

Integration software-ergonomischer Forschungsergebnisse in die betriebliche Software-Entwicklung

Erfahrungen, Strategien, Potentiale und Probleme

Edmund Eberleh, Walldorf

mit Beiträgen von U. Arend, C. Kasten, T. Strothotte und J. Ziegler

Zusammenfassung

Seit mehr als einem Jahrzehnt findet mittlerweile Forschung zur Software-Ergonomie statt. Die Ergebnisse dieser Bemühungen sind in einer Fülle von Veröffentlichungen und Experimentalsystemen dokumentiert. Die meisten Benutzer müssen jedoch zwangsläufig mit *der* Software arbeiten, die von den Softwarehäusern entwickelt wird und auf dem Markt tatsächlich verfügbar ist. In diesem Sinne sind betriebliche Softwareentwickler die (oftmals unfreiwilligen) Vermittler und Umsetzer der software-ergonomischen Forschungsergebnisse. Die hehren Ziele der benutzergerechten und aufgabenangemessenen Softwaregestaltung sind nur insoweit tatsächlich realisiert, als sie sich auch in kommerziellen Softwareprodukten niederschlagen. Hier existieren jedoch eine Reihe von Barrieren zwischen Forschung und Entwicklung, die diesen wünschenswerten Ergebnistransfer hemmen (können). Nach einer kurzen Skizzierung dieser Barrieren werden in vier Beiträgen Erfahrungen und Lösungsstrategien zu diesem Problemkreis diskutiert.

1. Problemlage

Schauen wir zur Einstimmung in die Thematik einmal einer Kollegin bei ihrer Arbeit über die Schulter: Frau Hilfreich ist Mitarbeiterin in einem Softwarehaus im Bereich Software-Ergonomie. Sie arbeitet in einem Entwicklungsteam mit, welches ein bestehendes Programm für eine neue Version grundlegend überarbeitet. Frau Hilfreich hat dadurch die Möglichkeit, eine Reihe ergonomischer Verbesserungen in das neue Produkt einzubringen. Auch die Entwickler legen viel Wert auf eine verbesserte Benutzbarkeit der Software.

Als Grundlage der Neugestaltung dienen Designrichtlinien eines führenden Computerherstellers und bereits bestehende Standards der Oberflächengestaltung. Frau Hilfreich führt darüberhinaus eine Benutzer- und Aufgabenanalyse durch. Sie verwendet dazu standardisierte und validierte Verfahren aus der Literatur. Die Ergebnisse der Analysen führen im Rahmen der vorgegebenen Gestaltungsrichtlinien zur Entwicklung eines ersten Prototypen, der von einigen potentiellen Benutzern oder Pilotkunden beurteilt wird. Für die dabei identifizierten Problembereiche schaut Frau Hilfreich in der Fachliteratur nach Lösungen nach und diskutiert sie informell mit einigen Kollegen aus der Hochschule und von Forschungsgesellschaften. Ihre so entstandenen Vorschläge zur Verbesserung des Prototypen diskutiert sie anschließend mit den übrigen Mitgliedern des

Entwicklungsteams. Ein von ihnen nicht lösbares bzw. bearbeitbares Problem wird als Auftrag an eine Forschungsinstitution abgegeben. Zusammen mit den weiteren Ergebnissen des Prototyping fließen diese Lösungen in die vorläufig letzte Version des neuen Produkts ein. Es wird abschließend mit einem standardisierten Evaluationsverfahren auf Einhaltung der zentralen software-ergonomischen DIN-Kriterien geprüft.

So etwa könnte die Entwicklung eines Software-Produktes unter Berücksichtigung aktueller software-ergonomischer Erkenntnisse aussehen (vgl. Schmidt, 1990). In der Praxis wird dieser Idealablauf allerdings nur mit Abstrichen realisiert werden können (vgl. Aschersleben & Zang-Scheucher, 1989).

In diesem Papier bzw. in der Diskussionsgruppe sollen Erfahrungen und Strategien von Software-Ergonomen aus Forschung und Entwicklung ausgetauscht werden und Potentiale und Probleme der Integration software-ergonomischer Forschungsergebnisse in die betriebliche Softwareentwicklung diskutiert werden. Ziel und Hoffnung hierbei ist, frühere Mißerfolge und Sackgassen zukünftig zu vermeiden und Strategien zu ermitteln, die das Wissen um eine benutzergerechte Softwaregestaltung direkter und schneller in der in der Arbeitswelt benutzten Software abzubilden erlauben.

Zur Identifikation und Systematisierung möglicher Hemmnisse und Problembereiche sei die Entwicklung eines Software-Produktes grob und idealtypisch in verschiedene Phasen aufgeteilt: Eine theoretische Forschungsphase (I) und eine praktische Entwicklungsphase (II). Innerhalb beider Phasen lassen sich weitere Teilphasen identifizieren:

I Forschung	1 Fragestellung
	2 Lösung
II Entwicklung	3 Kenntnisnahme der Lösung
	4 Berücksichtigung in Entwicklung
	5 Produkt

Eine optimale Berücksichtigung software-ergonomischer Forschungsergebnisse bedingt einen vollständigen und schnellen Informationsfluß von Teilphase 1 zu Teilphase 5 (und von dort zurück zu Teilphase 1). Sowohl zwischen der Forschungs- und Entwicklungsphase als auch zwischen den einzelnen Teilphasen können jedoch Barrieren bestehen, die den Informationsfluß behindern.

2. Barrieren bei der Integration software-ergonomischer Erkenntnisse

Barrieren zwischen den Hauptphasen Forschung und Entwicklung

1. Die Fragestellungen und Lösungen sind für das Produkt irrelevant (Teilphase 1# Teilphase 5).
2. Das Produkt berücksichtigt nicht die relevanten Lösungen oder Anforderungen (5#1).

3. Die Lösungen werden nicht zur Kenntnis genommen bzw. sind (noch) nicht bekannt und zugänglich (2#3).

Barrieren zwischen Teilphasen innerhalb der einzelnen Hauptphasen

4. Die Problemlösung ist zeitlich und/oder personell so aufwendig, daß in absehbarer Zeit keine praktisch verwendbare Lösung zu erwarten ist (1#2).
5. Bekannte Problemlösungen werden bei der Produktentwicklung nicht berücksichtigt (3#4).
6. Bei alternativen Produktprototypen wird gegen eine benutzergerechte Version entschieden (4#5).

Beispielhafte Konkretisierung der einzelnen Barrieretypen

- 1a. Die Hochschulforschung ist für die betriebliche Softwareentwicklung irrelevant. Es besteht mangelnde Generalisierbarkeit und Übertragbarkeit der Forschungsergebnisse auf Anwendungsprobleme (vgl. Grünupp & Muthig, 1990). Die Fragestellungen sind oft nur auf wissenschaftlichen Kontext beschränkt (z.B. nur Bürosoftware oder wissenschaftliche Anwendungen). Die Forschung bearbeitet Probleme, die für die Praxis erst in Jahren relevant werden.
Die Hochschule kennt die betrieblichen Probleme nicht.
Die Hochschule ist oft mehr an Grundlagenforschung interessiert. Auftragsforschung ist oft nur Mittel zur Geldbeschaffung.
Zu allgemeine Lösungen, wie bei einer Reihe software-ergonomischer Checklisten. Das Problem steckt jedoch oft im Detail.
- 1b. Die zentrale Ergonomie-Abteilung eines Softwarehauses bearbeitet Fragestellungen, die für die Entwicklungsabteilungen irrelevant sind.
2. Die Betriebe, für die Software entwickelt wird, haben nicht die neueste Technologie bzw. Hardware. Bei der Software-Entwicklung für diese ältere Hardware können Forschungsergebnisse, die oft von neuester Hardware ausgehen, nicht angewendet werden.
3. Die Industrie ist aus dem "Verteiler" der Hochschulinformationen ausgeschlossen, nicht in deren Kommunikationskanäle integriert.
Lösungen werden so theoretisch-abstrakt formuliert, daß sie nur schwer von Praktikern mit länger zurückliegender (Hochschul-) Ausbildung verfolgt werden können.
Praktiker haben wenig Zeit für intensives und kontinuierliches Literaturstudium. Lohnt sich der Aufwand der kontinuierlichen Weiterbildung in der wissenschaftlichen Fachliteratur?
4. Im Labor theoretisch mögliche Lösungen werden in realistischen Anwendungsgebieten oft nicht mehr mit vertretbarem Zeit- und Kostenaufwand handhabbar (z.B. GOMS-Modellierung)
5. Der Entwicklungsprozeß ist so dynamisch, daß eine systematische und fundierte Anwendung und Übertragung der prinzipiell bekannten Methoden und Ergebnisse nicht möglich ist. Schnelle

Lösungen sind gefragt.

Vorschläge aus dem eigenen Hause werden nicht so gewichtet wie von externen Institutionen (Motto: der Prophet gilt nichts im eigenen Lande). Eine software-ergonomische Stabsabteilung schafft Distanz zu Entwicklern, ist oft aber nötig um Entwicklung im Überblick zu koordinieren und komplexe Produkte konsistent zu gestalten.

Kommunikationsschwierigkeiten und Konkurrenz zwischen Personen verschiedener Fachdisziplinen und Organisationshierarchien.

6. Eine zu kurz gegriffene Kosten-Nutzen Rechnung führt zur Ablehnung eines vorerst aufwendiger zu entwickelnden Software-Produktes. Die später entstehenden Kosten durch mangelnde Akzeptanz, Motivation, Krankheit und Verkaufsrückgang werden nicht mit einbezogen.

3. Einige Themenkreise, die auf dieser Grundlage diskutiert werden könnten:

- Gibt es aufgrund der Erfahrungen bestimmte häufig auftretende Barrieren? Gibt es besonders schwerwiegende Barrieren? Wie wurden diese Probleme bisher von den einzelnen bearbeitet oder gelöst? Lassen sich gewisse Problemlösestrategien identifizieren ?
- Haben wir einen Stand der software-ergonomischen Erkenntnisse erreicht, ab dem anwendungsbezogene Forschung nicht mehr so wichtig ist, da es bereits eine Reihe von Richtlinien und Normen gibt ? Bleibt nur noch eine gewisse Konkretisierung von kontextspezifischen Detailfragen übrig? Kann (muß) jedes Softwarehaus seine eigenen Probleme lösen?
- Ist das eigentlich benötigte Wissen vielleicht mehr ein implizites Designwissen (vgl. Carroll, 1989)? Bilden die Hochschulen dafür aus? Wie kann man dieses Wissen erwerben?
- Können allgemeine Designregeln formuliert werden, oder ist die Wirkung einer Oberfläche so sehr von Wechselwirkungen mit anderen Gestaltungsfaktoren abhängig, daß sowieso der konkrete Einzelfall untersucht werden muß?
- Wie ist das Verhältnis Hochschule-Hersteller: Soll die Hochschule überhaupt anwendungsrelevante, betriebliche Probleme lösen? Sind Hersteller überhaupt an Hochschul-Forschung interessiert? Probleme bei Aufträgen an Hochschule und Forschungsinstitutionen: Die wissenschaftlichen Mitarbeiter müssen sich in die oft komplexe Funktionalität einarbeiten, kennen interne sachliche und personelle Randbedingungen des Software-Hauses nicht.
- Verhältnis Staat-Wirtschaft: Was wird von öffentlichen Mitteln gefördert ? Nur die Forschung und Problemlösung oder auch die Umsetzung der Lösung in ein Produkt?

In den folgenden vier Beiträge (in alphabetischer Folge) wird dieser Fragenkomplex diskutiert von einem in einem Softwarehaus tätigen Software-Ergonomen, von einem Vertreter des BMFT-Programms Arbeit und Technik, einem Hochschulangehörigen sowie einem Mitarbeiter eines angewandt arbeitenden Forschungsinstituts.

Udo Arend, SAP AG, Walldorf

1. Software-Ergonomie innerhalb der SAP AG

Zur Zeit umfaßt die Software-Ergonomiegruppe bei der SAP AG sieben Ergonomen, die sich interdisziplinär zusammensetzen (Informatiker, Psychologen und eine Betriebswirtin). Basis für die Arbeit der Software-Ergonomiegruppe stellt die Entscheidung der Firmenleitung dar, bei der Weiterentwicklung des R/2-Systems für die Gestaltung der Bedienoberfläche die CUA-Richtlinien der IBM (Common User Access) zu verwirklichen.

Zu den Aufgaben der Ergonomiegruppe gehören die Beratung der Anwendungsentwickler bei der Oberflächengestaltung und die Erstellung von SAP-Standards zur Oberflächengestaltung. Einen breiten Raum der täglichen Arbeit nimmt die Beratung der Anwendungsentwickler und die Koordinierung der unterschiedlichen Anwendungsbereiche ein. Dabei zeigt sich, daß sich die größten Fortschritte in der Entwicklung der Bedienoberfläche dann ergeben, wenn Ergonom und Entwickler gemeinsam Konzepte entwickeln und realisieren. Die speziell entwickelten Normen bilden dabei den Rahmen, der eine Vereinheitlichung der Bedienoberfläche fördert.

Herausfordernde Randbedingungen bei der konzeptionellen Gestaltung der SAP-Software durch die Software-Ergonomie sind:

- a) Die SAP-Software ist ein funktional sehr mächtiges, vollständig aus verschiedenen Modulen integriertes Softwareprodukt, das den gesamten betriebswirtschaftlichen Bereich abdeckt. Die Komplexität des Systems und die herausragende Bedeutung der Interaktion über Folgen von Bildschirmmasken erlaubt jedoch keine einfache Übertragung der CUA-Richtlinien. Dazu kommt, daß die SAP-Software sowohl auf Großrechnern mit "dummen" Terminals als auch auf Workstations (UNIX-Welt) und PCs (OS/2) mit graphikfähigen Terminals ablauffähig sein muß.
- b) Komplexe betriebswirtschaftliche Abläufe müssen in eine funktional angepaßte Ablaufstruktur umgesetzt werden. Dazu ist eine genaue Kenntnis von typischen Arbeitsplätzen und deren Anforderungen in der Industrie erforderlich. Je nach Industriebranche und Größe des Betriebes, aber auch innerhalb eines Betriebes unterscheiden sich Arbeitsplätze mit ähnlichen Aufgaben jedoch beträchtlich hinsichtlich ihrer Anforderungen an die Funktionalität des Systems.

2. Forderungen der betrieblichen Ergonomie

Da zeitaufwendige Untersuchungen sowohl bei den Endanwendern (z.B. Aufgabenanalysen, Arbeitsplatzanalysen, Beobachtungen usw.) als auch aufwendiges Rapid Prototyping oder zeitverschlingende Evaluationsstudien im Haus nur eingeschränkt möglich sind, müßte der Input über diese Themenbereiche auf eine andere Weise erfolgen. Es existieren zwar diverse "Guidelines", diese sind jedoch entweder viel zu allgemein, so daß keine Designentscheidungen ableitbar sind, oder so spezifisch, daß sie für den genannten Problembereich nicht einsetzbar sind (z.B. ein Kompendium zu "Human Perception and Performance"). Handbücher - ähnlich wie in den Ingenieurwissenschaften - die entweder eine "harte" Entwurfsmethodik beinhalten oder wenigstens Wissen über bestimmte Themenbereiche (z.B. betriebswirtschaftliche Arbeitsabläufe, Arbeitsplatzbeschreibungen) katalogartig verfügbar machen, fehlen.

Die Methodik zur systematischen Aufgabenmodellierung bevor mit der Entwicklung begonnen wird, als auch "Toolbaukästen", die Usability Tests in einer schnellen standardisierten Form ermöglichen, sind nicht ausreichend entwickelt. Tools zum Rapid Prototyping existieren zwar, die damit entwickelten Programme sind aber nicht auf die SAP-Welt überführbar. Zudem erlauben solche Tools nicht die Definition eigener Oberflächenkonstrukte, sondern geben vorgefertigte Konstrukte vor.

So bleibt der Ergonom in der Praxis faktisch auf sein akademisch erworbenes ergonomisches Wissen und seine eigene Erfahrung beim Softwareentwurf bzw. im Umgang mit Anwendern angewiesen. Hat T. Winograd (Vortrag auf der CHI'90-Conference) also Recht, wenn er von Softwaregestaltung als einer "Designwissenschaft" ähnlich wie der Architektur spricht?

3. Barrieren zwischen Hochschule und Industriepraxis

Ich möchte bestehende Barrieren zwischen Hochschule und Praxis als Thesen formulieren:

These 1: Die Hochschulforschung kümmert sich nicht um Belange, die aus der Praxis kommen. Es wird immer noch über Texteditoren geforscht, obwohl diese im gesamten Softwaremarkt nur eine untergeordnete Rolle spielen.

These 2: Ergebnisse der Hochschulforschung sind oft so "akademisch", daß sie nicht auf praktische Anforderungen übertragbar sind, oder diese auch gar nicht berücksichtigen.

These 3: Methodenentwicklungen innerhalb der Hochschule erlangen oftmals nur Prototypencharakter. Direkt einsetzbare Methoden (als Toolkits ähnlich wie psychologische Tests) fehlen.

These 4: Die Anforderungen praktischer Softwareentwicklung sind um ein vielfaches komplexer als die im Forschungsbereich untersuchten Probleme. Beispielsweise werden Anforderungen, die durch Industrie-arbeitsplätze gestellt werden (z.B. Massenerfassung von Belegen), nicht berücksichtigt.

These 5: Die Forschung kümmert sich nicht darum, wie softwareergonomisches Arbeiten erfolgreich

unter industriellen Bedingungen erfolgen kann. Dazu gehörten beispielsweise Strategien zur Beratung von Softwareentwicklern oder auch Szenarien zur Entscheidung über grundsätzliche Designalternativen (costs and benefits).

C. Kasten, DLR PT-AUG, Bonn

Beiträge des Programms Arbeit und Technik zur Verbesserung der Softwareentwicklung

Die benutzungs- und aufgabengerechte Ausgestaltung von Softwaresystemen ist von vielfältigen Voraussetzungen und Einflußfaktoren des Herstellungsprozesses abhängig. Über ihr Wirkungsgefüge und Optimierungsmöglichkeiten liegen bisher wenige Erkenntnisse vor. Im Rahmen des Programms Arbeit und Technik des BMFT werden Forschungen und Entwicklungen zur Verbesserung der Softwareentwicklung gefördert. Als zentrale Aspekte, die es in Vorhaben zu erhellen gilt, haben sich aus der Sicht des Programms vor allem folgende herausgestellt:

1. Der Nachweis der Wirtschaftlichkeit der Entwicklung benutzungs- und aufgabengerechter Software ist unter Einbeziehung der Phase der Benutzung und Pflege zu erbringen. Damit lassen sich ggf. die Kostenrestriktionen für den Entwicklungsprozeß, die sich aus einem verengten Wirtschaftlichkeitsverständnis ergeben, aufbrechen.
2. Unterschiedliche Software-Entwicklungsansätze (z. B. objektorientierte, prozeßorientierte und Prototyping Ansätze versus z. B. produktorientierte Ansätze) sind hinsichtlich ihrer Beiträge zur verbesserten Software-Gestaltung vergleichend zu bewerten und auszubauen.
3. Für die benutzungs- und aufgabenbezogene Ausdifferenzierung und Flexibilisierung der Software sind Systemkonzepte und Verfahren einer benutzerseitigen Anwendungsanpassung weiter zu entwickeln.
4. Arbeits- und Betriebsorganisation von Softwarehäusern bzw. Herstellern bedürfen der Überprüfung, inwieweit sie genügend Raum für ganzheitliche anwendungs- und benutzungsbezogene Entwicklungsprozesse bieten. Es mangelt an zukunftsgerichteten ausgereiften organisatorischen Alternativen zu arbeitsteiliger Entwicklungspraxis, die auch eine verbesserte Qualitätskontrolle mit einschließen.

5. Die Ausgestaltung von Entwicklungswerkzeugen muß sich an Formen kooperativer wie auch benutzerorientierter Entwicklungsarbeit ausrichten und darf die Kreativität der Entwickler nicht einschränken. CASE kann auch für notwendige tutorielle Unterstützung genutzt werden.

6. Neben der Weiterentwicklung software-ergonomischer Gestaltungskriterien, insbesondere auf dem Gebiet der Anforderungen aus arbeitsorganisatorischer Sicht - z. B. bezüglich kooperativer Arbeit - und ihrer Differenzierung für bestimmte Bereiche und Branchen sind für die Entwicklungspraxis Vorgehensweisen und Verfahren zur Optimierung unterschiedlicher, zum Teil auch widersprüchlicher Benutzungs- und Entwicklungskriterien von Bedeutung.

7. Es mangelt an Weiterbildungscurricula und -Organisation zur Umsetzung neuer Methoden, Verfahren und Instrumente des Software-Engineering wie auch zur Vermittlung von Erfahrungswissen zur Technikanwendung und Organisationsgestaltung, differenziert nach Branchen und Bereichen.

8. Die Möglichkeiten neuer Ansätze der Ausbildung von Softwareentwicklern, wie z. B. ein neues Berufsbild Informatiker/Softwareergonom, sind in die Überlegungen zur Umsetzung von Software-Ergonomie einzubeziehen.

Thomas Strothotte, FU Berlin

Die heute eingesetzten Verfahren zur Mensch-Computer Interaktion beruhen auf jahrelanger Grundlagenforschung, die an Hochschulen, aber auch zum Teil in industriellen Forschungslabors ihren Ursprung hat. Dabei liegen oft viele Jahre zwischen der einschlägigen Forschung und ihrem Praxiserfolg. Als Beispiel kann hier die direkte Manipulation dienen, die Anfang der siebziger Jahre erforscht wurde, aber erst ein Jahrzehnt später ihren breiten Einsatz fand.

Zu einem gegebenen Zeitpunkt besteht also zwischen der Grundlagenforschung und der für die Entwicklung einsetzbaren Technik und den damit verbundenen Problemstellungen eine große Kluft, die meiner Erfahrung nach in starkem Maße mit den unterschiedlichen Zielsetzungen der jeweiligen Personengruppen verknüpft ist.

Wissenschaftler, die sich mit Grundlagenforschung beschäftigen, wollen neue Begriffe prägen und dabei prinzipielle Lösungsstrategien möglichst breitangelegter Problemklassen aufzeichnen. Ihre Ergebnisse werden möglichst abstrakt formuliert und wirken tiefsinnig ("formal"), auch wenn die zugrundeliegenden Ideen oft einfach sind. Arbeitsintensive Ausarbeitungen werden hier von Fachkollegen nur wenig honoriert. Entwickler dagegen stehen unter ständigem Zeitdruck, um

lauffähige Programme abzuliefern, die den Ansprüchen des Kunden genügen. Kurz gezeichnete Zeitpläne erlauben einem Entwickler nicht, abstrakte Formulierungen eines Wissenschaftlers zu durchforsten. Dazu kommt, daß die Entwickler auch schon von ihrer Ausbildung her das Interesse hierzu oft nicht mit sich bringen.

Bei heutigen Entwicklungsprojekten wird - im Gegensatz zu noch vor wenigen Jahren - fast ausschließlich mit umfangreichen Programmpaketen (sog. Tools) gearbeitet. Manche Interaktionstechniken werden somit einem Entwickler aufgezwungen, zu anderen wird er ermutigt. Enthält beispielsweise ein Tool umfangreiche Funktionen für Menüs, so werden diese eher benutzt, als daß ein Entwickler einen natürlichsprachlichen Dialog ermöglicht, auch wenn dieser ihm als angemessener erscheint. Dagegen hat er Freiheitsgrade bei Fragestellungen, die ihm in der Regel nicht vertraut sind und für die er auch nur wenig Zeit investieren kann (beispielsweise wo auf dem Bildschirm ein Menü plaziert werden soll).

Diese Sachlage eröffnet für Wissenschaftler neue Möglichkeiten, gezielt zu forschen:

1. Erstellen von Tools

Das Ergebnis eines Forschungsprojektes sollte möglichst nicht nur aus einer wissenschaftlichen Abhandlung und einer prototypischen Anwendung bestehen, sondern auch aus einem oder mehreren Tools, in denen sich die neuen Erkenntnisse widerspiegeln. Auf diese Weise sollten einem Entwickler die Erkenntnisse anwendungsnah zur Verfügung gestellt werden, damit er - ohne sich mit den wissenschaftlichen Details zu beschäftigen - an den technischen Problemen arbeiten kann.

2. Skilltransfer an Entwickler

Mehr Aufwand sollte von Wissenschaftlern getrieben werden, um die Handhabung der von ihnen den Entwicklern implizit überlassenen Freiheitsgrade zu dokumentieren und deren Ausnutzung zu vermitteln. Die Ausbildung eines Entwicklers sollte also nicht nur auf die Funktionalität eines Tools beschränkt werden ("Was kann damit alles gemacht werden"), sondern er sollte auch mit dessen Einsatz und seinen Grenzen ("Was sollte damit alles *nicht* gemacht werden") vertraut gemacht werden.

Jürgen Ziegler, FHG-IAO, Stuttgart

Die Software-Ergonomie kann bislang keinen ausgereiften und standardisierten Warenkorb an Erkenntnissen anbieten. Es kann deshalb nicht erwartet werden, daß es **einen** Königsweg zur Integration solcher Ergebnisse in die betriebliche Praxis gibt. Angesichts des heterogenen Leistungsspektrums müssen auch die Transfermechanismen differenziert werden.

Organisatorische und qualifikatorische Maßnahmen: Organisatorische Maßnahmen sind angezeigt, um z.B. die Einführung und Durchsetzung von Firmenstandards zu unterstützen. Aus der Sicht eines benutzerorientierten Entwicklungsprozesses ist andererseits eine organisatorische Trennung in (wenige) Software-Ergonomen und (viele) Entwickler nicht unproblematisch, da sich damit schon automatisch ein Transferproblem ergibt. Wichtig ist es, die Entwickler für eine software-ergonomische Sichtweise zu sensibilisieren und zu qualifizieren, da auch im Bereich der Software-Entwicklung eine zu starke Arbeitsteilung negative Effekte mit sich bringen kann (Wirtschaftlichkeit, Motivation etc.).

Methodeneinsatz: Software-Ergonomie muß zu einem normalen Bestandteil des Software-Engineering werden. Software-ergonomische Erkenntnisse sind vielfach prozeßorientiert und lassen sich deshalb nur über die Vorgehensmethodik nutzen. Dies ist angesichts der noch weit verbreiteten Unlust gegenüber methodengeleiteter Software-Entwicklung an sich eine schwierige Zielsetzung. Umso wichtiger ist es, bei der Einführung von Methoden (z.B. CASE) software-ergonomische Komponenten direkt mit einzubauen. Leider weiß die Software-Ergonomie bislang zu wenig darüber, wie diese Komponenten aussehen sollten. Hier liegt ein großes Defizit vor.

Bausteine als Richtlinienträger: Im Zuge der Entwicklung von Industriestandards für Benutzungsschnittstellen werden auch Software-Bausteine zur Verfügung gestellt, die eine Reihe von Gestaltungsrichtlinien direkt verkörpern. Dies gilt allerdings bislang nur für recht elementare Komponenten, bei denen oft auch die software-ergonomische Qualität in Frage gestellt werden muß. Dennoch ist dieser Weg bis jetzt sehr erfolgreich und sollte in Richtung software-ergonomisch abgesicherter Grundkomponenten und anwendungsspezifischer Bausteine weitergeführt werden.

Werkzeugunterstützung: Software-Werkzeuge bilden ein hervorragendes Mittel, um standardkonformes Vorgehen zu belohnen (und Abweichungen zu bestrafen). Jedes Werkzeug ist auf bestimmte Konzepte, Bausteine und Regeln hin ausgerichtet, die sich auch in den mit dem Werkzeug gefertigten Produkten wiedererkennen lassen. Bei der Schnittstellengestaltung kann dies z.B. die Verwendung bestimmter Widget-Sätze oder Dialoggrundstrukturen sein. Deshalb sind z.B. durch User Interface Design Systeme auch bestimmte software-ergonomische Konzepte implizit (oder explizit durch eingebautes Wissen) vermittelbar.

Bausteine und Werkzeuge können den Entwickler bei der Gestaltung der low-level Details entlasten. Allerdings ergibt die Summe software-ergonomischer Komponenten noch kein benutzergerechtes System. Deshalb müssen diese Maßnahmen Kapazität freimachen für anwendungs- und aufgabenbezogene Problemlösungen.

Literatur

- Aschersleben, G. & Zang-Scheucher, B. (1989). Der Prozeß der Software-Gestaltung. Eine Bestandsaufnahme in Wissenschaft und Industrie. In S. Maaß und H. Oberquelle (Hrsg.), Software-Ergonomie 89, S. 247-253. Stuttgart: Teubner.
- Carroll, J.M. (1989). Taking Artifacts Seriously. In S. Maaß und H. Oberquelle (Hrsg.), Software-Ergonomie 89, S. 36-50. Stuttgart: Teubner.
- Grünupp, A., & Muthig, K.-P. (1990). Software-Ergonomie als prospektive Gestaltung der Funktionalität von Werkzeugen. In A. Reuter (Hrsg.), GI-20. Jahrestagung (pp. 532-542). Berlin: Springer.
- Schmidt, K.F. (1990). Arbeitswissenschaftliche Aspekte des Software-Entwicklungsprozesses. Information Management 1, S. 48-55.

Anschriften der Diskussionsteilnehmer

Dr. Udo Arend
 Dr. Edmund Eberleh
 SAP AG
 Abt. SAA-C
 Max-Planck-Str. 8
 6909 Walldorf

C. Kasten
 Deutsche Forschungsanstalt
 für Luft- und Raumfahrt
 PT-AuG
 Südstr. 125
 5300 Bonn 2

Prof. Dr. Thomas Strothotte
 FU Berlin
 Institut für Informatik
 Nestorstr. 8-9
 1000 Berlin 31

Jürgen Ziegler
 Fraunhofer Institut
 für Arbeitswirtschaft und
 Organisation
 Senefelder Str. 26
 7000 Stuttgart 1