

9 Software-ergonomische Evaluation

Reinhard Oppermann & Harald Reiterer

9.1 Wieso gewinnt die software-ergonomische Evaluation immer mehr an Bedeutung?

Immer mehr Arbeitsplätze werden vom Einsatz von EDV-Systemen berührt. Dadurch wird für eine steigende Zahl von Berufstätigen die Benutzung eines EDV-Systems bei ihrer Aufgabenerfüllung zum Bestandteil ihrer täglichen Arbeit. Prognosen für die Industrieländer gehen davon aus, daß im Jahr 2000 ca. 90% der dort arbeitenden Bevölkerung vom Einsatz der EDV-Technologie – direkt oder indirekt – betroffen sein wird /Fährlich 87/. Nur eine Minderheit der Berufstätigen werden aber von ihrer Ausbildung dazu qualifiziert sein (EDV-Laien). Neben entsprechenden Aus- und Weiterbildungsmaßnahmen stellt daher für diese Berufsgruppe die einfache Benutzung der EDV-Systeme eine wesentliche Voraussetzung für eine erfolgreiche Bewältigung ihrer Aufgaben dar. Vor diesem Hintergrund gewinnt die Forderung nach software-ergonomisch gestalteten Benutzerschnittstellen – unter dem Schlagwort „Benutzerfreundlichkeit“ – zunehmend an Bedeutung. Diese Notwendigkeit zu einer vereinfachten Benutzung wurde von vielen Anbietern von Hard- und Software erkannt und ist gleichzeitig auch eine der Hauptursachen für die derzeit stattfindende massive Umstellung bzw. Neueinführung objekt-orientierter, graphischer und direkt-manipulativer Benutzungsoberflächen.

Dies stellt die Entwickler von Software, insbesondere diejenigen die mit der Realisierung der Benutzerschnittstelle befaßt sind, vor neue und anspruchsvolle Herausforderungen. Sie werden mit einer Reihe von – oft widersprüchlichen – Anforderungen konfrontiert und benötigen zu deren Bewältigung fundiertes Wissen im Bereich der Gestaltung und Evaluation von Benutzerschnittstellen. Dabei spielt die Auswahl der geeigneten Methoden und Werkzeuge eine wichtige Rolle. Im Bereich der Gestaltung gewinnen User Interface Management Systems (UIMS) immer mehr an Bedeutung. Es handelt sich dabei um leistungsfähige Entwicklungswerkzeuge für Benutzerschnittstellen. Deren Einsatz allein garantiert aber noch lange keine „benutzerfreundlichen“ Schnittstellen. Dazu bedarf es umfassenden software-ergonomischen Wissens, das den Entwicklern in Form von Normen, Gestaltungsrichtlinien und Style Guides (beispielsweise OSF/Motif Style Guide /OSF/MOTIF 90/, Open Look Style Guide /SUN 89b/, Common User Access Style Guide /IBM 89a, 89b, 91a, 91b/, Apple Human Interface Guide-

lines/Apple 87/, vgl. auch Eberleh, in diesem Band) zur Verfügung gestellt wird. Dem Bereich der Evaluation kommt im Entwicklungsprozeß die Aufgabe der Qualitätssicherung zu. Es wird evaluiert, inwieweit die gestellten Anforderungen auch tatsächlich im Entwicklungsprozeß umgesetzt wurden. Für Zwecke der software-ergonomischen Evaluation gibt es heute eine breite Palette von Methoden und Werkzeugen. Sie reicht von einfachen Checklisten bis hin zu aufwendigen Untersuchungslabors. In der Praxis besteht vor allem ein Bedarf an einfach und schnell einzusetzenden Evaluationsmethoden, während im Forschungsbereich aufwendige experimentelle Methoden zur Überprüfung von Hypothesen und zur Theoriebildung im Bereich der Mensch-Computer-Kommunikation im Vordergrund stehen.

Ein weiterer wichtiger Grund für die zunehmende Bedeutung von software-ergonomischen Evaluationsverfahren ist der sich schrittweise etablierende Gemeinsame Europäische Markt (bestehend aus EU und EFTA). Um in diesem übernationalen Wirtschaftsraum einheitliche Arbeitsbedingungen für Benutzer von Bildschirmgeräten zu schaffen, wurde von der EG-Kommission eine „Richtlinie über die Mindestvorschriften bezüglich der Sicherheit und des Gesundheitsschutzes bei der Arbeit an Bildschirmgeräten“ verabschiedet /EWG 90/. Bis zum 31. Dezember 1992 hatten die nationalen Regierungen der einzelnen EG-Mitgliedsstaaten Zeit, diese Richtlinie in nationales Recht umzusetzen. In diesem Umsetzungsprozeß spielt die internationale und europäische Normung eine wichtige Rolle. Insbesondere die ISO-Norm 9241 „Ergonomic Requirements for Office Work with Visual Display Terminals“ ist hier von großer Bedeutung /ISO 89/, /ISO 91a, 91b, 91c, 91d, 91e, 91f, 92a, 92b/. In ihr werden eine Vielzahl von hard- und software-ergonomischen Forderungen festgeschrieben, die zukünftig von EDV-Systemen im Bürobereich zu erfüllen sind. Die ISO-Norm soll gleichzeitig auch als europäische Norm (CEN 29241) verabschiedet werden. Diese Norm wird in Software-Spezifikationen und Pflichtenheften ihren Niederschlag finden. Dies bedeutet für die Entwickler von Software, daß sie in der Lage sein müssen, die Gestaltungsforderungen dieser Norm zu erfüllen sowie deren Einhaltung im Rahmen der Qualitätssicherung zu kontrollieren. Dazu benötigen sie praktisch einsetzbare Evaluationsverfahren (z.B. Prüflaufpläne, Evaluations-Software). Für die Käufer von Software bedeuten die Normen, daß sie in der Lage sein müssen, deren Einhaltung zu überprüfen. Dazu benötigen auch sie einfach einzusetzende Evaluationsverfahren als Prüfinstrumente.

Die obigen Sachverhalte erklären das stark steigende Interesse an praktisch einsetzbaren Entwicklungswerkzeugen für die ergonomische Gestaltung von Benutzerschnittstellen und an Evaluationsverfahren zur Überprüfung deren ergonomischer Qualität.

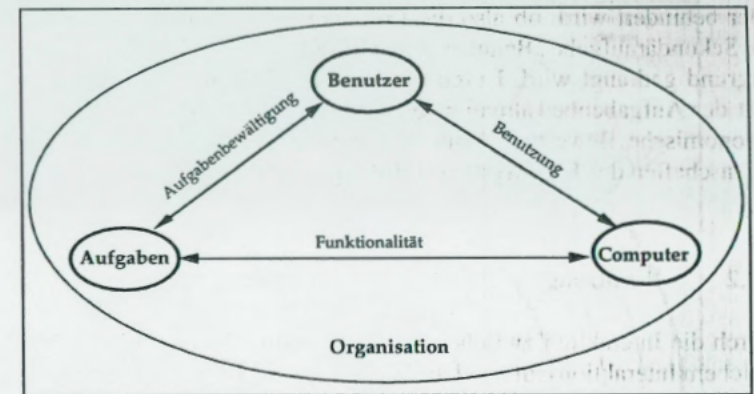


Abb. 9-1: Elemente und Beziehungen

9.2 Welche Elemente und Kriterien sind bei der Evaluation zu berücksichtigen?

Generell geht man heute im Bereich der Software-Ergonomie davon aus, daß bei der Gestaltung und Bewertung von EDV-Systemen die in Abb. 9-1 gezeigten Elemente sowie deren Beziehungen zueinander zu berücksichtigen sind /Frese, Brodbeck 89/. Vorgehensweisen des Software-Engineering und der Evaluation, die all diese Elemente und Beziehungen berücksichtigen, werden vielfach als ganzheitlich bezeichnet.

Die Ziele einer ganzheitlichen Evaluation lassen sich am besten an den oben dargestellten Beziehungen zwischen den zu bewertenden Elementen erklären:

9.2.1 Aufgabenbewältigung

Durch die Gestaltung der Beziehung zwischen Benutzer und Aufgabe wird 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 wesentliche 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-ergonomische Bewertung keine Verengung des Blicks nur auf Schnittstelleneigenschaften des EDV-Systems erfolgt („Schnittstellenkosmetik“).

9.2.2 Benutzung

Durch die 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, wie beispielsweise: „Welchen Aufwand muß der Benutzer zum Erlernen treiben?“ oder „Welche Möglichkeiten der individuellen Anpassung des EDV-Systems an seinen Arbeitsstil und seine Persönlichkeit werden ihm geboten (vgl. den Beitrag von Oppermann in diesem Band)?“. Die ergonomische Qualität der Benutzung ist zentraler Bewertungsgegenstand der software-ergonomischen Evaluation.

9.2.3 Funktionalität

Inwieweit das EDV-System aufgabenrelevant und aufgabenangemessen ist, wird durch das Ausmaß der Unterstützung der Aufgaben durch das EDV-System bestimmt. Darunter wird die Funktionalität des EDV-Systems verstanden. 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.

Aus einer ganzheitlichen Betrachtungsweise folgt auch eine spezifische Richtung bei der Gestaltung und Bewertung, wie dies in Abb. 9-2 verdeutlicht wird.

Zuerst ist der organisatorische Bereich zu gestalten bzw. zu bewerten. Dazu finden sich in der ISO 9241 Teil 2 eine Reihe von wünschenswerten Aufgabencharakteristika. Diese Charakteristika haben in einer Reihe von arbeitswissenschaftlichen Kriterien der Organisations- und Aufgabengestaltung ihren Niederschlag gefunden /Reiterer 90/.

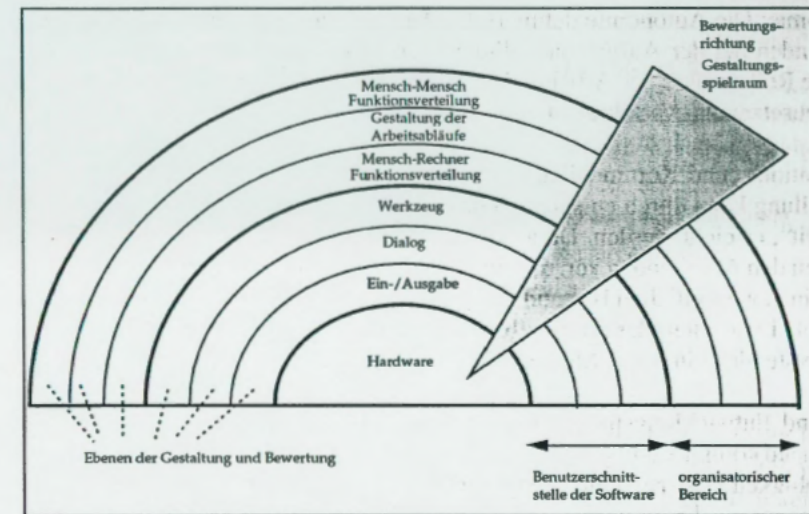


Abb. 9-2: Richtung und Ebenen der Evaluation

Anforderungsvielfalt: Die Anforderungsvielfalt einer Aufgabe soll sowohl den Einsatz unterschiedlicher Fähigkeiten, Kenntnisse und Fertigkeiten ermöglichen als auch die Möglichkeit zur körperlichen Aktivität bei der Erledigung der Aufgabe eröffnen.

Ganzheitlichkeit: Die Ganzheitlichkeit einer Aufgabe soll sicherstellen, daß der Arbeitende ein „ganzes Stück“ Arbeit vollendet und nicht (für ihn) sinnlose Einzelteile einer Aufgabe bearbeiten muß. Dadurch soll der Arbeitende den Bedeutungsgehalt und Stellenwert seiner Aufgabe im betrieblichen Arbeitsablauf klar erkennen können (Aufgabenidentität und -relevanz).

Durchschaubarkeit: Die Durchschaubarkeit einer Aufgabe soll es ermöglichen, daß der Arbeitende die technischen Bedingungen und Zusammenhänge, die organisatorischen Strukturen (der raum-zeitliche und inhaltliche Zusammenhang seiner Arbeitsaufgabe), die Ereignisse im Arbeitsprozeß und die Folgen des eigenen Eingreifens in den Arbeitsprozeß durchschauen und vorhersehen kann.

Rückmeldungen (Feedback): Rückmeldungen im Zuge der Aufgabenerfüllung bewirken, daß der Arbeitende Meldungen über den laufenden Fortgang seines Arbeitsablaufes (Ablauf-Feedback) und das Resultat seiner Arbeit (Resultat-Feedback) erhält.

Autonomie: Die Autonomie definiert das Ausmaß an (Selbst)Kontrolle, das dem Arbeitenden bei der Aufgabenerfüllung zugestanden wird. Der Arbeitende soll über die Reihenfolge der Arbeitsschritte, das Tempo der Aufgabenerfüllung und die einzusetzenden Verfahren angemessen entscheiden können.

Kooperations- und Kommunikationsförderlichkeit: Kooperation bei der Aufgabenerfüllung kann durch entsprechende Organisationskonzepte, wie z.B. „Gruppenarbeit“, erreicht werden. Derartige Konzepte sehen eine Aufgabenverteilung zwischen den Arbeitenden vor, die eine gemeinsame Aufgabenerfüllung ermöglichen. Einen wesentlichen Bestandteil dieser Kooperation bildet die Kommunikation, wobei vor allem der unmittelbaren persönlichen Kommunikation zwischen den Arbeitenden ein hoher Stellenwert zukommt.

Lern- und Entwicklungsmöglichkeiten: Dies setzt Arbeitsaufgaben voraus, die ausreichend komplex und gestaltbar sind, so daß sie es dem Arbeitenden erlauben, seine Fähigkeiten, Fertigkeiten und Kenntnisse weiterzuentwickeln. Dazu müssen die vorhandenen Erfahrungen und Fertigkeiten der Arbeitenden berücksichtigt sowie die notwendigen qualifikatorischen Voraussetzungen geschaffen werden, die eine Erfüllung der Arbeitsaufgabe ermöglichen.

Belastung und Beanspruchung: Der Arbeitende soll bei der Aufgabenerfüllung weder unterfordert werden (z.B. durch fachlich und inhaltlich wenig anspruchsvolle Aufgaben) noch soll er überfordert werden (z.B. durch Zeitdruck oder fachliche Überforderung). Als belastungsmindernd erweist es sich, wenn der Arbeitende die Möglichkeit hat, zwischen anspruchsvolleren Aufgaben auch weniger anspruchsvollere (z.B. Routineaufgaben) durchzuführen.

In einem nächsten Schritt ist die eigentliche Software zu gestalten bzw. zu bewerten. Vom ergonomischen Standpunkt kommt dabei der Benutzerschnittstelle eine hervorgehobene Rolle zu, da sie das „Fenster“ des Benutzers zur eigentlichen Anwendung darstellt. In der ISO 9241 Teil 10 /ISO 91b/ finden sich eine Reihe von Kriterien (dialogue principles), deren Umsetzung eine benutzerfreundliche Mensch-Rechner-Interaktion ermöglichen soll. Im folgenden wird ein Überblick über diese Kriterien gegeben, wobei auch eine Zuordnung zu den Kriterien der entsprechenden DIN 66 234 Teil 8 erfolgt (s. auch Tabelle 9-1).

Aufgabenangemessenheit, Suitability for the task: „Ein Dialog ist aufgabenangemessen, wenn er die Erledigung der Arbeitsaufgabe des Benutzers unterstützt, ohne ihn durch Eigenschaften des Dialogsystems unnötig zu belasten“ /DIN 88, S. 2/.

Selbstbeschreibungsfähigkeit, Self-descriptiveness: „Ein Dialog ist selbstbeschreibungsfähig, wenn dem Benutzer auf Verlangen Einsatzzweck sowie Lei-

Tab. 9-1: Kriterien nach ISO und DIN

ISO-Norm 9241 Part 10	DIN 66234 Teil 8
Suitability for the task	Aufgabenangemessenheit
Self-descriptiveness	Selbstbeschreibungsfähigkeit
Controllability	Steuerbarkeit
Conformity with user expectations	Erwartungskonformität
Error tolerance	Fehlerrobustheit
Suitability of individualization	
Suitability for learning	

stungsumfang des Dialogsystems erläutert werden können und wenn jeder einzelne Dialogschritt unmittelbar verständlich ist oder der Benutzer auf Verlangen dem jeweiligen Dialogschritt entsprechende Erläuterungen erhalten kann“ /DIN 88, S. 2/.

Steuerbarkeit, Controllability: „Ein Dialog ist steuerbar, wenn der Benutzer die Geschwindigkeit des Ablaufs sowie die Auswahl und Reihenfolge von Arbeitsmitteln oder Art und Umfang von Ein- und Ausgaben beeinflussen kann“ /DIN 88, S. 3/.

Erwartungskonformität, Conformity with user expectations: „Ein Dialog ist erwartungskonform, wenn er den Erwartungen der Benutzer entspricht, die sie aus Erfahrungen mit bisherigen Arbeitsabläufen oder aus der Benutzerschulung mitbringen sowie den Erfahrungen, die sie sich während der Benutzung des Dialogsystems und im Umgang mit dem Benutzerhandbuch bilden“ /DIN 88, S/.

Fehlerrobustheit, Error tolerance: „Ein Dialog ist fehlerrobust, wenn trotz erkennbar fehlerhafter Eingaben das beabsichtigte Arbeitsergebnis mit minimalem oder ohne Korrekturaufwand erreicht wird. Dazu müssen dem Benutzer die Fehler zum Zwecke der Behebung verständlich gemacht werden“ /DIN 88, S. 5/.

Suitability of individualization (Individualisierbarkeit): „Dialogue systems are said to support suitability for individualisation if the system is constructed to allow for adaptation to the user's individual needs and skills for a given task“ /ISO 91b/. „Dialogsysteme unterstützen die Individualisierbarkeit, wenn sie so konstruiert sind, daß die Anpassung an die individuellen Bedürfnisse und Fähigkeiten des Benutzers ermöglicht wird.“¹

¹ Die Übersetzung der Definition aus ISO 9241 Teil 10 erfolgte durch die Autoren.

Suitability for Learning (Erlernbarkeit): „Dialogue systems are said to support learnability if they guide the user through the learning stages minimizing the learning time. Reducing complexity and maintaining consistency are the prerequisite goal for this principle“ /ISO 91b/.

Dialogsysteme unterstützen die Erlernbarkeit, wenn sie den Benutzer durch den Lernprozeß führen und die dabei aufzuwendende Lernzeit minimieren. Reduzierung der Komplexität und Erhaltung der Konsistenz sind hierfür die Voraussetzungen.

Darüber hinaus finden sich in den Teilen 12 bis 16 der ISO 9241 eine Vielzahl von software-ergonomischen Forderungen, die sich auf die Präsentation der Information (Teil 12), die Unterstützung der Benutzerführung (Teil 13) und die gängigen Dialogtechniken (Teile 14 bis 16) beziehen.

Den letzte Schritt sollte die Auswahl geeigneter hardware-technischer Systeme darstellen. Damit sollte vermieden werden, daß hardware-technische Gegebenheiten zum Vorwand für einen technikzentrierten Entwicklungsprozeß herangezogen werden. Die Teile 3 bis 9 der ISO 9241 enthalten eine Reihe von hardware-ergonomischen Forderungen, die bei einem Auswahl- bzw. Evaluationsprozeß zu berücksichtigen sind.

9.3 Welche Methoden der Evaluation sind verfügbar?

9.3.1 Subjektive Evaluationsmethoden

Subjektive Evaluationsmethoden knüpfen unmittelbar an die Beurteilung durch den Benutzer an. Er ist die Quelle der Evaluation, u.U. sogar ihr Initiator. Bei subjektiven Evaluationsmethoden werden daher eher „weiche“ Daten gewonnen – ob die Benutzung des Systems bequem, angenehm, klar, einsichtig ist etc. – und nicht so sehr „harte“ Daten – ob die Benutzung des Systems hinsichtlich quantitativer Maße wie Zeit und Fehler effizient erfolgt. Bei der Befragung werden dem Benutzer eines EDV-Systems Fragen über bestimmte Systemeigenschaften gestellt, die er aufgrund seiner Erfahrung mit dem System beantworten muß. Die Befragung kann aber auch dazu dienen, bestimmte Charaktereigenschaften der Benutzer zu erfassen oder um Aussagen über die Arbeitszufriedenheit zu erhalten. Bei Befragungen ist zu unterscheiden zwischen mündlichen und schriftlichen Befragungen sowie zwischen solchen, die ein Evaluator mit Probanden durchführt und solchen, die er bei sich selbst anwendet (Durchführungsprotokoll). Als eine weitere subjektive Evaluationsmethode ist die Methode des „lauten Denkens“ zu nennen, bei

der ein Benutzer eine Aufgabe ausführt und dabei seine Überlegungen, Probleme, Handlungsalternativen etc. laut formuliert und dadurch dem Untersuchungsleiter Anhaltspunkte für die Interpretation des Versuchsgeschehens bietet. Als Alternative zu dieser manchem Teilnehmer künstlich erscheinenden Vorgehensweise wird gelegentlich die Methode des „Partnerggesprächs“ (constructive interaction) angewendet, bei der zwei Benutzer eine Aufgabe gemeinsam bearbeiten und sich gegenseitig „erzählen“, was sie jeweils empfinden, tun bzw. zu tun gedenken etc. und dadurch auf „natürlichere“ Weise Interpretationsmaterial liefern.

Vor- und Nachteile: Die Vorteile der subjektiven Evaluationsmethoden liegen darin, daß man Rückschlüsse bzgl. der Akzeptanz ziehen kann, daß eine wenig aufwendige und leichte Durchführung möglich ist, daß sie zur Einkreisung unstrukturierter Probleme eingesetzt werden können, etc. Nachteile können darin bestehen, daß subjektive Evaluationsmethoden anfällig sind für Übertreibungen, daß man einen repräsentativen Querschnitt an Probanden benötigt, daß es zu hohen Ausfallsquoten beim Rücklauf von schriftlichen Antworten kommen kann, daß sie Suggestionen durch die Untersuchungsfragestellung fördern, daß sie eine Vielzahl von Daten produzieren und sich damit die Auswertung aufwendig gestalten kann, daß von den Befragten in der Regel nur einfache Fragen beantwortet werden können, etc.

Beispiele: Praktisch einsetzbare Beispiele für schriftliche subjektive Evaluationsmethoden sind der „Questionnaire for User Interaction Satisfaction (QUIS 5.0)“ von /Norman, Shneiderman 89/ oder die „Evaluation Checklist“ von /Ravden, Johnson 89/. Ein neuer und vielversprechender Ansatz für ein subjektives Evaluationsverfahren ist der „Software Usability Measurement Inventory“ (SUMI), der im Rahmen des MUSiC-Projektes realisiert wird /MUSiC 92/.

9.3.2 Objektive Evaluationsmethoden

Bei objektiven Methoden versucht man, subjektive Einflüsse weitgehend auszuschalten. Die gängigste Methode der objektiven Evaluation ist die Beobachtung. Sie kann einerseits ohne technische Unterstützung anhand von Beobachtungsprotokollen erfolgen, die in der Regel in strukturierter Form vorliegen und die zu beobachtenden Sachverhalte vorgeben. Der Evaluator kann sich beim Ausfüllen der Protokolle passiv verhalten, oder er kann an die beobachtete Person Fragen zur näheren Abklärung bestimmter Sachverhalte stellen (Beobachtungsinterview). Beobachtungsinterviews werden häufig im Rahmen von Arbeits- und Aufgabenanalysen eingesetzt, um ergonomische Aspekte der Aufgabenerfüllung zu evaluieren. Bei der Evaluation von software-ergonomischen Fragestellungen spielt die technisch unterstützte Protokollierung eine große Rolle. Dabei macht man sich die technischen Möglichkeiten des Beobachtungsgegenstandes zunutze,

indem man ihn gleichzeitig zur Protokollierung von interessierenden Sachverhalten heranzieht. Gleichzeitig lassen sich dadurch etwaige subjektive Einflüsse durch den Benutzer oder den Beobachter verhindern. Bei dem technisch übermittelten Protokollieren des Benutzerverhaltens ist das Logfilerecording die am häufigsten gebrauchte Methode. Sie ist allerdings auf den Umgang des Benutzers mit dem System beschränkt, d.h. es können nur Operationen erfaßt werden, die durch die Nutzung von Eingabemedien manifest werden. Aktionen oder Interaktionen des Benutzers anderer Art, wie z.B. mimische/pantomimische Ausdrucksweisen, Exklamationen, Handbuchgebrauch, Kommunikation mit anderen Personen etc., können nicht berücksichtigt werden. Daher werden automatische Protokollierungsmethoden in der Regel durch Videoaufzeichnungen ergänzt, um die vorgenannten Informationen ebenfalls berücksichtigen zu können.

Vor- und Nachteile: Der wesentliche Vorteil besteht darin, daß Aspekte evaluiert werden können, die sich einer Beurteilung durch den Benutzer entziehen (z.B. nicht bewußtseinsfähige und nicht bewußtseinspflichtige Operationen, die nicht verbalisiert werden können). Desweiteren werden subjektive Einflüsse der Beurteilung ausgeschaltet, was bei der Verallgemeinerung von Ergebnissen eine wichtige Rolle spielt. Der wesentliche Nachteil besteht darin, daß objektive Evaluationsmethode in der Regel aufwendig durchzuführen sind und daher vielfach auf den Einsatz in Untersuchungslabors beschränkt sind. Die Beschränkung auf die Erfassung von Verhaltensdaten engt den Anwendungsbereich zusätzlich ein. So bleibt offen, welche Hypothesen des Benutzers beziehungsweise welche Vorstellungen vom EDV-System zu den mit dieser Methode aufgezeichneten Operationen geführt haben /Moll 87/.

Beispiele: Ein Beispiel für einen umfassenden objektiven Evaluationsansatz ist der „Diagnostic Recorder for Usability Measurement“ (DRUM), der im MUSiC Projekt verfolgt wird /MUSiC 92/. Es handelt sich dabei um eine Kombination zwischen Logfilerecording und Videoaufzeichnungen. Mittels Computerunterstützung wird der Evaluationsprozeß geplant, durchgeführt und ausgewertet.

9.3.3 Leitfadenorientierte Evaluationsmethoden

In gewissem Sinn eine Zwischenstufe zwischen subjektiven und objektiven Evaluationsmethoden sind leitfadenorientierte Evaluationsmethoden. Das Expertenurteil ist die am häufigsten verwendete leitfadenorientierte Evaluationsmethode. Hierbei wird ein EDV-System durch einen Experten geprüft, der sich dabei jedoch – anders als der Benutzer bei den oben besprochenen Befragungen – weniger an (s)einer Aufgabe mit dem zu prüfenden System, als an software-ergonomischen Fragestellungen orientiert. Subjektiv sind diese Verfahren insofern, als ein Subjekt aufgrund seiner Einschätzung eine software-ergonomische Fragestellung selbst

prüft und beantwortet, objektiv sind diese Verfahren insofern, als die software-ergonomischen Prüfkriterien soweit operationalisiert und präzisiert sind, daß der Prüfer seine Antwort aufgrund eindeutiger Testvorschriften und intersubjektiv nachvollziehbarer Bedingungen gibt. Wenn dem Evaluator Prüfkriterien und die anzuwendenden Methoden vorgegeben werden, spricht man von einem methodengeleiteten Expertenurteil /Piepenburg, Rödiger 89/. Die Prüfkriterien und einzusetzenden Methoden werden in einem Prüflitfadensystem festgeschrieben, den der Evaluator bei der Durchführung der Evaluation anleitet. Auf diese Weise wird auch eine größtmögliche Nachvollziehbarkeit der Prüfergebnisse gewährleistet.

Vor- und Nachteile: Das methodengeleitete Expertenurteil verbindet Vorteile der subjektiven und objektiven Evaluationsmethoden, da es mit relativ geringem Durchführungsaufwand eingesetzt werden kann und gleichzeitig nachvollziehbare Ergebnisse liefert. Nachteilig ist, daß es auf objektivierbare Kriterien mit bekannten Kriteriumswerten begrenzt ist und einen entsprechend qualifizierten Evaluator voraussetzt.

Beispiele: Es gibt gegenwärtig eine Reihe von Prüflitfäden für Expertenurteile, die zur Prüfung von software-ergonomischen Qualitäten eingesetzt werden können und die mehr oder weniger den genannten Bedingungen der Operationalisierung und Präzisierung der Fragestellung und der Kontextbedingungen der Prüfung entsprechen. Praktische Beispiele für leitfadenorientierte Evaluationsmethoden sind das Prüfverfahren des TÜV Bayerns /Lang, Peters 88/ oder die umfangreiche Sammlung von Fragebögen zur Bewertung des Einsatzes neuer Technologien im Unternehmen in /Clegg et al. 88/. Viele leitfadenorientierte Evaluationsverfahren überlassen es dem Evaluator, wie er mit dem zu prüfenden System umgeht, um Antworten auf die Fragen zu bekommen. Einige Verfahren bieten neben den eigentlichen Prüffragen auch eine Durchführungsvorschrift. Hierzu gehört z.B. das im Abschnitt 9.5 näher beschriebene EVADIS II-Verfahren.

9.3.4 Experimentelle Evaluationsmethoden

Das Ziel beim Einsatz experimenteller Evaluationsmethoden besteht darin, bestimmte Hypothesen zu testen. Sie spielen daher beim Testen von Theorien im Forschungsbereich eine große Rolle. Prominente Beispiele sind Untersuchungen, die das GOMS-Modell von /Card, Moran, Newell 83/ testen. In dem GOMS-Modell wird angenommen, daß routinemäßige kognitive Fähigkeiten zur Benutzung von Anwendungssystemen durch die Goals, Operations, Methods und Selection rules der Benutzer beschrieben werden können. Experimente werden so zugeschnitten, daß sie diese Annahme zu überprüfen vermögen. Dabei wird z.T. explizit auf Poppers Falsifikationsprinzip Bezug genommen, d.h. der Untersuchungsrahmen wird

so gestaltet, daß wohl-definierte Bedingungen für Möglichkeiten des Scheiterns der Theorievorhersage gegeben sind.

Bei den experimentellen Verfahren spielt auch der Benchmark-Test eine prominente Rolle. Bei Benchmark-Tests werden Systeme anhand von standardisierten Aufgaben im Vergleich untersucht. Als Beispiel kann die Untersuchung von /Roberts, Moran 83/ dienen. In der Studie wurden 9 Texteditoren verglichen. Vergleichsbasis war eine standardisierte Aufgabe, bei der fest definierte Operationen an einem Text vollzogen werden mußten. Kennzeichen der Methode war ihre Objektivität, die Generalität der Evaluationsaspekte, d.h. die Erfassung verschiedener Variablen wie Ausführungszeit, Fehler, Erlernbarkeit und Funktionalität, sowie die Leichtigkeit der Anwendung. Benchmark-Tests kommen als vergleichende Tests nicht zu absoluten Aussagen über ein System, sondern ordnen verschiedene Systeme anhand definierter Kriterien auf einer Rangskala.

Vor- und Nachteile: Die Stärke experimenteller Evaluationsverfahren liegt darin, daß sie bei sorgfältiger Durchführung Kausalschlüsse über die Art der Mensch-Rechner-Interaktion ermöglichen und daher wesentliche Beiträge zur Theoriebildung liefern können.

Das Hauptproblem bei der Planung von Experimenten liegt in der angemessenen Bestimmung der abhängigen und unabhängigen Variablen, ein anderes in der angemessenen Wahl der Untersuchungsumgebung. Bei der Zusammenstellung von unabhängigen Variablen kommt man leicht auf einige Dutzend, wenn nicht hunderte von Merkmalen. Schon bei einfacher Hard- und Software-Konfiguration kommt man auf astronomische Zahlen an Kombinationen möglicher Einflußfaktoren auf software-ergonomisch relevante Merkmale. Mit verschiedenen Aufgaben, Tageszeit- oder Motivationsbedingungen hat man weitere Beispiele der zu berücksichtigenden unabhängigen Variablen. Ebenso geht es bei den abhängigen Variablen. Auch hier kann man sich beliebig viele Merkmale vorstellen, die zu erfassen wären (beispielsweise Erlernzeit, Bearbeitungszeit, Behaltenszeit, Fehlerzahl/-art, Ermüdung, Konzentration, Akzeptanz). In einem einzelnen Experiment werden jeweils nur ganz wenige Variablen erfaßt, und tatsächlich ist etwas anderes auch kaum realisierbar. Trotzdem ist es unbefriedigend, aus Praktikabilitätsgründen einfach bei zwei, drei unabhängigen Merkmalen stehen zu bleiben (z.B. zwei Typen von Kommandoaktivierungen und zwei Gruppen von Benutzern) und alle anderen möglichen Merkmale zu ignorieren (z.B. Art der Aufgabe, Relevanz von Fehlern, Streßbedingungen etc.). Dies führt dazu, daß die Ergebnisse der experimentellen Evaluation nur mit großen Vorbehalten generalisierbar sind. Hinzu kommt der Mangel an grundlegender Theoriebildung zur Mensch-Maschine-Interaktion für die Bestimmung der relevanten Variablen. Die Festlegung der zu berücksichtigenden Merkmale ist daher meist der Phantasie und Einfühlung des Forschers überlassen. Wenn man also aus praktischen und

theoretischen Gründen nur unzureichend in der Lage ist, die relevanten Variablen explizit zu kontrollieren, so sollten jedoch wenigstens die Bedingungen, unter denen Experimente durchgeführt werden, wenn sie schon nicht systematisch variiert werden können, so doch sorgfältig dokumentiert werden, um die Möglichkeit zu Nachprüfungen zu bieten. All diese Beschränkungen bedingen, daß experimentelle Evaluationsmethoden auf den Forschungsbereich beschränkt bleiben und in der Regel auch nur in Untersuchungslabors durchgeführt werden können.

9.3.5 Kombination von Evaluationsmethoden

Vor dem Hintergrund der Anforderungen an eine ganzheitliche Evaluation, wie sie im Abschnitt 9.2 beschrieben wurden, und basierend auf der obigen Klassifikation wurde in /Reiterer 90/ eine Vielzahl von Evaluationsverfahren analysiert, wie beispielsweise /Baitsch, et al. 89/, /Clegg et al. 88/, /ETH-LAO 86/, /Hoyos et al. 90/, /Lang, Peters 88/, /Smith, Mosier 86/, /MITRE 91/, /Norman, Shneiderman 89/, /Oppermann et al. 88/, /Ravden, Johnson 89/, /Sherwood-Smith 89/, /Siemens 87, 90/, /Simes, Sirsky 85/, /TBS 91/, /Tullis 88/. Dabei wurde gezeigt, daß es nicht die „beste“ Evaluationsmethode gibt, sondern daß deren Eignung vom Gegenstand und Einsatzzweck der Evaluation bestimmt wird. Jede Methode hat ihre spezifischen Vorteile und Evaluationsschwerpunkte. Es empfiehlt sich daher, eine Kombination unterschiedlicher Evaluationsmethoden einzusetzen, um zu ganzheitlichen Aussagen bzgl. der software-ergonomischen Qualität der Benutzerschnittstelle zu kommen. Die Tabelle 9-2 zeigt eine derartige Methodenkombination, die auf spezifische Bedürfnisse der Praxis Rücksicht nimmt, wie beispielsweise einfache Handhabung, möglichst geringer Aufwand, usw. Daher wurden keine experimentellen Evaluationsmethoden berücksichtigt.

Der Schwerpunkt der software-ergonomischen Evaluation liegt auf der Benutzerschnittstelle. Hierzu sind eine Reihe ausgereifter subjektiver, objektiver und leitfadenorientierter Methoden und Werkzeuge verfügbar. Anhand eines Expertenurteils lassen sich mit relativ geringem Evaluationsaufwand Aussagen bzgl. der ergonomischen Qualität der Software gewinnen /Hammond et al. 85/. Expertenurteile haben primär qualitativen Charakter. Will man zusätzlich quantitative Aspekte der Benutzung, wie beispielsweise Fehlerhäufigkeit, Erlernzeit, Hilfeforderung erfassen, bietet sich der Einsatz von technisch unterstützten Protokollierungsinstrumenten an (z.B. Logfilerecording, Videoaufzeichnung). Dies bedeutet allerdings auch eine erhebliche Steigerung des Evaluationsaufwandes und setzt entsprechendes Know-how und Equipment voraus. Um Fragen der Benutzerakzeptanz zu erfassen, können einfache Fragebögen herangezogen werden. Dies stellt eine sinnvolle Ergänzung zu den experten- und technikbasierten Evaluationsmethoden dar und kann durchaus zu widersprüchlichen Bewertungs-

Tab. 9-2: Kombination von Evaluationsmethoden

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

ergebnissen führen. Den Aussagen der Benutzer – als eigentlich Betroffenen – sollte dabei auch ein entsprechender Stellenwert eingeräumt werden.

Im Sinne einer ganzheitlichen Evaluation sind die Einflüsse der Benutzer und der Aufgaben auf die Benutzung der Software zu berücksichtigen. Durch Beobachtung des Arbeitsplatzes und der Aufgaben können Aussagen über die ergonomische Qualität der Aufgaben und der Organisation gemacht werden. Dazu eignen sich vor allem arbeitspsychologische Arbeitsanalyseverfahren, wie beispielsweise KABA, /Dunckel 89/, /Zölch, Dunckel 91/; VERA/B, /Rödiger, Nullmeier, Oesterreich 86/; TBS-GA, /Rudolph, Schönfelder, Hacker 87/; VBBA, /Bonitz et al. 88/, /89/. Ergonomische Mängel in diesem Bereich beeinflussen maßgeblich die Benutzung der Software. Deren Berücksichtigung bei der software-ergonomischen Evaluation soll sicherstellen, daß es zu keiner Verengung des Blicks nur auf Schnittstelleneigenschaften kommt („Schnittstellenkosmetik“). Verfahrenstechnisch weniger aufwendig sind Befragungen der Benutzer, beispielsweise unter Verwendung einfacher Fragebögen (z.B. Fragebogen zur Subjektiven Arbeitsanalyse (SAA, siehe /Baitsch et. al. 89/), die deren Arbeitszufriedenheit und damit deren Sicht der ergonomischen Qualität der Aufgabe erfassen. Derartige Verfahren können auch ergänzend zu Arbeits- und Aufgabenanalyseverfahren eingesetzt werden.

Zur Erfassung der Benutzereigenschaften eignet sich ebenfalls die Methode der Befragung. Dafür gibt es eine Reihe von Fragebögen, die von den Benutzern auszufüllen sind (z.B. Methode zur Ermittlung der Benutzercharaktere, in /Urbanek 92/). Dabei werden Aspekte wie Arbeitsbereich des Benutzers, EDV-Erfahrung, Nutzungshäufigkeit von EDV-Systemen, usw. erfaßt. Die Ergebnisse der Befragung können zur Gewichtung der Bedeutung der software-ergonomischen Kriterien für die Benutzergruppe herangezogen werden und ermöglichen somit eine differenziertere Evaluation der Benutzerschnittstelle.

Wenn eine Kombination von Methoden zur Anwendung gelangt, ist es wichtig, daß es eine Art Durchführungsvorschrift gibt, die den Anwender genau anleitet, wann er welche Methode wie einzusetzen hat und wie die einzelnen Ergebnisse ineinandergreifen und zu einem abschließenden Qualitätsurteil zusammengefaßt werden müssen. Eine derartige Durchführungsvorschrift fungiert einerseits als Handlungsanleitung, ermöglicht andererseits eine Nachvollziehbarkeit der Evaluationsergebnisse. Ein Beispiel für ein derartiges kombiniertes Evaluationsverfahren ist das im Abschnitt 9.5 vorgestellte Verfahren EVADIS II.

9.4 Wie können Evaluationsmethoden in den Entwicklungsprozeß eingebunden werden?

Der Software-Entwicklungsprozeß wird typischerweise in Phasen strukturiert, welche die Aktivitäten der Entwickler beschreiben. Es gibt eine Vielzahl von Phasenmodellen mit sehr unterschiedlichen Phasenbezeichnungen. Inhaltlich sind die Überschneidungen groß, wobei man typischerweise die folgenden Phasen unterscheidet /Olle et al. 88/:

- Planung
- Analyse
- Design
- Programmierung
- Installation und Test
- Betrieb und Wartung.

Ziel der Aufteilung des Softwareentwicklungsprozesses in eine Reihe von wohldefinierten Phasen war, die Projektplanung und -steuerung zu erleichtern. Es sollten gewissermaßen ingenieurmäßige Methoden der Softwareentwicklung zum Einsatz kommen. Die sequentielle Struktur der ersten Phasenmodell ist vielfach kritisiert worden, vor allem vor dem Hintergrund der Berücksichtigung software-ergonomischer Anforderungen. Als Alternative wurden evolutionäre Phasenmodell entwickelt, die ein iteratives Design unterstützen /Floyd 84/. Sie haben unter dem Schlagwort „Software Prototyping“ sowohl im Bereich des Software-Engineering als auch im Bereich der Software-Ergonomie große Bedeutung erlangt, da sie einerseits den sich rasch verändernden Entwicklungsanforderungen während des Software-Lebenszyklus besser Rechnung tragen und andererseits eine bessere Berücksichtigung von Benutzeranforderungen im Entwicklungsprozeß ermöglichen /Eason 82/, /Mambrey, Oppermann, Tepper 86/. Prototypingorientierte Entwicklungsansätze stellen auch eine wichtige Voraussetzung für die effektive Berücksichtigung von Evaluationsergebnissen dar. Nur durch sie kann gewährleistet werden, daß im Rahmen der Qualitätssicherung mittels einer projektbegleitenden software-ergonomischen Evaluation (formative Evaluation) festgestellte Defizite rechtzeitig und mit vertretbarem Aufwand behoben werden können /Sherwood-Smith 89/. Viele der heute verfügbaren Evaluationsverfahren sind vor allem für eine nachträgliche Evaluation der Software geeignet (summative Evaluation).

Für den Softwareentwickler stellt sich nun die Frage, wann kann er im Entwicklungsprozeß welches Evaluationsverfahren einsetzen und welcher Evaluationsaufwand ist damit verbunden? Dazu haben /Kishi, Kinoo 91/ vier Kriterien entwickelt, anhand deren man Evaluationsverfahren beurteilen und dann entscheiden

kann, wann und mit welchem Aufwand man sie im Entwicklungsprozeß einsetzen kann:

- Die Zeit, die man für den Einsatz der Evaluationsmethode benötigt. Sie variiert sehr stark von Methode zu Methode und wird vor allem von dem zu treibenden methodischen Aufwand bestimmt (z.B. einfache Checkliste versus experimentelle Evaluation in einem Versuchslabor).
- Die Art und Anzahl von Problemen, die durch den Einsatz einer Evaluationsmethode erkannt werden sollen. Dies wird dadurch bestimmt, für welchen Einsatzbereich das Evaluationsverfahren entwickelt wurde und ob Eigenschaften der Benutzer und Fragen der Organisations- und Aufgabengestaltung in die Evaluation miteinbezogen werden (ganzheitlicher Ansatz). Je umfassender ein Evaluationsverfahren konzipiert wurde, desto mehr Probleme unterschiedlicher Art können auch erkannt werden.
- Die notwendigen Voraussetzungen zum Einsatz des Evaluationsverfahrens. Auch hier gibt es eine große Bandbreite an notwendigen Voraussetzungen, in Abhängigkeit vom gewählten Verfahren (z.B. Anzahl der Versuchspersonen, erforderliches Know-How des Evaluators, notwendige Werkzeuge).
- Der Grad der Meßqualität des Evaluationsverfahrens. Spielen Fragen der Validität und der Reliabilität von Evaluationsergebnissen eine wichtige Rolle (z.B. bei experimentellen Evaluationen), wird man bei der Auswahl von Evaluationsverfahren auf die entsprechende empirische und statistische Absicherung achten.

Es gibt kein Evaluationsverfahren, das alle vier Kriterien gleich optimal erfüllen kann. Evaluationsverfahren, die beispielsweise wenig Zeit beanspruchen, können in der Regel auch nur eine beschränkte Art und Anzahl von Problemen erfassen. Will man eine hohe Qualität der Meßergebnisse erreichen, bedarf es meist umfangreicher Voraussetzungen und damit auch entsprechend viel Zeit. So gibt es eine Reihe von Abhängigkeiten und Zielkonflikten zwischen diesen Kriterien, und so muß der Evaluator von Fall zu Fall entscheiden, welche Priorität er den einzelnen Kriterien für seine Evaluationsaufgabe zuordnet. In der Praxis kommt vor allem dem Kriterium Zeit eine herausragende Bedeutung zu. Die erforderlichen Voraussetzungen dürfen ebenfalls nicht zu anspruchsvoll sein, da sie sonst mit dem vorhandenen Know-How und verfügbaren Werkzeugen nicht zu erfüllen sind. Die Art und Anzahl der Probleme, die im Rahmen der software-ergonomischen Evaluation erfaßt werden sollen, bezieht sich in der Regel schwerpunktmäßig auf die Benutzungsoberfläche. Der Grad der Meßqualität hat für Evaluationsverfahren in der Praxis nicht den Stellenwert, wie beispielsweise für experimentelle Evaluationen im Forschungsbereich.

Die im Abschnitt 9.3.5 vorgestellte Kombination von Evaluationsmethoden stellt – vor dem Hintergrund des Aufwandes und der Einsetzbarkeit im Entwicklungs-

prozeß - einen pragmatischen Kompromiß dar. Es wurde versucht, den Gesamtevaluationsaufwand zu minimieren, dabei aber eine möglichst ganzheitliche Sichtweise zu verfolgen, die notwendigen Voraussetzungen gering zu halten und einen Grad an Meßqualität zu erreichen, der zumindest eine Nachvollziehbarkeit der Evaluationsergebnisse erlaubt. Darüber hinaus ist eine einfache Einbettung der einzelnen Methoden und Werkzeuge der vorgeschlagenen Kombinationen in den Entwicklungsprozeß möglich. Im Rahmen der Analyse können die Benutzereigenschaften erfaßt (Befragung mittels Fragebogen) und die vorhandenen Aufgaben analysiert werden (Beobachtung mittels Arbeits- und Aufgabenanalyse). Da dies zu den typischen Tätigkeiten, die im Rahmen der Softwareentwicklung innerhalb dieser Phase zu erbringen sind, gehört, ist damit kein großer Zusatzaufwand verbunden. In der Designphase sollte idealerweise ein prototypingorientierter Entwicklungsansatz verfolgt werden, bei dem bereits zu einem frühen Zeitpunkt erste Prototypen der Benutzerschnittstellen erstellt werden (z.B. mittels des Einsatzes von User Interface Management Systems). Diese Benutzerschnittstellen können dann bzgl. ihrer software-ergonomische Qualität evaluiert werden (Expertenurteil mittels Leitfaden und Evaluations-Software). Wenn kein Prototypingansatz verfolgt wird, muß die Evaluation der Schnittstelle in der Installations- und Testphase erfolgen. Auch beim zunehmend wichtiger werdenden Einsatz von Standard-Software erfolgt der Einführungsprozeß anhand von Phasenmodellen. Hier werden die Phasen Design sowie Programmierung durch eine Phase: Auswahl von Standard-Software ersetzt /Koch, Reiterer, Tjoa 91/. Auch hier kann eine ähnliche Zuordnung der vorgeschlagenen Evaluationsmethoden vorgenommen werden, wobei im Rahmen des Auswahlprozesses die software-ergonomische Evaluation der Benutzerschnittstelle erfolgt (Expertenurteil mittels Leitfaden und Evaluations-Software).

9.5 EVADIS II- Ein neuer Evaluationsansatz

EVADIS II wurde in dem Forschungsbereich Mensch-Maschine-Kommunikation der GMD in enger Kooperation mit dem Institut für Statistik und Informatik der Universität Wien entwickelt. Das Evaluationsverfahren wurde in /Oppermann et al. 92/ veröffentlicht.

EVADIS II basiert auf einer Kombination von Methoden, wie sie in Abschnitt 9.3.5 beschrieben wurde und stellt damit einen ersten Schritt in Richtung eines umfassenden Evaluationsansatzes dar. Es ist aber auch klar, daß mit dieser pragmatischen Kombination von Methoden nicht alle Aspekte einer umfassenden software-ergonomischen Evaluation abgedeckt werden können. Die Grenzen von EVADIS II werden in Abschnitt 9.5.6 beschrieben.

9.5.1 Vorgehensweise beim Einsatz des Evaluationsverfahrens

Das Evaluationsverfahren EVADIS II besteht aus den folgenden fünf Verfahrensschritten, die im Detail im Handbuch zum Verfahren beschrieben sind (Durchführungsvorschrift):

1. Installation und Exploration der zu evaluierenden Software.
2. Exploration und Evaluation der Aufgaben; Auswahl von relevanten Prüffragen und Erstellen der Prüfaufgabe(n).
3. Exploration der Benutzereigenschaften.
4. Evaluation der Software anhand der Prüfaufgabe(n).
5. Interpretation der Ergebnisse und Schreiben des Prüfberichtes.

Abb. 9-3 gibt einen Überblick über die Verfahrensschritte und zeigt die EVADIS II-Verfahrenskomponenten, die erforderlich sind, um die fünf Evaluationschritte auszuführen. Die ersten drei Schritte können weitgehend gleichzeitig ausgeführt werden. Das Ergebnis dieser drei Schritte besteht in einer Prüfaufgabe, die zum Evaluieren der Software benutzt wird und in einer geordneten Liste von software-ergonomischen Kriterien. Die Rangordnung dieser Liste reflektiert die Bedeutung eines jeden ergonomischen Kriteriums für die spezielle Benutzergruppe. Damit soll sichergestellt werden, daß die Benutzereigenschaften berücksichtigt werden, sowohl während der Evaluation der Benutzerschnittstelle als auch während der Interpretation der Untersuchungsergebnisse. Schritt 4 ist der zentrale Schritt des Evaluationsverfahrens. Hier werden alle ausgewählten Prüffragen beantwortet. Sie sind in die Prüfaufgabe eingebettet, so daß der Evaluationsprozeß im Wechsel zwischen der Bearbeitung der Prüfaufgabe und der Beantwortung der zugehörigen Prüffragen erfolgt. Das Ergebnis dieses Vorgehens ist ein Prüfprotokoll, das die Basis für die Interpretation der Ergebnisse und den zu schreibenden Prüfbericht darstellt.

9.5.2 Die Verfahrenskomponenten

Die folgenden Komponenten von EVADIS II werden während der Evaluationen benutzt, um den Evaluator anzuleiten (vgl. Abb. 9-3).

- A Fragebogen zur Evaluation der Aufgaben
- B Leitfaden zur Erstellung der Prüfaufgaben
- C Fragebogen zur Erfassung von Benutzereigenschaften
- D Prüffragensammlung
- E Liste typischer Funktionen von Bürosoftware
- F Sammlung von Beispielprüfaufgaben
- G Leitlinien für die Abfassung des Prüfberichtes.

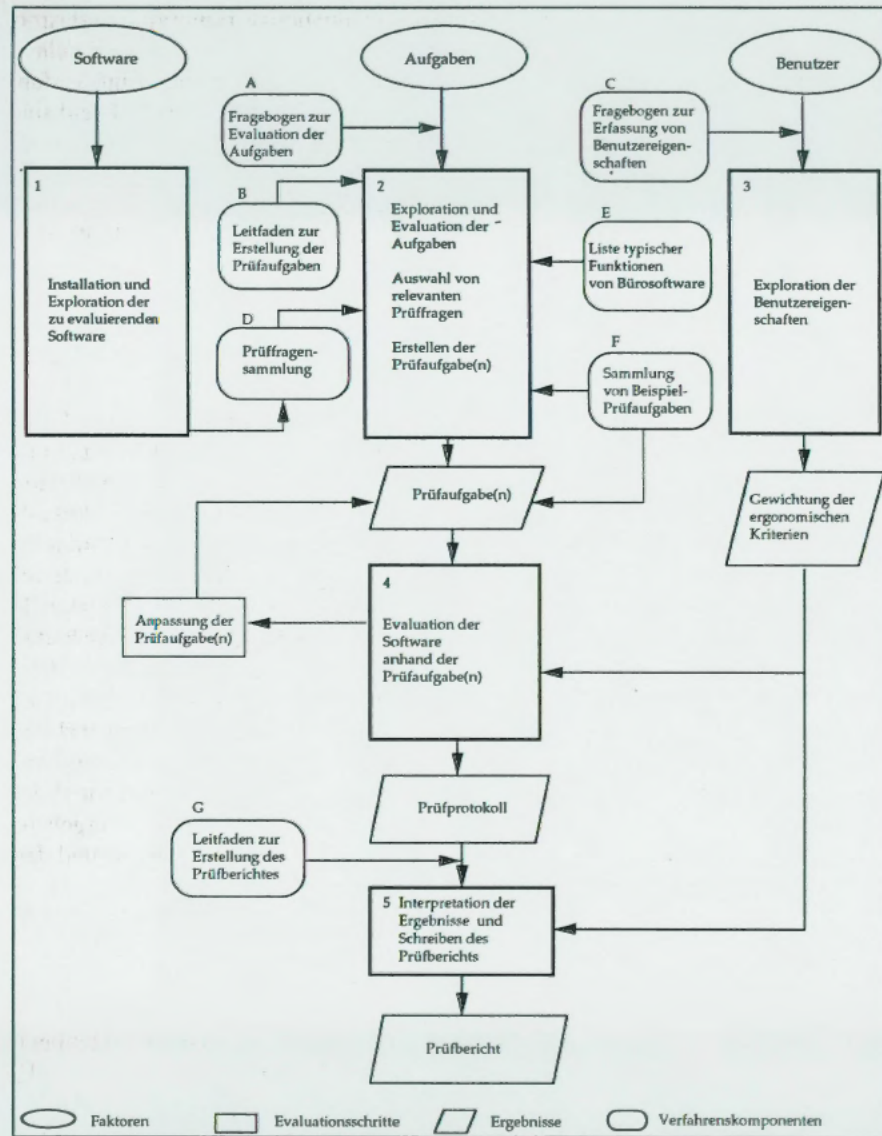


Abb. 9-3: Vorgehensweise beim Einsatz von EVADIS II

Übersicht 9-1: Prüffrage zur Evaluation der Arbeitsqualität

Prüffragennummer: 422.05.10

Bewertungsbereich 422: Gestaltung der Arbeitsabläufe

Kriterium 05: Autonomie

Prüffrage: Kann der Arbeitende die auszuführenden Aufgaben selbständig planen, vorbereiten und kontrollieren (Entscheidungsspielraum)?

Quelle: in Anlehnung an Bonitz; 1989 (VBBA Frage 6)

Antwortvorgaben:

nein

teils

ja

Anmerkungen:

Die folgende Beschreibung der wesentlichen Komponenten von EVADIS II soll den Leser mit den zugrundeliegenden Konzepten des Evaluationsansatzes vertraut machen.

Der Fragebogen zur Evaluation der Aufgaben (Komponente A) besteht aus 25 Fragen, die der Evaluator während der Analyse der Aufgaben und der Arbeitsbedingungen zu beantworten hat (Schritt 2). Der Fragebogen prüft die Eigenschaften der gegebenen Aufgaben und basiert auf den in der ISO 9241 Teil 2 vorgeschlagenen Aufgabencharakteristika:

- Nutzen die Aufgaben eine Bandbreite von Fachkenntnissen entsprechend den Fähigkeiten des Benutzers?
- Lassen die Aufgaben die Identifikation von ganzheitlichen Einheiten der Arbeit erkennen?
- Bieten die Aufgaben dem Benutzer ein ausreichendes Maß an Autonomie?
- Bieten die Aufgaben dem Benutzer eine Rückmeldung über die Aufgabenausführung?

Die Übersicht 9-1 zeigt eine typische Frage für die Evaluation der Arbeitsqualität:

Die Interpretation der Resultate ist einfach, weil die Antwortvorgaben auf die Prüffragen vorgegeben sind. Ein einfaches Auswertungsformular hilft dem Evaluator, eine abschließende Aussage über die ergonomische Qualität der gegebenen Aufgabe zu erstellen. Diese Aussagen werden in den Prüfbericht aufgenommen

(Schritt 5). Es ist klar, daß mithilfe eines solch kurzen Fragebogens unmöglich alle ergonomischen Begrenzungen und Defizite der Aufgaben am Arbeitsplatz entdeckt werden können. Dies ist auch nicht das Ziel des Fragebogens. Es ist lediglich beabsichtigt, dem Evaluator einen ersten Eindruck von der ergonomischen Qualität der Arbeit zu geben und ihm bedeutsame Defizite aufzuzeigen. Falls solche Defizite entdeckt werden, müßten umfangreichere Aufgabenanalysemethoden wie VERA/B (vgl. /Rödiger, Nullmeier, Oesterreich 86/), KABA (vgl. /Dunckel 89/, /Zölch, Dunckel 91/) oder TBS-GA (vgl. /Rudolph, Schönfelder, Hacker 87/) zu einer detaillierten Analyse herangezogen werden.

Der Leitfaden zur Erstellung der Prüfaufgaben (Komponente B) besteht aus detaillierten Instruktionen und einer Sammlung von Antwortblättern (Schritt 2). Mithilfe dieses Leitfadens einer vereinfachten Arbeits- und Aufgabenanalyse kann der Evaluator die Aufgaben untersuchen, die der Benutzer mit der Software ausführt. Indem der Evaluator eine Kombination von Beobachtung und Befragung benutzt (Beobachtungsinterview), füllt er die Antwortblätter aus. Diese Antwortblätter schließen Fragen zu folgenden Gegenständen ein:

- Beschreibung der Arbeitsumgebung und des Arbeitsplatzes, wo die Software eingesetzt wird (organisatorische Einbettung).
- Überblick über die Aufgaben des Benutzers an seinem Arbeitsplatz und Bestimmung der Aufgaben, die mit der Software unterstützt werden.
- Beschreibung der unterstützten Aufgaben und Erstellen einer Liste der genutzten Software-Funktionen.
- Überblick über die Hardware-Ausstattung, die für die Software erforderlich ist.

Bevor der Evaluator die Prüfaufgabe(n) zusammenstellt, hat er die vollständige Prüffragensammlung zu lesen (Komponente D), um die relevanten Fragen auszuwählen und in die Prüfaufgabe aufzunehmen. Basierend auf den Ergebnissen der Beobachtung und Befragung ist der Evaluator in der Lage, die Prüfaufgabe zusammenzustellen. Die Aufgabenanalyse zeigt dem Evaluator, welche der Softwarefunktionen normalerweise benutzt werden, um typische Aufgaben zu bearbeiten. Dabei zeigt sie auch deren Bedeutung für die Aufgabenerfüllung. Somit stellt die endgültige Prüfaufgabe ein „Drehbuch“ dar, das aus allen benutzten Software-Funktionen besteht, die für die Ausführung einer oder mehrerer typischer Aufgaben erforderlich sind. Die einzelnen Prüffragen werden in Abhängigkeit von ihrem Inhalt den Funktionen und Prüfoperationen zugeordnet. Übersicht 9-2 zeigt einen kleinen Ausschnitt aus einer Prüfaufgabe.

EVADIS II enthält eine Sammlung von Beispielprüfaufgaben (Komponente F), die den Evaluator bei der Zusammenstellung von Prüfaufgaben anleiten sollen.

Übersicht 9-2: Beispielhafter Ausschnitt aus einer Prüfaufgabe

...

...

9. Löschen einer Datei

Machen Sie eine Sicherungskopie der Datei „Bericht“ und löschen Sie die Originaldatei

Prüffragennummer: 241.07.20 Inhalt: irreversible Aktion

Prüffragennummer: 320.05.20 Inhalt: Löschen

10. Drucken einer Datei

Drucken Sie die Datei „Bericht“ auf dem lokalen Drucker. Spezifizieren Sie das Layout des Dokumentes bevor Sie es drucken (Format, Größe, Anzahl der Kopien, usw.).

Prüffragennummer: 120.10.20 Inhalt: Wahlmöglichkeiten

Prüffragennummer: 320.03.10 Inhalt: Funktionalität

Prüffragennummer: 140.06.10 Inhalt: Ausgabemedien

...

...

Diese Beispiele zeigen die typische Struktur von Prüfaufgaben und wie Prüffragen in die Prüfaufgabe eingebettet werden.

Der Fragebogen zur Erfassung von Benutzereigenschaften (Komponente C) ist eine Zusammenstellung von Fragen, die die Software-Benutzer zu beantworten haben (Schritt 3). Der Fragebogen enthält 12 Fragen über Benutzereigenschaften, wie beispielsweise Kenntnisse und Erfahrungen im Umgang mit Hard- und Software. Übersicht 9-3 zeigt eine repräsentative Frage aus diesem Fragebogen.

Ein Interpretationsverfahren und eine Klassifikation von typischen Benutzergruppen leitet den Evaluator während der Interpretation der Ergebnisse an. EVADIS II unterscheidet zwischen vier verschiedenen Benutzergruppen: erfahrene und regelmäßige Benutzer, erfahrene und gelegentliche Benutzer, unerfahrene und regelmäßige Benutzer und unerfahrene und gelegentliche Benutzer (vgl. /Triebel, Wittstock, Schiele 87/).

Jeder Benutzergruppe wurde eine entsprechende Rangliste von softwareergonomischen Kriterien zugeordnet, die die Bedeutung jedes Kriteriums für diese Benutzergruppe zeigt. Die Ranglisten basieren auf der psychologischen Theorie des Kontrollkonzeptes (nach /Spinas 87/). Diese Ordnung wird in den Schritten 4 und 5 benutzt und dient der Gewichtung der Resultate der Evaluation. Bei der abschließenden Einschätzungsübersicht in Schritt 5 – generiert durch die

Übersicht 9-3: Beispiel einer Frage aus dem Fragebogen zur Erfassung der Benutzereigenschaften

Frage: Welche Inhalte wurden Ihnen bei Ihrer Ausbildung – oder bei Schulungsmaßnahmen – vermittelt bzw. haben Sie sich selbst angeeignet? (Mehrfachantworten möglich!)

Antwortvorgaben:

Kenntnisse der Bedienung/Nutzung von Programmen: _____

Grundkenntnisse über EDV und eingesetzte Programme: _____

Kenntnisse bzgl. neuer Arbeitsabläufe bedingt durch den Einsatz von Programmen: _____

weitergehende Kenntnisse über EDV und eingesetzte Programme: _____

– Programmierkenntnisse: _____

– Kenntnisse über Betriebssysteme: _____

– Kenntnisse über Entwurf und Anwendung von Datenbanksystemen: _____

– sonstige: _____

Kenntnisse der Ergonomie und der sozialen Folgen des Einsatzes von EDV: _____

Anmerkungen: _____

Evaluations-Software – werden die Kriterien in der Reihenfolge ihrer Gewichtung sortiert. Basierend auf der Gewichtung im Vergleich mit dem durchschnittlichen Rang ist der Evaluator in der Lage, eine differenzierte Benutzbarkeitseinschätzung zu formulieren.

Die Prüffragensammlung (Komponente D) besteht aus etwa 150 Fragen und ist der Kern von EVADIS II. Die Fragen werden benutzt, um die verschiedenen Eigenschaften der Benutzerschnittstelle während der Bearbeitung der Prüfaufgabe zu bewerten (Schritt 4). Die Sammlung basiert auf ausführlichen Recherchen der verfügbaren Literatur (besonders der Normen, der Richtlinien und der Style Guides von Herstellern), auf dem Wissen und der Erfahrung der Autoren und auf einer Auswertung von existierenden Evaluationsverfahren. Um die Anzahl der Prüffragen zu reduzieren, wurden logisch verbundene ergonomische Anforderungen in jeweils einer Frage zusammengefaßt. Die verschiedenen Anforderungen wurden mithilfe eines breiten Spektrums von Antwortmöglichkeiten repräsentiert. Der Vorteil einer solchen Zusammenfassung besteht in einer leichteren Handhabbarkeit und einer größeren Überschaubarkeit der Prüffragensammlung. Trotzdem enthält die Sammlung der Prüffragen lediglich eine repräsentative Auswahl von ergonomischen Anforderungen. Es bleibt die Aufgabe des Evaluators,

basierend auf der Analyse der Aufgaben, der Benutzergruppe und der Software, diese Prüffragen anzupassen oder neue zu formulieren. Die Evaluations-Software bietet hierzu einige nützliche Erleichterungen.

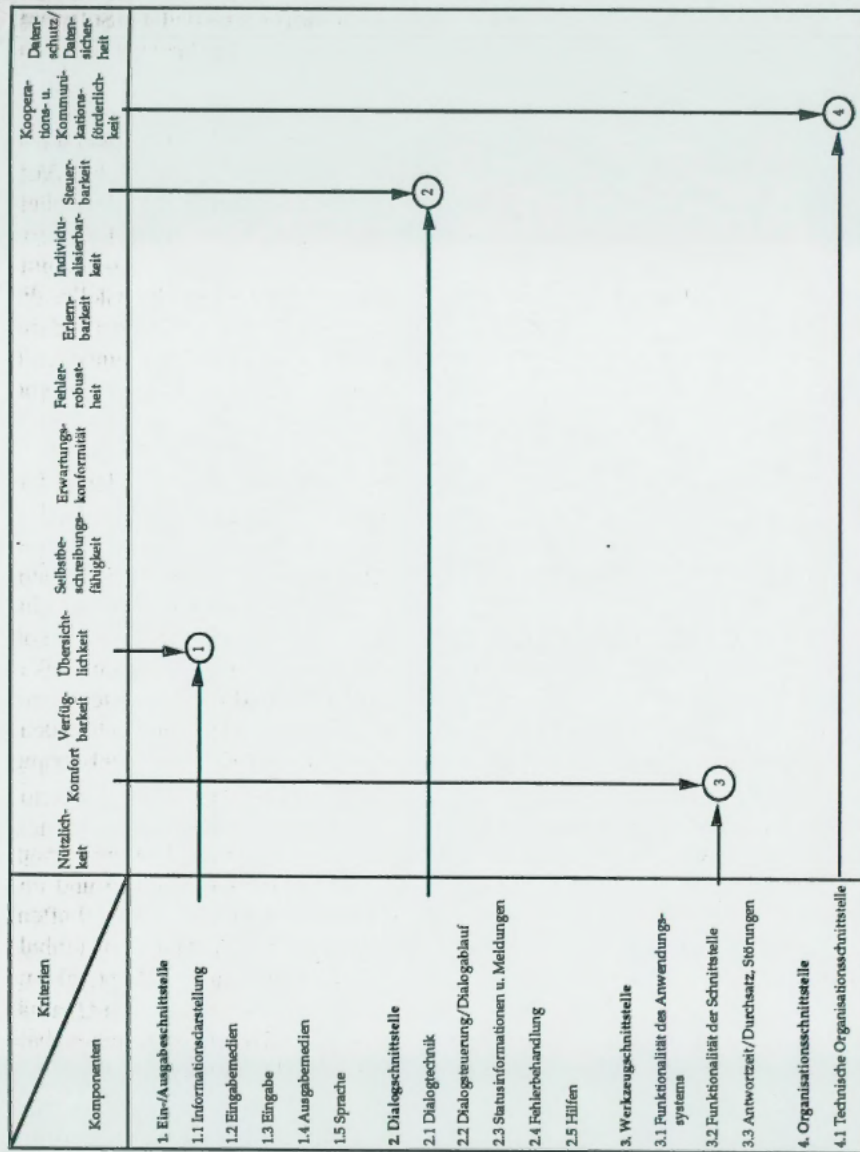
Alle Prüffragen sind eingebettet in einen zweidimensionalen Bezugsrahmen. Abb. 9-4 zeigt diesen Bezugsrahmen im Detail, weil er einige besondere Vorteile des Verfahrens gegenüber anderen Evaluationsmethoden bietet. Die Einbettung der Prüffragen wird anhand von Beispielen erläutert. Die erste Dimension bilden die technischen Systemkomponenten, die 4 Ebenen der Benutzerschnittstelle unterscheiden: die Ein-/Ausgabeschnittstelle, die Dialogschnittstelle, die Werkzeugschnittstelle und die Organisationsschnittstelle. Diese Unterscheidung basiert auf dem IFIP-Modell der Benutzerschnittstelle (vgl. /Dzida 83/) und wurde für die Zwecke der software-ergonomischen Evaluation weiter differenziert und detailliert.

Die zweite Dimension bilden die software-ergonomischen Kriterien. Diese basieren im wesentlichen auf den Dialogprinzipien wie sie bei DIN 66 234 Teil 8 und ISO 9241 Teil 10 (bzw. CEN 29241) vorgeschlagen werden. Der Grund für die Benutzung der ISO/CEN-Kriterien ist ihre steigende Bedeutung vor dem Hintergrund der EG-Verbindlichkeit. EVADIS II schließt 4 weitere Kriterien ein: Verfügbarkeit der Hard- und Software, Übersichtlichkeit der Präsentation von Informationen, der Einfluß von Software auf Kooperations- und Kommunikationsbeziehungen und Maßnahmen des Datenschutzes und der Datensicherheit. Diese 4 Prinzipien wurden hinzugefügt, weil sie unserer Meinung nach bedeutende ergonomische Anforderungen repräsentieren, die außerhalb der Zielsetzung der ISO/CEN-Prinzipien liegen.

Die Repräsentation der Prüffragen in diesem zweidimensionalen Bezugsrahmen, erleichtert dem Evaluator den Zugang zu den Inhalten der Prüffragen und unterstützt ihn bei der Suche nach spezifischen zu untersuchenden Eigenschaften. Die Zweidimensionalität ermöglicht auch unterschiedliche Zugänge zu den Inhalten der Prüffragen. Der Softwareentwickler wird die Dimension der Systemkomponenten bevorzugen, während ein mehr an Fragen der ergonomischen Qualität Interessierter den Zugang über die Kriterien bevorzugen wird. Der Rahmen sichert ebenso die Vollständigkeit der Zusammenstellung von Prüffragen.

Inhalte der Beispielprüffragen:

1. Wie ist der Bildschirm strukturiert?
2. Kann in unterschiedlichen Dialogsituationen (z.B. Arbeit am Dokument, in einem Verzeichnis, in einer Dialogbox) zwischen den verschiedenen Dialogtechniken wahlfrei gewechselt werden?



3. Können Objekte von einem Anwendungsmodul/-programm in ein anderes kopiert werden?
4. Sind Möglichkeiten zum Informationsaustausch in das Anwendungssystem integriert?

Abb. 9-4: Bezugsrahmen der Prüffragen

Übersicht 9-4: Beispiel einer Prüffrage aus der Prüffragensammlung

Kriterium 05:

Prüffragen-Nr.: 230.05.70

Selbstbeschreibungsfähigkeit

Komponente 230: Statusinformationen und Meldungen

Gruppe 1: In der Prüfaufgabe in einzelnen Sequenzen prüfen

Prüffrage:

Sind Meldungen des Anwendungssystems für den Benutzer jederzeit unterscheidbar in Rückmeldungen (bzw. Statusinformationen), Nachfragen des Anwendungssystems, Warnungen und Fehlermeldungen?

Antwortvorgaben:

- ja, Unterscheidungsmerkmale (z.B. Plazierung, Invertierung, Umrahmung, Farbe, Akustik?)
- nein

Kommentar:

Meldungen des Systems sollten jederzeit für den Benutzer unterscheidbar sein, damit er sofort aus der Art der Meldungen auf den derzeitigen Systemzustand schließen und die entsprechenden Reaktionen ableiten kann. Die Unterscheidung sollte durch entsprechende Hervorhebungen unterstützt werden.

Bewertung:

Note: _____ Kriteriumsgewichtung: _____

Beründung: _____

Notiz: _____

Wie das Beispiel in Übersicht 9-4 zeigt, enthält jede Prüffrage neben der Zuordnung zu den Systemkomponenten und Kriterien eine Frage, eine Reihe von möglichen Antworten, einen Kommentar, ein Feld für eine Note, ein Feld für die Begründung der Note, ein Feld für die Kriteriumsgewichtung und ein Notizfeld. Die Gruppenzugehörigkeit definiert, ob ein Prüffitem innerhalb oder am Ende der Prüfaufgabe zuzuordnen ist.

Die Antwortvorgaben decken das generelle Spektrum von möglichen Antworten ab, die dem gegenwärtigen Stand der Forschung und Entwicklung in der Software-Ergonomie entsprechen. Sie helfen dem Evaluator, die ergonomische

Qualität der Software zu beschreiben. Sie sind allerdings nicht vollständig. Erstens sind technische Entwicklungen oder neue Erkenntnisse möglich, die andere Antwortmöglichkeiten erfordern als die bisher formulierten. Zweitens bieten die gegebenen Antwortmöglichkeiten dem Evaluator lediglich einen groben Anhaltspunkt in der Beantwortung und der Prüfung von Funktionen. Einzelne spezielle Eigenschaften können so nicht betrachtet werden, sondern müssen vom Evaluator in bezug auf die spezifische Gegebenheit selbst formuliert werden.

Kommentare wurden bei jeder Prüffrage gegeben, um dem Evaluator die Interpretation und Einschätzung der Fragen und Antworten zu erleichtern. Sie basieren auf dem gegenwärtigen software-ergonomischen Wissen und geben an, welche der Antwortvorgaben gegenwärtig als die besten Schnittstelleneigenschaften angesehen werden. Selbstverständlich sind solche Kommentare Gegenstand raschen Wandels. Sie sind der beste Indikator für den Fortschritt, der auf dem Gebiet der Software-Ergonomie gemacht wird und werden entsprechend regelmäßig aktualisiert.

Basierend auf dem Vergleich der zu analysierenden ergonomischen Qualität mit der wünschenswerten ergonomischen Qualität, die in den Kommentaren beschrieben wird, kann der Evaluator seine Note vergeben. Diese Note kann Werte zwischen 1 (voll zufriedenstellend) und 5 (nicht zufriedenstellend) annehmen und hilft dem Evaluator während der Interpretation der Ergebnisse. Es ist wichtig, darauf hinzuweisen, daß eine solche metrische Evaluation der software-ergonomischen Qualität sehr schwierig ist, weil die Interpretation zwischen verschiedenen Evaluatoren variieren kann.

In einem Extrafeld hat der Evaluator die Gelegenheit, die Note zu begründen. Die Begründung der Einschätzung der ergonomischen Qualität ist für die Interpretation der Ergebnisse und die Zusammenstellung des Prüfberichtes sehr wichtig.

Die Bearbeitung der Prüfaufgabe mit der zu testenden Software kann durch Ereignisse unterbrochen werden, die nichts mit den Prüffragen zu tun haben, z.B. bei Eintritt von Systemfehlern, bei fehlerhaften Operationen und fehlerhaften oder offensichtlich unverständlichen oder unsinnigen Systemantworten. Solche Besonderheiten sind vom Evaluator in dem Feld Notizen festzuhalten.

9.5.3 Interpretation und Präsentation der Evaluationsergebnisse

Die Ergebnisse der Evaluation werden in einem Prüfbericht zusammengestellt (Schritt 5). Eine Richtlinie für die Abfassung des Prüfberichtes (Komponente G) ist integraler Bestandteil von EVADIS II. Diese Richtlinie schlägt die Struktur und den Inhalt des Prüfberichtes vor. Die Struktur basiert auf den Vorschlägen

von ISO 9241 Teil 11. Der Prüfbericht gibt eine Darstellung der software-ergonomischen Eigenschaften der Benutzerschnittstelle und diskutiert die Antworten auf und die Benotung der Prüffragen einschließlich aller zusätzlichen Testprotokollnotizen. Die Einschätzung des Evaluators stützt sich im wesentlichen auf die Noten und die Begründungen der Noten. Die Interpretation schließt auch eine Bezugnahme auf andere Schnittstellen-Eigenschaften ein, z.B. bezüglich der Berücksichtigung des Prinzips der internen Schnittstellenkonsistenz.

Die Reihenfolge, in welcher die Antworten im Prüfbericht präsentiert werden, wird von Fall zu Fall variieren, in Abhängigkeit vom jeweiligen Evaluationszweck. Der Prüfbericht kann einerseits gemäß den technischen Komponenten geordnet werden. Solch eine Anordnung orientiert sich an den Bedürfnissen des Entwicklers, der präzise wissen möchte, wo ergonomische Defizite an der Benutzerschnittstelle lokalisiert sind. Für andere Bedürfnisse, beispielsweise wenn im Zuge einer Kaufentscheidung mehrere Software-Produkte bzgl. ihrer ergonomischen Qualität verglichen werden sollen, ist ein kriterienorientierter Prüfbericht angemessener.

In Tabelle 9-3 wird ein Beispiel einer Übersichtseinschätzung aufgeführt, die automatisch durch die Evaluations-Software generiert wird, nachdem die Evaluation abgeschlossen ist. Die Übersicht zeigt die durchschnittlichen Noten für jedes Kriterium. Die Kriterien sind sortiert nach ihrem Gewicht, welches in Schritt 1 durch die Rangliste bestimmt wurde. Als eine nützliche Zusatzinformation wird die Prozentverteilung jeder Note angegeben. So kann beispielsweise eine durchschnittliche Note von 3.0 das Resultat einer sehr guten (50% Note 1) und einer sehr schlechten (50% Note 5) Einschätzung sein, aber sie kann ebenfalls auf einer 100 %igen Einschätzung der Note 3 basieren. Es ist für den Evaluator sehr nützlich zu wissen, ob die durchschnittliche Note aus einer extremen oder einer ausbalancierten Realisierung einer Benutzerschnittstelleneigenschaft resultiert.

9.5.4 Evaluations-Software

EVADIS II bietet Computerunterstützung bei der Durchführung der Evaluationsprozedur. Dazu wurde eine Evaluations-Software in Clipper implementiert, die unter DOS auf einem IBM-kompatiblen PC läuft. Für die computerunterstützte Evaluation sind zwei Rechner erforderlich. Ein Rechner für die zu evaluierende Software und ein Rechner für die Evaluations-Software (z.B. ein Laptop). Die Evaluations-Software unterstützt den Evaluator bei der Erstellung der Prüfaufgaben (Texteditor mit vorgefertigten Beispielprüfaufgaben) und präsentiert in Form eines interaktiven Fragebogens alle Prüffragen in der Reihenfolge der zu bearbeitenden Prüfaufgabe. Der Evaluator hat die entsprechenden Antworten anzukreuzen bzw. zu ergänzen, eine Note zu vergeben, diese Note zu begründen

Tab. 9-3: Beispielübersicht über eine Notenverteilung

Kriterium	Gewichtung	Durchschnitt	Prozentverteilung der Bewertung					
			1	2	3	4	5	
Nr. Name								
02 Aufgabenangemessenheit	hoch	3.5	0.0	0.0	50.0	50.0	0.0	
05 Erwartungskonformität	hoch	3.6	0.0	0.0	40.0	60.0	0.0	
01 Verfügbarkeit	hoch	2.5	16.7	33.3	33.3	16.7	0.0	
04 Selbstbeschreibungsfähigkeit	mittel	1.5	50.0	50.0	0.0	0.0	0.0	
11 Datenschutz/Datensicherheit	mittel	2.3	0.0	66.7	33.3	0.0	0.0	
09 Steuerbarkeit	mittel	3.6	0.0	0.0	40.0	60.0	0.0	
06 Fehlerrobustheit	mittel	3.0	50.0	0.0	0.0	0.0	50.0	
08 Individualisierbarkeit	gering	3.0	0.0	0.0	100.0	0.0	0.0	
10 Kooperations- und Kommunikationsförderlichkeit	gering	0.0	0.0	50.0	0.0	50.0	4.0	
03 Übersichtlichkeit	gering	1.0	100.0	0.0	0.0	0.0	0.0	
07 Erlernbarkeit	gering	1.5	50.0	50.0	0.0	0.0	0.0	

und evtl. eine Notiz einzutragen. Nachdem die Prüfaufgabe abgeschlossen ist, erstellt die Evaluations-Software automatisch eine Übersichtseinschätzung der zu evaluierenden Benutzerschnittstelle. Es werden dabei Durchschnittswerte für alle Noten der ergonomischen Kriterien errechnet und Sortierungen der Prüffragen entweder nach technischen Komponenten oder nach ergonomischen Kriterien vorgenommen. Damit erfährt der Evaluator eine wesentliche Unterstützung bei der Erstellung des Prüfberichtes. Diese wird zusätzlich durch die Bereitstellung der Prüfberichtsstruktur als Textdatei erleichtert. Durch den Einsatz der Evaluations-Software wird der Evaluator von Routinearbeit befreit und kann daher seine Aktivitäten auf den eigentlichen Evaluationsprozeß konzentrieren.

9.5.5 Typische Benutzer von EVADIS II

Typische Benutzer von EVADIS II sind Softwareentwickler oder Anbieter von Bürosoftware (Einsatzgebiet: software-ergonomische Qualitätssicherung) sowie Organisationen, die im Begriff sind, Bürosoftware einzusetzen (Einsatzgebiet: Auswahlentscheidung). Auch Unternehmensberater und gewerkschaftliche Beratungsinstitutionen setzen EVADIS II ein (Einsatzgebiet: Beratung).

Ein anderer wichtiger Einsatzbereich von EVADIS II ist die Ausbildung. Eine Reihe von Universitäten (beispielsweise Universität Wien, Koblenz, Dresden und Berlin) benutzen EVADIS als ein Lehrmittel für die Ausbildung von Studenten der Informatik bzw. Wirtschaftsinformatik. Diese Universitäten berichten über gute Erfahrungen mit der Benutzung von EVADIS im Rahmen von Praktika. Die Studenten werden durch die Anwendung von EVADIS II in die Lage versetzt, die grundlegenden Konzepte der Software-Ergonomie sowie den praktischen Einsatz einer Evaluationsmethoden zu erlernen. Dabei spielt die Evaluations-Software als Medium einer „spielerischen Wissensvermittlung“ eine wichtige Rolle.

9.5.6 Schwerpunkte und Grenzen

Im Abschnitt 9.4 wurden Kriterien definiert, die die Einsetzbarkeit eines Evaluationsverfahrens im Entwicklungsprozeß bestimmten. Sie werden im folgenden dazu verwendet, Einsatzschwerpunkte und Grenzen von EVADIS II aufzuzeigen.

EVADIS II basiert schwerpunktmäßig auf einem methodengeleiteten Expertenurteil, so daß der Evaluationsaufwand (z.B. Zeit, notwendige Voraussetzungen) beschränkt werden kann. Außerdem bietet EVADIS II Computerunterstützung, die den Routineeinsatz erleichtern kann.

Eine wesentliche Voraussetzung für den erfolgreichen Einsatz ist, daß ein Experte mit tiefgehenden Kenntnissen der Software-Ergonomie verfügbar ist. Als weitere Voraussetzung benötigt EVADIS II eine ablauffähige Software sowie Zugang zu realen Benutzern und Aufgaben, so daß der Einsatzzeitpunkt für die Evaluation im Software-Entwicklungsprozeß nach der Analysephase (nach einer Aufgabenanalyse und Erfassung der Benutzereigenschaften) und der Designphase (z.B. nach Entwurf eines Prototyps) liegt. Deshalb kann EVADIS II nicht in der Spezifikationsphase des Softwareentwicklungsprozesses genutzt werden, und es sollte ein prototyping-orientierter Entwicklungsansatz gewählt werden. Es liegt auf der Hand, daß EVADIS II für die nachträgliche Evaluation, beispielsweise für die Evaluation von Standardsoftware-Produkten im Zuge einer Kaufentscheidung, benutzt werden kann.

Der Evaluationsschwerpunkt von EVADIS II liegt auf der Software. Daher beziehen sich Art und Umfang der Probleme, die mit EVADIS II entdeckt werden können, im wesentlichen auf die Benutzbarkeit der Software und nicht auf die Qualität der Arbeit oder das Benutzerverhalten.

Der Grad der Meßqualität wird durch eine detaillierte Evaluationsvorgabe, die den gesamten Evaluationsprozeß beschreibt, bestimmt. Damit sollen Unterschiede in der Einschätzung zwischen verschiedenen Evaluatoren reduziert werden. Trotz-

dem kann das Evaluationsergebnis in einem gewissen Maße durch das subjektive Urteil des Experten geprägt sein, beispielsweise in bezug auf die Benotung sowie auf die Gewichtung einzelner Kriterien. Bisher wurden keine experimentellen Tests durchgeführt, um die Validität und Reliabilität von EVADIS II im test-theoretischen Sinne zu prüfen.

9.6 Zusammenfassung

Aufgrund der zunehmenden Bedeutung der „Benutzerfreundlichkeit“ als Marketingargument und vor dem Hintergrund der EG-Richtlinie bezüglich der Mindestanforderungen an Sicherheit und Gesundheit für Bildschirmarbeit /EWG 90/ gibt es einen steigenden Bedarf an praktisch einsetzbaren Evaluationsmethoden für Benutzerschnittstellen. Dabei setzt sich zunehmend die Einsicht durch, daß im Zuge der Evaluation von Benutzerschnittstellen auch Faktoren wie vorhandene Benutzereigenschaften und Aufgaben sowie deren Einbettung in den Organisationskontext zu berücksichtigen sind. Die Anforderungen einer solch umfassenden Evaluation bilden den Rahmen für die Konstruktion oder Selektion von angemessenen Methoden. Eine Klassifikation vorhandener Methoden zeigt, daß nicht eine beste Evaluationsmethode existiert, sondern daß eine Kombination von verschiedenen Evaluationsmethoden erforderlich ist.

Die neue Evaluationsmethode EVADIS II, die hier als Beispiel näher präsentiert wurde, basiert auf einer solchen Methodenkombination. EVADIS II überwindet damit einige typische Beschränkungen der bekannten Evaluationsmethoden, insbesondere berücksichtigt sie die Aufgaben, den Benutzer und den Organisationskontext und bietet eine Software-Unterstützung beim Einsatz des Evaluationsverfahrens.

Literatur

/Apple 87/

Apple: *Human Computer Interface Guidelines: The Apple Desktop Interface*, Wokingham: Addison-Wesley, 1987

/Baitsch et al. 89/

Baitsch, Ch., Katz, Ch., Spinas, Ph., E. Ulich: *Computerunterstützte Büroarbeit*, Zürich: Verlag der Fachvereine, 1989

/Bonitz 89/

Bonitz, D.: *Verfahren zur Beschreibung und Bewertung von Arbeitstätigkeiten (VBBA) (Entwurf)*, Gesamthochschule Kassel, 1989

/Bonitz et al. 88/

Bonitz, D., Nachreiner, F., Benz, C., Wäger, M.: *Zur Veränderung von Tätigkeitsstrukturen und Arbeitsinhalten durch den Einsatz von Rechnern*, Zeitschrift für Arbeitswissenschaft, 4, 1988, 211-221

/Card, Moran, Newell 83/

Card, S., Moran, T. und Newell, A.: *The Psychology of Human-Computer Interaction*. Hillsdale, London: Lawrence Erlbaum, 1983

/Clegg et al. 88/

Clegg, C.W., Warr, P., Green, T., Monk, A., Kemp, N., Allison, G., Lansdale, M.: *People and Computers – How to Evaluate Your Company's New Technology*, Chichester: Ellis Horwood, 1988

/DIN 88/

DIN 66 234 Teil 8: *Bildschirmarbeitsplätze, Grundsätze der Dialoggestaltung*, Berlin: Beuth-Verlag, 1988

/Dunckel 89/

Dunckel, H.: *Arbeitspsychologische Kriterien zur Beurteilung und Gestaltung von Arbeitsaufgaben im Zusammenhang mit EDV-Systemen*, in: S. Maaß, H. Oberquelle (Hrsg.), *Software-Ergonomie '89 – Aufgabenorientierte Systemgestaltung und Funktionalität*, Stuttgart: Teubner, 1989, 69-79

/Dzida 83/

Dzida, W.: *Das IFIP-Modell für Benutzerschnittstellen*, Office-Management, Sonderheft, 31, 1983, 6-8

/Eason 82/

Eason, K. D.: *The Process of Introducing Information Technology*, Behaviour and Information Technology, 1, 1982, 197-213

/ETH-LAO ;86/

ETH-LAO: *Fragebogen zur Beurteilung von Dialog-Bildschirmssystemen*, Zürich: ETH Zürich, Lehrstuhl für Arbeits- und Organisationspsychologie, 1986

/EWG 90/

EWG: *Richtlinie des Rates vom 29. Mai 1990 über die Mindestvorschriften bezüglich der Sicherheit und des Gesundheitsschutzes bei der Arbeit an Bildschirmgeräten* (Fünfte Einzelrichtlinie im Sinne von Artikel 16 Absatz 1 der Richtlinie 89/391/EWG), 90/270/EWG, Amtsblatt der Europäischen Gemeinschaft, Nr. L 156/14, 21.6.1990

/Fährnich 87/

Fährnich, K.-P.: *Software-Ergonomie*, München: Oldenbourg, 1987

/Floyd 84/

Floyd, C.: *A Systematic Look at Prototyping*, in: R. Budde, K. Kuhlenkamp, L. Mathiasen, H. Züllighoven (Hrsg.), *Approaches to Prototyping*, Berlin: Springer, 1984

/Frese, Brodbeck 89/

Frese, M., Brodbeck F.: *Computer in Büro und Verwaltung*, Berlin: Springer, 1989

/Hammond et al. 85/

Hammond, N., Hinton, G., Barnard, P., MacLean, A., Long, J., Whitefield, A.: *Evaluating the Interface of a Document Processor: A Comparison of Expert Judgement and User Observation*, in: B. Shackel (Hrsg.), *Human-Computer Interaction – INTERACT '84*, Chichester: Elsevier Science Publishers, 1985, 725-729

/Hoyos et al. 90/

Hoyos, C., Müller-Holz auf der Heide, B., Hacker, S., Bartsch, T.: *PROTOS Menschengerechte Gestaltung von Bürokommunikationssystemen: Entwicklung von Methoden zur Herstellung und Bewertung von Prototypen für Benutzeroberflächen*, Zwischenbericht 8/90, München: TU München, Inst. f. Psychologie u. Erziehungswissenschaften, Lehrstuhl für Psychologie, 1990

/IBM 89a/

IBM: *Systems Application Architecture, Common User Access, Basic Interface Design Guide*, IBM Corporation, 1989

/IBM 89b/

IBM: *Systems Application Architecture, Common User Access, Advanced Interface Design Guide*, IBM Corporation, 1989

/IBM 91a/

IBM: *Systems Application Architecture, Common User Access, Guide to User Interface Design*, IBM Corporation, 1991

/IBM ;91b/

IBM: *Systems Application Architecture, Common User Access, Advanced Interface Design Reference*, IBM Corporation, 1991

/ISO 89/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 2, Guidance on Task Requirements, International Standard, November 1989

/ISO 91a/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 1, General Introduction, International Standard, November 1991

/ISO 91b/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 10, Dialogue Principles, Committee Draft, September 1991

/ISO 91c/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 13, User Guidance, Working Draft, August 1991

/ISO 91d/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 14, Menu Dialogues, Draft International Standard, 1991

/ISO 91e/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 15, Command Dialogues, Working Draft, Juli 1991

/ISO 91f/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 16, Direct Manipulation Dialogues, Working Paper, Februar 1991

/ISO 92a/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 11, Usability (Principles), Committee Draft, Januar 1992

/ISO 92b/

ISO 9241 *Ergonomic Requirements for Office Work with Visual Display Terminals*, Part 12, Principles for Presentation of Information, Working Draft, 1992

/Kishi, Kinoc 91/

Kishi, N., Kinoc, Y.: *Assessing Usability Evaluation Methods in a Software Development Process*, in: H.-J. Bullinger (Hrsg.), *Human Aspects in Computing: Design and Use of Interactive Systems and Work with Terminals*, Chichester: Elsevier Science Publishers, 1991, 597-601

/Koch, Reiterer, Tjoa 91/

Koch, M., Reiterer, H., Tjoa, A.: *Software-Ergonomie, Gestaltung von EDV-Systemen – Kriterien, Methoden und Werkzeuge*, Wien, New York: Springer, 1991

/Lang, Peters 88/

Lang, J. und Peters, H.: *Erhebung ergonomischer Anforderungen an Software, die überprüfbar und arbeitswissenschaftlich abgesichert sind*, Institut für Software-Ergonomie, München: TÜV Bayern, 1988

/Mambrey, Oppermann, Tepper 86/

Mambrey, P., Oppermann, R., Tepper, A.: *Computer und Partizipation. Ergebnisse zur Gestaltungs- und Handlungspotentialen*, Opladen: Westdeutscher Verlag, 1986

/MITRE 91/

MITRE: *Dynamic Rules for User Interface Design, DRUID, Version 2.0*, Bedford: MITRE Corporation, 1991

/Moll 87/

Moll Th.: *Über Methoden zur Analyse und Evaluation interaktiver Computersysteme*, in: Fähnrich, K.-P. (Hrsg.), *Software-Ergonomie*, München: Oldenbourg, 1987, 179-190

/MUSIC 92/

MUSIC: *Metrics for Usability Standards in Computing*, (ESPRIT II Project 5429), Productinformation, Teddington: National Physical Laboratory, UK, 1992

/Norman, Shneiderman 89/

Norman, K., Shneiderman, B.: *Questionnaire for User Interaction Satisfaction (QUIS 5.0)*, University of Maryland: HCI-Lab, College Park, 1989

/Olle et al. 88/

Olle, T., Hagelstein, J., MacDonald, I., Rolland, C., Sol, H., Assche, F.V.: *Information Systems Methodologies*, Wokingham: Addison-Wesley, 1988