Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

USABILITY ENGINEERING NOW!

Matthias Rauterberg

1. Einleitung

Die Verlagerung der Computerleistung an den Arbeitsplatz bedeutete eine einschneidende Veränderung der Arbeitsbedingungen vieler Angestellter. Die Aufgabenerfüllung erfolgt nun im direkten Kontakt - vermittelt über Bildschirm und Tastatur/Maus - zum Computer. Dabei besteht die Gefahr, daß die Benutzung und Beherrschung des Computers zu einer eigenständigen, mühsamen Aufgabe werden kann, obwohl das System eigentlich zur Unterstützung der originären Aufgaben gedacht war. Der Computer als Arbeitsmittel dient primär dazu, den Benutzer bei seinen Arbeiten zu unterstützen und von unzumutbaren, ständig auszuführenden, z.B. hochgradig routinisierten Aufgaben zu entlasten. Hier kann der Computer hilfreich sein. 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 erträgliches Ausmaß reduziert werden.

Analysiert man aktuelle Software-Entwicklungsprozeße, so erkennt man eine Reihe von Problemen und Schwachstellen. Die Ursachen hierfür sind sowohl in den verwendeten theoretischen Konzepten und den traditionellen Vorgehensweisen (insbesondere Projektmanagement), als auch in unzureichenden Kostenrechnungsmodellen begründet.

Eines der wichtigsten Probleme besteht ganz allgemein darin, ein gemeinsames Verständnis aller betroffenen Personengruppen über den zu automatisierenden Anteil im betroffenen Arbeitssystem herzustellen, also *gemeinsam* verbindliche Antworten auf die Fragen "ob", "wo" und "wie" des geplanten Technikeinsatzes zu finden (siehe Abbildung 1). Hierzu gehört insbesondere die Bestimmung aller Eigenschaften des neu zu gestaltenden Arbeitssystems und der neuen Arbeitsaufgaben.

Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

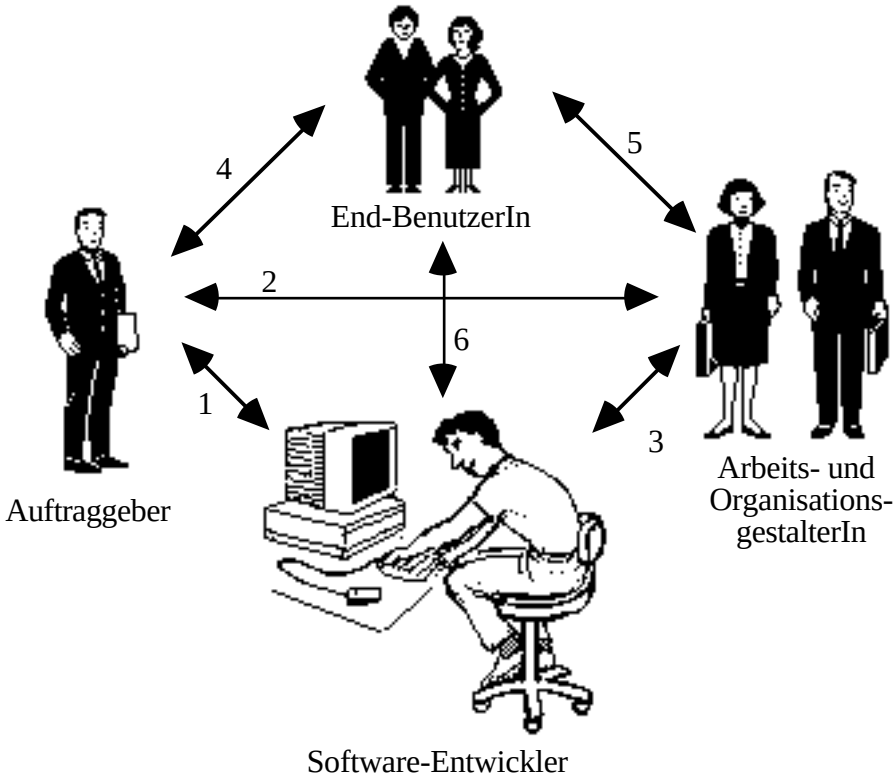

Abbildung 1: Die vier Personengruppen mit den sechs Kommunikations"schienen".

Bei der benutzungsorientierten Systemgestaltung müssen das Wissen der BenutzerInnen, der Arbeits- und Organisationsgestalter sowie der Informatiker integriert werden. Einerseits kann der Benutzer sein Wissen über die Bearbeitung von Aufgaben beisteuern, andererseits muß er Wissen über die Handhabung des neuen Systems erlernen. Dies kann zur Zeit am besten in einem gemeinsamen Vorgehen zwischen BenutzerInnen, Software-Entwicklern, Arbeits- und OrganisationsgestalterInnen sowie dem Auftraggeber erreicht werden.

Heute lautet die Frage nicht mehr, *ob* Benutzer überhaupt beteiligt werden sollen, sondern *wie* durch Benutzerbeteiligung das Fachwissen der Benutzer zur Entwicklung guter, aufgaben- und benutzerorientierter Software genutzt werden kann!

2. Die Überwindung der Barrieren

Aufgrund verschiedener Ansätze in der konkreten Praxis der Softwareentwicklung liegt bereits eine zahlreiche Sammlung von Lösungsmöglichkeiten in der Literatur vor, welche eindeutig auf die Vorteile der Partizipation aller betroffenen Gruppen hinweist. Bei der Analyse dieser Fallbeispiele lassen sich drei wesentliche Barrieren erkennen: die Spezifikations-Barriere, die Kommunikations-Barriere und die Optimierungs-Barriere. Die *Kommunikationsbarriere* läßt sich an den sechs verschiedenen Kommunikationsschienen wie folgt verdeutlichen (siehe Abbildung 1).

Kommunikationsschiene-1: Oft werden über die Köpfe der Betroffenen hinweg Systeme in Auftrag gegeben, bei denen der Auftraggeber meistens nicht genau weiß, was eigentlich notwendig und wichtig wäre.

Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

Kommunikationsschiene-2: Wenn Arbeits- und Organisationsgestalter vom Auftraggeber hinzugezogen werden, dann oft nur, um an den Betroffenen vorbei weitere Rationalisierungspotentiale herauszufinden. Nachweisbar vorteilhafter wäre eine rechtzeitige Einbeziehung der BenutzerInnen, um die notwendigen Verbesserung zu erkennen und mit allen gemeinsam umzusetzen. Arbeits- und Organisationsgestalter haben dabei primär die Rolle eines Moderators und Vermittlers.

Kommunikationsschiene-3: Die eigentlich wichtigste Aufgabe der Arbeits- und Organisationsgestalter ist es, zu verhindern, daß das neue EDV-System schlechte Arbeits- und Organisationsformen "zementiert".

Kommunikationsschiene-4: Der größte Schaden wird durch den Auftraggeber meistens dadurch erreicht, daß er garnicht oder zu spät die Betroffenen informiert und von den relevanten Entscheidungsprozessen ausschließt.

Kommunikationsschiene-5: Wenn Arbeits- und Organisationsgestalter nur an Rationalisierungen interessiert sind, dann ist eine angemessene Beteiligung der BenutzerInnen unmöglich. Besser wäre es, gemeinsam nach wirklichen Verbesserungen in der Arbeitsteilung und der Organisationsform zu suchen.

Kommunikationsschiene-6: Am größten scheinen die Kommunikationsschwierigkeiten zwischen den BenutzerInnen als Fachvertreter und den Software-Ingenieuren als Systementwickler zu sein. Die nicht-technischen Fakten der BenutzerInnen fallen durch das begriffliche Raster der technischen Fachsprache der Ingenieure. Diese Schwierigkeit läßt sich nur begrenzt mit rein sprachlichen Mitteln überbrücken, da die verwendeten Begriffe unscharf sind. Um diese Unschärfe zu überwinden, müssen gemeinsam erlebte, sinnlich erfahrbare Kontexte hergestellt werden. Hierbei sind neben der verbalen Kommunikation im wesentlichen visuelle Kommunikationsmittel geeignet. Je stärker die konkreten Probleme des jeweils Anderen sinnlich erfahrbar wird, desto besser läßt sich diese Kommunikationsbarriere überwinden.

Kasten 1: Beispiel für einen Benutzer (B)-Entwickler (E) Dialog.

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 DB"
B :"ja...? dann nach Autoren und Jahrgang sortieren, auswählen und verschiedene Listen erstellen und ausdrucken."
E:"Ja, dann machen wir ein Menu '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 Menu '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.

Je nach Form, Grad und Inhalt der Benutzerbeteiligung werden schon heute schon bei der Entwicklung von Systemen Benutzer beteiligt. Die am häufigsten eingesetzten Methoden sind: Workshops, Review-Sitzungen, 'user group', betriebsinterne Umfragen, alpha-, beta-Tests, Kontakt zu speziellen Kunden, 'hot-line', etc. Diese Methoden dienen jedoch fast ausschließlich

der Überprüfung auf funktionale Vollständigkeit und Korrektheit. Benutzungsprobleme im handlungs- und arbeitspsychologischen Sinne werden dabei kaum berücksichtigt. Hier setzt die Methode der *benutzungs-orientierten Benchmarks, bzw. Usability-Tests* an. Benutzerbeteiligung ist deshalb notwendig, weil bestimmte Eigenschaften interaktiver Systeme nur in der konkreten Interaktion meßbar sind!

Erst wenn die Benutzungsoberfläche insgesamt software-ergonomisch benutzungsgerecht gestaltet wurde, ist die individuelle Nutzungsweise kein erschwerendes Hindernis, sondern eine echte Unterstützung und Entlastung des Benutzers. Zusammengefaßt kann man sagen, daß Normen, Kriterien und Richtlinien einerseits notwendige Hinweise liefern, andererseits aber auch die Möglichkeit, Schnittstellen zu vereinheitlichen, bieten. 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 Anforderungslisten 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 beteiligten Gruppen dienen.

Dennoch hat es sich gezeigt, daß Kriterien und Richtlinien alleine bei weitem nicht ausreichen, um angemessene Systeme zu enwickeln. Daher ist es notwendig, alle Betroffenen in einem ausreichenden Maße zu beteiligen. Dies erfolgt in der Regel durch die Bildung von Projektteams und Arbeitsgruppen.

3. Das Projekt-Team

In vielen modernen Projektmanagementbüchern wird übereinstimmend Teamarbeit als die adäquate Arbeitsorganisation für die Abwicklung von Projekten dargestellt. Die Ergebnisse einer Analyse zu Projekterfolgskriterien zeigen, daß sehr verschiedene Merkmale eines Projektteams wie z.B. Fachkompetenz, Entscheidungskompetenzen, Partizipation, Motivation, Kontinuität/ Fluktuation im Zusammenhang mit Projekterfolgskriterien stehen. Das Verständnis darüber, wie Teamarbeit zu gestalten ist, damit eine kompetente und motivierte Aufgabenerfüllung erfolgt, ist allerdings in der betrieblichen Praxis sehr unterschiedlich.

Starke Funktions- und Arbeitsteilung in und zwischen Projektteams resultiert in einem Mangel an Transparenz, Eigeninitiative und Verantwortungsübernahme bei den Teammitgliedern, die Lern- und Entwicklungsmöglichkeiten bei der Arbeit sind eingeschränkt, geringe Motivation und Produktivität können die Folge davon sein.

Effiziente, weitgehehend selbständig arbeitende Projektteams sind (in Anlehnung an Ulich, 1991) durch folgende Merkmale gekennzeichnet: 1. relative Unabhängigkeit des Projektteams, 2. innerer Aufgabenzusammenhang im Projektteam sowie 3. Einheit von Produkt und Projektteam.

Dies bedeutet, daß einer Gruppe von Analytiker/Programmierern eine ganzheitliche Projektaufgabe - welche idealtypischerweise möglichst viele interdependente Teilaufgaben von A-Z umfaßt - übetragen wird. In gemischten Teams ist eine solche Gruppe z.B. durch Benutzervertreter ergänzt. Die Übertragung einer ganzheitlichen Projektaufgabe ist eine wesentliche Voraussetzung für das Verständnis einer gemeinsamen Aufgabe im Team. Dieses

Verständnis ermöglicht ein höheres Maß an Selbstregulation und gegenseitiger Unterstützung. Selbstregulation beinhaltet dabei, daß das Team neben inhaltlichen Aufgaben auch organisatorische Aufgaben wahrnimmt. Die Organisation der Arbeit erfolgt weitgehend selbständig durch das Projektteam.

4. Das iterativ-zyklische Ablaufmodell

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 mit anderen Worten: es ist zu überlegen, welche Methoden eingesetzt werden, ob sie vorhanden sind bzw. gekauft oder entwickelt werden müssen und ob sie eine spezielle Schulung verlangen oder das Vorhandensein bestimmter Geräte wie z.B. Prototypingwerkzeug, Videokamera, etc. bedingen! Müssen sie während des Projektes entwickelt oder gekauft und geschult werden so ergeben sich unliebsame Verzögerungen im Ablauf und/ oder improvisierte Methoden!

Die beim Start und beim Durchlauf des iterativ-zyklischen Ablaufmodells (siehe Abbildung 2) zu beachtenden Aspekte werden im folgenden diskutiert. Als eine wesentliche Rahmenbedingung von moderner Software-Entwicklung hat sich der Typ der zu entwickelnden Software herausgestellt. Es lassen sich die folgenden vier Typen unterscheiden:

Typ-A: Spezifische Anwendung für firmen-interne Fachabteilung; Fachabteilung und Softwareentwicklungsabteilung gehören derselben Firma an.

Typ-B: Spezifische Anwendung für externe Anwender; Fachabteilung und Softwareentwicklungsabteilung gehören unterschiedlichen Firmen an.

Typ-C: Standardbranchenlösung für externe Anwender; dieser Typ geht oftmals aus Projekten des Typs A oder B hervor, indem spezifische Anpassungen von Individualsoftwarelösungen (Typ-A, B) für weitere Anwender vorgenommen werden.

Typ-D: Standardsoftware für weitgehend anonymen Anwenderkreis.

Der Einstieg in das Ablaufmodell bei einer Neuentwicklung erfolgt über den Start-A, wohingegen bei einer Weiterentwicklung der Einstieg bei Start-B erfolgt. Je nach Projekttyp kommen kontextspezifisch unterschiedliche lokale Optimierungszyklen zur Optimierung spezifischer Zwischenergebnisse zum Einsatz. Die Entscheidung über die jeweils aktuelle Vorgehensweise gehört zur Aufgabe des Projektmanagements und schlägt sich in der gewählten Aufbauorganisation nieder.

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.

Die Kernelemente eines Beteiligungsmodells sind: (1.) Form der Beteiligung; (2.) Grad, bzw. Ausprägung der Beteiligung; (3.) Inhalt der Beteiligung; (4.) Repräsentativität der Beteiligten; (5.) Methoden der Beteiligung; sowie (6.) Zeitpunkt der Beteiligten. Für eine ausführliche Diskussion dieser einzelnen Kernaspekte siehe bei Rauterberg et al. (1994).

Wir haben ein iterativ-zyklischen Ablaufmodell entwickelt, welches an verschiedenen Stellen im Entwicklungsprozeß die Beteiligung von BenutzerInnen vorsieht (siehe Abbildung 2). Der

gesamte Prozeß ist dabei in vier Quadranten aufgeteilt: (I) Analyse und Bewertung des Arbeitssystems; (II) Spezifikation und Entwurf des EDV-Systems; (III) Realisierung der technischen Komponenten; (IV) Benutzung und Wartung des EDV-Systems. Für die Erstellung der verschiedenen Zwischenergebnisse stehen eine Reihe von unterschiedlichen Methoden zur Verfügung (siehe Tabelle 2).

Als *evaluative Beteiligung* bezeichnet man ein Vorgehen, welches die Benutzer zu definierten Zeitpunkten - beim Vorliegen eines (Zwischen-)Ergebnisses - zur Beurteilung und Bewertung von Vorschlägen, Prototypen oder System-Modulen einbezieht. Diese eher passive Form der Beteiligung kann in unterschiedlicher Weise erfolgen.

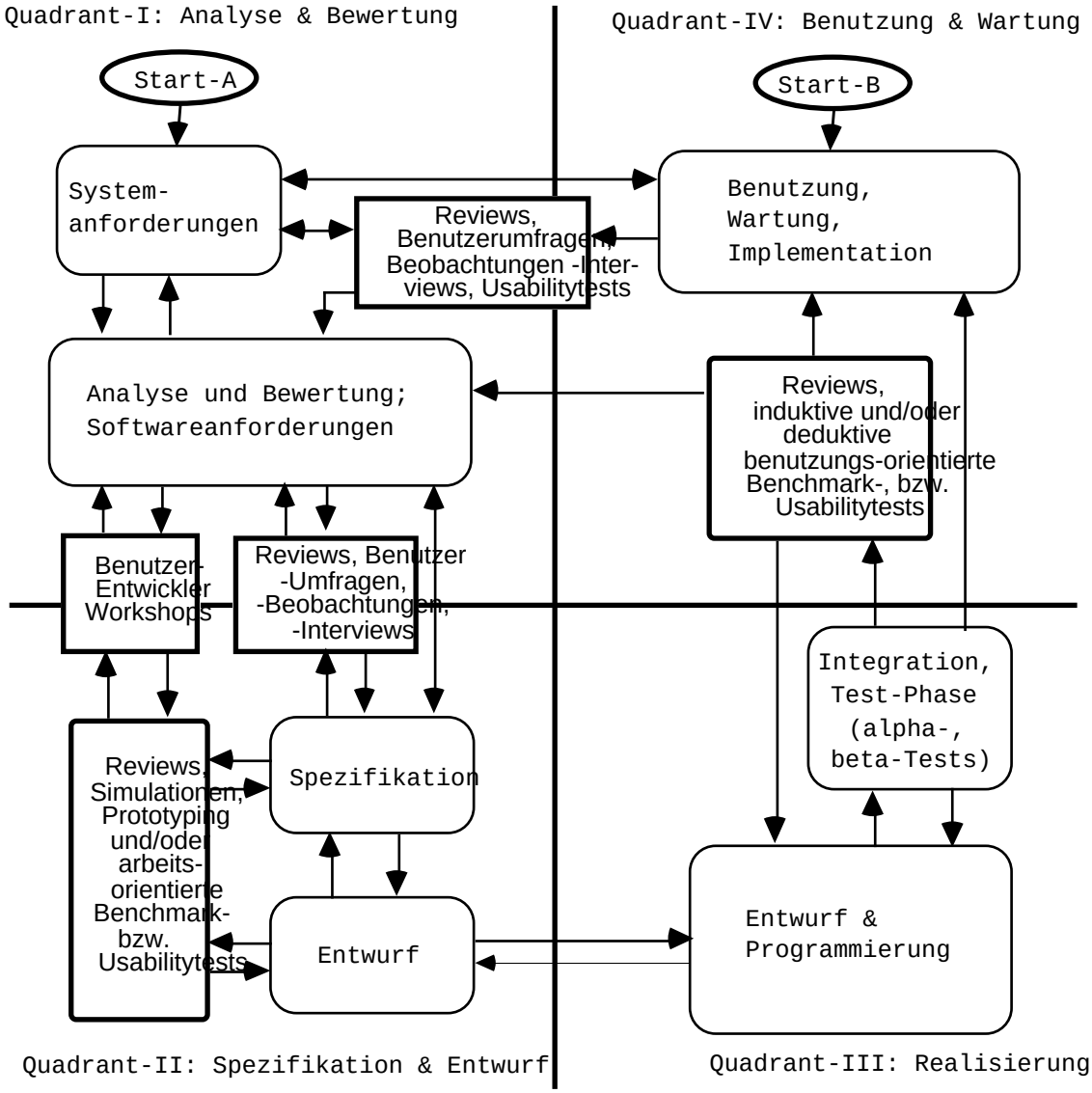

Abbildung 2: Ablaufmodell eines partizipativen Softwareentwicklungskonzeptes mit einer Übersicht über die verschiedenen Optimierungszyklen innerhalb und zwischen den einzelnen Quadranten (siehe Rauterberg et al 1994).

Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

Kasten 2: Beispiele für evaluative Beteiligungsformen zur Beurteilung der Maskengestaltung.

(1) Die Vorschläge werden den Benutzern 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? und subjektive Daten, z.B. Benutzerurteil, etc., erhoben.

Als *prozessuale Beteiligung* wird bezeichnet, wenn die Benutzer bzw. ihre Vertreter zu einem Teil ihrer Arbeitszeit kontinuierlich - von Anfang bis Ende - im Projekt mitarbeiten und dabei Teilaufgaben übernehmen; 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 3: Beispiele für prozessuale Beteiligungsformen.

(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.

Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

Tabelle 2: Übersicht über die verschiedenen Möglichkeiten zur Benutzerpartizipation.

Methode	Aktivität	"Test"	Ergebnis	Zyklus-Dauer
Diskussion	verbale Kommunikation	rein verbale Interpretation	globale Design-Entscheidungen	Sekunden - Minuten
	Metaplan, Flip-Charts, etc.	visuelle + verbale Interpretation	spezifische Design-Entscheidungen	Minuten - Stunden
Simulation	Handskizzen, Szenarien, "wizard of oz", etc.	visuelle + verbale Interpretation	Spezifikation der Ein/ Ausgabeschnittstelle	Minuten - Tage
	Erstellung von Ablaufplänen, etc. mittels (semi)-formaler Methoden	visuelle + verbale Interpretation bei entsprechender Qualifikation	(semi)-formale Beschreibungs-Dokumente	Stunden - Wochen
Prototyping	horizontales Prototyping	"lautes Denken", "walk-through"	Spezifikation der Dialogkomponente	Tage - Wochen
	partiell vertikales Prototyping	heuristische Evaluation	Spezifikation von Teilen der Anwendungskomponente	
	vollständig vertikales Prototyping	aufgaben-orientierte Benchmark-Tests	Spezifikation der Anwendungskomponente	
Versioning	Durchlauf des gesamten Entwicklungszyklus	induktive Benchmark-Tests	erste weitgehend vollständige Version	Monate - Jahre
	Durchlauf des gesamten Entwicklungszyklus	deduktive Benchmark-Tests	mehrere weitgehend vollständige Versionen	

Eine weitere Unterscheidungsdimension betrifft die Frage, ob die Benutzer *direkt* (z.B. alle betroffenen Benutzer in einem Workshop mit Entwicklern) oder *indirekt* (über Benutzervertreter, Koordinatoren, Betriebsrat etc.) in die Entwicklung einbezogen werden. Anders betrachtet geht es um den Weg, den Informationen vom Benutzer bis zum Entwickler zurücklegen müssen; der direkte Einbezug eröffnet Chancen zum 1:1-Austausch von Informationen, ist aber bei großer Benutzerzahl schwierig zu realisieren (Entwickler: "Ich kann ja nicht mit allen 150 Benutzern sprechen ..."). Aus zeitlichen, organisatorischen und ökonomischen Gründen bietet sich deshalb häufig der indirekte Einbezug an. Wie bei allen Informationsflüssen, die über mehrere Stellen verlaufen, besteht aber die Gefahr, daß die ursprünglichen Informationen gefiltert und verzerrt am Bestimmungsort ankommen.

Kasten 4: Beipiel für die Aufgaben der BenutzervertreterInnen.

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!
Gegenbeispiel:
Ein Projektleiter und ein Entwickler verbrachten drei Monate mit der Aufarbeitung (inklusive Rückfragen) von 367 Benutzeranforderungen; wichtige widersprüchliche Anforderungen wurden mit den Benutzern direkt in einem Workshop geklärt.

Die Auswahl der BenutzervertreterInnen bestimmt maßgeblich das Ergebnis der Mitarbeit; werden unmotivierte, schlecht qualifizierte, zur Zeit in der Fachabteilung gerade abkömliche 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!

5. Der evaluative Ansatz mittels Checklisten

Checklisten sind speziell für die Bewertung der Benutzungsfreundlichkeit entwickelt worden uind sind zumeist in einer englischen Fassung erhältlich. Deutschsprachige Checklisten sind in Baitsch et al (1989), EDV im Büro (1990), sowie Siemens Nixdorf Styleguide (1992) zu finden.

Stärken:

- Gegenüber Tests mit Benutzern verringert sich der Bewertungsaufwand beim Einsatz von Checklisten beträchtlich. Dadurch ist das wirtschaftliche Interesse an diesem Verfahren groß.
- Checklisten helfen zu verhindern, daß wichtige Aspekte bei der Bewertung vergessen werden.
- Checklisten eignen sich besonders als Hilfsmittel bei der Bearbeitung von Routinefällen.
- Checklisten enthalten in komprimierter Form die als relevant anzusehenden Aspekte.

Schwächen:

- Checklisten können Sachverstand nicht ersetzen; das Arbeiten mit ihnen bedingt hinreichende Kenntnis von Aufgaben, Problemen, Zusammenhängen und Lösungsansätzen.
- Die Zuverlässigkeit ist geringer als bei anderen Verfahren, da bewußtes oder unbewußtes Mißverstehen der Fragen möglich ist.
- Nur begrenzte Möglichkeiten zur eingehenderen Erläuterung der einzelnen Fragen sind bei Papier-Versionen gegeben.
- Spezielle Aspekte, die nicht in das Raster der Checkliste hineinpassen, lassen sich nicht adäquat bei der Auswertung berücksichtigen.

6. Der partizipative Ansatz mittels Workshops

Da es sich bei der Begriffswelt eines Unternehmens fast immer um einen bereichsübergreifenden Analysebereich handelt, kann dieser nicht in Zusammenarbeit mit nur einer einzelnen Person geklärt werden. Es empfiehlt sich daher, alle oder repräsentativ ausgewählte Mitarbeiter in der Form eines Workshops am Prozeß der Bedeutungsanalyse

wichtiger unternehmensweiter Konzepte und Begriffe zu beteiligen. Unklarheiten, Differenzen und Begriffsdefekte lassen sich so schneller ausräumen als bei einem Interview mit einzelnen Benutzern. Der Vergleich der Begriffsdefinitionen und die Klärung von Differenzen und Wiedersprüchen erfolgt direkt in der Gruppenarbeit und ermöglicht die Vereinheitlichung der Begriffswelt. Vagheiten werden aufgedeckt, da sich die Beteiligten in der Diskussion zu größerer Genauigkeit und Objektivität in den Aussagen veranlaßt fühlen. Der Gesamtaufwand für einen Workshop ist gegenüber den Interviews mit einzelnen Benutzern stark reduziert.

Um die Produktivität eines Workshops zu gewährleisten, sollte die Anzahl der Beteiligten nicht mehr als sieben bis zehn Teilnehmer betragen. Die Teilnehmer sollten rechtzeitig und ausreichend über die Aufgabenstellung und Zielsetzung des Workshops informiert worden sein und die notwendigen Arbeitsunterlagen (z.B. Begriffsliste zur Bedeutungsanalyse) erhalten haben.

Der Gesprächsverlauf sollte vom Moderator optische aufgezeichnet und strukturiert werden. Diese Visualisierung dient als Grundlage ("roter Faden") der Gruppendiskussion. Die Komplexität des Analysegegenstands kann reduziert werden. Kein Gedanke geht verloren, Wiederholungen werden vermieden und Redundanzen aufgedeckt. Um die Auswertung der Workshopergebnisse zu erleichtern, empfiehlt es sich, die Visualisierungsmethode an den Regeln semantischer Datenmodelle zu orientieren. Die Visualisierung erleichter die Beteiligung aller Gruppenmitglieder, da zurückhaltende Mitarbeiter ihre Beiträge leichter in der optischen Diskussion einbringen können. Es entsteht gleichzeitig während des Gesprächs ein graphisches Protokoll an den Stellwänden. Das Problem der kontinuierlichen, gesprächsbegleitenden Aufzeichnung kann somit gelöst werden.

7. Der benutzungsorientierte Ansatz mittels Tests

Es werden drei Methoden zur partizipativen Entwicklung von [Standard]-Software vorgestellt: (1.) aufgabenorientierte, (2.) induktive und (3.) deduktive Benchmark-, bzw. Usability Tests. Mit diesen Methoden ist es möglich, unter Verwendung von konkreten Aufgaben das zu bewertende interaktive System durch ausgewählte Benutzer zu testen (siehe Dumas & Redish 1993; Nielsen 1993; sowie Nielsen 1994).

Ganz allgemein läßt sich die Methode der benutzungs-orientierten Benchmark, bzw. Usability-Tests (BUTs) zur Gewinnung von Gestaltungsvorschlägen bei der Entwicklung neuer Systeme einsetzen (induktive BUTs). Wenn bereits verschiedene Systemalternativen zur Verfügung stehen, kann man zwischen diesen Alternativen durch BUTs sinnvolle Entscheidungen herbeiführen (deduktive BUTs). Zusätzlich können deduktive BUTs zur Überprüfung von getroffenen Designentscheidungen im Rahmen der partizipativen Entwicklung eingesetzt werden.

Die *aufgabenorientierten BUTs* werden bei der Evaluation eines Prototypen zur Gewinnung von Gestaltungsvorschlägen für die benutzerangemessenste Aufgabenbearbeitung eingesetzt. BUTs lassen sich ebenso zur Gewinnung von Verbesserungsvorschlägen, bzw. zur Analyse von Schwachstellen in der Benutzbarkeit einsetzbar.

Jeder Test wird durch einen oder zwei Testleiter vorbereitet und durchgeführt. Es empfiehlt sich, daß ein Produktverantwortlicher und ein Repräsentant der Entwicklungsabteilung als passive Beobachter beteiligt sind, um die Vermittelbarkeit der Ergebnisse zu gewährleisten. Die Durchführung eines BUTs erfolgt in der Regel in einem speziell eingerichteten Usability-Labor (siehe Nielsen 1994), kann aber auch unter besonderen Bedingungen am Arbeitsplatz stattfinden.

Die *induktiven BUTs* sind bei der Evaluation einer (Vor)-Version zur Gewinnung von Gestaltungs- und Verbesserungsvorschlägen, bzw. zur Analyse von Schwachstellen in der

Benutzbarkeit einsetzbar. Induktive BUTs können immer dann zum Einsatz kommen, wenn nur *eine* Version der zu testenden Software vorliegt. Demgegenüber verfolgen *deduktive BUTs* primär den Zweck, zwischen mehreren Alternativen (mindestens zwei Prototypen, bzw. Versionen) zu entscheiden. Zusätzlich lassen sich jedoch auch mit deduktiven BUTs Gestaltungs- und Verbesserungsvorschläge gewinnen.

7.1. Aufgabenorientierte Usability-Tests

Wie bei den Szenarien wird bei aufgabenorientierter BUTs eine möglichst realistische Arbeitssituation nachgestellt. Im Unterschied zu den Szenarien handelt es sich nicht um eine verbale Beschreibung, sondern um das möglichst realistische Nachbilden der jeweiligen Arbeitssituation. Diese BUTs lassen sich daher z.B. in Benutzer-Entwickler-Workshops einsetzen, um eine typische Arbeitssituation im Rollenspiel nachzuspielen. Dadurch kann dem Entwickler eine direkte, anschauliche Rückmeldung über die zunächst verbal beschriebene Situation gegeben werden. Dabei übernimmt ein Benutzer seine Rolle als Mitarbeiter in einer typischen, bzw. geplanten Arbeitssituation mit entsprechend abgesprochenen Aufgaben. Unter Einsatz des erstellten Prototypen wird die Aufgabenbearbeitung durchgespielt. Wenn noch kein Prototyp vorhanden ist, kann dieser auch mit möglichst einfachen Hilfsmitteln (Pappschachtel als Computerattrappe, Folien oder Papierblätter als Maskenersatz, etc.) simuliert werden. Die anderen Benutzer übernehmen neben dem Testleiter die Rolle der Beobachter und Kommentatoren, wenn sie Unterschiede zu der von ihnen bevorzugten Bearbeitungsweise feststellen. Am besten läßt sich dieser BUT direkt am Arbeitsplatz oder in unmittelbarer Nähe dazu durchführen.

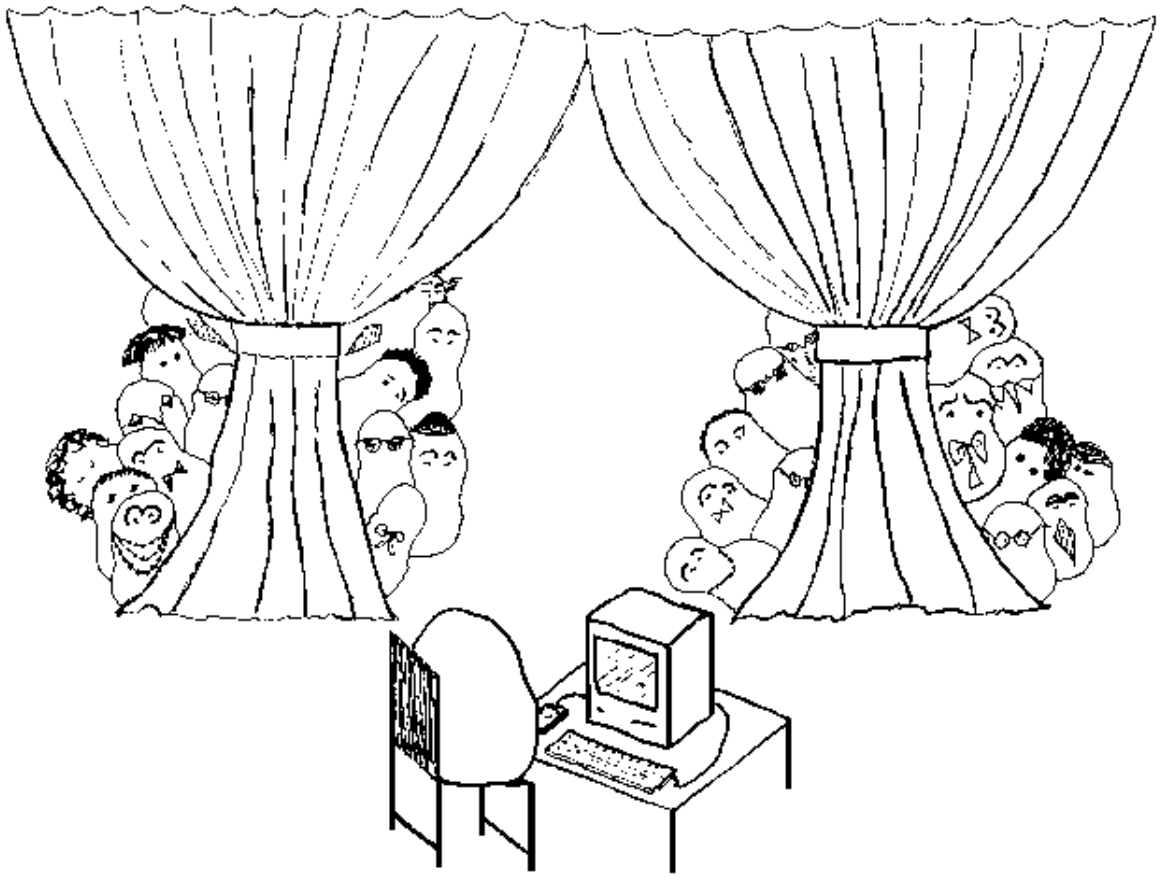

Abbildung 3: Darstellung eines Benchmark-, bzw. Usability-Tests mit allen Betroffenen.

Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

7.2. Induktive Usability-Tests

Bei der Durchführung eines induktiven BUTs ist eine Anzahl von Bedingungen zu beachten. Da die meisten Bedingungen zur Durchführung eines induktiven BUTs mit denen eines deduktiven BUTs übereinstimmen, werden bei der Darstellung deduktiver BUTs nur noch die Änderungen und Ergänzungen erwähnt.

Damit keine unnötigen Fehler während eines induktiven BUTs - z.B. hervorgerufen durch unvollständige Systemfunktionalität - auftreten können, sollte das zu testende System ein möglichst realitätsgerechtes Systemverhalten besitzen.

Als erstes sind eine oder mehrere Aufgaben festzulegen, welche auf die zu testenden Systemteile abgestimmt sind. Diese Aufgaben sind dem typischen Aufgabenkontext der zukünftigen End-Benutzer zu entnehmen. Sofern dieser Aufgabenkontext nicht oder nur vage bekannt ist (z.B. bei neuen Systemen), sollten die Aufgaben zumindest jedoch typische Teil-Aufgaben enthalten. Eine einzelne Aufgabe sollte nicht zu komplex, aber auch nicht zu einfach sein; die Bearbeitungszeiten liegen in der Regel zwischen fünf und fünfzehn Minuten.

Die Formulierung der Aufgaben sollte den Benutzern während der Aufgabenbearbeitung schriftlich vorliegen. Die Aufgabenbeschreibung hat das unterschiedliche fachliche Vorwissen der Benutzer zu berücksichtigen; bei hohem fach-spezifischem Vorwissen ist die Beschreibung möglichst *problem-orientiert* abzufassen, bei geringem fach-spezifischen Vorwissen ist die Beschreibung *handlungs-orientiert* zu halten (siehe Kasten 5). Die handlungs-orientierte Aufgabenbeschreibung soll verhindern helfen, daß die beobachteten Benutzungsprobleme überwiegend aufgrund fehlenden Fachwissens zustande gekommen sind.

Kasten 5: Beispiele für problemorientierte und handlungsorientierte Aufgabenbeschreibungen.

> Problemorientierte Aufgabenbeschreibung:
> "Erstellen Sie bitte einen Brief mit folgendem Inhalt: ... und bereiten Sie ihn zum Eintüten und Versenden an folgende Adressen: ... vor. Benutzen Sie dabei bitte als Briefvorlage das Dokument mit dem Namen: ..."
> Handlungsorientierte Aufgabenbeschreibung:
> "Laden Sie bitte das Textdokument mit folgendem Namen: ... und ergänzen es um den folgenden Inhalt: ... Versehen Sie dieses Textdokument mit allen für die Serienbrieferstellung notwendigen Steueranweisungen. ... usw."

Im Unterschied zum aufgabenorientierten BUT sind beim induktiven und deduktiven BUT *mehrere* Benutzer als Software-Tester beteiligt. Die Benutzergruppe sollte möglichst *repräsentativ* für die Population der End-Benutzer sein und mindestens sechs Testpersonen umfassen. Die Auswahl der Benutzer kann z.B. *zufällig* aus dem potentiellen oder aktuellen Benutzerkreis erfolgen. Die ausgewählten Benutzer sollten das zu testende System nicht kennen. Da sich diese Bedingung bei größeren Entwicklungsphasen, bzw. Weiterentwicklungen oftmals nicht einhalten läßt, muß die Vorerfahrung der Benutzer mit dem System, bzw. ähnlichen Systemen kontrolliert werden.

Anzustreben ist eine den realen Einsatzbedingungen möglichst entsprechende Testumgebung. Da es für die spätere Auswertung sehr wichtig ist, das Verhalten der einzelnen Benutzer sowie das entsprechende Systemverhalten auf Video, Tonband, "Screen-recording", etc. aufzuzeichnen, empfiehlt es sich, einen ruhigen, abgeschlossenen Raum zu benutzen.

Der Testleiter sollte sich stets darum bemühen, jedem Benutzer das Gefühl der Wichtigkeit und Wertschätzung zu vermitteln. Die folgenden drei Aspekte sind für die Schaffung eines

vertrauensvollen und für Kritik offenen Klimas hilfreich. (1.) Mit möglichst allgemein verständlichen Worten wird dem Benutzer Ziel und Zweck des BUTs erläutert (z.B. Test eines Prototypen in einem frühen Entwicklungsstadium, etc.). (2.) Es ist besonders wichtig, dem Benutzer verständlich zu machen, daß das System und *nicht* er als Benutzer getestet werden soll. Jede Art von Schwierigkeiten seitens des Benutzers bei der Aufgabenbearbeitung sind von besonderem Interesse! (3.) Jedem Benutzer muß zugesichert werden, daß er die Durchführung des BUTs jederzeit unterbrechen und sogar abbrechen kann, ohne daß irgendwelche negativen Konsequenzen (z.B. Wegfall einer zugesagten Bezahlung, etc.) erfolgen. Die vollständige Freiwilligkeit der Teilnahme und Zusicherung der Vertraulichkeit ist unabdingbar für die Gewinnung von brauchbaren Testergebnissen.

Um die Handlungsziele der Benutzer während der Aufgabenbearbeitung erfassen zu können, werden sie gebeten "laut zu denken". Viele Benutzer haben jedoch Schwierigkeiten, ihre Gedanken während der Bearbeitung einer Aufgabe laut zu äußern. Es empfiehlt sich daher, dem Benutzer zu verdeutlichen, was mit "lautem Denken" gemeint ist, und mit ihm einige Übungsbeispiele gemeinsam durchzugehen.

Als Testkriterien werden alle erhobenen Meßwerte bezeichnet, welche bei der Auswertung Aufschluß über die Güte der Benutzbarkeit des zu testenden Systems Auskunft geben können. Die Menge der Testkriterien teilt sich auf in die Menge der *qualitativen* Meßgrößen (problematische Benutzungssituationen, Fehlerarten, etc.) und die Menge der *quantitativen* Meßgrößen auf (Bearbeitungsdauer, Anlernzeit, Überlegungszeit, Anzahl Fehler).

Der Testleiter hält sich während der Aufgabenbearbeitung ruhig im Hintergrund. Für ihn ist es sehr wichtig, seinen Wunsch, dem Benutzer in problematischen Situationen sofort zu helfen, zurückzuhalten. Ein zu frühes Eingreifen des Testleiters hindert den Benutzer, eine eigene Lösung zu finden. Als Testleiter sollten daher keine an der Entwicklung des zu testenden Systems unmittelbar beteiligten Personen eingesetzt werden. Es wird sogar verschiedentlich empfohlen, daß der Testleiter sich völlig passiv verhält und dies dem Benutzer vorher deutlich macht.

Während der Durchführung eines BUTs wird man immer wieder überraschende und sehr informative Benutzungsweisen ("critical incidents", "Stolpersteine") beobachten können. Es lohnt sich, diese auf Video aufzuzeichnen, um sie dann mit den Entwicklern später detailliert diskutieren zu können. Wichtig ist dabei, daß die beobachtbaren Schwierigkeiten niemals dem Benutzer, sondern ausschließlich der Benutzbarkeit des zu testenden Systems angelastet werden! Wenn mehrere Benutzer dieselben Schwierigkeiten hatten, kann es nicht an ihnen liegen!

7.3. Deduktive Usability-Tests

Deduktive BUTs dienen primär der Entscheidungsfindung zwischen verschiedenen Systemalternativen, bzw. zur Kontrolle der erreichten Verbesserung und erst sekundär der Gewinnung von Gestaltungsvorschlägen. Es ergeben sich daher einige Unterschiede in den Anforderungen an die Durchführung von deduktiven BUTs.

Im Gegensatz zu induktiven BUTs müssen die verschiedenen zu testenden, alternativen Systemversionen beim deduktiven BUT verstärkt der Forderung nach einem - an realen Einsatzbedingungen gemessenen - realistischen Systemverhalten (d.h. möglichst funktionale Vollständigkeit, adäquates Systemantwortzeitverhalten, etc.) genügen. Diese Anforderungen sind deshalb wichtig, weil die Entscheidung zwischen den Systemalternativen primär mittels quantitativer Meßgrößen gefällt wird.

Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

8. Fazit

Zur Beteiligung von Benutzern bei der Software-Entwicklung gibt es viele, unterschiedliche Möglichkeiten, aber kein Patent-Rezept oder einen "one best way"!

Die praktische Realisierung in konkreten Fällen ist u.a. vom *Umfang*, vom *Inhalt* und der *Neuartigkeit* des *Vorhabens*, der *Anzahl* der betroffenen *Mitarbeiter* sowie von den *personellen* und *infrastrukturellen Voraussetzungen* abhängig. Die Effizienz des Entwicklungsprozesses und die Qualität des Produkte werden dabei maßgeblich von der Projektorganisation, den fachlichen und vor allem sozialen Qualifikationen der Beteiligten sowie den vorhandenen Methoden und Werkzeugen beeinflußt.

Zusammen mit einer gründlichen Ausbildung der Benutzer werden gute Voraussetzungen geschaffen, daß das System zu einem echten Hilfsmittel bei der Aufgabenerfüllung wird.

Am wichtigsten ist jedoch, daß wir anfangen zu lernen, Technik, Organisation und den Einsatz menschlicher Qualifikation gemeinsam zu planen. Betrachten wir also die Technik als eine Option, welche es uns gestattet, unsere Lebens- und Arbeitsräume menschengerecht und lebenswert zu gestalten.

Matthias Rauterberg
Institut für Arbeitspsychologie
Eidgenössische Technische Hochschule
Nelkenstraße 11, CH-8092 Zürich

9. Literatur

Baitsch, C., Katz, C., Spinas, P. & Ulich, E. (1989). Computerunterstützte Büroarbeit. Verlag der Fachvereine.

Dumas, J. & Redish, J. (1993) A practical guide to usability testing. Ablex Publishing.

EDV im Büro (1990) Handbuch zur menschengerechten Gestaltung. Oldenbourg.

Nielsen, J. (1993) Usability Engineering. Academic Press.

Nielsen, J. (1994, Hrsg.) Usability Laboratories. *Behaviour and Information Technology* Vol. 13, Nr.1 & 2.

Rauterberg, M., Spinas, P., Strohm, O., Ulich, E. & Waeber, D. (1994) Benutzerorientierte Software-Entwicklung. Verein der Fachverlage.

Siemens Nixdorf Informationssysteme AG (1992) Alpha-Styleguide-Checkliste V1.0. (Bestell-Nr.: U8557-J-Z147-1), München.

Siemens Nixdorf Informationssysteme AG (1992) Styleguide-Checkliste. (Bestell-Nr.: U6615-J-Z97-1), München.

Ulich, E. (1991) Arbeitspsychologie. Poeschel Verlag.

Published in: P. Pastors (Hrsg.) Wirtschaftlichkeit der Unternehmensweiten Informationsverarbeitung. Wuerzburg: Vogel Vlg; pp. 61-78

10. Fragen für den Entscheider:

Bitte lesen Sie die folgenden Fragen in Ruhe durch und beantworten Sie sie zügig. Anschliessend bieten wir Ihnen eine einfache Auswertung mit Entscheidungshinweisen an.

Glauben Sie, daß …

1. bei der Einführung von moderner Technologie in Ihrem Unternehmen die Beteiligung der Mitarbeiter in den meisten Fällen von Vorteil sein kann? [ ] ja [ ] nein
2. die folgenden Interessengruppen in der Projektorganisation vertreten sein sollten: Management, Stäbe/Spezialisten, betroffene Fachabteilungen / zukünftige Benutzer und eventuell externe Berater? [ ] ja [ ] nein
3. ein Konzept über die Projektabwicklung mit Zielsetzung, Zeitplan, Ressourcen, Aktivitäten, etc. notwendig ist [ ] ja [ ] nein
4. die Funktion, Aufgabe, Kompetenz und Verantwortlichkeit der gebildeten Gremien – v.a. bezüglich der Entscheidungsprozesse – definiert sein müssen? [ ] ja [ ] nein
5. ein zyklisches Ablaufmodell einem linearen Phasenmodell in der Regel überlegen ist? [ ] ja [ ] nein

Sind Sie sicher, daß …

1. die von Ihnen vorgesehenen Projekt- und Arbeitsgruppen ihren Aufgaben entsprechend personell richtig besetzt sind? [ ] ja [ ] nein
2. die Benutzervertreter repräsentativ für die gesamte, zukünftige Benutzerpopulation sind? [ ] ja [ ] nein
3. ein mit hinreichenden Kommunikationskanälen ausgestattetes Informationsverteilungsnetz vorhanden ist? [ ] ja [ ] nein
4. ein Dokumentationssystem eingerichtet wurde? [ ] ja [ ] nein
5. alle Voraussetzungen methodischer und technischer Art für den Projektbeginn überprüft und erfüllt sind? [ ] ja [ ] nein
6. der Inhalt und die Zielsetzung des Projektes klar definiert sind? [ ] ja [ ] nein
7. Inhalt und Zielsetzung des Projektes allen Beteiligten bekannt sind und ein Konsens darüber besteht? [ ] ja [ ] nein
8. genügend Zeitreserven für die verschiedenen Zyklen eingeplant sind? [ ] ja [ ] nein
9. besonders für die frühen Zyklen des Projektes (Analyse, Anforderungsermittlung und Grobkonzept) ausreichend Zeit vorgesehen ist, sodaß auch mehrere Lösungsvarianten entwickelt werden können? [ ] ja [ ] nein
10. alle Beteiligten den erforderlichen Ausbildungsstand haben? [ ] ja [ ] nein

Falls Sie mit der Beantwortung aller Fragen fertig sind, zählen Sie bitte die Anzahl der Nein-Antworten. Falls Sie mehr als fünf Nein-Antworten haben, bitten wir Sie, den Artikel noch einmal aufmerksam durchzulesen.