

Benutzungsoberflächen und ihr Einfluss auf die Akzeptanz und Nutzung der DV

Matthias Rauterberg
ETH Zurich

1994



Congress Report "exponet'94" (Band 1, S. 111-142).
München: Gesellschaft für Europäische Wirtschaftsinformation.

Biographie

Matthias Rauterberg
Dipl.Inf., Dipl.Psych.

Institut für Arbeitspsychologie
Eidgenössische Technische Hochschule (ETH)
Nelkenstrasse 11, CH-8092 Zürich

Herr Rauterberg ist Dozent für Informatik und Arbeitspsychologie an der Eidgenössischen Technischen Hochschule (ETH) in Zürich. Er leitet z.Z. ein interdisziplinäres Forschungsprojekt zur Gestaltung von Multimedia-Systemen.

Herr Rauterberg ist Mitglied im Editorial-Board der internationalen Fachzeitschrift "Interacting with Computers", sowie Mitglied im Advisory-Board der Zeitschrift "PC professional". Er ist darüberhinaus als Gutachter in Programmkomitees verschiedener wissenschaftlicher Konferenzen im deutschsprachigen und internationalen Bereich tätig. In den letzten Jahren hat er über 80 Fachartikel publiziert. Er hat verschiedene nationale und internationale Workshops zum Stellenwert und Einsatz von Usability-Tests in der Softwareentwicklung durchgeführt.

Herr Rauterberg war einer der ersten, der an einer deutschsprachigen Hochschule Usability-Tests als Methode zur Gestaltung und Verbesserung interaktiver Software entwickelt und eingesetzt hat. Er hat in enger Zusammenarbeit mit deutschen Softwareherstellern Usability-Tests auf ihre Praxistauglichkeit geprüft und optimiert.

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.

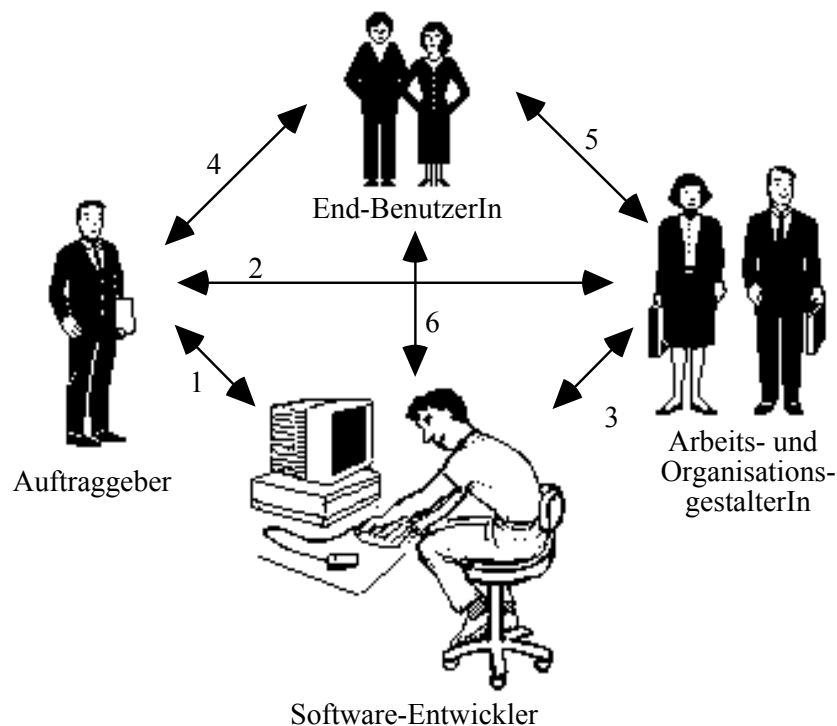


Abbildung 1: Die vier Personengruppen mit den sechs Kommunikations"schienen".

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 Abb. 1). Hierzu gehört insbesondere die Bestimmung aller Eigenschaften des neu zu gestaltenden Arbeitssystems und der neuen Arbeitsaufgaben.

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!

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 Kommunikationsschienen wie folgt verdeutlichen (siehe Abb. 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.

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 Verbesserungen 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 Arbeitsweisen und Organisationsformen "zementiert".

Kommunikationsschiene-4: Der größte Schaden wird durch den Auftraggeber meistens dadurch erreicht, daß er gar nicht 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 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 *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 entwickeln. Daher ist es notwendig, alle Betroffenen in einem ausreichenden Maße zu beteiligen. Dies erfolgt in der Regel durch die Bildung von Projektteams und Arbeitsgruppen.

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, weitgehend 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 - übertragen 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.

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 Abb. 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).

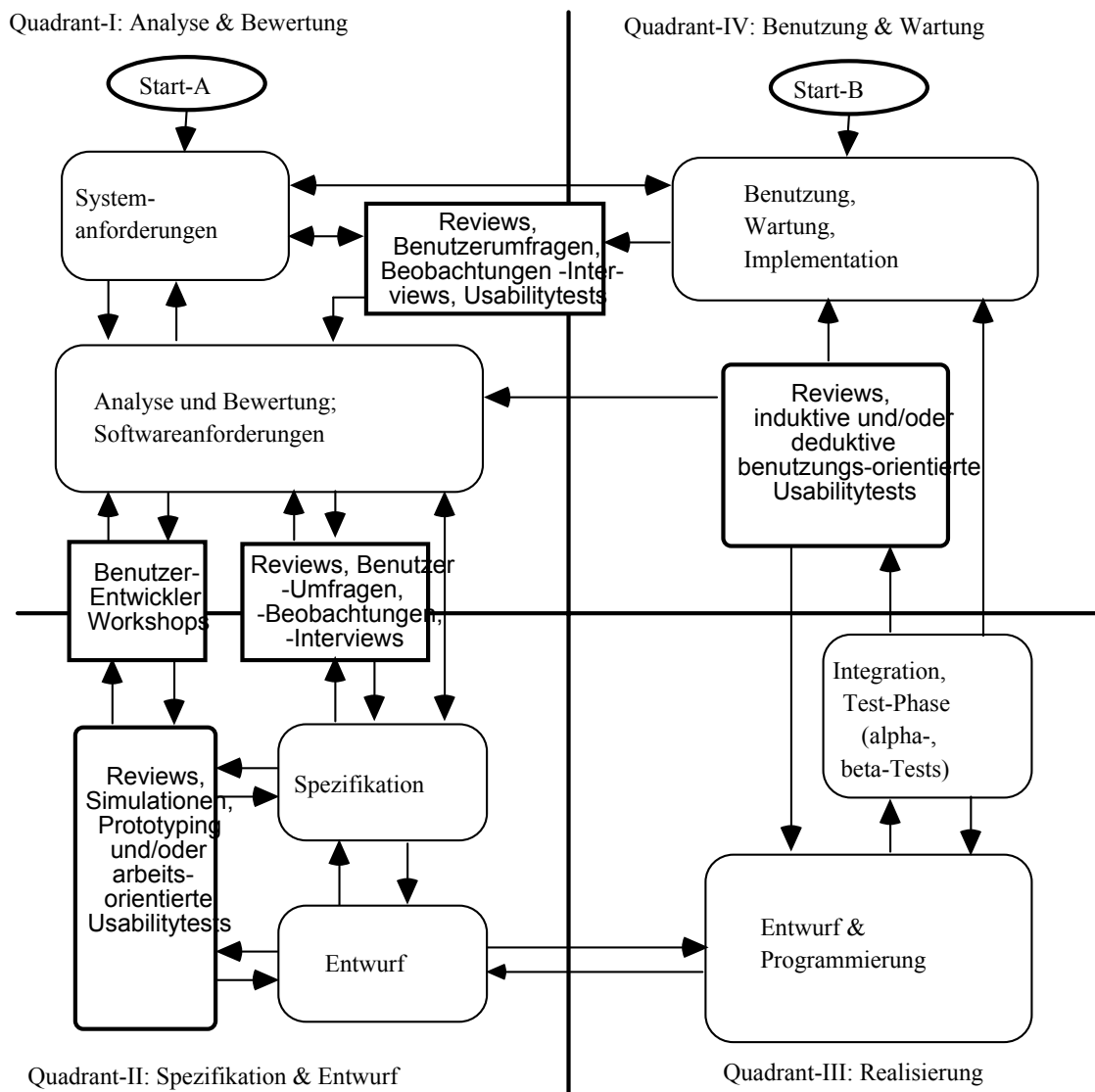


Abbildung 2: Ablaufmodell eines partizipativen Softwareentwicklungskonzeptes mit einer Übersicht über die verschiedenen Optimierungszyklen innerhalb und zwischen den einzelnen Quadranten (siehe Rauterberg et al. 1994).

Wir haben ein iterativ-zyklischen Ablaufmodell entwickelt, welches an verschiedenen Stellen im Entwicklungsprozeß die Beteiligung von BenutzerInnen vorsieht (siehe Abb. 2). Der gesamte Prozeß ist dabei in vier Quadranten aufgeteilt: (Quadrant-I) Analyse und Bewertung des Arbeitssystems; (Quadrant-II) Spezifikation und Entwurf des EDV-Systems; (Quadrant-III) Realisierung der technischen Komponenten; (Quadrant-IV) Benutzung und Wartung des EDV-Systems. Für die Erstellung der verschiedenen Zwischenergebnisse stehen eine Reihe von unterschiedlichen Methoden zur Verfügung (siehe Tab. 1).

Tabelle 1: Übersicht über die verschiedenen Möglichkeiten zur Benutzerpartizipation.

Methoden	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	Tage - Wochen
	vollständig vertikales Prototyping	aufgaben-orientierte Usability-Tests	Spezifikation der Anwendungskomponente	Tage - Wochen
Versioning	Erster Durchlauf des gesamten Entwicklungszyklus	induktive Usability-Tests deduktive Usability-Tests	erste weitgehend vollständige Version	Monate - Jahre
	Weiterer Durchlauf des gesamten Entwicklungszyklus	induktive Usability-Tests deduktive Usability-Tests	mehrere weitgehend vollständige Versionen	Monate - Jahre

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.

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 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.

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.

Kasten 4: Beispiel 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.

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 die Information vom Benutzer bis zum Entwickler zurücklegen muß; 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.

Die Auswahl der BenutzervertreterInnen 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!

Der evaluative Ansatz mittels Checklisten

Checklisten sind speziell für die Bewertung der Benutzungsfreundlichkeit entwickelt worden und sind zumeist in einer englischen Fassung erhältlich. Deutschsprachige Checklisten sind in Baitsch et al (1989), EDV im Büro (1990), Siemens Nixdorf Styleguide (1992), sowie ISO 9241/10 (1993) 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.

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 Widersprüchen 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 optisch 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 erleichtert 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, gesprächs begleitenden Aufzeichnung kann somit gelöst werden.

Der benutzungsorientierte Ansatz mittels Usability-Tests

Es werden drei Arten von Usability-Tests zur partizipativen Entwicklung von [Standard]-Software vorgestellt: (1.) aufgabenorientierte, (2.) induktive und (3.) deduktive 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). Diese drei Arten lassen sich sowohl in einem speziellen Usability-Labor als auch vor Ort am Arbeitsplatz durchführen.

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

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

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 Usability-Tests erfolgt in der Regel in einem speziell eingerichteten Usability-Labor (siehe Nielsen 1994), kann aber auch unter besonderen Bedingungen am Arbeitsplatz stattfinden.

Die *induktiven* Usability-Tests sind bei der Evaluation einer (Vor)-Version zur Gewinnung von Gestaltungs- und Verbesserungsvorschlägen, bzw. zur Analyse von Schwachstellen in der Benutzbarkeit einsetzbar. Induktive Usability-Tests können immer dann zum Einsatz kommen, wenn nur *eine* Version der zu testenden Software vorliegt. Demgegenüber verfolgen *deduktive Usability-Tests* primär den Zweck, zwischen mehreren Alternativen (mindestens zwei Prototypen, bzw. Versionen) zu entscheiden. Zusätzlich lassen sich jedoch auch mit deduktiven Usability-Tests Gestaltungs- und Verbesserungsvorschläge gewinnen.

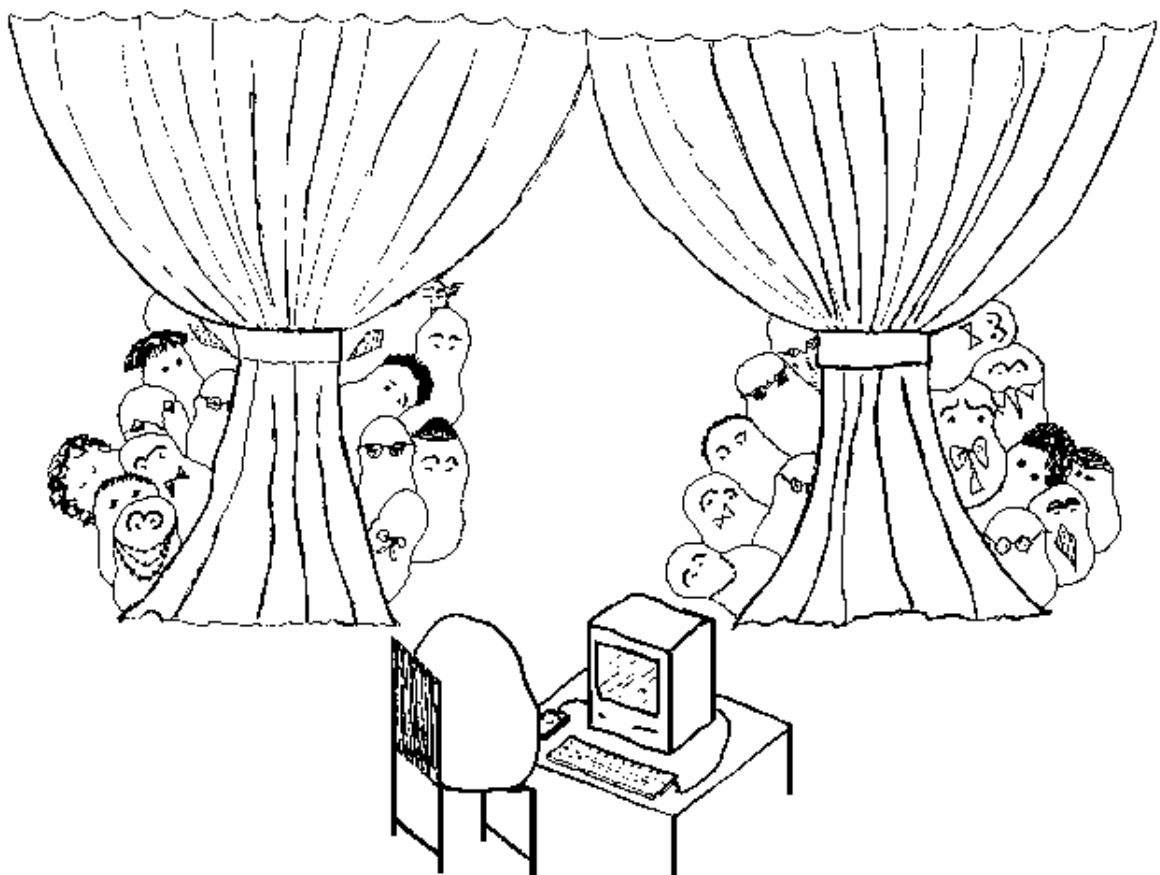


Abbildung 3: Karikaturistische Darstellung eines Usability-Tests mit allen Betroffenen.

Aufgabenorientierte Usability-Tests

Wie bei den Szenarien wird bei aufgabenorientierten Usability-Tests eine möglichst realistische Arbeitssituation nachgestellt. Der aufgabenorientierte Usability-Tests läßt 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 Computeratrasse, 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 Usability-Test direkt am Arbeitsplatz oder in unmittelbarer Nähe dazu durchführen.

Induktive Usability-Tests

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

Damit keine unnötigen Fehler während eines induktiven Usability-Tests - 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 Bearbeitungszeit einer (Teil-)Aufgabe liegt in der Regel zwischen fünf und fünfzehn Minuten.

Im Unterschied zum aufgabenorientierten Usability-Test sind beim induktiven und deduktiven Usability-Test *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.

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.

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 unterteilt sich in die Menge der *qualitativen* Meßgrößen

(problematische Benutzungssituationen, Fehlerarten, etc.) und die Menge der *quantitativen* Meßgrößen (Bearbeitungsdauer, Anlernzeit, Planungszeit, Anzahl Fehler).

Während der Durchführung eines Usability-Tests 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!

Deduktive Usability-Tests

Deduktive Usability-Tests 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 Usability-Tests.

Im Gegensatz zu induktiven Usability-Tests müssen die verschiedenen zu testenden, alternativen Systemversionen beim deduktiven Usability-Test 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.

Kosten–Nutzen Analysen von Usability-Tests

Wenn man deduktive Usability-Tests durchführt und Performanzmaße als abhängige Meßgrößen erhebt, kann man den erreichten Grad der Verbesserung auch in finanzieller Hinsicht abschätzen. Hierzu ist eine erweiterte Kosten-Nutzen Betrachtung notwendig. Die finanziellen Aufwendungen, welche durch den Einsatz von Usability-Tests entstehen, werden den zu erwartenden Einsparungen beim Gebrauch der verbesserten Version gegenübergestellt.

Es lassen sich folgende Berechnungsgrößen für eine erweiterte Kosten-Nutzen-Analyse zugrunde legen, welche zunächst investiert werden müssen:

- | | |
|---|-----|
| • Kosten für den Testleiter (als Jahresgehalt) | KTL |
| • Anzahl an eingesetzten Testleitern | ATL |
| • Zeitdauer der Durchführung (in Personen-Monaten) | ZPM |
| • Kosten für die Teilnahme pro TestpartnerIn | KTP |
| • Anzahl der teilnehmenden TestpartnerInnen | ATP |
| • Kosten für zusätzliche Geräte (Videoanlage, etc.) | KZG |
| • Kosten für sonstige Ausgaben (Sekretariat, etc.) | KSA |

Formel 1 zur Berechnung der gesamten Investitionskosten:

$$IGK = (ATL * KTL / 12) * ZPM + ATP * KTP + KZG + KSA$$

Zur Berechnung des Grades der erreichten Verbesserung (VG) wird der Quotient aus dem ausgewählten Performanzmaß (PM; z.B. die Test-Bearbeitungs-Zeit "TBZ") für die neue und die alte Software-Version gebildet:

Formel 2 zur Berechnung der Verbesserungs-Grades der neuen Version:

$$VG = 100\% - (PM_{\text{neu}} / PM_{\text{alt}}) * 100\%$$

Die folgenden Berechnungsgrößen können als Grundlage zur Abschätzung der Einsparungen durch den Einsatz der verbesserten Softwareversion herangezogen werden:

- | | |
|--|-----|
| • Anzahl der End-Benutzer | AEB |
| • Durchschnittlicher Stundenlohn des End-Benutzers | DSL |
| • Anzahl an Arbeitsstunden der End-Benutzer | AAS |
| • Anschaffungspreis der Software | APS |
| • "Upgrade"-Preis für die verbesserte Version | UPS |
| • Verbesserungs-Grad (Angabe in Prozent) | VG |

Formel 3 zur Berechnung der durchschnittlichen Kosten *aller* End-Benutzer:

$$DBK_{\text{alt}} = (AEB * (PM_{\text{alt}} / PM_{\text{alt}}) * DSL) * AAS$$

$$DBK_{\text{neu}} = (AEB * (PM_{\text{neu}} / PM_{\text{alt}}) * DSL) * AAS$$

Um den "return of investment" abzuschätzen, kann man das folgende Verfahren anwenden. Zuerst berechnet man DBK_{alt} (nach Formel 3) mit der alten Version. Dann bestimmt man anhand der gemessenen Performanz-Maße im Usability-Test den Grad der Verbesserung (VG) in Prozent (z.B. der Quotient aus der Bearbeitungszeit (TBZ) mit der neuen und der alten Version). Wird nun der Wert von DBK_{alt} mit dem Prozentwert multipliziert, erhält man DBK_{neu} . Die Differenz $EK = (DBK_{\text{alt}} - DBK_{\text{neu}})$ ergibt die eingesparten Kosten (EK). Diesem Wert wird der Wert IGK gegenübergestellt. Da man jedoch die Größe AAS in Formel 3 praktisch nicht genau angeben kann, empfiehlt es sich, einen Kosten-Nutzen-Graphen (Abb. 4) aufzuzeichnen. Aus dieser Graphik kann man dann unmittelbar die minimale Anzahl Arbeitsstunden (MAAS) der Population aller End-Benutzer ablesen, ab derer sich das Arbeiten mit der neuen Version für die Gruppe aller End-Benutzer "bezahlt" gemacht hat.

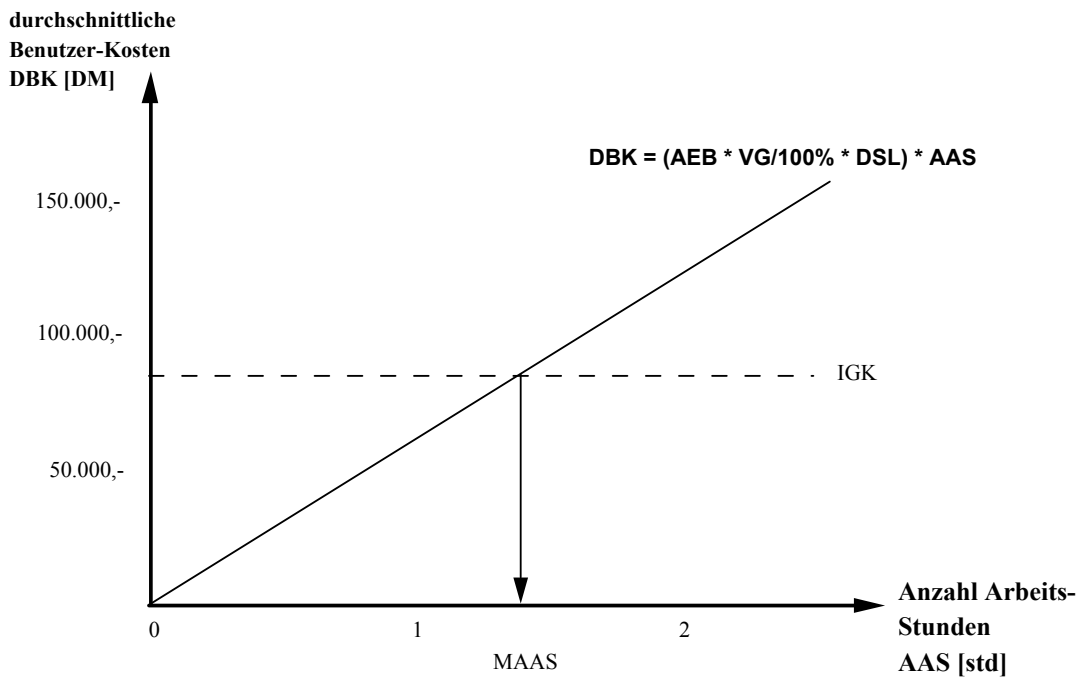


Abbildung 4: Der Kosten-Nutzen-Graph für die Abschätzung der minimalen Anzahl Arbeitsstunden (MAAS) bzgl. aller End-benutzer.

Mittels des Kosten-Nutzen-Graphes (Abb. 4) kann man lediglich die minimale Anzahl Arbeitsstunden für die *gesamte* Population der End-Benutzer ermitteln. Für den einzelnen End-Benutzer sieht der zu leistende minimale Arbeitsaufwand (MAAS) etwas anders aus. Der einzelne End-Benutzer muss nämlich erst den Anschaffungs-Preis der Software (APS; bzw. den Upgrade-Preis UPS) "erwirtschaften".

Formel 4 zur Berechnung der durchschnittlichen Kosten *eines* End-Benutzers:

$$EBK_{alt} = ((PM_{alt} / PM_{alt}) * DSL) * AAS + UPS$$

$$EBK_{neu} = ((PM_{alt} / PM_{neu}) * DSL) * AAS$$

Tragen wir die beiden Geradengleichungen aus Formel 4 in den individuellen Kosten-Nutzen-Graph für den einzelnen End-Benutzer ein, so können wir die minimale Anzahl Arbeitsstunden (MAAS_{APS}; bzw. MAAS_{UPS}) ermitteln (Abb. 5). Die Steigung der Geraden EBK_{neu} setzt sich aus dem geschätzten durchschnittlichen Stundenlohn gewichtet mit dem erreichten Verbesserungs-Grad (PM_{alt}/PM_{neu}) zusammen. Wenn z.B. VG = 50% ist, dann ist der Benutzer (PM_{alt} / PM_{neu}) = 2.0 mal so schnell mit der verbesserten Version. Bei einem durchschnittlichen Stundenlohn von 12,50 [DM/Std] benötigt der Neueinsteiger MAAS_{APS} = 30 [Std] Arbeitsstunden (EBK_{neu} = APS; d.h.: MAAS_{APS} = (APS / DSL) / (PM_{alt} / PM_{neu})), um den Anschaffungs-Preis (z.B. APS = 750,- DM) zu erwirtschaften. Für den Umsteiger dagegen ist die Gewinnzone [EBK_{neu} > EBK_{alt}] erst nach EBK_{neu} = EBK_{alt} erreicht ; d.h.:

$MAAS_{UPS} = (UPS / DSL) / ((PM_{alt} / PM_{neu}) - 1) \Rightarrow MAAS_{UPS} = 40$ [Std] Arbeitsstunden (siehe Abb. 5).

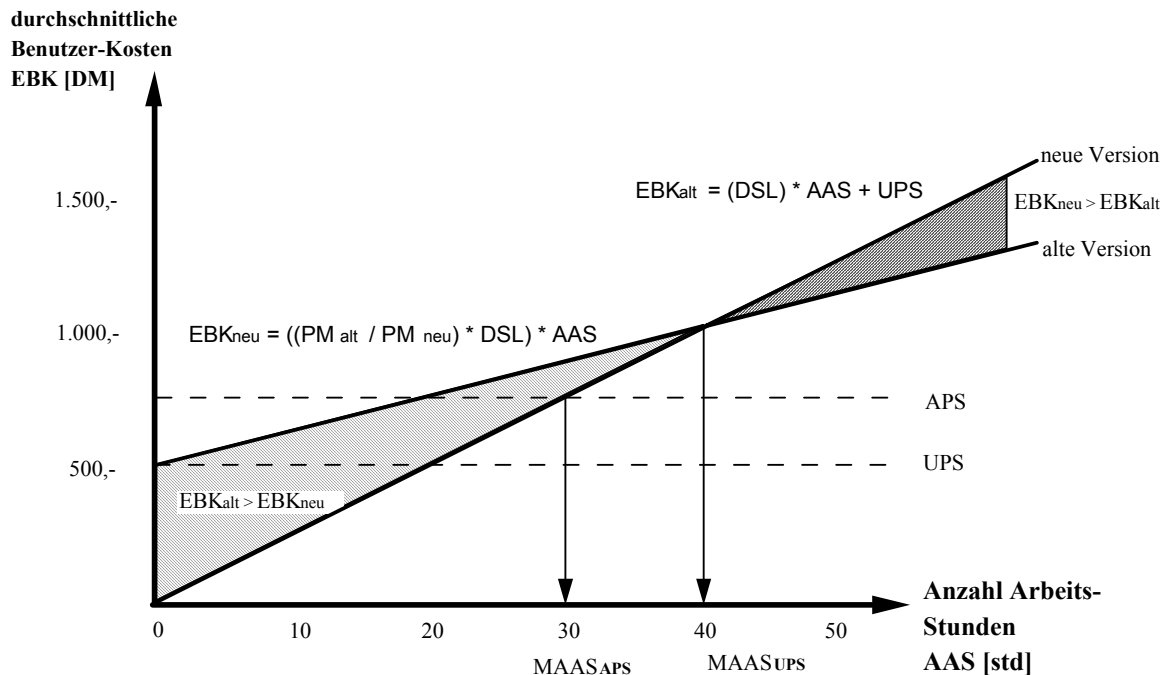


Abbildung 5: Der Kosten-Nutzen-Graph für die Abschätzung der minimalen Anzahl Arbeitsstunden (MAAS), ab derer sich die investierten Kosten (UPS, bzw. APS) für jeden einzelnen End-Benutzer bei einem durchschnittlichen Stundenlohn (DSL) bezahlt gemacht haben.

Damit nun der Umsteiger nicht mehr als der Neueinsteiger investieren muß ($MAAS_{UPS} > MAAS_{APS}$, siehe Abb. 5), gilt die folgende Bedingung: $MAAS_{UPS} = MAAS_{APS}$; dies kann erreicht werden, indem der Neupreis $APS = [1 - (PM_{neu} / PM_{alt})] * UPS$ beträgt. Durch diese Gewichtung des Upgrade-Preises wird erreicht, daß der Neupreis im Verhältnis zu den vom Umsteiger zu investierenden Kosten vergleichbar wird.

Der erreichte Verbesserungsgrad einer neuen graphischen Oberfläche für eine Standardsoftware gegenüber der alten Version betrug in einem konkreten Beispiel 35%. Die Benutzer sind mit der verbesserten Benutzungsoberfläche ungefähr 1.5 mal schneller als mit der alten Oberfläche. Bei ca. 35.000 Benutzern ergibt sich bei einem durchschnittlichen Stundenlohn von 15,- DM und dem Verbesserungsgrad von 35% eine minimale Anzahl Arbeitsstunden (MAAS) von 0,35 [Std]; wenn also alle 35.000 Benutzer nur 21 Minuten mit der neuen Version arbeiten würden, haben sich die Investitionskosten in diesen Usability-Test insgesamt gesehen ausgezahlt.

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 Produktes werden dabei maßgeblich von der Projektorganisation, den fachlichen und vor allem sozialen Qualifikationen der Beteiligten sowie den vorhandenen Methoden und Werkzeugen beeinflusst.

Die Methode der Usability-Tests läßt sich nicht nur zur Gewinnung von Gestaltungsvorschlägen, sondern auch zur Entscheidung zwischen Systemalternativen, bzw. zur Überprüfung von getroffenen Designentscheidungen sinnvoll im Rahmen der partizipativen Entwicklung von Software einsetzen. Wenn man die zu investierenden Kosten für die Methode des Usability-Tests mit denjenigen Kosten vergleicht, die sich bei der Benutzung der verbesserten Version ergeben, so wird man feststellen müssen, daß sich der Einsatz von Methoden zur Benutzerbeteiligung außerordentlich schnell auszahlt.

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.

Literatur

- Baitsch, C., Katz, C., Spinas, P. & Ulich, E. (1989). Computerunterstützte Büroarbeit. Zürich: Verlag der Fachvereine.
- Dumas, J. & Redish, J. (1993) A practical guide to usability testing. Ablex Publishing.
- EDV im Büro (1990) Handbuch zur menschengerechten Gestaltung. München: Oldenbourg.
- ISO 9241/10 (1993) Fragebogen zur Beurteilung von Software auf der Grundlage der Internationalen Ergonomie-Norm ISO 9241/10. Dr. Prümper & Partner, Holzhofenstrasse 8, D-81667 München.
- Nielsen, J. (1993) Usability Engineering. Amsterdam: 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. Zürich: 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. Stuttgart: Poeschel Verlag.

Benutzungsoberflächen und ihr Einfluß auf die Akzeptanz und Nutzung von DV

Matthias Rauterberg

1994



Gebrauchstauglichkeit
(Utility)

ist gleich

Benutzbarkeit
(Functionality)

plus

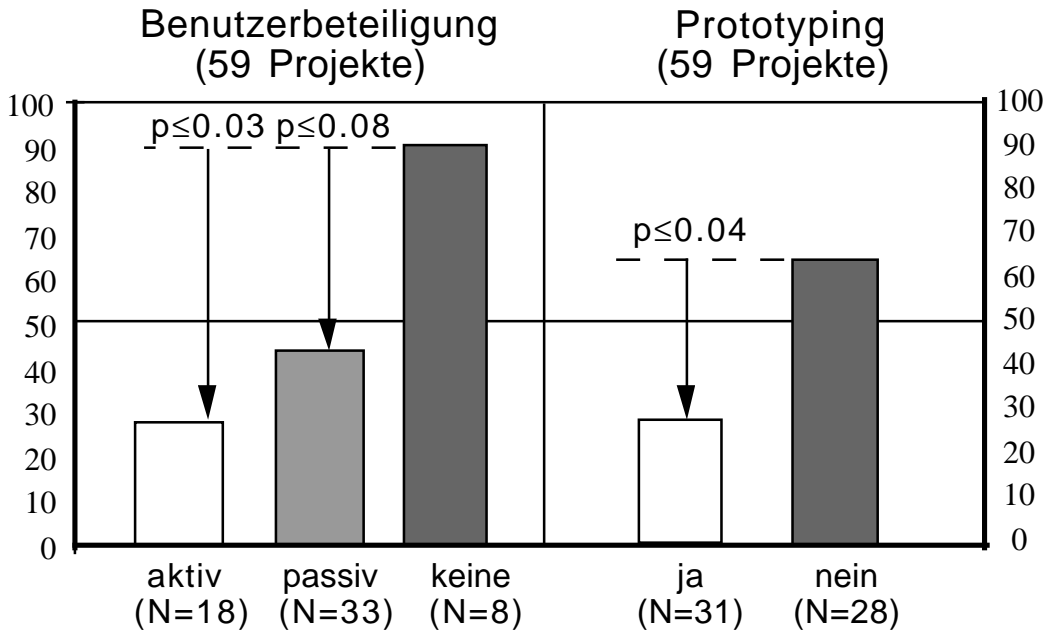
Benutzungsfreundlichkeit
(Usability)



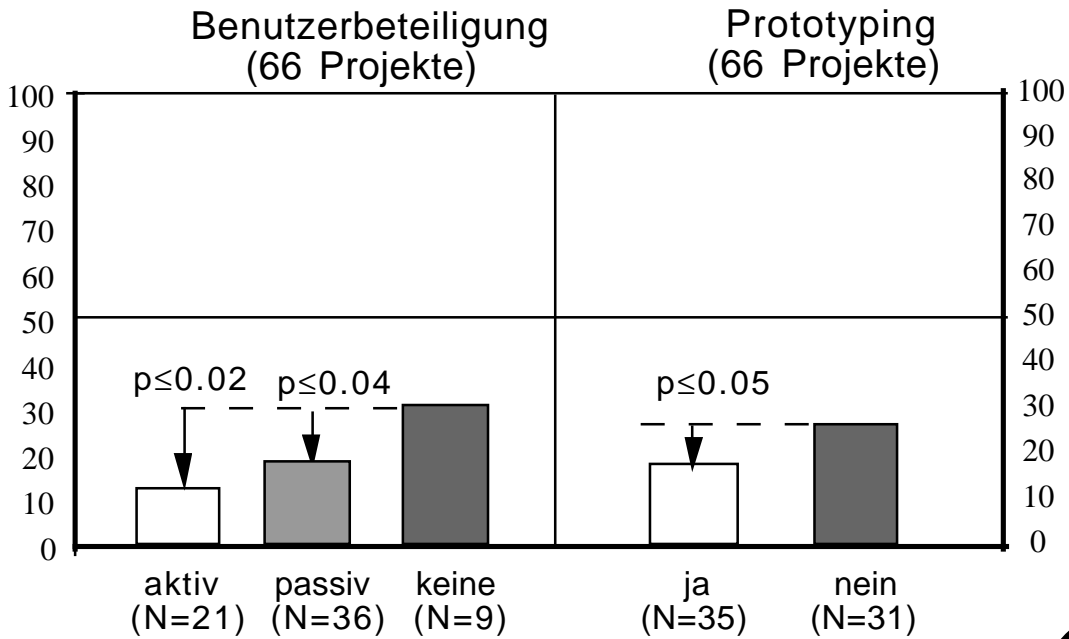
Vorteile von Benutzerbeteiligung

[Rauterberg & Strohm 1992]

% Kostenüberschreitung

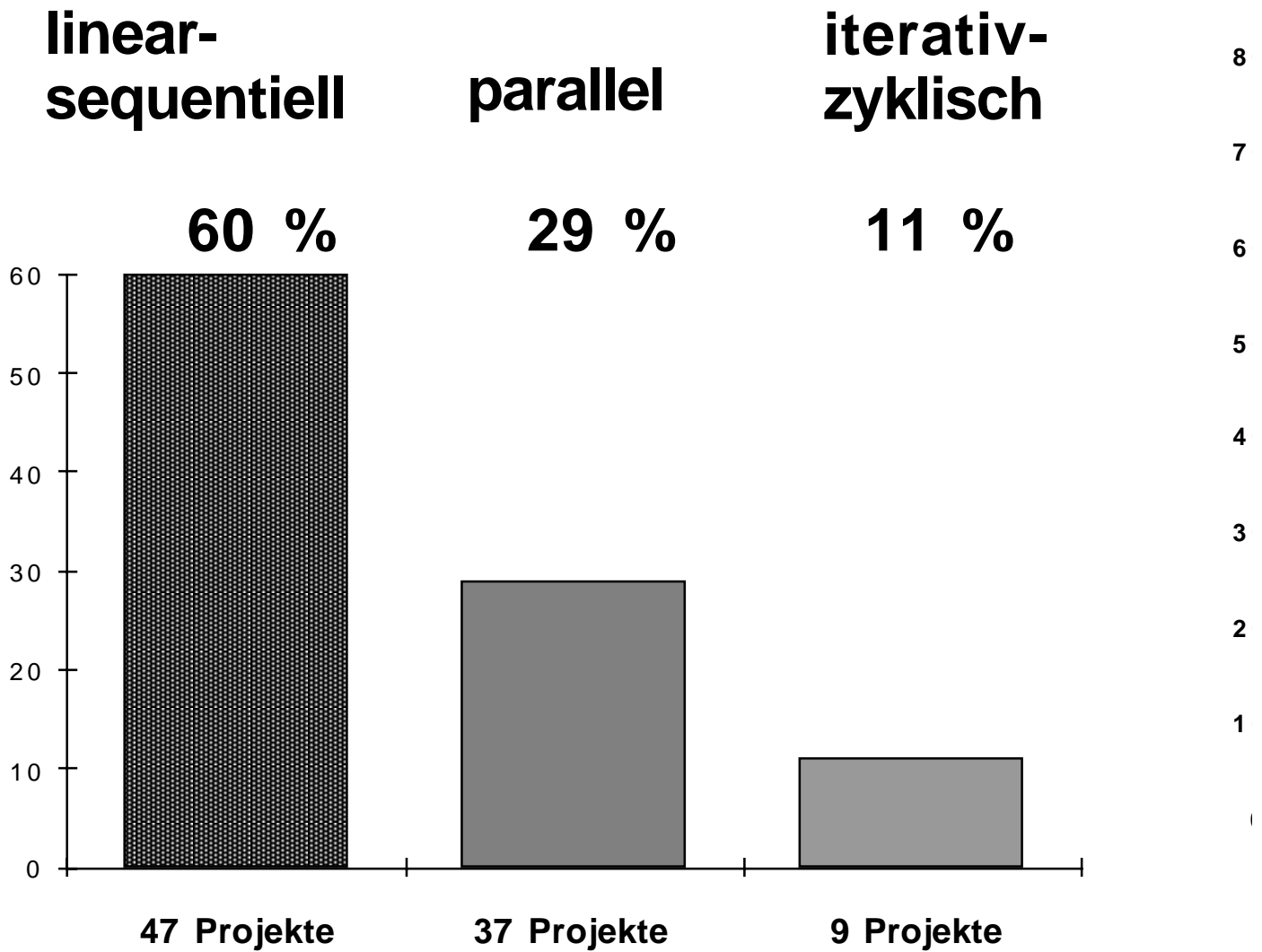


% Zeitüberschreitung



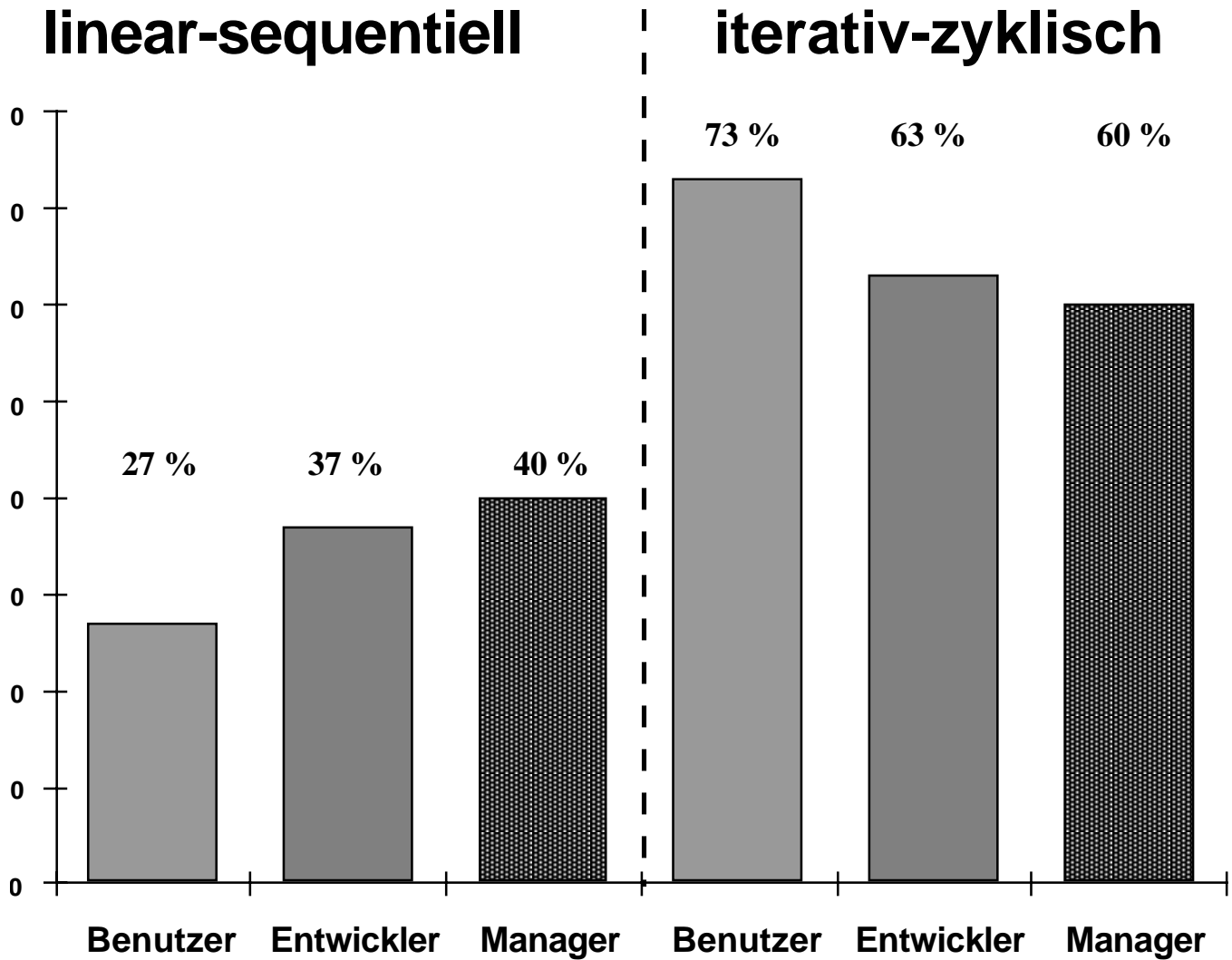
Typ des Ablaufmodells (IST-Vorstellung)

[STROHM 1990, analysierte Projekte N=79]



Typ des Ablaufmodells (SOLL-Vorstellung)

[SPINAS & WAEBER, 1991, analysierte Firmen N=10]



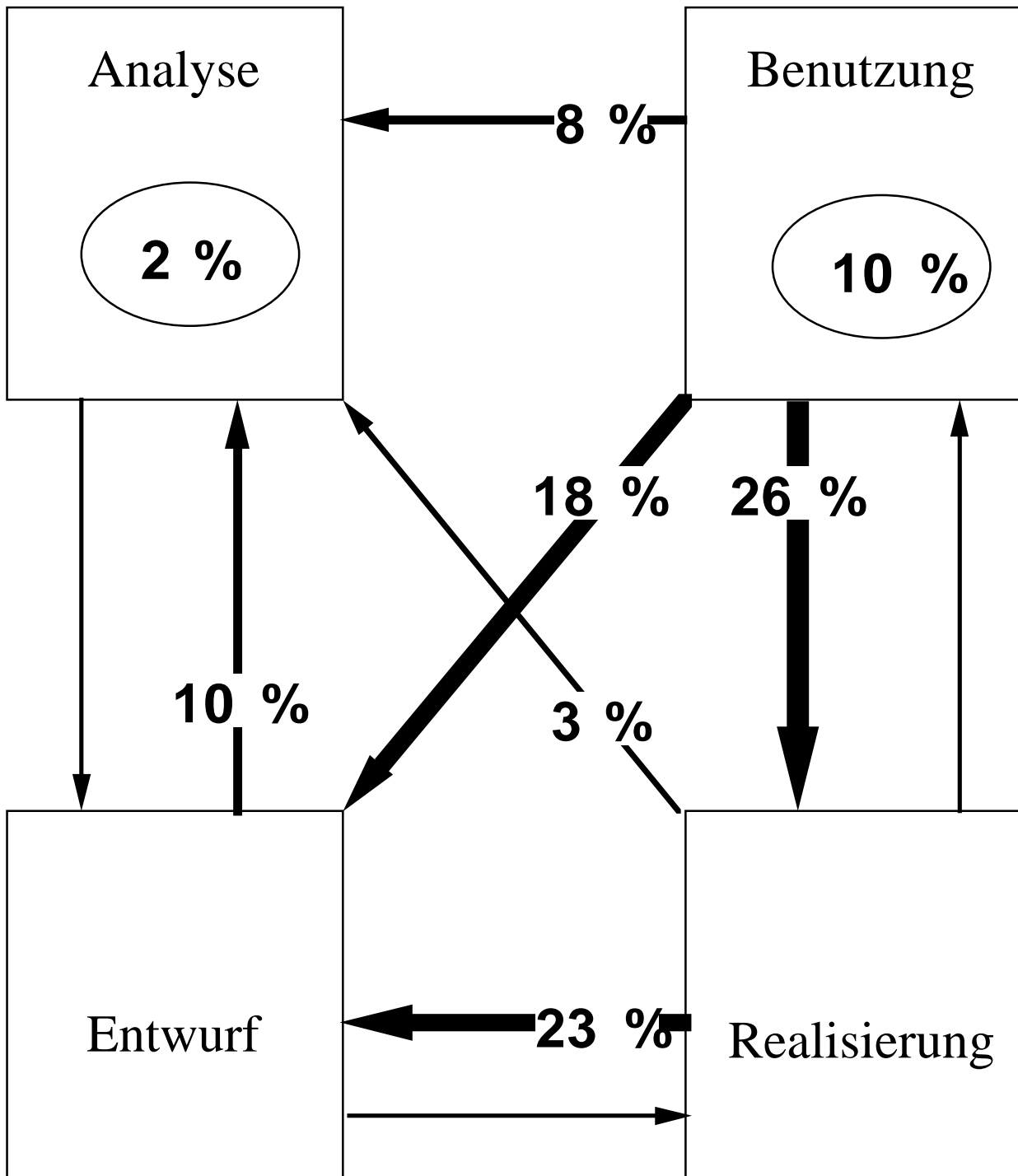
N=157 befragte Personen:

N= 65 befragte Benutzer,
N= 61 befragte Entwickler,
N= 31 befragte Manager



Häufigkeit von Phasenrückschritten

(IPAS-Projekt, Hesse 1993)



Stichprobe:

39 Berichte von expliziten

Schleifen

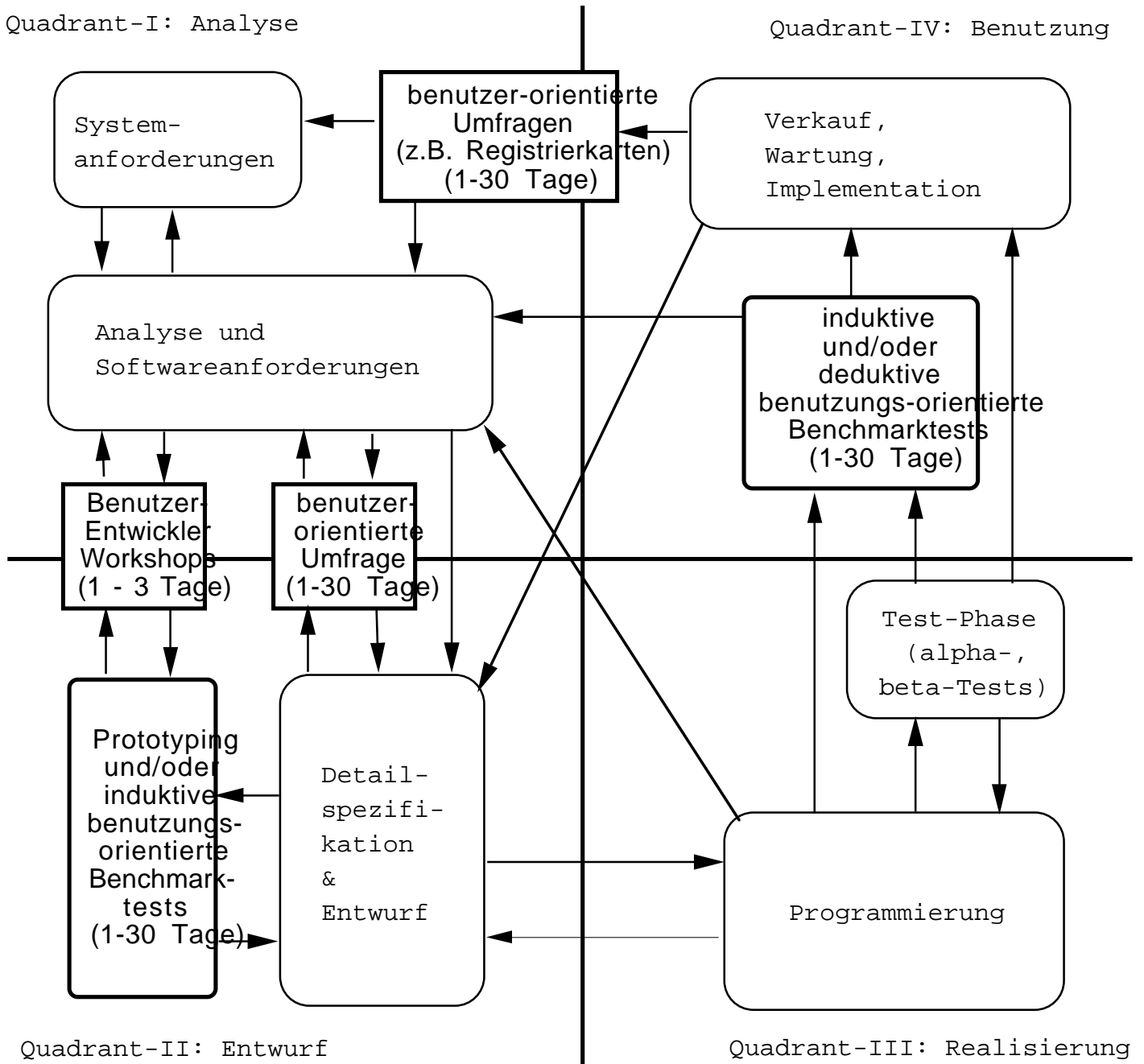
29 Software-Projekte

156 Antworten zum Phasenverlauf



Das Quadranten-Modell

[BOSS-Projekt, Rauterberg 1991]



Was ist ein Usability-Test ?

Ein Usability-Test ist...

eine Methode zur Erfassung und Messung
der interaktiven Eigenschaften
eines interaktiven Systems.



Wer kann Usability testen ?

Jeder mit ...

- Interesse an einer inter-aktiv(er)en Software,
- Neugierde an dem Verhalten von End-Benutzern,
- Aufgeschlossenheit gegenüber Kritik.

Hilfreiche Kenntnisse sind...

- software-ergonomisches Wissen,
- experimental-psychologische Grundlagen,
- angewandte Statistik.



Wozu dienen Usability-Tests ?

Formative Evaluation:

- => Hinweise auf interaktive Schwachstellen;
- => Entdeckung von interaktiven "Deadlocks";
- => Gestaltungshinweise zur Dialogstruktur.

Summative Evaluation:

- => Hinweise auf interaktive Schwachstellen;
- => Entdeckung von interaktiven "Deadlocks";
- => Gestaltungshinweise zur Dialogstruktur;
- => Entscheidung zwischen System-Varianten;
- => Kontrolle von Gestaltungsmaßnahmen.



der interaktions-zentrierte Meß-Ansatz

**He !
Ich Chef - du Werkzeug !
Begreifen ?**



Usability-Test 1

- Sie fahren nach Österreich in die Ferien.

Für die ersten Tage benötigen Sie Bargeld.

Wie hoch ist der aktuelle Wechselkurs für den Österreichischen Schilling?

Ihre Antwort:



Usability-Test 2

- Am 1. Juli 1991 wurde auf dem K o n t o '3999999-31' eine Buchung durchgeführt.

Wie hoch ist der Betrag?

Ihre Antwort:

- Welche Buchungen sind vorgemerkt?

Ihre Antwort:



Usability-Test 3

Situation:

Heute ist der 9. Januar 1992 und Sie sind Wertschriften-Sachbearbeiter bei der Bankstelle Zürich-Hauptsitz, zuständig für alle Depotbewegungen auf den Kundendepots.

Ausgangslage:

Wegen Differenzen mit der Dresdner Bank soll die Depotstelle 'Dresdner Bank, Frankfurt' (DST 26(00)) aufgelöst werden. Dies bedeutet, dass Kundenbestände, die bei dieser Depotstelle lagern, umgelagert werden müssen.

Tatbestand:

Herr Amsler, Liebhaber von Bier und Autos, wohnhaft in Zürich, ist Deponent bei der Bankstelle 0835 und weist folgenden Depotbestand aus (Depotnummer: 0835-258310-0):

V-Nr.	Valoren-Text.	DST/EDST2
335915	Akt. Henninger-Bräu AG	2600
335915	Akt. Henninger-Bräu AG	2600
	Zusatztext: zugunsten Olga Amsler	
154208	N-Akt. Brau. Feldschlöss.	9903

Nach der Umlagerungsaktion sollen seine Bestände wie folgt liegen:

335915	Akt Henninger-Bräu AG	2605
335915	Akt. Henninger-Bräu AG	2605
	Zusatztext: zugunsten Olga Amsler	
154208	N-Akt. Brau. Feldschlöss.	9903

Abwicklung:

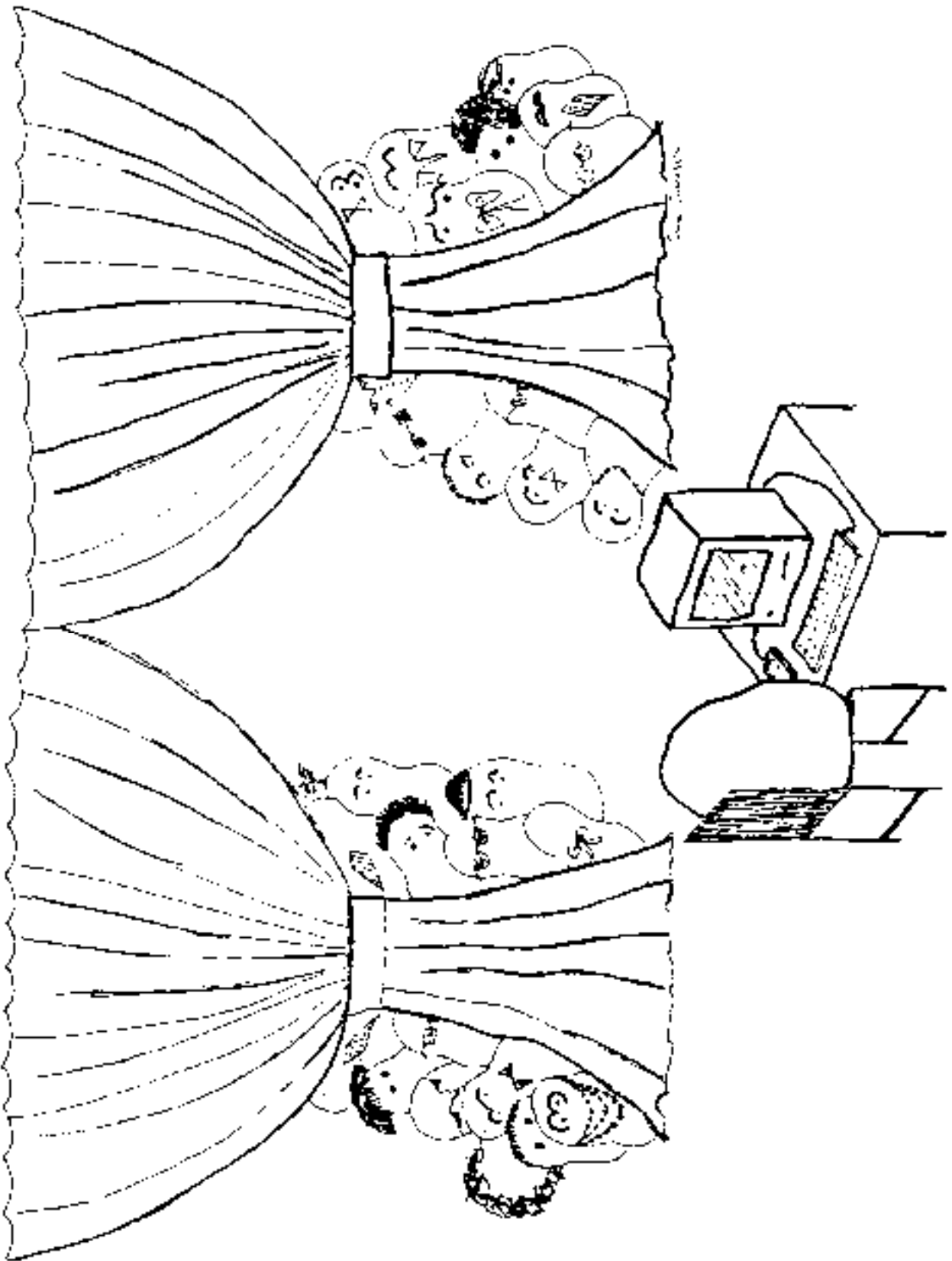
Die Volkswagen-Aktien sollen via SEGA umgelagert werden, die restlichen Titel physisch.

Es sollen die ordentlichen Spesen belastet werden (Ausnahme: die Umlagerung der Volkswagen-Aktien soll spesenfrei erfolgen).

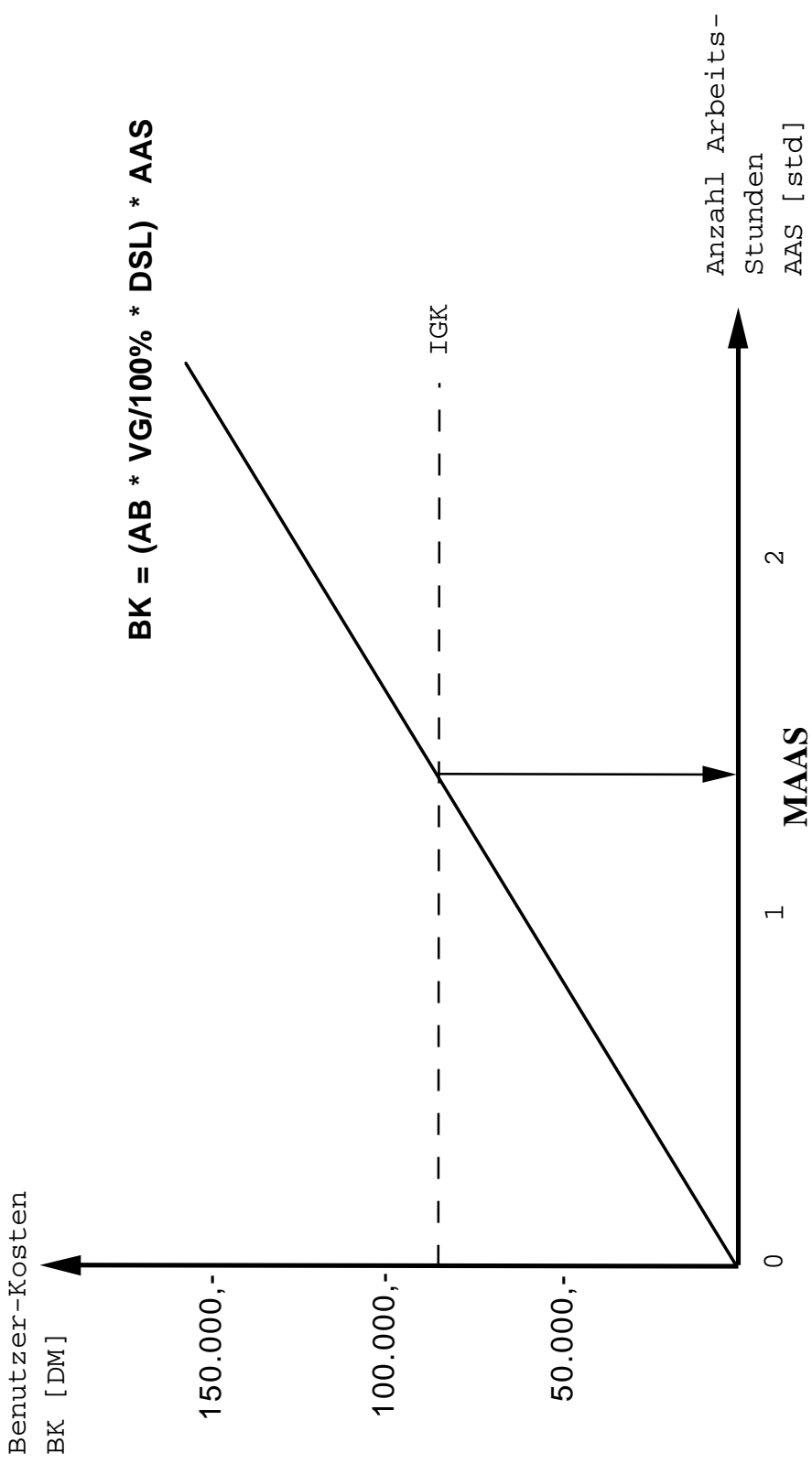


DIN 66 234 Teil 8 (1988)	ISO 9241 Entwurf (1991)	EG Richtlinie 90/270/EWG (1990)	VDI Richtlinie 5005 (1990)	ISO 9241 Entwurf (1991)
Aufgabengemessenheit	Aufgabengemessenheit	Tätigkeitsangepasstheit	Aufgabengemessenheit	Aufgabengemessenheit
Selbsterklärungsfähigkeit	Selbsterklärungsfähigkeit	Angaben über die jeweiligen Systemabläufe		Selbsterklärungsfähigkeit
Erwartungskonformität	Erwartungskonformität	Format- und temporechte Informationsdarstellung		Erwartungskonformität
	Erlernbarkeit		Kompetenzförderlichkeit	Erlernbarkeit
	Individualisierbarkeit	Anpassbarkeit an Kenntnis- und Erfahrungsstand		Individualisierbarkeit
Steuerbarkeit	Steuerbarkeit		Handlungsflexibilität	Steuerbarkeit
Fehlerrobustheit	Fehlertoleranz			Fehlertoleranz
		Unterrichtung und Unterweisung		
		Anhörung und Beteiligung		





Kosten-Nutzen-Graph für Usability-Tests



Legende:

- IGK = investierte Gesamtkosten, MAAS = minimale Anzahl Arbeitsstunden,
- BK = totale Benutzungskosten, AB = Anzahl Benutzer,
- VG = Verbesserungsgrad in %, DSL = durchschnittlicher Stundenlohn



**Läßt sich die
Gebrauchstauglichkeit
messen?**

Ja

**man muß es nur
tun...**

