

Ergonomie-Reviews und Usability-Testing als Beratungs- und Qualifizierungsinstrumente

Peter Ansorge und Uwe Haupt

Universität Bremen, Technologie-Zentrum Informatik,
Institut für Software-Ergonomie und Informationsmanagement¹

Zusammenfassung

Das Projekt „Software-Ergonomie-Transfer“ (SET) berät auf regionaler Ebene in einem Verbundvorhaben kleine und mittlere Softwarehäuser zur Fragen der Software-Ergonomie. Die dabei in allen Phasen der Software-Entwicklung eingesetzten punktuellen Reviews und Usability-Tests fungieren keineswegs nur als Prüfverfahren, sondern zusätzlich als Qualifizierungsinstrumente für die Software-Entwickler.

Für die Phasen der Anforderungsdefinition, des Entwurfs und der Implementierung wurden Reviews mit einer Kombination von Walkthrough und checklistenbasierter Analyse erfolgreich erprobt. Sie können Usability-Tests, in denen weitere, in der Regel schwerwiegendere ergonomische Defizite zutage treten, vorbereiten und effizienter gestalten.

1 Ausgangslage: Veränderung des Qualitätsbewußtseins

Im Rahmen allgemeiner Qualitätssicherungsverfahren (Stichwort ISO 9000) werden Bewertungsverfahren auch von kleinen und mittleren Softwarehäusern zunehmend als unumgänglich angesehen und akzeptiert. Allerdings stehen hierbei Tests und Erprobungen auf der *funktionalen* Ebene immer noch Vordergrund, während Ergonomie-Überprüfungen weitestgehend vernachlässigt werden. Mit steigender, aber immer ähnlicher werdender Funktionalität von Programmen wird die Benutzbarkeit der Software als wichtiges Qualitätsmerkmal erkannt und so zu einem die Kaufentscheidung wesentlich beeinflussenden Faktor. Die Forderungen der EU-Bildschirmrichtlinie, der Arbeitsschutzgesetze, der Bildschirmarbeitsverordnung und der hierdurch referenzierten Normen werden – zwar noch zögerlich, aber in wachsendem Maße – von Anwendern und Benutzern gegenüber Softwareherstellern geltend gemacht. Ergonomische Konformitätsnachweise, z. B. als Zertifizierungen, werden am Abschluß eines Software-Entwicklungsprozesses gefordert und von Herstellern nicht nur passiv als notwendiges Übel hingenommen, sondern aktiv als Vermarktungshilfe akzeptiert und benötigt.

Unabhängig von der Motivation werden Produktbewertungen als fester Bestandteil einer Software-Entwicklung im Produktionsplan als Meilenstein berücksichtigt, auf den hingearbeitet wird. Anders verhält es sich mit ergonomischen Schulungs- und Qualifizierungsvorhaben für Software-Entwickler, die zwar nach Überblicksveranstaltungen als durchaus notwendig angesehen werden, deren Durchführung aber nicht selten dem allgemeinen Termindruck der Softwarebranche anheim fällt.

¹ Diese Arbeit entstand im Rahmen des Projektes *SOFTWARE-ERGONOMIE-TRANSFER (SET)*, gefördert vom Landesprogramm Arbeit und Technik des Landes Bremen.

Eine Kombination von Reviews und Qualifizierungsmaßnahmen, von Akzeptiertem und Notwendigem, erweist sich nicht nur für kleine und mittlere Softwarehäuser als effektives und effizientes Verfahren zur Steigerung der ergonomischen Produktqualität. Hierbei sind allerdings einige Besonderheiten zu beachten, die einerseits durch die Beteiligten, andererseits durch die Verfahren bedingt sind.

2 Software-Ergonomie-Transfer

Im Forschungsprojekt Software-Ergonomie-Transfer (SET) der Universität Bremen, Informatik, werden Möglichkeiten des Transfers software-ergonomischer und arbeitswissenschaftlicher Erkenntnisse untersucht und erprobt. Adressaten der Transfermaßnahmen sind Benutzer, Anwender und insbesondere die Hersteller von Software. Innerhalb der dreijährigen Projektlaufzeit sollen die Grundlagen für eine dauerhaft Beratungseinrichtung für Software-Ergonomie-Transfer gelegt werden.

2.1 Akteure

Die Kooperation von Softwareherstellern und Software-Ergonomie-Spezialisten birgt eine Reihe von Problemen und setzt daher eine präzise Kenntnis der jeweiligen Bedingungen bei den einzelnen Akteuren voraus [vgl. auch 9]. Dies gilt insbesondere dann, wenn Software-Ergonomie-Berater als Externe den Software-Entwicklern gegenüberreten, wie etwa bei kleinen und mittleren Unternehmen, die sich noch keine eigenen Ergonomien leisten (können), und wenn durch die Kontroll- und Kritikfunktion eines Reviews Befürchtungen und Abwehrhaltungen auf Seiten der Software-Entwickler induziert werden könnten.

Die kleinen und mittelgroßen Unternehmen der Softwarebranche wickeln typischerweise kleine und mittelgroße Projekte ab, sehr häufig als Branchenlösungen. Langjährige und persönliche Kontakte zur Anwenderorganisation sind als Stärken zu nennen. Wesentliche Kennzeichen sind:

- informelle Kooperationsverfahren zwischen den beteiligten Entwicklern und teilweise auch in der Kundenbeziehung,
- geringer Einsatz analytischer Verfahren,
- Dominanz funktionaler Sichten,
- geringe ergonomische Kenntnisse der Entwickler und geringe Fähigkeiten zur Operationalisierung ergonomischer Ansprüche,
- ständige Zeitknappheit,
- praktisch nur schwach ausgeprägte Phasenabgrenzung,
- weitgehenden Verzicht auf Dokumentation.

Externe Berater neigen dazu, vor Vielschichtigkeit und Facettenreichtum der Software-Ergonomie die akuten Nöte und Bedürfnisse der Entwickler aus den Augen zu verlieren. Sie lassen sich charakterisieren durch:

- mangelnde Vertrautheit mit der aktuellen Entwicklungsaufgabe,
- nicht per se vorhandenem Wissen um die zu lösenden Arbeitsaufgaben,
- keine Kenntnisse über Anwender und Benutzer,
- geringe Kenntnisse über den Wissensstand der Entwickler,

- breites Wissen zu allgemeinen ergonomischen Fragen, Vorschriften und Regeln,
- Erfahrung durch unternehmensübergreifende Betrachtungen.

2.2 Maßnahmen

Maßnahmen von SET zum Transfer software-ergonomischer und arbeitswissenschaftlicher Erkenntnisse sind u. a.:

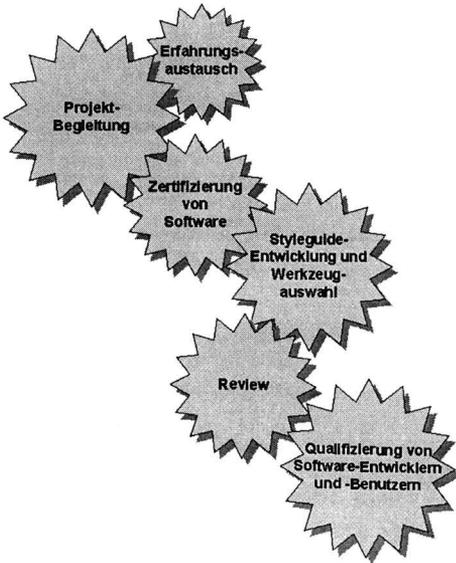


Abb. 1: Maßnahmen zum Software-Ergonomie-Transfer

- Review bereits erstellter Softwareprodukte in Bezug auf Formularlayout, Dialoggestaltung und arbeitsorganisatorische Erfordernisse
- Vermittlung von Review-Strategien für den späteren selbständigen Review von Softwareprodukten durch die Hersteller
- Projektbegleitende Beratung zur software-ergonomischen Erstellung und Gestaltung eines neuen Softwareproduktes
- Entwicklung von software-ergonomischen unternehmensspezifischen Styleguides und Beratung und Auswahl zum Einsatz von Entwicklungswerkzeugen
- Qualifizierung von Mitarbeitern in firmeninternen Schulungen oder öffentlichen Seminaren
- Seminare zur Verbesserung der innerbetrieblichen Kommunikationsstrukturen

Bei den zentralen Transfermaßnahmen Qualifizierung und Review treten im Kontext mit kleinen und mittelständischen Softwareherstellern folgende Probleme auf:

2.2.1 Probleme klassischer Qualifizierung

Betriebliche Weiterbildung ist in kleinen und mittleren Unternehmen im Vergleich zu Großunternehmen sowohl bezüglich Quantität als auch hinsichtlich einer systematischen Planung nur schwach ausgeprägt. Es dominieren ad-hoc-Qualifizierungsmaßnahmen am Anfang von Entwicklungsaufträgen, in denen es um das Erlernen neuer Werkzeuge bzw. Sprachen geht. Selbstinitiiertes Lernen durch Lektüre von Handbüchern oder Ausprobieren wird häufig gar nicht als solches erkannt und spielt trotz der immens hohen – allerdings im allgemeinen Personalbudget versteckten – Kosten eine überproportional große Rolle. „Klassische Kurse“ werden wegen des überschaubaren zeitlichen, finanziellen und personellen Aufwands nur zögernd wahrgenommen. Hinzu kommen die relativ geringen Wirkungen derartiger Qualifizierungsmaßnahmen, die durch Streuverluste aufgrund nicht angepaßten Lernstoffes sowie durch die Überbetonung reiner Kenntnisvermittlung gegenüber der Aneignung von Fähigkeiten

ten verursacht wird. Durch die zeitliche Entkopplung von der Umsetzungsphase sinkt der Nutzen weiter.

Die ohnehin problematischen, aber in kleinen und mittleren Unternehmen vorherrschenden Formen der Weiterbildung sind für software-ergonomische Themen in besonderem Maße ungeeignet: Software-Ergonomie erfordert grundsätzlich andere Denk- und Herangehensweisen, die sich von den gewohnten, technisch dominierten Denkweisen der Software-Entwickler deutlich unterscheiden. Anschlußlernen ist somit nicht möglich. Damit wird die Einarbeitung ohne fremde Unterstützung besonders aufwendig. Hinzu kommt, daß auch heute noch selbst die Grundzüge der Software-Ergonomie nur an wenigen deutschsprachigen Hochschulen vermittelt werden.

2.2.2 Defizite von Reviews

Produktreviews, Audits und Bewertungen werden ex-post durchgeführt, um die Qualität von Produkten anhand von Vorschriften und Vorgaben *festzustellen* und sie insbesondere gegenüber Kunden *nachzuweisen*. Adressaten eines Reviews sind in der Regel also nicht die Software-Entwickler.

Voraussetzungen für Softwareprüfungen sind insbesondere das ablauffähige Programm, die Dokumentationen zum Programm, Prüfvorschriften, Anforderungen, Spezifikationen, Gesetze, Normen, Styleguides und der Prüfauftrag.

Abschließende Beurteilungen kommen für eine Einwirkung auf die Produktqualität zu spät. Verbesserungen lassen sich erst in Nachfolgeprodukten / -versionen realisieren oder durch kostenintensive Nacharbeit – was bei groben ergonomischen Fehlern mit einem Neuanfang gleichzusetzen ist.

Die Durchführung von fundierten Ergonomie-Überprüfungen stellt sich durch das Fehlen oder den mangelnden Bekanntheitsgrad geeigneter Ergonomie-Prüf- und Beratungseinrichtungen de facto als ein Problem dar. Methoden der Selbstkritik und -beurteilung sind zwar interessante Ansätze, dennoch kann auf unbeteiligte Prüfer nicht verzichtet werden. In kleinen und mittleren Softwarehäusern fehlt aber häufig selbst die Kapazität für hausinterne Bewertungen durch Unbeteiligte, die nicht der Gefahr der „Betriebsblindheit“ unterliegen.

3 Entwicklungsbegleitende Reviews

Die „Verbesserung der ergonomischen Qualität“ von Programmen setzt die Qualifizierung der Entwickler voraus [vgl. z. B. 14]. Um effizient wirksam werden zu können, muß fachliche Beratung an den konkreten Problemen in der Entwicklung aufsetzen; die Probleme lassen sich am besten an den entstehenden Programmen durch ergonomische Inspektionen ermitteln.

3.1 Informationsgewinnung für Prüfer – Qualifikation für Entwickler

DIN 66 285 [5] schreibt vor, ohne spezielle Unterlagen dürfe nicht geprüft werden, aber diese strenge Forderung, die ihre Berechtigung bei Testat-erteilenden Beurteilungen wie Zertifizierungen hat, ist wenig praxisgerecht, wenn es um Begutachtungen im Zusammenhang mit Beratungen und Schulungen geht. Realiter wird aber in fast allen kleinen und mittleren Ent-

wicklungsvorhaben auf die Erstellung spezieller Ergonomie-Anforderungen und -Spezifikationen („ErgoSpec“) bisher verzichtet.

Prüfer sind dennoch auf diese Angaben angewiesen. Die Erhebung dieser Informationen bei den Software-Entwicklern hat einen eigenständigen Wert als Qualifizierungsinstrument: Sie macht nicht nur Berater/Prüfer mit dem entstehenden Programm vertraut und versetzt sie in die Lage, eine Minimal-/Ersatzspezifikation anzufertigen, sondern verdeutlicht den Entwicklern, welche Punkte klärungsbedürftig, mindestens aber mißverständlich sind. Hier wirkt allein die Übermittlung der bisher möglicherweise nur implizit unterstellten Annahmen erkenntnisfördernd.

Neben den spezifischen Dokumenten müssen selbstverständlich allgemeingültige wissenschaftliche Erkenntnisse, Gesetze und Normen, ggf. plattformspezifische Regelungen („Betriebssystem-Styleguides“) herangezogen werden. Hier können aus der DIN EN ISO 9241 [6] abgeleitete und operationalisierte Kriteriensysteme Verwendung finden.

3.2 Formulierung einer Prüfaufgabe – Vertrauensbildung

In der Regel steht bei kleinen und mittleren Entwicklungsvorhaben zunächst die Unterstützung *einer* speziellen Arbeitsaufgabe im Vordergrund oder dominiert die Anwendung auch bei einer Weiterentwicklung und Erweiterung noch stark, so daß Externen der Zugang nicht durch übergroße Komplexität erschwert wird. Hieraus kristallisiert sich bei beharrlichem Nachfragen mindestens eine während des Entwicklungsprozesses als typisch angesehene Programmnutzung heraus. Leitfragen zur Ermittlung können dabei z. B. Fragen nach der geschätzten Nutzungshäufigkeit oder einer Beeinträchtigung bei Funktionsverlust sein. Als zweckmäßig zur Abschätzung der unterstellten Relevanz haben sich auch Erhebungen über die Menge zugrundeliegender / verwendeter (Papier)Formulare oder sonstiger Datenquellen bzw. über die Menge der Ausgaben herausgestellt.

Aus den erhaltenen Hinweisen über den Zweck und Anleitungen zur Programmnutzung (meist müssen während der Entwicklung noch besondere Vorkehrungen zur Nutzung auf Fremdrechnern vorgenommen werden) und einer Grobsichtung des Programms läßt sich ein Eindruck über den Funktionsumfang und die geplante Nutzung gewinnen. Danach besteht die Notwendigkeit, die Prüfung zu strukturieren, da sich eine Totaluntersuchung meist aus Kapazitätsmangel verbietet. Hierzu formuliert der Prüfer in Zusammenarbeit mit Anwendern und Benutzern eine *Prüfaufgabe*, mit der das Programm untersucht werden soll. Hierdurch wird eine Beliebigkeit ausgeschlossen; eine Annäherung an die spätere Nutzungssituation wird ermöglicht, und die Entwickler erhalten Vertrauen, daß nicht willkürlich überall nach Fehlern gesucht wird. Dabei wird festgelegt, welche Arbeitsaufgabe mit welchen Testdaten erledigt werden soll und welche Schritte als notwendig angesehen werden. Vorzugsweise handelt es sich um *einen* vertikalen Durchlauf durch das Programm. Die Vorgaben werden den Entwicklern vorab vorgelegt und mit ihnen abgestimmt. Nach deren Zustimmung – die hier besonders wichtig ist, um nicht möglicherweise in noch nicht implementierte Module zu gelangen – wird der Prüfplan festgelegt.

3.3 Prüfungsdurchführung

Es hat sich bewährt, die Prüfaufgabe zunächst einmal ohne jegliche Anmerkungen oder Aufzeichnungen vom Prüfer durcharbeiten zu lassen – auch Prüfer müssen sich mit dem System vertraut machen. Im zweiten Durchgang erfolgt die eigentliche Prüfung anhand der gestellten Aufgabe. Hierzu werden die Arbeitsschritte minutiös dokumentiert – sinnvollerweise durch technische Unterstützung der Verlaufprotokollierung. Hier kommen Verfahren zum Einsatz, wie sie üblicherweise bei Protokollierungen des Nutzungsverhaltens in Usability-Tests eingesetzt werden, z. B. statische Dokumentation durch „Screenshots“, dynamische Aufzeichnung der Präsentationen durch „Screengraber“, Aufzeichnung der Interaktionen mit Registrierung der Eingaben und Festhalten einer „Mausspur“ (vgl. Abb. 2), Audioanmerkungen, Videoaufzeichnung. Anders als bei der Beobachtung späterer Benutzer dienen diese Instrumente lediglich der eigenen Auswertung durch den Prüfer selbst bzw. der Illustration seines Berichts.

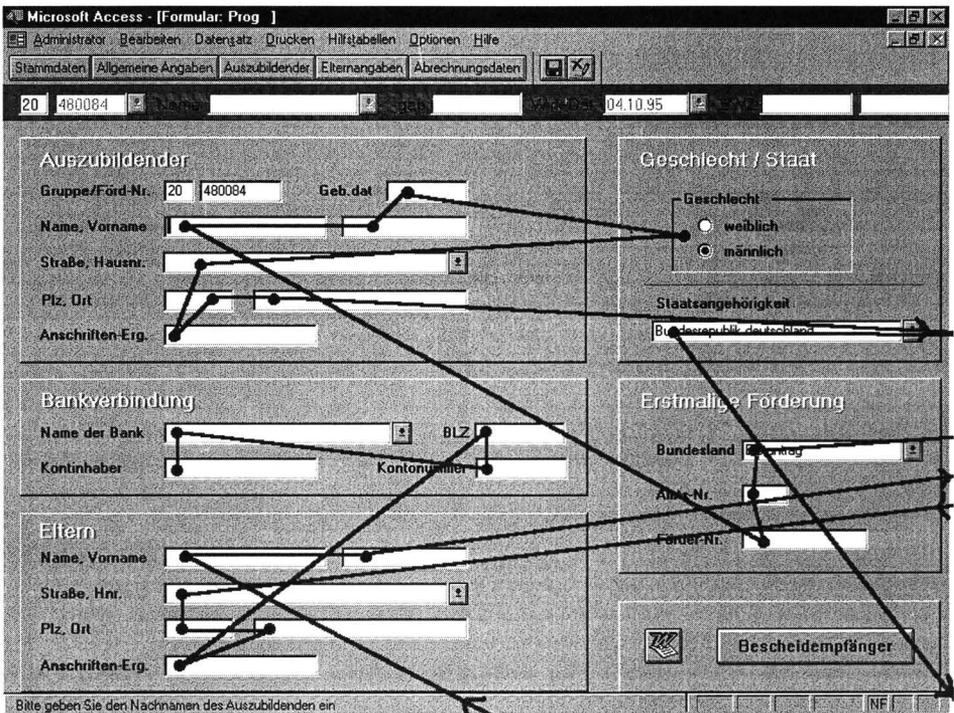


Abb. 2: Ergebnis eines Reviews

Sachbearbeiter hatten die Aufgabe, Daten von einem (bundeseinheitlichen) Papierformular in den Rechner einzugeben. Die Feldanordnung der Vorlage und des Bildschirmformulars unterschieden sich deutlich. Die Eingabe der Daten in der Reihenfolge des Papierformulars erforderte einen Navigation über den Bildschirm entlang des oben markierten, komplizierten Weges.

Die Bewertung erfolgt anhand eines Kriteriensystems. Der Prüfer dokumentiert jeden durchgeführten Schritt in einem Prüfbericht. Insbesondere sind Formular-/Maskenübergänge, Feldpositionierungen, Eingaben und sonstige Interaktionen festzuhalten. Als zweckmäßig hat sich

zeigt, die Kriterienliste („Checkliste“) zunächst zu *jedem* neuen Bildschirmformular einmal zu durchlaufen und sämtliche Prüfkriterien zu beantworten, wobei festgestellte Verstöße begründet zu protokollieren sind. Zur Unterstützung dieser Tätigkeit hat sich der Einsatz von Datenbanksystemen bewährt, bei dem einerseits die Kriterien durch Beispiele konkretisiert werden, andererseits alle bisher jemals zu diesem Kriterium gemachten Angaben (auch bei anderen zu prüfenden Programmen) abgerufen werden können. Hierbei werden interpersonale Bewertungsspielräume relativiert, außerdem entsteht mit zunehmender Prüfungsanzahl entsprechende Bewertungssicherheit.

Bei abschließenden Bewertungen, deren Ergebnisse sich vorzugsweise an Dritte wenden, sind Aussagen wie „Kriterium erfüllt“ bzw. „nicht erfüllt“ vollkommen ausreichend. Zur Qualifikation werden aber weitere Elemente benötigt: So ist sehr exakt zu benennen, welche Kriterien aus welchen Gründen nicht erfüllt werden. Hierbei ist auf die entsprechenden Stellen in den der Prüfung zugrundeliegenden Dokumenten wie Normen und Styleguides zu verweisen – diese Dokumente sollen ja später von vornherein beachtet werden, sind also Qualifizierungsgegenstand.

3.4 Konstruktive Kritik

Beurteilungsfähigkeit setzt besondere Qualifikationen der Prüfer voraus. Einem erfahrenen Prüfer fällt nicht nur aufgrund von Checklisten ein Kriterienverstoß auf, sondern auch durch Kenntnis von anderen Programmen und durch eigene Nutzungserfahrungen. Es liegen also auch bei Prüfern, die mit der konkreten Arbeitsaufgabe weniger vertraut sind als die späteren Anwendungsspezialisten, bestimmte Erwartungshaltungen mindestens über weniger aufgabenabhängige Teile vor.

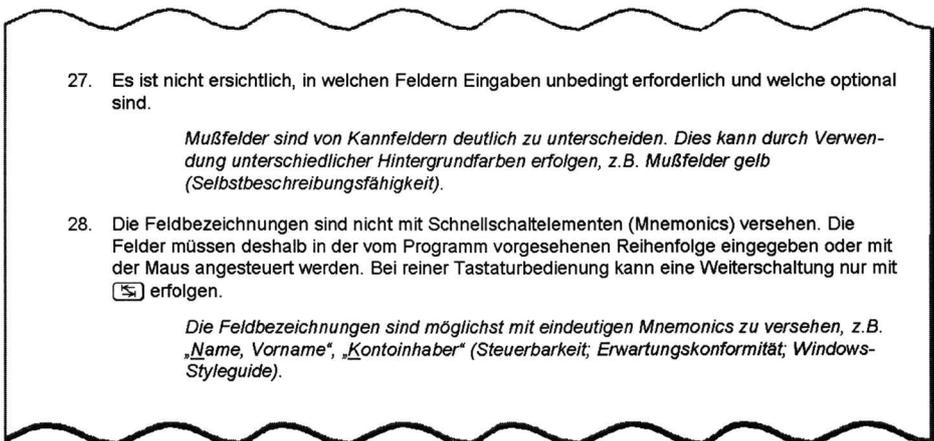


Abb. 3: Ausriß aus einem Prüerbericht

Hilfreich für die Korrektur von Fehlern ist also die Benennung der üblicherweise zu erwartenden Elemente der Benutzungsschnittstelle. Verfügen die Prüfer darüber hinaus auch über eine Gestaltungsfähigkeit, so können sie nicht nur analytisch Hinweise über den Grund von Feh-

lern („Warum?“) und auf das „Was?“, sondern auch konstruktiv über die Mittel („Wie?“) geben (vgl. Abb. 3). Werden Angaben zur ergonomischen Verbesserungen von Programmen angegeben, so wird mindestens verdeutlicht, daß es nicht nur den *einen* von den Entwicklern verfolgten Weg gibt. Wird die Alternative auch noch durch ablauffähige Prinziplösungen mindestens als machbar belegt, so wird die Zurückweisung, ergonomische Forderungen seien unrealistisch, von vornherein vermieden.

3.5 Präsentation und Vermittlung

Die Ergebnisse eines entwicklungsbegleitenden Reviews werden in einem kompakten Bericht, bestehend aus der Dokumentation des Walkthroughs anhand der Prüfaufgabe, einer checklistenbasierten Bewertung und einer exemplarischen Einzellösung eines zentralen Problems des geprüften Programmes zusammengefaßt. Die Präsentation dieser Dokumentation stellt einen zentralen Vermittlungsschritt innerhalb des Qualifizierungsvorhabens dar. In einer halbtägigen Präsentation des Reviews vor den Software-Entwicklern läßt sich sowohl an der Praxis der Entwickler orientiertes ergonomisches Fachwissen vermitteln als auch gleichzeitig ein konkreter Beitrag zum laufenden Entwicklungsprojekt leisten.

Reviewpräsentationen weisen ein erhebliches Konfliktpotential auf, zumal Schwächen der Software deutlich zutage treten, so daß Software-Entwickler zunächst eine vernichtende, unproduktive Kritik ihrer Arbeit befürchten. Die Diskussion kann nicht allein durch den Duktus der schriftlichen Dokumentation und der Präsentation von der Ebene einer Schuldzu- und -zurückweisung ferngehalten werden; vielmehr gelingt dies durch Konzentration auf konstruktive Elemente. Wider Erwarten beeinträchtigte die Beteiligung der zuständigen Geschäftsführer und auch der Pilotanwender nicht die produktive Auseinandersetzung.

Kritikbereitschaft wird durch die Darstellung von konkreten Lösungen und die Erläuterung möglicher Ursachen hergestellt: Häufig gelingt es, Folgefehler defizitärer Entwicklungswerkzeuge oder unzureichende Vorgaben der Anwender zu identifizieren oder anhand von Hintergrundmaterial aufzuzeigen, welcher Aufwand zur Herstellung ergonomisch geeigneterer Lösungen bei anderen Entwicklungsvorhaben notwendig war. Besonders einprägsam sind Beispiele durch Veränderungen des Originalprogramms, allerdings erfordert dies vom Reviewer nicht nur ein hohes Maß an Beurteilungs-, sondern auch an Gestaltungskompetenz.

3.6 Reaktionen auf Reviews

In den unterschiedlichen Ebenen der Softwarehäuser wurden gravierende Konsequenzen aus den von uns durchgeführten Reviews gezogen. Die monierten Defizite wurden von Entwicklern durchgängig ernst genommen und in der Regel unverzüglich beseitigt. Lediglich nach sehr spät im Entwicklungsverfahren ansetzenden Erstprüfungen konnten in wenigen Punkten Korrekturen nicht mehr durchgeführt werden, so daß Lerneffekte erst in Folgeprojekten wirksam wurden.

Auf der Ebene der Entscheidungsträger wurden – auch aufgrund des positiven Echos bei den Entwicklern – häufig deutlich weitreichendere Konsequenzen gezogen, die bis hin zur vollständigen Neuentwicklung einzelner Systeme reichten. Teilweise folgte auf den Review eine grundlegende Umstellung der Entwicklungsmethode z. B. auf objektorientierte Techniken, wobei unternehmensspezifische Klassenbibliotheken in Kooperation mit den Reviewern

erstellt wurden. Auch die Ergonomie der eigenen Werkzeuge wurde in Frage gestellt. Die Bereitschaft zur Beteiligung an systematischen Qualifikationsveranstaltungen wurde erhöht.

Innerhalb der Softwarehäuser traten durchweg positive Effekte auf: Das Verhältnis zwischen kundenorientierten Mitarbeitern – meist mit langjährigen Erfahrungen mit Vorversionen – und den Methodenspezialisten – häufig mit noch geringer Projekterfahrung – verbesserte sich nachhaltig. Auf bereits geplante Umorganisationen durch strenge Aufteilung in Vertrieb und Programmierung wurde zugunsten eines integrierten Entwicklungsteams verzichtet.

Gerade die Diskussion ergonomischer Fragen anhand der „eigenen“ Produkte befähigte die Führungskräfte, die Programmverbesserungen auch bei Kundenkontakten aufzugreifen und in Abgrenzung zu Mitbewerbern verkaufswirksam einzusetzen.

3.7 Reichweite des entwicklungsbegleitenden Reviews als Qualifizierungs- und Bewertungsinstrument

Kriterien wie Fehlertoleranz und Steuerbarkeit, eingeschränkt auch Individualisierbarkeit und Lernförderlichkeit, sowie Fragen der Anordnung und Codierung lassen sich weitgehend durch Experten abprüfen. Bei der Beurteilung der Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit und der Erwartungskonformität können Experten allerdings nicht mehr auf der Basis von allgemeingültigen ergonomischen Erkenntnissen auf Benutzerverhalten und -belastung rückschließen, so daß diese Aspekte unmittelbar mit Benutzern zusammen erhoben werden müssen.

Die punktuellen Reviews und deren Präsentationen ersetzen auch in der hier skizzierten Form keine software-ergonomischen Grundlagenschulungen, sondern bereiten sie vor. Mit vertretbarem zeitlichen Aufwand werden produktbezogene und aufgabenangemessene Empfehlungen gegeben, wodurch die Glaubwürdigkeit unterstrichen und die Akzeptanz der vertretenen Ansätze deutlich erhöht wird. Die Aufnahmebereitschaft für theoretische Grundlagen wird geweckt.

Viele Defizite können auch externe Experten nicht erkennen, insbesondere wenn keine oder nur geringe Angaben über Benutzer, Benutzungssituationen und organisatorisches, technisches und qualifikatorisches Umfeld vorliegen. Durch ihre umfangreiche Erfahrung im Umgang mit Rechnern verkennen Experten ebenso wie Entwickler schnell Probleme von Rechnernovizen und können fast nie beurteilen, wie Arbeitsgänge realiter ablaufen, ob nicht durch nichtformalisierbare Usancen falsche Dialogstrukturen und Werkzeugzuordnungen vorgenommen werden. Fragen danach werden durch einen Review provoziert, wobei von den Entwicklern erkannt wird, daß es einen „one-best-way“ nicht gibt und zur Vermeidung von Umwegentwicklungen eine frühzeitige Benutzerbeteiligung unverzichtbar ist. Benutzerbeteiligung darf sich aber nicht auf Ermittlung von Vorgaben beschränken, sondern muß zur Validierung auch Usability-Tests einbeziehen.

Eine benutzerzentrierte und aufgabenangemessene Problemlösung hat weitreichende Konsequenzen für die strategische Ausrichtung von Softwarehäusern: Aus den Reviews wird deutlich, daß der Entwicklungsprozeß neu strukturiert werden muß, daß wesentliche qualitäts- und damit wertsteigernde Aktivitäten nicht mehr in der Programmierphase, sondern in vorgelagerten Analysephasen zu erbringen sind. Da die Budgets in der Regel festgelegt sind bzw. nicht beliebig erweitert werden können, müssen Maßnahmen zur Kostenumverteilung inner-

halb der Projekte erfolgen, z. B. durch Einsatz effizienterer Entwicklungswerkzeuge. Der dadurch geschaffene finanzielle Spielraum für Analysearbeiten positioniert Softwarehäuser in neuen, dem Wettbewerb weniger stark ausgesetzten Kompetenzfeldern.

Verstanden sich Softwarehäuser bis dahin nicht selten als reine „Software-Ersteller“, so wurden ihre Aktivitäten, provoziert durch die Auseinandersetzung mit „Aufgabenangemessenheit“ und den konkreten Anwendungssituationen, zwangsläufig erweitert. Beratungsdienstleistungen werden wesentlich stärker mit den Kunden integriert als dies bei einfachen Zulieferungen der Fall ist. Hierdurch können weitergehende ergonomische Forderungen berücksichtigt werden.

4 Usability-Test – Einbeziehen der Benutzer

Eine Grunderkenntnis der Software-Ergonomie ist, daß sie ohne Beteiligung der Benutzerinnen und Benutzer kaum zufriedenstellend umgesetzt werden kann. Ein Transfer-Konzept darf sich daher nicht auf die Unterstützung bei der Lösung eines „ergonomischen Programmierproblems“ durch Expertenberatung beschränken. Es muß vielmehr die arbeitswissenschaftliche Begleitung von der Aufgabenanalyse über die software-ergonomische Spezifikation bis zur Realisierung und Einführung umfassen. Dabei wird deutlich, daß auch die Kunden eines Softwarehauses und die dort arbeitenden späteren Benutzer der Software in den Transferprozeß einbezogen werden müssen.

Eine wesentliche Rolle spielt hierbei die Diskussion der Prüfaufgabe mit den Entwicklern, bei der allen Beteiligten deutlich wird, daß eine Perspektivübernahme – also das Hineinversetzen in die Benutzer – notwendig, aber nicht vollends möglich ist. Daher sind Usability-Tests, also die direkte Konfrontation der Benutzer mit Programmen (oder Vorläufern) unverzichtbar.

4.1 Stand der Forschung

Mit expertengestützten Prüfmethoden wie heuristischer Evaluation [19] oder „Cognitive Walkthrough“ [18] kann eine Vielzahl ergonomischer Gestaltungsfehler aufgedeckt werden. So erwies sich in Untersuchungen von JEFFRIES, MILLER, WHARTON und UYEDA [17] die expertengestützte, heuristische Evaluation als diejenige Methode, mit der die größte Anzahl ergonomischer Gestaltungsfehler entdeckt werden konnte. Die unmittelbare Einbeziehung der Benutzer in die Softwareüberprüfung (Usability-Testing) fördert demgegenüber in der Regel zwar eine geringere Anzahl von ergonomischen Gestaltungsfehlern zutage, doch sind die beim Usability-Test entdeckten Fehler schwerwiegender und „globaler“. Sie treten häufiger auf als die von den Experten entdeckten Fehler. Die Hälfte der bei Usability-Tests gefundenen Fehler blieb bei Anwendung expertenbasierter Prüfmethoden unentdeckt [vgl. 16]. Daraus ergibt sich die Notwendigkeit zur Kombination expertengestützter Verfahren und Usability-Tests.

Ausschließlich expertenorientierte Verfahren wie z. B. EVADIS II [20] und die ausschließlich auf Benutzerbefragungen basierenden Verfahren wie QUIS [24], SUMI [21], ISONORM 9241-10 [22] und das Verfahren von RAVDEN und JOHNSON [23] werden um kombinierte Verfahren ergänzt, die expertenbasierte Evaluationen mit Benutzerbefragungen und Usability-Tests im Labor verbinden (z. B. die Verfahren von CLEGG et. al [2] und von ENGLISCH [10] sowie der ERGOguide von DZIDA, WIETHOFF und ARNOLD [8]).

Im Rahmen von Arbeits- und Aufgabenanalysen sollen zwar die aus der Forderung nach Aufgabenangemessenheit und Erwartungskonformität resultierenden Anforderungen ex-ante erhoben werden, jedoch erweist sich dieser Ansatz für sich allein als unzureichend [vgl. 11]. Er kann demnach lediglich am Anfang eines Systementwicklungsprozesses stehen, der in seinem weiteren Verlauf auch benutzerorientierte Evaluationsschritte beinhaltet.

Die in der ISO 9000, Teil 3, dem Leitfaden für die Anwendung von ISO 9001 auf die Entwicklung, Lieferung und Wartung von Software, niedergelegten Regelungen beziehen „usability tests“ als Testempfehlung ausdrücklich mit ein [15, S. 19]; allerdings fehlen in der Norm hierzu noch Maßnahmen zur Umsetzung. Hier besteht ein akuter Handlungsbedarf bezüglich eines wissenschaftlich fundierten Methodentransfers.

Das Kriterium Benutzbarkeit wird in den derzeit gängigen, ausschließlich produktbezogenen Qualitätssicherungsmethoden für die Software-Entwicklung nicht ausreichend berücksichtigt. Beispielsweise beschränkt DUNN [7, S. 179f] Benutzbarkeitstests auf die Auslieferung von Alpha- und Betaversionen. Prüfungen nach DIN 66 285 umfassen nur Teilaspekte der DIN 66 234, Teil 8 (resp. DIN EN ISO 9241, Teil 10), und in den Verfahren nach DGQ-NTG [3] sowie nach ASAM, DRENKHARDT und MAIER [1] nimmt der Teil zur Benutzbarkeit nur einen sehr geringen Raum ein [vgl. 13, S. 49-52].

4.2 Usability-Tests – vorbereitet durch Experten-Reviews

Im Vergleich zu entwicklungsbegleitenden Reviews sind Usability-Tests hinsichtlich des zeitlichen, finanziellen und personellen Ressourcenbedarfs sehr aufwendig. Dieser Aufwand kann bei systematischer Vorbereitung insbesondere durch einen entwicklungsbegleitenden Review deutlich gesenkt bzw. die Ergebnisqualität der Usability-Tests erheblich gesteigert werden.

In allen Phasen der Systementwicklung, in denen Benutzer mit frühen Versionen des Endproduktes oder mit Prototypen konfrontiert werden, läßt sich zuvor mit Hilfe eines entwicklungsbegleitenden Reviews mit relativ geringem Aufwand ein Eindruck über die ergonomische Qualität der Software gewinnen. Die hierdurch möglich gewordene Behebung offensichtlicher Ergonomie-Defizite noch *vor* einem Usability-Test macht den Blick frei auf die wesentlichen, nur mit dieser Methode erkennbaren Fehler. Die konstruktiven Hinweise im Review erlauben meist eine umgehende Behebung der Bugs. Bestandene Reviews dienen somit einer (auch formalen) „Freigabe zum Usability-Test“. Ohne die Vorschaltung besteht die Gefahr, sich während des aufwendigen Usability-Tests in der Erkennung von Fehlern bei Anordnung und inkonsistenten Bezeichnungen o. ä. zu verlieren.

Der Review kann darüber hinaus Hinweise auf offene Fragen enthalten und damit einen Beitrag für die Planung der Usability-Tests leisten. Die in den Usability-Tests zu bearbeitende Aufgabe sollte beispielsweise bereits als Review-Prüfungsaufgabe formuliert sein. Auf diese Weise erhält auch der Reviewer als Nebenergebnis eine Rückmeldung über die Validität seiner Kritik und der vorgeschlagenen exemplarischen Musterlösungen.

4.3 Usability-Tests in sämtlichen Entwicklungsphasen

Usability-Tests lassen sich sowohl als Feldversuche am Arbeitsplatz als auch im Labor unter kontrollierten Bedingungen durchführen. In Feldversuchen muß bereits ein lauffähiges,

produktiv einsetzbares System vorliegen. Typischerweise ist dies erst kurz vor oder am Ende der Entwicklung der Fall, wenn ausgereifte Beta-Versionen bzw. schon das Endprodukt zur Verfügung stehen. Zu diesem Zeitpunkt ist der Gestaltungskorridor bereits soweit verengt, daß nur noch Geringfügigkeiten im Endprodukt verbessert werden können.

Um die spezifische Kompetenz der Benutzer bereits in möglichst frühe Entwicklungsphasen in den Entwicklungsprozeß einfließen zu lassen, sollte in Usability-Tests nicht nur eine Konfrontation mit dem Endprodukt bzw. Prototypen durchgeführt werden, sondern auch mit Alt- und Alternativsystemen.

Am Anfang eines Entwicklungsprozesses liegt häufig mit einem abzulösenden *Altssystem* eine bisher zu gering genutzte Informationsquelle vor. Ein Feld-Usability-Test gibt hier wertvolle Aufschlüsse über die zu lösenden Aufgaben und vor allem über das Arbeitsumfeld (z. B. Unterbrechungen, Umgebungsbedingungen). Eine so erstellte Analyse der Arbeitsaufgaben und der Arbeitsumfeldes liefert durch ihren konkreten Bezug zu Programmen einen von Entwicklern besser umsetzbaren Beitrag zu Anforderungsdefinitionen und Spezifikationen. Diese Dokumente werden ihrerseits dann in den folgenden Reviews zur Formulierung von Prüfaufgaben herangezogen.

Während die Qualifizierungsfunktion entwicklungsbegleitender Reviews im wesentlichen auf die Software-Entwickler gerichtet ist, kann mit Usability-Tests ein Beitrag zur Qualifizierung aller Beteiligten (Benutzer und Entwickler) geleistet werden. Gerade am Anfang beteiligungsorientierter Projekte ist die Herstellung von Beurteilungs- und Beteiligungskompetenz wichtig. Durch Usability-Tests, hier in der Ausprägung einer *vergleichenden Betrachtung* unterschiedlicher, auf dem Markt für ähnliche oder gleichartige Zwecke angebotenen Programme, können Benutzer wie Entwickler zu wesentlich aussagekräftigeren Festlegungen gelangen.

Eine auch als informelle Version geeignete Variante sind Ad-hoc-Usability-Tests, beispielsweise durch Einladung von Benutzern zu Entwicklerbesprechungen und dem gemeinsamen Durchgehen von Skizzen und Prototypen. Sie erfordern geringen Aufwand und führen in der Regel zu Ergebnissen, wie sie in streng formalisierten Verfahren nicht zu gewinnen sind.

4.4 Usability-Test-Methoden

Usability-Tests – verstanden als Konfrontation von Benutzern mit Programmen oder Vorläufern hierzu – kann 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.

- Rapid Prototyping („low fidelity“-Prototypen)
Entwickler und Benutzer skizzieren auf Papier (nicht auf Rechnern, das begünstigt die Entwickler) Bildschirmformulare für typische Arbeitsaufgaben.
- Prototypenentwicklung
Entwickler führen den Benutzern Bildschirmwürfe vor. Benutzer kommentieren die Entwürfe direkt. Die Ausführungen werden aufgezeichnet oder protokolliert.

- „Papier und Bleistift“-Simulationen
Entwickler legen Benutzern Sätze von ausgedruckten Bildschirmformularen vor. Benutzer arbeiten an einer typischen Sachaufgabe, tragen Ergebnisse in Felder ein und skizzieren durch Linien die typischerweise mit der Maus zurückzulegenden Strecken. Viele „Zacken“ in den Linienzügen lassen auf eine nichtangepaßte Felddaufteilung schließen, häufige Übergänge zu anderen Formularen zeigen Probleme des Maskenzuschnittes auf.
- Lautes Denken
Bei der Erprobung von Prototypen werden alle Aktionen mit dem Programm laut vorgetragen und aufgezeichnet / protokolliert. Die direkte Kommentierung beeinflusst stark den Prüfablauf und sollte auf Einzelaspekte, wie z. B. Wahrnehmbarkeit von Interaktionselementen, beschränkt werden.
- Benchmark-Test (bezogen auf die Benutzung)
Bei Benchmark-Tests wird eine Programmentwicklung dadurch überprüft, daß mehrere Benutzer unter reproduzierbaren Umfeldbedingungen typische Arbeitsaufgaben erledigen. Mit objektivierbaren Methoden wie Zeitmessungen, Interaktionsverfolgung und Fehleranalysen lassen sich Hinweise auf Ergonomiedefizite gewinnen. Sinnvoll ist hierbei die Überprüfung unterschiedlicher Programmvariationen („Welches Modul zeigt die größten Fehlerhäufigkeiten?“).
- Videokonfrontation (im Labor oder der tatsächlichen Arbeitsumgebung)
Benutzer führen am Rechner eine definierte Prüfaufgabe mit dem zu bewertenden Softwareprodukt durch. Sämtliche Interaktionen werden aufgezeichnet, im Anschluß den Benutzern vorgeführt und von ihnen kommentiert. Angaben wie „Hier hatte ich vor, ... zu tun.“, „Von Programm ... bin ich gewohnt, ...“ oder „Das konnte ich nicht erkennen.“ geben Gestaltungshinweise zu allen Ebenen der Benutzungsschnittstelle.

5 Resümee

Der hier vorgestellte Methodenmix, die inzwischen mehrfach von uns praktisch erprobten Reviews und die Usability-Tests, versprechen sich derart zu ergänzen, daß trotz Reduzierung des Gesamtaufwands für Prüfungen und Qualifizierungen eine präzise, schnelle und frühzeitige Fehlererkennung und als Lernerfolg eine nachhaltige Fehlervermeidung zu qualitativen Verbesserungen bei der Softwareherstellung beiträgt.

Der Methodenverbund von Experten-Reviews und Usability-Tests führt zu folgenden Synergien:

- Durch vorgeschaltete Experten-Reviews wird sichergestellt, daß Benutzer sich intensiv mit Fragen zur Aufgabenangemessenheit befassen können und Usability-Tests damit nicht auf „Bugsuche“ reduziert werden.
- Die Ergebnisse der Usability-Tests stehen für einen zweiten Review-Zyklus zur Verfügung und erlauben den Experten eine deutlich bessere Perspektivenübernahme.

Beide Prüfverfahren leisten ihren spezifischen Beitrag zur Qualifizierung der Beteiligten:

- Entwickler erlernen bei der Auseinandersetzung mit dem Prüfungen ihrer konkreten Projekte effizient die Grundzüge der Software-Ergonomie und erkennen den Bedarf für eine tiefergehende Auseinandersetzung mit diesem Gebiet.
- Anwender und Benutzer entwickeln im Zuge von Usability-Tests eine gesteigerte Beurteilungskompetenz.

Der Einsatz des Methodenmixes erfordert von den Experten sowohl software-ergonomische Beurteilungs- und Gestaltungskompetenz (Review) als auch die Fähigkeit zur Durchführung der Usability-Tests, insbesondere zur Moderation des Kommunikationsprozesses zwischen den sich meist unstrukturiert äußernden Benutzern und den Entwicklern, die diesen Äußerungen in der Regel skeptisch gegenüberstehen.

Während einige Prüfmethode von Entwicklern selbständig durchgeführt oder initiiert werden können, so bedarf es für die avancierteren Usability-Tests entsprechender Infrastruktur, die von kleinen und mittelständischen Softwareherstellern allein kaum vorgehalten werden kann. Gemeinschaftliche Einrichtungen, z. B. von Universitäten und Softwarehäusern, stellen hier einen Beitrag zur Problemlösung dar.

6 Literatur

- [1] Asam, R.; Drenkard, N.; Maier, H.-H.: Qualitätsprüfung in Softwareprodukten. Berlin, 1986
- [2] Clegg, C.W.; Warr, P.; Green, T.; Monk, A.; Kamp, G.; Allison, G.; Lansdale, M.: People and Computers – How to Evaluate Your Companys New Technology. Chichester, 1988
- [3] DGQ-NTG Deutsche Gesellschaft für Qualität e.V. (DGQ) und Nachrichtentechnische Gesellschaft im VDE (NTG): Software-Qualitätssicherung. Berlin, 1986
- [4] DIN 66 234: Bildschirmarbeitsplätze, Grundsätze ergonomischer Dialoggestaltung. 1988
- [5] DIN 66 285: Anwendungssoftware – Gütebedingungen und Prüfbestimmungen. August 1990
- [6] DIN EN ISO 9241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten; Teil 10: Grundsätze der Dialoggestaltung. 1995 (noch als Entwurf)
- [7] Dunn, R.H.: Software Qualität. München u.a., 1993
- [8] Dzida, W.; Wiethoff, M.; Arnold, A.G.: ERGOguide. The Quality Assessment Guide to Ergonomic Software. St. Augustin, 1993
- [9] Ehrlich, K.; Rohn, J.A.: Cost Justification of Usability Engineering: A Vendors Perspective. In: Bias, R.G.; Mayhew, D.J.: Cost Justifying Usability. Boston u.a., 1994. 71-110
- [10] Englisch, J.: Systematische Entwicklung von Evaluationsverfahren zur Beurteilung der Benutzungsfreundlichkeit von CAD-Systemen. Forschungsbericht aus dem Institut für Arbeitswissenschaft und Betriebsorganisation der Universität Karlsruhe Band 4 (1992)
- [11] Hamborg, K.-C.; Schweppenhäuser, A.: Zur Bedeutung psychologischer Arbeits- und Aufgabenanalyse für die Softwaregestaltung. In: Rödiger, K.-H. (Hrsg.): Software-Ergonomie '93. Stuttgart, 1993. 227-235
- [12] Hampe-Neteler, W.: Software-ergonomische Bewertung zwischen Arbeitsgestaltung und Softwareentwicklung (Diss.), Bremen, 1994
- [13] Hampe-Neteler, W.; Rödiger, K.-H.: Software-Ergonomie. Verfahren der Evaluierung und Standards zur Entwicklung von Benutzungsoberflächen. Bericht 2/1992, Fachbereich Mathematik/Informatik, Universität Bremen, 1992
- [14] Hüttner, J.; Wandke, H.: What do system designers know about software-ergonomic and how to improve their knowledge. In: H Luczak; A.Cakir; G. Cakir: Work With Display Units. New York u.a., 1992. 304-308

- [15] ISO 9000, Teil 3: Leitfaden für die Anwendung von ISO 9001 auf die Entwicklung, Lieferung und Wartung von Software. Juni 1992
- [16] Jeffries, R.; Desurvire, H.: Usability Testing vs. Heuristic Evaluation: Was there a contest? In: SIGCHI Bulletin Vol 24, Nr.4. October 1992
- [17] Jeffries, R.; Miller, J.R.; Wharton, C.; Uyeda, K.M.: User interface evaluation in the real world: A comparison of four techniques. In: Proceedings of CHI'91. New York, 1991. 119-124
- [18] Lewis, C.; Polson, P.; Wharton, C.; Rieman, J.: Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-an-Use Interface. In: Proceedings of CHI'91. 1991. 235-242
- [19] Nielsen, J.; Molich, R.: Heuristic Evaluations of User Interfaces. In: Proceedings of CHI'90. 1990. 249-256.
- [20] Oppermann, R; Murchner, B.; Reiterer, H.; Koch, M.: Software-ergonomische Evaluation. Der Leitfaden EVADIS II. Berlin u.a., 1992
- [21] Porteus, M; Kirakowski, J.; Corbett, M.: SUMI-Handbook. Human Factor research Group. Cork, 1993
- [22] Prümper, J.; Anft, M.: Die Evaluation von Software auf Grundlage des Entwurfs zur internationalen Ergonomie-Norm ISO 9241 Teil 10 als Beitrag zur partizipativen Systemgestaltung - ein Fallbeispiel. In: K.-H. Rödigier (Hrsg.): Software-Ergonomie '93. Stuttgart, 1993. 145-156
- [23] Ravden, S.; Johnson, G.: Evaluating Usability of Human Computer Interfaces. Chichester, 1989
- [24] Wallace, D.F.; Norman, T.J.: Approaches to Interface-Design. In: Interacting with Computers No. 5., 1993. 259-278 s. a. <http://www.lap.umd.edu/QUISFolder/quisHome.html>

Adressen der Autoren

Dipl.-Inform. Peter Ansoerge
 Universität Bremen
 Fachbereich Mathematik / Informatik
 Technologie-Zentrum Informatik
 Institut für Software-Ergonomie
 und Informationsmanagement
 Bibliothekstraße 1
 28359 Bremen

E-Mail: ansorge@informatik.uni-bremen.de
<http://selab24.informatik.uni-bremen.de/ansorge>

Dipl. oec. Dipl.-Inform. Uwe Haupt
 Universität Bremen
 Fachbereich Mathematik / Informatik
 Technologie-Zentrum Informatik
 Institut für Software-Ergonomie
 und Informationsmanagement
 Bibliothekstraße 1
 28359 Bremen

E-Mail: haupt@informatik.uni-bremen.de
<http://selab24.informatik.uni-bremen.de/haupt>

