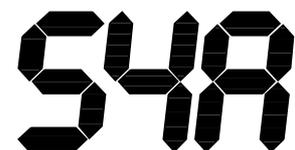


Ergonomie

Einführung in die Software Ergonomie

Datum: Nov. 2004
Version: 1.3
Autor: Markus Müller
Adresse: Solution for All
Abernstrasse 25
CH-8632 Tann
Telefon: ++41 79 437 1446
E-Mail: info@s4a.ch
Internet: <http://www.s4a.ch>



Inhaltsverzeichnis

Inhaltsverzeichnis	2
1. Einleitung.....	3
2. Die Benutzerbeteiligung bei der Software-Entwicklung	4
3. Hausinterne Richtlinien und Normen	5
4. Technisch-organisatorische Rahmenbedingungen.....	7
5. Software-Ergonomie und Qualitätssicherung	9
6. Die Wirtschaftlichkeit der Software-Ergonomie	10
7. Schlussbemerkung	11

1. Einleitung

In diesem einleitenden Kapitel soll erläutert werden, was Software-Ergonomie ist. Damit verbunden ist ein kurzer Abriss der historischen Entwicklung von Software-Ergonomie. Die Pionierzeit der Software-Ergonomie gehört heute, gut ein Jahrzehnt nachdem sie im deutschsprachigen Raum grössere Beachtung gewonnen hat, der Vergangenheit an. Aus der neuen Strömung im Wissenschaftsbetrieb, die vielerorts als Spielwiese für arbeitswissenschaftlich angehauchte Informatiker oder Psychologen galt, ist ein Arbeitsgebiet gewachsen. Vereinzelt sind die Ergebnisse oder Forderungen zwar noch umstritten, jedoch die Fragestellungen der Software-Ergonomie werden zunehmend akzeptiert.

Was ist Software-Ergonomie?

Die Software-Ergonomie beruht auf der Basis des Grundverständnisses von Ergonomie. Das heisst, der Anpassung von technischen Systemen an den Menschen und nicht umgekehrt! Die Software-Ergonomie beschäftigt sich vor allem mit dem Verhältnis von Software-Systemen und ihren Benutzern. Dieses Verhältnis (Mensch-Computer-Interaktion) spielt bei der Analyse, Gestaltung und Bewertung interaktiver Computersysteme eine wichtige Rolle. Hervorgehoben werden diejenigen Aspekte der Mensch-Computer-Interaktion, die sich aus der Gestaltung von Software ergeben. Dieser Themenbogen kann sich sehr weit spannen:

zum Beispiel:

- Arbeitsabläufe
- Fenstersysteme (Windowing Systems)
- Menü-Hierarchien
- Kommandosprachen
- Farb- und Schriftwahl (Information Design)
- Funktionsaufteilung zwischen Mensch und Computer
- etc..

Die Frage, wie man während dem Gestaltungsprozess zu ergonomischer Software gelangt, ist weithin eine Diskussion der hierzu benötigten Werkzeuge (z.B. User Interface Management Systeme, GUI-Builder etc.) oder des Prototyping.

Sollen die Erfahrungen aus der Software-Ergonomie in die Praxis umgesetzt werden, sind jedoch noch weitere Themen von grosser Wichtigkeit zu beachten:

- woher erfährt man, wie die Benutzungsschnittstelle aussehen soll?
- wann / wo / wie wird sie definiert?
- wie überprüft man ein gewähltes Design?
- was für technische und andere Rahmenbedingungen sollten beachtet werden?

Diese Fragen, werden in der vorliegenden Einführung im Mittelpunkt stehen. Bewusst ausser acht gelassen wird die Frage nach technischen Aspekten oder nach unterstützenden Werkzeugen. Ziel ist es, sowohl Notwendigkeiten und Möglichkeiten, als auch Probleme und Schwierigkeiten aufzuzeigen.

2. Die Benutzerbeteiligung bei der Software-Entwicklung



Die Notwendigkeit einer Benutzerbeteiligung ist unter anderem dadurch begründet, dass die wachsende Komplexität heutiger Anwendungen vom Software-Designer häufig nicht mehr hinreichend durchschaut und bewältigt werden kann; diese Komplexität erfordert eine enge Zusammenarbeit von Benutzern als Experten für das zu unterstützende Fachgebiet und Designern als Experten für Informatikwerkzeuge.

Die Motivation zur Mitarbeit ist bei den Benutzern, die das System bereits kennen, in der Regel hoch einzuschätzen. Dort wo das System bisher Mängel aufwies, kann jetzt durch Eigeninitiative des direkten Benutzers persönlich eine Verbesserung der Situation bewirkt werden. Zu berücksichtigen ist immer auch der Aspekt, inwieweit der Benutzer grundsätzlich in der Lage ist, das Design einer Benutzungsschnittstelle zu beurteilen bzw. eigene Vorschläge zu machen. Zieht man Benutzer bei, muss unter anderen folgender Punkt beachtet werden.

- Was der Benutzer gewohnt ist, findet er zunächst einmal eher positiv. Dies kann natürlich zur Frage führen, ob im Fall einer subjektiven Zufriedenheit überhaupt etwas an der Benutzungsschnittstelle geändert werden soll. Gleichwertige Designalternativen werden in solchen Fällen immer wieder als schlecht beurteilt.

Im einzelnen sind bei der Festlegung der Projektorganisation folgende Punkte zu berücksichtigen:

- Wer wird beteiligt? (Benutzervertreter, Programmierer, Software-Ergonomen) Welche Fachbereiche, welche Hierarchie-Ebenen werden einbezogen, wer vertritt die direkt und indirekt Betroffenen? Erst wenn diese Fragen geklärt sind, ist auch sichergestellt, dass alle benötigten Personen für einen reibungslosen Ablauf in Kenntnis gesetzt wurden.
- Wie wird die Form der Aufbauorganisation gestaltet? Mit welchen Mitarbeitern werden Projekt- und Arbeitsgruppen besetzt? Wie werden Rollen, Aufgaben und Kompetenzen verteilt? Wie wird der Informationsfluss gewährleistet?

- Wann verlangt die Anwendung eher ein zeitlich klar strukturiertes Phasenkonzept oder eher ein schrittweises, rückgekoppeltes Vorgehen, z.B. Prototyping? Zu welchem Zeitpunkt werden verstärkt Benutzerbeiträge erwartet? Erstellung eines Zeitplanes mit geplanten Aktivitäten, Meilensteinen und Prioritäten; dabei müssen genügend Zeitreserven für Unvorhergesehenes eingeplant werden.
- Was bildet den Inhalt der Software-Entwicklung? Auflistung und Systematisierung von Teilanwendungen. Zu welchen Inhalten werden die Benutzer sowie Software-Ergonomen beigezogen: Maskendesign, Dialoggestaltung, Funktionen, Datenkonzept?
- Welche Methoden und Werkzeuge werden benötigt? Welche stehen zur Verfügung, welche müssen beschafft werden?
- Speziell: Wie wird der Einbezug der Benutzer realisiert?
- Repräsentativität: Werden alle, speziell ausgewählte oder repräsentative Vertreter der Benutzer einbezogen?

Ausprägung: Beschränkt sich die Partizipation auf einen Informationsaustausch, werden die Benutzer(-vertreter) an Entscheidungen beteiligt, oder werden sie zur aktiven Mitwirkung bei der Entwicklung (z.B. des Dialogdesigns) beigezogen?

3. Hausinterne Richtlinien und Normen



Normen, Standards und Richtlinien beginnen sich im Bereich der Benutzungsschnittstellen zunehmend einen festen Platz zu erobern. Dies gilt einerseits für Standards von offiziellen Standardisierungsgremien (in Deutschland insbesondere DIN, auf internationaler Ebene ISO). Andererseits ist Windows heute im PC-Bereich schon Standard geworden, der mittels Style-Guides verankert ist. Dasselbe gilt für Motif, welches im Bereich der UNIX-basierten Mehrbenutzersysteme eine

ähnliche Stellung einnimmt.

Standards, die aufgrund von Erfahrungen mit bisherigen Entwicklungen basieren, sind heute für die Praxis sehr viel bestimmender als Richtlinien aus offiziellen Standardisierungsgremien. Letztere sind allerdings derzeit auch noch weitgehend in der Definitions- bzw. in der Abstimmungsphase.

Allgemeine Standards und Normen lassen noch einigen Spielraum für die Software-Entwicklung. In zunehmendem Masse ist man heute bestrebt, innerhalb einer Firma durch hausinterne Standards ergänzend zu wirken. Man versucht so, den Wildwuchs, der sich sonst – wie die Vergangenheit immer wieder gezeigt hat – rasch ausbreiten kann, rechtzeitig zu verhindern.

Software, die nach solchen Richtlinien erstellt ist, wird damit im Besonderen der Forderung nach einer möglichst einheitlichen Benutzungsschnittstelle gerecht. Sie stellt sicher, dass sogenannte generische Funktionen, d.h. Funktionen, die über die verschiedenen Applikationen hinweg die gleiche Semantik haben, in gleicher Weise aufgerufen werden können. Solche Funktionen sind z.B. "löschen", "speichern", "drucken", etc. Wenn diese Funktionen in unterschiedlichen Anwendungen einheitlich aufgerufen werden, **verkürzen sich Lernzeiten und es steigt die Verlässlichkeit** im Sinn des Erfüllens einer Erwartungshaltung der Benutzer. Damit sind wesentliche Forderungen der Software-Ergonomie erfüllt. Wie wichtig dies ist und wie schnell diese Eigenschaft von Benutzern erkannt und eingefordert wird, ist an den Reaktionen zu erkennen, wenn diese Einheitlichkeit verletzt wird bzw. wenn der Benutzer irrtümlich glaubt, eine bestimmte Einheitlichkeit erkannt zu haben. Solche Fälle treten in der Tat immer wieder auf und sind zum Teil auch gar nicht vermeidbar zum Beispiel bei einem aus vielen Einzelapplikationen bestehenden System der Bürokommunikation.

Soll eine Richtlinienammlung zur Vereinheitlichung der Benutzungsschnittstelle in einer Firma erstellt werden, wird dies in der Regel durch ein dafür vorgesehenes Team geschehen. Die Erfahrung zeigt, dass hierbei unbedingt auf die folgenden Punkte geachtet werden muss:

- **Die Neutralität dieses Teams muss abgesichert sein.**
Richtlinien zu schaffen, die zur Vereinheitlichung dienen, bedeutet dass in der Regel jeder Entwickler Dinge teilweise anders tun muss als er dies in der Vergangenheit gewohnt war. Es kann vorkommen, dass Entscheidungen akzeptiert werden müssen, die man vielleicht nicht für richtig hält. Dies fällt nicht zuletzt bei Fragen der Benutzungsschnittstelle manchen Software-Entwicklern schwer. Für die Akzeptanz solcher Richtlinien ist es von hoher Bedeutung, dass der Eindruck vermieden wird, es würde einfach eine persönliche Meinung festgesetzt und durchgedrückt, ohne Rücksicht auf den für andere dadurch entstehenden Zusatz- oder Änderungsaufwand.
- **In dem Team müssen entwicklungserfahrene Personen mitarbeiten.**
Es geht hier nur zum Teil darum, die sachliche Richtigkeit der Entscheidungen sicherzustellen, vielmehr um deren Akzeptanz. Nach wie vor ist gerade bei "gestandenen" Entwicklern die Ansicht verbreitet, Software-Ergonomie sei eine "soft science", deren Ergebnisse nur persönliche Vermutungen und Meinungen ohne objektiven Hintergrund seien. Was eine gute Benutzungsschnittstelle sei, würden sie als erfahrene Entwickler ohnehin besser wissen. Solchen Ansichten kann auch dadurch entgegengewirkt werden, dass erfahrene und in ihrer Kompetenz als Entwickler anerkannte Personen im Team mitarbeiten. Dadurch wird ein weiterer wichtiger Vorteil sichergestellt, nämlich dass die Möglichkeit der technisch-praktischen Umsetzung der Richtlinien immer mitbetrachtet wird.
- **Alle betroffenen Stellen müssen frühzeitig informiert und einbezogen werden.**
Diese Forderung betrifft sowohl Informationen über Ziele und Nutzen der Richtlinie als auch die ständige Rückkopplung über erreichte (Zwischen-) Ergebnisse. Letztlich

handelt es sich hierbei um das Analogon zur Benutzer-Beteiligung bei der Software-Entwicklung.

- **Die Balance zwischen Einheitlichkeit und Freiheitsgraden muss beachtet werden.**
Zwar sollen Richtlinien für Einheitlichkeit sorgen, doch muss andererseits für die Anwendung noch genügend Flexibilität möglich sein. Dies ist nötig, damit man sich ihren spezifischen Erfordernissen anpassen kann. Insofern wird man in Richtlinien zur Benutzungsoberfläche kaum um "Kann"- und "Soll"-Bestimmungen herumkommen. Dabei sollte aber darauf geachtet werden, so restriktiv zu sein, dass gerechtfertigte Ausnahmen als Teil des Qualitäts-Sicherungsprozesses speziell diskutiert und entschieden werden können. Es kann allerdings auch vorkommen, dass Entwickler von solchen Richtlinien erwarten, dass ihnen alle Design-Entscheidungen abgenommen werden. Dies ist natürlich nicht möglich, denn das hiesse, über die Richtlinien vorab alle speziellen Benutzungsschnittstellen zu spezifizieren. Die Verantwortung für deren Definition bleibt beim Entwickler, die Richtlinien können nur in Teilen Vorgaben machen.
- **Die Machtpromotoren müssen hinter der Aufgabe stehen.**
Wie bereits angedeutet, ist die Frage der Akzeptanz von Richtlinien zur Benutzungsschnittstelle von grosser Bedeutung. Die Erfahrung in vielen Unternehmen zeigt, dass hier oft grössere Schwierigkeiten zu überwinden sind. Wird explizit die Unterstützung durch die Machtpromotoren gefordert um die Durchsetzungsfähigkeit sicherzustellen, so heisst das nicht, dass damit der Software-Ergonomie über den Anweisungsweg Rückhalt verschafft werden kann. Es gilt vielmehr auch sicherzustellen, dass die nötigen Ressourcen sowohl bei der Richtliniendefinition als auch später in konkreten Gestaltungsprozessen zur Verfügung stehen. Software-Ergonomie ist nicht zum Nulltarif zu haben.

Die fachliche Güte von Richtlinien zur Software-Ergonomie ist zwar eine notwendige, keineswegs aber eine hinreichende Voraussetzung, um die Gestaltung von Benutzungsschnittstelle entsprechend zu beeinflussen. Genauso wichtig ist das Prozedere der Gestaltung.

4. Technisch-organisatorische Rahmenbedingungen

Die Entwicklung von Benutzungsschnittstellen vollzieht sich im Normalfall **nicht in einem luft- bzw. konfliktfreien Raum**. Sie steht in der Regel im Wettbewerb mit anderen Zielen und Rahmenbedingungen.

Es wurde vorher bereits angemerkt, dass Software-Ergonomie nicht zum Nulltarif zu haben ist. Zeit und Kosten sind heute mehr denn je von eminenter Bedeutung. Speziell die Informationstechnik ist von einem raschen Preisverfall und einem extremen Innovationsdruck und -tempo gekennzeichnet. Im Hardwarebereich liegt die Halbwertszeit eines Produktes

heute nur noch bei ca. 6 Monaten, im Fall vieler Komponenten liegt sie sogar noch darunter. Kostensenkung und die Verkürzung des **"time-to-market"** sind entscheidende Faktoren im Fertigungs- und Entwicklungswettbewerb geworden.

Dies darf natürlich nicht eine Entschuldigung dafür sein, nicht in die software-ergonomische Qualität eines Produktes investieren zu wollen. Aber es wird im Einzelfall immer klar zu überlegen und zu begründen sein, warum der geforderte Aufwand zu treiben ist. Dadurch wird ersichtlich, ob Wettbewerbsfähigkeit oder sogar Wettbewerbsvorteile erzielbar sind. Der Entwickler ist gleichzeitig verpflichtet, sorgfältig darauf zu achten, welcher Aufwand auch zu welchem Mehr an Qualität führen kann. Auch Softwareprodukte haben heute (v.a. bei Herstellern, System- und Softwarehäusern) in der Regel nur ein kleines Zeitfenster, um ihren Erfolg sicherzustellen. Dieses Zeitfenster lässt sich nicht ohne weiteres verschieben. Die Produkte-Entwicklung, einschliesslich der Entwicklung der Benutzungsschnittstelle muss sich an ihm orientieren.

Mit Software-Ergonomie, die zu spät kommt, ist niemandem gedient.

In diesen Komplex gehört auch die Frage nach Funktionalität versus Design bei gegebenen Entwicklungsressourcen. Bei allen, am Entwicklungsprozess Beteiligten muss Kompromissfähigkeit bezüglich der Benutzungsschnittstelle vorhanden sein. Dies darf jedoch nicht mit grundsätzlichem Nachgeben verwechselt werden. Es gibt meistens nicht eine einzige gute Lösung, in der Regel gibt es eine ganze Bandbreite von qualitativ ausreichenden oder sogar guten Möglichkeiten. Wichtig ist es, mit der gewählten Lösung innerhalb dieser Bandbreite zu bleiben. Man darf nie vergessen, wie wichtig den Anwendern (Endbenutzer eines Systems), die Funktionalität ist. Solange die Schaffung einer möglichst grossen Funktionalität und die Realisierung einer ergonomischen Benutzungsschnittstelle im Wettbewerb um knappe Ressourcen stehen, wird es Ziel sein müssen, für die Benutzern eine insgesamt zufriedenstellende Lösung zu finden.

Auch die technischen Rahmenbedingungen können den Wünschen Grenzen setzen. Die zu verwendende Hardware bzw. die Leistungsfähigkeit der vorhandenen Plattformen engen die möglichen Realisierungen noch immer häufig ein. Obwohl die technische Entwicklung heute schon ganz andere Voraussetzungen bietet als noch vor einigen Jahren, wie zum Beispiel die Entwicklung graphisch orientierter Benutzungsschnittstellen.

Die Ernsthaftigkeit und Glaubwürdigkeit der Software-Ergonomie muss sich jedoch auch darin zeigen, wie sie mit bestehenden Welten zurechtkommt. Auch mit schon vorhandenen Applikationen muss sie es schaffen, akzeptable oder sogar gute Lösungen bereitzustellen. Software-ergonomische Erkenntnisse und Ideen müssen auf ihre zugrundeliegenden Prinzipien untersucht werden.

Es darf keine technische Lösung als die einzig richtige oder mögliche festgeschrieben werden. Hierzu ein Beispiel aus der Alltagsarbeit, um dies zu verdeutlichen: Das Prinzip des

"Elektronischen Schreibtisches", wie wir es von vielen Systemen mit graphischen Benutzungsschnittstellen kennen (z.B. Apple-Macintosh), kann weitgehend auch in einem zeichenorientierten Umfeld realisiert werden. Die grosse Herausforderung an Entwickler ist in solchen Fällen, das als richtig erkannte Prinzip so umzusetzen, dass sich die Benutzungsschnittstelle auch bei einem durch neue Technik ermöglichten Wechsel (etwa von der Tastatur zur Maus) dem Benutzer gegenüber weitestgehend identisch darstellt.

5. Software-Ergonomie und Qualitätssicherung



Solange der Aspekt einer ergonomischen Gestaltung der Benutzungsschnittstelle allzu häufig noch stiefmütterlich behandelt wird, muss dessen Umsetzung über formale Wege und Methoden durchgesetzt werden. Dasselbe gilt für die Bereitschaft, vorhandene Richtlinien und Kenntnisse konsequent in der eigenen Arbeit umzusetzen.

Es darf insbesondere nicht zugelassen werden, dass die Frage der Benutzungsschnittstelle an das Ende des Entwicklungsprozesses verschoben wird. Die Gefahr, dass zu diesem Zeitpunkt unter starkem terminlichem Druck entwickelt werden muss, ist sehr hoch. Dann wird die Funktionalität bzw. Qualität im Sinn von Stabilität und Fehlerfreiheit prioritär behandelt. Die Benutzungsschnittstelle wird dann nicht mehr mit der erforderlichen Aufmerksamkeit realisierbar.

Die **Definition der Benutzungsschnittstelle muss zu einem möglichst frühen Zeitpunkt** im Projektablauf und in einer grossen Detailtiefe **überlegt und spezifiziert werden**. Am sinnvollsten geschieht dies bei der Erstellung des Systementwurfs (ggf. bei der Spezifikation des ersten Protoyps). Es kann unter Umständen auch bereits bei der Erstellung des Pflichtenhefts sinnvoll sein. Darin sollten zumindest die Anforderungen an die Benutzungsschnittstelle, soweit sie sich aus den intendierten Benutzergruppen etc. ableiten lassen, aufgeführt sein.

Idealerweise sind zu diesem Zeitpunkt auch schon Skizzen des Dialog Designs und des Screen-Layouts so detailliert als möglich anzufertigen. Es ist klar, dass alle solchermaßen früh getroffenen Designentscheidungen möglicherweise später noch revidiert werden müssen. Wichtig ist jedoch, dass bei der Realisierung bereits eine klare und vollständige Vorgabe als Orientierung vorliegt. Zudem verfügt man heute in zunehmendem Masse über Werkzeuge in der Software-Entwicklung, die ein vergleichsweise rasches Ändern der Benutzungsschnittstelle erlauben. Dies ist umso erfreulicher, da die Erfahrung folgendes klarstellte: erst bei einer realitätsnahen Anwendung **implementierter Software** kann die ergonomische Güte der Benutzungsschnittstelle schlüssig und unzweifelhaft beurteilt werden. Definitionen auf Papier, so gut diese als schnelles und kreativitätsförderndes Mittel sein mögen, genügen hierfür nicht mehr. Dem Prinzip des **"Rapid Prototyping"** mit der technischen Entwicklungsumgebung kommt somit eine hohe Bedeutung zu, wenn ergonomische Güte und kostengünstige Entwicklung gleichermaßen erreicht werden sollen.

Durch die frühe Spezifikation der Benutzungsschnittstellen ist auch gesichert, dass ihre Prüfung bereits zu einem frühen Zeitpunkt in den regulären Prozess der Qualitätssicherung einfließen kann. Auch hier gilt, wie eigentlich generell, dass Qualität als ein Produkt nicht hineingeprüft werden kann, sondern dass die Schaffung von Qualität und die zugehörige Prüfung und Absicherung die Entwicklung von Beginn an parallel begleiten müssen.

6. Die Wirtschaftlichkeit der Software-Ergonomie



Der wirtschaftliche Nutzen der Software-Ergonomie ist auf den ersten Blick nur schwer fassbar. Es ist eher von einer indirekten Wirtschaftlichkeit auszugehen, im Sinne der Vermeidung oder Verringerung sogenannter Opportunitätskosten. Das heisst, es sind beispielsweise die Folgen für den Fall abzuschätzen, dass ein System nicht benutzungsgerecht gestaltet ist.

Bury (1984) beschreibt die erfolgreiche Entwicklung eines Programms, bei der zuerst nur ein Prototyp der Benutzungsschnittstelle erstellt und mit späteren Benutzern getestet wurde. In mehreren Durchgängen wurde der Prototyp weiter verbessert. In jeder Stufe wurde gemessen, wieviel Prozent der Benutzer die verschiedenen Testaufgaben ohne grössere Hilfe lösen konnten. Die Werte steigerten sich für jede Aufgabe im Laufe der Verfeinerungen um ca. 25%. Erst im Anschluss daran wurde die zugehörige Anwendung für die untersuchte Benutzungsschnittstelle entwickelt.

Nicht nur die Anwender ziehen ökonomischen Nutzen aus der Software-Ergonomie. Die ganzheitliche Ausrichtung zeigt auch positive Effekte im Software-Entwicklungs-Prozess. Zwar werden intensivere Vorarbeiten in früheren Projektphasen erforderlich, diese Anstrengungen zahlen sich allerdings beim Einsatz software-ergonomisch orientierter Prototyping-Werkzeuge für Design und Implementierung rasch durch Effizienz- und Qualitätssteigerung aus.

Mehr als 80% der Programmierkapazität wird heute durch die Pflege von Programmen beansprucht. Das entspricht dem zwei- bis dreifachen der ursprünglichen Erstellungskosten. Häufig entfallen bis zu 50% des Programmieraufwands auf die Gestaltung der Benutzungsschnittstellen. Eine drohende Kostenexplosion dürfte durch den Einsatz software-ergonomisch orientierter Entwicklungsmethoden und entsprechender Software-Werkzeuge abgeschwächt werden. Dies gilt sowohl bei Migrationsproblemen, als auch bei Neuentwicklungen. Die anfallenden Kosten amortisieren sich um so leichter, je benutzergerechter die entwickelte Software ausfällt und je stärker sich der Entwicklungsaufwand reduzieren lässt. Dies setzt jedoch voraus, dass die Software auch optimal eingesetzt wird.

Effizienzsteigerungen können in Zukunft nicht nur durch neue Techniken sondern vor allem bei der Optimierung der Mensch-Computer-Interaktion erwartet werden. Diese Techniken müssten dem wachsenden Bedarf nach Einbindung verschiedener Medien wie Text, Graphik, Animation, Audio/Video und natürliche Sprache gerecht werden. Wirtschaftliche und benutzungsgerechte Lösungen für die neu entstehenden Interaktionsprobleme werden vor allem von der Software-Ergonomie zu erwarten sein.

7. Schlussbemerkung

Wie in der Einleitung bereits erwähnt, war Gegenstand dieser Kurzbeschreibung nicht das, was die Ergonomie an Eigenschaften von der Software fordert und in welchen Gestaltungsmerkmalen sich dies widerspiegelt. Sie sollte vielmehr einen kurzen Abriss zur Umsetzung und ihren praktischen Schwierigkeiten liefern.



Beides zusammen müssen Fragen der Software-Ergonomie sein. Wir brauchen ihre Qualitätsmerkmale als Ziel. Doch dürfen wir nicht vergessen, ihr auch den Weg dahin zu ebnen.