.4.5 Thinking Aloud

Dieses Verfahren findet häufig bei der formativen Evaluation seine Verwendung (s. Nielsen 1993:224). Zwischen 3 bis 5 Nutzer werden gebeten, während einer Aufgabenbewältigung ihre Überlegungen, Probleme und ihre jeweiligen Handlungsalternativen *laut* vor sich her zu sagen. Diese Technik kann auch ohne konkrete Aufgabenstellung angewandt werden. Hierbei exploriert der Benutzer das System in eigener Initiative. Diese „freie Exploration“ ist besonders dann hilfreich, wenn ein genereller Eindruck vom System ermittelt werden soll (s. Lin et al. 1997; Nielsen 1993:195-198). Bei Thinking-Aloud-Untersuchungen ist es oft nötig, den Teilnehmer durch kurze Hinweise wie „Können Sie uns sagen, was sie gerade denken?“ oder „Haben sie dieses Verhalten erwartet?“ zum lauten Denken zu animieren

Vorteile der Thinking Aloud Technik
Es wird sofort ersichtlich, an welchen Stellen der Nutzer das im System verfolgte Konzept zur Benutzungsschnittstelle falsch interpretiert und warum diese Fehlinterpretation aufgetreten ist.
Leichter und kostengünstiger Einsatz ohne aufwendige Testapparaturen zu jedem Zeitpunkt des Entwicklungsprozesses.

Nachteile der Thinking Aloud Technik
Lin et al. (1997) betonen, das die Methode von vielen Nutzern als ungewohnt und verwirrend empfunden wird. Die Benutzer haben erhebliche Schwierigkeiten, ihre Gedanken fortlaufend zu verbalisieren. Fälschlicherweise glauben die Benutzer bestimmte Erwartungen erfüllen zu müssen, d.h. das „Problem der sozialen Erwünschtheit“ kann deren Äußerungen verfälschen (s. Bortz & Döring 2002:233-236).

Hackman und Biers (1992) untersuchten die Methode des lauten Denkens als Einzel und Gruppenmethode. Die Ergebnisse der Untersuchung zeigten, dass

Evaluatoren im Zweier-Team insgesamt mehr Zeit mit dem Verbalisieren als einzelne Evaluatoren verbrachten. Im Team wurde mehr Zeit dazu verwendet, qualitativ hochwertige Verbalisierungen vorzunehmen. Die Autoren halten für wichtig, dass zwei Personen mehr Zeit mit dem Verbalisieren verbringen und dabei auch mehr hochwertige Informationen als eine Person generieren. Sie bewerten die Ergebnisse dahingehend, dass die Teammethode im gleichen Zeitraum wie die Einzelmethode mehr Informationen liefert und daher zeiteffizienter ist. Ferner verweisen sie darauf, dass im Team andere Anmerkungsqualitäten als bei der Einzelevaluation entstehen. Sie zählen dazu besonders auch Anmerkungen, die Unsicherheit mit dem Interface und dessen Bedienung ausdrücken.

4.4.6 Gruppendiskussion und -gespräch

Gruppendiskussionen können mit repräsentativen Nutzern und/oder Entwicklern und Designern durchgeführt werden. In Group Design Reviews evaluieren mehrere Nutzer gemeinsam in der Gruppe eine Benutzungsschnittstelle. Hierbei wird vorausgesetzt, dass mindestens ein Human-Factor-Experte anwesend ist. Weitere Mitglieder können beispielsweise Dokumentations- und Trainingsspezialisten, Marketingfachleute, User Interface Designer und auch Nutzer sein. Das Ziel dieser Art der Evaluation besteht darin, mögliche Usability-Probleme des untersuchten Softwaresystems aufzudecken. Nach allgemeiner Auffassung ermuntert die Gruppendiskussion gehemmte Teilnehmer durch die sichtbare Auskunftsbereitschaft anderer Gesprächspartner. Andererseits hat die Kleingruppenforschung gezeigt, dass Gruppen bei bestimmten Aufgaben einem Einzelnen weit überlegen sein können. Es handelt sich dabei hauptsächlich um Aufgaben „vom Typ des Suchens“.

4.4.7 Constructive Interaction

Dies ist eine Modifikation der Thinking Aloud Methode. Hier erforschen zwei Versuchspersonen gemeinsam ein System. Dabei soll die Kommunikation zwischen beiden die Thinking Aloud Methode ersetzen (s. Nielsen 1993:198).

Vor- und Nachteile der Constructive Interaction
Die Hemmschwelle mit einem zweiten Teilnehmer (im günstigsten Fall einem Arbeitskollegen) zu sprechen, ist deutlich geringer, als seine Gedanken für sich selbst zu formulieren. Die Aufzeichnungen einer solchen Sitzung könnten irreführend und möglicherweise unbrauchbar sein, wenn die kooperierenden Teilnehmer verschiedene Methoden des Lernens und Problemlösens verfolgen.

4.4.8 Retrospective Testing

Retrospective Testing ist ebenfalls eine Variante des Thinking Aloud Tests, bei der Videoaufzeichnungen ihren Einsatz finden. Das Verfahren ist im Labor oder am Arbeitsplatz einsetzbar. Die Benutzer führen am Rechner eine definierte Prüfaufgabe mit dem zu bewertenden Softwareprodukt durch. Dabei werden sämtliche Interaktionen aufgezeichnet und im Anschluss den Benutzern vorgeführt und von ihnen kommentiert (s Nielsen 1993:199). Durch die Digitalisierung der Videodaten besteht die Gelegenheit, einzelne Elemente aus dem Video herauszulösen, um diese den Probanden vorzulegen. Somit kann beispielsweise ein Standbild ausgedruckt oder ein Geräusch herausgeschnitten werden, dass bei einem zweiten Gespräch von der befragten Person kommentiert wird (s. Irion 2002). Retrospective Testing ist besonders sinnvoll, wenn eine repräsentative Stichprobe nicht verfügbar ist.

Tabelle 4 enthält einen Überblick über die Methoden, die Anzahl der Benutzer sowie die Vor- und Nachteile der einzelnen Methoden (Nielsen 1993:224):

Tabelle 4: Übersicht der Testmethoden (modifiziert nach Nielsen 1993:224)

Methode	Anzahl der Benutzer	Vorteile	Nachteile
Thinking Aloud	ca. 3-5	Zeigt Missverständnisse der Benutzer, Sehr günstiger Test.	Unnatürlich für die meisten Benutzer.
Interviews	ca. 5	flexibel	Zeitaufwendig: Aufwendige Analyse und schlechte Vergleichsmöglichkeiten
Fragebogen	> 30	Subjektive Benutzerwünsche Leicht wiederholbar	Pilot-Tests notwendig
Logging Actual Use	> 20	Zeigt wenig/häufig benutzte Funktionen Läuft kontinuierlich	Großes Datenaufkommen, Verletzung der Privatsphäre

4.5 Methoden der Datenerfassung

Hier werden die Methoden zur Datenerfassung beschrieben, welche am häufigsten zum Einsatz kommen:

- Befragungsmethoden,
- Logging Actual Use,
- Incident Diaries,
- Videoaufzeichnung.

4.5.1 Befragungsmethoden

Nach Hasebrook (1995:254) muss das erste Kriterium eines Evaluationsprozesses die Akzeptanz einer Software bei einem realistischen Einsatz in der Zielgruppe sein. Dabei ist darauf zu achten, dass Akzeptanzdaten bei einer ausreichend großen Stichprobe und mit geeigneten Fragebogen- und Interviewmethoden erhoben werden. Akzeptanzdaten sind sehr wichtig, reichen aber zur Beurteilung eines Systems und seines Einsatzes nicht aus. Befragungsmethoden sind unverzichtbar, wenn es darum geht, Auskunft über die subjektive Zufriedenheit der Benutzer zu erhalten und die zu erwartende Akzeptanz gegenüber einem Softwareprodukt abzuschätzen. Mit Hilfe dieses Verfahrens lassen sich auch entscheidende Schwachstellen und Defizite eines Produktes lokalisieren, es gibt aber wenig Hinweise auf detaillierte Designfragestellungen. Weil sich Experten über die Effektivität und Effizienz eines Produktes irren können, sind stets die Benutzer nach ihrem subjektiven Eindruck über Effektivität und Effizienz zu befragen, ein Urteil das als Zufriedenstellung interpretiert wird. Bei einer Benutzerbefragung ist auf die folgenden Gütekriterien zu achten:

- Sind die befragten Benutzer repräsentativ?
- Haben die Fragen einen definierten Aufgabenbezug?
- Sind relevante Merkmale des Nutzungskontextes beschrieben?
- Ist der Übungsgrad der Benutzer festgestellt?
- Ist die bisherige Nutzungsdauer festgestellt?
- Ist eine Wiederholungsmessung nach der Einarbeitungsphase geplant (Falls der Benutzer sich gerade einarbeitet)?

Zu den Befragungsmethoden zählen die schriftliche (Fragebögen) und die mündliche Befragung (Interview). Aus der Usability-Perspektive zählen Fragebögen und Interviews zu den indirekten Methoden. Fragebögen und Interviews erheben die Meinung und Einstellung des Benutzers bezüglich der Benutzungsschnittstelle. Der wesentliche Unterschied in der Interview- und Fragebogenmethodik liegt in der Standardisierung der Fragen, der Durchführung

und der Auswertung. Die Wahl zwischen Fragebogen und Interview hängt vor allem von der Zielsetzung der Evaluation ab. Sollen viele Benutzer angesprochen werden, ist ein Fragebogen das ökonomischere Instrument. Die Qualität der Antworten hängt stark vom Aufbau eines Fragebogens oder Interview ab. Für weitere grundlegende Fragen wird auf Bortz und Döring (2002) verwiesen.

4.5.1.1 Fragebögen

Die Evaluation anhand von Fragebögen ist eine gute Technik, wenn eine große Anzahl von Probanden herangezogen werden soll. Die Erfassung von Daten über einen Fragebogen bietet die Gelegenheit, Meinungen und Einstellungen zu einem bestimmten Bereich des Systems zu erfragen. Durch Fragebogen lassen sich leicht quantifizierbare Aussagen über Häufigkeiten, Bewertungen und Vorlieben gewinnen. Der Fragebogen solle nicht zu lang sein und mögliche Antwortalternativen vorgeben. Die Fragen sollte auch sehr präzise formuliert sein, da ein Nachfragen von Seiten der Benutzer oft nicht möglich ist. Durch Antwortvorgaben können die Ergebnisse leichter miteinander verglichen werden. Um Verbesserungsvorschläge aufnehmen zu können, sollte ausreichend Platz vorhanden sein und die Benutzer im Vorfeld aufgefordert werden, frei über ihre Eindrücke zu schreiben. Man muss aber auch damit rechnen, dass der Benutzer die Lust verliert und keine ehrlichen und konstruktiven Antworten mehr gibt.

An erster Stelle werden Daten zur subjektiven Einschätzung erfasst, die für eine spätere Analyse kategorisiert und zusammengefasst werden. Hierfür sind standardisierte Fragebögen, wie „Questionnaire for User Interaction Satisfaction“ (QUIS[8]) oder „Software Usability Measurement Inventory“ (SUMI[9]) vorzuziehen, da bereits Erfahrungen mit der Validität vorliegen. Spezifischere Fragebögen leisten aber für den konkreten Fall zumeist mehr. Fragebögen sind aber besonders stark vom Erinnerungsvermögen, Aufmerksamkeit, Selbsterkenntnis etc. abhängig und sowohl für unwillkürliche Fehler und Verzerrungen als auch für absichtliche Verfälschungen anfällig. Ein Vorteil von Fragebogenmethoden besteht darin, dass sie mit geringeren finanziellen und organisatorischen Aufwand eingesetzt werden können. Bei geschlossenen Fragen wird vom Befragten eine Bewertung auf einer mehrstufigen Skala erwartet. Hier sollten Skalen mit nur drei, fünf oder sieben Stufen eingesetzt

---

[8] siehe URL: http://www.ce.uni-linz.ac.at/research/sw_erg_pages/quis.htm

[9] siehe URL: http://www.ucc.ie/hfrg/questionnaires/sumi/index.html

werden. Was darüber hinaus geht, kann vom Befragten kognitiv nicht mehr richtig differenziert werden.

Die Bewertung von Häufigkeiten:

nie	selten	gelegentlich	oft	immer
0	1	2	3	4

Die Beschreibung von Wahrscheinlichkeiten:

keinesfalls	wahrscheinlich nicht	vielleicht	ziemlich wahrscheinlich	ganz sicher
-2	-1	0	+1	+2

Die Abgabe einer Bewertung

völlig falsch	ziemlich falsch	unentschieden	ziemlich richtig	völlig richtig
-2	-1	0	+1	+2

oder

trifft gar nicht zu	trifft wenig zu	teils, teils	trifft ziemlich zu	trifft völlig zu
-2	-1	0	+1	+2

Background Questionnaire
Hiermit wird der Wissenshintergrund des Benutzers erfragt, um sein Verhalten bei der späteren Analyse besser interpretieren zu können. Mit Hilfe des Fragebogens kann geprüft werden, ob der zum Test erschienene Benutzer den Anforderungen des Benutzerprofils auch wirklich entspricht.

Questionnaire for User Interaction Satisfaction (QUIS)
QUIS, von Norman und Shneiderman (1989; zit. nach Lin et al. 1997) entwickelt, stellt ein zuverlässiges und konsistentes Instrument zur Messung der Zufriedenheit bei Softwareanwendungen dar und ist in dieser ursprünglichen Form nicht für die Web-Technology entwickelt worden. Die Kurzform des QUIS umfasst 5 Kategorien mit je 46 Items,. die einzelnen Items auf einer Skala von 0-9 eingestuft. QUIS umfasst Fragen an die Benutzer wie demographischer Hintergrund, Terminologie und Systeminformation (Systemstatus, Anweisungen, Fehlermeldungen etc), Erlernbarkeit des verwendeten Systems und Systemfähigkeiten (Schnelligkeit, Zuverlässigkeit usw.). Der Benutzer gibt den Grad seiner Zufriedenheit in einer Skala an, die von 1 bis 7 reicht, wobei 1 als absolut unzufrieden und 7 als absolut zufrieden gewichtet wird. Nach jedem Abschnitt gibt es noch Platz für Kommentare. QUIS kann

aber auch für Web-Site-Untersuchungen als geeignetes Instrument für die Zufriedenheitsmessung des Nutzers verwendet werden, indem einige irrelevante Fragen durch Fragen, die sich auf Hypermedia-Anwendungen beziehen, ersetzt werden. Wichtig dabei ist, die Fragen möglichst neutral zu formulieren, um nicht durch die Art der Fragestellung den Benutzer diesbezüglich zu beeinflussen.

Adjektivskalen zur Einschätzung der Stimmung (SES)
Die Adjektivskalen zur Einschätzung der Stimmung sind eine systematische Weiterentwicklung einer von Saretz (1969) konzipierten Skala zur Selbsteinschätzung der augenblicklichen Stimmungslage. In der Instruktion des SES wird der Proband aufgefordert, mit Hilfe von Eigenschaftswörtern seine augenblickliche Stimmungslage bzw. sein augenblickliches Befinden selbst einzuschätzen (Hampel 1977: 46).

Die augenblickliche Stimmungsskalierung weist einen größeren Vorteil für Wiederholungsuntersuchungen in kürzeren Zeitabständen bzw. für Messungen der Stimmungsänderungen auf. Die folgenden drei adjektivische Skalenformen zur Einschätzung der augenblicklichen Stimmungsalge liegen vor: Die SES-Langform (SES-L) umfasst 84 Items, welche durch folgende Dimensionen (Faktoren) dargestellt werden:

1.Gehobene Stimmung, 2. Gedrückte Stimmung, 3. Missstimmung, 4. Ausgeglichene Stimmung, 5. Trägheit (Deaktiviertheit)[10], 6.Müdigkeit.

Des weiteren werden sechs Items als Pufferitems verwendet, so dass die SES-Langform insgesamt 90 Items umfasst Neben jedem Stimmungsitem ist eine siebenstufige unipolare Intensitätsskala mit ihren verbalen Kennzeichnungen angeordnet:

1	2	3	4	5	6	7
überhaupt nicht zureffend	ein bisschen zutreffend	etwas zutreffend	ziemlich zutreffend	überwiegend zutreffend	fast völlig zutreffend	vollkommen zutreffend

Abbildung 6: siebenstufige Intensitätsskala (Hampel 1977:60)

[10] Für manche Anwendungsbereiche mag es nach Hampel (1977:48) sinnvoll sein, die Items des Faktors „Trägheit“ umzupolen und ihn dann als Faktor der „Aktiviertheit“ zu beschreiben.

Beim SES gibt der Proband pro Stimmungsitem eine Gewichtszahl von 1 bis 7 an, wobei 1 den geringsten und 7 den stärksten Ausprägungsgrad des adjektivischen Stimmungsindex charakterisiert. Durch Summation der Gewichtszahlen aller Items einer Skala bekommt man die individuellen Skalenwert (Hampel 1977:53).
Der SES A[11] und SES B als äquivalente (parallele) Halbformen bestehen jeweils aus 42 Items.

Measuring Usability of Systems in Context (MUSiC)
MUSiC ist ein Verfahren zur Erfassung der Usability bzw. „Quality in Use" von Software in Anlehnung an die Norm EN ISO 9241-11, die im Rahmen des ESPRIT Projektes entwickelt wurde. Zu diesem Zweck wird zunächst der Nutzungskontext systematisch analysiert, um anschließend anhand von Tätigkeitsmessungen in einem repräsentativen Kontext die Effektivität und Effizienz, mit der die Benutzer ihre Aufgaben erledigen, zu untersuchen (s. Bevan & Macleod 1994; Bevan 1997). Die Indikatoren für Effektivität und Effizienz sind der Grad der Vollständigkeit, mit dem bestimmte Aufgaben bearbeitet werden und die Zeit, die zur Aufgabenbearbeitung benötigt wird. Zur Bewertung der Tätigkeiten (performance) wird eine Benutzungssituation mit einem Videorecorder und logfile recording aufgenommen und mit Hilfe der DRUM-Software (Diagnostic Recorder for Usability Measurement), einem umfassenden, objektiven Evaluationsverfahren, analysiert (s. Bevan & Curson 1997). DRUM ermöglicht unter anderem eine Differenzierung zwischen produktiv verbrachter Zeit und der Zeit, die dafür aufgewendet wird, um z.B. Benutzungsfehler zu korrigieren oder Hilfe zu suchen. Ferner wird die mentale Beanspruchung bei der Bearbeitung der mit der Software zu verrichtenden Aufgaben erfasst. In diesem Rahmen können objektive Messungen (zum Beispiel Herzrate) und subjektive Messungen (persönlich erlebte Beanspruchung) vorgenommen werden. Außerdem wird die subjektive Bewertung der Software durch den SUMI-Benutzerfragebogen (Software Usability Measurement Inventory) erhoben.

Software Usability Measurement Inventory (SUMI)
Die subjektive Zufriedenheit von Software wird durch den Benutzerfragebogen SUMI erhoben. Dieses Verfahren dient Benutzern zur selbständigen Beurteilung von Software. SUMI ist ein Verfahren zur Erhebung subjektiver

[11] Hampel (1977: 53) berichtet, dass SES–A und SES-B bisher nur mit 5 Intensitätsstufen (ganz- überwiegend – ziemlich – etwas - gar nicht) eingesetzt wurden.

Bewertungen (s. Bevan 1997). Der Anspruch, ein Inventar zur Messung von Software-Usability zu sein, kann aber nicht aufrechterhalten werden.

Der Fragebogen umfasst 50 Items, die sich den folgenden 5 Skalen zuordnen lassen:

- Affect (the user's general emotional reaction to the software);
- Control (the extent to which the users feel in control of the software);
- Efficiency (the degree to which users feel that the software assists them in their work;
- Helpfullness (the degree to which the software is self-explanatory, adequacy of documentation);
- Lernability (the ease with which the users feel that they have been able to master the system) (s. Kirakowski & Corbett 1993; Natt och Dach Regenell, Madsen & Aurum 2001).

Für einen effektiven Einsatz des Fragebogens schlagen Kirakowski und Corbett (1993) 10 Benutzer vor. Diese sollten bereits über Erfahrungen mit der zu evaluierenden Software verfügen. SUMI bietet keine besonderen Methoden zur Beschreibung des Kontextes einer Evaluation an. Das Verfahren bezieht sich ausdrücklich auf die Bewertung des Gesamtsystems. Es ist voll standardisiert und der Prüfungs- und Auswertungsaufwand ist als gering zu erachten. Der Fragebogen kann bei Vorliegen eines Prototypen und für fertige Dialogsysteme eingesetzt werden. Die Ergebnisse des Fragebogens sollten nach ihrer Erhebung unter Berücksichtigung des Nutzerkontextes interpretiert werden. Der Fragebogen wurde in mehrere Sprachen übersetzt, auch eine deutsche Version ist vorhanden. Die Auswertung des Fragebogens kann mit Hilfe der speziellen Software SUMISCO erfolgen. Das Verfahren ist geeignet, subjektive Bewertungen der Software zu erheben um Hinweise auf mögliche Normabweichungen zu erhalten. Durch den fehlenden Aufgabenbezug des Fragebogens ist im Anschluss an das Ausfüllen des Fragebogens ein Interview nötig. Dabei besteht die Gefahr, dass die Anonymität der Probanden gefährdet wird. SUMI dürfte momentan der psychometrisch am solidesten konstruierte Benutzerfragebogen zur globalen Bewertung von Dialogsystemen sein (s. Dzida et al. 2000).

Zusammenfassung

Die wichtigsten Vor- und Nachteile von standardisierten schriftlichen Befragungen werden hier sehr verkürzt dargestellt (s. Frieling & Sonntag 1999:66f.).

Vorteile:

- Standardisierung: Für alle Befragten sind die Formulierungen gleich, obwohl nicht immer sichergestellt ist, was die Befragten mit den verwendeten Begriffen meinen (d.h. es bestehen konnotative Unterschiede).
- Interviewereinfluss: Dieser ist nicht vorhanden, im Gegensatz zu mündlichen Befragungen.

Nachteile:

- Unvollständige Daten: Ohne Hilfestellung durch einen Interviewer sind bei schriftlichen Befragungen Missverständnisse nicht vermeidbar. Je nach Lust und Laune bleiben auch Fragen offen, so dass bei der Auswertung mit unvollständigen Datensätzen gerechnet werden muss.
- Mangelnde Flexibilität: Das starre Antwortschema lässt häufig keine individuellen Antwortvarianten zu; folglich sollten ergänzende, offene Antwortmöglichkeiten verwendet werden.

4.5.1.2 Interviews

Zur Durchführung eines Interviews muss die Software im Vorfeld vom Versuchsleiter begutachtet werden, um auf mögliche Schwachstellen in der Befragung einzugehen. Wenn die Anzahl der Probanden nicht so groß ist, ist das Interview eine gute Technik um Meinungen und Einstellungen herauszufinden. Der Aufbau eines Interviews reicht von einer festen Vorgabe von Fragen und möglichen Antworten (strukturierten Interview) bis hin zum unstrukturierten Interview. Bei einem strukturierten Interview sind Wortlaut und Abfolge der Fragen eindeutig vorgegeben und für den Interviewer verbindlich. Werden hier die Antwortvorgaben bekannt gegeben, erfährt der Interviewte, was der Interviewer für normal bzw. überzeugend hält, wodurch die Bereitschaft zu einer ehrlichen Beantwortung beeinträchtigt wird. Im Unterschied hierzu ist bei einem unstrukturierten (nichtstandardisierten oder qualitativen) Interview nur ein thematischer Rahmen vorgegeben. Die Gesprächsführung ist offen, d.h. es bleibt der Fähigkeit des Interviewers überlassen, ein Gespräch in Gang zu setzen. Vom Interviewer werden die Äußerungen in Stichworten mitprotokolliert oder mit einem Tonbandgerät aufgezeichnet. Das nichtstandardisierte Interview hat sich vor allem bei explorativen Studien bewährt, um eine Orientierung über Informationen und Meinungen zu einem Thema zu erfassen. Weitere Interviewformen sind die halb- oder teilstandardisierten Interviews mit teils offenen, teils geschlossenen Fragen und mit unterschiedlicher Standardisierung der Interviewdurchgänge. Bei dieser Interviewform verwendet der Interviewer einen Leitfaden, der ihm mehr oder weniger verbindlich die Art und die Inhalte des Gesprächs vorschreibt. Ein Vor-

teil liegt darin, dass ein Interview als Erfassungsmethode zu jedem Zeitpunkt im SWE-Prozess eingesetzt werden kann. Auf die speziellen Eigenschaften der Interviewsituation gehen z.B. Kromney (1995) Bortz & Döring (2002) ausführlich ein. Da es kaum standardisierte mündliche Interviewmethoden in der Softwareevaluation gibt, werden hier nur einige wenige Methoden diskutiert.

Eine halbstandardisierte Befragungsmethode ist die Repertory Grid Technique, eine Methode aus der Persönlichkeits- (Differenziellen) Psychologie, die zur Exploration des Gestaltungsraums eines Softwareproduktes eingesetzt werden kann. Anhand dieser Methode lassen sich diverse Gestaltungsentwürfe für die Software in einer systematischen Weise evaluieren. (s. Hamborg, Hassenzahl & Wessel, n.d.). In einem ersten Schritt werden jeweils drei aus einer Menge von *N* Gestaltungsentwürfe - häufig als Papierprototypen - gegenübergestellt. Die Probanden werden gefragt, in welche Weise die zwei Entwürfe zueinander passen und sich von der dritten unterscheiden (s. Dick 2000). Die gefundenen Ähnlichkeiten und Unterschiede werden mit der persönlichen Sichtweise der potentiellen Benutzer (ihre Erwartungen, Einstellungen, Bedürfnisse, etc.) in Form sogenannter „persönlicher Konstrukte“ erhoben. Solche Konstrukte sind z.B. „fachmännisch-unseriös“, oder „hat Spaß gemacht- ernsthaft, gut für die Arbeit“. Dadurch erhalten Entwickler die Gelegenheit zu erfahren, wie ihre Gestaltungsentwürfe von potentiellen Benutzern wahrgenommen werden und welcher Entwurf den Anforderungen an die Software am ehesten genügt. Die Repertory Grid Technique zielt auf „Verstehen durch Gestalten“ ab, da konkrete Entwürfe benötigt werden. Für Hamborg, Hassenzahl & Wessel (n.d.) ist es auch vorstellbar, diese Technik mit abstrakten Bedienkonzepten durchzuführen.

Formen von Interviews

Der Variantenreichtum mündlicher Befragungen (Interviews) ist enorm und kann daher hier nur lückenhaft dargestellt werden. Interviews unterscheiden sich (s. Bortz & Döring 2002:238f.):

- nach dem Ausmaß der Standardisierung (strukturiert, halb strukturiert, unstrukturiert),
- nach der Art des Kontaktes (direkt, telefonisch-elektronische Medien, schriftlich),
- nach der Anzahl der befragten Personen (Einzelinterview, Gruppeninterview, Survey),
- nach dem Autoritätsanspruch des Interviewers (weich, neutral, hart).

Bei der Durchführung von Interviews ist darauf zu achten, dass es einen großen Unterschied zwischen den Selbstauskünften von Benutzern (Fragebo-

gen/Interview) und ihrem tatsächlichen Verhalten geben kann, d.h. wenn sie die Software mit einem echten Anliegen benutzen.

Die Benutzer könnten beispielsweise antworten, dass die Software gut sei, weil sie höflich sein wollen, Ihre Mühe honorieren wollen oder Sie ermutigen wollen, mit ihrem Projekt fortzufahren. Genauso gut könnten sie auch äußern, dass die Benutzungsschnittstelle schlecht sei, da sie möglicherweise kein schickes Design hat. Tatsächlich war sie aber bei jeder Recherche erfolgreich, weil sie stattdessen klar und einfach ist.

Mündliche Befragungen haben eine Reihe von Vor- und Nachteilen, die hier sehr stark gekürzt wiedergegeben werden (s. Frieling & Sonntag 1999:72):

Vorteile mündlicher Befragungen

- Flexibilität: Der Interviewer kann sich den Bedürfnissen des Befragten anpassen und unverständliche Fragen erläutern.
- Spontaneität: Die impulsiven Reaktionen des Befragten geben teilweise mehr Aufschluss als durchdachte Reaktionen. Für den Interviewer ergibt sich die nicht immer leichte Aufgabe, die Aussagen angemessen festzuhalten.
- Nonverbale Reaktionen: Reaktionen wie Gesten, Lachen, Erröten etc. können neben verbalen Reaktionen aufschlussreiche Zusatzinformationen bieten, die aber immer sehr zurückhaltend interpretiert und ausgewertet werden sollten.
- Identifikation: Der Befragte muss persönlich Stellung nehmen, er kann sich nicht hinter einer anonymen Antwort (wie beim Fragebogen) verstecken.
- Vollständigkeit: der Interviewer kann dafür Sorge tragen, dass alle Fragen, soweit sie für den Befragten beantwortbar sind, auch beantwortet werden.
- Lese- und Schreibfähigkeit: Für Personen mit geringen Schreib- und Lesekenntnissen ist das mündliche Interview am angebrachtesten; selbst bei schlechten Deutschkenntnissen kann vieles durch Umschreibungen und einfache Erläuterungen verdeutlicht werden.

Nachteile mündlicher Befragungen

- Kostenaufwand: Werden größere Stichproben in die Untersuchung einbezogen, sind die Kosten für Reisen, Terminvereinbarungen, Interviewertraining, etc. ungleich höher als bei schriftlichen Befragungen.

- Eingeschränkte Anonymität: Die Aufhebung der Anonymität im Gespräch kann als Bedrohung empfunden werden, die zu einer Verfälschung der Antworten oder zur Teilnahmeverweigerung führt.
- Interviewereinfluss: Persönliche Merkmale des Interviewers wie Alter, Geschlecht, Dialekt, äußere Erscheinung und Auftreten können zu systematischen Fehlern führen.

4.5.2 Logging Actual Use (Logfile Recording)

Hier wird von einem Erfassungssystem automatisch ein Protokoll des Systemverhaltens und der Benutzereingaben erstellt. Erfasst werden alle interaktiven Daten, die bei der Arbeit mit dem zu untersuchenden System anfallen. Dies betrifft beispielsweise Tatstatureingaben und Mausbewegungen (s. Nielsen 1993; vergl. auch Kapitel 2.6.1).

4.5.3 Incident Diaries

Hierbei handelt es sich um Mini-Fragebögen, die der Benutzer während des Tests immer dann ausfüllt, wenn ein Problem bei der Nutzung des Testobjekts auftritt. Hier kann erfragt werden, welches Problem aufgetreten ist, wie der Benutzer es gelöst hat (wenn er es lösen konnte) und als wie schwerwiegend er das Problem einstuft. Bei dieser Methode muss der Benutzer sein Problem in seiner eigenen Sprache vermitteln. Anhand der Ausführungen kann abgeleitet werden, wie gut der Benutzer das System verstanden hat und welche Aspekte des Systems in Bezug auf Benutzerfreundlichkeit noch zu verbessern sind (Nielsen 1994).

4.5.4 Videoaufzeichnung

Verschiedene Gründe können dafür sprechen, den Test auf Video aufzuzeichnen. Dabei kann in einer nachträglichen Auswertung die Interaktionssituation detaillierter analysiert werden. Videosequenzen von Bedienproblemen können eine starke Argumentationshilfe für geforderte Designänderungen sein. Als Ausgangspunkt für detailliertere Auswertung kann mit Hilfe von Videoaufzeichnungen systematisch eine Liste von Fehlerhäufigkeiten erstellt werden. Um den Einfluss der sozialen Erwünschtheit in der Erhebungssituation oder Hemmungen vor dem Aufzeichnungsgerät möglichst gering zu halten, fordert Irion (2002), dass die Probanden über den Zweck der Aufzeichnung informiert und in einer Vorlaufphase an das Gerät gewöhnt werden. Während der Aufzeichnung sollte das Aufnahmegerät in Vergessenheit geraten. Es sollte

auch vor der Aufzeichnung festegelegt werden, ob es nötig ist einen Zeitcode dauerhaft in die Aufnahme einzublenden (s. Irion 2002).

Weitere mögliche Untersuchungsziele sind die Zeit, die mit Problemen und deren Behebung zugebracht wird, die Zahl der Benutzer, die ein bestimmtes Problem haben und die Zahl der Schritte, um ein bestimmtes Ziel zu erreichen. Videoaufnahmen sind sehr hilfreich, wenn ein Entwicklerteam zu überzeugen ist, das hartnäckig an einem alten Entwurf festhalten will.

4.6 Vergleich von Software-Evaluationsmethoden

Mittlerweile gibt es eine Vielzahl von Studien, in denen Evaluationsmethoden in Bezug auf deren Effektivität, Effizienz und ihren Nutzen untersucht werden. Insbesondere wurde der Vergleich von Inspektionsmethoden als kostengünstige Varianten zu Testmehtoden verfolgt (s. Nielsen 1993; Nielsen & Mack 1994). Vergleichsuntersuchungen zur Effektivität und Effizienz von Inspektions- und Testmethoden zeigen widersprüchliche Ergebnisse. Während die Ergebnisse von Jeffries et al. (1991) zugunsten der Inspektionsmethoden ausfallen, bewerten Karat et al. (1992) Inspektionsmethoden im Vergleich zu Testmethoden als weniger effektiv. Werden der finanzielle und zeitliche Aufwand einer empirischen Überprüfung gescheut, so sind die Inspektionsmethoden ein wertvolles Werkzeug, das als Ersatz zu Usability-Tests dienen kann und als Alternative zum Nicht-Testen dienen soll. Sollten aber genauere Leistungsmaße bei einer Untersuchung erhoben werden, können empirische Tests nicht ersetzt werden. Analysen zeigen, dass Inspektionsmethoden im Gegensatz zu Usability-Tests nicht in der Lage sind, Ziele klar zu evaluieren. Inspektionsmethoden decken Genauigkeits-, Systematik- oder Konsistenzprobleme auf. Inspektionsmethoden sind besser geeignet Probleme zu finden als Problemlösungen vorzuschlagen. Wenn es darum geht, die Ganzheitlichkeit einer Aufgabe zu sehen, sind empirische Testmethoden aber von Vorteil. Wenn ausreichend Ressourcen zur Verfügung stehen, ist eine optimale Strategie, die Vorteile beider Methoden zu verwenden, indem sie kombiniert werden. Im Idealfall werden beide Methoden kombiniert (Nielsen n.d.). Zu bedenken ist, dass ein Test immer eine ungewöhnliche Situation für die Versuchspersonen darstellt und die ermittelten Ergebnisse notwendigerweise nicht verlässlich sind. Außerdem sind die Versuchspersonen als nicht voll repräsentativ im Sinne der Zielbenutzer einzustufen.

5 Discount Usability Engineering

In den vorherigen Kapiteln wurden Methoden und Verfahren vorgestellt, um die Usability von Softwareprodukten zu evaluieren. Alle genannten Methoden haben ihre Vor- und Nachteile, die sorgfältig gegeneinander abgewogen werden müssen, um für bestimmte Usability-Ziele die bestmögliche Methode auszuwählen und einzusetzen. Der Discount Usability-Ansatz bewertet spezielle Vorgehensweisen und Methoden zum Testen der Usability eher aus einer „gut genug für die Praxis"-Perspektive als mit wissenschaftlichen Maßstäben. Laut Nielsen (1993) ergeben sich die folgenden Hauptnachteile:

- Hohe Kosten
- Hoher Aufwand
- Enormer Zeitbedarf
- Mangelndes Verständnis bei Entwicklern und Software-Managern

Daraufhin entwickelte Nielsen ein Verfahren, um die offensichtlichen Nachteile zu beseitigen. Dabei sei es vorteilhafter, einfache und verständliche Methoden einzusetzen, die jedoch nicht die Qualität komplexer Verfahren erreichen. Diese Verfahren werden eher ausgewählt und eingesetzt, da sie weniger Aufwand und dadurch geringere Kosten verursachen (s. Nielsen 1993). Die Discount Usability-Methoden[12] setzen sich aus verschiedenen Verfahren zusammen, die teilweise in den vorigen Kapiteln vorgestellt wurden (s. Nielsen n.d.; Nielsen 1993:17):

- Benutzer- und Aufgabenbeobachtung
- Szenarien
- Vereinfachtes Lautes Denken
- Heuristische Evaluation (s. Kapitel 3.2.2)

Die entscheidende Rolle bei diesem Vorgehen spielen das frühzeitige Einbeziehen der Endbenutzer (z.B. am Arbeitsplatz).

5.1 Benutzer- und Aufgabenbeobachtung

Beim Einsatz von Discount Usability-Methoden ist der Fokus auf den Benutzer gerichtet. Die Grundannahme der Aufgabenbeobachtung bzw. -analyse besteht darin, den Benutzer nur zu beobachten und ihn nicht bei der Arbeit zu

[12] siehe URL: http://www.useit.com/papers/guerrilla_hci.html

stören. Bis auf die Heuristische Evaluation, die bereits detailliert dargestellt wurde, werden weitere Methoden näher beschrieben.

5.2 Szenarien

Nielsen (1993:18) versteht hierunter eine vereinfachte und günstige Art von Prototypen, die durch das T-Modell Prototyping realisiert werden. Unter Szenarien sind Arbeitsaufgaben zu verstehen, die die Benutzer unter Beobachtung abarbeiten müssen. Diese Aufgaben können nicht nur auf fertigen System-Oberflächen, sondern auch als Papierentwurf (paper mock-ups) durchgeführt werden (s. Nielsen 1993). Die Schnittmenge aus den horizontalen (Anzahl der Funktionalitäten) und vertikalen Prototypen (Ausprägung der Funktionalität) ergibt das Szenario (s. Abbildung 7). Das Szenario kann genau auf das Ziel des Tests hin konzipiert und gewählt werden.

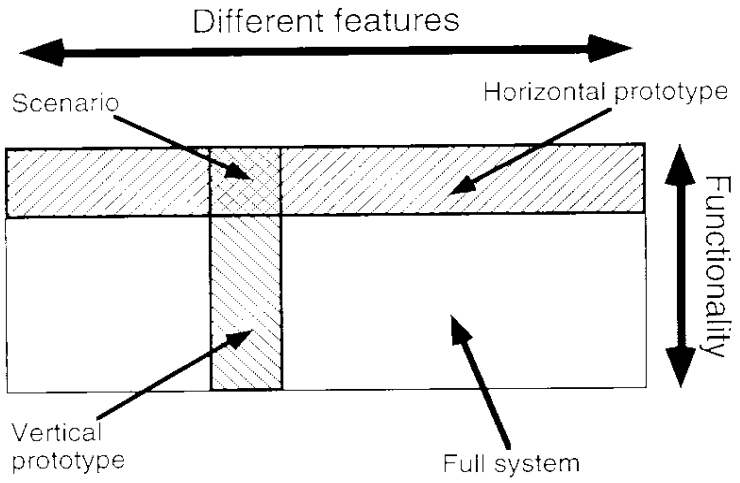

Abbildung 7: Die zwei Dimensionen des Prototyping (Nielsen 1993:94)

5.3 Vereinfachtes Lautes Denken

Die Methode des Vereinfachten Lauten Denkens greift auf die gleichen Inhalte zurück wie die Methode des Lauten Denkens. Die Divergenzen liegen in der Expertise der Testbegleiter und der Methode zur Aufzeichnung der Äußerungen des Benutzers. Bei der traditionellen Methode muss der Testbegleiter (oft ein Psychologe) über ein fundiertes Fachwissen verfügen, die Daten des Tests werden aufgezeichnet. Beim Discount Usability-Test wird auf den Experten und eine komplette Dokumentation verzichtet. Software-Entwickler

sind durch eine kurze Einführung in diese Verfahren im Stande, diese Methode einzusetzen. Dabei stützt sich die Analyse auf die Notizen, die während des Tests anfallen. Laut Niesen (1993) reicht die gekürzte Version aus, um einen großen Anteil der Usability-Probleme erkennen zu können (s. Nielsen 1993:18f.).

6 Modellierung der Benutzungsschnittstelle

Das Ziel der Modellierungsverfahren ist es, die analysierten Aufgaben zu beschreiben und zu validieren. Dies betrifft einerseits die eigentlichen Arbeitsabläufe, andererseits die mentalen Modelle der Benutzer über die Aufgabe und darüber, wie das Anwendungssystem die Aufgabe abbildet. Man spricht hier auch von kognitiver Modellierung. Mit Hilfe kognitiver Modelle lassen sich Aussagen übe die Leistungen bei der Arbeit mit einer Anwendung, über deren Erlernbarkeit und übe die Möglichkeit des Transfers von Wissen sowie über die Gedächtnisleistung und über wahrscheinliche Fehler ableiten. Bekannte Vertreter der kognitiven Modellierung sind die GOMS-Modellierung und die TAG-Modellierung.

6.1 GOMS-Modell[13]

Das GOMS-Modell von Card et al. (1983; zit. nach Lin et al. 1997) dient zur Beschreibung des prozeduralen Benutzerwissens. Da das GOMS-Modell eine Aufgabenanalyse impliziert, eignet es sich ebenfalls als Leitfaden und Strukturierungshilfe für eine Aufgabenanalyse. Das Modell geht immer von einem geübten Benutzer aus und modelliert daher routinierte Aufgabenbewältigung. Bei der GOMS-Modellierung (Goals, Operators, Methods, Selections rules) werden Aufgaben bzw. Computerprogramme durch eine Hierarchie von Zielen, Operatoren, Methoden und Regeln für die Auswahl dieser Methoden beschrieben (s. Nielsen 1993:257). Als Operatoren werden die elementaren Aktionen der Wahrnehmung, der Entscheidung und des Handelns bezeichnet. Methoden setzen sich aus Unterzielen und Operatoren zusammen. Die Selektionsregeln bestimmen, welche Methoden für welches Ziel verwendet werden. Da für bestimmte Operatoren wie Mausbewegungen und Tastaturanschläge der Zeitaufwand für die Ausführung bekannt ist, kann die Effizienz der Benutzungsschnittstelle für die modellierten Aufgaben durch Summierung dieser

[13] Siehe URL: http://www.gomsmodel.org/

Zeiten abgeschätzt werden. Das GOMS-Modell (s. Tabelle 5) setzt sich aus vier wesentlichen Elementen zusammen, von denen es auch seinen Namen erhalten hat:

Tabelle 5: Die vier Elemente des GOMS-Modells (Wandmacher, 1994)

Ziele Das GOMS-Modell postuliert eine hierarchische Struktur von Zielen und Teilzielen auf verschiedenen Ebenen. Die oberen Ebenen der Zielstruktur sind aufgabenspezifisch und mehr oder wenige unabhängig vom Softwaresystem. Zum Beispiel ist das Ziel „Erstelle Grafik“ eine relativ abstrakte systemunabhängige Zielformulierung. Die unteren Ebenen der Zielformulierung hängen dagegen von der jeweiligen Benutzungsoberfläche des Systems ab. **Operatoren** Das Verhalten des Benutzers wird im Rahmen des GOMS-Modells als eine Folge von nacheinander ausgeführten Operatoren beschrieben. Ein Operator ist eine elementare, kognitive, perzeptuelle oder motorische Aktion, durch welche der Zustand des Benutzers oder die Aufgabenumgebung verändert wird. **Methoden** Eine Methode ist eine Prozedur, mit der ein bestimmtes Teilziel erreicht werden kann. Eine Methode kann aus einem einzigen Operator bestehen, sie kann aber auch sehr komplex sein und aus einer Hierarchie von untergeordneten Teilzielen bestehen, wobei jedem Teilziel eine Folge von Operatoren zugeordnet sein kann. **Selektionsregeln** Ein Teilziel kann möglicherweise mit verschiedenen Methoden erreicht werden. Die Auswahl einer Methode wird im GOMS-Modell durch Selektionsregeln gesteuert. Die ausgewählte Methode sollte die im Hinblick auf die aktuelle Aufgabenumgebung zweckmäßigste Methode sein. Eine Selektionsregel spezifizierte die Bedingungen, unter denen eine bestimmte Methode ausgewählt wird.

Das GOMS-Modell erlaubt eine Funktions- und Bedienbarkeitsanalyse auf unterschiedlichen Ebenen von der Erledigung bestimmter Arbeitsziele bis hin zu einfachen Tastenkombinationen. Die exorbitanteste Leistung des GOMS-Modells besteht laut Hasebrook (1995:249) darin, dass es erstmals eine psychologische Theorie der Mensch-Maschine-Interaktion in den Softwareentwurf einführte. Hasebrook (1995:250) führt aber eine Reihe zum Teil weitreichender Nachteile zum GOMS-Modell auf:

- Das Modell beschränkt sich auf bereits erfahrene Nutzer (s. Nielsen 1993:257).
- Es lässt sich nicht eindeutig festlegen, welche Analyseebene verwendet werden soll, um bestimmte Entwurfsprobleme zu lösen.
- Da nur jeweils existierende Programme nachträglich analysiert werden können, ist man in frühen Phasen des Programmentwurfs erfahrungsgemäß auf Spekulationen und bestenfalls Prototypen angewiesen.

- Komplizierte Bediensituationen wie die sogenannten Doppelaufgaben (beispielsweise Tippen und Lesen synchron) und neuere Erkenntnisse zur Arbeitsweise des Arbeitsgedächtnisses werden regelrecht ignoriert.

6.2 TAG-Modellierung

Die Task Action Grammar (TAG[14]) Analyse von Payne und Green (1986; zit. nach Lin et al. 1997) besteht aus der grammatikalischen Beschreibung gültiger Interaktionssequenzen in Form von Ersetzungsregeln. Hier können ähnliche Ersetzungsregeln zu sogenannten Meta-Regeln zusammengefasst werden.

Einzelregeln

<Wähle Kreis>::= <Wähle Kreissymbol> <Drücke OK-Knopf>

<Wähle Linie>::= <Wähle Liniensymbol> <Drücke OK-Knopf>

<Wähle Rechteck>::= <Wähle Rechtecksymbol> < Drücke OK-Knopf>

Metaregeln

<Wähle FORM>::= <Wähle FORMsymbol> <Drücke OK-Knopf>

FORM besteht aus Kreis, Linie, Rechteck.

Da nur syntaktisch und semantisch konsistente Ersetzungsregeln zu Meta-Regeln zusammengefasst werden können, ist die Anzahl der zur Beschreibung einer Dialogsprache notwendigen Meta-Regeln ein Maß für den Grad de Konsistenz einer Benutzungsschnittstelle. Eine TAG-Modellierung beginnt mit der Bestimmung sogenannter *Elementarer Aufgaben*. Das sind einfache Teilaufgaben, deren Ausführung keinerlei Problemlösen beinhaltet und stark von der Benutzungsschnittstelle abhängt. Für die in den elementaren Aufgaben vorkommenden Konzepte werden Mengen mit zulässigen bzw. relevanten Merkmalsausprägungen bestimmt (z.B. „Richtung“ besteht aus „hoch, runter, rechts, links“). Ausgehend von diesen semantisch abgeleiteten Wertemengen wird dann versucht, die Ersetzungsregeln auf möglichst wenige Ersetzungsregeln zu reduzieren. Die TAG-Modellierung kann unter Umständen Inkonsistenzen einer Benutzungsschnittstellensprache aufdecken. Die Anzahl der Regeln dient als Maß für die Komplexität einer Schnittstellensprache. Es drückt aus, wie viele Regeln ein Benutzer lernen bzw. wissen muss um alle

[14] Siehe URL: http://collide.informatik.uni-duisburg.de/Lehre/DidaktikSeminar/BuA/page08.php

gültigen Eingaben ableiten zu können. Die Modellierung ist jedoch auf die systemabhängige Interaktionsebene begrenzt und erfasst nicht semantisch höher stehende Planungsprozesse.

Die beiden Methoden sind sehr gut geeignet eine überzeugende analytische Beschreibung des Mensch-Computer-Systems zu liefern, aber weniger praktikabel für größere Anwendungen.

7 Planungsphasen und Grundlagen des Experimentierens

Die Versuchsplanung steht vor der Datenerhebung und muss mit entsprechender Sorgfalt betrieben werden. Die Datenerhebung wird im Rahmen einer Untersuchungsstrategie verwendet, die die Beantwortung einer bestimmten Fragestellung zum Ziel hat. Die Untersuchungsstrategie führt oft zur Festlegung eines Versuchsplans (Design). Im Versuchsplan ist erwähnt, welche Erhebungen vorzunehmen sind und in welcher logischen Beziehung sie zueinander und zur Fragestellung stehen (s. Zimbardo 1995:26). Unter einem Versuchsplan (Untersuchungsplan) versteht Huber (2002) den logischen Aufbau des Versuchs im Hinblick auf die Hypothesen.

7.1 Auswahl der Variablen

In jedem Experiment spielen drei Arten von Variablen eine wichtige Rolle (Roth 1994:220):

- Unabhängige Variable (UV): Ihr Einfluss soll untersucht werden. Dazu werden sie vom Versuchsleiter planmäßig variiert. Man findet oft für unabhängige Variablen die Bezeichnungen Behandlung (Treatment), gelegentlich auch Faktor (factor). Die Stufen der UV werden Bedingungen oder experimentelle Bedingungen genannt.
- Abhängige Variable (AV): Sie hängt von der Wirkung der unabhängigen Variablen und von Störeinflüssen ab.
- Störvariable: Alle Variablen die sonst noch einen Einfluss auf die abhängige Variable haben. Sie müssen kontrolliert werden, da sie sonst die Eindeutigkeit der Interpretation (interne Validität) gefährden.

Zur Analyse des Einflusses einer unabhängigen Variablen auf eine oder mehrere abhängige Variablen wird häufig ein experimentelles Vorgehen gewählt. Hierbei findet eine systematische Manipulation der unabhängigen Variablen statt, deren Einfluss auf abhängige Variablen untersucht wird. Um sicherzu-

gehen, dass dabei ausschließlich die unabhängige Variable einen Einfluss auf die abhängige Variable hat, versucht der Versuchsleiter, die Wirkungen aller äußeren Bedingungen zu kontrollieren. Dies kann auf drei Arten geschehen:

- durch den Einsatz von Experimental- und Kontrollgruppen,
- durch zufällige Zuweisung zu den Versuchsbedingungen,
- und durch die Standardisierung des Experimentablaufs (s. Zimbardo 1995:26).

In der Experimentalpsychologie treten die Bezeichnungen Experimentalgruppe und Kontrollgruppe auf. Die Experimentalgruppe ist die Gruppe, bei der diejenige Stufe der UV realisiert wird, die den Forscher interessiert. Sie erhält das experimentelle Treatment (Behandlung). Die Kontrollgruppe erhält dagegen das Treatment nicht, ansonsten gibt es keine Unterschiede in den Versuchsbedingungen, für Experimental- und Kontrollgruppe. Ein Vergleich der Kontrollgruppenergebnisse mit jenen der Experimentalgruppe ermöglicht eine Aussage über Ausmaß und Richtung und Wirkung der experimentellen Bedingungen (s. Dorsch 1994:404).

Variablen können sein (s. Roth 1994:222f.):

- Verbale Stellungnahmen: Sie sind sowohl bei experimentellen als auch bei nicht-experimenteller empirischer Forschung einsetzbar. Sie haben den Vorteil, dass sie leicht erhoben werden können. Hierbei finden oft Fragebögen ihren Einsatz.
- Leistungen der Versuchsperson: Die Versuchsperson hat beispielsweise ein Problem zu lösen oder auf ein Signal möglichst schnell zu reagieren. Vorwiegend werden hier die benötigte Zeit, die Menge (Zahl der gelösten Aufgaben) und die Fehler ausgewertet.

7.2 Die Kontrolle von Störvariablen

Zur Kontrolle von Störvariabeln gib es eine Reihe von experimentellen Techniken, die dazu dienen, sicherzustellen, dass die experimentellen Bedingungen sich *nicht* auch in anderer Hinsicht unterscheiden (s. Huber 2002).

Die Störvariablen lassen sich in zwei Gruppen gliedern: Störvariablen der Versuchsperson und Störvariablen der Untersuchungssituation (Versuchsleiter, Untersuchungsraum und Reihenfolge von Fragen, etc.).

7.2.1 Störvariablen der Versuchsperson

Bortz und Döring (2002:525) verstehen unter personengebundenen Störvariablen, „wenn sich die Untersuchungsteilnehmer der einen Stichprobe von den Untersuchungsteilnehmern der anderen Stichprobe nicht nur bezüglich der unabhängigen Variablen, sondern auch in bezug auf weitere, mit der abhängigen Variablen zusammenhängende Merkmale unterscheiden“.

Zufallsaufteilung (Randomisierung) ist die wichtigste Technik zur Kontrolle personengebundener Störvariablen. Durch zufällige Zuweisung der Vpn zu den Untersuchungsbedingungen werden Experimental- und Kontrollgruppe im Wege des statistischen Fehlerausgleichs vergleichbar. Diese Technik sorgt für Äquivalenz bezüglich aller personengebundener Störvariablen. Durch Zufallsaufteilung ist es möglich, eine Vielzahl von Störvariablen, die inhaltlich nicht einmal bekannt zu sein brauchen, auf einmal unter Kontrolle zu bekommen. Äquivalenz ist um so mehr sichergestellt, je größer die zu vergleichenden Stichproben sind (Empfehlung: mindestens 20 Untersuchungsteilnehmer pro Experimental- und Kontrollgruppe).

Falls eine Randomisierung nicht möglich ist, stehen für die Kontrolle personengebundener Störvariablen in quasiexperimentellen Untersuchungen die nachfolgenden Techniken zur Verfügung (s. Bortz & Döring 2002:527; Zimbardo 1995:26f.):

- **Konstanthalten**
  Personengebundene Störvariablen beeinflussen die Unterschiedlichkeit von Vergleichsgruppen nicht, wenn sie herausgehalten werden.
- **Parallelisieren (matching)**
  Die Störvariable wird zunächst bei jeder Versuchsperson gemessen. Danach unterteilt man die Versuchspersonen in Gruppen ein, die von ihren Durchschnittswerten möglichst gleich sein sollten. Parallelisieren ist vor allem bei kleinen Stichproben besser als das Randomisieren, weil beim Randomisieren bei kleinen Stichproben die Wahrscheinlichkeit zu groß ist, zufällig Extremgruppen zu erzeugen. In der Regel erfordert die Parallelisierung eine eigene Vortest-Sitzung. Dabei wird von allen Vpn die Variable erhebt, nach der parallelisiert werden soll. Für K (=Anzahl) der experimentellen Bedingungen benötigt man K Parallelisierte Gruppen (s. Roth 1994:226).
- **Matched Samples**
  Dieses Verfahren findet vor allem bei kleineren Stichproben (nicht mehr als ca. 20 Untersuchungsteilnehmer pro Vergleichsgruppe) Anwendung. Dabei werden die Untersuchungsteilnehmer der Stichprobe

einander paarweise (bei zwei Vergleichsgruppen) in Bezug auf das oder die zu kontrollierende Störvariablen zugeordnet.

- **Mehrfaktorielle Pläne**
  Der Einfluss einer Störvariablen lässt sich kontrollieren, wenn man sie als gesonderten Faktor in einem mehrfaktoriellen Versuchsplan mit berücksichtigt (s. Bortz & Döring 2002:531ff.).
- **Kovarianzanalytische Kontrolle**
  Mittels der Kovarianzanalyse kann auf rechnerischem Wege die Beeinflussung einer abhängigen Variablen durch personengebundene Störvariablen kontrolliert werden (s. Bortz & Döring 2002:544f.).

Neben Variablen, die der Versuchsleiter leicht manipulieren kann, gibt es auch zahlreiche Variablen – wie beispielsweise Alter, Geschlecht, Religionszugehörigkeit und Intelligenz – die der Versuchsleiter nicht aktiv manipulieren kann. Damit entfällt hier die Randomisierung als wirkungsvolle Kontrollmöglichkeit, weshalb eine isolierende Variation nicht manipulativer Variablen nur schwer bis gar nicht durchzuführen ist (s. Roth 1994:221f.).

Als eine weitere wichtige Einflussgröße bei der Aufgabenbearbeitung hat sich die Vorerfahrung der Versuchspersonen herausgestellt. Diese vermittelnde (intervenierende) Variable sollte gemessen werden, um sie später bei der statistischen Auswertung berücksichtigen zu können. Die Vorerfahrung kann dabei mit einem Fragebogen erhoben werden. Hierbei ist es ratsam, getrennt nach Nutzungsdauer (ND; in Jahren, Monaten, Wochen, etc.) und Nutzungshäufigkeit (NH; in Stunden/Woche, etc.) von verschiedenen interaktiven Systemen zu ermitteln, um anschließend einen Nutzungs-Intensitäts-Index (NII= ND* NH; in Stunden)zu berechnen (Rautenberg 1991: 8).

7.2.2 Störvariablen bei mehreren experimentellen Bedingungen pro Versuchsperson

Hier wird auf zwei spezielle Typen von Störvariablen eingegangen – Positionseffekte und Carry-Over-Effekte – und ihre Kontrolle behandelt. Diese Störvariablen treten bei Versuchsplänen auf, bei denen eine Versuchsperson (Vp) nicht nur einer einzigen experimentellen Bedingung ausgesetzt wird, sondern mehreren. Positionseffekte und Carry-Over-Effekte können auch kombiniert auftreten. Der Hauptvorteil eines Versuchsplans mit Messwiederholung an denselben Vpn besteht in seiner besonderen Ökonomie. Der Hauptnachteil liegt darin, dass die erste Messung Folgewirkungen hat, die leicht mit den experimentellen Effekten konfundieren können.

	Zeitpunkt t1 UV (a1)	Zeitpunkt t2 AV	Zeitpunkt t3 UV (a2)	Zeitpunkt t4 AV
Vp1				
Vp2				
…..				

Abbildung 8: Versuchsplan mit Messwiederholung

Bei einem Versuchsplan mit Messwiederholung ist die Parallelisierung vieler Störvariablen, die von der Versuchsperson her stammen (Alter, Geschlecht, etc.), von Vorteil. Man braucht weniger Versuchspersonen, da ja jede Versuchsperson mehrere experimentelle Bedingungen durchläuft.

Problematisch wäre bei diesem Design, wenn alle Versuchspersonen die gleiche Reihenfolge von Bedingungen erhalten. Denn die Reihenfolge ist eine Quelle von zahlreichen und konsequenzenreichen Störvariablen.

7.2.2.1 Positionseffekt und dessen experimentelle Kontrolle

Ein Positionseffekt ist eine Störvariable, die von der Position einer experimentellen Bedingung in der Reihenfolge bestimmt ist. Beispiele hierfür sind: Langeweile, Übungseffekte, etc. Die Kontrolle von Positionseffekten wird durch die nachfolgenden Verfahren näher beschrieben:

Vollständiges Ausbalancieren
Alle möglichen Reihenfolgen werden erzeugt und dem entsprechenden Anteil an Versuchspersonen zugeordnet. Beispiel: bei 3 experimentellen Bedingungen x, y und z gibt 3! =3*2*1 mögliche Reihenfolgen:

- x-y-z
- x-z-y
- y-x-z
- y-z-x
- z-x-y
- z-y-x

Die Positionseffekte werden somit über alle Versuchspersonen hinweg ausgeglichen.
Der Nachteil ist, dass sehr viele Versuchspersonen benötigt werden. Mit dem Hinzukommen experimenteller Bedingungen explodiert die Zahl möglicher

Reihenfolgen kombinatorisch (bei n experimentellen Bedingungen ergeben sich n! Reihenfolgen).

Unvollständiges Ausbalancieren
Nur eine Unterzahl aller Möglichkeiten wird realisiert.

- *Zufallsauswahl:*
  Aus allen möglichen Reihenfolgen von experimentellen Bedingungen werden mit Hilfe eines Zufallsverfahrens diejenigen ausgewählt, die den Versuchspersonen zugeordnet werden. Jede Versuchsperson erhält eine andere Reihenfolge. Sollte nur angewendet werden, wenn viele Versuchspersonen am Experiment teilnehmen. Es ermöglicht eine Kontrolle des Positionseffektes.
- *Spiegelbildmethode*
  Nur eine einzige Reihenfolge von experimentellen Bedingungen wird herausgegriffen, die dann in umgekehrter Reihenfolge angehängt wird: Die Reihenfolge x-y-z wird herausgegriffen, bei der Spiegelbildmethode ergibt sich die folgende Sequenz: x-y-z-z-y-x. Alle Versuchspersonen erhalten diese Sequenz.
- *Methode des Lateinischen Quadrats*
  Hier greift man genau so viele Reihenfolgen heraus, wie es experimentelle Bedingungen gibt. Die Versuchspersonen werden gleichmäßig auf die Reihenfolgen aufgeteilt. Die Reihenfolgen werden ganz gezielt nach einem bestimmten Schema konstruiert.

	Position 1 exp.Bed.	Position 2 exp.Bed.	Position 3 Exp.Bed.	Position 4 exp.Bed.
	a b c d	a b c d	a b c d	a b c d
Gruppe 1	x	x	x	x
Gruppe 2	x	x	x	x
Gruppe 3	x	x	x	x
Gruppe 4	x	x	x	x

Abbildung 9: Lateinisches Quadrat als faktorieller Versuchsplan
In diesem Versuchsplan werden nur die mit einem x gekennzeichneten Kombinationen realisiert.

Der besondere Vorteil dieser Methode ist, dass er in einen faktoriellen Versuchsplan integriert werden kann. In diesem Versuchsplan wird die Position der einzelnen Bedingungen als UV eingeführt. Jeder Reihenfolge wird einer Gruppe von Versuchspersonen zugeordnet. Der Vorteil

eines derartigen Versuchsplanes ist, dass der Positionseffekt nicht nur kontrolliert werden kann, sondern gleichzeitig in seiner Wirkung überprüft wird. Der Versuchsplan für das Lateinische Quadrat wird in Abbildung 9 dargestellt.

7.2.2.2 Carry-Over-Effekt und dessen Kontrolle

Dies ist eine Störvariable, die davon herrührt, dass eine frühere experimentelle Bedingung eine spätere inhaltlich beeinflusst. Hier ist nicht die absolute Stellung einer experimentellen Bedingung wichtig, sondern nur, welche speziellen andere(n) Bedingung(en) ihr vorausgegangen sind. Für den Carry-Over-Effekt ist es unerheblich, ob die experimentelle Bedingung x an 5. oder 8. Position in der Reihenfolge auftritt, sondern nur ob ihr beispielsweise die experimentelle Bedingung y vorausgeht.

Bei der Kontrolle von Carry-Over-Effekten sind zwei Maßnahmen wirksam:

- Ist die Ursache des Carry-Over-Effektes bekannt, so sollte(n) die experimentelle(n) Bedingung(en) so umgestaltet werden, dass die Ursache und somit auch der Carry-Over-Effekt beseitigt wird.
- Rückkehr zu einem Versuchsplan, bei dem jede Versuchsperson nur einer einzigen experimentellen Bedingung ausgesetzt wird. Dadurch kann der Carry-Over-Effekt auf keinen Fall auftreten.

7.2.3 Störvariablen aus der sozialen Situation des Experiments

Es darf nicht außer Acht gelassen werden, dass ein Experiment immer auch eine soziale Situation mit all ihren Konsequenzen ist.

7.2.3.1 Versuchsleiter-Erwartungseffekt

Der Versuchsleiter wird als Quelle für eine Vielzahl von Störvariablen besprochen (s. Roth 1994:252f.), z.B. für den Versuchsleiter-Erwartungseffekt (Rosenthal-Effekt).

- Die Versuchsperson bekommt durch Bemerkungen, Gestik und Mimik des Versuchsleiters Hinweise zum richtigen Umgang mit dem System.
- Vermittelt der Versuchsleiter einen zu kompetenten Eindruck, so kann dies dazu führen, dass die Versuchsperson vorschnell Fragen an den Versuchsleiter richtet, wenn ein Problem aufgetreten ist. Im Idealfall vermittelt der Versuchsleiter den Versuchspersonen ein Gefühl der Art, dass er bezüglich des Wissens um das System genauso unbeholfen ist wie die Versuchsperson selbst. Dies hat zur Folge, dass die Versuchs-

person ein wesentlich natürlicheres Agieren und erhöhte Anstrengungsbereitschaft zur Lösung auftretender Probleme zeigt.

Zur Kontrolle des Versuchsleiter-Erwartungseffekts können folgende Techniken eingesetzt werden

- Standardisierung der Versuchsbedingungen (Instruktion per textueller Vorgabe oder Video)
- Training des Versuchsleiters (hinsichtlich Mimik, Gestik, Sprachweise, Betonung, usw.)

Dementsprechend ist es ratsam, auch die Art und Anzahl der gegebenen Hilfestellungen des Versuchsleiters während der (experimentellen) Untersuchung auf einem Protokollbogen oder anhand eines entsprechenden Protokollierungssystems aufzuzeichnen. Die Anzahl an Hilfestellungen muss positiv mit der Aufgabenbearbeitungszeit korrelieren, um zu gewährleisten, dass die Hilfestellungen durch den Versuchsleiter nicht zu einer künstlichen Verkürzung der Bearbeitungszeit geführt haben.

7.2.3.2 Versuchspersoneneffekte

Die Erwartungen der Versuchsperson können ihr Verhalten in einer wissenschaftlichen Untersuchung stark beeinträchtigen. Motive, die das Verhalten im Experiment beeinflussen:

- Kooperation - Nichtkooperation
  Dies betrifft die Frage, ob sich eine Versuchsperson dem Versuchsleiter gegenüber kooperativ verhält oder unkooperativ. Im ersten Fall wird die Versuchsperson die Anweisungen befolgen und insgesamt versuchen, dem Versuchsleiter zu helfen (z.B. dadurch, dass sie sich entsprechend seiner Hypothese verhält). Im zweiten Fall wird die Versuchsperson die Anweisungen des Versuchsleiters - wenn möglich - unterlaufen und insgesamt gegen den Versuchsleiter arbeiten.
- Testangst - Bewertungsangst (evaluation apprehension)
  Versuchspersonen haben Angst, an einem Test teilzunehmen. Sie fürchten, durchschaut zu werden. Die Unsicherheit und Angst kann das Verhalten der Versuchsperson beeinträchtigen. Die Test- und Bewertungsangst kann auch ein Motiv für die Ablehnung der Teilnahme an einem Experiment sein.

7.2.4 Störvariablen der Untersuchungssituation

Untersuchungsbedingte Störvariablen sind eine weitere Ursache für mangelnde Validität. Sie lassen sich durch die folgende Weise kontrollieren (Huber 2002):

- Elimination: Betrifft die völlige Ausschaltung von Störvariablen. (z.B. Lärm eliminieren, der in den Untersuchungsraum eindringt).
- Konstanthalten: Für die Dauer des Versuchs wird die Störvariable konstant gehalten; alle Versuchspersonen sind ihr gleich ausgesetzt. Dazu gehört, dass für eine gleichmäßige Raumbeleuchtung gesorgt wird und dass der Versuchsleiter sich allen Versuchspersonen gegenüber gleich verhält. Der Experimentator hofft, dass mit der Zufallsaufteilung der Versuchsperson die Störvariable wenigstens über die Gruppen hinweg konstant wirkt. Ein Nachteil des Konstanthaltens von Störvariablen ist, dass es Auswirkungen auf die Generalisierung des Ergebnisses haben kann.

7.3 Einteilung von Experimenten nach dem Ziel

7.3.1 Prüfexperimente

Diese haben das Ziel, eine oder mehrere Hypothesen zu prüfen. Wenn ohne nähere Kennzeichnung von einem Experiment gesprochen wird, ist normalerweise ein Prüfexperiment gemeint.

7.3.2 Erkundungsexperiment

Mit einem Erkundungsexperiment wird das Ziel verfolgt, Daten zu sammeln welche die Bildung einer neuen Hypothese gestatten. Man variiert also eine UV (oder mehrere) noch ohne Hypothesen formuliert zu haben, um zu erkunden, wie die AV darauf regiert. Allgemein wird die empirische Untersuchung mit Erkundungsabsicht auch *pilot study* genannt.

7.3.3 Vorexperiment

Ein Vorexperiment ist ein (meist kleines) Experiment, das im Rahmen der Planung eines (Prüf- oder Erkundungs-) Experimentes durchgeführt wird. Sein Zweck ist die Erprobung und Verbesserung der Durchführung des Experimentes, der Operationalisierungstechniken, etc.

7.4 Experimente versus Quasi-Experimente

Bei einem (echten) Experimente ist der Versuchsleiter in der Lage,

- mindestens eine UV aktiv zu variieren, und
- die Wirkung der relevanten Störvariablen auszuschalten

Bei einem **Quasi-Experiment** ist die zweite Bedingung nicht erfüllt. Oft kann der Versuchsleiter bei einem solchen Experiment nicht zufällig festsetzen, welche Versuchsperson welcher Stufe der UV ausgesetzt wird. Quasi-Experimente sind für Untersuchungen im Feld bzw. unter feldähnlichen Bedingungen besonders wichtig, weil sie hier oft das methodisch schärfste Werkzeug sind, das eingesetzt werden kann. Generell wird quasi-experimentellen Untersuchungen eine geringere interne Validität als experimentellen Untersuchungen zugesprochen. Weil quasiexperimentelle Untersuchungen weniger aussagekräftig sind als experimentelle, ist die Kontrolle personengebundener Störvariablen bei diesem Untersuchungstyp äußerst wichtig. Bei einem Quasi-Experiment ist der Schluss von der AV auf die Wirkung der UV natürlich nicht im gleichen Maß möglich wie bei einem echten Experiment, weil eben UV und Störvariable(n) konfundiert sind. Es muss abgeschätzt und belegt werden, wie stark der Einfluss der nicht kontrollierten Störvariablen ist. Je geringer der (vermutliche) Einfluss der Störvariablen ist, mit desto größerer Sicherheit kann auf die Wirkung der UV geschlossen werden (s. Huber 2002).

7.5 Planung der statistischen Auswertung

Vor der Datenerhebung sollte feststehen, welches Auswertungsverfahren man anwenden möchte. Dabei ist es wichtig, das man die Datenerhebung nicht nach der Auswertungsmethode gestaltet. Vielmehr sollte sich die Auswertungsmethode nach den inhaltlichen Kriterien der Untersuchung richten. In der Auswertungsphase sollten die Daten betrachtet und statistisch ausgewertet werden. Dabei fällt der erste Blick auf die Test-Gütekriterien von Untersuchungen. Diese sind unter Objektivität, Reliabilität und Validität zusammengefasst. Diese haben vor allem die pragmatische Funktion, dass unterschiedliche Tester hinsichtlich einer bestimmten Testung zu vergleichbaren Ergebnissen kommen.

Objektivität
Die Objektivität betrifft die Standardisierung des gesamten Testvorgangs. Die Objektivität eines Tests umfasst die Unterformen: Durchführungsobjektivität,

Auswertungsobjektivität und Interpretationsobjektivität (s. Bortz & Döring 2002:194; Fisseni 1997:66-70; Roth 1994; Zimbardo 1995:522f.).

Validität

Die Validität befasst sich mit der Frage, ob das Ergebnis des Usability-Tests auch die Zielsetzung des Tests widerspiegelt. Es muss dementsprechend sichergestellt werden, dass der Test methodisch korrekt gestaltet wird. Dies betrifft also alle Punkte, die die Gestaltung eines Test betreffen, wie beispielsweise die richtige Auswahl der Versuchspersonen, eine geeignete Testumgebung, sowie die richtige Wahl der Testaufgaben bzw. Testscenarios. Auf die diversen Validitätsarten soll hier nur kurz eingegangen werden. Die einzelnen Validitätsarten sind nicht unabhängig voneinander: So setzt die externe Validität voraus, dass die anderen drei Validitätskriterien erfüllt sind. Eine ausführliche Darstellung finden sie bei Bortz & Döring (2002), Schmid (1992:54-56), Fisseni (1997:93ff.) oder bei Roth (1994).

- **Interne Validität**
  Dieses Gütekriterium hat das Ziel, die bedeutsamen Störvariablen zu kontrollieren. Ist eine wichtige Störvariable mit der UV konfundiert, so ist das Experiment wertlos, weil nicht auf die Wirkung der UV auf die AV geschlossen werden kann. Ein denkbarer Effekt könnte durch die Störvariable verursacht sein (s. Bortz & Döring 2002:504; Roth 1994:247).
- **Externe Validität**
  Die externe Validität liegt vor, wenn das in einem Stichprobenergebnis gefundene Ergebnis auf andere Personen, Methoden der Operationalisierung, Situationen oder Zeitpunkte verallgemeinert werden kann. Haben sich z.B. nur Informatikstudenten an eine Untersuchung beteiligt, so wäre eine Generalisierung auf die Gesamtbevölkerung zumindest fragwürdig (s. Roth 1994:247).
  Ausführlich werden die Einflussfaktoren, die die interne und externe Validität gefährden, z.B. in Bortz & Döring (2002:504). behandelt.
- **Konstruktvalidität**
  Mit Konstruktvalidität ist die Güte der Operationalisierung von UV und AV gemeint. Falls die Operationalisierung zweifelhaft ist, so ist auch der Schluss auf die Sachhypothese kritisch (s. Bortz & Döring 2002:200). Nach Zimbardo (1995:524) bezieht sich die Konstruktvalidität auf das Ausmaß, in dem ein Test, der ein bestimmtes Konstrukt erfassen soll, mit den Ergebnissen anderer Tests, Verhaltensmessungen oder experimentellen Ergebnissen, die bereits als valide Indikatoren des zu messenden Konstrukts gelten, zusammenhängen. Crombach (1960;

zit. nach Schmid 1992:55) fordert drei Schritte bei der Konstruktvalidierung:

- Welche Konstrukte sind für die Testleistung verantwortlich?
- Hypothesenbildung auf Grundlage der Theorie, in der das Konstrukt enthalten ist.
- Empirische Überprüfung der Hypothesen.

Validität statistischer Schlussfolgerungen
Statistische Verfahren lassen sich immer dann rechnen, wenn Zahlen vorliegen. Es stellt sich die Frage, ob die Anwendung des speziellen statistischen Verfahrens überhaupt gerechtfertigt ist. Dies hängt einmal davon ab, ob das statistische Verfahren der Fragestellung und dem Versuchsplan angemessen ist. Es ist ebenfalls zu prüfen, ob das für ein bestimmtes statistisches Verfahren vorausgesetzte Skalenniveau auch tatsächlich erfüllt ist, ob die Verteilungsform nicht zu sehr von der vorausgesetzten abweicht, ob Varianzen homogen sind, usw. (s. Bortz & Döring 2002:57; Huber 2002).

Reliabilität
Die Reliabilität ist neben der Objektivität und der Validität eines der Hauptgütekriterien und äußert sich als Zuverlässigkeit des Testergebnisses. Von Interesse ist die Frage, ob die Wiederholung eines Tests mit demselben System (Retest-Reliabilität) und mit anderen Versuchspersonen in einem bestimmten Zeitabstand ein gleiches oder zumindest ähnliches Ergebnis liefert (s. Bortz & Döring 2002:195ff.). Bei der Parallel-Test Reliabilität wird zu einem Test eine möglichst gleichwertige Parallelform entwickelt und beide Formen zur Testung von zwei identischen Systemen eingesetzt. In der Praxis ist es jedoch schwierig, inhaltlich und formal zufriedenstellende Parallelformen zu entwickeln.

7.6 Signifikanztest für Unterschiedshypothesen

Die Hypothesenprüfung im Kontext empirischer Untersuchungen erfolgt üblicherweise über sogenannte Signifikanztests in Abhängigkeit der zu prüfenden Hypothesenart. Der Signifikanztest ermittelt die Wahrscheinlichkeit (Irrtumswahrscheinlichkeit), mit der das gefundene empirische Ergebnis sowie Ergebnisse, die noch extremer sind als das gefundene Ergebnis, auftreten können, wenn die Populationsverhältnisse der Nullhypothese entsprechen. Unter der Irrtumswahrscheinlichkeit ist diejenige Wahrscheinlichkeit zu verstehen, mit der wir uns irren würden, wenn wir die Nullhypothese (H_o) fälschlicherweise zugunsten der Alternativhypothese (H_1) verwerfen. Das Signifikanzniveau wird charakterisiert durch α, für das per Konvention die Werte 5% und 1% festgelegt sind. Ist die Irrtumswahrscheinlichkeit kleiner als α%,

wird das Stichprobenergebnis als statistisch signifikant bezeichnet (s. Bortz & Döring 2002:496). Auf eine detaillierte Darstellung über den Aufbau eines Signifikanztests wird auf Bortz und Döring (2002, Abschnitt 1.3) verwiesen. Als Unterschiedshypothesen bezeichnen Bortz und Döring (2002:525) Hypothesen, die sich auf die Wirksamkeit einer Maßnahme bzw. Treatment beziehen. Bei Unterschiedshypothesen bei quantitativen Variablen muss man zuvor bestimmen, welcher Aspekt der abhängigen Variablen untersucht werden soll: Geht es um eine Hypothese über die Lokation (Mittelwert, zentrale Tendenz, etc.) die Dispersion (Varianz, Streubereich, etc.) oder die gesamte Verteilung. Weitere Fragen betreffen die Anzahl der zu vergleichenden Stichproben mögliche Abhängigkeit zwischen ihnen, die Form der Verteilung aus denen die Stichproben stammen, und deren Varianzen. Die Tests für Unterschiedshypothesen sind dem von Vorberg und Blankenberger (1999) entwickelten Entscheidungsbaum zu entnehmen. Das Ziel dieses Abschnitts ist es nicht, einen Überblick über möglichst viele statistische Verfahren zu geben. Deshalb wird ein inferenzstatistischen Verfahren zur Auswertung von Unterschiedshypothesen kurz skizziert.

7.6.1 Varianzanalyse (Analysis of Variance, ANOVA) - Vergleich von mehr als zwei Gruppen

Das Gemeinsame aller varianzanalytischen Versuchspläne ist darin zu sehen, dass sie die Unterschiedlichkeit von Versuchspersonen in Bezug auf ein Merkmal (abhängige Variable) auf eine oder mehrere unabhängige Variablen zurückführen. Die bedeutsamsten Einteilungskriterien für Varianzanalysen sind:

Einfaktorielle Versuchspläne und einfaktorielle ANOVA
Der Zwei-Gruppen-Plan ist der einfachste einfaktorielle Plan. Bei diesem Zwei-Gruppen-Plan aus der Psychologie bzw. Medizin arbeitet man mit einer zweifach gestuften unabhängigen Variablen (Abbildung 10) und einer oder mehreren abhängigen Variablen.

Treatmentgruppe	Kontrollgruppe
S_1	S_2

Anmerkung:
$S_{1=}$ Stichprobe 1
$S_{,=}$ Stichprobe 2

Abbildung 10: Zwei-Gruppen-Plan mit Kontrollgruppe

Beim Usability Test kann zwischen den folgenden zwei experimentellen Designs gewählt werden:

Between groups
Zwei oder mehr Gruppen von Versuchspersonen (s. Abbildung 11) nehmen an der Untersuchung teil. Jede Gruppe testet nur eines der Systeme.

Interface 1	Interface 2
S_1	S_2

Abbildung 11: Between groups design

Within groups
Nur eine Gruppe von Versuchspersonen (s. Abbildung 12) nimmt an der Untersuchung teil. Jede Versuchsperson testet alle Systeme. Die Vorteile liegen darin, dass im Vergleich zum Between groups design weniger Versuchspersonen benötigt werden. Problematisch sind aber die eventuell entstehenden Lerneffekte. Auf die Störvariablen und deren Kontrolle bei diesem experimentellen Design wird in Kapitel 7.2.2 näher eingegangen.

Interface1	Interface 1
S_1	S_1

Abbildung 12: Within groups design

Zu einem einfaktoriellen Plan zählt auch der Mehr-Gruppen-Plan. Dieser arbeitet mit einer mehrfach gestuften unabhängigen Variablen und einer oder mehrerer abhängiger Variablen.

Prüfbare Hypothese
Mit einem Versuchsplan mit mehreren Stufen *einer* UV werden grundsätzlich Hypothesen geprüft, die Behauptungen über Unterschiede aufstellen (z.B.: UV-Stufe X unterscheiden sich von den Stufen Y und Z) und über Rangreihen von Wirksamkeit (je höher der Wert der UV, desto höher sollte der Wert der AV sein). Charakteristisch für statistische Verfahren zum Prüfen von Hypothesen mit einfaktoriellen Versuchsplänen sind die (einfaktorielle) Varianzanalyse (für AVn auf Intervallskalenniveau) und die darauf aufbauenden Verfahren, oder entsprechende statistische Modelle für ordinalskalierte AVn (z.B. die Kruskal-Wallis[15] einfaktorielle Varianzanalyse).

[15] Der Kruskal-Wallis H-Test (vgl. Büning & Trenkler 1994:184ff) ist ein nichtparametrisches Pendant zur einfaktoriellen ANOVA

Einfaktorielle Varianzanalyse
Im allgemeinen ist das Ziel der Varianzanalyse (ANOVA), die Signifikanz der Unterschiedlichkeit mehrer Mittelwerte zu testen. Die Bezeichnung wird jedoch aus der Tatsache abgeleitet, dass beim Testen der statistischen Signifikanz von Mittelwertsdifferenzen eigentlich Varianzen (Streuungen) verglichen werden. (Auch Messwiederholungen können durchgeführt werden). Der Begriff einfaktoriell zeigt an, dass der Einfluss nur eines Faktors der in p Stufen vorliegt, auf eine abhängige Variable (AV) untersucht wird. Der Faktor wird auch als unabhängige Variable (UV) oder als Treatment bezeichnet. Dabei liegen die folgenden Hypothesen vor (s. Bortz 1993:397ff.):

- Nullhypothese: $H_0: \mu_1 = \mu_2 = \ldots = \mu_m$.
- Alternativhypothese: $H_1: \mu_l \neq \mu_i$. Mindestens zwei Mittelwerte unterschieden sich, d.h. zwischen irgendwelchen Gruppen ergibt sich ein Treatmenteffekt.

Was macht die Varianzanalyse?
Sie zerlegt die Varianz in den Daten in verschiedene Bestandteile, nämlich in diejenige Varianz, die innerhalb der einzelnen Treatment-Gruppen auftritt und die Varianz zwischen den verschiedenen Gruppen (die also auf das Treatment zurückzuführen ist). Die Zerlegung der Varianzen erfolgt nach der Formel (s. Backhaus, 1989, S. 48-50):

$SS_{tot} = SS_{bt} + SS_{wt}$.

Dies bedeutet im einzelnen:

SS_{tot}: Die Gesamtvarianz, also die quadrierte Abweichung aller Messwerte vom Gesamtmittelwert.

SS_{bt}: „between treatments“, also die Varianz zwischen den Versuchsbedingungen. Dabei handelt es sich um ein Maß für die Größe der Effekte (Mittelwertsunterschiede).

SS_{wt}: „within treatmens“, also die Varianz innerhalb der Gruppen (in diesem Fall als Fehlervarianz zu betrachten).

Aus diesen Werten wird ein F-Wert berechnet: $F = \frac{SS_{bt} / df_{bt}}{SS_{wt} / df_{wt}}$

Dabei bedeutet df_{bt} die Anzahl der Freiheitsgrade zwischen den Gruppen (also die Anzahl der Faktorenstufen m-1) und df_{wt} die Anzahl der Freiheitsgrade innerhalb der einzelnen Faktoren (also die Differenz zwischen Stichprobenumfang und Anzahl der Gruppen n - m). Der resultierende F- Wert wird mit einer theoretische F-Verteilung verglichen. Ist der F-Wert signifikant, heißt dies, dass sich zumindest zwei der Mittelwerte signifikant unterscheiden. Zu-

sätzlich kann man mit sog. ***Einzelvergleiche*** oder ***Kontraste*** überprüfen, ob sich bestimmte Treatments signifikant unterscheiden (s. Bühl & Zöfel 2000:284).

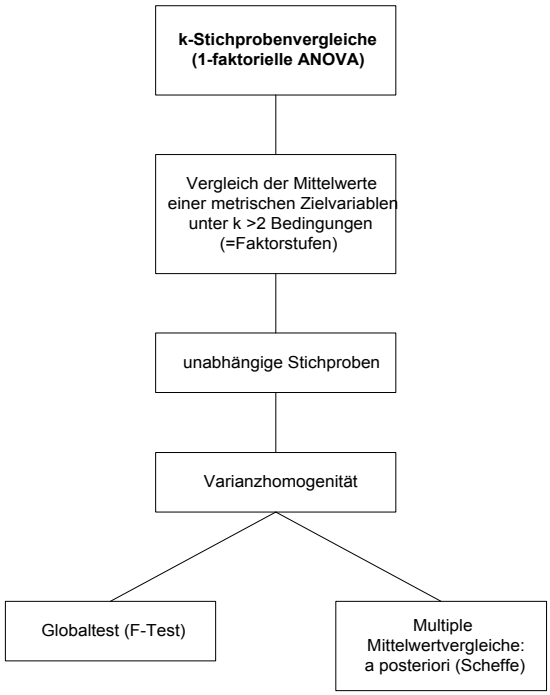

Abbildung 13: Ablauflogik einer einfaktoriellen Varianzanalyse

Hier könnte nun die Frage offen blieben, ob nicht ein t-Test zum gleichen Ergebnis führt. Solange aber nur zwei Stichproben miteinander verglichen werden, ergeben einfaktorielle Varianzanalysen und t-Test dieselben Ergebnisse.

Zweifaktorielle Pläne

Dieser kontrolliert gleichzeitig die Bedeutung von zwei unabhängigen Variablen (Faktoren) für eine abhängige Variable. Ferner informiert dieser Plan über die Kombinationswirkung (Interaktion oder Wechselwirkung) der beiden unabhängigen Variablen (Bortz & Döring, 2002:531).

		UVB	
		b1	b2
UV A	a1	a1b1	a1b2
	a2	a2b1	a2b2
	a3	a3b1	a3b2

Abbildung 14: Versuchsplan mit 2 UVn in Matrixform (3x2 Plan)

Bei diesem Versuchsplan (siehe Abbildung 14) werden die erhobenen Daten mit einer zweifaktoriellen Varianzanalyse statistisch ausgewertet. Hierbei kann man Hypothesen über drei Effekte prüfen: Haupteffekt A, Haupteffekt B und die Interaktion erster Ordnung A x B.

Prüfbare Hypothesen: Haupteffekte und Interaktionen
Mit einem mehrfaktoriellen Versuchsplan lassen sich zunächst einmal Hypothesen über jede der UVn überprüfen. Diese Hypothesen werden auch als Hypothesen über Haupteffekte bezeichnet. Ein Haupteffekt ist also die Wirkung einer UV. Darüber hinaus kann man auch Hypothesen über die Interaktion von Faktoren prüfen.
Eine Interaktion zwischen zwei (oder mehreren) UVn besagt, dass die Wirkung einer UV *nicht unabhängig* von der anderen UN(n) ist.
Bei einem dreifaktoriellen Experiment (UVn A, B, C) gibt es drei mögliche Interaktionen zwischen je zwei Faktoren und eine Interaktion zwischen den drei Faktoren:

Interaktion zwischen A und B,
Interaktion zwischen A und C,
Interaktion zwischen B und C und
Interaktion zwischen A, B und C.

Näheres zur mathematischen Definition von Haupteffekten und Wechselwirkung siehe Bortz und Döring (2002:532ff.) oder Lehrbücher der Varianzanalyse.

Zweifaktorielle Varianzanalyse
Mit der zweifaktorielllen Varianzanalyse wird überprüft, wie eine AV von zwei UVn (Faktoren) beeinflusst wird. Hierbei können nicht nur die Effekte der beiden Faktoren analog zur einfaktoriellen ANOVA getrennt beurteilt werden, sondern auch die Wechselwirkung bzw. die Interaktion der Faktoren kann getestet werden (Bortz 1993:383ff.).

Versuchspläne mit mehreren Uvn und mehrfaktorielle Varianzanalysen
Versuchspläne zur gleichzeitigen Variation mehrerer UVn werden als multi- oder mehrfaktorielle Versuchspläne bezeichnet. Abhängig von der Zahl der betreffenden UVn spricht man von einem zweifaktoriellen, dreifaktoriellen, vierfaktoriellen, etc. Versuchsplan. Bei einem Experiment mit mehreren UVn werden die Stufen der verschiedenen UVn miteinander kombiniert. Bei dem Versuchsplan in Abbildung werden alle die Stufen der UVA mit allen beiden Stufen der UV B kombiniert.
Bei mehrfaktoriellen Varianzanalysen werden die Stufen mehrerer unabhän-

giger Variablen sowie deren Kombinationen in Bezug auf eine intervallskalierte abhängige Variable verglichen.

Varianzanalyse für unabhängige Stichproben
bei denen die untersuchten Gruppen unabhängig voneinander sind oder Varianzanalysen für abhängige Stichproben bzw. mit Messwiederholungen, bei denen wiederholte Messungen einer oder mehrer Stichproben miteinander verglichen werden.

Einfaktorielle Varianzanalyse mit Messwiederholung
Die einfaktorielle Varianzanalyse mit Messwiederholung prüft, ob zwischen den Mittelwerten von p abhängigen Stichproben Unterschiede bestehen.

Abhängige Stichproben resultieren, wenn
a) eine Stichprobe von *n* Personen sukzessive allen p Untersuchungsbedingungen ausgesetzt wird,
b) an n Personen zu p Zeitpunkten Messwerte erhoben werden.

Kovarianzanalyse
Wenn eine für die AV relevante Drittvariable (Kontrollvariable) bekannt ist, kann man sie in die ANOVA mit aufnehmen. Falls diese Drittvariable diskret ist und wenige Abstufungen hat, kann sie als weiterer Faktor aufgenommen werden (2F-ANOVA). Wenn aber die Drittvariable stetig ist oder sehr viele Abstufungen hat, müsste sie zuerst dichotomisiert werden, wodurch Informationen und damit Teststärke verloren ging. Besser ist in diesem Fall die Aufnahme der Drittvariable als Kovarianz in die ANOVA, die damit zur ANCOVA wird. Vom Prinzip her rechnet man dazu zunächst eine Regression mit der Drittvariablen als Prädiktor für das Kriterium AV. Die Residuen dieser Regression sind dann die AV für die folgende Varianzanalyse. Man sagt auch: Die Drittvariable wird aus der AV herauspartialisiert. Dadurch wird der Einfluss der Drittvariablen auf die AV praktisch nachträglich herausgerechnet und so „statistisch konstant gehalten“. Das Vorgehen entspricht so einer nachträglichen Parallelisierung (s. Bortz 1993:332-353).

Univariate Varianzanalysen
Dort werden beliebig viele unabhängige Variablen bzw. Gruppen in Hinblick auf nur eine abhängige variable untersucht werden oder multivariate Varianzanalysen (Multivariate Analysis of Variance, MANOVA), bei denen beliebig viele unabhängige Variablen bzw. Gruppen im Hinblick auf mehrere abhängige Variable untersucht werden (s. Bortz & Döring 2002:693).

7.6.2 Voraussetzungen für die Varianzanalyse

Die Anwendbarkeit der Varianzanalyse als statistisches Auswertungsverfahren ist an gewisse Voraussetzungen gebunden. Zunächst werden die allgemeinen Annahmen (Zufallstichprobe und Unabhängigkeit der Messungen), die für alle inferenzstatistischen Verfahren wichtig sind, erörtert.
Als nächstes wird auf das Skalenniveau eingegangen, das für eine bestimmte Gruppe von Verfahren (parametrische Auswerteverfahren) bedeutsam ist, um schließlich zu den eigentlichen Voraussetzungen der Varianzanalyse (Varianzhomogenität und Normalverteilung) zu kommen. Als letztes wird eine Voraussetzung behandelt, die für eine Untergruppe varianzanalytischer Auswertungen (Messwiederholungspläne) relevant ist, nämlich die Zirkularität.

Zufallsstichprobe
Fast alle statistischen Verfahren gehen von einer Zufallsstichprobe aus, die aus der Population gezogen wird. Die Ziehung einer Zufallsstichprobe ist unter folgendem Aspekt relevant:
Voraussetzung zur Konstruktion der Stichprobenkennwerteverteilung ist, dass die Konstruktion der Stichprobenkennwerteverteilung, die dann die Grundlage für die Signifikanzentscheidung bildet, von einer Zufallsstichprobe ausgeht. Ohne Kenntnis dieser Verteilung kann keine Entscheidung darüber getroffen werden, ob ein bestimmter Mittelwertunterschied überzufällig ist oder nicht. Um das Prinzip des Signifikanztests auch auf eine nicht-zufällig gezogene Stichprobe anwenden zu können, konstruiert man zu der gezogenen Stichprobe eine **hypothetische Grundgesamtheit**, aus der die untersuchte Stichprobe als Zufallsstichprobe angesehen wird. Für diese hypothetische Grundgesamtheit gilt dann die entsprechende Stichprobenkennwerteverteilung und ermöglicht so den statistischen Schluss, nämlich - im Fall eines signifikanten Ergebnisses - die Absicherung dieses Ergebnisses gegen eine Zufallserklärung. Damit ist gemeint, das es unwahrscheinlich ist, das der gefundene Gruppenunterschied rein zufällig, durch besonders extreme Gruppenzusammensetzung entstanden ist.

Unabhängigkeit der Messungen
Die Voraussetzung der Unabhängigkeit der Messwerte bzw. die Unabhängigkeit der Fehlerkomponenten bedeutet, dass der Einfluss von Störvariablen für jede Messung unabhängig vom Einfluss der Störvariablen jeder anderen Messung ist. Diese Voraussetzung gilt sowohl innerhalb der Stichproben als auch zwischen den Stichproben. Diese Annahme liegt dem Datenmodell der Varianzanalyse zugrunde und wird nicht weiter überprüft. Sie muss durch Maßnahmen bei der Versuchsplanung und -durchführung gesichert werden (indem

die Vpn einzeln aus der Population gezogen und dann randomisiert den Versuchsbedingungen zugewiesen werden).

Das Skalenniveau
Da die Varianzanalyse eine Aussage über Mittelwerte macht, folgt, dass sie nur auf Daten angewendet werden sollte, für die eine Aussage über Mittelwerte sinnvoll ist. Dies gilt für Messungen, die mindestens Intervallskalenniveau erreichen (vgl. Dorsch 1994:840). Häufig besteht aber Unsicherheit darüber, ob Werte auf einer Intervall- oder Ordinalskala liegen. Das Skalenniveau[16] lässt sich aber statistisch nicht ermitteln, sondern nur inhaltlich begründen. Die sinnvolle Anwendung bestimmter statistischer Prozeduren lässt sich auch im Rückschluss aus der sinnvollen Interpretierbarkeit der Ergebnisse ableiten: Auch wenn sich darüber diskutieren lässt, ob Schulnoten auf eine Intervall- oder Rangskala liegen, so ergibt die Aussagen, dass in Gruppe A die durchschnittliche Note besser ist als in Gruppe B einen inhaltlichen Sinn und rechtfertigt somit den Einsatz z.B. einer Varianzanalyse (wenn die übrigen Voraussetzungen gegeben sind).

Normalverteilung und Varianzhomogenität
Die mathematischen Voraussetzungen der Varianzanalyse sind, dass die Stichproben normalverteilten und varianzhomogenen Stichproben entstammen. Mit Varianzhomogenität ist gemeint, dass die Varianzen in den einzelnen Versuchsbedingungen sich nicht systematisch unterscheiden, denn nur unter diese Voraussetzung ist es gerechtfertigt, eine gemeinsame Fehlervarianz aus allen Bedingungen zu berechnen. Es gibt verschiedene Verfahren zur Überprüfung der Varianzhomogenität (z.B. Bartlett-Test), die aber wiederum bestimmt Probleme mit sich bringen. Sie sind ihrerseits wieder an bestimmte Voraussetzungen geknüpft (z.B. die de Normalverteilung) und reagieren auf Verletzungen teilweise sensibler als die Varianzanalyse selbst. Der Levence Test ist ebenfalls ein Verfahren zur Überprüfung der Varianzhomogenitätsvoraussetzung, der nicht so empfindlich auf die Verletzung der Normalverteilungsannahme reagiert. Beim Levence Test bedeutet ein signifikantes Ergebnis („Signifikanz“ < .05), dass die Varianzen in den einzelnen Gruppen nicht gleich sind, ein hoher Wert dagegen, dass die Varianzen sich nicht stark unterscheiden. Man ist in der Regel an einem nicht signifikanten Ergebnis interessiert (Bortz 1993:261ff).

[16] Auf eine Behandlung der wichtigsten Skalenarten wird auf Bortz und Döring (2002:70ff) und Benninghaus (2002:20ff) verwiesen.

Zirkularität (bei Varianzanalyse mit Messwiederholung)
Für die Varianzanalyse mit Messwiederholung gilt eine weitere Voraussetzung, nämlich die sogenannte „Zirkularität“ oder „Sphärizität“ Damit ist gemeint, dass die Varianzen der Differenzen der Messwerte einer Person zwischen allen Bedingungen gleich sein müssen.

8 Auswertung von Retrieval Tests

Information-Retrieval-Systeme können hinsichtlich Effektivität und/oder Effizienz evaluiert werden, die aus zwei Arten von Gütekriterien basieren. Bei der Effektivität wird die Güte des Systems gemessen, also dessen Fähigkeit, relevante Dokumente zu selektieren und irrelevante zurückzuhalten. Bezüglich der Effizienz werden Kosten-Nutzen-Faktoren wie Systemressourcen, Antwortzeiten etc. betrachtet (s. Womser-Hacker 1989:23). Auf den Aufbau von Retrieval Tests aus experimenteller Sicht wird hier auf Womser-Hacker (1989: 23f) verwiesen. Für den Retrievaltest werden pro Fragestellung und getestetem System die folgenden Basiszahlen ermittelt (s. Tabelle 6).

Tabelle 6: Elementarparameter (nach Womser-Hacker 1990:52)

Bewertung System	relevant	nicht relevant	Summe
nachgewiesen	a	b	L
nicht nachgewiesen	c	d	L*
Summe	C	C*	N

Die nachgewiesenen, relevanten Dokumente heißen „Treffer“ (a), die nachgewiesenen, nichtrelevanten Dokumente „Ballast“ (b). Nicht nachgewiesene, jedoch relevante Dokumente werden als „vermisste relevante“ oder „sillence“ bezeichnet (c), während die nicht-nachgewiesenen und nicht relevanten „umgangene Dokumente“ (d) heißen.

Die Werte werden in eine Datenbank oder Exceltabelle eingetragen, die neben den Angaben aus der Tabelle die folgende Inhalte enthält:

- Zahl der bearbeiteten Fragen
- Zahl der gefundenen Dokumente
- Zahl und Dokumentennummer der relevanten und nicht-relevanten Dokumente
- Zahl und Dokumentennummern mit unbekannten Relevanzurteil

- Precision je Frage
- Recall je Frage

Aufgrund dieser Fragen, werden die nachfolgenden Auswertungen vorgenommen.

8.1 Recall und Precision

Recall und Precision sind die grundlegenden Kennzahlen des Information Retrieval. Recall und Precision als abhängige Variablen weisen ein metrisches[17] Skalenniveau auf. Der Recall ist definiert als das Verhältnis zwischen nachgewiesenen relevanten Dokumenten und im Dokumentenbestand vorhandenen relevanten Dokumenten (Womser-Hacker 1990:53):

$$r = \frac{a}{a+c}$$

Anmerkungen:
r = Recall
a = Anzahl der nachgewiesenen relevanten Dokumente
c = Anzahl der im Dokumentenbestand enthaltenen relevanten, aber nicht nachgewiesenen Dokumente

Der Wertebereich des Recall geht von 0 bis 1 ($0 \leq r \geq\_ 1$). Ein Recall von 0 wird für das schlechteste Ergebnis, 1 für das bestmögliche vergeben. Womser-Hacker (1990:53) kritisiert am Recall die folgenden zwei Punkte:

- Der Recall bezieht die Ballastquote nicht mit ein und reicht somit nicht zur Bewertung der Retrievalergebnisse aus.
- Bei größerem Dokumentenbestand ist eine Bewertung aller Dokumente bezüglich ihrer Relevanz nicht möglich, daher muss ein Annäherungswert (= c') verwendet werden.

Die Precision ergänzt den Recall dadurch, dass sie als Indikator für die Fähigkeit eines System gilt, irrelevante Dokumente herauszufiltern. Die Precision drückt aus, welcher Anteil der tatsächlich gefundenen Dokumente für die Fragestellung relevant ist.

$$p = \frac{a}{a+b}$$

Anmerkungen:

[17] Der theoretische Hintergrund metrischer Skalen wird z.B. in Schwarze (1994:34ff) beschrieben.

p = Precision
a = Anzahl der nachgewiesenen relevanten Dokumente
b = Anzahl der nachgewiesenen nicht relevanten Dokumente

Anhand beider Messzahlen wird versucht, die Qualität der zu testenden Systeme zu beurteilen.

8.2 Mittelwertbildung

Im Bereich Information Retrieval werden zur Mittelwertbildung von Recall und Precision zwei Methoden unterschieden (Womser-Hacker 1990: 56):

- „Bei der *Mikrobewertung* addiert man die Grundparameter des Retrievalergebnisses (=Anzahl der relevanten und irrelevanten Antwortdokumente) über den gesamten Aufgabenbestand des Tests; dann erfolgt auf dieser Basis die verallgemeinernde Berechnung des entsprechenden Bewertungsmaßes
- Bei der *Makrobewertung* wird das gewählte Bewertungsmaß für jedes Retrievalergebnis (d.h. für jede Aufgabe) getrennt berechnet; anschließend werden diese Werte arithmetisch gemittelt."

8.3 Signifikanztests

Die hier untersuchten „fiktiven Hypothesen" sind mit Unterstützung der Statistik – Software SPSS 11.0 für Windows (s. Bühl & Zöfel 2000) berechnet worden. Als Grundlage zur Berechnung diente der Datensatz aus der Vergleichsuntersuchung Messenger-Fulcrum (Binder et al 2000). Die Hintergrundinformationen zur Vergleichsuntersuchung sind dem Arbeitsbericht zu entnehmen. Das Ziel der folgenden statistischen Untersuchung ist zu überprüfen, ob sich Versuchspersonen mit boolesche Erfahrung von denen ohne boolesche Erfahrung hinsichtlich Recall signifikant voneinander unterscheiden. Recall wies als zentrale abhängige Variable ein metrisches Skalenniveau auf. Die unabhängige Variable boolesche Erfahrung wies hingegen Nominalskalenniveau auf. Als Verfahren bot sich hier die einfaktoriellen Varianzanalyse (s Kapitel 7.6.1) an, die Vergleiche auf der Grundlage der Mittelwerte anstellt[18].

[18] Eine analoge Methode für den Vergleich von Mittelwerten bei unabhängigen Variablen mit zwei Gruppen ist der t-Test (s. Bortz 1993: Kapitel 5). Die ANOVA ist eine Variante davon, die auch mehr Gruppen zulässt.

Zur Berechnung einer einfaktoriellen ANOVA wurden im Menü von SPSS 11.0 für Windows folgende Schritte durchgeführt:

- Analysieren
- Mittelwerte vergleichen
- Einfaktorielle ANOVA ...
- Tragen Sie die Variable Boole-Erfahrung als Faktor ein.
- Erklären Sie in der erscheinenden Dialogbox Recall als abhängige Variable.
- Wählen Sie unter „Optionen ...“ die Erstellung deskriptiver Statistiken und die Überprüfung auf Varianzhomogenität.
- Starten Sie die Berechnungen mit *OK*.

Die wichtigsten Ergebnisse aus dem Viewer von SPSS11.0 für Windows werden hier kurz erläutert:

Tabelle 7: Test der Homogenität der Varianzen, Recall (AV)

Levene-Statistik	df1	df2	Signifikanz
,280	1	142	,597

Der Levene Test dient zur Überprüfung auf Varianzhomogenität. Üblicherweise verwirft man die Gleichheit der Varianzen (Varianzhomogenität) falls der Levene Test ein p kleiner 0.05 ergibt. Die gegebenen Fallgruppen unterscheiden sich nicht signifikant voneinander, da das Signifikanzniveau (Irrtumswahrscheinlichkeit[19]) hier 0.597 beträgt (s. Tabelle 7).

Somit ist eine weitere Voraussetzung zur Berechnung der ANOVA erfüllt.

Tabelle 8: ONEWAY ANOVA, Recall nach boolescher Erfahrung

	Quadratsumme	df	Mittel der Quadrate	F	Signifikanz
Zwischen den Gruppen	,024	1	,024	,388	,535
Innerhalb der Gruppen	8,903	142	,063		
Gesamt	8,927	143			

[19] Zur Erläuterung der Irrtumswahrscheinlichkeit sei auf Bühl und Zöfel (2000:109ff) verwiesen.

Tabelle 8 ist zu entnehmen, dass zwischen den beiden Gruppen (boolesche Erfahrung / ohne boolesche Erfahrung) kein signifikanter Unterschied besteht ($F_{(1,141)}$=.39 , p >.535)

Folglich hat die boolesche Erfahrung keinen Einfluss auf den Recall.

9 Literatur

Backhaus, K. (Hrsg.). (1989). Multivariate Analysemethoden. Eine anwendungsorientierte Einführung. Berlin: Springer

Baumgartner, P. (1999). 10 Testmethoden in der Evaluation interaktiver Lehr- und Lernmedien. In: K. Lehmann (Hrsg.), Studieren 2000- Alle Inhalte in neuen Medien?. Münster: Waxmann Verlag

Benninger, H. (2002). Deskriptive Statistik. Eine Einführung für Sozialwissenschaftler (9. überarb. Aufl.) Wiesbaden: Westdeutscher Verlag

Bevan, N. & Macleod, M. (1994). Usability measurement in context. Behavior and Information technology 13 (S. 132-145)

Bevan, N. & Curson, I. (1997). Methods of Measuring Usability. In: Proceedings of the sixth IFIP conference on human-computer interaction, Sydney, Australia, July 1997

Bevan, N. (1997). Quality in Use: Incorporating Human Factors into the software engineering lifecycle. In: Proceedings of the Third International Symposium and Forum on Software Engineering Standards, ISESS'97 conference, August 1997

Binder, G., Stahl, M., Faulborn, L. (2000). Projektbericht. Vergleichsuntersuchung MESSENGER-FULCRUM. IZ-Arbeitsbericht Nr. 18. Bonn: Informationszentrum Sozialwissenschaften. Verfügbar über: [Zugriffsdatum: 03.02.03] http://www.gesis.org/Publikationen/Berichte/IZ_Arbeitsberichte/pdf/ab18.pdf

Bortz, J. (Hrsg.). (1993). Statistik für Sozialwissenschaftler. (4 Aufl.). Berlin: Springer

Bortz, J. & Döring, N. (2002). Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler (3., überarb. Aufl.). Berlin: Springer

Bortz, J. & Lienert, G. A. (1998). Kurzgefasste Statistik für die klinische Forschung. Ein praktischer Leitfaden für die Analyse kleiner Stichproben. Berlin [u.a.]: Springer

Bühl, A. & Zöfel, P.: (2000) SPSS Version 10 .Einführung in die moderne Datenanalyse unter Windows. München: Addison-Wesley

Büning, H., Trenkler, G. (1994). Nichtparametrische statistische Methoden (2., erw. u. völlig überarb. Aufl.). Berlin [u.a.]: de Gruyter

CI (1994) Usability engineering now! [www document]. Abrufbar über [Zugriffsdatum: 09.01.03]: http://www.ipo.tue.nl/homepages/mrauterb/publications/COMPUSAB94paper.pdf

DATech (2001). DATech – Prüfhandbuch Gebrauchstauglichkeit. Leitfaden für die softwareergonomische Evaluierung von Software auf Grundlage von DIN EN ISO 9241, Teile 10 und 11. Deutsche Akkreditierungsstelle Technik e.V., Frankfurt/Main

Dick, M. (2000). Die Anwendung narrativer Gridinterviews in der psychologischen Mobilitätsforschung. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [Online Journal], 1 (2). Verfügbar über: [Zugriffsdatum: 20.09.02] http://www.qualitative-research.net/fqs-texte/2-00/2-00dick-d.pdf

Dorsch, F. (Hrsg.). (1994). Psychologisches Wörterbuch. (12. überarbeitete und erweiterte Auflage). Bern [u.a.]: Huber

Dzida, W., Hoffmann, B., Freitag, R., Redtenbacher, W., Baggen, R., Geis, T., Beimel, J., Hurheiden, C., Hampe-Neteler, W., Hatwig, R., Peters, H. (2000) . Gebrauchstauglichkeit von Software. ErgoNorm: ein Verfahren zur Konformitätsprüfung von Software auf der Grundlage von DIN EN ISO 9241 Teile 10 und 11. Schriftreihe der Bundeszentrale für Arbeitsschutz und Arbeitsmedizin. Forschung F1693, Dortmund

Fisseni, H.J. (Hrsg.) (1997). Lehrbuch der psychologischen Diagnostik. Mit Hinweisen zur Intervention (2., überarb. und erw. Aufl.). Göttingen [u.a.]: Hogrefe

Frieling, E. & Sonntag, K. (Hrsg.). (1999). Lehrbuch Arbeitspsychologie (2., vollst. überarb. und erw. Aufl.). Bern [u.a.] : Huber

Glaser, B.G. & Strauss, A.L. (1967). The discovery of grounded theory. Strategies for qualitative research. New York: de Gruyter

Glaser, B. G. & Strauss, A. (1998). Grounded Theory. Strategien qualitativer Forschung. Bern: [u.a.] Huber.

Görner, C., Ilg, R. (1993). Evaluation der Mensch-Rechner-Schnittstelle. In: Ziegler, J., Ilg, R. (Hrsg.): Benutzergerechte Software-Gestaltung. Standards, Methoden und Werkzeuge. München: Wien: Oldenbourg

Hackman, G. S. & Biers, D. W. (1992). Team usability testing: Are two heads better than one? Proceedings oft the 36 th annual meeting of the Human Factors society. 36, S.1205-1209

Hamborg, K.C., Hassenzahl, M. & Wessel, R. (n.d.). Workshop: Gestaltungsunterstützende Methoden für benutzer-zentrierte Softwareentwicklung [www document]. Abrufbar über: [Zugriffsdatum: 20.09.02] http://www.tu-darmstadt.de/fb/fb3/psy/soz/veroeffentlichungen_mh/Workshopm_c_eingerecht.pdf

Hampel, R. (1977). Adjektiv-Skalen zur Einschätzung der Stimmung (SES). Diagnostica, 23, 43-60

Hasebrook, J. (Hrsg.) (1995). Multimedia-Psychologie: eine neue Perspektive menschlicher Kommunikation. Heidelberg [u.a.]: Spektrum, Adad. Verlag

Honold, P. (2000). Interkulturelles Usability Engineering. Eine Untersuchung zu kulturellen Einflüssen auf die Gestaltung und Nutzung technischer Produkte.. Düsseldorf: VDI Verlag GmbH.

Holz auf der Heide, B. (1993). Welche software-ergonomischen Evaluationsverfahren können was leisten? In: K.-H. Rödiger : Von der Benutzungsoberfläche zur Arbeitsgestaltung: [gemeinsame Fachtagung des German Chapter of the ACM]

Huber, O. (2002). Das Psychologische Experiment: Eine Einführung. (3.Auflage). Bern [u.a.] : Huber

Irion, T. (2002). Einsatz von Digitaltechnologien bei der Erhebung, Aufbereitung und Analyse multicodaler Daten. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [Online Journal], 3 (2). Verfügbar über: [Zugriffsdatum: 18.09.02] http://www.qualitative-research.net/fqs-texte/2-02/2-02irion-d.htm

EN ISO 9241-10 (1995). Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Teil 10: Grundzüge der Dialoggestaltung

ISO DIN 9241-11(1995). Ergonomic requirements for office work with display terminals (VDTs): Guidance on usability. Berlin:Beuth

Jeffries, R., Miller, J.R., Wharton, C. & Uyeba, K.M. (1991). User interface evaluation in the real world: a comparison of four techniques, Procedings of ACM CHI'91 conference on Human Factors in Computing Systems, 119- 124 (Association for Computing Machinery, New York)

Karat, J. (1997). User-Centered Software Evaluation Methodologies. In: Helander, M. G., Landauer, T.K., Prabhu, P. (Hrsg.), Handbook of Human-Computer Interaction. Amsterdam: Elsevier Science B.V.

Karat, C.-M., Campell, R.L., Fliegel, T. (1992). Comparison of empirical testing in user interface evaluation. Proccedings ACM CHI'92 Conference. Monterey CA. New York: ACM, S. 397-404

Kirakowski, J. & Corbett, M. (1993). SUMI: the Software Usability Measurement. British Journal of Educational Technoloy. 24 (3) S. 210-212

Kirakowski, J. (1995). The software usability measurement inventory: background and usage. In: P. Jordan, B. Thomas, and B. Weerdmeester (eds.): Usability Evaluation in Industry. Taylor & Francis: London

Kromrey, H. (1995). Empirische Sozialforschung. Modelle und Methoden der Datenerhebung und Datenauswertung (7. Auflage). Opladen: Leske + Budrich

Lin, H. X., Choong, Y., Salvendy, G. (1997). A proposed index of usability: a method for comparing the relative usability of different software systems. Behavior and Information Technology 16 (4/5) 267-278

Lisch, R., & Kriz, J. (1978). Grundlagen und Modelle der Inhaltsanalyse. Bestandsaufnahme und Kritik. Reinbek: Rowohlt

Lewis, C. & Wharton, C. (1997). Cognitive Walkthroughs. In: Helander, M. G., Landauer, T.K., Prabhu, P. (Hrsg.), Handbook of Human-Computer Interaction. Amsterdam: Elsevier Science B.V.

Mekelburg, H.-G. (2002). hgm's Reisen durch den Cyberspace. Der Recherchekompass. [www document]. Abrufbar über: http://home.nordwest.net/hgm/index.html [Zugriffsdatum: 10.09.02]

Mayring, P. (Hrsg.). (1996). Einführung in die qualitative Sozialforschung. (3. Aufl.). Weinheim: Deutscher Studienverlag

Myers, G. J. (1999). Methodisches Testen von Programmen. (6.Aufl.) München: Wien: Oldenbourg

Natt och Dach, J., Regnell, B., Madsen, O.S., Aurum, A. (2001). An Industrial Case Study of Usability Evaluation in Marked Driven Package Software Development

Nielsen, J. (1994). Heuristic evaluation, in: J. Nielsen & R.L. Mack (Hrsg.). Usability inspection methods, New York: John Wiley

Nielsen, J. (1993). Usability Engineering. San Francisco, Calif. : Kaufmann

Nielsen, J. (n.d.). useit.com: Jakob Nielsen's Website. [www document] Abrufbar über: http://www.useit.com/ [Zugriffsdatum: 10.09.02]

Oppermann, R., Reiterer, H. (1994).Software-ergonomische Evaluation. In: Eberleh, E. Oberquelle, H. & Oppermann, R. Einführung in die Software-Ergonomie. Gestaltung graphisch-interaktiver Systeme: Prinzipien, Werkzeuge, Lösungen. Berlin: New York: Walter de Gruyter

Rautenberg, M (1991). Benutzungsorientierte Benchmark Tests: eine Methode zur Benutzerbeteiligung bei de Entwicklung von Standardsoftware. In: Projektberichte zum Forschungsbericht Benutzungsorientierte Softwareentwicklung und Schnittstellengestaltung (BOSS) Nr.6 (Spinas, P. ; Rautenberg, M.; Strohm, O.; Waeber, D. & Ulich, E; Hrsg.); ETH-Zürich; Institut für Arbeitspsychologie

Roth, E. (1994). Sozialwissenschaftliche Methoden : Lehr- und Handbuch für Forschung und Praxis. München: Wien: Oldenbourg

Rubin, J. (1994). Handbook of Usability Testing- How to Plan, Design and Conduct Effective tests. New York: Wiley

Saretz, S. (1969). Theoretische Überlegungen und empirische Untersuchungen für die Entwicklung einer Skala zur Selbsteinschätzung der augenblicklichen Stimmungslage. Unveröff. Dipl.-Arbeit, Freiburg

Schmid, H. (1992). Psychologische Tests: Theorie und Konstruktion. Freiburg, Schweiz: Univ.-Verl.; Bern [u.a.]: Huber

Scriven, M (1980).The Logic of Evaluation. Invernetss, CA: Edgepress

Schnell, R. Hill, P., Esser, E. (1989). Methoden der empirischen Sozialforschung (2., überarb. und erw. Aufl.). München: Wien: Oldenbourg

Schwarze, J. (1994). Grundlagen der Statistik I. Beschreibende Verfahren (7 Aufl.) Berlin: Neue Wirtschafts-Briefe

Strauss, A. & Corbin, J. (1996). Grounded Theory: Grundlagen Qualitativer Sozialforschung. Weinheim: Beltz Psychologie Verlags Union.

Ulich, E. (1986). Aspekte der Benutzerfreundlichkeit. In W. Remmele, M. Sommer (Hrsg.) Arbeitsplätze morgen.: [gemeinsame Fachtagung des German Chapter of the ACM]. Stuttgart: B.G. Teubner.

Ulich, E. (2001). Arbeitspsychologie (5., vollst. überarb. und erw. Aufl.). Zürich: vdf, Hochschulverlag, an der ETH Zürich; Stuttgart: Schäfer-Poeschel.

Urbanek, W. (1991). Software-Ergonomie und benutzerangemessene Auswahl von Werkzeugen bei der Dialoggestaltung. Berlin: de Gruyter

Vorberg, D., Blankenberger, S. (1999). Die Auswahl statistischer Tests und Maße. Psychologische Rundschau, 50 (3) 157-164. Göttingen [u.a.]: Hogrefe.

Wandmacher, J. (1993). Software-Ergonomie. Mensch-Computer-Kommunikation. Grundwissen: 2. Berlin: New York: Walter de Gruyter

Womser-Hacker, Ch. (1989). Der PADOK-Retrievaltest. In: Hellwig, P., Krause, J. (Hrsg.) Sprache und Computer. Hildesheim [u.a.]: Georg Olms Verlag.

Womser-Hacker, Ch. (1990). Die statistische Auswertung des Retrievaltests. In. Krause, J., Womser-Hacker, Ch. (Hrsg.) Das Deutsche Patentinformationssystem. Entwicklungstendenzen, Retrievaltests und Bewertungen. Köln [u.a.]: Carl Heymanns Verlag KG

Wottawa, H., Thierau, H. .(Hrsg.). (1998). Lehrbuch Evaluation. Bern [u.a.]: Huber

Wottawa (2001): Evaluation. In Weidemann, B., Krapp, A.: Pädagogische Psychologie (4., vollst. überarb. Aufl). Weinheim : Beltz PVU

Zimbardo, P.G. (1995). Psychologie (6., neu bearb. und erweitere Aufl). Berlin [u.a.]: Springer