

Benutzerorientierte, iterative Software-Entwicklung in der Praxis^a

Jochen Prümper
DATA TRAIN GmbH

Zusammenfassung

Es wird von einer Softwareentwicklung für klein- und mittelständische Verlage berichtet, die unter arbeitswissenschaftlicher Betreuung in einem iterativen Vorgehen in enger Zusammenarbeit mit den potentiellen Endbenutzern durchgeführt wurde. Zur Sprache kommen: (1) die Anforderungen an die Anwendungssoftware und an das Software-Entwicklungs-Tool; (2) die arbeitswissenschaftlichen Schulungen mit den Softwareentwicklern; (3) die Vorgehensweise bei den Arbeitsanalysen; (4) die verschiedenen an der Software-Entwicklung beteiligten Gremien.

Abstract

This article describes a software development project for small and middle publishing companies that has been carried out in an iterative process under industrial scientific supervision in close cooperation with potential end users. The following will be discussed: (1) the demands for the software application and for the software development tool; (2) the industrial scientific training procedure for the software engineers; (3) the work analysis procedure; (4) the different committees participating in the software development.

Résumé

Cet article traite du développement d'un programme pour des maisons d'éditions de petites et moyennes tailles. Ce projet a été effectué sous un éclairage scientifique (lié au monde du travail), dans une démarche itérative, et en étroite collaboration avec les utilisateurs potentiels. Les sujets traités sont: (1) les exigences liées à l'application du programme et à l'outil de son développement; (2) la formation de base des analystes programmeurs dans les sciences du travail; (3) le procédé d'analyse du travail, (4) les différents groupes intervenant au développement du programme.

1 Einleitung

Eine Methode der Software-Entwicklung, die in letzter Zeit immer mehr in den Blickpunkt des Interesses rückt, ist das benutzerzentrierte Prototyping (vgl. z.B. Connell & Shafer, 1989; Gunzenhäuser, 1988; Hallmann, 1990; Heeg & Neuser, 1988; Kieback, Lichter, Schneider-Hufschmidt & Züllighoven, 1991). Dabei handelt es sich um eine Produktentwicklungsstrategie, bei der eine frühzeitige, interaktive

^a Dieser Beitrag entstand im Rahmen des Projektes "Verlag 2000: Eine benutzerfreundliche integrierte Lösung für die mittelständische Verlags- und Druckereibranche unter Berücksichtigung von zu verbessernden Arbeitsbedingungen für die Beschäftigten." Dieses Projekt wird gefördert vom Bundesministerium für Forschung und Technologie, Projektträger "Arbeit und Technik" (Förderkennzeichen: 01 HK 601). Dank gilt an dieser Stelle Michael Anß, Michael Fresno, Klaus Hartmannsgruber und Dankmar Scheuchl.

Auseinandersetzung mit den potentiellen Anwendern und ihren Arbeitsaufgaben betrieben wird. Hierzu werden in einem iterativen Designprozeß anhand typischer Arbeitsaufgaben systematische empirische Bewertungen des Produktes vorgenommen, die dann wieder in einem zyklischen Entwicklungsprozeß in den Überarbeitungen des Prototypen Verwertung finden. Trotz der Tatsache, daß eine Anzahl von Autoren den finanziellen und zeitlichen Gewinn einer derartigen Vorgehensweise propagieren (z.B. Baroudi, Olson & Ives, 1986; Bewley, Roberts, Schroit & Verplanck, 1983; Boehm, Gray & Seewald, 1984; Karat, 1990; Mantey & Teory, 1988; Strohm, 1991), so muß man im großen und ganzen feststellen, daß diese Art der Software-Entwicklung noch wenig Eingang in die Praxis gefunden hat (vgl. Aschersleben & Zang-Scheucher, 1989; Rosson, Maass & Kellogg, 1987). Der Grund hierfür ist möglicherweise darin zu sehen, daß Softwareentwickler von einer negativen Beeinflussung durch die an der Entwicklung beteiligten Benutzer sowohl auf die Qualität der Software, als auch auf den Softwareentwicklungsprozeß an sich berichten (vgl. Heinbokel, Frese, Stolte, Brodbeck & Sonntag, 1993). Da sich jedoch an dieser Stelle die Vermutung aufdrängt, daß derartig negative Erlebnisse und Vorbehalte zu großen Teilen auch auf die geringen Erfahrungen der Softwareentwickler mit dieser Art von Produktentwicklungsstrategie zurückzuführen sind (vgl. Zang & Gstalter, 1987), soll im folgenden von einem Softwareentwicklungsprojekt berichtet werden, in dem ein benutzerzentriertes, iteratives Verfahren durchgeführt wurde. Die hier beschriebene Vorgehensweise erhebt dabei keineswegs den Anspruch, ein musterhafter Leitfaden für eine partizipative, iterative Software-Entwicklung *per se* zu sein, sondern möchte lediglich anhand eines Softwareentwicklungsprojektes aus der Praxis - im Sinne eines Erfahrungsaustausches - einige Anregungen vermitteln.

Berichtet wird im folgenden von: (1) den Anforderungen an die Anwendungssoftware und an das Software-Entwicklungs-Tool; (2) der Vorgehensweise bei den arbeitswissenschaftlichen Schulungen mit den Softwareentwicklern; (3) der Vorgehensweise bei den Arbeitsanalysen; (4) der Vorgehensweise bei den verschiedenen an der Software-Entwicklung beteiligten Gremien.

2 Anforderungen an die Anwendungssoftware und das Software-Entwicklungs-Tool

Der folgende Abschnitt liefert: (1) einen Einblick in die Anforderungen an die zu entwickelnde Verlagssoftware und (2) eine Beschreibung des Software-Entwicklungs-Tools.

2.1 Die Anwendungssoftware

Das Ziel der Entwicklung war eine Standardsoftware für klein- und mittelständische Verlage. Die zu entwickelnde Verlagssoftware sollte vier Module umfassen: ein Adreß-, Buch-, Anzeigen- und Abonnenten-Modul. Die EDV-Aufgabenstellung reicht im Verlagswesen auf der Anwenderseite von der Unterstützung für Marketing-Maßnahmen über Auftragsabwicklung mit Fakturierung und Versand, Bestell- bzw. Herstellunterstützung bis zu modernem Rechnungswesen mit Buchhaltung und Kostenrechnung und letztendlich einem integrierten Informationssystem für alle Arbeitsbereiche und Führungsebenen. Eine besondere Herausforderung bestand in der Integration der im Verlagswesen sehr breiten Vielfalt unterschiedlichster Anforderungen. Diese reicht von Anzeigenblättern und Büchern, über Loseblatt-Werke bis hin zu Zeitschriften und Zeitungen. Zudem gibt es eine Reihe von Spezialbereichen, wie z.B. Kartographische Werke oder Musikverlag-Produkte.

2.2 Das Software-Entwicklungs-Tool

Eine entscheidende Frage war die Wahl eines geeigneten Software-Entwicklungs-Tools. Eine wichtige Bedeutung wurde dabei der Frage beigemessen, inwieweit sich ein derartiges Werkzeug zum Prototyping eignet. Da die Softwareentwicklung auf einer IBM AS/400 stattfinden sollte, fiel die Entscheidung zugunsten LANSAs aus. Bei LANSAs (vgl. Kuscher, 1992) handelt es sich um ein CASE-Tool (CASE: Computer Aided Software Engineering), welches darauf ausgelegt ist, den Software-Entwicklungsprozeß von der Planung bis hin zur Wartung zu automatisieren (vgl. Focus, 1991; Vonk, 1990). Aufgrund der Überzeugung, daß in Zukunft Computern der mittleren Datentechnik zunehmend eine reine Server-Funktion zukommen wird und die eigentliche Mensch-Computer Interaktion so gut wie ausschließlich an Personal-Computern/Work-Stations stattfinden wird, machte der besondere Reiz bei LANSAs die Möglichkeit aus, daß für dieses CASE-Tools ein GUI (Graphical User Interface) angekündigt war, welches die Möglichkeit bieten sollte, die Windows-PC-Oberfläche für AS/400-Programme nutzbar zu machen. Von diesem GUI versprachen wir uns ein wirkliches *Rapid*-Prototyping, da dadurch die Gestaltung der Benutzeroberfläche unabhängig von der Funktionalität vorgenommen werden konnte.

Die Entscheidung, ein CASE-Tool als Entwicklungswerkzeug zu verwenden, hat sich im großen und ganzen bewährt. Allerdings stellte es eine Ernüchterung dar, daß die aktuelle GUI-Version von LANSAs noch zu unausgegoren war, als daß es als *Rapid*-Prototyping-Tool hätte eingesetzt werden können. Dadurch sahen wir uns gezwungen, statt eines "Rapid-Prototypings" mit PC-Support ein "Versioning" an der

AS/400 zu betreiben, das den Entwicklungsprozeß in einzelne kleinere Teilentwicklungen aufteilte (vgl. Peschke & Wittstock, 1987). Neben dem Umstand, daß dadurch die Maskengestaltung nicht in ihrer endgültigen Form evaluiert werden konnte, bedeutete dies insbesondere für die Softwareentwickler einen erheblichen Mehraufwand.

Zudem werden - und wurden auch in diesem Projekt - die Auswirkungen, die ein derartig mächtiges Tool auf den Software-Entwicklungsprozeß haben können, unterschätzt (vgl. Karer, 1992). CASE-Tools stellen aufgrund ihrer Komplexität höhere Anforderungen an das Projektmanagement als herkömmliche Verfahren (vgl. Mährländer, 1991). Die Einführung eines CASE-Tools, so zeigt unsere Erfahrung, sollte nicht gleichzeitig mit dem eigentlichen Softwareentwicklungsprojekt beginnen. Die Einführung eines CASE-Tools stellt ein eigenes Projekt dar und erfordert im Rahmen eines langfristig geplanten Gesamtkonzeptes eine intensive Betreuung der betroffenen Softwareentwickler.

3 Arbeitswissenschaftliche Schulungen

Eine Software-Entwicklung unter Benutzerbeteiligung stellt an die Softwareentwickler besondere Anforderungen an ihre software-ergonomischen Kenntnisse und sozialen Fertigkeiten. Ähnlich wie bei Hacker, Holz auf der Heide & Aschersleben (1991) wurden deshalb zu Beginn des Projektes mit den Softwareentwicklern eintägige arbeitswissenschaftliche Schulungen durchgeführt. Folgende Themenbereiche kamen dabei zur Sprache: (1) Einführung in eine handlungsorientierte Fehlertaxonomie in der Mensch-Computer Interaktion; (2) Software-Ergonomie und Probleme bei der Einführung neuer Technologien; (3) Partizipative, iterative Software-Entwicklung; und (4) Kommunikation mit dem Endbenutzer.

In der ersten Schulung "*Handlungsorientierte Fehlertaxonomie in der Mensch-Computer Interaktion*" erhielten die Softwareentwickler einen Einblick in ein handlungstheoretisches Kategoriensystem zur Einordnung von Fehlern in der Mensch-Computer Interaktion (vgl. Frese & Zapf, 1991; Zapf, Brodbeck & Prümper, 1989). Anschließend stellten die Softwareentwickler selbst beobachtete oder erlebte Fehler vor, und klassifizierten diese entsprechend der Taxonomie. Anhand dieser selbstgenerierten Fehler wurden mit den Softwareentwicklern mögliche Verbesserungen für die jeweiligen Softwareprogramme erarbeitet. Der inhaltliche Schwerpunkt lag dabei sowohl auf der Darstellung der Möglichkeiten zur Fehlervermeidung (vgl. Zapf, Frese, Irmer & Brodbeck, 1991) als auch zum Fehlermanagement (vgl. Frese, Irmer & Prümper, 1991).

In der zweiten Schulung "*Software-Ergonomie und Probleme bei der Einführung*

neuer Technologien" wurden zum einen spezielle software-ergonomische Fragen behandelt und zum anderen Probleme bei der Einführung neuer Technologien aufgezeigt. Den Schwerpunkt des software-ergonomischen Teils bildete der von der *International Organisation for Standardisation* vorgelegte Normentwurf zur ergonomischen Dialoggestaltung (ISO 9241/10, 1992), sowie allgemeine Grundsätze der Informationsdarstellung, Methoden und Werkzeuge zur Masken- und Menügestaltung, Empfehlungen für Zeichen, Markierungen, etc. (vgl. Brown, 1989; Hoffmann, Klose & Martin, 1989; Shneiderman, 1992; Smith & Mosier, 1986). Im Vordergrund des Schwerpunktes "Probleme bei der Einführung neuer Technologien" standen die Punkte "Partizipation und Einführung neuer Techniken" und "Organisationsumfeld und neue Technologien" (vgl. Frese & Brodbeck, 1989).

In der dritten Schulung "*Partizipative, iterative Software-Entwicklung*" wurden den Softwareentwicklern das Vorgehen einer benutzerzentrierten, iterativen Softwareentwicklung vorgestellt. Zur Sprache kamen die Prinzipien für ein "design for usability" nach Gould & Lewis (1985), die Darstellung von Prototyping-Prozessen nach Floyd (1984) und Williges, Williges und Elkerton (1987) und Methoden zur Bewertung von Software (Aschersleben, Gstalter, Kaiser, Strube & Zang-Scheucher, 1989).

In der vierten Schulung "*Kommunikation mit dem Endbenutzer*" wurde mit den Softwareentwicklern Regeln zur Kommunikation mit dem Endbenutzer erarbeitet. Inhalte waren hier Regeln der Moderation (Klebert, Schrader & Straub, 1985) und Präsentation (Hierhold, 1990), Interviews entsprechend der "critical incident technique" (Flanagan, 1954), Strukturierung von Diskussionen (Bieger, Feltes, Schmalzriedt & Sieber, 1981) und die Methode der visualisierten Diskussion (Mauch, 1981).

Die arbeitswissenschaftlichen Schulungen stießen bei den Softwareentwicklern auf eine sehr positive Resonanz und trugen entscheidend dazu bei, die Bedeutung arbeitswissenschaftlicher und software-ergonomischer Ergebnisse in das Projekt zu integrieren.

Allerdings wurde bald deutlich, daß vereinzelte Schulungen alleine nicht genügen, damit aus traditionellen Softwareentwicklern benutzerorientierte Softwaredesigner werden. Darüber hinaus empfiehlt es sich nicht, einen geschlossenen Schulungsblock am Anfang eines Projektes zu plazieren.

Den Softwareentwicklern fiel es häufig schwer, während ihrer alltäglichen Arbeit software-ergonomische Überlegungen zu berücksichtigen und bei kleineren Referaten im Kreis von Benutzern eine angemessene Sprache zu finden. Die Vernachlässigung software-ergonomischer Aspekte ist dabei zu Beginn des Entwicklungsprozesses verständlich, da in dieser Zeit ein Großteil der konzeptionellen Arbeit geleistet wird und Fragen bezüglich Masken- oder Menügestaltung noch nicht so relevant sind. Im Rahmen dieser *Entwurfskonzeptuali-*

sierung stellten sich vielmehr die Fragen: (1) "Was sind die Arbeitsaufgaben der Benutzer?", (2) "Wie setze ich diese Anforderungen software-technisch um?" und (3) "Wie melde ich den Benutzern meine Überlegungen zurück?". Erst im späteren Projektverlauf der *Oberflächenkonzeptualisierung* kamen dann noch die Fragen auf: (4) "Wie übertrage ich die konzeptuellen Überlegungen auf die Oberfläche?" und (5) "Wie bewerte ich mit den Benutzern die Oberfläche?". Da in manchen Fällen die Software-Entwickler zudem mit der Aufgabe der Dokumentation und Schulung der Software betraut gemacht werden sollten, stellten sich für die *Ausbildungskonzeptualisierung* die Fragen (6) "Wie dokumentiere ich Software?" und (7) "Wie gestalte ich eine Software-Schulung?".

Entsprechend dieser Überlegungen bietet es sich an, arbeitswissenschaftlichen Schulungen im Softwareentwicklungsprozeß folgendermaßen zu gestalten (vgl. Abb. 1):

- | |
|---|
| <p>A. Schulungen zur Entwurfkonzeptualisierung</p> <ol style="list-style-type: none"> 1. arbeitsanalytische Verfahren 2. Methoden zur Datenstrukturierung 3. Kommunikations- und Präsentationstechniken <p>B. Schulungen zur Oberflächenkonzeptualisierung</p> <ol style="list-style-type: none"> 4. Software-Ergonomie 5. Kriterien und Methoden zur Bewertung von Software <p>C. Schulungen zur Ausbildungskonzeptualisierung</p> <ol style="list-style-type: none"> 6. Dokumentation von Software 7. Schulung von Software |
|---|

Abb. 1: Möglicher Aufbau einer arbeitswissenschaftlichen Schulung im Softwareentwicklungsprozeß

Von entscheidender Bedeutung ist, daß diese Schulungen nicht als "theoretische Trockenübungen" durchgeführt werden, sondern daß sie eng mit dem natürlichen Projektverlauf verknüpft werden. Dies verlangt nicht nur hohe Koordinationsanforderungen an das Projektmanagement, sondern bedeutet insbesondere, daß es unerlässlich ist, den jeweiligen Schulungen eine angemessene praktische Umsetzung folgen zu lassen. Die Software-Entwickler müssen in deutlichem Maße dazu angehalten werden, Arbeitsanalysen bei den beteiligten Benutzern in den Unternehmen durchzuführen, verschiedene Konzepte zur Datenstrukturierung zu entwerfen und mit Kollegen zu diskutieren, Referate entsprechend der gelernten Techniken vorzubereiten, alternative Oberflächenentwürfe zu konzipieren und lernen zu begründen, Methoden zur Softwareevaluation bei verschiedenen Systemen anzuwenden und daraus Gestaltungshinweise abzuleiten, Hilfe- und Informationstexte zu formulieren und auch diese im Rahmen des

Prototypings den Benutzern vorzulegen, Schulungseinheiten zu konzeptualisieren und anzuwenden, etc.

Die arbeitswissenschaftliche Tätigkeit darf sich dabei nicht nur auf die Durchführung der Schulungen beschränken. Sie muß insbesondere die Aufgabe erfüllen, ein ständiges Bindeglied zwischen den an der Softwareentwicklung beteiligten Entscheidungsträgern, Benutzern und Entwicklern zu sein und die praktische Umsetzung zu kontrollieren und aktiv mitzugestalten.

4 Arbeitswissenschaftliche Analysen

Da Softwaregestaltung immer auch Arbeitsgestaltung ist, gilt es nicht nur die Software nach neuesten ergonomischen Gesichtspunkten zu entwickeln, sondern auch bereits während der Planungsphase die Auswirkungen der neuen Technologie auf den gesamten Arbeitsablauf zu berücksichtigen. Dies bedeutet, daß in einem soziotechnischen Ansatz sowohl (1) eine Analyse der Software als auch (2) eine Analyse der Arbeitssituation Beachtung finden muß. Aus methodischer Sicht wollten wir uns dabei in beiden Punkten um eine Ergänzung von qualitativen und quantitativen Analysen bemühen.

4.1 Analyse der Software

Die *qualitativen Analysen der Software* fanden in den an der Entwicklung beteiligten Verlagen statt und bestanden aus einer Reihe von Beobachtungen und Befragungen an einzelnen Computerarbeitsplätzen. Das Anliegen der qualitativen Analyse der Software war eine möglichst getreue Abbildung einzelner Arbeitsabläufe, der dort auftretenden Schwierigkeiten und die Aufnahme von Verbesserungs- und Ergänzungsvorschlägen. Grundlage dieser Analysen war eine bestehende Verlagssoftware, die von der neu zu entwickelnden Verlagssoftware abgelöst werden sollte.

Nachdem wir per Interview eine *Grobanalyse* des Arbeitsablaufs erstellt hatten, baten wir die Benutzer, uns zwecks *Feinanalyse* die einzelnen Arbeitsschritte bei der Bearbeitung einer Standardaufgabe am Bildschirm möglichst so detailliert darzustellen, daß daraus ein Ablaufdiagramm erstellt werden konnte, in dem die einzelnen Arbeitsschritte nach Ober- und Unterzielen gegliedert wurden. Ergänzt wurde diese Erfassung der Bearbeitung von Standardaufgaben um die Frage nach Problemen und Schwierigkeiten, die bei *Ausnahmen von derartigen Standardaufgaben* auftreten und um die *Frage nach vorbereitenden und nachbereitenden Tätigkeiten* des Benutzers, von Kollegen und Vorgesetzten.

Abbildung 2 zeigt Auszüge einer derartigen Feinanalyse.

LN	Arbeitsschritt	Auffälligkeiten
4	Der Benutzer erhält telephonisch einen Auftrags- eingang.	Zur Auftragsbearbeitung benötigt der Benutzer zunächst eine Kundennummer.
4.1.1	Der Benutzer aktiviert den Menüpunkt "Adress- verwaltung" um dem Kunden eine Kunden- nummer zuzuordnen.	Die Maske, die daraufhin erscheint, trägt den Titel "Adressbearbeitung".
4.1.2	Der Benutzer gibt in der Maske "Adressbearbei- tung" die Adresse des Kunden ein und erhält daraufhin für den Kunden eine Kundennummer.	Da die Kundennummer für die eigentliche Auf- tragserfassung benötigt wird, notiert der Benutzer die Kundennummer auf einem Zettel. Es besteht keine Möglichkeit, diese Nummer in die folgende Auftragsbearbeitung mitzunehmen.
4.1.3	Der Benutzer verläßt die "Adressbearbeitung" und befindet sich daraufhin wieder im Hauptmenü.	
4.2.1	Der Benutzer aktiviert den Menüpunkt "Auftragsbearbeitung" und erhält eine Auftrags- nummer.	Die Maske, die daraufhin erscheint, trägt den Titel "Auftragsverwaltung". Da die Auftragsnummer zu einem späteren Zeit- punkt noch einmal benötigt wird, überträgt der Benutzer auch diese Zahl auf Papier. Dies geht schneller, als diese Nummer später über das Auf- tragssuchprogramm ausfindig zu machen.

Abb. 2: Auszüge einer Feinanalyse: Arbeitsschritte und Auffälligkeiten

Bereits in diesen ersten Analysen wurde deutlich: die bestehende Software unterstützte die Arbeitsaufgabe der Benutzer nur unvollständig, die Menüpunkte trugen andere Bezeichnungen als die entsprechenden Masken, die Software verwendete Begriffe, die nicht aus der Arbeitswelt des Benutzers stammten, sie verlangte überflüssige Eingaben, verwendete eine unverständliche Sprache, war für den Benutzer nach nicht nachvollziehbaren Ordnungskriterien organisiert, störte den Arbeitsablauf durch unnötige Systemmeldungen, unterstützte den Benutzer nicht bei der Suche nach Informationen, verlangte umständliche Eingaberoutinen, enthielt nicht alle benötigten, dafür aber viele redundante oder für den Benutzer nicht interpretierbare Informationen und war an manchen Punkten so kompliziert gestaltet, daß der Benutzer die Aufgaben effizienter "per Hand" löste. Zudem stürzte die Software bei bestimmten Tastenkombinationen ab, oder bot Optionen an, die nicht mit Funktionalität hinterlegt waren.

Der Vorteil insbesondere der Feinanalysen war darin zu sehen, daß wir über sie einen weitgehend nahtlosen Einblick in die computerunterstützten Tätigkeiten einzelner Mitarbeiter erhielten. Zudem wurden von den Benutzern bereits schon an dieser Stelle eine Reihe von Verbesserungsvorschlägen generiert.

Einschränkend muß erwähnt werden, daß die sequentielle Erarbeitung der einzelnen Module (Adresse, Buch, Anzeige und Abonnement) mit einzelnen Benutzern dann nicht die optimale Vorgehensweise sein kann, wenn die innerbetrieblichen Abläufe eine enge Verzahnung unterschiedlicher Anforderungen

verlangen. Die gesamten betrieblichen Abläufe sowie die Arbeitstätigkeit einer Vielzahl von Mitarbeitern haben einen Grad an Komplexität, Vernetztheit und Dynamik, der durch die beschriebene arbeitsanalytische Vorgehensweise noch ungenügend transparent wurde. Um einen umfangreichen Beitrag zu einer ganzheitlich orientierten Optimierung der sozialen und technischen Subsysteme zu leisten, sind weitere Analysen notwendig, die es gestatten, die synchrone Repräsentation eines integrierten Netzwerkes von Arbeits- und Organisationseinheiten abzubilden.

Zur *quantitativen Analyse der Software* wurde ein Verfahren auf Grundlage der sieben Grundsätze Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Erwartungskonformität, Fehlerrobustheit, Individualisierbarkeit und Erlernbarkeit des Entwurfs zur internationalen Ergonomie-Norm ISO 9241/10 (1992) entwickelt. Neben der standardisierten Bewertung von Software soll es als Grundlage für die Moderation mit Benutzergruppen dienen, wenn mit ihnen Designanforderungen erarbeitet werden.

Im Vordergrund der Software-Entwicklung stand die Form der aktiven Partizipation (Heilmann, 1981), bei der den Benutzern die Möglichkeit gegeben wird, den Systemgestaltungsprozeß aktiv mitzubeeinflussen. Da allerdings in den meisten Fällen lediglich ein ausgewählter Kreis von Endbenutzern die Möglichkeit hat, an den Benutzergruppentreffen teilzunehmen, sollten die restlichen von der Softwarelösung betroffenen Personen zumindest den Status der passiven Partizipation erhalten. Deshalb beurteilten auch andere in den Verlagen tätige Personen ihre Software anhand dieses Verfahrens. Mittels der individuellen Fragebogenergebnisse wurden dann mit einer Anzahl von Benutzern vor Ort Interviews durchgeführt, in denen sie Verletzungen der Grundsätze erläuterten. In den Benutzergruppensitzungen wurden die kumulierten Ergebnisse dazu herangezogen, um Mittels der Technik der Kartenfrage (Klebert, Schrader & Straub, 1985) Benutzer und Entwickler im Plenum miteinander ins Gespräch zu bringen und dabei grundlegende Designprobleme zu erörtern. Das Ergebnis war letztendlich eine Fülle von konkreten Hinweisen auf Schwachstellen, die das Ergebnis der Fragebogenerhebung sinnvoll bereicherten und erweiterten (zur Vertiefung siehe: Prümper & Anft, 1993).

Zur Illustration seien exemplarisch für die beiden Grundsätze Steuerbarkeit und Erwartungskonformität jeweils eine Operationalisierung und ein Ergebnis der Kartenabfrage dargestellt (vgl. Abb. 3).

Grundsatz	Item	Verletzung des Grundsatzes
Steuerbarkeit	Die Software erzwingt eine unnötig starre Einhaltung von Bearbeitungsschritten.	Bei der Adreß-Neuanlage wird zwingend in die Erfassungsmaske für Konditionsvereinbarungen verzweigt, obwohl meist keine Konditionen benötigt werden.
Erwartungskonformität	Die Software erschwert die Orientierung durch eine uneinheitliche Gestaltung.	In der Adreß-Maske sind die Funktionstasten anders belegt als zum Beispiel in den Bereichen Auftragserfassung, Buchhaltung oder Abonnement.

Abb. 3: Die Grundsätze Steuerbarkeit und Erwartungskonformität der ISO 9241 Teil 10, Beispielimens für ihre Operationalisierung und exemplarische Ergebnisse

4.2 Analyse der Arbeitssituation

Die ersten Analysen der Software vermittelten bereits: in den an der Softwareentwicklung beteiligten Verlagen bestand eindeutig ein großer Bedarf nach einer neuen, software-ergonomisch gut gestalteten Software. Naturgemäß war bei den Arbeitswissenschaftlern und den Softwareentwicklern kein umfassendes Wissen des Verlagswesens vorhanden. Zur *qualitativen Analyse der Arbeitssituation* wurde von arbeitswissenschaftlicher Seite versucht, Einsicht in die Arbeitsabläufe eines Verlages über die - insbesondere qualitative - Analyse der Software und über Interviews mit Benutzern zu erlangen. Die Softwareentwickler näherten sich dem Verlagswesen über die Beschäftigung mit der bestehenden Software.

Beide Vorgehensweisen genügen nicht, um einen für die Softwareentwicklung ausreichenden Einblick in die Anforderungen eines Unternehmens zu gewinnen.

Aus diesem Grund wurde ein Verlagsberater berufen, der seine langjährige Erfahrung mit verlagsspezifischer Software in das Projekt einbrachte. Zudem könnte es hilfreich sein, wenn - wie einige Male seitens der Benutzer angeregt - Arbeitswissenschaftler und Softwareentwickler für einige Zeit als reguläre Angestellte an dem Arbeitsprozeß teilnahmen.

Zur *quantitativen Analyse der Arbeitssituation* wurden bestehende arbeitswissenschaftliche Instrumente danach untersucht, inwiefern sie sich sowohl für unsere um Praxisnähe bemühte Herangehensweise als auch für die zu untersuchenden Büroarbeitsplätze eigneten.

In die engere Wahl kamen folgende Instrumente: der "Job Diagnostic Survey" (JDS) von Hackman und Oldham (1975), der "BMS" (Ermüdung-Monotonie-Sättigung-Streß) von Plath und Richter (1984), der "Erhebungsbogen zur Erfassung des Betriebsklimas" von von Rosenstiel, Falkenberg, Hehn, Henschel und Warns (1983), das "Verfahren zur subjektiven Arbeitsanalyse" (SAA) von Udris und Alioth

(1980) und das "Instrument zur streßbezogenen Arbeitsanalyse" (ISTA) von Semmer (1984) bzw. ISTA-C für Computerarbeitsplätze von Zapf (1991).

Da der größte Teil dieser Instrumente eine uneinheitliche Frage- und Antwortstruktur aufweist, sich manche Items nicht als trennscharf erwiesen haben, uns der zeitliche Aufwand all dieser Instrumente zu groß erschien oder sich manche Items nicht für die Untersuchung von Büroarbeitsplätzen anbieten, entschlossen wir uns, aus diesen arbeitsanalytischen Instrumenten für unsere Zwecke ein zeitökonomisches Verfahren mit einheitlicher Frage- und Antwortstruktur und ausgewählten Items zu generieren.

Der aus diesen Instrumenten erwachsene Fragebogen zur "Einschätzung der Arbeitstätigkeit" (ESAT) enthält 31 Items zu den Konstrukten "Handlungsspielraum", "Arbeitskomplexität", "Anforderungsvielfalt", "Monotonie", "Aspekte sozialer Interaktion", "Soziale Kohäsion", "Rollenkonflikt", "Ganzheitlichkeit", "Belastungen", "Betriebliche Leistungen" und "Information und Mitsprache" (zur Vertiefung siehe: Prümper & Hartmannsgruber, 1993).

Darüber hinaus wurden Items zur Klärung der Fragestellung konstruiert, ob eine intensive Computerarbeit durch den Wegfall "sinnlicher Erfahrung" (vgl. Böhle & Milkau, 1988; Rose, 1984) eine Belastung darstellt und auch dann für die Gestaltung von Mischarbeitsplätzen plädiert werden muß, wenn es um Maßnahmen zum 'job-enrichment' geht.

Diese quantitative Analyse der Arbeitssituation sollte dabei nicht nur eine standardisierte Ergänzung zu den qualitativen Arbeitsanalysen darstellen, sondern insbesondere den Einfluß einer veränderten Arbeitsplatzgestaltung im Längsschnitt Rechnung tragen.

Bislang liegen zu dem Einsatz dieses Verfahrens noch keine Ergebnisse vor. Neben der reinen quantitativen Erfassung der Arbeitssituation ist jedoch seine Anwendung ebenfalls so geplant, wie oben bei der Darstellung des Fragebogens zur Softwareevaluation beschrieben wurde: als "screening"-Instrument zur Moderation mit Benutzergruppen.

5 Arbeit in den verschiedenen Gremien

Im Rahmen des Projektes wurden drei Gremien ins Leben gerufen: (1) eine Benutzergruppe, (2) eine Kundengruppe und (3) ein Projektbeirat.

5.1 Die Benutzergruppe

Die Benutzergruppe definierte und überprüfte die Feinanforderungen an die zu entwickelnde Verlagssoftware. In die Benutzergruppe wurden aus jedem Verlag je

(1980) und das "Instrument zur streßbezogenen Arbeitsanalyse" (ISTA) von Semmer (1984) bzw. ISTA-C für Computerarbeitsplätze von Zapf (1991).

Da der größte Teil dieser Instrumente eine uneinheitliche Frage- und Antwortstruktur aufweist, sich manche Items nicht als trennscharf erwiesen haben, uns der zeitliche Aufwand all dieser Instrumente zu groß erschien oder sich manche Items nicht für die Untersuchung von Büroarbeitsplätzen anbieten, entschlossen wir uns, aus diesen arbeitsanalytischen Instrumenten für unsere Zwecke ein zeitökonomisches Verfahren mit einheitlicher Frage- und Antwortstruktur und ausgewählten Items zu generieren.

Der aus diesen Instrumenten erwachsene Fragebogen zur "Einschätzung der Arbeitstätigkeit" (ESAT) enthält 31 Items zu den Konstrukten "Handlungsspielraum", "Arbeitskomplexität", "Anforderungsvielfalt", "Monotonie", "Aspekte sozialer Interaktion", "Soziale Kohäsion", "Rollenkonflikt", "Ganzheitlichkeit", "Belastungen", "Betriebliche Leistungen" und "Information und Mitsprache" (zur Vertiefung siehe: Prümper & Hartmannsgruber, 1993).

Darüber hinaus wurden Items zur Klärung der Fragestellung konstruiert, ob eine intensive Computerarbeit durch den Wegfall "sinnlicher Erfahrung" (vgl. Böhle & Milkau, 1988; Rose, 1984) eine Belastung darstellt und auch dann für die Gestaltung von Mischarbeitsplätzen plädiert werden muß, wenn es um Maßnahmen zum 'job-enrichment' geht.

Diese quantitative Analyse der Arbeitssituation sollte dabei nicht nur eine standardisierte Ergänzung zu den qualitativen Arbeitsanalysen darstellen, sondern insbesondere den Einfluß einer veränderten Arbeitsplatzgestaltung im Längsschnitt Rechnung tragen.

Bislang liegen zu dem Einsatz dieses Verfahrens noch keine Ergebnisse vor. Neben der reinen quantitativen Erfassung der Arbeitssituation ist jedoch seine Anwendung ebenfalls so geplant, wie oben bei der Darstellung des Fragebogens zur Softwareevaluation beschrieben wurde: als "screening"-Instrument zur Moderation mit Benutzergruppen.

5 Arbeit in den verschiedenen Gremien

Im Rahmen des Projektes wurden drei Gremien ins Leben gerufen: (1) eine Benutzergruppe, (2) eine Kundengruppe und (3) ein Projektbeirat.

5.1 Die Benutzergruppe

Die Benutzergruppe definierte und überprüfte die Feinanforderungen an die zu entwickelnde Verlagssoftware. In die Benutzergruppe wurden aus jedem Verlag je

nach zu entwickelndem Modul 1-3 Anwender entsandt. Die Benutzergruppe tagte drei bis fünf Mal pro Modul.

Unter der Anleitung von Arbeitswissenschaftlern kamen Anwender und Entwickler in regelmäßigen Abständen zusammen, um in zweitägigen Sitzungen die einzelnen Designphasen miteinander abzustimmen. Jede Benutzergruppensitzung bestand aus drei Bausteinen: (1) Überprüfung, ob die Festlegungen der vergangenen Benutzergruppensitzung erfüllt wurden; (2) Klärung offener Fragen; (3) Bearbeitung und Überprüfung des Prototypen anhand konkreter Arbeitsaufgaben (vgl. Abb. 4).

Festlegungen	Offene Fragen	Konkrete Arbeitsaufgaben
<ul style="list-style-type: none"> - Haben die Ansprechpartner die selbe Kundennummer wie die Firma? - Können zusätzliche Telefon- und Faxnummern hinterlegt werden? - Funktioniert die Änderung von Privat- zu Firmenadresse? - Gibt es in der Adreßerfassung die Funktionstaste, die direkt zum Ansprechpartner verzweigt? 	<ul style="list-style-type: none"> - Wann werden Adressen inaktiv gesetzt? - Was passiert mit persönlichen Rechten, wenn der Mitarbeiter ausscheidet? - Reicht die Objektnummer für die Zuordnung der Bankverbindung zum Objekt? - Besteht die Möglichkeit, Vornamensabellen einzulesen? - Welche Auswirkungen hat die neue Postleitzahlenregelung? 	<p>Legen Sie folgende Adresse an und überprüfen Sie die Auswirkung auf die Browseliste und die Etikettenfunktion:</p> <p>Ehrwürdige Schwester Else Mack Leiterin der Berufsfachschule für Sozialwesen Kloster St. Kathrin Klostersgasse 8 8000 München 80 Tel. 089 - 448 37 82 Fax: 089 - 448 08 31</p>

Abb. 4: Drei Arbeitsblöcke einer Benutzergruppensitzung mit entsprechenden Beispielen

Die thematische Gliederung in diese drei Bausteine bewährte sich. Dadurch, daß die Liste der Festlegungen fortlaufend weitergeführt und kommentiert wurde, war über den Projektverlauf hinweg eine ständige Kontrolle über das Erreichen der Arbeitsschritte möglich. Die Beantwortung der offenen Fragen bot eine weitere Möglichkeit der Überwindung arbeitsteiliger Projektorganisation, da alle an der Benutzergruppe Beteiligten für die Beantwortung einzelner Fragen verantwortlich zeichneten. Die konkreten Arbeitsaufgaben erlaubten einen einheitlichen roten Faden der Evaluation und bildeten gleichzeitig einen Aufsatzpunkt, von dem aus die Benutzer den Prototypen selbstständig explorierten.

Bei der Formulierung der Arbeitsaufgaben ist zu beachten, daß ihr Komplexitätsgrad so gewählt wird, daß der Handlungsfluß bei der Bearbeitung des Prototypen nicht zu stark gestört wird und dennoch die Schwachstellen des Prototypen in aller Deutlichkeit zum Vorschein kommen, die für den jeweiligen Entwicklungsstand relevant sind.

Die gewählte Vorgehensweise diente sowohl der Veranschaulichung der Software für die Benutzer als auch zur Strukturierung von Ideen der Entwickler. Zudem wurde den Softwareentwicklern klar, daß bei der wachsenden Komplexität heutiger Software eine Beteiligung der Benutzer unverzichtbar ist.

Sobald die Anwender darangingen, die jeweiligen Prototypen anhand konkreter

Arbeitsaufgaben zu überprüfen, mußte allerdings immer wieder eine schlagartige Flucht von einigen Entwickler hinter ihre Bildschirme verhindert werden. Desgleichen machte sich häufig Unmut bei den Benutzern breit, wenn der zur Testung vorgelegte Prototyp nicht ihren Erwartungen entsprach, d.h. erster Linie nicht die Erledigung ihrer Arbeitsaufgaben erfüllte.

Softwareentwickler haben offenbar Schwierigkeiten, unvollständige Produkte zur Bewertung durch Benutzer vorzulegen, und Benutzer sehen nicht immer den Unterschied zwischen einer sich in der Entwicklung befindlichem Produkt - eben einem Prototypen - und einem fertigen Produkt.

Hier ist es die Aufgabe der Arbeitswissenschaft, sowohl die Softwareentwickler hinsichtlich Benutzerorientierung, d.h. stärkerer Offenheit und Toleranz gegenüber Kritik zu qualifizieren, als auch die Benutzer darüber aufzuklären, daß ein Prototyping gezielt Unvollständigkeit impliziert, um frühzeitig Korrekturen vornehmen zu können. Deshalb empfiehlt es sich aufgrund dieser Erfahrungen, neben den Schulungen mit den Entwicklern, vor den ersten Benutzergruppensitzungen eine Schulung für die Benutzer konzipieren, in der ihnen die grundlegenden Möglichkeiten und auch Restriktionen des Software-Entwicklungs-Tools und die Vorgehensweise eines Prototypings erläutert werden.

Ferner ist es wichtig, daß bei Benutzergruppensitzungen ein unabhängiger Moderator anwesend ist, der immer wieder eine Brücke zwischen Benutzern und Entwicklern schlägt. Desweiteren sollte jeder Benutzergruppensitzung ein Résumé folgen, in dem Benutzer und Softwareentwickler ihre gegenseitigen Probleme zusammenfassen.

5.2 Die Kundengruppe

Die Kundengruppe wurde mit dem Ziel installiert, die gemeinsamen Grobanforderungen an die zu entwickelnde Verlagssoftware zu definieren. An der Kundengruppe nahmen aus jedem der vier an der Entwicklung beteiligten Verlage 1-2 Personen teil. Bei diesen Personen handelte es sich um Mitglieder des höheren und mittleren Managements sowie um EDV-Leiter. Die Kundengruppe tagte ein bis zwei Mal pro Modul.

Der Gewinn dieser Kundengruppen lag darin, daß dadurch zum einen eine Partizipation von betriebliche Entscheidungsträgern an dem Entwicklungsprozeß bewirkt wurde und daß zum anderen für die Benutzergruppen eine Strukturierung der wichtigen Themenbereiche erarbeitet wurde.

Allerdings herrschte einige Male Uneinigkeit bezüglich der Anforderungen, die auf der einen Seite von der Kundengruppe und auf der anderen Seite von der Benutzergruppe formuliert wurden. Dabei lagen die Gründe nicht nur darin, daß das Management in manchen Fällen zu wenig über die aktuellen Arbeitsabläufe wußte.

Ein Problem stellte auch der Umstand dar, daß in den Verlagen der Informationsfluß zwischen der EDV-Leitung und den Benutzern der einzelnen Abteilungen nicht immer optimal war. Auch dies hatte zur Folge, daß Festlegungen und damit bereits durchgeführte Teilentwicklungen redefiniert werden mußten.

Es empfiehlt sich deshalb, sowohl vor den ersten Grobdefinitionen als auch im weiteren Verlauf einer Softwareentwicklung die arbeitswissenschaftlich Arbeit auf die beteiligten Unternehmen auszuweiten verlagern, damit dort eine bessere Kommunikation zwischen den betrieblichen Entscheidungsträgern und den Benutzern stattfindet.

5.3 Der Projektbeirat

Der Projektbeirat überprüfte den Verlauf des Vorhabens insbesondere hinsichtlich der Erfüllung der finanziellen und inhaltlichen Förderung durch den Projektträger A&T. Der Projektbeirat setzte sich aus Vertretern von Interessenverbänden, der Universität, des BMFT und des Softwarehauses zusammen.

Der Projektbeirat, der durch Quartalsberichte auf die vierteljährlich stattfindenden Sitzungen vorbereitet wurde, bewährte sich als nützlicher Ratgeber, der dazu beitrug, inhaltliche Anregungen für weitere Projektschritte zu formulieren. Obendrein wurde insbesondere durch die Teilnahme der Interessenvertreter der Zugang zu weiteren Verlagen ermöglicht. Dies schaffte die Voraussetzung dafür, daß in weiteren Verlagen mittels des Softwareevaluationsverfahrens und des arbeitsanalytischen Verfahrens bei weiteren Stichproben vergleichende Daten erhoben werden konnten (zu ersten Ergebnissen siehe: Prümper, 1993).

6 Zusammenfassung

In diesem Beitrag wurde von einer Softwareentwicklung für klein- und mittelständische Verlage berichtet, die unter arbeitswissenschaftlicher Betreuung in enger Zusammenarbeit mit den potentiellen Endbenutzern durchgeführt wurde.

Bei der Wahl des *Software-Entwicklungs-Tools* fiel die Entscheidung auf das CASE-Tool LANSa. Aufgrund des Umstandes, daß die aktuelle GUI-Version von LANSa noch zu unausgegoren war, konnte die Maskengestaltung nicht in ihrer endgültigen Form evaluiert werden. Zudem zeigte sich, daß CASE-Tools aufgrund ihrer Komplexität höhere Anforderungen an das Projektmanagement stellen als herkömmliche Verfahren.

Zu Beginn des Projektes wurden *arbeitswissenschaftliche Schulungen* durchge-

führt. Es zeigte sich, daß dieser geschlossene Schulungsblock keine praxisgerechte Art der Seminarorganisation darstellt. Es wird empfohlen, eine natürlichere Verknüpfung der arbeitswissenschaftlichen Schulungen mit dem Projektverlauf anzustreben und den jeweiligen Schulungen eine intensive praktische Umsetzung folgen zu lassen.

Bei der *qualitativen Analysen der Software* stand eine Feinanalyse im Vordergrund, in der die einzelnen Arbeitsschritte und die jeweils auftretenden Probleme bei der Bearbeitung einer Standardaufgabe in ein Ablaufdiagramm gebracht wurden. Diese Vorgehensweise bewährte sich, da sie einen guten Einblick in die computerunterstützten Tätigkeiten einzelner Mitarbeiter ermöglichte. Zu einer ganzheitlich orientierten Optimierung sind allerdings weitere Analysen notwendig, welche es gestatten, die synchrone Repräsentation eines integrierten Netzwerkes von Arbeits- und Organisationseinheiten abzubilden.

Zur *quantitativen Analyse der Software* wurde ein Benutzer-Fragebogen auf Grundlage des Entwurfs zur internationalen Ergonomie-Norm ISO 9241 Teil 10 entwickelt. Neben der standardisierten Bewertung der Software diente er als Grundlage für die Moderation mit Benutzergruppen, als mit ihnen erste Designanforderungen erarbeitet wurden. Das Ergebnis dieser Vorgehensweise war eine Fülle von konkreten Hinweisen auf Schwachstellen.

Die *qualitative Analyse der Arbeitssituation* wurde über die Analyse der Software und über Interviews mit Benutzern angestrebt. Es zeigte sich, daß dies nicht genügt, um einen ausreichenden Einblick in die Abläufe eines Unternehmens zu gewinnen.

Zur *quantitativen Analyse der Arbeitssituation* wurde aus bestehenden arbeitswissenschaftlichen Instrumenten ein Fragebogen zur "Einschätzung der Arbeitstätigkeit" entwickelt. Neben der quantitativen Erfassung der Arbeitssituation bietet er sich "screening"-Instrument zur Moderation mit Benutzergruppen an.

Im Rahmen des Projektes wurden eine *Benutzergruppe*, eine *Kundengruppe* und ein *Projektbeirat* ins Leben gerufen. Die Benutzergruppe definierte und überprüfte die Feinanforderungen und die Kundengruppe die Grobanforderungen an die zu entwickelnde Verlagssoftware. Der Projektbeirat überprüfte den Verlauf des Vorhabens hinsichtlich der Erfüllung der Förderung durch den Projektträger. In den Benutzergruppen bewährte sich eine thematische Gliederung in die drei Bausteine Überprüfung von Festlegungen, Klärung offener Fragen und Überprüfung des Prototypen anhand konkreter Arbeitsaufgaben. Es zeigt sich, daß der Einbezug der Benutzer für die Gestaltung der Software unabdingbar war. Einige Male herrschte Uneinigkeit hinsichtlich der Anforderungen, die zum einen von der Kundengruppe und zum anderen von der Benutzergruppe formuliert wurden. Es empfiehlt sich, die arbeitswissenschaftliche Arbeit auf die beteiligten Unternehmen auszuweiten, um dort eine bessere Kommunikation zwischen den betrieblichen Entscheidungsträgern und den Benutzern zu koordinieren. Der Projektbeirat bewährte sich als nützlicher

Ratgeber, durch dessen Teilnahme der Zugang zu weiteren Verlagen ermöglicht wurde, in denen ergänzende Untersuchungen stattfinden konnten.

Literatur

- Aschersleben, G., Gstalter, H., Kaiser, F., Strube, V., & Zang-Scheucher, B. (1989). Prototyping als Verfahren zur Software-Entwicklung. Literaturanalyse und Expertengespräche. *Zeitschrift für Arbeitswissenschaft*, *43*, 42-47.
- Aschersleben, G., & Zang-Scheucher, B. (1989). Der Prozeß der Software-Gestaltung - Eine Bestandsaufnahme in Wissenschaft und Industrie. In S. Maaß & H. Oberquelle (Hrsg.) *Software-Ergonomie '89*, 244-253. Stuttgart: Teubner.
- Baroudi, J., Olson, M., & Ives, B. (1986). An empirical study of the impact of user involvement on system usage and information satisfaction. *Communication of the ACM*, *29*, 232-238.
- Bewley, W.L., Roberts, T.L., Schroit, D., & Verplanck, W.L. (1983). Human factors testing in the design of XEROX's 8010 'Star' office work station. *Proceedings of CHI'83 conference on human factors in computing systems*, 72-77. Boston.
- Bieger, E., Feltes, T., Schmalzriedt, L., & Sieber, U. (1981). *Spielregeln für Kursleiter. Wie plane und leite ich Kommunikationskurse*. Gelnhausen: Burckhardtthaus.
- Böhle, F., & Milkau, B. (1988). *Vom Handrad zum Bildschirm. Eine Untersuchung zur sinnlichen Erfahrung im Arbeitsprozeß*. Frankfurt am Main: Campus.
- Boehm, B.W., Gray, T., & Secwald, T. (1984). Prototyping versus specifying: a multiproject experiment. *IEEE Transactions on Software Engineering*, *10*, 224-236.
- Brown, C.M. (1989). *Human-Computer Interface Design Guidelines*. Norwood, New Jersey: Ablex.
- Connell, J.L., & Shafer, L. (1989). *Structured rapid prototyping: an evolutionary approach to software development*. Englewood Cliffs, New York: Yourdon Press.
- Flanagan, J.G. (1954). The critical incident technique. *Psychological Bulletin*, *51*, 327-358.
- Floyd, C. (1984). A systematic look at prototyping. In R. Budde, K. Kuhlenkamp, L. Mathiassen, & M. Züllighoven (Eds.), *Approaches to prototyping* (S.1-18). Berlin: Springer.
- Focus (1991). CASE hat viele Gesichter. *Focus 3* (Beilage zur Computerwoche vom 28.6.1991).
- Frese, M., & Brodbeck, F.C. (1989). *Computer in Büro und Verwaltung*. Berlin: Springer.
- Frese, M., Irmer, C., & Prümper, J. (1991). Das Konzept Fehlermanagement: Eine Strategie des Umgangs mit Handlungsfehlern in der Mensch-Computer Interaktion, In M. Frese, Chr. Kasten, C. Skarpelis & B. Zang-Scheucher (Hrsg.), *Software für die Arbeit von morgen. Bilanzen und Perspektiven anwendungsorientierter Forschung*, (S.241-251). Berlin: Springer.
- Frese, M. & Zapf, D. (Hrsg.) (1991). *Fehler bei der Arbeit mit dem Computer. Ergebnisse und Befragungen im Bürobereich*. Bern, Göttingen, Toronto: Huber.
- Gould, J.D., & Lewis, C. (1985). Designing for usability - Key principles and what designers think. *Proceedings of CHI'83 conference on human factors in computing systems*, 50-53, Boston.
- Gunzenhäuser, R. (Hrsg.) (1988). *Prototypen benutzergerechter Computersysteme*. Berlin: de Gruyter.
- Hacker, S., Müller-Holz auf der Heide, B., & Aschersleben, G. (1991). Prototyping in einem Designteam - Vorgehen und Erfahrungen bei einer Software-Entwicklung unter Benutzerbeteiligung. In M. Frese, Chr. Kasten, C. Skarpelis, & B. Zang-Scheucher (Hrsg.) *Software für die Arbeit von morgen: Bilanz und Perspektiven anwendungsorientierter Forschung* (S. 179-189). Berlin, Heidelberg, New York: Springer.
- Hackman, J.R., & Oldham, G.R. (1975). Development of the job diagnostic survey. *Journal of Applied Psychology*, *60*, 159-170.
- Hallmann, M. (1990). *Prototyping komplexer Softwaresysteme: Ansätze zum Prototyping und Vorschlag einer Vorgehensweise*. Stuttgart: Teubner.

- Heeg, F.J., & Neuser, R. (1988). *Nutzergerechte Ausgestaltung von Software durch Prototyping - Grundlagen, Vorgehensweise, Wirtschaftlichkeitsaspekte*. Düsseldorf: VDI-Verlag.
- Heilmann, H. (1981). *Modelle und Methoden der Benutzermitwirkung in Mensch-Computer-Systemen*. Stuttgart, Wiesbaden: Forkel.
- Heinbokel, T., Frese, M., Stolte, W., Brodbeck, F.C., & Sonnentag, S. (1993) *Don't underestimate the problems of user involvement in software development - there are many!* Manuscript.
- Hierhold, E. (1990). *Sicher präsentieren - wirksam vortragen*. Wien: Ueberreuter.
- Hoffmann, T., Klose, H.G., & Martin, H. (1989) *Handbuch zur software-ergonomischen Gestaltung von Bildschirmmasken*. Düsseldorf: VDI-Verlag.
- ISO 9241: Ergonomic requirements for office work with visual display terminals (VDTs). (1992). *Part 10: Dialogue principles - 1st DIS*, November 1992.
- Karat, C.M. (1990). Cost-Benefit-Analysis of Iterative Usability testing. In D. Diaper, D. Gilmore, G. Cockton, & B. Shackel (Hrsg.), *Human-Computer Interaction - INTERACT '90*, 351-356. Amsterdam: Elsevier.
- Karer, A. (1992). CASE-Erfahrungen haben die "Hysterie" gebremst. *Computerwoche*, 30, 23-25.
- Kieback, A., Lichter, H., Schneider-Hufschmidt, M., & Züllighoven, H. (1991) *Prototyping in industriellen Software-Projekten*. Sankt Augustin: GMD.
- Kleibit, K., Schrader, E., & Straub, W.G. (1985). *KurzModeration: Anwendung der Moderationsmethode in Betrieb, Schule und Hochschule, Kirche und Politik, Sozialbereich und Familie bei Besprechungen und Präsentationen*. Hamburg: Windmühle.
- Kuscher, G. (1992). Sanftes Hinüberwachsen. Software-Entwicklung am Beispiel des CASE-Tool Lansa. *DV-Dialog*, 4, 18.
- Mährländer, H.J. (1991). Die CASE-Einführung braucht ein gutes Projektmanagement. *Computerwoche*, 38, 12-14.
- Mantey, M.M., & Teorey, T.J. (1988). Cost/benefit analysis for incorporating human factors in the software lifecycle. *Communications of the ACM*, 31, 428-439.
- Mauch, H. (1981). *Werkstattzirkel - wie Arbeiter und Meister an der Lösung betrieblicher Probleme beteiligt werden können*. Quickborn: McIaplan GmbH.
- Peschke, H., & Wittstock, M. (1987). Benutzerbeteiligung im Software-Entwicklungsprozeß. In K.P. Fährich (Hrsg.), *Software-Ergonomie* (S.81-92). München: Oldenbourg.
- Plath, H.E., & Richter, P. (1984) *Ermüdung - Monotonie - Sättigung - Streß (BMS). Handanweisung*. Berlin: Psychodiagnostisches Zentrum, Sektion Psychologie der Humboldt Universität.
- Prümper, J. (1993). Wie benutzerfreundlich ist Ihre Software? Eine kritische Bestandsaufnahme der Softwaresituation im Verlagswesen. *Börsenblatt*, (im Druck).
- Prümper, J., & Anft, M. (1993). 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. Rödiger (Hrsg.), *Software-Ergonomie '93*. Stuttgart: Teubner, (im Druck).
- Prümper, J., & Hartmannsgruber, K. (1993). *ESAT - Einschätzung der Arbeitstätigkeit aufgrund arbeitswissenschaftlicher Erkenntnisse*. Manuskript.
- Rose, H. (1984). Neue Belastungsformen an computergestützten Arbeitsplätzen. In R. Crusius & J. Stebani (Hrsg.), *Neue Technologien und menschliche Arbeit: Stand und Entwicklung der Steuerungs-, Informations- und Kommunikationstechnologien in der Arbeitswelt*. Berlin: Die Arbeitswelt.
- Rosenstiel, L. von, Falkenberg, T., Hehn, W., Henschel, E., & Warns, I. (1983) *Betriebsklima heute*. München: Bayerisches Staatsministerium für Arbeit- und Sozialordnung.
- Rosson, M.B., Maass, S., & Kellogg, W.A. (1987). Designing for designers: An analysis of design practice in the real world. *Proceedings of CHI'87 conference on human factors in computing*

- systems*, 137-142. Toronto.
- Semmer, N. (1984). *Stressbezogene Tätigkeitsanalyse*. Weinheim, Basel: Beltz.
- Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction*. Reading, Massachusetts: Addison-Wesley.
- Smith, S.L., & Mosier, J.N. (1986). *Guidelines for designing user interface software*. Bedford, Mass.: MITRE.
- Strohm, O. (1991). Projektmanagement bei der Software-Entwicklung: Eine arbeitspsychologische Analyse und Bestandsaufnahme. In D. Ackermann & E. Ulich (Hrsg.) *Software-Ergonomie '91*, 46-58. Stuttgart: Teubner.
- Udris, I. & Alioth, A. (1980). Fragebogen zur subjektiven Arbeitsanalyse. In E. Martin, U Ackermann, I. Udris & K. Oegerli (Hrsg.), *Monotonie in der Industrie* (S.204-207). Bern, Göttingen, Toronto: Huber.
- Vonk, R. (1990). *Prototyping: the effective use of CASE technology*. New York: Prentice Hall.
- Williges, R.C., Williges, B.H., & Elkerton, J. (1987). Software interface design. In G. Salvend (Hrsg.), *Handbook of human factors* (S.1416-1449). New York: John Wiley & Sons.
- Zang, B., & Gstalter, H. (1987). Erfahrungen bei der Entwicklung und Einführung von rechnergestützten Systemen im Bürobereich. *Zeitschrift für Arbeits- und Organisationspsychologie*, **31**, 115-118.
- Zapf, D. (1991). Stressbezogene Arbeitsanalyse bei der Arbeit mit unterschiedlichen Bürosoftwaresystemen. *Zeitschrift für Arbeits- und Organisationspsychologie*, **35**, 2-14.
- Zapf, D., Brodbeck, F.C., & Prümper, J. (1989). Handlungsorientierte Fehler-taxonomie in der Mensch-Computer Interaktion. *Zeitschrift für Arbeits- und Organisationspsychologie*, **33**, 178-187.
- Zapf, D., Frese, M., Irmer, C., & Brodbeck, F.C. (1991). Konsequenzen von Fehleranalysen für die Softwaregestaltung. In M. Frese & D. Zapf (Hrsg.), *Fehler bei der Arbeit mit dem Computer. Ergebnisse und Befragungen im Bürobereich* (S.177-191) Bern, Göttingen, Toronto: Huber.

Quelle:

Prümper, J. (1993). Benutzerorientierte, iterative Software-Entwicklung in der Praxis. In W. Coy, P. Gorny, I. Kopp & C. Skarpelis (Hrsg.), *Menschengerechte Software als Wettbewerbsfaktor* (S. 630-647). Stuttgart: Teubner.