**Rauterberg, M., Spinas, P., Strohm, O., Ulich, E. & Waeber, D. (1994).
Benutzerorientierte Software-Entwicklung Konzepte, Methoden und Vorgehen zur Benutzerbeteiligung.
(Mensch - Technik - Organisation. Band 3), vdf Hochschulverlag AG an der ETH Zurich.**

Matthias Rauterberg, Philippe Spinas,
Oliver Strohm, Eberhard Ulich und Daniel Waeber

BENUTZERORIENTIERTE SOFTWARE-ENTWICKLUNG

Konzepte, Methoden und Vorgehen zur Benutzerbeteiligung

Inhaltsverzeichnis

Vorwort

Bei Projekten zur Computerunterstützung von Bürotätigkeiten stellt sich die Grundfrage: *Wie* können *Benutzerbedürfnisse bzw. Anforderungen* von Fachabteilungen nach Computerunterstützung von Arbeitsabläufen so *in Software umgesetzt* werden, dass diese den Anforderungen auch wirklich entspricht und dadurch eine *echte Hilfe* bei der Bewältigung der anfallenden Aufgaben darstellt? Allzu oft sind schon Entwicklungsprojekte gescheitert, oder die entwickelten Lösungen haben den Benutzern nicht die erwartete Unterstützung gebracht, weil ihre Bedürfnisse und ihre Aufgaben im Prozess der Softwareentwicklung zu wenig berücksichtigt wurden. Warum? Viele Beispiele zeigen, dass es für den Softwareentwickler sehr schwierig ist, sich in die Lage eines zukünftigen Systembenutzers zu versetzen und dessen Denkvorgänge sowie Handlungen im Arbeitsprozess zu antizipieren. Dies gilt je länger je mehr, denn die Komplexität der Anwendungen steigt immer schneller an, und für den Softwareentwickler wird es zunehmend schwieriger oder gar unmöglich, den fachlichen Hintergrund einer Anwendung zu durchschauen und zu verstehen. Diese Komplexität erfordert eine enge *Zusammenarbeit zwischen Benutzern als Experten für das zu unterstützende Fachgebiet und Entwicklern als Experten für Informatikwerkzeuge*, um optimale, auf die konkrete Situation angepasste Lösungen zu finden. Über die Notwendigkeit einer Beteiligung von Benutzern bei der Softwareentwicklung besteht heute bei Fachleuten verschiedenster Disziplinen weitgehend Einigkeit. Probleme bereitet allerdings häufig die praktische Realisierung.

Das vorliegende Buch ist das Ergebnis eines vierjährigen Projektes zur benutzerorientierten Softwareentwicklung, das vom Bundesministerium für Forschung und Technologie (Projektträger 'Arbeit und Technik, Bonn; Förderkennzeichen 01HK706) finanziell gefördert wurde. Dabei wurden die Entwicklungsprozesse in über 100 Projekten analysiert, Methoden und Vorgehensweisen zur Beteiligung von (End-)Benutzern entwickelt und erprobt sowie Benutzungstests zur Messung der Gebrauchstauglichkeit von interaktiven EDV-Systemen durchgeführt. Die Ergebnisse dieser Analysen, Feldversuche und Benutzungstests wurden aufgearbeitet und in diesem Buch dargestellt. Damit soll Softwareentwicklern (Projektleiter, Analytiker, Programmierer), Benutzern, Organisatoren, Ergonomiebeauftragten, Koordinatoren zwischen EDV- und Fachabteilung und Ausbildnern eine Unter-

stützung bei der Entwicklung benutzerorientierter EDV-Systeme angeboten werden.

Obwohl bereits vor mehr als zehn Jahren eine Reihe von – zum Teil heute noch aktuellen Büchern – zum Thema der Benutzerbeteiligung erschienen sind, erscheint es sinnvoll, dieses Buch zu veröffentlichen. Die drei folgenden Gründe haben uns dazu bewogen:

(1) Es wurden in den letzten Jahren weltweit viele neue Methoden, Techniken und Formen für Benutzerbeteiligungen entwickelt und erprobt.

(2) Dieses Buch zeichnet sich durch eine integrative Sichtweise aus – die Integration von arbeitspsychologischen und softwaretechnischen Aspekten.

(3) Da die meisten Bücher über moderne Methoden englischsprachig sind, erschliesst dieses deutschsprachige Buch den Zugang zu dieser Literatur und gibt eine erste Orientierung.

Neben der Vermittlung (arbeits-)psychologischer Grundlagen werden Konzepte der Benutzerbeteiligung vorgestellt, Hinweise und Ideen für die praktische Realisierung gegeben sowie mögliche Probleme und Lösungsansätze aufgezeigt. Fallbeispiele, Checklisten und Übersichten unterstützen die Umsetzung der Anregungen in das Vorgehen bei konkreten Projekten. Wenn auch noch nicht alle Probleme im Detail gelöst sein mögen und bereits vorhandene Lösungsmöglichkeiten aufgrund des begrenzten Seitenumfangs zum Teil nicht weiter ausgeführt werden konnten, so gibt dieses Buch dennoch eine Fülle an Anregungen und Hinweisen auf weiterführende Literatur.

An dieser Stelle sei allen Beschäftigten in den Betrieben gedankt, durch deren Kooperationsbereitschaft im Laufe der letzten vier Jahre viele wertvolle Erkenntnisse und Erfahrungen in dieses Buch eingeflossen sind. Insbesondere gilt unser Dank dem Geschäftsführer Herrn Dr. Karl Schlagenhauf und dem Produktmanager Herrn Raimund Mollenhauer der ADI Software GmbH in Karlsruhe für die fruchtbare Zusammenarbeit.

Zürich, im Juli 1994

Matthias Rauterberg, Philipp Spinas, Oliver Strohm, Eberhard Ulich und Daniel Waeber

Lesehinweise

Für dieses Buch gibt es verschiedene Leseweisen, welche sich nach den unterschiedlichen Interessen der Leser ausrichten. Wir bieten daher die folgenden Leseweisen an:

Als Nachschlagewerk – Zuerst im Stichwort- oder Inhaltsverzeichnis die relevanten Verweise heraussuchen und dann an den entsprechenden Textstellen weiterlesen.

Falls bei den folgenden Lesehinweisen einige Begriffe an der jeweiligen Stelle nicht ausreichend erklärt sein sollten, so kann dies daran liegen, dass diese Begriffe im Grundlagenteil A eingeführt wurden.

Hinweise auf Methoden – Im Teil B ist im Kapitel 4 eine weitgehend vollständige Auflistung und Darstellung der verschiedenen, zur Zeit bekannten Methoden zur Benutzerbeteiligung gegeben. Zur Vertiefung bei der einen oder anderen Methode kann es unumgänglich sein, auf die weiterführende Literatur zurückzugreifen.

Fragen zur Projektleitung und zum Projektaufbau – Im Teil B sind in Kapitel 2 verschiedene Ansätze zur Projektorganisation mit Benutzerbeteiligung dargestellt. Falls einige Begriffe nicht ausreichend erklärt sein sollten, so kann dies daran liegen, dass diese Begriffe im Grundlagenteil A eingeführt wurden.

Fragen an den Projektablauf und die Prozessgestaltung – Im Teil B ist in Kapitel 3 ein iterativ-zyklisches Ablaufmodell vorgestellt, welches die Zuordnung der Methoden zum Benutzereinbezug an den verschiedenen Stellen im Entwicklungsprozess vorsieht.

Für den Einsatz in der Lehre – Anhand der dargestellten Fälle im Teil C lassen sich verschiedene Aspekte für Benutzerbeteiligung herausarbeiten. Je nach Fragestellung werden dann entsprechende Vertiefungen in den Teilen A und B bzw. der weiterführenden Literatur notwendig.

Einleitung

Mit der fortschreitenden Automatisierung von Arbeitsprozessen – man denke etwa an Schlagworte wie Bürokommunikation, Büroautomation oder CIM, CAD, CAM im Produktionsbereich – werden verschiedene Ziele verfolgt. Beispielhaft seien hier die Steigerung von Effizienz und Produktivität, die Verbesserung von Dienstleistungen, die Erhöhung von Wirtschaftlichkeit und Konkurrenzfähigkeit, die Verkürzung der Durchlaufzeiten von Werkstücken und Dokumenten, die Verbesserung von Arbeitsbedingungen, die Erhöhung der Auskunftsbereitschaft durch Integration von Informationen und verbesserten Zugriffsmöglichkeiten usw. genannt. Diese Ziele werden allerdings von den unterschiedlichen Interessengruppen im Betrieb – z.B. Management, Systementwickler, Organisatoren und Benutzer – vielfach unterschiedlich gewichtet, was zu Interessenkonflikten führen kann. Immer häufiger stellt sich deshalb bei Vorhaben zur Computerunterstützung von Arbeitstätigkeiten die folgende Grundfrage: Wie können Anforderungen von Fachabteilungen bzw. Bedürfnisse von Benutzern nach Computerunterstützung von Arbeitsabläufen so in Software umgesetzt werden, dass diese den Anforderungen auch wirklich entspricht und dadurch eine echte Hilfe bei der Bewältigung der anfallenden Aufgaben darstellt?

In den letzten Jahren liessen sich in vielen Softwareentwicklungsprojekten grosse Probleme feststellen, die manchmal zum Scheitern eines Projektes oder gar zum Konkurs einer Firma führten. Die Ursachen können vielfältiger Natur sein, denn für die Akzeptanz eines neuen Systems durch die Benutzer spielt die optimale Gestaltung folgender, unterschiedlicher Aspekte eine Rolle:

(1) Arbeitsteilung und -organisation,

(2) Mensch-Computer-Funktionsteilung,

(3) Benutzungsoberfläche (Softwareergonomie),

(4) Ergonomie von Hardware sowie Arbeitsplatz und -umgebung,

(5) Entwicklungs- und Einführungsprozess, Benutzerbeteiligung.

In diesem Buch befassen wir uns *zur Hauptsache* mit den Möglichkeiten der Benutzerbeteiligung im Prozess der Softwareentwicklung (siehe auch Mambrey und Oppermann 1983, Mumford und Welter 1984, Mambrey,

Oppermann und Tepper 1986, Peschke 1986, Kissler 1988, Koslowski 1988 und Ruch, Spinas und Ulich 1989). In bezug auf die anderen Aspekte verweisen wir auf die entsprechende Fachliteratur. Von besonderer Bedeutung ist die Gestaltung der Arbeitsteilung und -organisation; sie hat vor bzw. zumindest gleichzeitig mit dem Einkauf oder der Entwicklung technischer Systeme zu erfolgen!

Bei den erwähnten Problemen handelte es sich aber meist weniger um technische Schwierigkeiten. Vielmehr ist es die fachliche Komplexität und Dynamik heutiger Anwendungen, welche die Softwareentwickler vor Probleme stellt. Ein EDV-Spezialist ist heutzutage häufig nicht mehr in der Lage, die fachliche Seite einer Anwendung hinreichend zu durchschauen und zu bewältigen. Deshalb ist eine enge Zusammenarbeit von Benutzern als Experten für das zu unterstützende Fachgebiet und Softwareentwicklern als Experten für Informatikwerkzeuge unabdingbar. In manchen Fällen kann auch zusätzlich ein 'neutraler' Prozessmoderator notwendig sein. Im Rahmen partizipativer Softwareentwicklung sind verschiedene Komponenten und Aspekte zu unterscheiden, die in diesem Buch ausführlich dargestellt werden: Rahmenbedingungen, Projektmanagement, Aufbau- und Ablauforganisation, Information, Ausbildung, Partizipation sowie verschiedene Methoden zur Unterstützung von Benutzerbeteiligung.

Entgegen häufig vorgebrachten Argumenten – wie z.B. die Benutzer wollen oder können gar nicht an einer Entwicklung beteiligt werden – ist die Bereitschaft zur Zusammenarbeit von Entwicklern und Benutzern nach den Ergebnissen unserer Untersuchungen viel höher als gemeinhin angenommen. Ein Projektleiter formulierte es – stellvertretend für viele andere – wie folgt: "Ohne die aktive Mitarbeit der Benutzer könnten wir heute keine Anwendung mehr realisieren. Der *erforderliche Zeitaufwand* ist im Verhältnis zu den *erzielten Vorteilen gering*."

Mit dem Einbezug von Benutzern in den Entwicklungsprozess werden primär folgende Ziele verfolgt:

Innovation Es werden innovativere Lösungen entwickelt, in welche neben dem technischen Wissen der Entwickler auch das Fachwissen der Benutzer einfliesst.

Identifikation Die Entwickler wie auch die Benutzer können sich mit der Entwicklungsarbeit sowie auch mit der Lösung besser identifizieren (Akzeptanz!)

Zur Erreichung dieser Ziele müssen folgende Grundbedingungen erfüllt sein:

Integration Aufgabenspezifische, benutzerspezifische und technische Anforderungen müssen *integrativ* analysiert und in der Umsetzung berücksichtigt werden.

Interaktion Entwickler und Benutzer müssen die Möglichkeit haben, direkt miteinander zu interagieren, damit sich gegenseitiges Verständnis entwickeln kann.

Iteration Der Entwicklungsprozess muss so angelegt sein, dass innerhalb einzelner Phasen wie auch über Phasen hinweg iteriert werden kann. Nur so können Veränderungen in den Anforderungen aufgefangen, getroffene Entscheidungen überprüft, Gestaltungsvorschläge, Konzepte evaluiert und gegebenenfalls frühzeitig korrigiert werden.

Inzwischen haben auch namhafte Hersteller sowie Anwender mit eigener Entwicklungsabteilung die Vorteile des Einbezugs von Endbenutzern erkannt. So wirbt z.B. ein Grossproduzent von Standardsoftware in einem doppelseitigen Inserat mit dem Argument der Benutzerbeteiligung für sein Produkt: "An ... haben 300 Leute mitgearbeitet, die keine Ahnung vom Computer haben. ... Das Usability Lab, ein Speziallabor, in dem wir alle unsere neuen Programme sowohl von PC-Freaks als auch von blutigen Laien testen lassen ... Dadurch konnten wir aus jedem Fehler lernen und schwierige Aufgaben ... für jedermann verständlich machen." Ferner ist uns bekannt, dass zwei Schweizer Banken die Beteiligung von Benutzern bezüglich Funktion, Kompetenzen und Verantwortung in Projekthandbüchern verbindlich beschreiben und ausserdem ein Usability Lab bzw. eine Fachstelle für Softwareergonomie eingerichtet haben. In ähnlicher Weise wird der Benutzereinbezug auch bei verschiedenen Unternehmen (z.B. Versicherungen) in Projekthandbüchern verbindlich festgelegt. Diese Beispiele zeigen, dass Benutzerbeteiligung kein theoretisches Konzept mehr, sondern inzwischen praktische Realität ist!

Dieses Buch ist zur besseren Orientierung in drei Teile gegliedert: In *Teil A* werden die wichtigsten *Grundlagen* für aufgaben- und benutzerorientierte Softwareentwicklung erläutert. *Teil B* beschreibt das *praktische Vorgehen*. In *Teil C* sind dann *konkrete Fallbeispiele* beschrieben. Die Fallbeispiele dienen der Veranschaulichung und können auch in der Ausbildung Einsatz finden. Im *Anhang* finden sich *Checklisten* und *Übersichten.* Die Checklisten erlauben direkten Zugriff bei Fragen und Problemen der Alltagsarbeit.

An dieser Stelle scheint uns auch der Hinweis wichtig, was dieses Buch *nicht* beabsichtigt: Es wird *kein* Grundlagenwissen hinsichtlich Organisationsentwicklung und Arbeitsstrukturierung, Softwareentwicklung und Betriebswirtschaftslehre vermittelt. Dieses Buch ersetzt somit kein anderes Buch zu den genannten Bereichen. Das Wissen über die anderen Bereiche wird zum Teil vorausgesetzt bzw. muss ergänzend zur Kenntnis genommen werden. Entsprechende Literaturhinweise finden sich an den jeweiligen Textstellen. Um Missverständnissen vorzubeugen, seien hier einige grundlegende Begriffe kurz erläutert:

(1) Unter Partizipation bzw. Beteiligung oder Einbezug von Benutzern ist deren Mitwirkung an der Planung und Entwicklung eines Systems und die Einflussnahme auf Entscheidungen zu verstehen.

(2) Als Benutzer oder Endbenutzer werden diejenigen Personen (Sachbearbeiter, Sekretärin, Manager usw.) bezeichnet, die das System nach abgeschlossener Entwicklung im Rahmen ihrer täglichen Arbeit benutzen. Im Gegensatz dazu ist mit Käufer der Betreiber oder Anwender gemeint (Organisation, Abteilung oder deren Repräsentanten).

(3) Bezeichnungen wie Entwickler, Designer, Programmierer werden in umfassendem Sinne synonym verwendet für Personen, die EDV-seitig an der Konzeption und Realisierung eines Systems beteiligt sind. In einzelnen Fällen wird eine bestimmte Gruppe – z.B. Projektleiter, Analytiker – besonders hervorgehoben.

(4) In diesem Buch wird fast ausschliesslich die männliche Form (Benutzer, Entwickler usw.) verwendet. Wir gehen aber davon aus, dass beide Geschlechter gemeint sein können (BenutzerIn, EntwicklerIn usw.).

Abschliessend sei noch auf folgende wichtige Unterscheidung hingewiesen: Wir haben es bei Softwareentwicklungsprojekten sowohl mit der Aufbau- und Ablauforganisation eines Unternehmens bzw. der betroffenen Fachabteilungen zu tun als auch mit der Aufbau- und Ablauforganisation des Softwareentwicklungsprojektes selbst. Diese Organisationsformen können unabhängig voneinander sein, aber auch weitgehend identisch. Wir haben diesem Umstand Rechnung getragen, indem wir die arbeitspsychologischen Kriterien für humane Arbeit, welche primär auf die Aufbau- und Ablauforganisation eines Unternehmens ausgerichtet sind, auch auf die Aufbau- und Ablauforganisation der Softwareentwicklung selbst übertragen und entsprechend angepasst haben.

TEIL A: GRUNDLAGEN

1 Arbeitspsychologische Grundlagen

Damit Arbeitstätigkeiten und Arbeitsstrukturen menschengerecht gestaltet werden können, müssen bestimmte Zielvorgaben für die Bewertung bekannt sein. Die Arbeitspsychologie bewertet Arbeit dann als menschengerecht, wenn sie die folgenden vier Kriterien erfüllt (Ulich 1994):

(1) Schädigungsfreiheit,

(2) Beeinträchtigungslosigkeit,

(3) Persönlichkeitsförderlichkeit,

(4) Zumutbarkeit.

Das Kriterium der Schädigungsfreiheit meint, dass Arbeit keine physischen und psychophysischen *Schädigungen* beim Arbeitenden hervorrufen darf. Schädigungen zeichnen sich vor allem dadurch aus, dass sie

(1) objektiv feststellbar sind,

(2) in den normalen Erholungszeiten nicht zurückgebildet werden können und

(3) meistens der Behandlung bedürfen.

Schädigungen durch Arbeit am Computer können beispielsweise in Form degenerativ-rheumatischer Erkrankungen infolge ausschliesslich sitzender Tätigkeit auf ergonomisch schlechten Stühlen bei ganztägiger, intensiver Bildschirmarbeit auftreten.

Im Gegensatz zu den Schädigungen sind *Beeinträchtigungen* des psychosozialen Wohlbefindens weniger gut objektivierbar und können kaum eindeutig auf bestimmte Arbeitsbedingungen zurückgeführt werden. Als psychosoziale Beeinträchtigungen bei Bildschirmarbeit gelten zum Beispiel depressive Verstimmungen als Folge sozialer Isolation am Arbeitsplatz (Einzelarbeit) oder Einschränkungen im Sozialleben bei langjähriger Schichtarbeit in der EDV-Überwachung. Es muss beachtet werden, dass aus Beeinträchtigungen, die über längere Zeit bestehen bleiben, psychosomatische Schädigungen entstehen können.

Aus den genannten Beispielen zu möglichen Schädigungen und Beeinträchtigungen bei der Bildschirmarbeit geht bereits hervor, dass hier nicht die Technik als solche verantwortlich gemacht werden kann, sondern dass deren Einsatzform – ganztägige Bildschirmarbeit, isolierter Arbeitsplatz – bestimmend wirkt. Es ist daher sicherlich eine Aufgabe der Softwareentwicklung, solche Einsatzformen durch entsprechende Gestaltungsmassnahmen in der Software zu verhindern oder wenigstens nicht zwingend festzuschreiben.

Das Kriterium der *Persönlichkeitsförderlichkeit* geht davon aus, dass sich die Persönlichkeitsentwicklung des erwachsenen Menschen vor allem in der Auseinandersetzung mit seiner Arbeitstätigkeit vollzieht. Obwohl nicht abschliessend beantwortet werden kann, welche Persönlichkeitsdimensionen in der Arbeit besonders gefördert werden können, weisen bisherige Untersuchungen klar darauf hin, dass die Ausprägung

(1) der kognitiven Kompetenz,

(2) der sozialen Kompetenz,

(3) des Selbstkonzepts und

(4) der Leistungsmotivation in direktem Zusammenhang mit Merkmalen der Arbeit zu sehen sind.

Als wichtigste Merkmale der Arbeit wirken sich dabei vor allem die Arbeitsinhalte und die Anforderungen, die an die Qualifikationen der Arbeitenden gestellt werden, auf die Persönlichkeitsentwicklung aus. So erhalten

und steigern beispielsweise abwechslungsreiche, herausfordernde Aufgaben die geistige Beweglichkeit.

Zentrale Voraussetzung für die persönlichkeitsförderliche Wirkung von Arbeitstätigkeiten ist deren vollständige Struktur. *Vollständigkeit* ist nach Ulich (1994) gegeben, wenn

(1) Ziele selbständig gesetzt und in übergeordnete Ziele eingebettet werden können,

(2) selbständige Handlungsvorbereitung im Sinne der Wahrnehmung von Planungsfunktionen möglich ist,

(3) die Mittel einschliesslich der erforderlichen Interaktionen zur adäquaten Zielerreichung ausgewählt werden können,

(4) während der Ausführung Rückmeldungen möglich sind, damit allenfalls korrigiert werden kann und

(5) das Ergebnis auf Übereinstimmung mit den gesetzten Zielen überprüft werden kann.

Die Vollständigkeit der Arbeitstätigkeit ist eng verknüpft mit der Arbeitsaufgabe. Sie bildet deshalb den Ansatzpunkt für die Gestaltung menschengerechter Arbeit.

Die *Zumutbarkeit* ist ein gruppenspezifisches Kriterium, welches stark von gesellschaftlichen Normen und Werten bestimmt ist. Ob eine Arbeit für eine bestimmte Person zumutbar ist oder nicht, hängt in erster Linie von den Wertvorstellungen, Bedürfnissen und Ansprüchen der direkt Betroffenen ab. In diesem Sinne kann es auf der Ebene jedes der hier genannten Kriterien angewendet werden.

1.1 SOZIOTECHNISCHES SYSTEMKONZEPT

In der Praxis ist oft zu beobachten, dass die Entwicklung computerunterstützter Arbeitssysteme 'von innen nach aussen' erfolgt. Das heisst: Man geht von der Hardware aus, entwirft Software dafür, setzt als letztes noch eine Benutzungsoberfläche darauf und fragt sich ganz am Schluss, wozu ein solches System benutzt werden kann. Ein derartiges Vorgehen verlangt nicht nur vielfältige nachträgliche Korrekturen, sondern führt möglicherweise auch zu Veränderungen von Tätigkeits- und Organisationsstrukturen, die ursprünglich gar nicht beabsichtigt waren. Wie für die industrielle Fertigung gilt demzufolge auch für computerunterstützte Büroarbeit, dass die

Systeme 'von aussen nach innen' entwickelt werden sollten. Das heisst: Zunächst muss das soziotechnische System, in das das Computersystem eingebettet werden soll, analysiert bzw. konzipiert werden, erst daran anschliessend wird die geeignete Hard- und Software beschafft bzw. angepasst oder neu entwickelt.

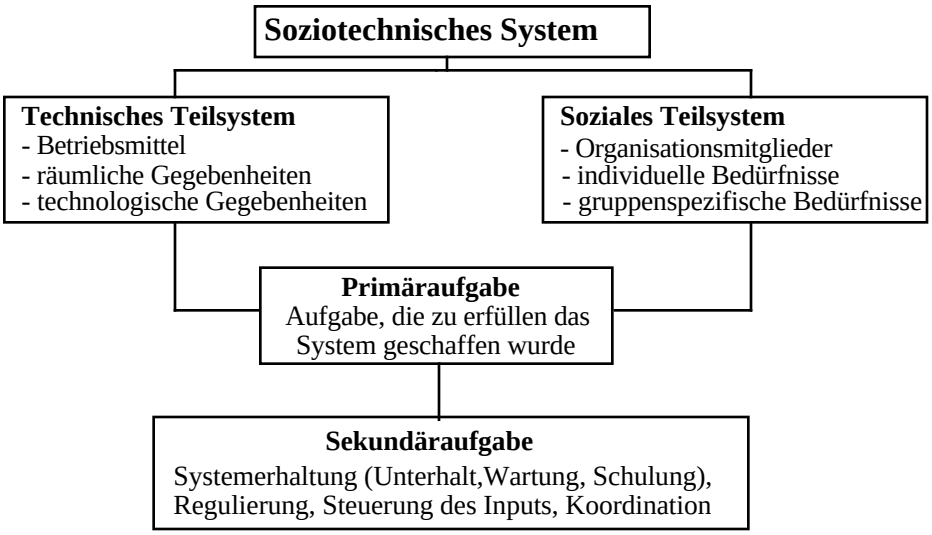

Abbildung 1: Das soziotechnische Systemkonzept.

Jeder Betrieb, jede Organisation muss als soziotechnisches System betrachtet werden, das aus zwei Teilsystemen besteht (siehe Abbildung 1). Von entscheidender Bedeutung ist dabei:

(1) Das technische und das soziale Teilsystem 'funktionieren' nach unterschiedlichen Regeln.

(2) Nur durch eine *gemeinsame Optimierung* des technischen *und* sozialen Teilsystems kann das *ganze* System optimal gestaltet werden.

Um optimale Lösungen zu erreichen, kann man also nicht jedes Teilsystem getrennt für sich optimieren oder ein Teilsystem an das andere anpassen, wie dies häufig zu beobachten ist. Das Konzept der soziotechnischen Systemgestaltung postuliert *explizit* die Notwendigkeit, den Einsatz von Technologie und die Veränderung der Organisation *gemeinsam* zu optimieren.

Das bedeutet nun aber keinesfalls, dass bei Innovationen immer der gesamte Betrieb oder eine ganze Abteilung reorganisiert werden muss. Wesentlich ist jedoch, dass für derartige Vorhaben eine soziotechnische Einheit des Betriebes ausgewählt wird, die ein gegenüber anderen Betriebsbereichen klar abgegrenztes Aufgabengebiet bearbeitet.

Ausgangspunkt für die verknüpfte Optimierung einer soziotechnischen Betriebseinheit ist deren Primäraufgabe und die Berücksichtigung einer Reihe von *Prinzipien* zur soziotechnischen Systemgestaltung (vgl. Ulich, Baitsch und Alioth 1987, S. 24f.):

(1) *Bildung von relativ unabhängigen Organisationseinheiten*
Den relativ unabhängigen Organisationseinheiten sind – als Mehrpersonenstellen – ganzheitliche Aufgaben zu übertragen, so dass sie aufgrund ihrer Unabhängigkeit und der Ganzheitlichkeit der Aufgaben in der Lage sind, Schwankungen und Störungen am Entstehungsort aufzufangen und selbst zu regulieren.

(2) *Aufgabenzusammenhang innerhalb einer Organisationseinheit*
Die verschiedenen Teilaufgaben innerhalb der Organisationseinheit müssen einen inhaltlichen Zusammenhang aufweisen, damit das Bewusstsein einer gemeinsamen Aufgabe entsteht und erhalten werden kann.

(3) *Einheit von Produkt und Organisation*
Aufbau- und Ablaufstrukturen müssen so gestaltet sein, dass Arbeitsergebnisse sowohl qualitativ als auch quantitativ Organisationseinheiten zugeordnet werden können.

Diese drei Prinzipien ermöglichen einerseits die Selbstregulation von Schwankungen und Störungen, welche an ihrem Entstehungsort aufgefangen werden können und somit nicht unkontrolliert auf eine andere Organisationseinheit übertragen werden. Andererseits bedingen sie die Grenzregulation durch den Vorgesetzten: Die Hauptaufgabe des Vorgesetzten besteht darin, die Beziehungen zwischen den verschiedenen Organisationseinheiten sicherzustellen und die Selbstregulation und Unabhängigkeit der Organisationseinheiten zu gewährleisten.

Diese Prinzipien verhindern unter anderem das Entstehen von technischen 'Sachzwängen', die sich üblicherweise dann ergeben, wenn technische Systeme ohne Berücksichtigung von Organisationsanforderungen konzipiert werden.

Tabelle 1: Merkmale der Aufgabengestaltung (nach Ulich 1994, S. 161).

Gestaltungsmerkmal	**Ziel / Absicht Vorteil / Wirkung**	**Realisierung durch ...**
Ganzheitlichkeit	• Mitarbeiter erkennen Bedeutung und Stellenwert ihrer Tätigkeit • Mitarbeiter erhalten Rückmeldung über den eigenen Arbeitsfortschritt aus der Tätigkeit selbst	... umfassende Aufgaben mit der Möglichkeit, Ergebnisse der eigenen Tätigkeit auf Übereinstimmung mit gestellten Anforderungen zu prüfen
Anforderungsvielfalt	• Unterschiedliche Fähigkeiten, Kenntnisse und Fertigkeiten können eingesetzt werden • Einseitige Beanspruchungen können vermieden werden	... Aufgaben mit planenden, ausführenden und kontrollierenden Elementen bzw. unterschiedlichen Anforderungen an Körperfunktionen und Sinnesorgane
Möglichkeiten der sozialen Interaktion	• Schwierigeiten können gemeinsam bewältigt werden • Gegenseitige Unterstützung hilft Belastungen besser ertragen	... Aufgaben, deren Bewältigung Kooperation nahelegt oder voraussetzt
Autonomie	• Stärkt Selbstwertgefühl und Bereitschaft zur Übernahme von Verantwortung • Vermittelt die Erfahrung, nicht einfluss- und bedeutungslos zu sein	... Aufgaben mit Dispositions- und Entscheidungsmöglichkeiten
Lern- und Entwicklungsmöglichkeiten	• Allgemeine geistige Flexibilität bleibt erhalten • Berufliche Qualifikationen werden erhalten und weiterentwickelt	... problemhaltige Aufgaben, zu deren Bewältigung vorhandene Qualifikationen erweitert bzw. neue Qualifikationen erworben werden müssen
Zeitelastizität und stressfreie Regulierbarkeit	• Wirkt unangemessener Arbeitsverdichtung entgegen • Schafft Freiräume für stressfreies Nachdenken und selbstgewählte Interaktionen	... Schaffen von Zeitpuffern bei der Festlegung von Vorgabezeiten
Sinnhaftigkeit	• Vermittelt das Gefühl, an der Erstellung gesellschaftlich nützlicher Produkte beteiligt zu sein • Gibt Sicherheit der Übereinstimmung individueller und gesellschaftlicher Interessen	... Produkte, deren gesellschaftlicher Nutzen nicht in Frage gestellt wird ... Produkte und Produktionsprozesse, deren ökologische Unbedenklichkeit überprüft und sichergestellt werden kann

1.2 AUFGABENGESTALTUNG

Arbeitsaufgaben stellen die Schnittstelle zwischen der Organisation und dem Individuum dar und bestimmen, welche Ziele eine Person oder Gruppe im Rahmen eines Gesamtauftrages erreichen soll.

Für die Erfüllung der Aufgaben stehen bestimmte Arbeitsmittel oder Werkzeuge zur Verfügung. Der Computer als technisches Unterstützungsmittel bei der Aufgabenbearbeitung verkörpert in diesem Sinne also ein Werkzeug.

Der zentrale Ansatzpunkt für die Gestaltung von Aufgaben bezieht sich auf deren *Motivationsgehalt*. Eine Aufgabe sollte so gestaltet sein, dass der Mensch sie als Herausforderung erlebt und Freude an ihrer Bewältigung hat. Eine solche innere Motivation aus der Aufgabe heraus entsteht dann, wenn bei der Gestaltung von Arbeitsaufgaben die folgenden sieben Gestaltungsmerkmale berücksichtigt werden (siehe Tabelle 1).

Aufgaben, die nach den Merkmalen 'Ganzheitlichkeit', 'Anforderungsvielfalt', 'Möglichkeiten der sozialen Interaktion', 'Autonomie', 'Lern- und Entwicklungsmöglichkeiten', 'Zeitelastizität' sowie 'Sinnhaftigkeit' gestaltet sind, fördern den Menschen in seiner sozialen und kognitiven Kompetenz, optimieren seine Beanspruchung, steigern sein Selbstwertgefühl und berücksichtigen seine Entwicklungsfähigkeit (Ulich 1994). Aufgaben werden vor allem auch dann als interessant und motivierend erlebt, wenn in ihnen planende, ausführende und kontrollierende Funktionen integriert sind. Dies kann zum Beispiel durch die Sachbearbeitung nach dem 'Fall-Prinzip' – ein Angestellter erfüllt alle im Zusammenhang mit einem Geschäftsfall zu erledigenden Teilaufgaben – realisiert werden.

1.3 SOFTWARE-ERGONOMIE

Wenn man den Menschen mit dem Computer vergleicht, so kann man feststellen, dass es sowohl für den Menschen als auch für den Computer spezifische Stärken und Schwächen gibt (siehe Tabelle 2). Der Computer als Arbeitsmittel dient primär dazu, den Benutzer bei seinen Arbeiten zu unterstützen und von unzumutbaren, sich ständig wiederholenden, z.B. hochgradig routinisierten Aufgaben zu entlasten. Dies bedeutet jedoch nicht, dass die vom Menschen bearbeiteten Aufgaben ausschliesslich kreativer Natur sein sollten, vielmehr sollte der Routineanteil durch den Einsatz von EDV auf ein sinnvolles Ausmass reduziert werden.

Tabelle 2: Vergleich von Mensch und Computer [LZG=Langzeitgedächtnis, KZG= Kurzzeitgedächtnis].

	Mensch	**Computer**
Stärken	sehr mächtige Fähigkeit zur Mustererkennung; Fähigkeit zur selektiven Aufmerksamkeitssteuerung; lebenslange Lernfähigkeit; 'unbegrenzte' Kapazität des LZG; assoziativer Zugriff auf Gedächtnisinhalte; komplexe multi-modale Speicherung; hohe Kontextsensitivität (z.B. bei Entscheidungen usw.); kreative Tätigkeiten;	permanente Verfügbarkeit des Speichers; zuverlässiger Speicherzugriff; relativ grosse Speicherkapazität; sehr schnelle Verarbeitungsgeschwindigkeit; 'fehlerfreie' Datenverarbeitung; Routineoperationen;
Schwächen	geringe Speicherkapazität des KZG; schnelle Vergessensrate des KZG; langsame Verarbeitungsgeschwindigkeit; fehleranfällige Verarbeitung; unzuverlässiger Zugriff auf das LZG.	sehr eingeschränkte Möglichkeit zur Mustererkennung; begrenzte Kapazität des 'LZG'; begrenzte 'Lernfähigkeit'; multi-modale Speicherung; geringe Kontextsensitivität.

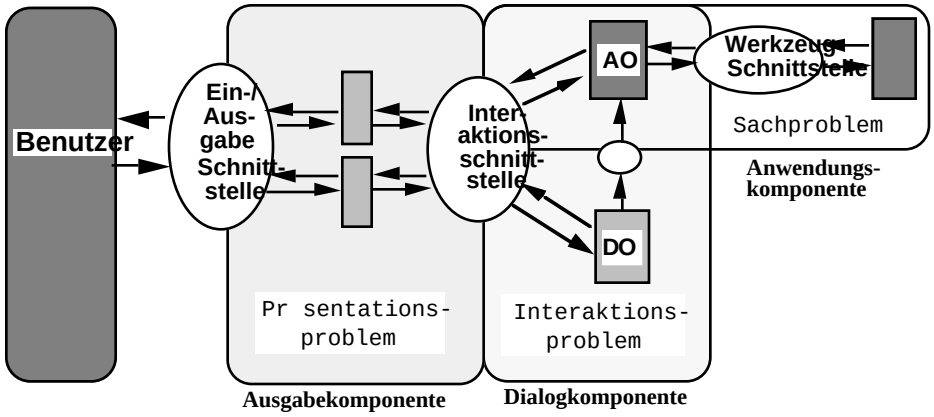

Abbildung 2: Übersicht über die verschiedenen Schnittstellen eines interaktiven EDV-Systems. [AO: Anwendungsoperatoren; DO: Dialogoperatoren]

Wenn zur Aufgabenbearbeitung interaktive EDV-Systeme eingesetzt werden, gibt es drei Problem- bzw. Gestaltungsbereiche: Das *Sachproblem*, das *Interaktionsproblem* und das *Präsentationsproblem* (siehe Abbildung 2).

Das *Sachproblem* besteht darin, dem Benutzer zur Unterstützung seiner Aufgabenbearbeitung alle notwendigen Funktionen und entscheidungsrelevanten Informationen auf einem geeigneten 'Granulations- und Darstellungsgrad' anzubieten (die Gestaltung der 'Werkzeugschnittstelle'). Diese aufgabenbezogenen Funktionen werden vom Benutzer durch die Eingabe entsprechender Anwendungsoperatoren (AO) ausgelöst bzw. aktiviert. Die Eingabe eines Dialog- bzw. Anwendungsoperators wird auch Dialog- bzw. Anwendungs-Operation genannt. Die vom System angebotenen Operationen und Informationen dürfen aber den Benutzer in seinem individuellen Problembearbeitungsprozess nicht unnötig einschränken. Die Lösung des Sachproblems bedeutet deshalb die aufgaben- und benutzerangepasste Gestaltung der Werkzeugschnittstelle mit allen notwendigen Anwendungsoperationen und -informationen zur Bearbeitung der gestellten Aufgaben.

Das *Interaktionsproblem* kommt im wesentlichen durch die Begrenzungen der Ausgabe- und Interaktionsschnittstelle zustande (siehe Streitz 1990). Damit der Benutzer (aufgabenbezogene) Informationen erhalten und die notwendigen Operationen durchführen kann, muss er die entsprechenden Anwendungs- bzw. Dialogoperatoren eingeben (AO bzw. DO). Hierzu stehen zur Zeit unterschiedliche Techniken zur Verfügung (Kommandos, Menüs, maussensitive Bereiche, PF-Tasten usw.; siehe Helander 1991).

Fast jede heutzutage existierende Benutzungsoberfläche verlangt nun vom Benutzer einige Dialogoperationen, welche lediglich zur Gestaltung der Ein-/Ausgabe- und der Interaktionsschnittstelle (z.B. Steuerung der Interaktion) dienen und nichts zur unmittelbaren Aufgabenbearbeitung beitragen. Der Unterschied zwischen diesen speziellen Dialogoperationen und den Anwendungsoperationen kann man sich am besten wie folgt merken: Alle Operationen, welche den Zustand des jeweiligen Anwendungsobjektes (z.B. Textdokument, Datenbank usw.) verändern, gehören zu den Anwendungsoperationen. Alle anderen Operationen (z.B. die Veränderungen von Fenstern, wie Öffnen, Schliessen, Verschieben usw.) sind 'reine' Dialogoperationen. Die Handhabung dieser Dialogoperationen sollte so einfach wie möglich sein und dem Benutzer keinen unnötigen zusätzlichen kognitiven Aufwand abfordern. (Dies darf jedoch nicht mit der Komplexität der Werkzeugschnittstelle und seiner Anwendungsoperationen verwechselt

werden, welche sich ausschliesslich nach der Komplexität der zu bearbeitenden Aufgaben zu richten hat!)

Das *Präsentationsproblem* besteht darin, die Ausgaben des Computers so zu gestalten, dass der Benutzer bei der Aufgabenbearbeitung mit allen notwendigen Informationen in seinem aktuellen Aufmerksamkeitsfokus auf der Ein-/Ausgabeschnittstelle versorgt wird und hierzu möglichst wenig Dialogoperationen benötigt (die Gestaltung der 'Ein-/Ausgabeschnittstelle'; siehe Abbildung 2). Die Ausgabe von aufgabenbezogenen Informationen auf der Ausgabeschnittstelle ist für den Benutzer das 'externe' Gedächtnis. Wie sich die drei Problembereiche gestalten lassen und was dies im einzelnen bedeuten kann, hängt von den spezifischen Eigenschaften menschlicher Wahrnehmungsleistungen, von Gedächtnisleistungen und den Handlungsmöglichkeiten ab.

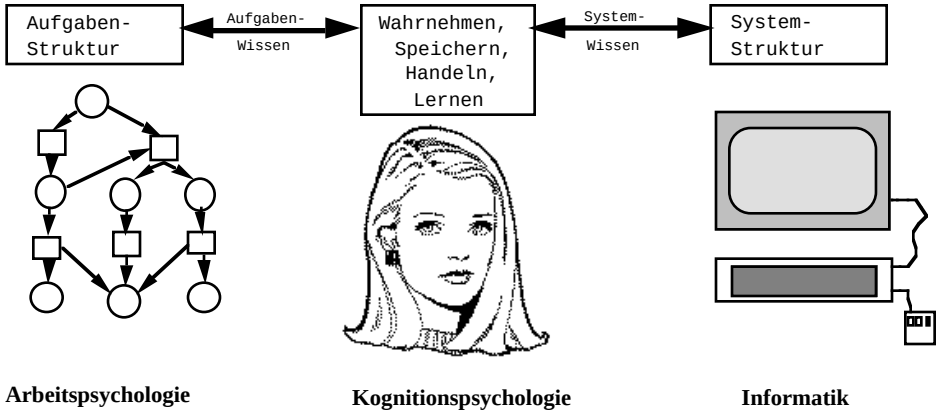

Abbildung 3: Übersicht über die drei verschiedenen Disziplinen.

Bei der benutzungsorientierten Systemgestaltung müssen Ergebnisse der Arbeitspsychologie, der Kognitionspsychologie und der Informatik integriert werden. Einerseits muss der Benutzer Wissen über die Bearbeitung von Aufgaben haben, andererseits muss er Wissen über die Handhabung des interaktiven Systems besitzen (siehe Abbildung 3).

Zunächst einmal sind die softwaretechnischen Kriterien der *Effizienz* und der *Zuverlässigkeit* von grundlegender Bedeutung. Die Entwicklung preiswerter Hardware und ihr Einsatz im Bereich der Arbeitsplatzrechner (PCs

und Workstation) lassen eine positive Entwicklung erkennen, bei der dem Benutzer eine möglichst grosse technische 'Power' zur Verfügung gestellt wird.

Tabelle 3: Übersicht über verschiedene Bewertungsmodelle der Mensch-Computer-Interaktion (siehe auch Reiterer 1990 und Ilg 1993).

Dzida, et al. (1977)	**DIN 66234 Teil 8 (1988)**	**VDI-Richtlinie 5005 (1990)**	**EG-Richtlinie 90/270/EWG (1990)**	**ISO 9241 Entwurf (1991)**	**Ulich (1994)**
	Aufgabenangemessenheit	Aufgabenangemessenheit	Tätigkeitsangepasstheit	Aufgabenangemessenheit	• Ganzheitlichkeit • Anforderungsvielfalt, • Autonomie, • soziale Interaktionsmöglichkeiten
Selbsterklärungsfähigkeit	Selbsterklärungsfähigkeit		Angaben über die jeweiligen Systemabläufe	Selbsterklärungsfähigkeit	• Transparenz, • Unterstützung
Rückkopplungsfähigkeit					• Feedback
	Erwartungskonformität		Format- und tempogerechte Informationsdarstellung	Erwartungskonformität	• Konsistenz • Kompatibilität
Dialogflexibilität		Handlungsflexibilität			• Flexibilität
einfache Anwendbarkeit				Erlernbarkeit	• Lernmöglichkeiten
		Kompetenzförderlichkeit	Unterrichtung und Unterweisung		• Entwicklungsmöglichkeiten
			Anpassbarkeit an Kenntnis- und Erfahrungsstand	Individualisierbarkeit	• individuelle Auswahl, • individuelle Anpassung
			Anhörung und Beteiligung		• Partizipation
	Steuerbarkeit			Steuerbarkeit	• Flexibilität
Zuverlässigkeit	Fehlerrobustheit			Fehlertoleranz	[wird vorausgesetzt]

Erst wenn diese grundlegenden Eigenschaften des technischen Systems gewährleistet sind, lassen sich sinnvolle Gestaltungsmassnahmen aus arbeitspsychologischer Sicht für die interaktive Software umsetzen. Es wird daher im weiteren vorausgesetzt, dass das technische System *keine wesentlichen Beschränkungen* hinsichtlich Antwortzeit und Speicherplatz aufweist und auf jeden Fall *zuverlässig* arbeitet.

Arbeitspsychologische Kriterien lassen sich den verschiedenen Phasen eines vollständigen Handlungsvollzuges zuordnen. Für die Unterstützung der *Orientierung* ([1] Zielsetzung, [2] Planung, [4] Kontrolle) zu Beginn und während einer Handlung sind die Kriterien *Transparenz*, *Feedback*, *Kompatibilität* und *Konsistenz* sowie der Bereich *Unterstützung* wichtig.

Die verschiedenen Bewertungsmodelle der Mensch-Computer-Interaktion (Dzida et al. 1977; DIN 66234 1988; VDI 5005 1990; ISO 9241 1991; Ulich 1994) zeigen unterschiedliche Schwerpunkte und Differenzierungsgrade (siehe Tabelle 3). Das Modell von Ulich (1994) ist nicht nur handlungspsychologisch begründbar und empirisch abgesichert, sondern auch am weitesten differenziert. Die meisten Übereinstimmungen mit dem Modell von Ulich finden sich in der EG-Richtlinie und der ISO 9241.

Kasten 1: Ein Beispiel für das Kriterium *Transparenz*.

> Benutzer können erkennen, ob ein eingegebener Befehl behandelt wird oder ob das System auf weitere Eingaben wartet. Bei längeren Vorgängen sollte das System Zwischenzustandsmeldungen abgeben [können].

Die *Transparenz* eines Systems soll dem Benutzer die Bildung eines Struktur- und Prozessmodells des Systems im Gedächtnis erleichtern, was ihm die notwendigen Orientierungsgrundlagen für die Benutzung bietet. Der Benutzer muss sich ja über den Inhalt (Funktionen bzw. Operationen, Informationen) und dessen Organisation im System (Zugriffspfade, Ordnungskriterien) generell sowie über den aktuellen Dialogzustand ein 'Bild' machen können, um die technische Unterstützung auch effizient nutzen zu können. Ein System sollte dementsprechend seine Nutzungs- und Kontrollmöglichkeiten sowie aktuelle Bearbeitungszustände dem Benutzer an der

Benutzungsoberfläche strukturiert und verständlich offenlegen (siehe auch Maass 1994).

Nach Wandke (1988, S. 4) kann es für den Benutzer – insbesondere bei abfragenden Systemen mit 'prompting-technique' – in Ermangelung erkennbarer Ziele und eines Bezugssystems zu Ziel- und Orientierungslosigkeit sowie einer Überlastung des Kurzzeitgedächtnisses kommen, da "alle vorausgegangenen und nachfolgenden Nutzereingaben und Systemausgaben, die einen Einfluss auf die aktuelle Nutzereingabe haben, im Arbeitsgedächtnis behalten werden müssen". Typische Benutzeraussage: "Wenn ich etwas zurückverfolgen will, muss ich immer wieder von Bild zu Bild gehen, um die Informationen abzurufen, vorher habe ich einfach meine Papiere durchgeblättert und hatte alles auf *einen Blick.*" Insgesamt sind integrale, parallele Informationsdarstellungen gegenüber sequentiell dargebotenen Informationen von Vorteil. Benz und Haubner (1983) demonstrierten eindrücklich den positiven Einfluss, den nach Gestaltgesetzen strukturierte, 'geordnete' Bildschirm-Masken auf Leistung, Beanspruchung und Akzeptanz der Benutzer ausüben. Der positive Effekt strukturierter Menüs ist bekannt. Die höhere Transparenz der direktmanipulativen Benutzungsoberfläche gegenüber einer menü- und funktionstastenorientierten Oberfläche ist u.a. auf den grösseren, permanent sichtbaren Informationsgehalt sowie eine deutlich grössere Flexibilität in der Interaktion zurückzuführen.

Zusammengefasst ist festzuhalten, dass die *Transparenz* ein zentrales, aber schwer fassbares bzw. isolierbares Kriterium darstellt, das viele Bezüge zu anderen Kriterien – wie Konsistenz, Unterstützung, Feedback – aufweist und ausserdem stark von der Erfahrung des Benutzers abhängt.

Kasten 2: Ein Beispiel für das Kriterium *Feedback*.

> Die für die Planung und Ausführung von Tätigkeiten zentralen Rückmeldungen (Feedback) werden im aktuellen Dialogkontext für den Benutzer wahrnehmbar ausgegeben.

Zu den wesentlichen Merkmalen des Kriteriums *Feedback* im Rahmen einer vollständigen Aufgabenbearbeitung gehören: 'Ausführungsfunktionen mit Ablauffeedback zur allfälligen Handlungskorrektur' sowie 'Kontrolle mit Resultatfeedback und der Möglichkeit, Ergebnisse der eigenen Handlungen auf Übereinstimmung mit den gesetzten Zielen zu überprüfen'.

Im Bereich der Softwareergonomie haben sich verschiedene Untersuchungen mit einer speziellen Form von Rückmeldung – den Fehlermeldungen – beschäftigt, die erst dann erforderlich wird, wenn der Benutzer eine Aktion nicht systemgerecht durchgeführt hat. So hat Norman (1983) eine differenzierte Fehlerklassifikation vorgestellt und Hinweise für die Gestaltung von Schnittstellen und Fehlermeldungen abgeleitet. Dabei hat sich gezeigt, dass geeignetes Feedback zur Vermeidung mancher Fehler beitragen würde. Fehler sollten allerdings nicht um jeden Preis vermieden bzw. vom System toleriert werden, sondern auch gezielt für Lernprozesse eingesetzt werden. Zapf und Frese (1989) betrachten Fehler als 'mismatch' (Unangepasstheit) von Aufgabe und Computer (Funktions- bzw. Sachproblem) sowie von Computer und Benutzer (Nutzungs- bzw. Interaktionsproblem). Die meisten Fehler treten bei Operationen auf, die ein hohes Mass an Vorausplanung erfordern, wenn kein geeignetes Feedback gegeben wird. Fehlermeldungen mit mangelnder Angabe von Fehlerursache und Korrekturmöglichkeiten werden von den Benutzern auch dementsprechend schlecht beurteilt.

Kasten 3: Beispiele für das Kriterium *Konsistenz*.

> Das System sowie dessen *Antwortverhalten* sollten für Benutzer transparent und konsistent sein. Ähnliche Aktionen sollten ähnliche Ausführungen bewirken, andernfalls muss dies durchschaubar gemacht werden.
> Z.B. die Belegung der Funktionstasten, Menüoptionen, Schaltflächen auf dem Bildschirm sollte über alle Dialogkontexte hinweg innerhalb eines Programmes wie auch über die anderen verwendeten Programme hinweg einheitlich und damit konsistent sein.

Das Kriterium der *Konsistenz* bezieht sich auf die Berechenbarkeit des Systemverhaltens, wodurch Erwartungen des Benutzers erfüllt und Überraschungseffekte vermieden werden. Ein konsistent-regelhafter Aufbau von Dialogstruktur, Semantik und Syntax der Operationen (z.B. Kommandos) ermöglicht dem Benutzer die Rekonstruierbarkeit und entlastet dadurch das Gedächtnis von faktisch präziser und damit aufwendiger Speicherung. Generell lässt sich sagen, dass ein System um so konsistenter ist, je weniger Regeln man zur Beschreibung des Systems benötigt. Norman (1983) stellte fest, dass Inkonsistenzen in der Systemhandhabung die Quelle mancher Benutzerfehler sind.

Unter dem Kriterium *Kompatibilität* versteht man den Grad an Übereinstimmung zwischen den semantischen Gedächtnisstrukturen des Benutzers und den – durch die Aufgabenerfordernisse bestimmten – Benutzeraktionen. Norman (1989) spricht in diesem Zusammenhang auch von *natürlichen Mappings*, womit die Nutzung äusserer Analogien, Metaphern und kultureller Standards gemeint ist, welche zu einem *unmittelbaren Verständnis* führen (z.B. Drehen des Autolenkrades im Uhrzeigersinn lenkt das Auto nach rechts usw.). Da viele natürliche Mappings *analoger Natur* sind, empfiehlt sich oftmals der Einsatz von grafischen, *direkt-manipulierbaren Benutzungsoberflächen*.

Kasten 4: Ein Beispiel für das Kriterium *Kompatibilität*.

> Bei der *Darstellungsform* für Einzelinformationen sollte ebenso wie für ganze Bilder ggf. auf Übereinstimmung mit entsprechenden gedruckten Vorlagen oder Unterlagen geachtet werden. *Sprache* und begriffliche Komplexität des Dialoges sollten an den Gepflogenheiten und Kenntnissen des spezifischen Benutzerkreises orientiert sein. Anstelle von EDV-Kürzeln sollte mit den jeweils fachspezifischen Begriffen der Benutzer gearbeitet werden können.

Eine direkte Manipulation erfolgt durch analoges Bewegen, Verändern oder in eine bestimmte Relation Setzen (z.B. Anwendung einer Funktion) von grafisch visualisierten Objekten auf dem Bildschirm. Die analogen Operationen werden in der Regel durch ein entsprechendes Zeigegerät (z.B. Maus, Joystick usw.) ausgeführt. Da der Benutzer Objekte in einer Scheinwelt (z.B. die 'Desktop'-Metapher) direkt manipulieren kann, hat er oftmals den Eindruck, mit realen Objekten zu agieren. Direktmanipulierbare Benutzungsoberflächen zeichnen sich durch folgende Eigenschaften aus:

(1) Die permanente Sichtbarkeit der relevanten Objekte;

(2) schnelle, umkehrbare, einstufige Operationen mit unmittelbarer Rückmeldung und

(3) die Ersetzung komplexer, 'digitaler' Kommandos durch physische, 'analoge' Aktionen (z.B. Mausbewegungen, Selektionsoperationen usw.).

Zuerst wählt der Benutzer das relevante Objekt (O) aus und wendet dann eine entsprechende Funktion (F) darauf an. Diese Art der Interaktion nennt man daher auch O-F-Struktur. Im Gegensatz dazu herrscht bei den traditionellen Benutzungsoberflächen (Frage-Antwort, Kommando, Menü, Formular-Dialog) meistens die F-O-Struktur vor. Sehr viele alltägliche Handlungen mit Objekten der realen Welt sind jedoch nur mit der O-F-Struktur kompatibel.

Um eine entsprechende Kompatibilität auf der Ein-/Ausgabeschnittstelle zu erreichen, werden zur Zeit multimediale Benutzungsoberflächen eingesetzt. Multimediale Benutzungsoberflächen bedingen die Ausarbeitung entsprechender Metaphern und die dramaturgische Erarbeitung nichtsequentieller Vermittlungsformen. Multimediale Benutzungsoberflächen erhöhen drastisch die Chancen, dass sich die aufgabenangemessene Darstellung der Bearbeitungsobjekte auf eine benutzungsgerechte Systemnutzung positiv auswirken kann. Dabei führt das multimediale Design zur Auflösung und Neufassung der traditionellen Gestaltungsregeln der verschiedenen, beteiligten Einzelmedien: Text, Bild, Ton und Film. Beim Informationstransfer zwischen den Medien werden nicht nur Daten transportiert, viel wesentlicher ist die expressive und inhaltliche Gestaltung, die sich durch die Übertragung von Information in ein neues Darstellungsmedium ergibt. So ändert sich z.B. die Wirkung numerischer Daten, sobald diese in eine grafische Darstellung übertragen werden, mit der sich sogar interaktiv bestimmte Simulationen durchführen lassen. In dieser Wirkungsverschiebung liegen die herausragenden Vorteile multimedialer Benutzungsoberflächen.

Durch die Kompatibilität soll eine Reduktion mentaler Transformationsschritte des Benutzers und damit eine Entlastung des mit beschränkter Kapazität arbeitenden Kurzzeitgedächtnisses erreicht werden. Ulich (1986) hat in diesem Zusammenhang eine sinnvolle Unterscheidung in *Sprach*- und *Darstellungskompatibilität* vorgenommen. So haben Untersuchungen gezeigt, dass die Gestaltung von Dialogsprachen in Anlehnung an die natürliche (Fach-)Sprache (Vokabular; Abkürzungen; Reihenfolge von Subjekt, Verb, Objekt usw.) des Benutzers von Vorteil sein kann.

Das Kriterium *Unterstützung* bezieht sich auf rechnergestützte Lern- und Arbeitshilfen, die dem Benutzer über die normalen Dialogmeldungen hinausgehende Informationen zur Bewältigung von Problemsituationen zur Verfügung stellen. Diese Lern- und Arbeitshilfen sollen dem Benutzer die Orientierung erleichtern und ihm helfen, Software besser zu durchschauen (Informationen über wichtige Zusammenhänge, Erklärung von Ereignis-

sen, Handlungsergebnissen usw.) sowie Dialogabläufe besser vorherzusehen (Bilden von Hypothesen und Prognosen) und gemäss eigenen Zielen zu beeinflussen. *Unterstützung* muss aufgabenorientiert gestaltet sein und sich auf die Dialog- und Anwendungskomponente der Benutzungsoberfläche beziehen. Die Kontrolle über die rechnergestützten Lern- und Arbeitshilfen (z.B. Bestimmung des Umfangs der Hilfemeldungen) soll beim Benutzer und nicht beim System (Adaptivität) liegen. Möglichkeiten der Unterstützung von Benutzern werden unter so unterschiedlichen Begriffen wie Selbsterklärungsfähigkeit, 'advice-giving systems', 'on-line assistance', 'users support' oder 'user assistance' beschrieben. Sie lassen sich folgenden drei Grundformen zuordnen: Computerunterstützte Handbücher, Hilfesysteme und Tutorials.

Kasten 5: Ein Beispiel für das Kriterium *Unterstützung*.

> *Dialoghilfen* sowohl zu inhaltsbezogenen wie zu vorgehensbezogenen Aspekten sollten von den Benutzern während des Dialogs jederzeit abgerufen werden können. Das Betätigen einer allfälligen Help-Taste sollte gegenüber anderen Befehlen einen Sonderstatus einnehmen. Das System sollte eine *Rückfragemöglichkeit* derart bereitstellen, dass auf eine Aufforderung durch die Benutzer hin ggf. ausführlichere, aufgabenbezogene Antworten abgegeben werden.

Das elektronische, am Bildschirm zugängliche Handbuch wird verschiedentlich als Alternative zum traditionellen Handbuch angesehen. Der Hauptnachteil der am Bildschirm zur Verfügung gestellten Information liegt in der im Vergleich zur gedruckten Version deutlich schlechteren Lesbarkeit. Grössere Textpassagen sind deshalb immer in einem gedruckten Benutzerhandbuch zur Verfügung zu stellen. Hinweise zur Gestaltung gedruckter Handbücher und Bedienungsanleitungen befinden sich z.B. in Rupietta (1987) und Boedicker (1990).

Rechnergestützte Auskunfts- und Informationssysteme, die dem Benutzer Informationen über Kommandos, Abfragen oder Dialogzustände zur Verfügung stellen, sollten immer verfügbar, richtig und vollständig sein und ein – hinsichtlich Leseniveau, Umfang und Länge – einheitliches Format haben. Wichtig ist in diesem Zusammenhang, dass die Inhalte möglichst kontext-sensitiv, das heisst, auf den konkreten Bearbeitungszustand ausge-

richtet sind. An den meisten Hilfesystemen wird bemängelt, sie seien für Programmierer geschrieben und deshalb zu abstrakt, zu algebraisch und voll von Computerjargon. So lässt sich u.a. die Sprache dem Benutzer anpassen – Kriterium der Sprachkompatibilität – und statt systemorientierter Meldungen können aufgabenorientierte Informationen zur Verfügung gestellt werden. Die Ergebnisse empirischer Studien zeigen deutlich, dass auch Computerlaien von On-line-Hilfesystemen profitieren können, wenn auch die Hilfemeldungen den Kriterien benutzungsorientierter Dialoggestaltung entsprechen.

Die nun folgenden Kriterien *Flexibilität, individuelle Auswahl* und *individuelle Anpassung* beziehen sich primär auf die Planung, Auswahl der Mittel und Ausführung einer Handlung (siehe Teil A, Kapitel 1.2).

Kasten 6: Das Kriterium *Flexibilität*.

> Flexibilität eines interaktiven Systems wird durch das vorhandene Ausmass an Handlungsspielraum festgelegt. Der Handlungsspielraum ist die Summe (objektiv) vorhandener Freiheitsgrade zur selbständigen Setzung und Erreichung von (Teil-)Zielen durch variable Abfolge von (Teil-)Schritten.
> Der Benutzer kann aus einer vorgegebenen Menge von Alternativen auswählen und beliebig zwischen den verschiedenen Alternativen wechseln, da ihm alle Möglichkeiten jederzeit zur Auswahl zur Verfügung stehen (z.B. Tastatur und Maus).

Flexibilität gilt wohl zu Recht als eines der zentralen Kriterien der Dialoggestaltung, entscheidet sich doch an der Flexibilität bzw. Rigidität eines Systems, ob sich der Mensch als Benutzer oder als Bediener des Computers fühlt. Mit der Flexibilität ist die Beeinflussbarkeit des Systemverhaltens im Sinne objektiv vorhandener Freiheitsgrade – vermittelt über den Handlungsspielraum – gemeint. Der Handlungsspielraum dient zur selbständigen Setzung und Erreichung von (Teil-)Zielen durch variable Abfolge von (Teil-)Schritten. Die Steuerung des Dialogablaufes – und damit die Reihenfolge von Operationen und Arbeitsschritten – sollte in inhaltlicher und zeitlicher Hinsicht sowie bezüglich einzusetzender Mittel weitgehend dem Benutzer überlassen werden. Dadurch können unterschiedliche Vorgehensweisen – die durch sich ändernde Aufgabenerfordernisse und/oder unter-

schiedliche Arbeitsweisen der Benutzer bedingt sein können – ermöglicht werden.

Der Bedeutung des Kriteriums entsprechend haben sich auch weltweit Wissenschaftler damit befasst und sind zu mehrheitlich übereinstimmenden Ergebnissen gekommen: Flexibilität eines Dialogsystems ist unbedingt notwendig, damit der Benutzer in weiten Grenzen Teilaufgaben selbständig definieren, deren Bearbeitungsreihenfolge festlegen und auch bei unvorhergesehenen Ereignissen seine Pläne ändern kann. Der Benutzer sollte den Dialogablauf nach seinen kognitiven Prozessen ausrichten können, selbst assoziative Gedankensprünge sollten durch Sprungmöglichkeiten im Dialog realisierbar sein. Demgegenüber erzwingen rigide Dialogstrukturen eine Anpassung der kognitiven Prozesse des Benutzers an die im System festgelegte, abstrakt-logische Aufgabenstruktur, was sich in schlechteren Leistungen und vor allem auch im subjektiven Erleben der Benutzer in Form von Unzufriedenheit und geringer Akzeptanz des Systems ausdrückt. Benutzer derartiger Systeme fühlen sich laut ihren Aussagen in ein Schema gepresst, das ihren Bedürfnissen und ihrer Vorgehensweise bei der Aufgabenbewältigung nicht gerecht wird und sie zum Bediener des Computers degradiert.

Präzisierend ist allerdings zu ergänzen, dass Anfänger eine stärkere Systemführung – und damit geringere Flexibilität – begrüssen, nach relativ kurzer Zeit aber schon das Bedürfnis nach grösseren Kontrollmöglichkeiten äussern, wie dies in einigen empirischen Studien zum Ausdruck kommt. Dialogflexibilität ist offensichtlich nicht nur notwendig, um unterschiedlichen Vorgehensweisen der Benutzer Rechnung zu tragen, sondern darüber hinaus werden dadurch auch individuelle Bearbeitungsweisen berücksichtigt, die sich durch Lernprozesse über die Zeit ergeben.

Kasten 7: Das Kriterium *individuelle Auswahlmöglichkeit*.

> Der Benutzer kann das Systemverhalten durch die Einstellung von Systemparametern (Schaltern usw.) auf seine individuellen und aufgabenbezogenen Bedürfnisse abstimmen. Der Benutzer kann aus einer vorgegebenen Menge von Alternativen eine der angebotenen Möglichkeiten auswählen; anschliessend ist jeweils nur diese ausgewählte Möglichkeit aktiv und benutzbar.

Im Unterschied zur Flexibilität eines Systems, die für Gruppen von Benutzern Freiheitsgrade für unterschiedliche Vorgehensweisen ermöglicht, soll die *individuelle Auswahlmöglichkeit* und die *individuelle Anpassung* eines Systems allen Benutzern die Möglichkeit bieten, die Benutzungsoberfläche nach ihren persönlichen Bedürfnissen massgeschneidert einzurichten; im Englischen werden dafür häufig die Begriffe 'taylorized', 'customized', 'personalized' bzw. 'individualized' gebraucht.

Es sollte dem Benutzer eines Systems möglich sein, durch einfache Veränderung einiger Parameter bzw. Schalter für Systemeinstellungen die seinen Kenntnissen und Bedürfnissen gerechte Formen der Interaktion selbst zu realisieren (z.B. Darstellung von Informationen, Belegung von Funktionstasten usw.). In der Praxis haben sich Systeme bewährt, deren Benutzungsoberfläche unterschiedliche Interaktionsformen – z.B. Menü und Kommando, Maus und Tastatur – umfassen, die je nach Erfahrung des Benutzers von ihm selbst ausgewählt werden können.

Im Unterschied zur Flexibilität bietet die *individuelle Auswahl* alternative Nutzungsvarianten an, zwischen denen sich der Benutzer im Sinne von Wahlmöglichkeiten entscheiden kann. Sobald er jedoch seine Entscheidung getroffen hat, ist sein weiterer Bearbeitungsweg festgelegt und kann (im Gegensatz zur Flexibilität) nicht mehr unmittelbar neu gewählt werden. Erst nach einem zusätzlichen Auswahldialog (z.B. Setzen verschiedener Schalter der Systemeinstellungen) kann eine andere Nutzungsvariante aktiv werden.

Kasten 8: Das Kriterium *individuelle Anpassung*.

> Individuelle Anpassungsmöglichkeiten bieten die Möglichkeit der eigenständigen Gestaltung ('Eigenprogrammierung') und dementsprechend auch der Erweiterung objektiver Handlungsspielräume. Der Benutzer kann die Menge von Benutzungsalternativen erweitern.

Eine weitere Möglichkeit der Individualisierbarkeit wird als *individuelle Anpassung* bezeichnet. In diesem Fall erlaubt das System dem Benutzer Änderungen am System selber vorzunehmen (Adaptierbarkeit), oder das System verändert sich selbst aufgrund der Nutzungsart des Benutzers (Autoadaptierbarkeit, siehe Haaks 1992). Bei autoadaptiven Systemen sollte der Benutzer Kontrolle über diese Autoadaptierbarkeit haben.

Es zeigte sich im Rahmen einer empirischen Studie, dass bei der Benutzung eines simulierten Informationsabfragesystems eine fast doppelt so hohe Leistung von Benutzern erreicht wurde, wenn sie das jeweils aktive Deskriptorensystem individuell selbst entwickelt hatten. Demgegenüber musste eine Kontrollgruppe mit einer vorgegebenen 'Fremdkonstruktion' des Deskriptorensystems arbeiten.

Ein Dialogsystem ist dann benutzungsfreundlich, wenn dem Benutzer alle zur Aufgabenerfüllung benötigten Informationen und Funktionen in einer der verlangten Arbeitshandlung entsprechenden Form und Reihenfolge angeboten werden. Es konnte gezeigt werden, dass dem Benutzer für manipulative Operationen – wie z.B. Editieren von Texten – besser bildliche Symbole dargeboten werden sollten, für die Auslösung sequentiell strukturierter Ereignisse – wie z.B. Dateihandhabung – dagegen Kommandoworte angemessener sind. Ebenso fand sich ein Zusammenhang zwischen dem Aufgabentyp und einer Dialogtechnik, die dem Benutzer effizientes Arbeiten erlaubte. Die Frage der Gestaltung von Fenstern – überlappende versus nicht überlappende Fenster – ist nur in Abhängigkeit von der zu unterstützenden Aufgabe zu beantworten. Die Abhängigkeit der Dialogstruktur vom Aufgabengebiet und vom Arbeitsablauf des Benutzers wurde insbesondere in Untersuchungen zur Strukturierung von Menüsystemen nachgewiesen. Es konnte schliesslich noch gezeigt werden, dass mit steigendem Wiederholungsgrad und zunehmender Komplexität einer Aufgabe die Benutzer vermehrt von den Möglichkeiten zur Bildung von Makros Gebrauch machen.

1.4 FAZIT

Erst wenn die Benutzungsoberfläche insgesamt software-ergonomisch benutzungsgerecht gestaltet ist und dadurch das Interaktionsproblem minimiert wurde, ist die individuelle Nutzungsweise kein erschwerendes Hindernis, sondern eine echte Unterstützung und Entlastung des Benutzers bei seiner Aufgabenerfüllung. Zusammengefasst kann man sagen, dass Normen, Kriterien und Richtlinien einerseits notwendige Hinweise, wie benutzungsgerechte Oberflächen zu gestalten sind, liefern, andererseits aber auch die Möglichkeit bieten, Schnittstellen zu vereinheitlichen. Richtlinien erfüllen folgende zentrale Funktionen:

(1) Sie sind Hilfsmittel, individuelle Beiträge zur Schnittstellengestaltung zu koordinieren und zu vereinheitlichen.

(2) Entscheidungen werden nur einmal getroffen und nicht immer wieder von einzelnen Beteiligten verändert.

(3) Anhand von Richtlinien lassen sich detaillierte Anforderungskataloge erarbeiten.

(4) Die erstellten Anforderungskataloge können gleichzeitig als Grundlage für die Evaluation der Zwischenstadien und Endprodukte dienen.

(5) Richtlinien können als Grundlage für die Koordination der Zusammenarbeit verschiedener am Projekt beteiligter Gruppen dienen.

Im Rahmen dieses Kapitels wurden arbeitspsychologische Grundlagen und begründete Kriterien für die Gestaltung von Arbeitssystemen, Arbeitsaufgaben und Benutzungsoberflächen dargestellt. Diese Kriterien sind eine wichtige Orientierungshilfe für die Gestaltung technisch-organisatorischer Strukturen und Abläufe, die den Ansprüchen humaner und wirtschaftlich effizienter Arbeit genügen sollen. Diese Kriterien ersetzen allerdings keine Beteiligung von Mitarbeitern bei technisch-organisatorischen Veränderungen. Diese Kriterien können vielmehr einen solchen Beteiligungsprozess unterstützen, indem die Gestaltungshinweise dieser Kriterien mit den Beschäftigten für den konkreten Arbeits- und Aufgabenkontext spezifiziert und umgesetzt werden. Vor diesem Hintergrund werden im nächsten Kapitel Grundlagen der Benutzerbeteiligung vermittelt.

2 Grundlagen der Benutzerbeteiligung

In diesem Kapitel werden zuerst die Gründe für eine Beteiligung von Benutzern bei der Softwareentwicklung und die Voraussetzungen dazu beschrieben. Anschliessend werden spezifischer die unterschiedlichen Möglichkeiten der Kooperation von Benutzern und Softwareentwicklern dargestellt (siehe dazu auch Mambrey und Oppermann 1983, Mumford und Welter 1984, Kissler 1988, Koslowski 1988, Mambrey, Oppermann und Tepper 1986 sowie Peschke 1986).

2.1 GRÜNDE FÜR BENUTZERBETEILIGUNG

Vielfältige Praxiserfahrungen lassen folgende mögliche Vorteile von Benutzerbeteiligung erwarten.

Für den Anwender:

(1) Bessere Aufgabenangemessenheit der Software,

(2) höhere Akzeptanz durch Benutzerorientierung,

(3) geringere Kosten durch Vermeidung nachträglicher, kostenintensiver Korrekturen.

Für die Benutzer:

(1) Einsicht in Verständnis für Möglichkeiten und Grenzen neuer Technologien und die Arbeit der Entwickler,

(2) grössere Identifikation mit bzw. höhere Akzeptanz der Lösung,

(3) höhere Motivation zur Benutzung des Systems.

für die Entwickler:

(1) Feedback über ihr Produkt,

(2) Qualifizierungsmöglichkeiten in Fachfragen,

(3) bessere Einsicht in die Arbeit der Benutzer,

(4) grössere Sicherheit bei der Lösungsfindung.

2.1.1 Widerstand gegen Veränderung – Akzeptanz und Identifikation

Bei technisch-organisatorischen Veränderungen in Unternehmen sind auf allen hierarchischen Ebenen immer wieder *Ängste und Widerstände* bei den von der Veränderung Betroffenen zu beobachten, die sich auf verschiedene Weise manifestieren können: Minderung der Arbeitsleistung, Vermehrung der Absenzen, Gesuche um Versetzung, Stellenwechsel usw. Was sind aber die *Ursachen von Widerstand gegen Veränderungen?*

> Die wahre Natur des Widerstands: Unsicherheit und Ungewissheit über die *sozialen* Veränderungen.

Widerstand bezieht sich zumeist nicht auf die technischen, sondern auf die *sozialen Veränderungen*, mit denen in der Folge von angekündigten technisch-organisatorischen Veränderungen gerechnet wird. Hierzu zählen unter anderem die Veränderungen bezüglich zwischenmenschlicher Beziehungen, Selbständigkeit, Arbeitsinhalt, Status usw. Gerade die Einführung neuer Technologien löst bei den Betroffenen häufig Unsicherheit und Ungewissheit aus und führt zu Spekulationen über die zukünftige Arbeitssituation. Der Mensch hat aber ein grundlegendes Bedürfnis nach Klarheit über Ereignisse und Prozesse, die in seiner Umwelt ablaufen und ihn betreffen; zudem möchte er auch korrigierend eingreifen können, falls etwas seinen Interessen oder Ansichten zuwiderläuft. Auf betriebliche Veränderungen bezogen heisst dies:

Die Mitarbeiter möchten

(1) die Gründe für eine Innovation verstehen,

(2) die Folgen für die Veränderung ihrer Arbeitsbedingungen abschätzen und auch

(3) an der Veränderung mitwirken können, um ihre Meinung, ihre Erfahrungen und ihr Wissen einzubringen.

Bei betrieblichen Innovationsvorhaben ist allerdings häufig zu beobachten, dass vorwiegend Experten bzw. Spezialisten eingesetzt werden, welche eine Neuerung vorbereiten und einführen. Ihr formeller Auftrag bezieht sich häufig ausschliesslich auf technische und/oder organisatorische Probleme, zu deren Bewältigung *keine formellen* Kommunikationskanäle zu

den Betroffenen vorgesehen sind. Dies kann dazu führen, dass die Experten sich einseitig nur mit den *technischen* Aspekten neuer Ideen beschäftigen, sich vorwiegend an ihren eigenen Interessen und Vorstellungen orientieren und das Fachwissen der Betroffenen, deren Kenntnisse und Fertigkeiten aus der täglichen praktischen Erfahrung, unterschätzen.

> *Möglichkeiten zur Verringerung von Widerstand:* Information, Ausbildung und Beteiligung der Betroffenen!

Angesicht dieser Überlegungen scheinen die erforderlichen Massnahmen bei der Planung technisch-organisatorischer Neuerungen naheliegend zu sein. Durch eine frühzeitige und fortlaufende *Information* der Betroffenen über die Innovation können Ungewissheiten reduziert sowie Spekulationen und Gerüchtebildungen weitgehend vermieden werden.

Über die Information hinaus kann Befürchtungen der Betroffenen, die zukünftigen Anforderungen mit den bisherigen Kenntnissen und Fertigkeiten nicht mehr bewältigen zu können, durch rechtzeitige *Ausbildungsmassnahmen* begegnet werden.

Information und Ausbildung befähigen die Betroffenen auch zur *Mitwirkung an der Veränderung*. Für die Organisation ergibt sich dadurch die Chance, die in der täglichen Praxis erworbenen Erfahrungen der Mitarbeiter bezüglich einer effizienten Erfüllung der Arbeitsaufgaben zur Neugestaltung der Arbeitsorganisation zu nutzen. Die Nutzung dieses Fachwissens kann einen wesentlichen Beitrag zu Optimierung von Arbeitsabläufen und organisationalen Strukturen leisten. Ausserdem ist die Identifikation der Beteiligten mit dem Endprodukt grösser, wenn sie ihre Vorschläge verwirklicht sehen, und damit auch die Motivation, mit dem System zu arbeiten; das System erfährt eine insgesamt höhere Akzeptanz.

An dieser Stelle sei aber auch ausdrücklich vor jeder Form von *Pseudo*partizipation gewarnt: Solange man Partizipation nur als ein Mittel betrachtet, jemanden dazu zu bewegen, das zu tun, was man von ihm erwartet, wird diese Art der Pseudopartizipation nie befriedigende Ergebnisse zeitigen. Wirkliche Partizipation beruht stets auf Respekt.

Dabei darf jedoch nicht vergessen werden, dass die konkrete Gestaltung operativer Massnahmen eine gewichtige Bedeutung für den Erfolg oder Misserfolg eines Projektes besitzt. So können z.B. die Wahl einer ungün-

stigen Abteilung für den Einführungsversuch, die mangelnde Berücksichtigung von Verflechtungen eines Teilsystems mit dem Gesamtsystem oder der ungenügende Einbezug der Vorgesetzten ein Veränderungsprojekt zum Scheitern verurteilen, obwohl eine Beteiligung der Mitarbeiter praktiziert wird.

2.1.2 Fachliche Komplexität der Anwendung – Fachkompetenz der Benutzer

Bisherige, konventionelle Anwendungen von EDV-Systemen befassten sich mit der (Teil-)Automatisierung von gut formalisierbaren Routineaufgaben, dem Massengeschäft, mit vorwiegend einheitlichen Bearbeitungsverfahren. Demgegenüber unterstützen heutige Anwendungen vor allem ein breites Spektrum unterschiedlichster, schwer formalisierbarer und gering strukturierter Tätigkeiten mit individueller Aufgabenabwicklung. Anstelle der Bearbeitung von Routinefällen tritt dabei die Bearbeitung von Klassen von Sachfällen bzw. Einzelfällen in den Vordergrund. Zusätzlich erschwerend wirkt die hohe Veränderungsdynamik qualitativer, das heisst, schlecht formalisierbarer Anforderungen. Es wäre deshalb unrealistisch und verfehlt, von einem Softwareentwickler zu erwarten, dass er diese Komplexität ohne Mitarbeit der Benutzer bewältigen kann. Die Vielfalt moderner Entwicklungstools und technischer Möglichkeiten im allgemeinen erfordern von ihm in seinem Fachgebiet bereits hohe Qualifikationen!

Kasten 9: Beispiel für die Notwendigkeit von Benutzerbeteiligung.

> Eine Regel besagt, dass Optionen in einem Menü nach einem Kriterium (Inhalt, Sachlogik, Arbeitsablauf, Benutzungshäufigkeit, Alphabet) gruppiert sein müssen. Welches Kriterium im konkreten Fall Anwendung findet, kann am besten mit den Benutzern geklärt werden!

Damit ist ein Hauptgrund für die Beteiligung von Benutzern angesprochen: Das fachliche Erfahrungswissen der Benutzer kann durch nichts ersetzt werden! Insbesondere bei der Komplexität der heutigen Anwendungen verhelfen die Beiträge der Benutzer zur Entwicklung eines Systems, das bezüglich Funktionalität und Informationsbestand den Aufgaben der Benutzer und deren Arbeitsstil besser angepasst ist, als wenn es ein Entwickler 'im

stillen Kämmerlein' entwirft. Ausserdem erhält der Entwickler ein grösseres Sicherheitsgefühl. Er muss sich bei Problemen, offenen Fragen, Zweifelsfällen usw. nicht mehr den Kopf zerbrechen, 'wie das ein Benutzer sehen würde', sondern kann die Situation gemeinsam mit den Benutzern klären.

Sicherlich gibt es schon viele Fachbücher, die mit Kriterien, Richtlinien und Regeln dem Softwareentwickler Hilfe bei der Systementwicklung versprechen. Sie bieten ihm in der Regel einen Rahmen, eine Orientierungshilfe und allgemeine Hinweise zur *formalen Gestaltung* der Benutzungsoberfläche. Aufgrund ihres allgemeingültigen Anspruchs müssen sie aber notgedrungen die *inhaltliche Seite* – welche Funktionen soll das System anbieten, welche Informationen sollen wie auf Bildschirm-Masken aufgeteilt werden usw. – ausklammern. Das heisst, der Entwickler wird in seiner konkreten Projektarbeit auf Fragen stossen, die nur in Zusammenarbeit mit Benutzern, nicht aber durch Bücher beantwortet werden können.

Neben der besseren Anpassung der Software an Aufgaben und Unterstützungsbedürfnisse der Fachabteilung gibt es aber auch wirtschaftliche Gründe für den Einbezug von Benutzern. In einem Vergleich zwischen systemorientierten und benutzerorientierten Entwicklungsprojekten zeigten sich Vorteile bezüglich Wartungsaufwand, Kostenüberschreitung und Terminüberschreitung zugunsten einer Benutzerorientierung. Insbesondere die aktive Beteiligung von Benutzern in frühen Phasen der Entwicklung – bei der Problemanalyse und Anforderungsermittlung – kann zur Vermeidung nachträglicher, meist kostspieliger Korrekturen beitragen. Fehler in frühen Phasen bewirken Probleme in späteren Phasen und hohen Korrekturaufwand. Quantitative Analysen haben gezeigt, dass am meisten Fehler in der Problem- und Anforderungsanalyse entstehen, die dann auch den grössten Teil der Fehlerbehebungskosten verursachen.

2.1.3 Gegenseitiges Verständnis

Als grosser Vorteil sowohl für die Benutzer als auch für die Entwickler entpuppt sich der Gewinn an gegenseitigem Verständnis. Durch die Einsicht in die Entwicklungsarbeit begreifen die Benutzer die technischen Schwierigkeiten und Probleme, mit denen sich Entwickler bei der Realisierung von Benutzerwünschen häufig konfrontiert sehen, viel besser und akzeptieren auch eher eine abwehrende Haltung, wenn das Verhältnis von Aufwand zu Ertrag unverhältnismässig hoch ist. Umgekehrt staunt mancher Entwickler, wenn er die Unterschiede zwischen Stellenbeschreibungen, formalen Be-

schreibungen von Arbeitsabläufen usw. und dem realen Arbeitsverhalten durch die Zusammenarbeit mit Benutzern erfährt. Gewöhnlich sind Entwickler dann eher bereit, diese oder jene Funktion oder Änderung – die sie sonst als überflüssig erachtet hätten – noch zu realisieren, da sie deren Bedeutung für die Arbeit der Benutzer einsehen. Es finden auf beiden Seiten viele Aha-Erlebnisse statt!

Darüber hinaus gibt die Zusammenarbeit mit Benutzern dem Entwickler Feedback über die geleistete Arbeit, das über die Bemerkungen seines Vorgesetzten oder Testergebnisse hinausgeht. In vielen Fällen ist es das erste Mal, dass ein Entwickler derart detaillierte Rückmeldungen über Einsatz und Benutzung seines Arbeitsproduktes erhält, was auch sehr motivierend wirken kann.

Kasten 10: Beispiel für einen Benutzer(B)–Entwickler(E)-Dialog.

B: "Ich habe mich schon lange gefragt, warum man bei diesen Masken nur vorwärts-, aber nicht rückwärtsblättern kann. Das müsste doch einfach zu realisieren sein, und wäre für mich schon oft hilfreich gewesen!"

E: "Das liegt nicht an unseren Programmen, sondern an der Datenbank, auf der wir aufbauen, die kann das nicht, und wir müssten umfangreiche und komplizierte ..." (erläutert das Problem noch mehr).

B: "Aha, jetzt verstehe ich das, und ich dachte immer, ..., aber es ist nicht so schlimm, so wichtig ist das Rückwärtsblättern für mich auch wieder nicht, manchmal wäre es eben bequem!"

E: "Bei der nächsten Version der Datenbank, die wir ca. in einem halben Jahr bekommen, wird es dann möglich sein."

2.2 VORAUSSETZUNGEN FÜR BENUTZERBETEILIGUNG

2.2.1 Die erste Voraussetzung: Information

Eine der grundlegenden Voraussetzungen zur benutzerorientierten Planung und Durchführung organisationaler Veränderungen – wie z.B. der Entwicklung und Einführung neuer Anwendungssoftware – ist eine sachgemässe Information aller Betroffenen.

Warum

Fragmente und Details über geplante Projekte sickern – auch bei vertraulicher Behandlung von Dokumenten! – über informelle Kanäle, z.B. bei Pausengesprächen, zufälligen Treffen im Gang ("Hast Du schon gehört ...") usw. in den gesamten Betrieb, wo sie Unruhe erzeugen, Anlass zu Spekulationen geben und Keimzellen für Gerüchte bilden können.

Dies ist ein – vermutlich fast alltägliches – Beispiel für fehlende oder unzureichende Aufklärung über geplante Veränderungen. Gerüchte können aus Kleinigkeiten, unbestätigten Vermutungen, aufgeschnappten Bemerkungen usw. entstehen oder auch bewusst in die Welt gesetzt werden, z.B. um die 'Gegenposition' zu einer Stellungnahme zu provozieren. Ist ein Gerücht allerdings einmal entstanden, so entwickelt es eine Eigendynamik, die nur schwer zu stoppen ist.

Durch sachliche Information der betroffenen Mitarbeiter kann Ungewissheit reduziert und Spekulationen und Gerüchten vorgebeugt werden, wenn sie auch nicht immer ganz vermieden werden können! Verständliche Information von offiziellen Stellen – z.B. Projektleiter – mit Möglichkeiten der Rückfrage trägt zur Klärung offener Fragen bei und ermöglicht den Betroffenen auch eine bessere Abschätzung zukünftiger Entwicklungen; darüber hinaus hilft sie beim Aufbau eines vertrauenswürdigen Klimas im Betrieb.

In Deutschland ist die Information der Betroffenen gesetzlich geregelt; nach § 90 des Betriebsverfassungsgesetzes hat "der Arbeitgeber den Betriebsrat über die Planung ... 2. von technischen Anlagen, 3. von Arbeitsverfahren und Arbeitsabläufen oder 4. der Arbeitsplätze rechtzeitig zu unterrichten und die vorgesehenen Massnahmen insbesondere im Hinblick auf ihre Auswirkungen auf die Art der Arbeit und die Anforderungen an die Arbeitnehmer mit ihm zu beraten ..."

Wer

Wer soll überhaupt informiert werden? Selbstverständlich sind alle von einer Veränderung *direkt* Betroffenen zu informieren. Darüber hinaus kann es sich aber auch als sinnvoll erweisen, *indirekt* betroffene Mitarbeiter und Vorgesetzte anderer Abteilungen vom Vorhaben in Kenntnis zu setzen. Häufig können sich nämlich Veränderungen in den Arbeitsabläufen einer Abteilung auch ungeplant auf andere Abteilungen auswirken ("Seit ihr dieses neue System habt, kommen eure Dokumente viel schneller zu uns rüber; darauf waren wir gar nicht vorbereitet!"). Ausserdem kann es ratsam sein, über betrieblich vorgeschriebene Adressaten (wie Stabsstellen, Kommissionen, Standardisierungswesen usw.) hinaus auch Stellen zu informieren, die einen indirekten Einfluss auf das Projekt ausüben könnten.

Was

Worüber soll informiert werden? Die Mehrheit der Betroffenen interessiert sich in der Regel *nicht* so sehr für technische Aspekte wie Übertragungsraten, Speicherkapazitäten oder Zugriffszeiten von Computersystemen. Eine Ausnahme bilden heute z.T. qualifizierte PC-Benutzer. Informationen sollten deshalb vor allem über die (Hinter-)Gründe und Ziele des Vorhabens, die voraussehbaren Auswirkungen auf die Arbeitssituation und das weitere Vorgehen Klarheit schaffen. Den einzelnen Arbeitnehmer interessiert es zentral, wie sich seine Arbeitssituation, seine Rolle und seine Aufgaben, seine Kooperations- und Kommunikationsbeziehungen verändern werden. Kennt er noch den zeitlichen Horizont und die wichtigsten Schritte des Vorgehens, so kann er sich auch besser darauf einstellen.

Wie

Wie sollen die Informationen den Betroffenen übermittelt werden? Dazu gibt es verschiedene Möglichkeiten, deren Wahl grundsätzlich vom Umfang des Projektes und von der Anzahl der Betroffenen abhängt.

Für *kleinere* Projekte kann

- ein persönliches Gespräch,
- eine Orientierung am Anschlagbrett der Abteilung,
- ein persönliches Anschreiben,
- ein Zirkulationsschreiben in der Abteilung,
- eine Mitteilung in der Mailbox oder

- eine Orientierung durch den Vorgesetzten bei einer Sitzung

genügen. Für *grössere* Vorhaben lohnt es sich, einen dementsprechend grösseren Aufwand in die Information zu investieren, denn: Nachlässigkeiten und 'Sünden' zu Beginn eines Projektes führen zu einem schlechten Image, das sich nachträglich nur mit viel Mühe wieder aufbessern lässt!

Häufig benutzte Möglichkeiten sind

- Haus-, Betriebszeitung,
- Abteilungs-, Betriebsversammlung,
- Seminare, Workshops,
- Projektnachrichten,
- Projektbroschüre,
- Besuch eines anderen Betriebes oder eines Herstellers.

Letzteres ist eine viel zu selten praktizierte Möglichkeit. Indem man mit den Mitarbeitern einen Betrieb besucht, der eine ähnliche Veränderung schon durchgeführt hat, gibt man ihnen Gelegenheit, ein System im Alltagsbetrieb zu sehen und die Erfahrungen der Benutzer kennenzulernen.

Grundsätzlich ist darauf zu achten, dass die Information die Angesprochenen auch erreicht ("Was? Woher hast Du das? Ach, das steht in der Info-Box im Untermenü von 'Utilities'? Da schau' ich selten rein!"). In der Praxis werden deshalb auch meist mehrere der aufgelisteten Möglichkeiten kombiniert zur Verbreitung von Information genutzt. Für direkt Betroffene bzw. Beteiligte empfiehlt sich die Schaffung eines Verteilungsnetzes!

Wann

Wann soll die Information erfolgen? Diese Frage wird meist mit 'frühzeitig' beantwortet, weil in der Praxis die Information häufig zu spät erfolgt, die Zeitspanne aber auch schwerlich in Wochen oder Monaten angegeben werden kann. Sie hängt wiederum mit der Grösse des Vorhabens und den damit verbundenen betrieblichen Umwälzungen zusammen. Bei grösseren Projekten sind sechs Monate nach Projektetablierung eine untere Grenze, ein Jahr wird von den meisten Betroffenen als frühzeitig genug erachtet. Wenn sich eine Projektskizze konkretisiert und realisierbare Formen annimmt, wenn zuverlässige Informationen abgegeben werden können, ist meist auch bald der Zeitpunkt für eine Erstinformation gekommen. Dabei darf es aber nicht bleiben. Fortlaufende Information mit zunehmendem De-

taillierungsgrad soll die Betroffenen über den Stand der Arbeiten unterrichten! Wenn ein Projekt einmal begonnen wurde, ist es sinnvoll, die Betroffenen im Sinne eines Feedbacks periodisch über den Stand der Dinge, erzielte Erfolge ("Teilprojekt 'Automatische Agendierung' ist abgeschlossen. Die Anwendung wurde getestet und wird termingemäss verfügbar sein") wie auch Schwierigkeiten ("Bei der Verlegung der Kabel für das Netzwerk sind bauliche Probleme aufgetreten. Dadurch verzögert sich die Einführung des Systems um ca. drei Wochen") zu informieren.

Prinzipiell sollte bei jeder Information auch eine Person bzw. Kontaktstelle angegeben werden, die bei Unklarheiten oder weiterem Interesse der Benutzer Auskunft geben kann. Gegebenenfalls können auf diesem Wege auch schon erste Meinungen, Stellungnahmen zum Projekt eingeholt werden. Damit wird die Gefahr eines einseitigen Monologes vermieden und zum Dialog aufgefordert. Bezüglich Darstellung und Sprache ist auf Verständlichkeit der Information zu achten. Hinsichtlich der Menge an Information bzw. des Umfangs ist gerade bei Abgabe schriftlicher Informationen ein Mittelweg zwischen zu wenig Information und einer Überflutung mit Details zu finden, da sonst die Mitteilungen unverständlich bleiben oder nicht gelesen werden.

Letztlich bleibt es den Projektverantwortlichen überlassen, aufgrund von Erfahrung und Kenntnis der betrieblichen Gepflogenheiten aus den aufgelisteten Möglichkeiten ein der konkreten Situation angepasstes Informationskonzept zu entwickeln. Dabei sollte man allerdings nicht nur in 'Traditionen' verharren (" bei uns war das immer so ..."), sondern auch den Mut aufbringen, Neues zu wagen.

2.2.2 Die zweite Voraussetzung: Ausbildung

Als zweite wesentliche Voraussetzung für einen erfolgreichen Einbezug von Benutzern ist eine *angemessene Ausbildung* zu nennen. Es gibt zwei Qualifikationsarten: (1) die Qualifikation zur Teilnahme am Entwicklungsprozess und (2) die Qualifikation zur Benutzung des EDV-Systems. Im folgenden Abschnitt wird *nicht* über die Schulung aller Benutzer zur operationellen Handhabung des fertigen Systems gesprochen (siehe dazu Teil B, Kapitel 4.15), sondern speziell über die Qualifizierung der in die Entwicklung einbezogenen Benutzer(-vertreter).

Warum

Forderungen nach Benutzerbeteiligung wird manchmal mit dem allgemeinen Argument begegnet, die Benutzer hätten zuwenig Kenntnisse – z.B. in den Bereichen Organisation und Informatik – und könnten deshalb keine konstruktiven Beiträge leisten. Diesem Umstand kann durch eine entsprechende *Qualifizierung* der Benutzer, die sie *zu kompetenter Mitarbeit* befähigt, Rechnung getragen werden. Allerdings zeigen sich häufig auch bei Entwicklern Wissenslücken verschiedenster Art, weshalb im folgenden von *komplementärer Ausbildung* die Rede sein wird. *Die Qualifikationen von Benutzern und Entwicklern müssen komplementär erhöht und einander angenähert werden*, was auch zur Vermeidung bzw. Reduktion von Verständigungsproblemen führt.

Wer

Wie bereits angesprochen sollen nicht nur die beteiligten Benutzer, sondern auch die Entwickler qualifiziert werden. Grundsätzlich gilt, dass alle im Projekt involvierten Personen (Benutzer, Entwickler, Organisatoren, Koordinatoren usw.) – abhängig von ihrem Wissen und ihrer Erfahrung – eine Qualifikation erfahren sollen, die ihnen eine gemeinsame Basis vermittelt. Kombiniert mit dem jeweiligen Spezialwissen wird dadurch eine konstruktive Zusammenarbeit verschiedener Spezialisten gefördert.

Was

Sicher ist es für die Zusammenarbeit hilfreich, wenn den *Benutzern* ein *Grundlagenwissen in EDV* und *Kenntnisse* über die *Möglichkeiten und Grenzen* der aktuell verfügbaren *Technik* (z.B. "Was ist ein LAN? Wie sehen neuartige Benutzungsoberflächen mit windows, pull-down-Menüs aus" usw.) vermittelt wird. Genauso wichtig ist es aber, dass die *Entwickler* zumindest *Grundbegriffe des fachlichen Anwendungsbereiches* kennen. Wie sonst soll ein Entwickler eine Anwendung für eine Bank oder eine Versicherung erstellen, wenn er keine bankfachlichen oder versicherungstechnischen Kenntnisse hat? Um Missverständnisse zu vermeiden, sei aber festgehalten: Weder sollen Sachbearbeiter (als Benutzer) in einer Art 'Schnellbleiche' zu Programmierern noch sollen Entwickler zu Sachbearbeitern ausgebildet werden! Weiter hat es sich als sinnvoll erwiesen, wenn Benutzer und Entwickler Kenntnisse in *Organisation* und *(Software-)Ergonomie* erlangen. Darüber hinaus ist es speziell für Personen mit Vorgesetztenfunktion – z.B. Projektleiter und deren Stellvertreter – wichtig, dass sie Kenntnisse in *Gesprächsführung und Gruppenmoderation* erhalten. Da der

Arbeit in Teams grosse Bedeutung zukommt, ist eventuellen Qualifikationsdefiziten der Projektbeteiligten durch ein *Training sozialer Fertigkeiten und Kompetenzen* zu begegnen. Abhängig von den jeweiligen Aufgaben können sich zusätzliche spezifische Qualifikationserfordernisse (z.B. Präsentationstechnik, juristische Aspekte usw.) ergeben.

Wie

Prinzipiell ist der Erwerb der angesprochenen Inhalte in den üblichen betriebsinternen und -externen Kursen möglich. Für spezielle Inhalte – z.B. Gruppenmoderation, Gruppenarbeit – kann es sich jedoch als sinnvoll erweisen, einen auf die Projektbeteiligten und ihre konkrete Situation massgeschneiderten Kurs, Workshop zu organisieren. Meist sind es externe Experten, welche derartige Seminare durchführen. Selbststudium von Büchern und computergestütztes Lernen kann für die Aneignung bzw. Vertiefung von Grundwissen in einem Fachgebiet nützlich sein, darf aber interaktives Lernen in Gruppen nicht ersetzen!

Wann

Eigentlich, so möchte man meinen, ist es selbstverständlich, dass die Qualifizierung *vor* Beginn der Zusammenarbeit stattfindet. Praxiserfahrungen zeigen aber immer wieder, dass erst bei auftauchenden Problemen ad hoc improvisierte Massnahmen getroffen werden, was sich auf den Projektablauf störend auswirkt! Deshalb sei hier betont: *Vor* Beginn der eigentlichen Projektarbeit sollten Qualifikationsdefizite ermittelt und durch entsprechende Massnahmen ausgeglichen werden! Dies gilt im besonderen für die Anwendung neuer Entwicklungswerkzeuge, deren Erlernen und Beherrschung durch die Entwickler nicht in den Ablauf des Projektes verschoben werden sollte. Ebenso ist daran zu denken, dass Wissen und Kenntnisse in relativ kurzer Zeit erworben werden können, soziale Fertigkeiten aber über längere Zeit trainiert werden müssen, bis sie auch wirklich eingespielt sind.

2.3 KONZEPT DER BENUTZERBETEILIGUNG

Wie in einem Projekt Benutzerbeteiligung praktisch realisiert wird, hängt von einer Reihe von *Rahmenbedingungen* ab (siehe Teil B, Kapitel 2). Es sind dies z.B.:

(1) *Die Anzahl zukünftiger Benutzer:*

Bei nur 12 Benutzern ist eine *direkte* Beteiligung *aller* Benutzer noch gut möglich, mit steigender Benutzerzahl wird sich das Schwergewicht auf eine *indirekte* Beteiligung – die Mitwirkung von Benutzer*vertretern* – verschieben.

(2) *Der Projekttyp:*

Ob betriebsintern für eine Abteilung eine Individuallösung oder in einem Software-Haus ein Standard-Handelspaket entwickelt wird, bestimmt die Möglichkeiten des Zugangs zu Benutzern.

(3) *Die Komplexität der Anwendung:*

Je unstrukturierter, offener und komplexer die Aufgaben sind, desto unverzichtbarer ist eine aktive, kontinuierliche Mitarbeit von Benutzern.

Dies sind nur einige Rahmenbedingungen, die man berücksichtigen muss, bevor man das konkrete Beteiligungsmodell anhand der folgenden Kernelemente konzipiert.

2.3.1 Kernelemente eines Beteiligungsmodells

Form der Beteiligung

Wie erfolgt die organisatorische Eingliederung der Benutzer, wie wird die Zusammenarbeit zwischen Benutzern und Entwicklern geregelt: Aufbauorganisation, Aufgaben, Kompetenzen, Entscheidungsfindung?

Bei der Entwicklung *betriebsinterner* Anwendungen wird häufig ein Modell angewendet, das aus einem Lenkungsausschuss, einer Projektgruppe und Arbeitsgruppen besteht (siehe dazu Teil B, Kapitel 2). Bei der Entwicklung von Standardsoftware sind derartige Organisationsformen allerdings kaum möglich. Hier bietet sich z.B. die Bildung einer User-Group an, die sich in regelmässigen Abständen (mindestens einmal jährlich) mit den Entwicklern trifft, um z.B. Erweiterungen für neue Versionen eines Produktes zu diskutieren. Vielfach ist es in der Praxis auch der Fall, dass Pilotkunden Anregungen für neue Versionen liefern und diese dann – bevor sie in den Verkauf gelangen – einem Praxistest unterziehen. Viel zu wenig wird die Möglichkeit benutzt, über Verkäufer, Händler, Hotline oder Support ein Feedback über das Produkt zu erhalten. Eine systematisch geführte Dokumentation von Supportleistungen kann nicht nur auf Fehler in der Software hinweisen, sondern auch allgemeine Probleme der Benutzer im Umgang mit

der Software aufdecken und Verbesserungen anregen. Ferner besteht auch die Möglichkeit, über eine dem Produkt beigelegte Registrierkarte mit Fragebogen den Kontakt zu den ansonsten unbekannt bleibenden Benutzern herzustellen. Eine weitere Möglichkeit sind Umfragen in einschlägigen Fachzeitschriften.

Grad bzw. Ausprägung der Beteiligung

In welchem Ausmass werden die Benutzer in den Entwicklungsprozess einbezogen? Beschränkt sich die Beteiligung auf einen Informationsaustausch, werden die Benutzer(-vertreter) an Entscheidungen beteiligt und übernehmen sie auch Verantwortung und/oder wirken auch aktiv an der Gestaltungsarbeit (z.B. Maskendesign mit Hilfe eines Maskengenerators, bei der Datenmodellierung usw.) mit?

In einer Befragung von 157 Entwicklern, Benutzern und Führungskräften hat sich dazu ein interessanter Trend gezeigt: Alle drei Gruppen haben sich mehrheitlich dafür ausgesprochen, dass Benutzer nicht nur Informationen über ihre Aufgaben und Arbeitsabläufe liefern und Wünsche anbringen sollten, sondern vermehrt (Mit-)Entscheidungskompetenzen erhalten und Verantwortung übernehmen sollten. Teilweise – je nach Inhalt des Projektes – sollten Benutzer auch Teilaufgaben der Realisierung übernehmen, was wir auch in mehreren Projekten beobachten konnten! Dies ist ein Hinweis darauf, dass die bisher eher passive Rolle der Benutzer in einigen Unternehmen der Vergangenheit angehört und sich in Richtung einer Partnerschaft im Sinne der Kooperation von Experten verschiedener Gebiete entwickelt.

Inhalt der Beteiligung

An welchen Inhalten der Softwareentwicklung werden die Benutzer beteiligt?

In der Regel erstreckt sich die Mitarbeit auf Aufgaben, die mit der Systemfunktionalität, den Daten bzw. dem Datenkonzept und der Benutzungsoberfläche (Informationsdarstellung, Informationscodierung, Maskendesign, Dialogablauf) zu tun haben. Vorausgesetzt ist dabei, dass bei grösseren organisatorischen Veränderungen Fragen der Aufbau- und Ablauforganisation *vorher* mit den Benutzern geklärt wurden! Es wäre unsinnig, wenn Entwickler organisatorische Abläufe allein – ohne Benutzer – entwerfen und anschliessend für das Design von Masken die Benutzer einbeziehen!

Repräsentativität der Beteiligten

Werden alle, speziell ausgewählte oder repräsentative Vertreter der Benutzer einbezogen?

Eines der wichtigsten Kriterien der Benutzerbeteiligung ist die Repräsentativität! Dies ist eine Frage der 'richtigen' Zusammenstellung der Benutzerstichprobe, die in die Entwicklung einbezogen werden soll. In den meisten Fällen wird es kaum möglich sein, alle Benutzer kontinuierlich in die Entwicklungsarbeit zu integrieren. Die beteiligten Benutzer müssen aber repräsentativ für die ganze zukünftige Benutzerschaft sein. Je nach Projekt sind bei der Auswahl der beteiligten Benutzer folgende – unter Umständen auch noch andere! – Merkmale zu berücksichtigen, um Repräsentativität zu erreichen:

- Abteilungen, Fachbereiche,
- Aufgaben: Benutzer mit unterschiedlichen Aufgaben am System wie Eingabe, Bearbeitung, Abfrage,
- Benutzertypen: Benutzer mit unterschiedlicher Erfahrung und Benutzungshäufigkeit; oft können gerade Neulinge eine neue Sichtweise beisteuern; tägliche Benutzer haben andere Bedürfnisse als Benutzer, die einmal im Monat eine Abfrage durchführen,
- Hierarchiestufen,
- Sprachgruppen.

Methoden für Beteiligung

Welche Methoden (Instrumente, Werkzeuge, Tools) werden zu welchem Zeitpunkt im Entwicklungsprozess wofür (Analyse, Bewertung, Darstellung, Produktion, Test) eingesetzt?

Dies ist die Grundfrage der methodischen Gestaltung des Entwicklungsprozesses; sie sollte *vor* Projektbeginn beantwortet werden können! Das heisst, es ist zu überlegen, welche Methoden eingesetzt werden, ob sie vorhanden sind bzw. gekauft, gelernt, angepasst oder entwickelt werden müssen und ob sie eine spezielle Schulung verlangen oder das Vorhandensein bestimmter Geräte wie z.B. Videokamera und Rekorder bedingen! Müssen sie während des Projektes entwickelt oder gekauft und geschult werden, so ergeben sich unliebsame Verzögerungen im Ablauf und/oder improvisierte Methoden! Moderne Entwicklungswerkzeuge können vor allem in der Realisierungsphase den Programmieraufwand beträchtlich reduzieren. In der

Regel verlangen sie aber eine klar bestimmte Vorgehensweise, welche die Modellbildung und Entwurfsstrategie des Entwicklers – und damit auch seinen Handlungsspielraum – bestimmt. Unter Umständen können sie dadurch auch eine Benutzerbeteiligung erschweren.

Zeitpunkt der Beteiligung

Zu welchem Zeitpunkt, in welcher Phase im Projektablauf werden die Benutzer einbezogen?

Grundsätzlich sollen die Benutzer – abgesehen von den kontinuierlich mitarbeitenden Benutzervertretern – an *den* Arbeitsschritten beteiligt werden, wo sie auch einen konstruktiven Beitrag leisten können bzw. zu den Zeitpunkten, an denen Entscheidungen gefällt werden, die für die Benutzer bedeutsam sind. Wesentlich ist, dass der Einbezug *frühzeitig, das heisst schon bei der Problemanalyse, der Anforderungsermittlung und dem Grobkonzept erfolgt; es macht keinen Sinn, die Benutzer von diesen Aktivitäten auszuschliessen und sie dann bei der Detailspezifikation beizuziehen!* Normalerweise werden Benutzer bei der EDV-technischen Realisierung nicht einbezogen, da ihnen die fachlichen Kenntnisse fehlen. Bei Tests des – mehr oder weniger – fertigen Systems hingegen kann die Mitarbeit der Benutzer wichtig sein, wenn sich die Tests nicht nur auf programmiertechnische Aspekte beziehen, sondern auch den Umgang mit dem System enthalten. Letztlich wird die Beantwortung der Frage von der gewählten Beteiligungsform und den zur Diskussion stehenden Inhalten abhängig sein.

2.3.2 Evaluative Beteiligung

Die evaluative Beteiligung geht auf Mambrey und Oppermann (1983) zurück. Die Benutzer werden dabei zu definierten Zeitpunkten im Projekt – bei Vorliegen eines (Zwischen-)Ergebnisses, dessen Annahme oder Ablehnung über weitere Schritte entscheidet – zur Beurteilung und Bewertung von Vorschlägen, Prototypen, Systemmodulen einbezogen, die von den Entwicklern erarbeitet wurden. Diese eher passive Form der Beteiligung kann in unterschiedlicher Weise erfolgen (siehe Kasten 11).

Als *Vorteile evaluativer Beteiligung* sind zu nennen:

+ Die Entwickler erhalten ein Feedback und Input für die weitere Entwicklung, damit auch grössere Sicherheit, auf dem richtigen Weg zu sein.

+ Je nach Methode – z.B. bei Fragebogen – ergibt sich die Möglichkeit, eine grosse Zahl von zukünftigen Benutzern anzusprechen und damit ein repräsentatives Bild zu erhalten.

Kasten 11: Drei Beispiele für eine evaluative Beteiligung zur Beurteilung der Maskengestaltung.

(1) Die Vorschläge werden den Benutzern – auf Papier oder per EDV-Netz auf Terminal, PC – übermittelt. Mit einem Fragebogen wird die Beurteilung nach bestimmten Merkmalen erfragt und Verbesserungsvorschläge eingeholt ("Zeichnen Sie bitte selbst von Ihnen gewünschte Änderungen ein!"). Häufig werden dabei Alternativ-Vorschläge angeboten: "Welche Maske – A oder B – finden Sie besser?"

(2) In einem speziell einberufenen Workshop oder z.B. an einem Abteilungsmeeting werden die Vorschläge von den Entwicklern vorgestellt und mit den Benutzern diskutiert.

(3) Die Masken werden in einem Test mit einem Teil der Benutzer anhand realistischer Aufgaben geprüft. Dabei werden objektive Daten, z.B. wie schnell findet ein Benutzer die Information X auf Maske Y, evtl. im Vergleich mit Maske Z gemessen. Zusätzlich lassen sich subjektive Daten, z.B. Benutzerurteil, Einschätzung der Beanspruchung usw., erheben.

Als *Nachteile* sind zu werten:

– Die Beteiligung ist nur punktuell (zu bestimmten Zeitpunkten) und ausschnitthaft (zu bestimmten Aspekten). Aufgrund des organisatorischen Aufwandes wird eine derartige Beteiligung kaum mehr als zwei bis drei Mal in einem Projekt stattfinden. Die Benutzer können zeitlich und inhaltlich bedingte Zusammenhänge schwer erkennen; wichtige Bereiche werden ausgeklammert.

– Die Ergebnisse der Beteiligung 'verschwinden' bei den Entwicklern, man weiss nicht genau, was damit geschieht, und Umsetzungen werden erst viel später sichtbar und sind kaum mehr auf eigene Anregungen zurückzuführen ("Von meinen Vorschlägen kann ich nichts mehr

wiedererkennen; haben die das überhaupt zur Kenntnis genommen?"). Entsprechend ist die Motivation – und damit verbunden die Wahrscheinlichkeit kreativer Inputs – der Benutzer geringer, als wenn sie kontinuierlich in einen direkten interaktiven Austausch eingebunden sind.

2.3.3 Prozessuale Beteiligung

Die Benutzer bzw. ihre Vertreter arbeiten zu einem Teil ihrer Arbeitszeit kontinuierlich im Projekt mit und übernehmen dabei Teilaufgaben. Sie können nicht nur Stellung zu Vorgegebenem nehmen, sondern selbst bzw. in Zusammenarbeit mit Entwicklern Vorschläge entwerfen und dabei ihre Erfahrung konstruktiv umsetzen. Das Hauptgewicht dieser Art der Beteiligung liegt also auf aktiver, kontinuierlicher Mitarbeit bei der Entwicklung.

Kasten 12: Zwei Beispiele für eine prozessuale Beteiligung.

(1) Zwei Benutzer arbeiten zu 50% ihrer Arbeitszeit in der Projektgruppe. Sie wirken aktiv bei der Erarbeitung von Konzepten zum Funktionsumfang, zum Informationsgehalt und zur Benutzungsoberfläche mit; ausserdem ist es ihre Aufgabe, den Kontakt zu den übrigen 40 Benutzern sicherzustellen (Abgabe von Informationen, Einholen von Feedback, Organisation und Durchführung von Benutzer-Workshops).

(2) Vier Benutzer und ein Programmierer arbeiten in einer speziellen Arbeitsgruppe, die von der Projektgruppe den Auftrag hat, Masken, Formulare und Listen zu entwerfen und zu testen. Eine andere Arbeitsgruppe mit zwei Programmierern und einem Benutzer entwirft ein Konzept für die Benutzungsoberfläche und erstellt Prototypen.

Die zwei Beispiele in Kasten 12 zeigen, dass es auch für prozessuale Beteiligung unterschiedliche Organisationsformen geben kann.

Vorteile:

+ Durch die kontinuierliche Mitarbeit entwickeln die Benutzer(-vertreter) Einsicht in Zusammenhänge, was konstruktive, realistische Vorschläge und gegenseitiges Verständnis fördert.

+ Die Möglichkeit, von Grund auf eigene Vorschläge zu entwickeln und mit anderen Beteiligten zu diskutieren, wirkt motivierend.
+ Die Delegation von Teilaufgaben – wie z.B. Maskendesign, Kontakt mit übrigen Benutzern usw. – entlastet die Entwickler. Sie können sich vermehrt fachspezifischen Aufgaben (wie programmiertechnische Fragen, Schnittstellen zu anderen Systemen usw.) zuwenden.

Nachteile:

– Da die Benutzervertreter nur einen kleinen Teil aller zukünftigen Benutzer bilden, ist auch nicht immer gewährleistet, dass die übrigen Benutzer mit ihrer Arbeit bzw. Meinung übereinstimmen. Deshalb ist der ständige Kontakt zur gesamten Benutzerschaft wichtig!

Prozessuale und evaluative Beteiligung schliessen sich nicht aus, sondern lassen sich kombinieren: Die kontinuierliche, aktive Mitarbeit einiger Benutzervertreter wird zu definierten Zeitpunkten im Projekt durch die Stellungnahme aller oder zumindest eines repräsentativen Teils der Benutzer zu entscheidenden Fragen ergänzt!

Kasten 13: Gefahren bei indirekter Beteiligung: ein Beispiel.

Drei *Benutzervertreter* sammeln die Meinungen ihrer Arbeitskollegen zum vorgeschlagenen Maskendesign und übergeben sie einem *Koordinator;* dieser bereitet die 150 schriftlichen Beurteilungen auf, verdichtet sie, lässt – seiner Meinung nach – Unwichtiges weg und ergänzt Fehlendes; seine Interpretation des Haupttrends fügt er als Einleitung zum Bericht bei, den er dem *Projektleiter* übermittelt. Dieser liest vor allem die Einleitung sowie zusätzlich noch einige Abschnitte diagonal durch und übermittelt – befriedigt vom Gesamtergebnis in der Einleitung – den *Programmierern* einzelne Änderungshinweise. Nur schon ein grober Vergleich der originalen Benutzeraussagen mit den schliesslich getroffenen Änderungen ergab aber wesentliche qualitative und quantitative Diskrepanzen!

2.3.4 Direkte versus indirekte Beteiligung

Eine andere, grobe Unterscheidungsdimension betrifft die Frage, ob die Benutzer *direkt* (z.B. alle betroffenen Benutzer in einem Workshop mit

Entwicklern) oder *indirekt* (über Benutzervertreter, Koordinatoren, Betriebsrat usw.) in die Entwicklung einbezogen werden. Anders betrachtet geht es um den Weg, den Informationen vom Benutzer bis zum Entwickler zurücklegen müssen. Wie bei allen Informationsflüssen, die über mehrere Stellen verlaufen, besteht die Gefahr, dass die ursprünglichen Informationen gefiltert und verzerrt am Bestimmungsort ankommen. Aus zeitlichen, organisatorischen und ökonomischen Gründen bietet sich häufig der *indirekte Einbezug* an.

Die Auswahl der Benutzervertreter bestimmt massgeblich das Ergebnis der Mitarbeit; werden unmotivierte, schlecht qualifizierte, zur Zeit in der Fachabteilung gerade abkömmliche Benutzer beigezogen, so sind Probleme vorprogrammiert! Bei der Auswahl der Benutzervertreter ist deshalb – neben der Repräsentativität – sorgfältig auf deren Motivation, Qualifikation und soziale, kommunikative Fertigkeiten zu achten!

Die Möglichkeiten zur Benutzerbeteiligung sind vielfältig, aber es gibt kein Patentrezept. Vielmehr liegt es an den Verantwortlichen, ein auf ihr Projekt und ihren Betrieb massgeschneidertes Modell der Beteiligung zu konzipieren!

2.4 PRAXISPROBLEME BEI BENUTZERBETEILIGUNG

Es sei nicht verschwiegen: Auch bei noch so guter Planung können bei der Beteiligung von Benutzern Schwierigkeiten im Entwicklungsprozess auftreten. Nachfolgend sind deshalb einige der am häufigsten auftauchenden Probleme und mögliche Massnahmen zu ihrer Vermeidung genannt.

2.4.1 Doppelbelastung

Wenn ein Benutzer neben seinen täglichen Aufgaben *zusätzlich* noch als Benutzervertreter im Projekt Aufgaben übernehmen muss, so führt dies in der Regel zu einer Überforderung. Dies kann zur Folge haben, dass sein Engagement für das Projekt zunehmend sinkt, die Mitarbeit nur noch als Pflicht betrachtet und der Aufwand dafür minimal gehalten wird. Eine derartige Pflicht-Beteiligung ist für beide Seiten nicht nur nutzlos, sondern auch frustrierend! Deshalb ist es unbedingt notwendig, *Benutzervertreter* in Absprache mit ihrem jeweiligen Vorgesetzten offiziell zu einem gewissen Teil ihrer Arbeitszeit *von ihren täglichen Aufgaben zu entlasten.* Für einen Projektleiter erhöht sich die Gewissheit, dass er auf aktive Mitarbeit der Benutzervertreter zählen kann, da sie ja speziell dafür freigestellt wurden.

2.4.2 Fluktuation

Gerade bei länger dauernden Projekten kann die Fluktuation von Projektbeteiligten zu einem Problem werden, da mit den Personen ja nicht nur Wissen und Erfahrung verlorengehen, sondern auch eingespieltes Verhalten und zwischenmenschliche Beziehungen sich verändern. Sicherlich kann dieses Problem nicht restlos aus der Welt geschafft werden. Um einem ständigen Wechsel der Beteiligten vorzubeugen, sollten aber die *Bereitschaft und die objektiven Möglichkeiten bzw. Hindernisse* (bevorstehende Versetzung, Auslandaufenthalt, Weiterbildung usw.) für die Mitarbeit bis Projektende frühzeitig mit den Betreffenden *abgeklärt* werden. In Ausnahmefällen kann ein Personenwechsel auch Vorteile mit sich bringen (neue Ideen, weniger 'Betriebsblindheit' usw.).

2.4.3 Hierarchiegefälle

Bei hierarchisch gemischt zusammengesetzten Gremien (Projekt-, Arbeitsgruppen) können sich Probleme ergeben, wenn das alltägliche Rollenverhalten in der Gruppe beibehalten wird. Wenn Vorgesetzte dominieren und Unterstellte den Mut für Widerspruch und Kritik nicht aufbringen (dürfen?!), sind die Voraussetzungen für eine fruchtbare Zusammenarbeit, die auch unterschiedliche Meinungen zulässt, sehr schlecht. Neben der gemeinsamen Festlegung von *Regeln für Diskussionen und Verhalten in der Gruppe* kann sich ein *Training sozialer Fertigkeiten* für die Schaffung eines guten Gruppenklimas als nützlich erweisen.

2.4.4 Informationstransfer Beteiligte <–> Betroffene

Gerade bei länger dauernden Projekten kann sich die Gefahr ergeben, dass im Laufe der Zeit der Informationsfluss von den direkt Beteiligten (z.B. den Benutzervertretern) zu den übrigen Benutzern abnimmt oder gar versiegt. Die Gründe dafür können vielfältiger Natur sein, wie z.B. Nachlässigkeit, Vergessen, Überlastung, bewusste Zurückhaltung usw. Meist nimmt dabei aber auch der Informationsfluss in umgekehrter Richtung – von den Betroffenen zu den Beteiligten – ab, was dem Projektfortschritt abträglich ist. In Extremfällen kann es sogar zu einer Abschottung der Projektbeteiligten von den übrigen Betroffenen kommen, was einerseits zu sozialen Spannungen bzw. Konflikten und andererseits zu einer einseitigen Lösungsfindung mit entsprechend eingeschränkter Akzeptanz führen kann. Um derartige Entwicklungen zu vermeiden, ist die *Institutionalisierung des Informationsaustausches,* z.B. in Form *regelmässiger Meetings* oder dem

Aufbau eines *Informationsverteilungsnetzes* (siehe auch Teil A, Kapitel 2.2.1), erforderlich.

2.4.5 Fixierung, Struktur-Konservatismus

Ein allgemeines Problem besteht häufig darin, dass Projektgruppen sich allzu sehr an bestehenden Strukturen oder organisationalen Abläufen orientieren und der innovative Gehalt einer Lösung entsprechend bescheiden ausfällt. Oft passiert es auch, dass eine Gruppe sich zu früh auf eine erste sich ergebende oder die nächstliegende Lösung konzentriert bzw. fixiert, ohne Alternativen wirklich geprüft zu haben. Dieser Gefahr kann durch den Einsatz von Kreativitätstechniken und die Gewährung eines auch für anfänglich abwegig erscheinende Ideen *offenen Diskussionsklimas* in der Gruppe begegnet werden. Auch die Einführung einer 'Spielverderber'-Rolle im Sinne eines Advocatus Diaboli kann eine vorschnelle Einigung auf eine suboptimale Lösung verhindern helfen.

2.4.6 Motivationserhalt

Grosse, längerfristige Projekte haben häufig den Nachteil, dass Ergebnisse der Projektarbeit erst nach einigen Monaten oder gar Jahren sichtbar werden. Ein derart verzögertes Feedback dämpft aber die Motivation der Benutzer(-vertreter) und bewirkt nachlassendes Engagement und Initiative. Deshalb ist es sinnvoll, grosse Projekte in *kleinere, überschaubare Teilprojekte* aufzugliedern, deren Abwicklung den Beteiligten *sichtbaren Fortschritt und Erfolgserlebnisse* bringt.

2.4.7 Vorstellungsvermögen der Benutzer bzw. ihrer Vertreter

Die in der Softwareentwicklung häufig eingesetzten Methoden (formale Spezifikationen, Diagramme, Struktogramme, Pflichtenhefte usw.) überfordern in der Regel das Vorstellungsvermögen von Benutzervertretern. Anhand dieser Unterlagen können sie sich kaum eine gedankliche Vorstellung des zukünftigen Systems bilden. Im Gegensatz dazu sind *Prototypen anschauliche Modelle* für die Beurteilung und Ableitung von Verbesserungs- bzw. Änderungsvorschlägen seitens der Benutzer.

2.4.8 Verständigung zwischen Benutzern und Entwicklern

Eines der Hauptprobleme ist die Kommunikation zwischen Benutzern und Entwicklern. Vielfach bereitet es den beiden Personengruppen einige

Mühe, das Gemeinte dem jeweils anderen auch verständlich zu übermitteln. Unverständnis bzw. Missverständnisse sind die Folgen, die sich – bei anhaltenden Schwierigkeiten – zu Rückzug und gegenseitiger Ablehnung der beiden Gruppen ausweiten können. Manchmal stellt sich auch nachträglich heraus, dass unter dem gleichen Begriff Verschiedenes verstanden wurde. Verständigungsprobleme resultieren nicht nur aus dem unterschiedlichen Sprachgebrauch. Vielmehr sind sie Ausdruck für unterschiedliche

(1) Ausbildungen,

(2) Betrachtungsperspektiven des zu entwickelnden Systems und

(3) Denkstile bzw. Logik beim Problemlösen.

Entwickler denken meist in abstrakt-analytischen, systemanforderungsbezogenen Kategorien, Benutzer hingegen in aufgaben- und anwendungsbezogenen Begriffen.

Kasten 14: Beispiel einer unzureichenden Benutzer(B)–Entwickler(E)-Verständigung.

B: "Ich muss neue Bücher eingeben, verschlüsseln und ..."
E: "Ja, Schlüsselmerkmale müssen für Selektionen definiert werden."

B: "Ääh ... ich meine thematisch ...ja, und katalogisieren, Schlüssel ändern..."
E: "Mmh, nicht so einfach bei dieser relationalen DB."

B: "Ja...? Dann nach Autoren und Jahrgang sortieren, auswählen und verschiedene Listen erstellen und ausdrucken."
E: "Ja, dann machen wir ein Menü 'Erfassung, Mutation, Anzeigen', sauber getrennt, damit bei 'Anzeigen' auch nichts geändert werden kann."

B: "Aber ich muss doch Angezeigtes ändern können..."
E: "Ja,ja, dann ein Menü 'Selektion', wo Sie auch nach Autor oder Jahrgang sortieren können."

B: "... und Jahrgang, nach beidem!"
E: "Das geht bei dieser DB nicht! Wieviele Records haben Sie überhaupt?"

B: " Records ...?"
usw.

Derartige Verständigungsprobleme können durch *komplementäre Ausbildung* (siehe Teil A, Kapitel 2.2.2) und die Schaffung eines *gemeinsamen Begriffsglossars* verringert werden. Sie entschärfen sich auch zunehmend, da jüngere Benutzergenerationen schon in der Schulausbildung EDV-Kenntnisse erhalten. Weniger Kommunikationsprobleme treten auf, wenn Entwickler eine fachspezifische Vorbildung (z.B. kaufmännische Grundausbildung) besitzen!

2.4.9 Abwehr

Ein Problem kann sich dadurch ergeben, dass Benutzer zur Einbringung von Vorschlägen bzw. Anforderungen aufgefordert werden, eintreffende Vorschläge dann aber von Entwicklerseite pauschal ("... das geht nicht")

abgewehrt werden. Ob diese Abwehr nun gerechtfertigt ist (Aufwand, technische Restriktionen) oder nicht: Die Motivation der Benutzer wird dadurch zusehends reduziert und kippt möglicherweise in Resignation oder Aggression über. Es ist deshalb wichtig, schon zu Projektbeginn die *Erwartungen* auf einem *realistischen Niveau* zu halten und *frühzeitig* über *mögliche Restriktionen* zu *diskutieren*. Die Strategie, die Benutzer durch grosse Versprechen für das Projekt zu gewinnen und die Versprechen dann nicht zu erfüllen, kann sich als fatal erweisen!

2.4.10 Inkonsistenz von Benutzeranforderungen

Von Entwicklern wird häufig die Inkonsistenz von Benutzeranforderungen beklagt ("... vor drei Wochen wollten sie es so, nun sieht's wieder anders aus ...!"). Diese für Entwickler wirklich unangenehme Situation lässt sich nicht grundlegend beseitigen. Bei der Komplexität heutiger Anwendungen wäre die Annahme unrealistisch, dass alle zu Beginn ermittelten Anforderungen über die Projektlaufzeit hinweg konstant blieben. Eine Entschärfung des Problems ergibt sich durch eine *schrittweise Präzisierung und Differenzierung von Anforderungen z.B. durch Prototyping* (siehe Teil B, Kapitel 4.10).

Diese Problemauflistung soll niemanden entmutigen oder gar von einer Benutzerbeteiligung abhalten! Probleme werden in den meisten Projekten auftauchen, können aber besser gelöst werden, wenn man sie antizipiert. Mit zunehmender Erfahrung werden sie abnehmen, und Benutzerbeteiligung wird fester, 'normaler' Bestandteil der Softwareentwicklung werden.

2.5 FAZIT

Die Ausführungen in diesem Kapitel zeigen, dass es verschiedene, gute Gründe für die Beteiligung von Benutzern gibt. Die grundlegendste und vielleicht wichtigste Voraussetzung ist, dass die Benutzer frühzeitig und ausreichend informiert und ausgebildet werden, bevor ein auf das konkrete Projekt abgestimmtes Beteiligungskonzept gemeinsam entwickelt wird. Die aufgelisteten Probleme bei Beteiligungsprozessen sollen zeigen, dass auch Benutzerbeteiligung ein Lernprozess ist, der nur gemeinsam bewältigt werden kann.