

Vorlesung

Human Factors in Engineering

Teil 2, 1. Hälfte

(Folien für SS 2011)

Dozent: Prof. Dr. Thomas Stoffer

Themen der ersten Hälfte der Vorlesung:

1. Methoden ergonomischer Untersuchungen
2. Grundlagen der Software-Ergonomie

Folien downloaden:

www.psy.lmu.de/exp/index/de.html/education/courses/index.html

Literatur:

Wickens, C.D., Lee, J.D., Liu, Y. & Becker, S.E.G. (2004). An introduction to human factors engineering. Upper Saddle River, NJ: Pearson, 2nd Ed. (hieraus nur Kap. 2, 3 und 15)

Sarodnick, F. & Brau, H. (2006). Methoden der Usability Evaluation. Bern: Huber.

Kapitel 1: Methoden ergonomischer Untersuchungen

- Viele Produkte und technischen Systeme werden auch heute noch weitgehend ohne Berücksichtigung des menschlichen Faktors und ergonomischer Gesichtspunkte gestaltet.
- Designer konzentrieren sich i.d.R. auf technische und gestalterische Aspekte: Wenn es
 - funktioniert,
 - billig herzustellen ist und auch noch
 - passabel aussieht,sind alle Ziele des Designers erreicht.

- Heutige Geräte und technischen Systeme (z.B. Software) werden u.a. aufgrund von viel zu zahlreichen Funktionen immer komplizierter zu benutzen.

- **Das muss nicht sein:** Wenn man spätestens nach, aber besser noch (wegen der Entwicklungskosten) schon **während der Entwicklung** empirisch prüft, wie schwierig und fehleranfällig eine Benutzung ist, kann man dem leicht entgegenwirken.

→ Wir wollen uns im Folgenden mit den wichtigsten Methoden beschäftigen, die im Verlauf einer solchen empirischen Überprüfung eingesetzt werden können.

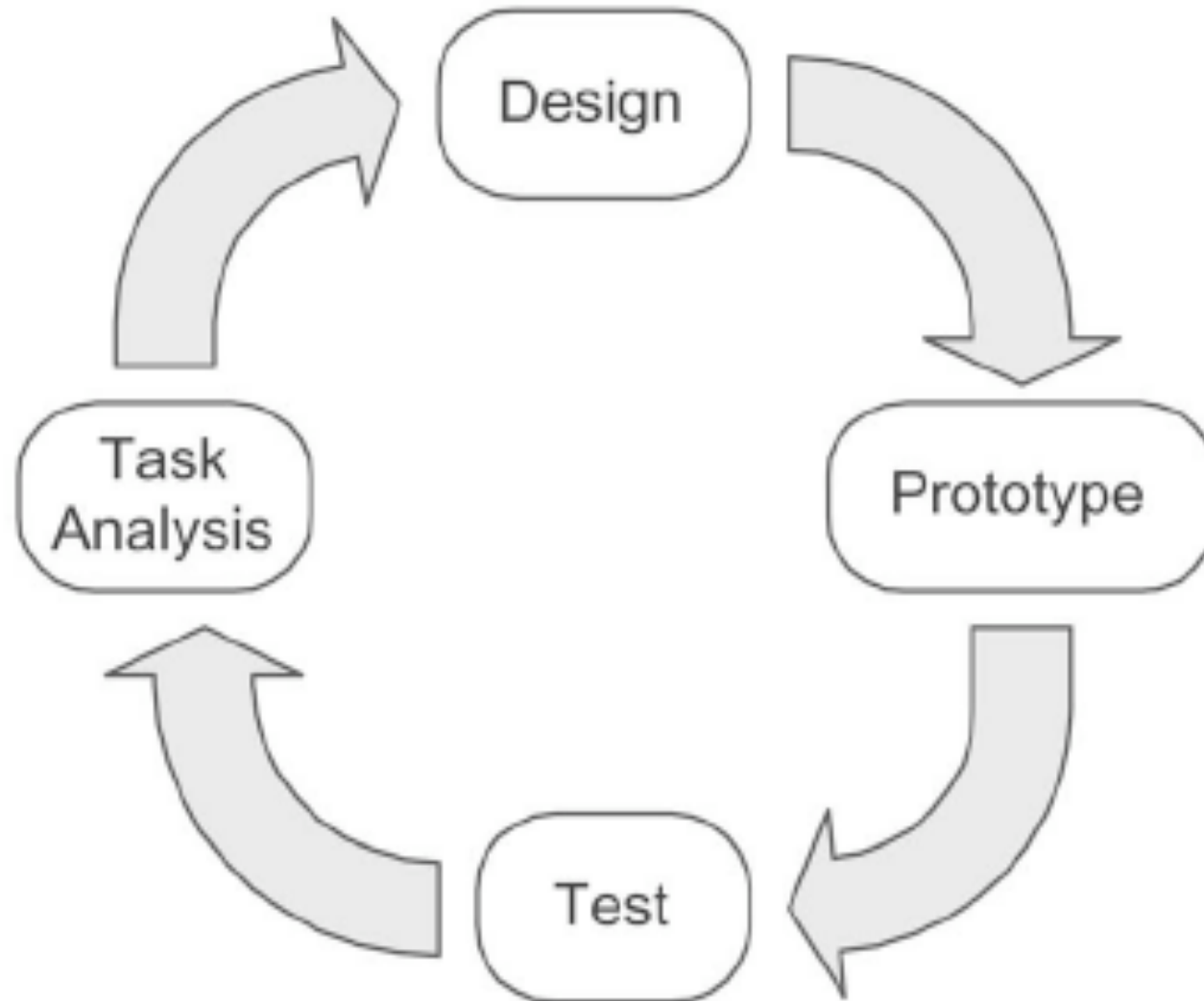
Benutzerzentriertes Design (User-Centered Design):

Ideal: Das Design eines Gebrauchsgegenstandes oder die Gestaltung eines Arbeitsplatzes und der Arbeitstätigkeiten sollten grundsätzlich *benutzerzentriert* erfolgen (Norman & Draper, 1986), d.h. der Benutzer solle Maßstab aller Designentscheidungen sein:

- seine körperliche Unversehrtheit,
- seine Bedürfnisse,
- die Begrenzungen seiner Handlungsmöglichkeiten,
- seine Zufriedenheit,
- seine mentalen und körperlichen Belastungen

müssen mit Priorität berücksichtigt werden;

technische und wirtschaftliche Gesichtspunkte dürfen nie dominant, erst recht nicht alleinige Richtschnur des Designs sein.



Daraus zu ziehende Konsequenzen:

- Was sind die Bedürfnisse des Benutzers?

Z.B. → Einfachheit der Handhabung vs. Multifunktionalität

→ Stabilität vs. preisgünstige Produzierbarkeit

→ abwechslungsreiche Tätigkeit vs. Monotonie

- Welche Aufgaben muss der Benutzer bewältigen, und unterstützt ihn dabei das zu entwickelnde Gerät bzw. die zu gestaltende Arbeitsumgebung?

Z.B. → Wird der Benutzer behindert oder wird seine Tätigkeit erleichtert ?

→ Muss sich der Benutzer an den Designgegenstand anpassen oder passt sich dieser dem Benutzer an?

Prinzipielles Vorgehen:

- **Früher Fokus auf den Benutzer:** Bereits vor Bau eines Prototyps sollten Benutzer befragt werden, welche Konsequenzen das zu entwickelnde Gerät für ihre Arbeit haben könnte.
- **Iteratives Design:** Ergonomisch begründete Designänderungen sollten evaluiert werden und schrittweise zu neuen Prototypen führen, die wiederum evaluiert werden.
- **Partizipation des Benutzers:** Während des gesamten iterativen Designprozesses sollten potenzielle Benutzer befragt und im Umgang mit den Prototypen beobachtet werden.
- **Empirische Daten als Grundlage aller Entscheidungen:** Grundsätzlich sollten alle ergonomisch relevanten Entscheidungen auf empirischen Daten über das Verhalten von Benutzern basieren.

1.1 Das Methodenrepertoire

- Zur **Definition des Gegenstandes** einer jeden Wissenschaft gehören neben der inhaltlichen Bestimmung des Gegenstandsbereichs auch Angaben zu ihren spezifischen **Methoden**.
- Human-Factors-Forschung und ergonomischer Forschung verwenden grundsätzlich nur **empirische Methoden**.
- **Empirische Methoden:** Aussagen über menschliches Erleben und Verhalten in ergonomisch relevanten Umwelten werden an der Realität überprüft (analog den naturwissenschaftlichen Forschungsprozeduren, die sich bereits seit langem bewährt haben und wissenschaftstheoretisch gut begründet sind).

Das bedeutet:

1. Man muss die Situationen aufsuchen oder künstlich herstellen, in denen das zu untersuchende Verhalten auftreten kann, d.h. man muss dafür sorgen, dass die **Kausalfaktoren** (**unabhängige Variablen** → **UV**), die das Verhalten beeinflussen, wirksam werden können (z.B. im **Experiment**);

2. man muss so exakt wie möglich **beobachten**, welche Verhaltensphänomene (**abhängige Variablen** → **AV**) auftreten und wie sie sich im Detail auswirken (→ systematische Verhaltensbeobachtung, Befragung);

3. diese Beobachtungen dürfen nicht durch Messungenauigkeiten, Täuschungseffekte und Konfundierungen von Kausalfaktoren (nicht auflösbare Vermischung mehrerer Kausalfaktoren) beeinflusst sein (→ experimentelle **Kontrolle**);

4. für die auftretenden Verhaltensphänomene muss geprüft werden, ob ihre Auftretenshäufigkeit noch durch Zufall zu erklären ist oder nicht mehr (→ Funktion der **Statistik**).

- **Prinzip der Hypothesenprüfung durch Vorhersagen:**

Vergleich von Vorhersagen (aufgestellt auf der Basis von Theorien oder empirischen Generalisationen) mit Beobachtungen physischer Ereignisse (Verhaltensphänomene).

Nur dann, wenn Vorhersagen und Beobachtungen physischer Verhaltensphänomene überzufällig übereinstimmen, kann eine Aussage als wissenschaftliche Aussage (einer empirischen Wissenschaft) akzeptiert werden!

→ Auch ergonomische Studien müssen den Kriterien einer empirischen Wissenschaft genügen, um für die Begründung ergonomischer Entscheidungen eine objektive Basis zu haben.

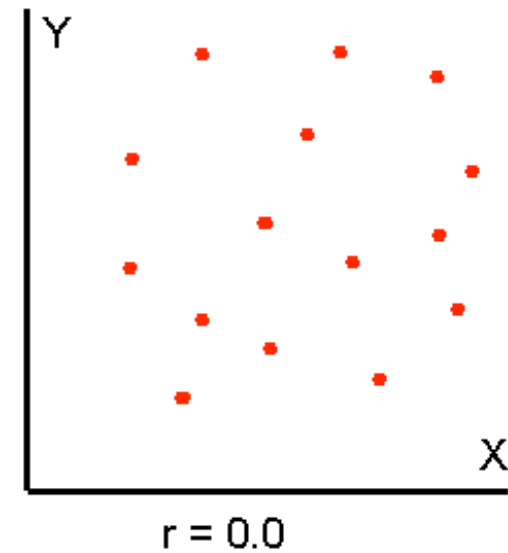
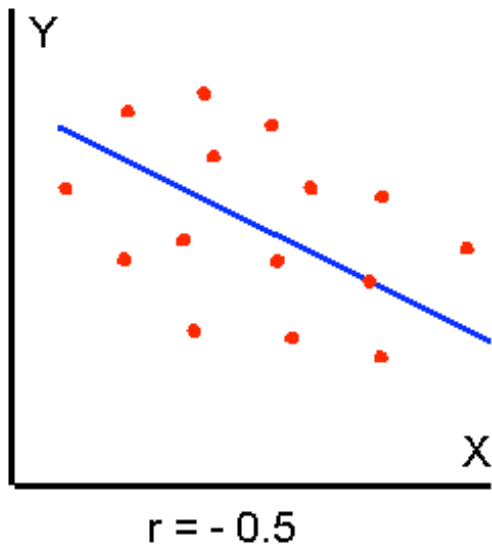
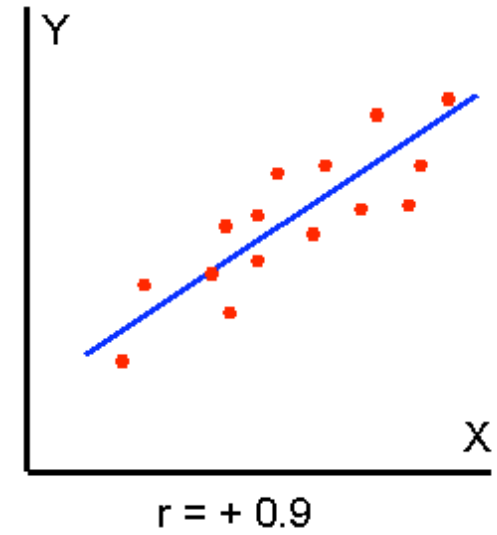
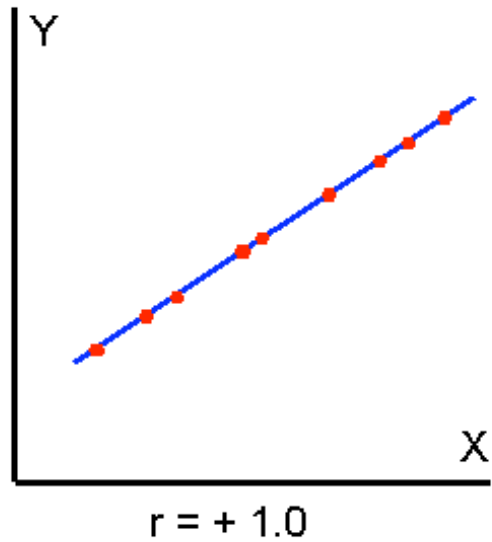
● Empirische Forschungsmethoden

1. Designs

a) *korrelatives Design*: Es wird lediglich die Stärke eines Kovariationszusammenhangs registriert, nicht seine Richtung (kausale Beeinflussungsrichtung),

d.h. es wird **keine Kausalrichtung** vom Typ „Wenn X (Kausalfaktor) dann Y (Verhaltensphänomen)“ ermittelt sondern lediglich die Stärke ihres Zusammenhangs „X und Y sind irgendwie voneinander abhängig“, indem man X und Y so misst, wie sie natürlich vorkommen.

Beispiele für Kovariationszusammenhänge:



b) *experimentelles Design*: Auch hier wird nur die Stärke eines Kovariationszusammenhangs registriert, aber die interessierenden Kausalfaktoren werden vom Experimentator systematisch variiert, so dass die kausalen Einflussgrößen und damit die Kausalrichtung systematisch variiert werden können;

d.h. es wird das Vorhandensein einer **Kausalrichtung** vom Typ „Wenn X (Kausalfaktor \rightarrow UV) dann Y (Verhaltensphänomen \rightarrow AV)“ auf der Grundlage einer Ermittlung der Kovariationsstärke von X und Y ermittelt.

2. Methoden der Datenerhebung

a) systematische Verhaltensbeobachtung: Beobachtung der physischen Merkmale von Erleben und Verhalten unter Verwendung exakter Definitionen der zu registrierenden Verhaltenseinheiten und -merkmale (häufig unter Verwendung eines Kategoriensystems zur Abbildung unterschiedlicher Verhaltensweisen, d.h. mehrere AVn).

Grenzfall: Registrierung nur einer AV, z.B. in einem Experiment (z.B. Reaktionszeit)

Beispiel: Die Benutzeroberfläche des *iPhone* wird mit der konventioneller Handys unter ergonomischen Gesichtspunkten verglichen.



Mögliche Hypothesen:

1. Die benötigte Zeit zur Durchführung einer bestimmten Aufgabe ist beim *iPhone* kürzer als beim Vergleichgerät.
2. Die Benutzerzufriedenheit ist beim *iPhone* größer als beim Vergleichgerät.

Beobachtungssituation: Den Vpn werden bestimmte Aufgaben gestellt (z.B. Telefon-Nr. aus Verzeichnis heraussuchen und Nummer wählen).

- AVn:**
1. Benötigte Zeit, bis letzte Handlung ausgeführt wurde.
 2. Häufigkeit bestimmter Handlungen → **Kategoriensystem**

Kategoriensystem (Handlungen pro Zeitintervall von 5 s):

- Taste drücken
- Bildschirm berühren
- Scrollen

3. Fragebogen zur Benutzerzufriedenheit

b) Fragenbogen und Interview: Befragungsmethoden (Interview, Fragebogen), die sich auf die subjektiven **Erinnerungen** der Befragten über ihre Wahrnehmungen und Bewertungen ihres eigenen Verhaltens in bestimmten Situationen stützt.

Anwendbar nur dann, wenn

- Befragte über ausreichend Erfahrung zum Befragungsgegenstand verfügen;

ist dies nicht der Fall → Vpn müssen die entsprechenden Erfahrungen unter kontrollierten Bedingungen machen (z.B. bei einem Usability-Test);

Fortsetzung: Anwendbar nur dann, wenn

- Inhalte bewusst und Erinnerungsfehler weitgehend ausgeschlossen sind;

ist das nicht der Fall → Experiment mit systematischer Variation der das Verhalten determinierenden Kausalvariablen.

- Befragungssituation keine Reaktivitätseffekte erzeugt (Reaktivität: Messvorgang verändert Messwert, z.B. soziale Erwünschtheit, Prestige eines Herstellernamens)

treten Reaktivitätseffekte auf → *unwissentliche* systematische Verhaltensbeobachtung

1.2 Experimentelle Überprüfung ergonomischer Fragestellungen

Ziel: Nachweis, dass eine **kausale Abhängigkeit** einer bestimmten Richtung bei bestimmten Verhaltensmerkmalen **von** bestimmten physischen **Ursachen** besteht, d.h. dass die im Experiment systematisch variierten UVn kausal verantwortlich sind für zu beobachtende Veränderungen der AVn.

Beispiel: Die Benutzung eines Handys während des Lenkens eines Fahrzeugs sollte nach allen Erkenntnissen der Aufmerksamkeitsforschung zu einer Reduzierung der auf das Führen des KFZ fokussierten Aufmerksamkeit führen.

Kausalität: Wenn Handy benutzt wird, dann Reduzierung der Aufmerksamkeit.

1.2.1 Fünf Schritte der Planung und Durchführung eines Experiments

Schritt 1: Formulierung der Fragestellung und der Hypothese

1a: Fragestellung: In der angewandten Forschung ergeben sich Fragestellungen i.d.R. aus Problemstellungen des Alltags, seltener aus dem Versuch, eine Theorie kritisch zu überprüfen.

→ s. obiges Beispiel der Konsequenzen der Handybenutzung während des Autofahrens

1b: Hypothese: Theoretisch begründete Aussage über Ursachen des Verhaltens evtl. auch mit genauen quantitativen Bestimmungen der relativen Stärke dieser Ursachen.

→ s. obige Hypothese über den ursächlichen Zusammenhang zwischen Handybenutzung und Aufmerksamkeitsablenkung

Schritt 2: Spezifikation des experimentellen Designs und Operationalisierung der UVn und AVn

Operationalisierung:

- *Spezifikation der in der Hypothese gemachten Aussagen sowohl über die Ursachen (UVn) als auch über die durch sie bedingten Verhaltensmerkmale (AVn) in Form von Angaben über deren physische Parameter.*
- *Verankerung der theoretischen Konzepte in der physischen Realität.*

1. Operationalisierung der UVn: Exakte Festlegung der physischen Parameter der als UVn im Experiment wirkenden Ursachen für das zu beobachtende Verhalten.

Beispiel: UV1 „Handybenutzung“ (Abk. H) →

Bedingung H1: Handy wird benutzt während V_p das KFZ durch den Verkehr steuern muss

Bedingung H2: Es wird kein Handy benutzt; ansonsten alles gleich.

UV2 „Verkehrsdichte“ (Abk. V) →

Die Hypothese soll noch etwas präzisiert werden: Die durch die Benutzung des Handys bewirkte Aufmerksamkeitsablenkung sollte sich bei relativ schwachem Verkehr nur geringfügig negativ auswirken; erst wenn die Verkehrsdichte relativ groß ist, sollten massive Effekte auftreten.

Bedingung V1: geringe Verkehrsdichte (3 zu überholende Autos pro Min. auf derselben + 3 zu beachtende Autos (Vorfahrtsregeln!) auf kreuzenden Fahrspuren)

Bedingung V2: hohe Verkehrsdichte (30 zu überholende Autos pro Min. auf derselben + 30 zu beachtende Autos (Vorfahrtsregeln!) auf kreuzenden Fahrspuren)

2. Operationalisierung der AVn: Exakte Festlegung der physischen Parameter der im Experiment zu bewirkenden und zu beobachtenden Verhaltensänderungen.

Beispiel: AV „Abweichungen von der Spur“ →

Aufgabe der Vp: Fahrsimulator, das Fahrzeug möglichst genau in der Mitte der Spur zu halten.

Zu beobachtender Parameter des Verhaltens: Messung der Abweichungen des Fahrzeugmittelpunkts von der Spurmitte (in cm)

Design: Als Design bezeichnet man die Konfiguration der verschiedenen experimentellen Bedingungen, die durch Kombination der AV_n entstehen.

Beispiel: Die Variablen **H** (Handybenutzung) und **V** (Verkehrsdichte) besitzen jeweils zwei Stufen (Bedingung 1 und 2). Ordnet man diese orthogonal an, ergibt sich aus der Kombination der 2x2 Bedingungen folgende Vierfelder-Tafel (2x2-faktorielles Design):

	geringe Verkehrsdichte V_1	hohe Verkehrsdichte V_2
keine Handybenutzung H_1	Kontrollgruppe/ Kontrollbedingung	Kontrollgruppe/ Kontrollbedingung
Handybenutzung H_2	experimentelle Gruppe	experimentelle Gruppe

Schritt 3: Durchführung der Untersuchung

1. Material:

Alle Geräte sowie die den Vpn gegebenen Informationen in Form von Instruktionen und bildlichen Darstellungen müssen exakt spezifiziert werden und für alle Vpn konstant sein (**Bedingungskonstanz**).



Fahrsimulator



Präsentation der visuellen Informationen im Fahr Simulator

2. Ablauf: Spezifikation, welche Aufgabenschritte wie häufig innerhalb welchen Zeitraums durchzuführen sind. Hier ist wichtig: Alle Vpn müssen exakt denselben Ablauf haben (**Bedingungskonstanz**).

3. Stichprobe:

a) Auf welche Personen soll das Ergebnis anwendbar sein?

alle Menschen → repräsentative Zufallsstichprobe aller Menschen

bestimmte Menschen, z.B. nur Autofahrer → repräsentative Zufallsstichprobe aus der Gruppe der Autofahrer

b) Wie zuverlässig und genau sollen die Daten sein?

- Stichprobengröße min. 1000 bei repräsentativer Stichprobe.
- Je genauer und zuverlässiger (reliabler) eine AV gemessen werden kann, desto kleiner kann i.d.R. die Stichprobe sein.

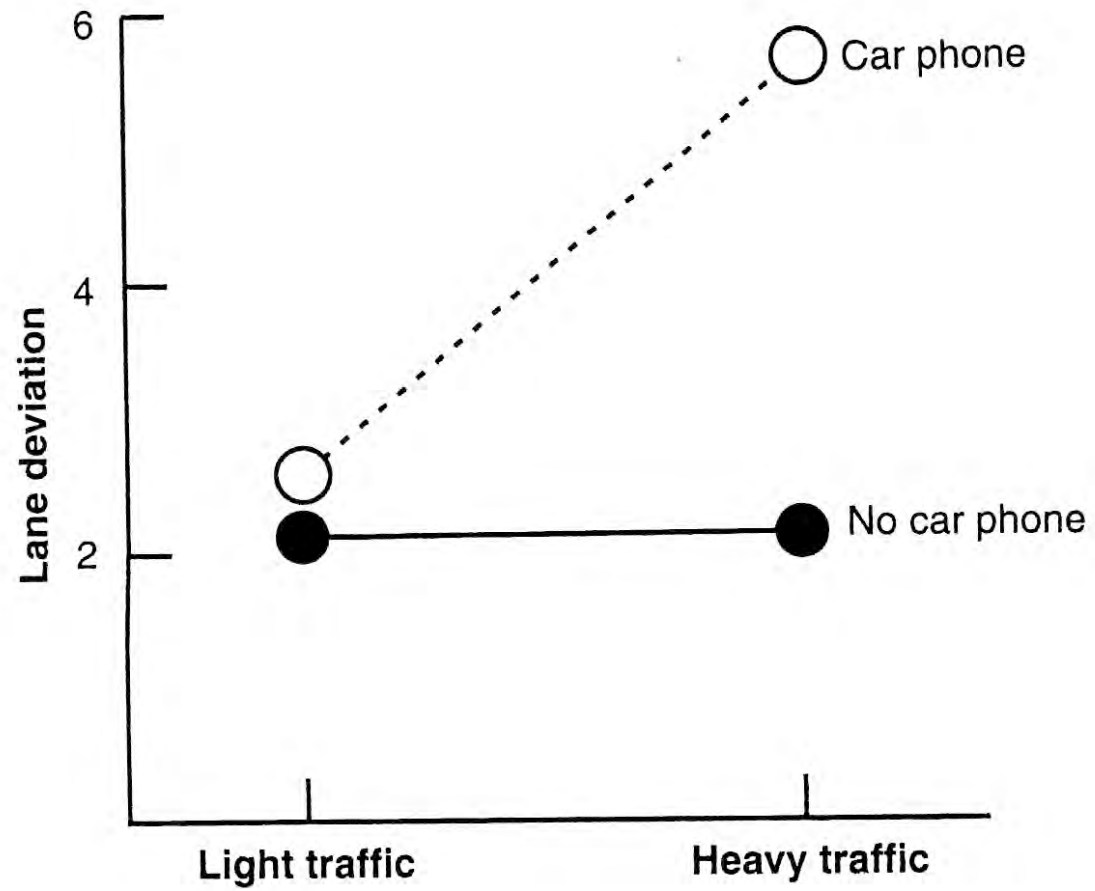
Schritt 4: Statistische Datenanalyse

- Die Funktion der Statistik besteht darin, eine Aussage darüber zu ermöglichen, wie groß die Wahrscheinlichkeit ist, dass die beobachteten Unterschiede zwischen den verschiedenen experimentellen Bedingungen noch per Zufall zu beobachten sein sollten.
- Nur solche Ergebnisse, für die diese Wahrscheinlichkeit deutlich kleiner als 5% ist, werden als überzufällig betrachtet.

Beispiel: 2x2-faktorielle Varianzanalyse

Erlaubt unabhängige Abschätzung der Stärke der UV „Handy“ und der UV „Verkehrsdichte“ sowie ihrer Wechselwirkung (Interaktion)

Ergebnisse:



Schritt 5: Ziehen von Schlussfolgerungen

Zeigt die statistische Analyse, dass die Differenzen zwischen den Bedingungen überzufällig > 0 sind → Ergebnis stützt Hypothese.

Beispiel: statistisch überzufällige Effekte zeigen

- die UV „Verkehrsdichte“ als **Haupteffekt**: Je dichter der Verkehr, desto größer die Spurabweichungen (gemittelt über beide Handybedingungen).
- die UV „Handy“ als **Haupteffekt**: Mit Handybenutzung sind die Spurabweichungen größer als ohne (gemittelt über beide Verkehrsdichtebedingungen).
- die **Wechselwirkung** beider UVn: Der Handyeffekt beschränkt sich auf die Bedingung mit hoher Verkehrsdichte (Geraden in Graphik verlaufen nicht parallel!).

Resümee:

- Nur mit **experimentellen Methoden** lassen sich **Kausalhypothesen** testen; korrelative Methoden können nur statistische Zusammenhänge ohne Klärung der Kausalrichtung analysieren.
- Deshalb sind **experimentelle Methoden** besonders gut geeignet, im Zusammenhang ergonomischer Fragestellungen nach **Ursachen** für menschliche Fehler und Schwierigkeiten bei der Bedienung von Geräten zu fahnden.
- Zur Erhebung *subjektiver* **Einschätzungen und Bewertungen** können **Fragebogenmethoden** verwendet werden.
- Die Analyse von Handlungsfehlern und von Schwierigkeiten bei der Benutzung von Geräten können am besten im Rahmen einer **systematischen Verhaltensbeobachtung der Handlungsausführung** im Kontext eines Experiments analysiert werden.

1.3 Spezielle Methoden im Zusammenhang mit der Gestaltung technischer Systeme und Arbeitsplätze

1.3.1 Kosten-Nutzenanalyse

- Empirisch gestützte, ergonomische Analysen kosten Geld.
 - Erhöhung der Produktionskosten, Reduzierung der Gewinns
- Lässt sich eine empirisch gestützte ergonomische Analyse dennoch wirtschaftlich rechtfertigen? → **Kosten-Nutzenanalyse**
- **Kosten-Nutzenanalyse:** Bestimmung der Differenz zwischen dem monetären Nutzen der Analyse und den Kosten ihrer Durchführung.

Kostenparameter:

- Vorbereitung des Materials/Prototyps
- Personalkosten (Mannstunden x Stundenlohn)
- Versuchspersonenkosten (Anzahl Vpn x Zeit (h) x Stundenlohn)
- Kosten für Beobachtungs- und Versuchsapparaturen (Usability-Labor mit Rechner, Bildschirm drei Videokameras usw.)
- Datenanalyse (EDV)
- Abfassung und Präsentation des Berichts
- Kosten für die Modifikation des Designs

Nutzenparameter:

- „*Ergonomisch getestet!*“ → Verbesserung des Absatzes
- reduzierte Trainingskosten (z.B. bei Software, komplexen Maschinen)
- reduzierte Kosten für Support beim Kunden
- reduzierte Wartungskosten
- erhöhte Produktivität des Nutzers
- verringerte Fehlerzahlen (Ausschuss in der Produktion) auf Seiten des Nutzers
- verminderte Anzahl an Beschwerden durch Nutzer bzw. Rückgaben des Produkts
- erhöhte Benutzerzufriedenheit → informelle Empfehlungen an andere Kunden, Wahl desselben Anbieters bei späterer Neuanschaffung
- geringere körperliche Belastungen und Erkrankungen beim Nutzer
- Reduzierung der Anzahl und Schwere von Unfällen
- Reduzierung der Kosten für juristische Auseinandersetzungen im Rahmen der Herstellerhaftung

Viele Untersuchungen im Bereich der Software-Ergonomie haben gezeigt, dass selbst so geringfügig erscheinende Verbesserungen wie z.B.

- Gestaltung einer übersichtlicheren Bildschirmmaske für die Eingabe von Daten (Zeitersparnis: 3 s),
- optimale Gestaltung von Menüs (z.B. flache Hierarchien) hinsichtlich Zeitbedarf zum Finden eines Menüpunkts,
- automatische Funktionen zwecks Einsparung von Dateneingaben (z.B. Excel: Speicherung von Eingaben und automatische Ergänzung bei Namen usw.)

enorme Einsparungen bewirken, wenn die Funktionen häufig, über eine lange Zeitspanne und von vielen Menschen benutzt werden (Größenordnungen von einigen zigtausend €).

→ Der Nutzen übersteigt immer dann die Kosten, wenn man als Nutzen nicht nur den für den Hersteller sich unmittelbar in Geld auszahlenden Nutzen sieht sondern auch den für den Kunden/Nutzer mit einkalkuliert!

1.3.2 Vorausgehende Analyseschritte vor der eigentlichen ergonomischen Untersuchung

Ziel: Analyse der Bedürfnisse des Benutzers und der Anforderungen, die die Arbeitssituation an ihn stellt.

Fragen: müssen vor Erarbeitung des Designs beantwortet werden

1. Wer sind die potenziellen Benutzer? Über welches Wissen, Erfahrungen, Fertigkeiten verfügen sie?
2. Worin bestehen die Hauptaufgaben/-funktionen des zu gestaltenden Gerätes/Arbeitsplatzes?
3. Unter welchen Umgebungsbedingungen kommt das Gerät zum Einsatz bzw. wird an dem Arbeitsplatz gearbeitet?
4. Welche Anforderungen stellt der Benutzer an das Gerät bzw. den Arbeitsplatz?

→ Diese Fragen werden beantwortbar aufgrund der im Folgenden beschriebenen Analysen.

1.3.2.1 Benutzeranalyse

Benutzeranalyse: Sie besteht in einer Registrierung der für die Bewältigung einer vorgegebenen Arbeitsaufgabe kritischen Merkmale der zukünftigen Benutzer/Operateure eines technischen Systems.

Zu erhebende Merkmale (sofern für den Umgang mit dem System kritisch):

- **physische Anforderungen:** Alter, Geschlecht, Körpergröße und -kräfte, Körperbehinderungen
- **psychische Anforderungen:** Belastbarkeit durch Stress, Daueraufmerksamkeit, Monotonie
- **intellektuelle Anforderungen:** aufgabenrelevante Kenntnisse, einschlägige Erfahrungen und Fertigkeiten (Ausbildungshintergrund)

Persona (Konzept von Cooper, 1999 → *zusammenfassende Darstellung der Ergebnisse der Benutzeranalyse*): Beschreibt eine idealtypische Person, die die oben genannten Anforderungen in idealer Weise erfüllt, beschreibt die Hierarchie der Handlungsziele, die einzelnen auszuführenden Handlungen zum Erreichen der verschiedenen Ziele, die physischen Arbeitsmittel sowie die Arbeitsumgebung.

Informationsbasis: **Interviews** mit potenziellen Benutzern und systematische **Verhaltensbeobachtungen** bei Verwendung eines Modells/Prototyps des Systems bzw. bei Ausführung der Arbeit ohne das System unter Einsatz der bis dahin üblichen Hilfsmittel.

1.3.2.2 Umgebungsanalyse

Umgebungsanalyse: Registrierung der für die Bewältigung einer vorgegebenen Arbeitsaufgabe kritischen Merkmale der räumlichen und klimatischen Verhältnisse, unter denen mit einem zu entwickelnden technischen Systems gearbeitet wird.

relevante Merkmale:

- Raumklima: Luftfeuchtigkeit, Temperatur, Luftaustausch
- räumliche Enge
- Arbeitskleidung, Schutzkleidung

1.3.2.3 Funktions- und Aufgabenanalyse

Funktionsanalyse: Zusammenstellung einer Liste von Funktionen, die das Mensch-Maschine-System (Benutzer in Interaktion mit dem zu entwickelnden technischen System) zu erfüllen hat in Form der zu erreichenden Arbeitsziele.

Beispiel: Eine neue Textverarbeitungssoftware soll folgende Aufgaben zu bearbeiten ermöglichen: 1. Eingabe von Texten; 2. Editieren von Texten; 3. Einbinden von Graphiken und Tabellen; 4. Text formatieren; 5. automatische Funktionen: Fußnotenverwaltung, Erstellen eines Inhaltsverzeichnisses, Erstellen eines Stichwortverzeichnisses; 6. Speicher- und Druckfunktionen usw.

Aufgabenanalyse: Wichtigstes Werkzeug zur Analyse der Interaktionen des Benutzers mit dem zu gestaltenden technischen System zwecks Bestimmung der *Ziele*, *Funktionen* und *Aufgaben* unter besonderer Berücksichtigung ihrer Abhängigkeiten.

Ziele: Von der eingesetzten Technologie unabhängige Arbeitsergebnisse (evtl. mit Spezifizierung einiger Zwischenziele) (z.B. Erstellen eines Textes)

Funktionen: Transformationen eines gegenwärtigen Zustands (Teilergebnis einer Arbeit) in einen anderen Zustand, die es ermöglichen, einem der zu erreichenden Ziele näher zu kommen (z.B. Formatieren des Textes)

Aufgaben: Von der eingesetzten Technologie abhängige spezifische Handlungen zum Erreichen bestimmter Ziele/Zwischenziele (z.B. Wahl des Parameters „Schriftgröße“ aus einem Formatierungsmenü)

Drei Schritte der Durchführung einer Aufgabenanalyse:

1. Design der Datenerhebung: Festlegung der Analyseziele, der einzusetzenden Erhebungsmethoden und Festlegung der zu erhebenden Daten.

2. Durchführung der Datenerhebung: Durchführung der Interviews und Beobachtungen bei zahlreichen potenziellen Benutzern.

3. Zusammenfassung und Analyse der Daten: Zusammenfassung der individuellen Daten so, dass über individuelle Tendenzen erkennbar werden. Statistische Analysen zwecks Bestimmung der Systematik der Beobachtungen bzw. der Größe von Zufallsschwankungen.

1. Design der Datenerhebung bei der Aufgabenanalyse

Analyseziele: Typische Analyseziele sind

- *Feststellung des Umfangs eines Trainings zur optimalen Bewältigung der Aufgaben:* Manche Tätigkeiten erfordern ein spezifisches, aufwändiges Training → Umfang des erforderlichen Trainings soll minimiert werden.



Beispiel: Fotoapparat → Gerät sollte selbsterklärungsfähig (selbstbeschreibungsfähig) sein, über Display dem Benutzer die jeweils notwendigen Informationen geben; Übung und Lesen von Gebrauchsanweisungen sollte überflüssig sein.

- *Identifizierung der Anforderungen an benötigte Software und/oder Hardware:* Software soll die Ausführung der Tätigkeiten unterstützen; welche Tätigkeiten können der Software überlassen werden, welche muss der Mensch ausführen; wie muss die einzusetzende Hardware beschaffen sein?

Beispiel: Fotoapparat

→ Software muss die Reihenfolge der Bedienung festlegen, notwendige Informationen darstellen, auf potenzielle Fehler aufmerksam machen (z.B. ungünstige Lichtverhältnisse).

→ Hardware: leicht, robuste Konstruktion, gute Sichtbarkeit der Displayinformationen; gute Erreichbarkeit der Einstellfunktionen, ästhetisch ansprechend gestaltet usw.

- *Ermittlung der Zuverlässigkeit des Systems:* Die Tätigkeit muss zuverlässig zum erwünschten Ergebnis führen; Ausschuss muss minimiert werden.

Beispiel: Fotoapparat → Belichtungssystem muss die gesamte Bandbreite von Lichtverhältnissen bewältigen können (z.B. auch Gegenlicht); Displaykontrast muss so groß sein, dass auch bei Sonnenschein Bildausschnitt kontrastreich dargestellt wird; grundsätzlich: eindeutige Rückmeldungen zwecks Fehlervermeidung.

- *Ermittlung der benötigten Zahl von Mitarbeitern und deren Voraussetzungen an Fertigkeiten und Kenntnissen*

Beispiel: Fotoapparat → hier nur eine Person ohne spezielle Kenntnisse

- *Bestimmung der kognitiven Anforderungen:* In der Vergangenheit wurden den kognitiven Anforderungen bei der Bedienung komplexer Systems zu wenig Aufmerksamkeit geschenkt.

→ Analyse kognitiver Anteile: *Welche kognitiven Vorgänge sind bei der Bedienung des Systems involviert?*

- Komplexe **Entscheidungsprozesse**, Lösen von Problemen, diagnostisches Denken;
- umfangreiche, sehr spezialisierte **Kenntnisse** sind erforderlich, um die gestellten Aufgaben zu bewältigen;
- wenn ein umfangreiches **Regelwerk** bei der Bedienung berücksichtigt werden muss, müssen kritische **situative Merkmale** korrekt berücksichtigt werden (z.B. Kernkraftwerk: Anlässe und Bedingungen für eine Notabschaltung).

Beispiel: Fotoapparat → Gedächtnisentlastung durch Benutzerführung

- *Bestimmung der Arbeitsbelastung*: Die physische Arbeitsbelastung muss minimiert, psychische Arbeitsbelastungen (z.B. aufgrund von Zeitdruck) dürfen erst gar nicht auftreten; aber gleichzeitig ist Langeweile zu vermeiden (bei monotonen Aufgaben, z.B. Kontrolltätigkeiten).

Beispiel: Fotoapparat → Auslöseknopf und andere Einstellelemente müssen ohne Kraftaufwand zu bedienen sein (z.B. auch von Kindern); das Gewicht muss minimal sein (→ Verwendung leichter, aber stabiler Materialien); die Komplexität der gesamten Bedienung muss so gering sein, dass kein Stress entstehen kann.

2. Durchführung der Datenerhebung in einer ergonomischen Untersuchung

je nach Fragestellung

- Fragebogen/Interview von Novizen und Experten;
- Beobachtung evtl. unter Vergleich verschiedener Bedingungen (Experiment)

zu erhebenden Daten: 4 Inhaltsbereiche (s.u.)

1. *Hierarchische Beziehungen von Aufgaben und Teilaufgaben:* Analyse der zu erledigenden Aufgaben in Teilaufgaben ermöglicht eine inhaltliche Gruppierung (wird benötigt z.B. für die Bildung von Lerneinheiten in Trainingsprogrammen; auch für die Organisation von Fließbandarbeit, die aus einer sinnvollen Abfolge von Teilaufgaben besteht).

Beispiel: Fotoapparat → Aufgabe „Bild machen“: 1. Kamera anschalten; 2. im Display Bildausschnitt wählen; 3. Auslöser drücken; 4. gemachtes Bild im Display kontrollieren.

2. Informationsfluss zwischen Menschen und Maschinen: In komplexen Aufgabensituationen ist eine ungestörte (z.B. Lärm in Werkshalle) und flüssige Kommunikation zwischen den beteiligten Menschen und technischen Systemen (z.B. schnelle Informationsdarbietung für Benutzer, schnelle Informationseingabe ins System) zur Vermeidung von Fehlern und zur Sicherstellung eines effizienten Ablauf notwendig.

Beispiel: Fotoapparat → hier besteht eine Kommunikation zwischen Benutzer und Display bzw. Eingabe von Befehlen an das System über Schalter.

3. Reihenfolge der Aufgabendurchführung: Registrierung der optimalen zeitlichen Abfolge und Dauer einzelner Teilaufgaben zur Sicherstellung möglichst effizienter Abläufe; Feststellung der Anzahl benötigter Personen bei simultaner Erledigung von Teilaufgaben, Registrierung von zeitlichen Koordinationsproblemen (z.B. wichtig für Fließbandarbeit).



Beispiel: Fotoapparat → System sollte Reihenfolge festlegen und dem Benutzer keine andere Reihenfolge gestatten: Auslöser drücken und damit erfolgreich ein Bild machen bevor Apparat angeschaltet wurde, wäre technisch machbar, würde aber zu Zufallsbildern, unnötige Batterieentleerung usw. führen, wenn unbeabsichtigt Auslöser gedrückt wird.

4. Räumliche und Umweltbedingungen: Räumliche Anordnung der Apparate, Werkzeuge, Materialien und Hilfsmittel bei der Durchführung der einzelnen Teilaufgaben unter dem Gesichtspunkt der guten Erreichbarkeit, des ausreichend großen Platzangebots zur unbehinderten Ausführung aller nötigen Bewegungen.

Beispiel: Fotoapparat → Ist das Gehäuse griffig, nicht zu schwer; Helligkeitskontrast des Displays zum Umgebungslicht ausreichend.

Berücksichtigt werden folgende Aspekte:

- Wege, die die einzelnen beteiligten Personen oder Transportfahrzeuge zurücklegen müssen;
- einengende physische Strukturen (z.B. Wände, Tische, Stühle);
- Werkzeuge und deren Positionen;
- klimatische und Beleuchtungsbedingungen sowie solche Aspekte der räumlichen Gestaltung, die dazu führen, dass sich die handelnden Personen wohl fühlen.



© Leonard Riegel

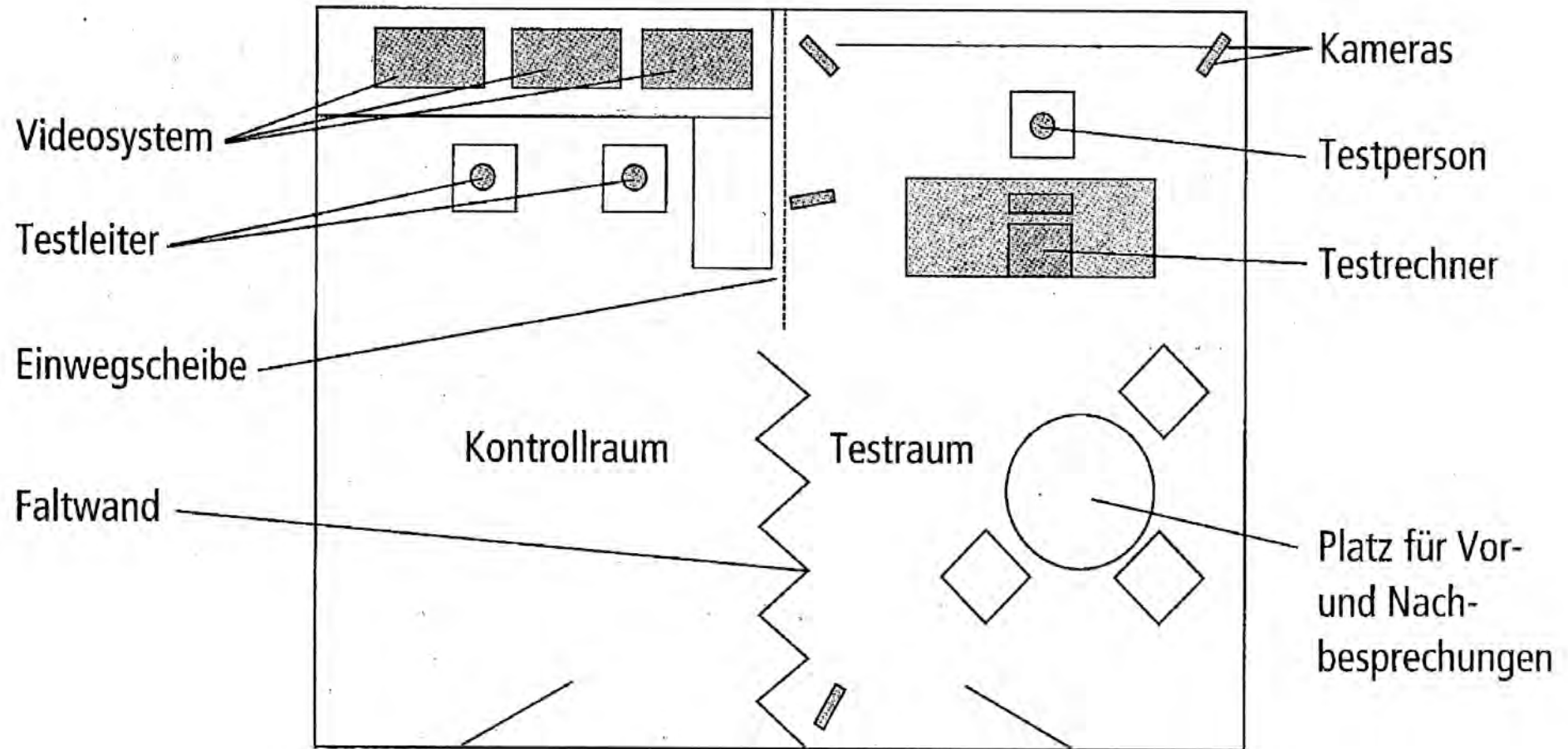
Welche Daten werden benötigt? Operationalisierung

- **Testpersonen:** Die Datenerhebung und -analyse wird nur mit Personen durchgeführt, die in Zukunft häufige Benutzer des zu evaluierenden Objekts sein werden.
- Die Art der benötigten Daten ist von der jeweiligen Fragestellung abhängig, besteht aber im Prinzip in der **Beobachtung** und/oder **Befragung** der zukünftigen Nutzer, während oder nachdem sie mit dem zu evaluierenden Objekt gearbeitet haben.

- **Beobachtung:**

- Potenziellen Nutzern werden alle möglichen Aufgaben gestellt, die sie mit Hilfe des zu evaluierenden Objekts oder technischen Systems bearbeiten sollen.
- Dabei werden sie beobachtet: meistens mit Videoaufzeichnung durch mehrere Kameras aus verschiedenen Blickwinkel (insbesondere die Gesicht und Hände in Großaufnahme).
- Beobachtung kann auch mit Fragen durch den Versuchsleiter kombiniert werden (z.B. Fragen danach, warum man gerade etwas macht und warum so und nicht anders).

Beispiel eines Beobachtungslabors



→ Warum ist Beobachtung eine wichtigere Methode als Befragung?

- Was Personen sagen und was sie tun stimmt nicht immer überein!
- Bei Befragung stützt sich die Antwort auf das *Gedächtnis* über die ausgeführten Handlungen, nicht auf die tatsächlich ausgeführten Handlungen.
- Gedächtnisbefunde sind deutlich weniger zuverlässig, weil
 - Details ausgelassen werden können;
 - subjektive Gewichtungen und Meinungen eine selektive Rolle spielen können;
 - Befragte sich die neue Technologie (wenn kein Prototyp vorliegt) nicht konkret genug vorstellen können;
 - sie verfälschen u.U. ihre Beschreibungen der Handlungen, um als besonders kompetent zu erscheinen;
 - es ist wesentlich schwieriger und daher fehleranfälliger zu beschreiben, was man tun würde, als die Handlungen tatsächlich durchzuführen.

- **Protokolle über „lautes Denken“:**

- Tonbandaufzeichnung dessen, was Personen begleitend zur Ausführung ihrer Handlungen denken und verbal formulieren.

- Dabei wird in der Instruktion betont, dass nichts ausgelassen werden soll, also alle Einfälle spontan wiedergegeben werden sollen.

- **Sinnvoll nur in Verbindung mit Videoaufzeichnungen** der Handlungen, da sonst nicht klar ist, worauf sich die Äußerungen jeweils beziehen!

- Erlaubt z.B. Einblicke in
 - die Festlegung von Handlungszielen (**Beispiel:** Fotoapparat → „Platz schaffen im Speicher: Alle schlechten Bilder löschen“)
 - die Planung und Modifikation von Handlungsstrategien (**Beispiel:** Fotoapparat → Bildqualität zu schlecht; Auflösung von 5 MB auf 10 MB pro Bild erhöhen)
 - das Vorgehen beim Lösen von Problemen (**Beispiel:** Fotoapparat → Bild unterbelichtet: 1. Batterie prüfen. 2. War Blitz abgeschaltet?)
 - Fällen und Begründungen von Entscheidungen (**Beispiel:** Fotoapparat → Hineinzoomen, weil zu wenige Details erkennbar waren.)
- Protokolle können „*simultan*“ oder „*retrospektiv*“ sein: Simultanes lautes Denken ist schwieriger, weil es höchste Konzentration erfordert (Doppelaufgabensituation); retrospektive Protokolle liefern daher in der Praxis die nützlicheren und zuverlässigeren Informationen (Ohnemus & Biers, 1993).

- **Befragung:**

- Befragung während der Aufgabenbewältigung: Variante der Methode des lautenden Denkens mit zusätzlichen, gezielten Fragen nach

- Zielen und Zwischenzielen

- Entscheidungsproblemen

- alternativen Handlungsstrategien

- Bewertungen und Präferenzen für Vorgehensweisen

- auch retrospektiv anwendbar (Gründe s.o.)

- unstrukturierte vs. strukturierte Interviews: Meistens zuerst unstrukturierte Befragung (Fragen werden ad hoc formuliert); später wird aufgrund der gewonnenen Erfahrungen ein strukturiertes Interview durchgeführt: Fragen sind vorformuliert, ihre Reihenfolge festgelegt, evtl. sind Antwortalternativen vorgegeben.

- Fokus-Gruppen: 6-10 potenzielle Nutzer kommen als Gruppe zusammen, besprechen die Aufgabenbewältigung, die Probleme dabei, diskutieren Bewertungen und alternative Vorgehensweisen sowie Stärken und Schwächen des zu evaluierenden Objekts oder Systems.

- Fragebogenuntersuchungen:
 - Schriftlich fixierte Fragen werden erst am Ende des Evaluationsprozesses eingesetzt, um die zuvor gesammelten Beobachtungs- und Interviewdaten auf ihre Zuverlässigkeit zu überprüfen;
 - manchmal werden so auch Häufigkeitsanalysen an größeren Gruppen durchgeführt, um festzustellen, wie häufig welche Aufgaben, wie gerne, unter Einsatz welcher der alternativen Vorgehensweisen durchgeführt werden, insbesondere bei Vergleich verschiedener Objekte (z.B. altes vs. neu gestaltetes Objekt oder System).

3. Analyse und Zusammenfassung der Daten

Listen, Hierarchien, Flussdiagramme: Die Ergebnisse der Aufgabenanalyse lassen sich z.B. in Form einer Liste von Hauptaufgaben und untergeordneten Teilaufgaben darstellen. Flussdiagramme sind geeignet, wenn komplexe zeitliche Abfolgen und verzweigte Entscheidungsbäume eine Rolle spielen. **Beispiel:** Fotoapparat →

Schritt 1: Finden eines guten Blickwinkels auf ein Motiv.

1.1 Wähle das Motiv.

1.2 Wähle den Standort so, dass Hindernisse aus dem Bild verschwinden.

1.3 Wähle den Blickwinkel unter Berücksichtigung der Richtung, aus der das Sonnenlicht kommt (Gegenlicht?).

Schritt 2: Bereite die Kamera auf die Aufnahme vor.

2.1 Falls erforderlich: Objektivdecke abnehmen.

2.2 Kamera anschalten.

2.3 Auswahl des Belichtungsprogramms.

Schritt 3: Aufnahme machen.

3.1 Bildausschnitt wählen.

3.2 Zoom einstellen.

3.3 Scharfstellen bzw. Ende der automatischen Scharfeinstellung abwarten.

3.4 Auslöser drücken.

Schritt 4: Qualität der Aufnahme kontrollieren.

4.1 Auf Bildbetrachtungsmodus umschalten.










4.2 Bild auswählen.

4.3 Bild kritisch auf Fehler analysieren.

1.3.3 Iteratives Design und Usability Evaluation

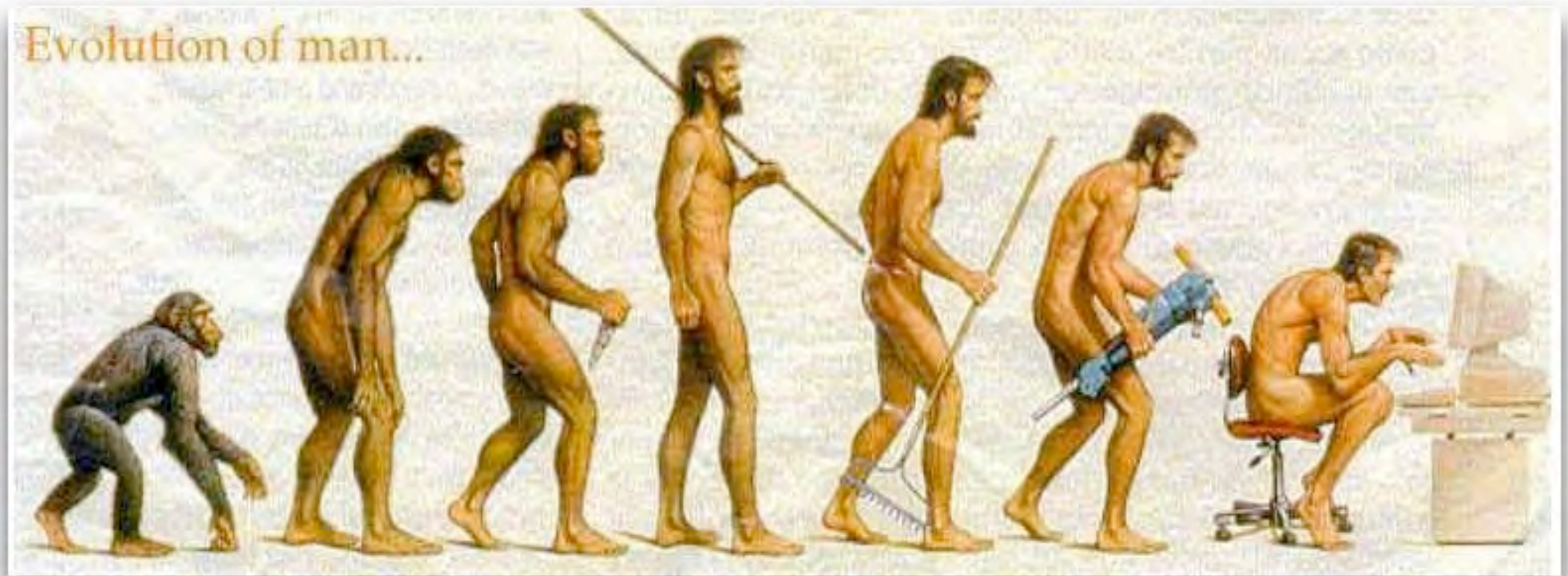
Erst nach Abschluss der dargestellten Präliminarien beginnt der eigentliche *iterative Designzyklus*.

Die einzelnen iterativen Zyklen (manchmal 10-20) bestehen aus:

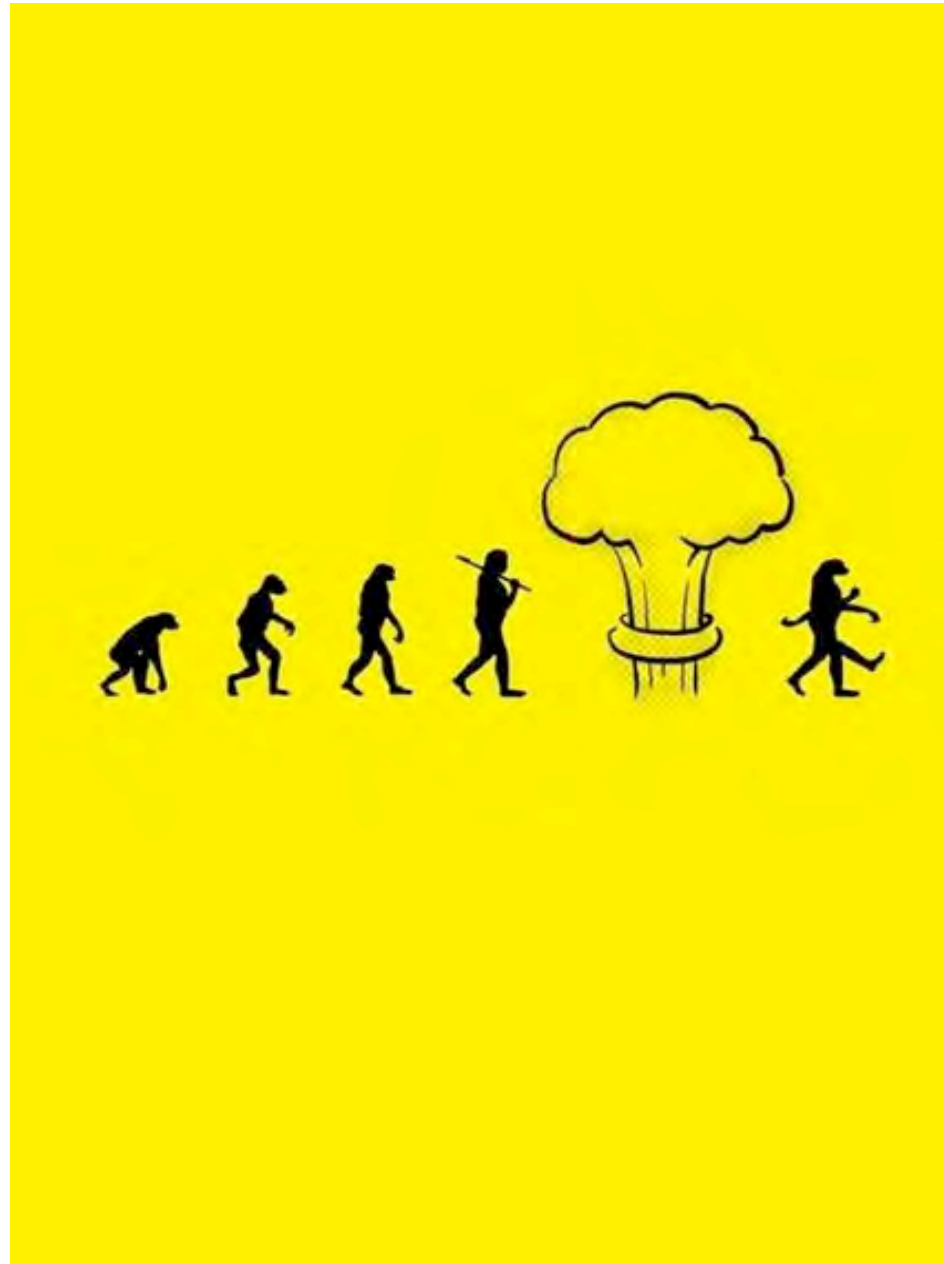
-  1. Vorschlägen der Ergonomen zu Aspekten der Gestaltung des Objekts oder Systems;
 -  2. Entwurf eines (verbesserten) Prototyps;
 -  3. ergonomische Evaluation des Prototyps (frühzeitiges und kontinuierliches Testen der Usability)
-      

Parallele zur Evolution

(keine *natürliche* Selektion sondern Selektion durch Bewährung in Usability-Studien):



fehlgeschlagenes iteratives Design = fehlgeschlagene Evolution:



- **Verwendung der Ergebnisse der Aufgabenanalyse zur System-spezifikation:**

1. Spezifikation notwendigerweise zu erreichender Handlungsziele: Parallel zum Prozess der Generierung verschiedener Designlösungen durch Industriedesigner und Techniker ist es die Aufgabe des Ergonomen, auf der Grundlage der Ergebnisse der Arbeitsanalyse **Vorgaben** zu machen, die sicherstellen sollen, dass

1. alle wünschbaren **Handlungsziele** mit dem Objekt/System **erreichbar**,
2. zu erwarten ist, dass die Handlungsziele **effizient** (schnell, fehlerfrei) und **sicher** erreichbar sind,
3. eine ausreichend hohe **Benutzerzufriedenheit** wahrscheinlich ist (aufgrund der Funktionsvielfalt ohne Komplizierung der Bedienung wegen zu großer Funktionsvielfalt).

→ **Diese Spezifikationen sind sehr allgemein und legen noch keine spezifischen technischen Realisationen fest.**

Beispiel: Fotoapparat → die sich aus den zu erreichenden Handlungszielen ergebenden Spezifikationen:

- Ausreichend Kapazität, um recht viele Bilder machen zu können.
- Fotos sollten unter (fast) allen Lichtverhältnissen möglich sein.
- Bilder sollten ohne Computeranschluss zu betrachten sein.
- Eine Selbstauslöserfunktion sollte vorhanden sein.
- Eine Zoommöglichkeit sollte gegeben sein.
- Die notwendigen Einstellungen sollten weitestgehend automatisch erfolgen; für Experten sollten aber Einflussmöglichkeiten bei Belichtungszeit und Blende zur Verfügung stehen.

2. Spezifikation der Anforderungen an das technische System:

Hierbei werden die zuvor festgelegten zu erreichenden **Handlungsziele in Form von technisch zu realisierenden Leistungsfähigkeiten** und Merkmalen zusammen mit den Technikern **konkretisiert**.

Beispiel: Fotoapparat

- Ausreichend Kapazität, um recht viele Bilder machen zu können.
 - Speicherkapazität für ca. 1000 Bilder bei 5 MB/Bild bzw. 500 Bilder bei 10 MB/Bild
- Fotos sollten unter (fast) allen Lichtverhältnissen möglich sein.
 - Stativbefestigung / Blitz mit Gegenlichtfunktion
- Bilder sollten ohne Computeranschluss zu betrachten sein.
 - ausreichend großes und detailreiches Display

Fortsetzung Beispiel: Fotoapparat

- Eine Selbstauslöserfunktion sollte vorhanden sein.

 - Selbstauslösermenü

- Eine Zoommöglichkeit sollte gegeben sein.

 - digitaler und optischer Zoom

- Die notwendigen Einstellungen sollten weitestgehend automatisch erfolgen; für Experten sollten aber Einflussmöglichkeiten bei Belichtungszeit und Blende zur Verfügung stehen.

 - Expertenmodus mit individuellen Einstellmöglichkeiten

- **Merkmals-Funktions-Matrix (QFD → Quality Function Deployment):**

- Die technischen Spezifikationen gewährleisten aber noch nicht zuverlässig, dass die Bedürfnisse der potenziellen Benutzer ausreichend befriedigt werden oder technische Spielereien (unübersichtliche Funktionsvielfalt) die Befriedigung ihrer Bedürfnisse nicht sogar konterkarieren.

- **Empirische Lösung:** Beurteilung der **Wichtigkeit der Handlungsziele** und der **technischen Umsetzungen/Merkmale** durch das **Objekt/System** durch potenzielle Nutzer und **Bestimmung der Passung**.

- Für Liste mit Handlungszielen und Liste mit Merkmalen des Systems: Skalierung der Wichtigkeit von 1 (relativ geringes Gewicht) bis 9 (sehr hohes Gewicht).

Beispiel Fotoapparat: Merkmals-Funktions-Matrix enthält Ratings, die wiedergeben, wie gut ein Merkmal des Produkts geeignet ist, die Ziele des Nutzers zu verwirklichen.

Produktmerkmale					
Ziele des Nutzers	Gewicht des Ziels	hohe Speicherkapazität	Blitz	Zoom	Automatik abschaltbar
viele Bilder	9	9	3	1	1
alle Lichtverhältnisse möglich	3	3	9	1	3
Bilder betrachten ohne Computer	9	3	3	1	1
unterschiedlich große Bildausschnitte	1	1	1	9	1
Summe: Gewicht * Rating		118	82	49	28

Summenwerte: Je höher, desto besser wird das Erreichen der Ziele des Nutzers durch das technische Merkmal des Produkts gefördert.

- **Berücksichtigung von Usability-Kriterien:**

Zu den Systemspezifikationen können zusätzlich auch Anforderungen an die Usability gehören:

- besonders schnelle Bedienbarkeit
- Fehlertoleranz
- besonders geringe Belastungen für die Muskulatur (z.B. PC-Tastatur) und Stützung der Wirbelsäule (z.B. Bürostuhl)
- Rückgängigmachen des letzten Betriebszustands (z.B. bei Software)
- Vermeidung von Datenverlust
- höchst mögliche Sicherheit gegenüber unberechtigtem Eindringen ins System (z.B. bei Computernetzwerken).

- **Dokumentationen: Handbücher und Gebrauchsanweisungen**

Gebrauchsanweisungen unterliegen ebenfalls ergonomischen Gestaltungsrichtlinien.

- **Gliederung nach Wichtigkeit:** Sie müssen stets das Wichtigste zuerst darstellen: alle sicherheitsrelevanten Dinge müssen so erklärt werden, das selbst der ungebildete Laien sie verstehen kann.
- **Untergeordnete Gliederung nach Handlungsablauf:** Aufgrund der Ergebnisse der Arbeitsanalyse weiß man, bei welchen Aufgaben in welcher Reihenfolge welche Handlungen durchzuführen sind; in genau dieser Reihenfolge sind die Anweisungen darzustellen.
- **Unterstützung durch Abbildungen:** „Ein Bild sagt mehr als tausend Worte“ → leicht verständliche, dreidimensionale Graphiken oder detailreiche Fotos
- **Empirische Prüfung:** Anleitungen sollten empirisch auf Verständlichkeit durch potenzielle Nutzer geprüft werden! Wird fast nie gemacht.

- **Prototypen:**

- **Prototyping:** Beginn zum frühest möglichen Zeitpunkt mit Prototypen, die mit 3D-Druckern oder computergesteuerten Fräsen hergestellt werden (rapid prototyping).

Prototyp: → reicht vom einfachen Papiermodell

→ über Produkt mit nur wenigen realisierten Funktionen ohne ästhetische Gestaltung und noch nicht maßstabgetreuem Aufbau

→ bis zu den ersten Exemplaren des voll funktionsfähigen Endprodukts

- Vorteile der Verwendung von Modellen und Prototypen:

- Erkenntnisse aus der Arbeitsanalyse können überprüft werden, z.B. bezüglich der Griffbarkeit und Rutschfestigkeit eines Gehäuses → nur Gehäuseprototyp erforderlich
- Möglichkeit, Ideen schnell zu konkretisieren und einer kritischen Begutachtung zu unterziehen; Erleichterung der Kommunikation
- Grundlage einer **heuristischen Evaluation** ↓
- Gegenstand von **Usability-Tests** ↓

- **Heuristische Evaluation:**

- Systematische Evaluation des Produkts bezüglich der Frage, inwieweit alle Richtlinien der ergonomischen Gestaltung (teilweise in Form von DIN-Normen festgelegt → s. Kap. 2 „Software-Ergonomie“) erfüllt wurden → Check-Listen
- wird von Usability-Experten vorgenommen ohne Beteiligung von potenziellen Nutzern;
- Vergleich verschiedener Designalternativen unter dem Gesichtspunkt, welche Alternative die Richtlinien ergonomischer Gestaltung am besten realisiert;
- ein Usability-Experte findet im Durchschnitt nur 35% der ergonomischen Probleme → wenigsten 3, besser 5 Experten notwendig
 - zunächst unabhängig von einander arbeitend,
 - am Ende werden in einer Konferenz die Ergebnisse diskutiert und Konsens über die endgültige Gestaltung hergestellt;
 - erst dann sollte man mit den Designern diskutieren.

- **Usability-Tests:**

Usability: Grad in dem ein Produkt/System leicht bzw. benutzerfreundlich zu bedienen ist unter Berücksichtigung mindestens folgender Faktoren:

- **Selbsterklärungsfähigkeit:** Das Produkt/System sollte dem Nutzer alle zu seiner Handhabung erforderlichen Informationen liefern (z.B. sollte man dem Gerät ansehen, was man damit machen kann) → das sollte Bedienungsanleitungen/Handbücher weitestgehend überflüssig machen!
- **Erlernbarkeit:** Der Umgang mit dem Produkt/System sollte schnell erlernbar sein (wenig Training erfordern).

- **Effizienz:** Nach Erlernen der korrekten Handhabung des Produkts/Systems sollte die Produktivität einen höheren Grad erreichen als ohne Verwendung dieses Produkts/Systems (z.B. im Vergleich zur Arbeit ohne maschinelle Unterstützung oder im Vergleich zu einem alten Produkt/System).
- **Erinnerbarkeit:** Die beim Erlernen der Handhabung des Produkts/Systems zu behaltenden Informationen sollten möglichst gering und leicht erinnerbar sein (ggf. sollte das System Erinnerungsunterstützungen anbieten → Selbsterklärungsfähigkeit).

- **Flexibilität/Adaptierbarkeit:** Das Produkt/System sollte auf die spezifischen Bedürfnisse der Nutzer einstellbar sein (z.B. bei Software Menüs für Novizen; Tastenkurbefehle für Experten).
- **Fehlertoleranz:** Die Handhabung des Produkts/Systems sollte zu möglichst wenigen Fehlern führen; wenn sie dennoch auftreten, sollten sie leicht korrigierbar sein (z.B. Rückgängigmachen einer bereits ausgeführten Anweisung bei Software); das Produkt/System darf durch keinen Handhabungsfehler in einen instabilen Zustand (fataler Fehler) mit Systemzusammenbruch geraten (z.B. Stabilität von Gehäusen sollte so groß sein, das Fallenlassen nicht gleich zu Defekten führt).

- **Benutzerzufriedenheit:** Die Handhabung des Produkts/Systems sollte den Nutzer zufrieden stellen, d.h. für ihn angenehm sein, ihn nicht überlasten, vielleicht sogar Spaß machen, ästhetischen Ansprüchen genügen (deswegen auch als „joy of use“ bezeichnet) (z.B. keinen Ärger über zu lange Reaktionszeiten eines Programms wegen fehlender Rückmeldungen provozieren).

Usability-Test: Empirische Überprüfung des Zutreffens der genannten Usability-Kriterien mit zahlreichen potenziellen Nutzern des Produkts/Systems.

- Sollte mehrfach im Rahmen des Evaluationszyklus mit den unterschiedlich fortgeschrittenen Prototypen geschehen: anfangs mit wenigen Versuchspersonen, mit zunehmend größeren Stichproben bei den späteren Iterationen.

- Typischer Bereich, in dem Usability-Tests von vielen Firmen standardmäßig durchgeführt werden, ist die Softwareentwicklung →

Software-Ergonomie

Kapitel 2: Grundlagen der Software-Ergonomie

Ziel der Software-Ergonomie: Gestaltung der Software so, dass sie den Bedürfnissen der sie benutzenden Personen insbesondere im Kontext der Arbeit gerecht wird.

Das erfordert die **Berücksichtigung psychologischer Erkenntnisse** über die menschliche Handlungssteuerung unter Beachtung folgender die Handlungsmöglichkeiten begrenzender Funktionen:

- **Wahrnehmung:** Der Dialog mit dem Computer geschieht mittels Bildschirm (Zeichen- und Objektdarstellung, Berührung von Objekten); deshalb spielen Bedingungen eine Rolle, die geeignet sind, die Wahrnehmung der dargebotenen Informationen zu erleichtern bzw. zu erschweren.

- **Aufmerksamkeit:** Gleichzeitig auf dem Bildschirm dargebotene Informationen erfordern eine Auswahl der augenblicklich benötigten Informationen (Selektion); komplementär dazu kann gerade nicht benötigte Information unter bestimmten Umständen ablenkend wirken (automatische Aufmerksamkeitsattraktion).
- **Gedächtnis und Wissen:** Manche Software beansprucht das Gedächtnis enorm, weil der Nutzer viel lernen und das Wissen stets verfügbar haben muss (z.B. die alten Betriebssysteme, die noch mit Befehlen arbeiteten); andere Software stellt dem Benutzer alle zur Entscheidung erforderliche Information bereit und entlastet so das Gedächtnis.
- **weitere Faktoren:** Emotionen (Zufriedenheit), Motivation (Neugier)

2.1 Psychologische Grundlagen der Software-Ergonomie

2.1.1 Wahrnehmung

Im Folgenden werden nur die für die Gestaltung von Benutzerschnittstellen (z.B. Fenstern) direkt relevanten Wahrnehmungsfaktoren skizziert:

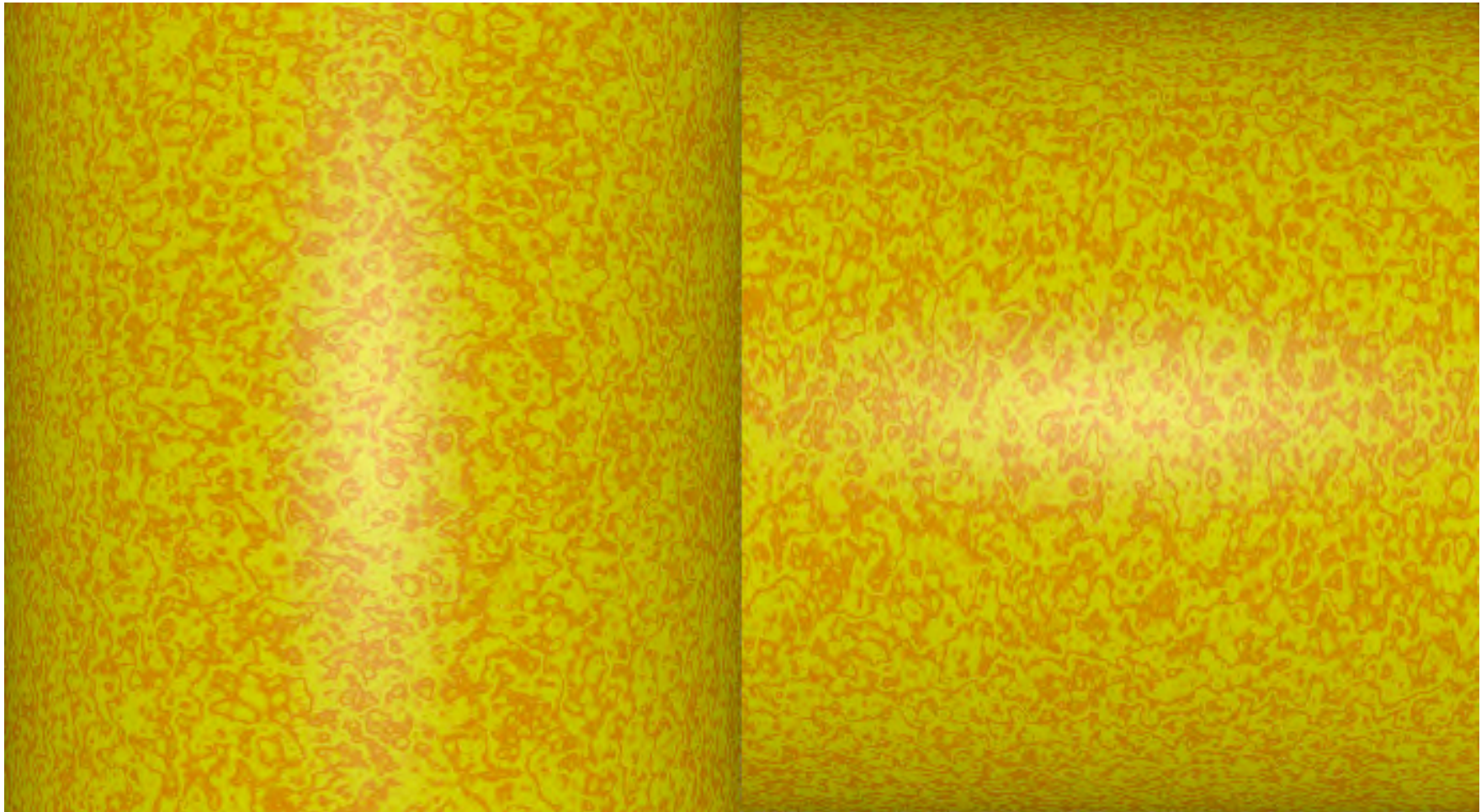
- **Figur-Grund-Differenzierung:** Leistung einer frühen Stufe der visuellen Informationsverarbeitung.
→ Funktion: Aufbau einer visuellen Repräsentation der Umwelt, in der einzelne Objekte von einem Hintergrund, auf dem sie sich befinden, zu trennen sind, damit die Objekte einzeln beachtet werden können.

Welchen Eigenschaften charakterisieren eine Figur?

- Symmetrisch geformte Bereiche,
- horizontal und/oder vertikal oder in der Tiefendimension konvexe (nach außen gewölbte) Bereiche,
- im Vergleich zu anderen relativ kleine Komponenten,
- streng vertikal oder horizontal ausgerichtete Flächen,
- Merkmale, die auf räumliche Tiefe eines Bereichs hindeuten
- Hell-Dunkel-Kontraste.



Rubinscher Becher

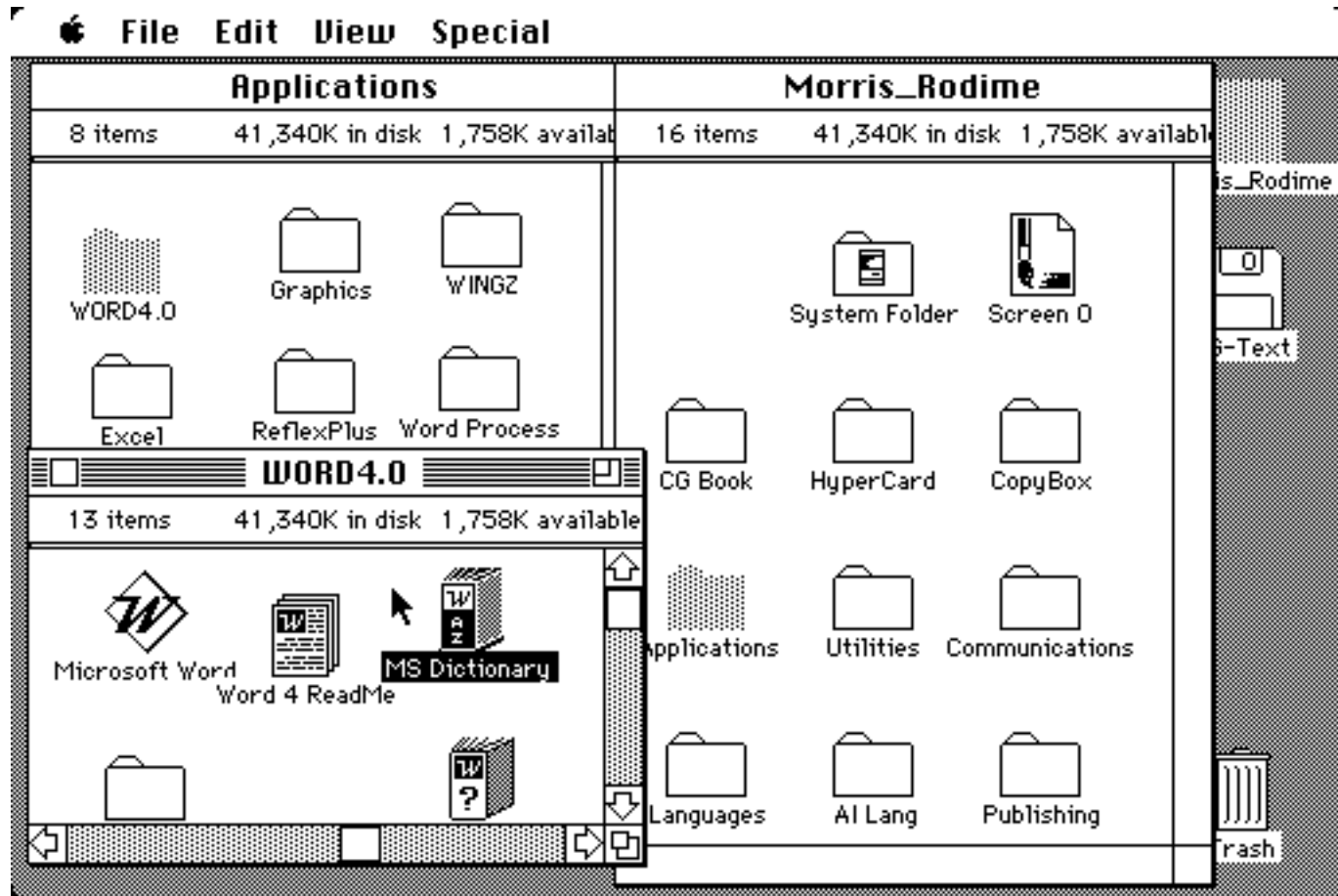


Tiefeninformation durch Texturierung und künstliche Lichteffekte

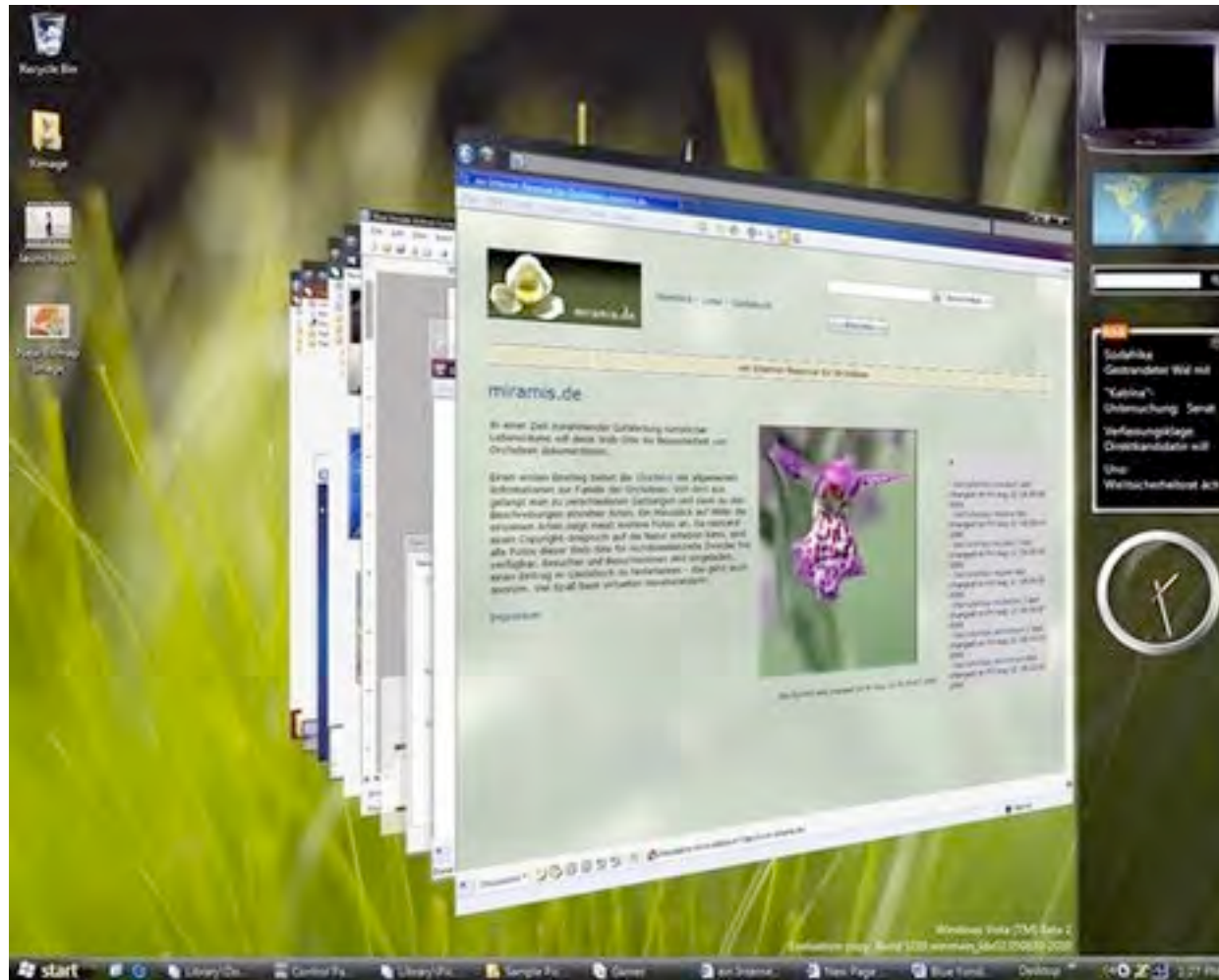
→ Konsequenzen für die Bildschirmgestaltung:

- **Merkmale**, die die Figur-Grund-Differenzierung unterstützen (s.o.), müssen die Graphiker bei der graphischen Gestaltung von Objekten (z.B. Fenstern, Icons) **berücksichtigen**, damit die auf diese Objekte bezogenen Handlungen (z.B. Mausbewegungen, Berührungen) optimal (schnell und fehlerfrei) gesteuert werden können.

- **Je mehr dieser Merkmale** gleichzeitig vorhanden sind, **desto deutlicher wird die Segregation** der Objekte von ihrem Hintergrund wahrgenommen und desto schneller und einfacher ist das Erkennen der Objekte und die Ausführung der auf sie bezogenen Handlungen.



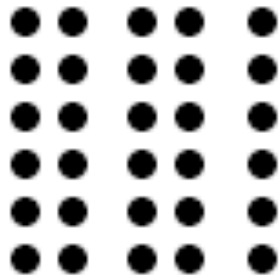
*Schlechte visuelle Trennung von Figur und Grund
beim alten Mac-Betriebssystem*



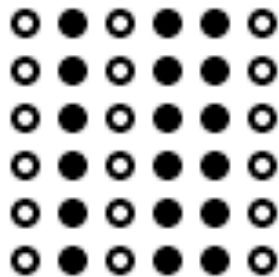
deutliche Trennung von Figur und Grund durch dreidimensionale Darstellung der Fenster und Verwendung von Farben

- **Ausgliederung von Objekten:** Leistung einer frühen Verarbeitungsstufe der visuellen Informationsverarbeitung
 - Funktion: Aufbau einer visuellen Repräsentation der Umwelt, in der die visuellen Merkmale so gruppiert sind, dass isolierte Objekte (Gestalten), die von anderen Objekten deutlich getrennt sind, repräsentiert werden; automatischer, intentional nicht beeinflussbarer Vorgang.

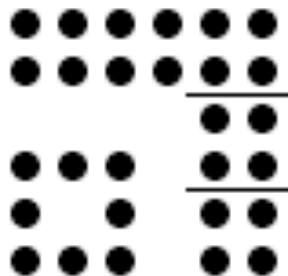
Die Regeln, nach denen das visuelle System diese Organisation des visuellen Feldes vornimmt, werden als **Gestaltgesetze** bezeichnet.



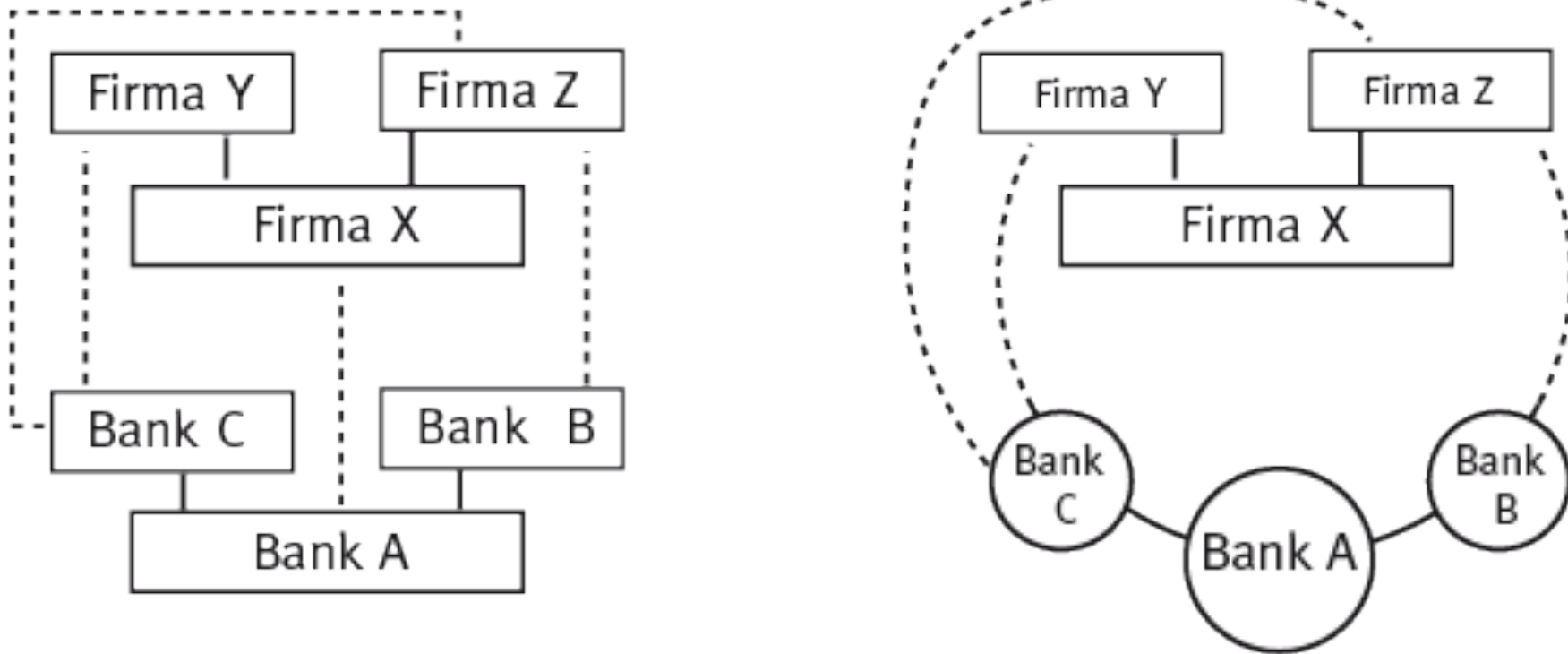
Gesetze der Nähe



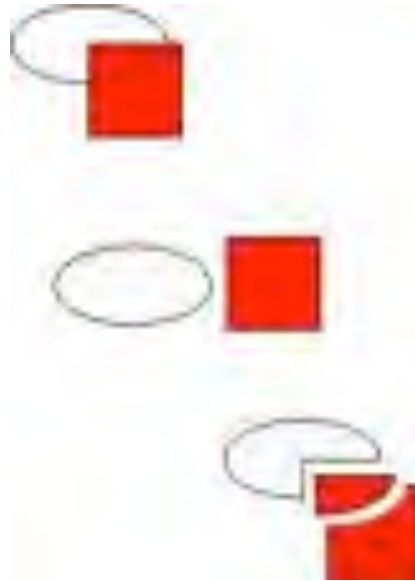
Gesetz der Ähnlichkeit



Gesetz der Geschlossenheit



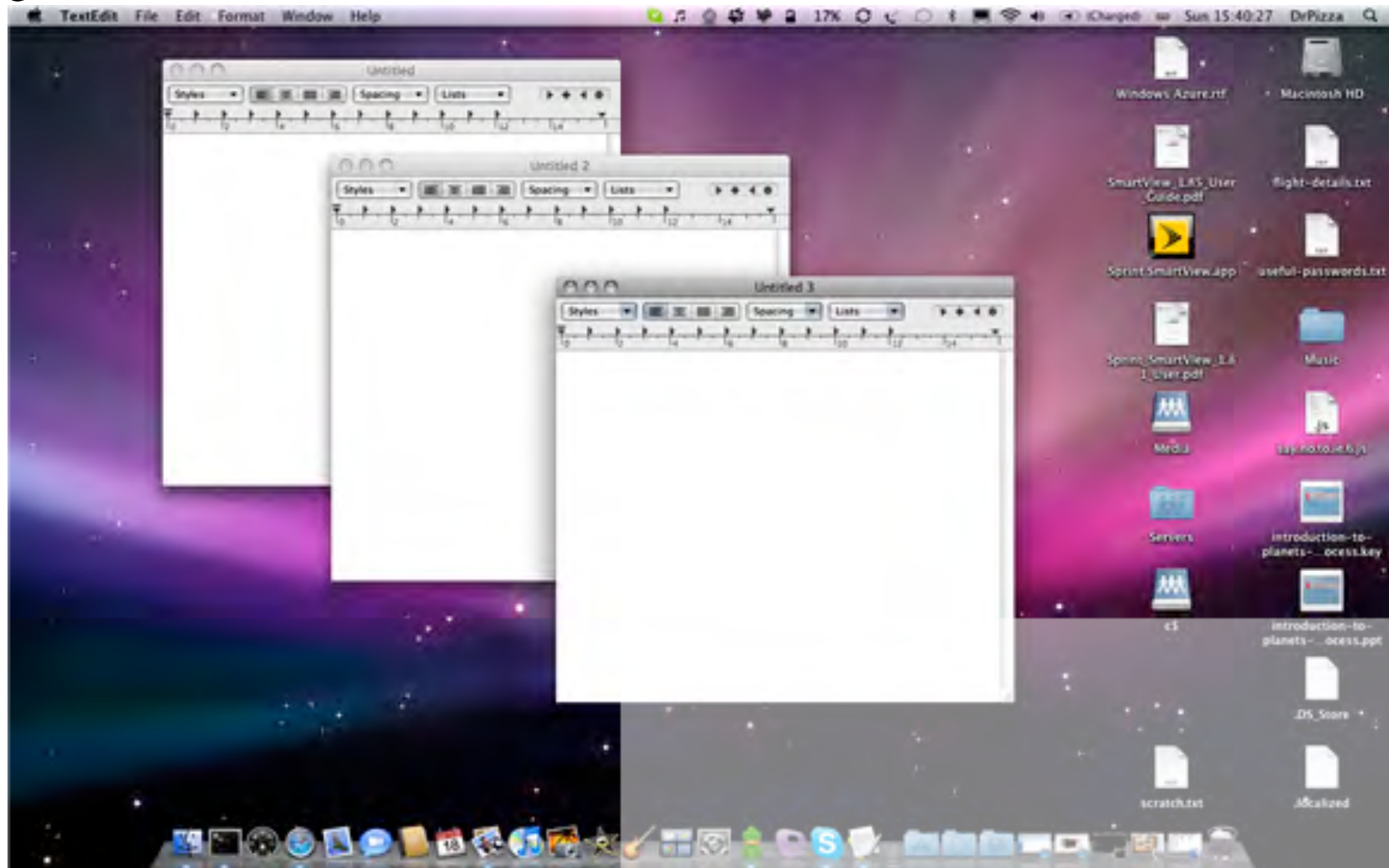
*Beispiel für eine deutlichere perzeptive Segregation von Bildschirmobjekten aufgrund Berücksichtigung der Gestaltgesetze der **Nähe** und der **Ähnlichkeit***



*Beispiel für die Wirkung des **Prägnanzgesetzes**:*

Die in der 1. Zeile dargestellten beiden Figuren werden nicht wie in der 3. Zeile dargestellt wahrgenommen sondern als zwei getrennte, sich überlappende Objekte.

Diese Form einer deutlichen Segregation von Objekten wird Platz sparend bei Betriebssystemen mit mehreren geöffneten Fenstern angewendet:



2.1.2 Aufmerksamkeit

selektive Aufmerksamkeit: Auf der Grundlage einer Repräsentation der Objekte wird aus der Menge wahrgenommener Objekte dasjenige fokussiert, dessen Merkmale zur Ausführung einer intendierten Handlung benötigt werden.

Zweck:

1. Die Merkmale aller anderen Objekte werden davon ausgeschlossen, die Handlung zu steuern (Vermeidung von Interferenz);
2. die Merkmale des beachteten Objekts werden schneller und genauer identifizierbar als die anderer Objekte.

→ **Anwendungen in der Softwaregestaltung:** Systematisches Durchsuchen einer Bildschirmseite ist ein zeitraubender, fehleranfälliger Vorgang. Besser ist es, automatisch die Aufmerksamkeit auf das jeweils zunächst zu beachtende Objekt einer Bildschirmseite zu lenken. → automatische Aufmerksamkeitslenkung

Willkürliche vs. unwillkürliche Aufmerksamkeitslenkung: Ein selektives Beachten eines Objekts kann willkürlich initiiert werden, aber auch unwillkürlich (automatisch). Dabei sind softwareergonomisch die automatischen Lenkungsformen am interessantesten.

Formen automatischer Aufmerksamkeitsattraktion:

- **Onsets/Offsets und Blinken:** Jeder Kontrastprung von dunkel nach hell (Onset) oder umgekehrt (Offset, schwächere Wirkung) bewirkt eine automatische Hinwendung der Aufmerksamkeit auf den Ort der Veränderung.

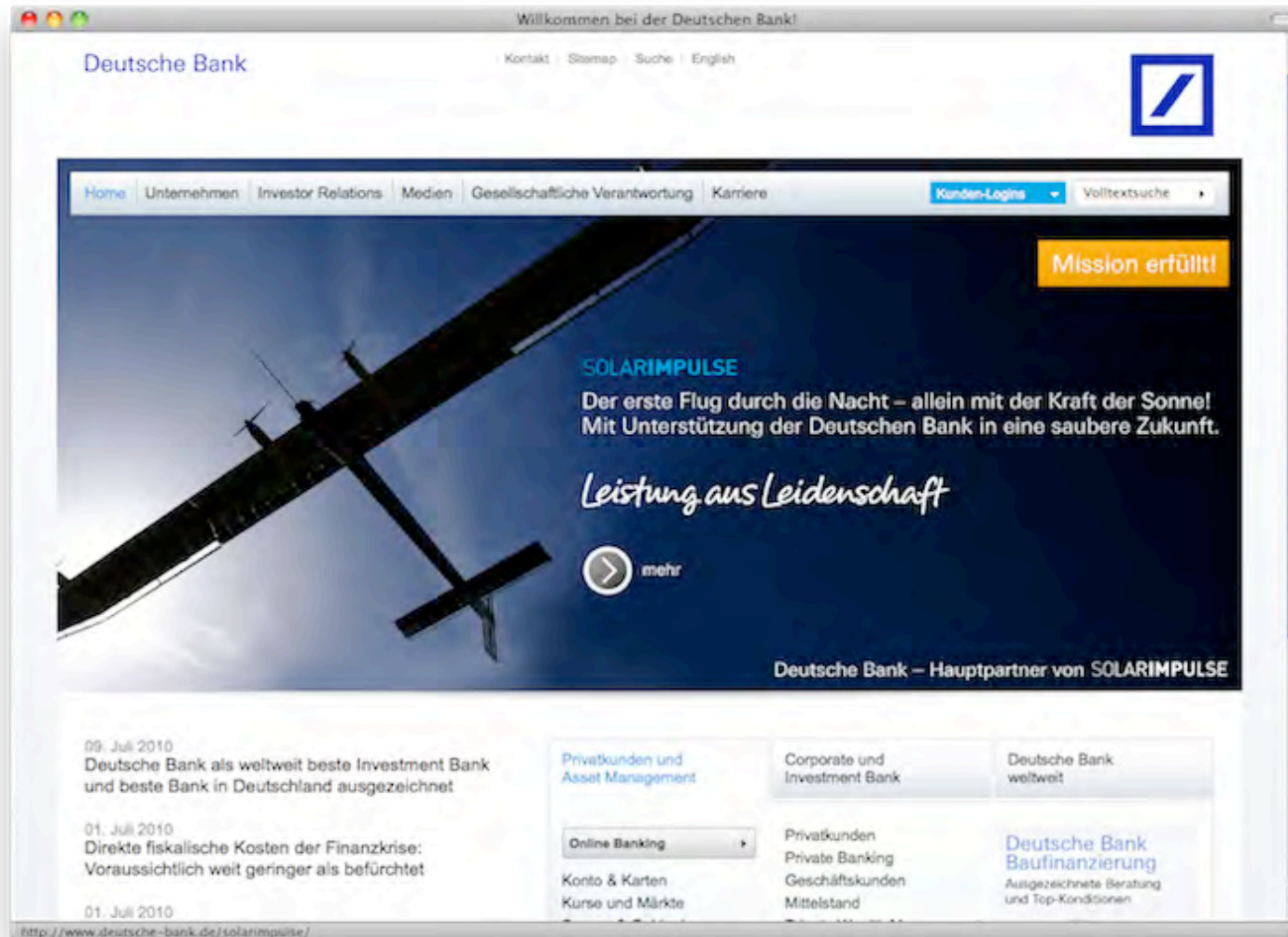
Beispiele: Blinken des Cursors zwecks schnellen Auffindens; Blinken des Textes einer Fehlermeldung.

- **Bewegungen:** Insbesondere in der Peripherie auftretende Bewegungen lösen eine Aufmerksamkeitszuwendung aus. Finden gleichzeitig an verschiedenen Orten Bewegungen statt, führt dies nicht mehr zu einer automatischen Aufmerksamkeitsattraktion.

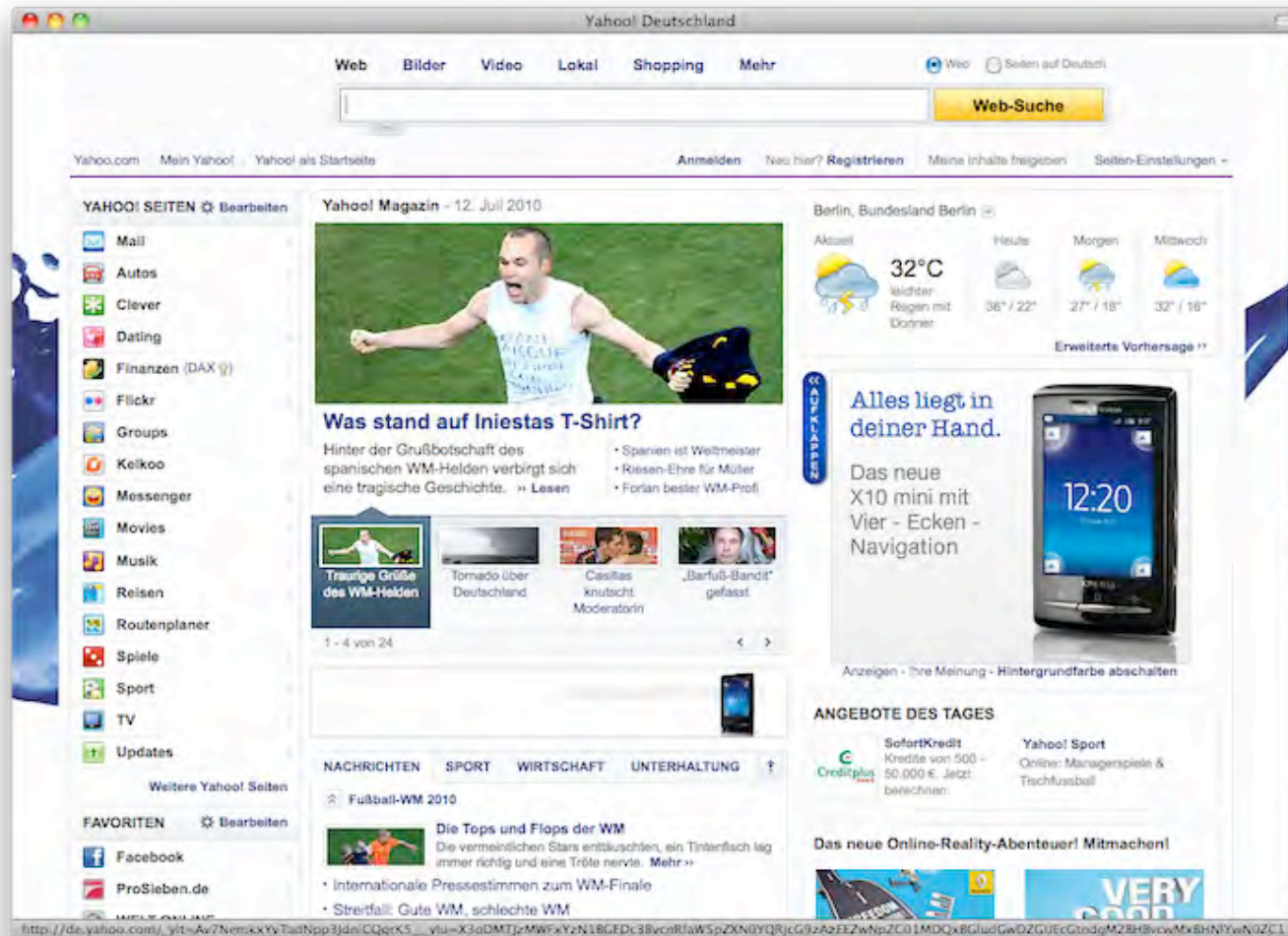
Beispiele: Viele gleichzeitig Bewegungsinformation enthaltende Werbebanner auf Internetseiten.

- **Farben:** Farben führen nur dann zu einer automatischen Aufmerksamkeitsattraktion, wenn **eine bestimmte Farbe für nur ein Objekt** eingesetzt wird und der Rest des Bildschirm farblich anders und möglich wenig bunt gestaltet ist (Popout-Effekt → s.u.).

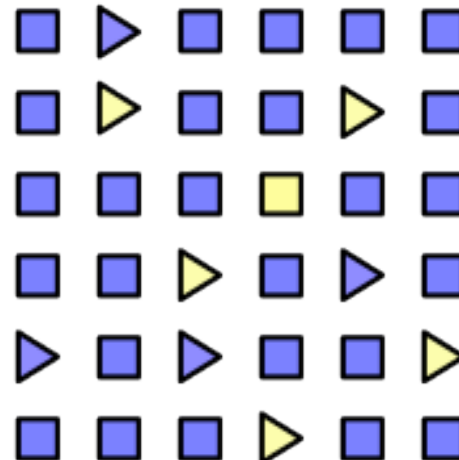
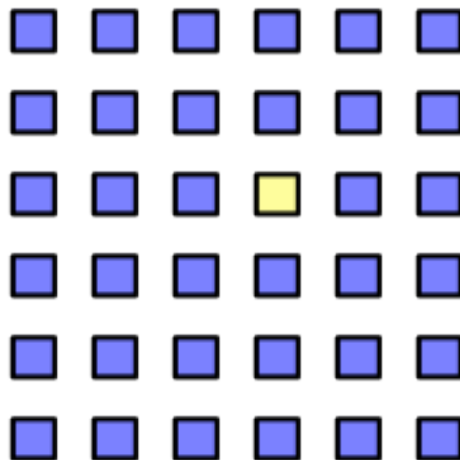
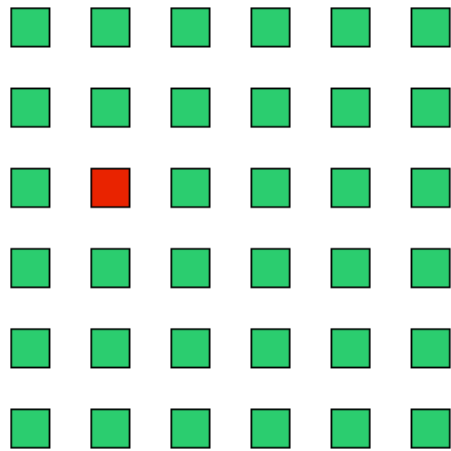
Login-Feld hier farblich markiert und daher gut zu finden (besser nicht blau, weil bereits viel blau verwendet wurde):



Hier ist das Login-Feld nicht singular farblich markiert und erfordert daher eine aufwendige Suche:



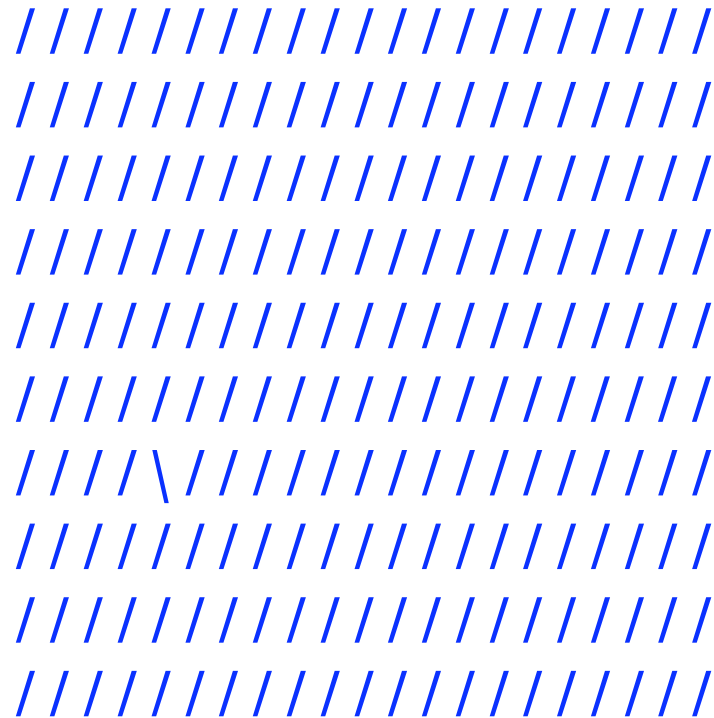
Popout durch Farbe als singuläres Merkmal:



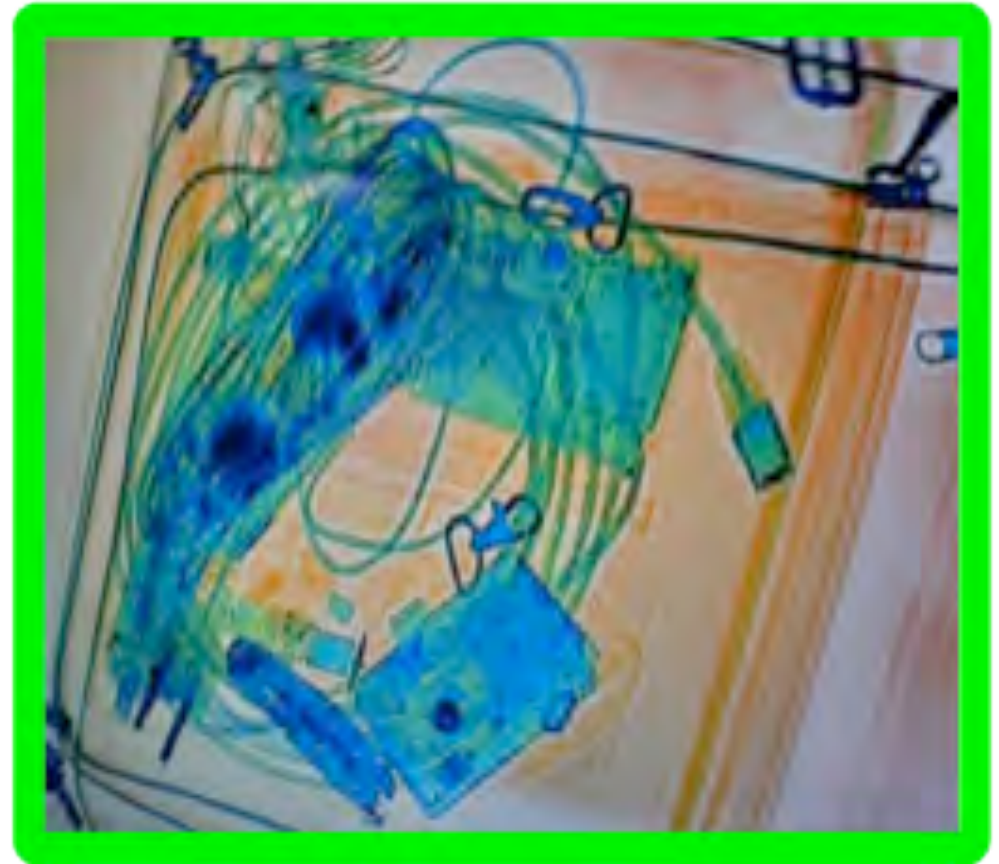
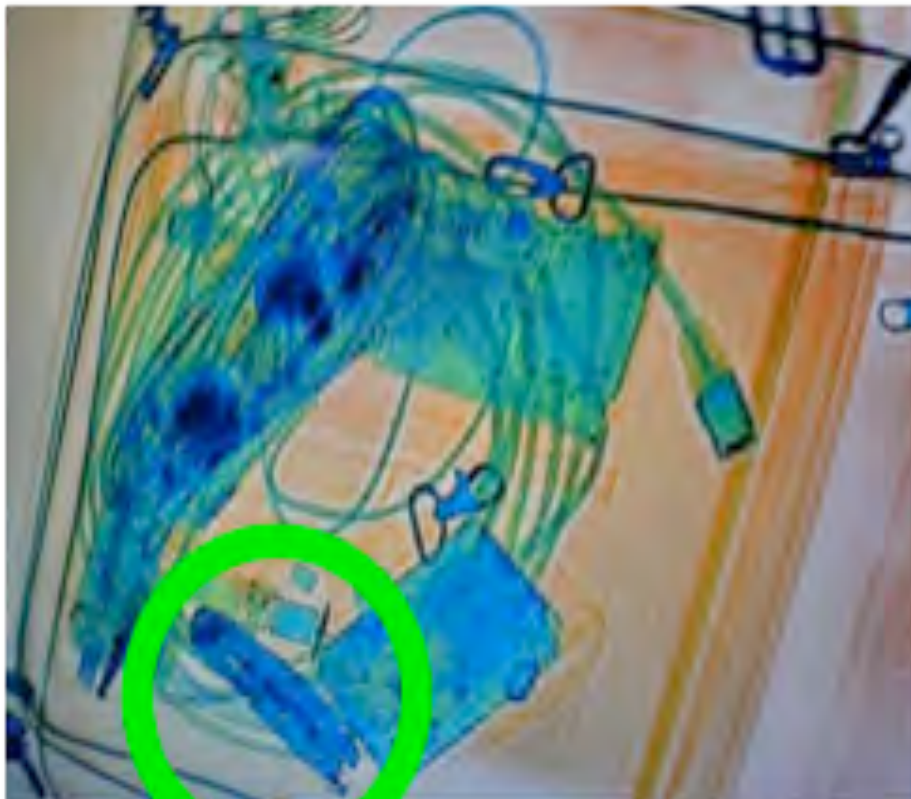
links → Popout, rechts → kein Popout (gerichtete Aufmerksamkeit bei Suche notwendig)

Popout aufgrund anderer **singulärer** Merkmale:





Popout von abweichenden Texturelementen



links → automatische Unterstützung des Suchens durch Popout-Markierung (Gefährlicher Gegenstand bei Fluggepäckuntersuchung)

2.1.3 Gedächtnis und Wissen

- Jede Software erfordert zumindest ein wenig **Lernen**, wie man damit die gewünschten Handlungsziele erreichen kann. Das Ergebnis des Lernens ist als Wissen im **Langzeitgedächtnis** gespeichert.
- Komplexes Wissen kann recht aufwändige **Abrufprozesse** erforderlich machen → **Erinnerungsfehler**
- Je differenzierter und daher umfangreicher benötigtes Wissen ist desto größer ist die Wahrscheinlichkeit von Abruffehlern → **Handlungsfehler**
- **Grundsatz:** Je weniger aus dem Gedächtnis abzurufendes Wissen für die Verwendung von Software nötig ist desto **benutzerfreundlicher** ist die Software.

- **Methoden zur Entlastung des Gedächtnisses bei der Softwarebenutzung:**

- **Konsistenz:** Wird für die Anwendung verschiedener Programme auch immer neues und ganz unterschiedliches Wissen benötigt, ist der Lernaufwand für jede einzelne Anwendung größer als wenn immer wieder weitgehend dasselbe Wissen verwendet werden kann.

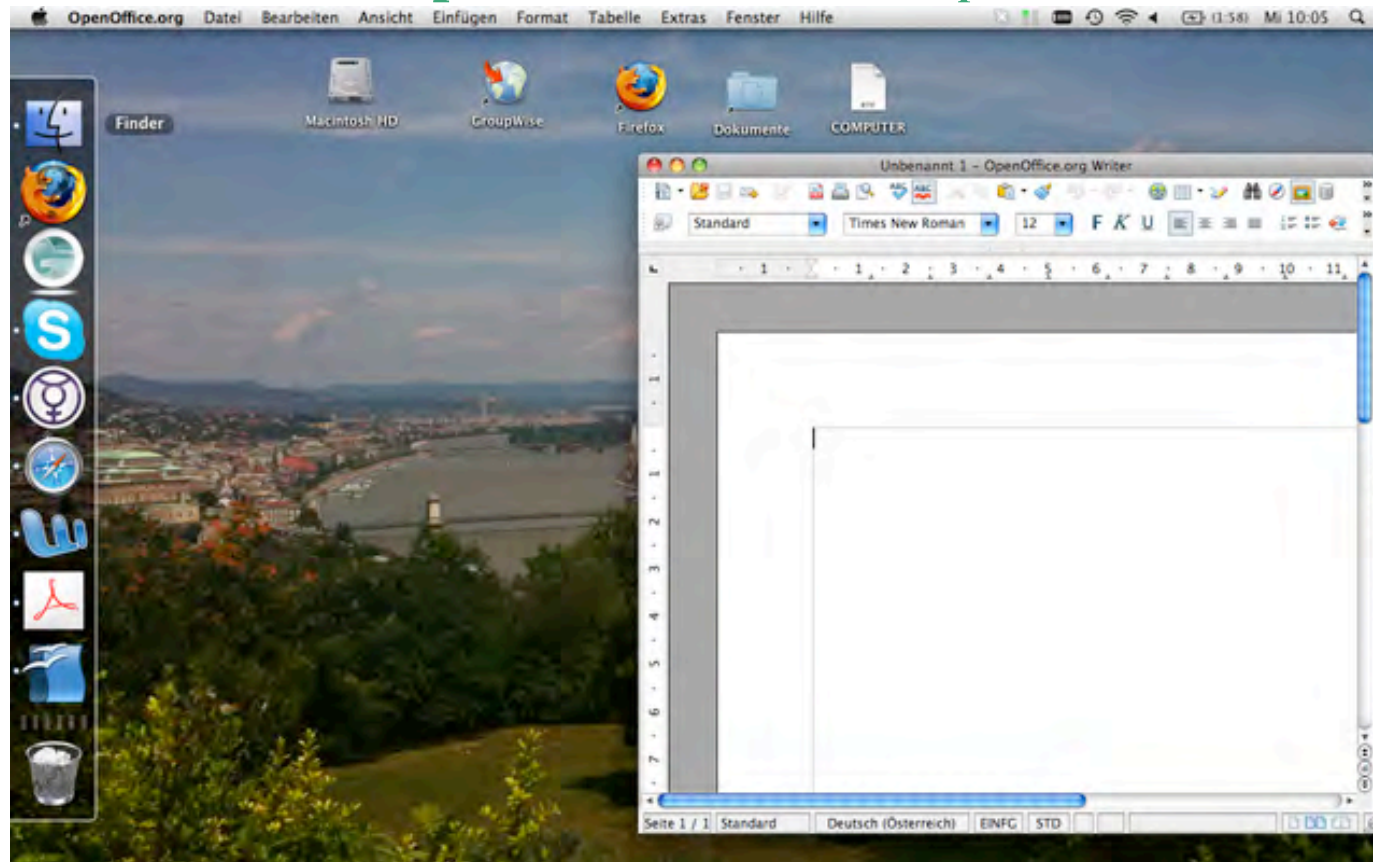
- **Konsistenz:** Die Methoden zur Bedienung der Software sind weitestgehend gleich unabhängig davon, um welche Software es sich handelt.

Beispiel: Konsistenz wird erreicht durch übereinstimmende Orte bestimmter Menüpunkte (→ Normierungen der Betriebssystemhersteller, z.B. alle Datei-Menüpunkte befinden sich beim Apple OS bei jedem Programm an der 2. Position von links in der Menü-Zeile).

- **Metaphern:** Das Langzeitgedächtnis wird entlastet, wenn bereits vor Einarbeitung in die Benutzung einer Software vorliegendes Wissen möglichst leicht in Form von **Handlungsanalogien** auf die neue Situation der Softwarebenutzung übertragen werden kann.

→ Einsatz von **Metaphern**

Beispiel: *Schreibtischmetapher*



Metapher: Transfer von Wissen von einer vertrauten Situation auf eine analog strukturierte andere Situation (hier: von der realen Welt, ihren bekannten Objekten und den damit durchgeführten Tätigkeiten auf die virtuelle, auf den Bildschirm übertragene Situation, die ganz ähnliche Handlungen zulässt).

Zur Schreibtischmetapher gehören weitere **metaphorische Objekte:**

Ordner



Tools



Papierkorb



Voraussetzung der Metaphernverwendung: Strukturelle Übereinstimmung der mit den Objekten durchführbaren Handlungen in der realen Welt und der metaphorischen Welt des Systems.

Ist nicht immer und in allen Aspekten erfüllt: z.B. realer Papierkorb steht nicht wie der virtuelle auf dem Schreibtisch.

→ Führt das zu Handlungsfehlern?

→ **Usability-Evaluation muss also auch die Verwendbarkeit von Metapher prüfen!**

Entwicklung geeigneter Metaphern:

- Die **Arbeitsanalyse** muss Ergebnisse liefern, die Aufschluss geben über erforderliche Gegenstände, Werkzeuge, Bedienelemente aus der realen Welt, deren virtuelle Entsprechungen eine vergleichbare Funktionalität ausweisen müssen.
- Die **Usability-Evaluation** muss dann potenzielle Benutzer dabei beobachten, wie sie vergleichbare Tätigkeiten virtuell bewältigen und inwieweit sie mit der Metapher zurechtkommen. Wenn möglich, sollten verschiedene Metaphern vergleichend auf unterschiedliche Brauchbarkeit überprüft werden.

mentale Modelle: Kognitive Repräsentation, die das Wissen des Benutzers über die von ihm vermutete Funktionsweise eines Gerätes/Systems zum Gegenstand hat und die ihn bei der Planung und Ausführung seiner Handlungen leitet.

Beispiel:

- Aus einem Reiseführer entnimmt man eine zutreffende verbale Beschreibung eines Wegs zu einem bestimmten Ziel.
- Die verbale Beschreibung wird in eine visuelle Vorstellung übersetzt → kognitive Landkarte (= mentales Modell des Wegs)
- Ist die Übersetzung in eine visuelle Vorstellung korrekt, findet man das Ziel leicht; stimmen mentales Modell und Realität nicht überein, kommt es zu Handlungsfehlern: man verirrt sich.

- **Mentales Modell ist korrekt:** Wenn das mentale Modell mit der tatsächlichen Funktionsweise übereinstimmt, kann der Benutzer sein Wissen ohne erneutes Lernen zur Steuerung seiner Handlungen einsetzen.

- **Mentales Modell weicht von Realität ab:** In dem Maße, in dem das mentale Modell in handlungsrelevanten Aspekten von der tatsächlichen Funktionsweise des Gerätes/Systems abweicht, kann es zu Handlungsfehlern kommen.

- **Passung von mentalem Modell und Realität herstellen:** Bei Abweichungen von mentalem Modell und tatsächlicher Funktionsweise ist ein verstärktes Training nötig, das zum Aufbau eines adäquaten mentalen Modells führt.

Beispiel für Diskrepanz zwischen durch Software suggeriertem mentalem Modell und Realität: Funktion des Papierkorbs beim Mac-OS

mentales Modell:

- Objekte in den Papierkorb zu schieben führt zu deren Löschung.
- Speichermedien in den Papierkorb zu schieben, sollte daher ebenfalls zur deren Lösung führen.

→ Problem: „Wie wirft man Speichermedien aus?“

- Hilfefunktion sagt: „In den Papierkorb schieben!“
- Widerspruch zum mentalen Modell!

→ Ungutes Gefühl bleibt, wenn man Speichermedien in den Papierkorb schiebt, weil mentales Modell durch eine Ausnahme ergänzt werden muss, die den Widerspruch jedoch nicht aufhebt.

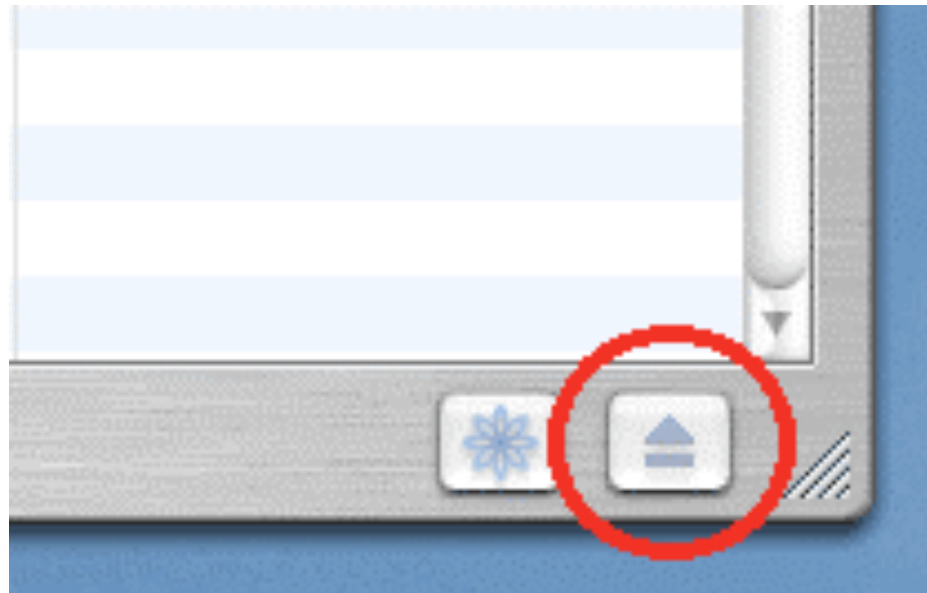
Lösung des Problems:

Auswurfschaltersymbol



auf dem Schreibtisch oder an

dem dem Speichermedium zugeordneten Fenster:



2.2 Ergonomische Grundsätze der Dialoggestaltung

- **DIN EN ISO 9241-110:** Für Gestaltung und Evaluation der Interaktion zwischen Benutzer und Software gibt es eine DIN-Norm (hier dargestellt in der Version von 2004), die allerdings **nicht rechtsverbindlich** ist, deren Einhaltung also nicht staatlich überwacht wird und nicht einklagbar ist.

- **Sieben Grundsätze:** Diese Norm schlägt sieben Grundsätze zur Gestaltung der Software-Schnittstelle zwischen Mensch und Computer vor, die aus allgemein gültigen Prinzipien der Ergonomie für den Spezialfall der Software abgeleitet wurden.

- **keine Überschneidungsfreiheit:** Diese sieben Grundsätze, erst recht die EVADIS-Richtlinien (s. Kap. 2.3), sind keineswegs als überschneidungsfreie Kategorien zu betrachten; das ist auch nicht notwendig, da es nicht auf eine eindeutige Klassifikation ankommt, sondern auf die Identifikation von ergonomischen Gestaltungsfehlern gleich welchen Typs.

- **Anwendung:** Schon bei der Planung und Programmierung der Software sollten sie berücksichtigt werden; spätestens bei der ersten und allen nachfolgenden Usability-Evaluationen werden sie zu berücksichtigen sein.

2.2.1 Aufgabenangemessenheit

Dieses Kriterium folgt aus den in der Vorgängerversion der DIN-Norm (ISO-9241-11) definierten Eigenschaften **Effektivität** und **Effizienz**, die von einem Softwaresystem erfüllt sein müssen.

- **Effektivität:** Genauigkeit und Vollständigkeit, mit der ein Nutzer Aufgaben zu lösen vermag.
- **Effizienz:** Verhältnis von Aufwand zu Genauigkeit und Vollständigkeit der Aufgabenergebnisse.

→ **Systemunterstützung:** Alle Tätigkeiten, die nicht direkt mit dem Erreichen eines Aufgabenziels zusammenhängen, sollten vom System übernommen werden (z.B. Speichern, Sicherheitskopien). Der Nutzer sollte sich nur mit so genannten externen Aufgaben, die sich direkt aus der Arbeitsaufgabe ergeben, beschäftigen müssen (z.B. Eingaben, Auswahl von Funktionen, die Bearbeitung einer Aufgabe ermöglichen) (zur Begründung der Kriterien s.a. Herczeg, 2005).

→ **Minimum an Arbeitsschritten:** Eine Aufgabe sollte mit so wenig Arbeitsschritten wie möglich realisierbar und die Benutzeroberfläche den Kenntnissen und Fertigkeiten des Nutzers angemessen gestaltet sein.

Bezeichnung des Kriteriums	Aufgabenangemessenheit
Definition nach ISO- Norm	<i>Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgaben effektiv und effizient zu erledigen.</i>
Beispiele für mögliche Anwendungen	<ul style="list-style-type: none"> • <i>Es werden nur Informationen ausgegeben, die mit der Erledigung der Arbeitsaufgabe zusammenhängen.</i> • <i>Wiederkehrende Tätigkeiten werden weitestgehend automatisiert (sollte aber abschaltbar sein).</i> • <i>Sinnvolle Standardwerte und Positionsmarken werden gesetzt, sind aber leicht änderbar.</i> • <i>Pflichtfelder in Formularen werden vermieden; wenn nötig (z.B. aus Sicherheitsgründen), müssen sie markiert sein.</i> • <i>Zwischenergebnisse werden abgespeichert, gerade benötigte und die ursprünglichen Daten sind jederzeit abrufbar (z.B. durch die Funktion „Rückgängigmachen“).</i> • <i>Für erfahrene Nutzer sind Short-Cuts (z.B. Tastaturkürzel) möglich.</i>

Beispiel: Adressangaben sollten in genau der Reihenfolge in eine Eingabemaske eingegeben werden, in der man diese Informationen auf ein Briefcouvert schreiben würde. **Hier gut gelungen?**

Adressenverzeichnis Encyclia (45/45)

22/10/1999 **Adressenverzeichnis**

Verzeichnisnr. **Merkmal**

 **Kontakt**

Intro **Anmerkungen**

2. Name

Adresse Experte für Fernand Léger

... / ...

Stadt **Telekommunikation**

PLZ Anschluß Nummer

Land Tel. privat

Einladung Fax

Beziehung zur Galerie **Korrespondenz**

Datum	Thema der Korrespondenz
26 Juni 2000	brief



Präferenzen

Kubismus



2.2.2 Selbstbeschreibungsfähigkeit (Selbsterklärungsfähigkeit)

- **Navigation: und Orientierung:** Der Nutzer muss während der Arbeit mit einer Anwendung immer wissen, wo er sich befindet, wie er dorthin gekommen ist und welche Möglichkeiten er hat, von dort weg zu kommen. Das sollte i.d.R. ohne Hilfefunktionen, Vorwissen, Handbücher usw. vom System unterstützt werden.
- **Hilfefunktionen:** Ist das nicht erreichbar, müssen verständliche, kontextabhängige Hilfen angeboten werden.
- **Sichtbarkeit:** Selbstbeschreibungsfähigkeit ist häufig auch an gute Sichtbarkeit geknüpft.

Bezeichnung des Kriteriums	Selbstbeschreibungsfähigkeit
Definition nach ISO- Norm	<i>Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer jederzeit erkennbar ist, in welchem Dialog er sich gerade befindet, an welcher Stelle des Dialogs er angekommen ist, welche Handlungsoptionen gegeben sind und wie sie ausgeführt werden können.</i>
Beispiele für mögliche Anwendungen	<ul style="list-style-type: none"> • <i>Anklickbare Objekte (Knöpfe, Links) werden so beschriftet, dass erkennbar ist, welche Funktion sich dahinter verbirgt.</i> • <i>Der Nutzer wird genau über den Systemstatus informiert.</i> • <i>Anklickbare Elemente sollten Aufforderungscharakter haben und als anklickbar zu erkennen sein (z.B. Links durch blaue Farbe und unterstrichene Schriftzeichen).</i> • <i>Ganz wichtig: Es werden differenzierte, kontextabhängige Rückmeldungen über das Ergebnis ausgeführter Handlungen gegeben.</i> • <i>Antizipierbarkeit: Dem Benutzer muss jederzeit klar sein, was als nächstes zu tun ist, um ein Handlungsziel zu erreichen.</i>

Beispiel: Icons sollten so gestaltet sein, dass ihre Bedeutung ohne Beschriftung und ohne Notwendigkeit, nachschlagen zu müssen, eindeutig erkennbar ist.



Motorola
RAZR V3i



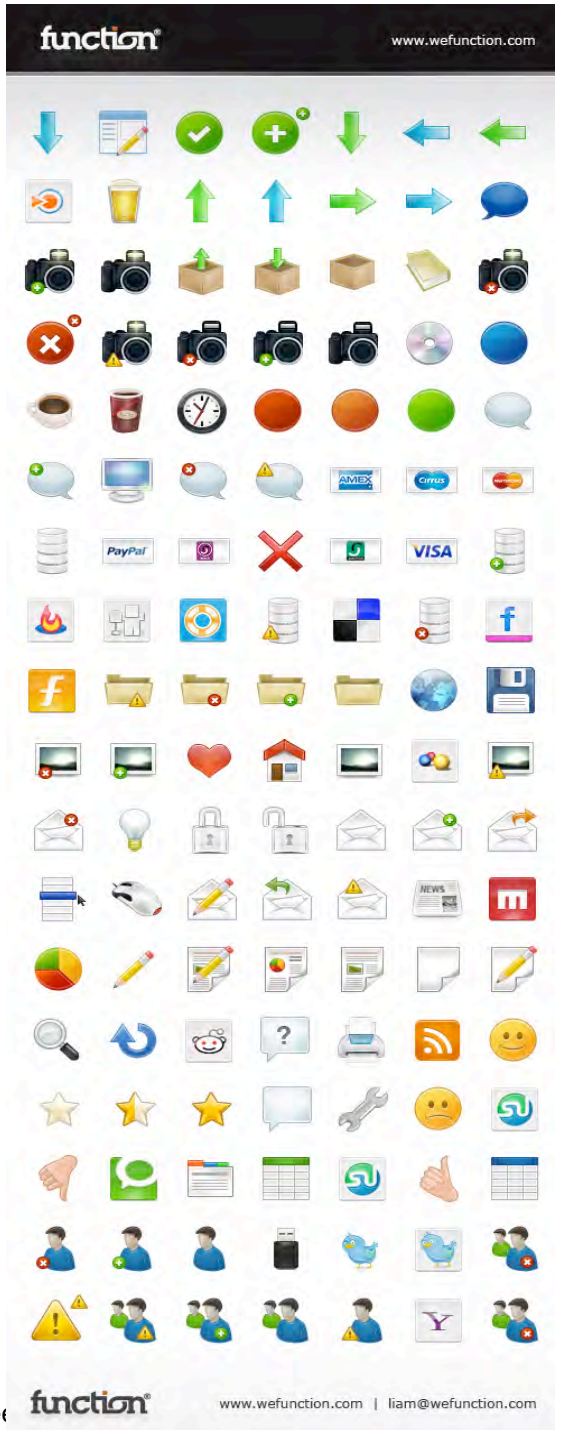
Samsung
SGH-D600



Sony-Ericson
V600i



Nokia
6230i



2.2.3 Steuerbarkeit

- **Kontrolle:** Der Nutzer muss das System möglichst weitgehend kontrollieren können und nicht das Gefühl haben, in seiner Arbeit kontrolliert und in seiner Entscheidungsfähigkeit eingeschränkt zu werden. Geschwindigkeit und Arbeitsreihenfolge muss den Wünschen des Nutzers anpassbar sein. Kontrollverlust durch Systemfehler darf es nicht geben.
- **Individualisierbarkeit:** Um unterschiedlichen Merkmalen von Nutzern gerecht werden zu können, sollte es möglich sein, Ein- und Ausgabemodalitäten der Software bedarfs- und kontextabhängig anzupassen.
- **System- und benutzergesteuerte Dialoge:** Bei komplexen Systemen ist ein gewisser Grad an Nutzerführung erforderlich, insbesondere bei sicherheitsrelevanten Systemen. Die Bearbeitung einer Aufgabe sollte in ihrem Tempo regulierbar sein und keinen Stress erzeugen.

Bezeichnung des Kriteriums	Steuerbarkeit
Definition nach ISO- Norm	<i>Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten und zu unterbrechen sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Arbeitsziel erreicht ist.</i>
Beispiele für mögliche Anwendungen	<ul style="list-style-type: none"> • <i>Keine zeitgesteuerten automatischen Aktionen einbinden.</i> • <i>Flexibilität: Alle im Prinzip möglichen Wege müssen beschreitbar sein.</i> • <i>Akustische Ausgaben können reguliert werden.</i> • <i>Der Nutzer kann die Darstellung und Sortierung von Daten auswählen.</i> • <i>Änderungen können rückgängig gemacht werden.</i> • <i>Der Nutzer hat mehrere Möglichkeiten der Navigation.</i> • <i>Der Nutzer kann den Systemablauf jederzeit beeinflussen.</i> • <i>Fehlertoleranz: Das System muss so stabil sein, dass kein Verlust der Steuerbarkeit auftreten kann. Bei fehlerhafter Bedienung muss das System auf die Fehler hinweisen und richtige Handlungsalternativen vorschlagen.</i>

Beispiel: Einschränkungen der Steuerbarkeit sind auf Internetseiten häufig zu beobachten →

- Werbefenster schieben sich über die gerade betrachtete Seite.
 - Werbebanner ziehen automatisch Aufmerksamkeit auf sich.
 - Fehlen von Rückmeldungen z.B. über den Beginn und Zeitbedarf eines Downloads.
 - Wichtige Stellen (z.B. Login) sind nicht leicht erkennbar.
 - Beim Navigieren verliert man den Überblick, weil man nicht mehr weiß, wie man zur übergeordneten Auswahlebene gelangen kann (außer durch aufwändiges, seitenweises Zurückblättern).
- usw. usw. usw.

2.2.4 Erwartungskonformität

- **Erwartungen:** Häufiger Umgang mit Computern lässt Erwartungen entstehen, die auf gemachten Erfahrungen beruhen. Ein wichtiges Kriterium ist, wie gut die visuelle Erscheinungsform und die Aktionen eines Systems diese Erwartungen erfüllen.
- **Übereinstimmung mit mentalem Modell:** Erwartungskonforme Systeme erleichtern dem Nutzer die Anwendung seines mentalen Modells auf sein Handeln mit dem System.
- **innere Konsistenz:** Ein System muss in sich selbst konsistent sein.
- **äußere Konsistenz:** Ein System muss vorhandene Konventionen befolgen (z.B. Styleguides).

- **metaphorische Konsistenz:** Besteht in der Konsistenz der virtuellen Handlungen, die mit dem Softwaresystem ausgeführt werden können, mit den Handlungen, wie sie in der realen Arbeitswelt ausgeführt werden.

- **Antwortzeiten:** Antwortzeiten des Systems, die mehr als 50% von der erwarteten Antwortzeit/Rückmeldezeit abweichen, erzeugen beim Nutzer Unsicherheit darüber, ob das System richtig funktioniert. Sind lange Antwortzeiten vorhersehbar, muss das System den Nutzer darüber rechtzeitig informieren.

Bezeichnung des Kriteriums	Erwartungskonformität
Definition nach ISO- Norm	<i>Ein Dialog ist erwartungskonform, wenn er den aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen sowie allgemein anerkannten Konventionen entspricht.</i>
Beispiele für mögliche Anwendungen	<ul style="list-style-type: none"> • <i>Bei der Gestaltung von Funktionen sollte man sich an vorhandenen Werkzeugen, Lösungen und Konventionen aus dem konventionellen Arbeitskontext orientieren.</i> • <i>Styleguides der Betriebssystemhersteller beachten.</i> • <i>Kenntnisse des Benutzers aus älteren Programmversionen sollten Verwendung finden.</i> • <i>Die graphische Gestaltung von Navigationsinstrumenten (z.B. Menüs) sollte immer gleich sein.</i> • <i>Häufig verwendete Funktionen sollten an immer gleichen Orten vorkommen (z.B. Felder zum Schließen eines Fensters).</i> • <i>Handlungen, wie z.B. Drag&Drop, sollten ausnahmslos in jedem Kontext anwendbar und nicht an bestimmte Voraussetzungen geknüpft sein.</i>

Beispiele:

- Nicht erwartungskonform ist es, wenn man eine Textstelle neu formatiert hat, weil man z.B. eine Textpassage eingefügt hat, und der Cursor bleibt nicht an der von ihm zuletzt eingenommenen Position stehen, sondern wird zurück verschoben (vgl. WORD for Mac)
- Erwartungskonform ist es, wenn Google die Eingabe nach Suchbegriffen trotz vieler sonstiger Änderungen stets gleich lässt.
- Erwartungskonform ist es, wenn auf Internetseiten das Verzeichnis der untergeordneten Seiten in Form von Links auf der linken Seite angeordnet ist.
- Auf einer Webseite sind blaue, unterstrichene Textabschnitt immer Links (wird häufig verletzt).

2.2.5 Fehlertoleranz

- **Handlungsfehler:** Das System sollte den Nutzer stets vor Fehlern bewahren, die er bei Verwendung eines Programms machen könnte, damit er sein Arbeitziel möglichst ohne Zeit- und Datenverlust erreichen kann.

Dies kann durch

- differenzierte,
- auch für Laien **verständliche**,
- kontextabhängige Fehlermeldungen sowie durch
- Beschreibung der möglichen Handlungsalternativen und evtl. eintretenden Fehlerkonsequenzen

geschehen (s. Beispiel u.).

Netzlaufwerksanbindung fehlgeschlagen



Fehlernr: -2147024829 Share: \\nwsrv02-w.lrz.lrz-muenchen.de\SWs\ existiert nicht
Fehlernr: -2147024891 Fehlertyp leider unbekannt

Evtl. ist Ihre PC-Kennung für Windows2000 noch nicht aktiviert
(Fehlernr: 2147023570)

Dies können Sie über folgende Webseite durchführen:

<https://tools.lrz-muenchen.de/pcentsperren>

Wählen Sie dort den Menüpunkt: PC-Kennung entsperren.

Bei anderen Fehlernummern setzen Sie sich mit der LRZ-Beratung,
bzw. mit den Systemadministratoren der PC-Gruppe in Verbindung.

Soll nun auf die o.a. Webseite gewechselt werden?

Ja

Nein

- **Systemfehler:** Das System darf nie durch einen Fehler des Nutzers in einen irreversiblen Zustand geraten, aus dem nur noch ein Neustart unter Verlust des bisherigen Arbeitsergebnisses Rettung bedeutet. Ist ein Fehler präventiv nicht vermeidbar, muss der Nutzer im Fehlerfall dabei unterstützt werden, den Fehler mit geringem Aufwand zu beheben. Über automatische Fehlerkorrekturen muss der Nutzer in Kenntnis gesetzt werden.

- **Fehlermeldungen:** Ist ein Fehler vom System nicht korrigierbar, muss eine Fehlermeldung ausgegeben werden. Sie sollte automatisch die Aufmerksamkeit des Nutzers auf sich ziehen (z.B. durch Verwendung besonderer Farben (z.B. rotes Warndreieck), Blinken ist besonders effektiv → vgl. Aufmerksamkeit). Die Ausgabe von weniger wichtigen Warnungen muss der Nutzer abstellen können.

- **Bestätigung einer Funktion:** Bei Funktionen, die eine nicht wieder herstellbare Veränderung bringen würde, muss das System nachfragen, ob die Funktion wirklich ausgeführt werden soll; evtl. müssen die Konsequenzen der Funktionsausführung explizit benannt werden (z.B. Datenlöschung; Neustart des Systems u.ä.)

- **Meldung von Fehlern an Hersteller:** Effiziente Methode zum Ausmerzen von Programmierfehlern; wirkt beim Benutzer dem Gefühl eines Kontrollverlust und dem daraufhin entstehenden Ärger entgegen (bei Einführung von Mac OS X waren die anfangs häufig auftretenden Systemfehler so nach ca. 2 Monaten behoben).



Das Programm „Cubase“ wurde unerwartet beendet.

Mac OS X und andere Programme sind nicht betroffen.

Klicken Sie auf „Neu starten“, um das Programm erneut zu starten.
Klicken Sie auf „Bericht“, um weitere Details anzeigen zu lassen
oder einen Bericht an Apple zu senden.

Ignorieren

Bericht ...

Neu starten

Bezeichnung des Kriteriums	Fehlertoleranz
Definition nach ISO- Norm	<i>Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.</i>
Beispiele für mögliche Anwendungen	<ul style="list-style-type: none"> • <i>Eingaben werden automatisch überprüft und fehlerhafte Eingaben abgefangen.</i> • <i>Der Nutzer wird auf Fehler, wenn möglich mit einer kontextsensitiven Hilfestellung, hingewiesen.</i> • <i>Bei bekannten aber nicht sofort behebbaren Fehlern des Systems sollte der Fehler erklärt werden und die notwendigen Maßnahmen zur Sicherung des bisherigen Arbeitsergebnisses angegeben werden.</i> • <i>Systemabstürze sollten auf der Betriebssystemebene abgefangen werden und nicht einen Neustart erforderlich machen, bei dem das bisherige Arbeitsergebnis i.d.R. verloren geht.</i> • <i>Wiederholt auftretende neuartige Fehler sollten dem Hersteller mit geringem Aufwand gemeldet werden können.</i>

Beispiele:

- In ein Eingabefeld eines Formulars, das nur für Zahlen vorgesehen ist (um z.B. das Alter anzugeben), Buchstaben einzugeben, führt zu einer Fehlermeldung mit Erklärung des Grundes.
- Automatische Rechtschreibkorrektur mit Korrigierbarkeit des gemachten Vorschlags, z.B. HEute → Heute (WORD)
- Solche Fehlermeldungen sollten nicht vorkommen:



2.2.6 Individualisierbarkeit

- **Wünsche und Vorstellungen der Nutzer:** Nutzer von Systemen haben individuelle Vorstellungen und Wünsche bezüglich der Gestaltung und Handhabung eines Systems. Die Möglichkeit der Individualisierung ersetzt aber keine gute Standardoberfläche. Zu vermeiden ist, dass Änderungen zu spürbaren Beeinträchtigungen führen.
- **Flexibilität:** Anwendungen sollten so flexibel gestaltet werden, dass Änderungen möglich sind, ohne dafür Entwickler zu benötigen (z.B. Reihenfolge in der Bearbeitung einer aus mehreren Teilaufgaben bestehenden Aufgabe; Farbwahl, Voreinstellungen, Formate u.ä.).

Bezeichnung des Kriteriums	Individualisierbarkeit
Definition nach ISO- Norm	<i>Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe, an die individuellen Vorlieben des Benutzers und an das jeweilige Kompetenzniveau des Benutzers zulässt.</i>
Beispiele für mögliche Anwendungen	<ul style="list-style-type: none"> • <i>Funktionen und Fenster sind vom Nutzer zu- und abschaltbar. Außerdem kann der Nutzer die Anordnung verändern.</i> • <i>Die Größe von Werkzeugpiktogrammen und Schaltflächen sowie von Feldern zum Ausfüllen müssen variabel gestaltet werden, um auf Sehschwächen insbesondere älterer Benutzer Rücksicht zu nehmen.</i> • <i>Es können Werkzeugleisten generiert und individuell arrangiert werden.</i> • <i>Die Benennung von Komponenten ist veränderbar.</i> • <i>Die Nutzung von Nutzerprofilen wird unterstützt.</i> • <i>Hat der Nutzer Daten hinterlegt, die in einem Formular abgefragt werden, werden diese automatisch eingesetzt.</i>

Beispiel: Autokorrektur



- Alles
- Bilder
- Videos
- News
- Mehr

Duisburg
Standort ändern

Das Web
Seiten auf Deutsch
Seiten aus Deutschland

Alle
Neueste
Letzte 3 Tage

Alle Ergebnisse
Websites mit Bildern

Mehr Optionen

Hamurg Suche

hamburg
hamburger abendblatt
hamburger sv
hamburger dom
hamurg

Weitere Informationen

Ungefähr 55.400.000 Ergebnisse (0,13 Sekunden) Erweiterte Suche

Ergebnisse für [hamburg](#). Stattdessen suchen nach: [Hamurg](#)

[Hamburg](#) [maps.google.de](#)



[hamburg.de](#) - Offizielles Stadtportal für die Hansestadt Hamburg ☆ 🔍

Alle Informationen über **Hamburg** übersichtlich und schnell gefunden. **Hamburg** von A bis Z für **Hamburger** Einwohner, Touristen und Unternehmen.
Hotels & Tourismus - Hamburg Mail - Bürger- Service
[www.hamburg.de/](#) - Im Cache - Ähnliche Seiten

[HAMBURG Tourismus GmbH](#) – Tickets, Hotels, Pauschalangebote ☆ 🔍

Offizielle **Hamburg**-Tourismus-Seite: Hotels, Pauschalangebote und Tickets online buchen. Ausführliche Informationen und viele Tipps zu allen ...
[www.hamburg-tourism.de/](#) - Im Cache - Ähnliche Seiten

Google Instant ist aktiviert. ▼

SafeSearch aus ▼

Anzeigen

[Preiswerte Hotels Hamburg](#)

Sparen Sie bei diesen Hotels in **Hamburg**. Alles Sterne, günstig!
[www.booking.com](#)

[225 Hotels in Hamburg](#)

Hotels ab 37,-€ pro Zimmer/Nacht mit Karte, Fotos, Hotelbewertungen!
[www.hotel.de/Hamburg](#)

[Hotel Hamburg Schnäppchen](#)

4 für 3, 3 für 2, Last Minute etc. Jetzt **Hamburg**-Sonderangebote buchen
[www.hamburg-tourism.de/Schnaepchen](#)

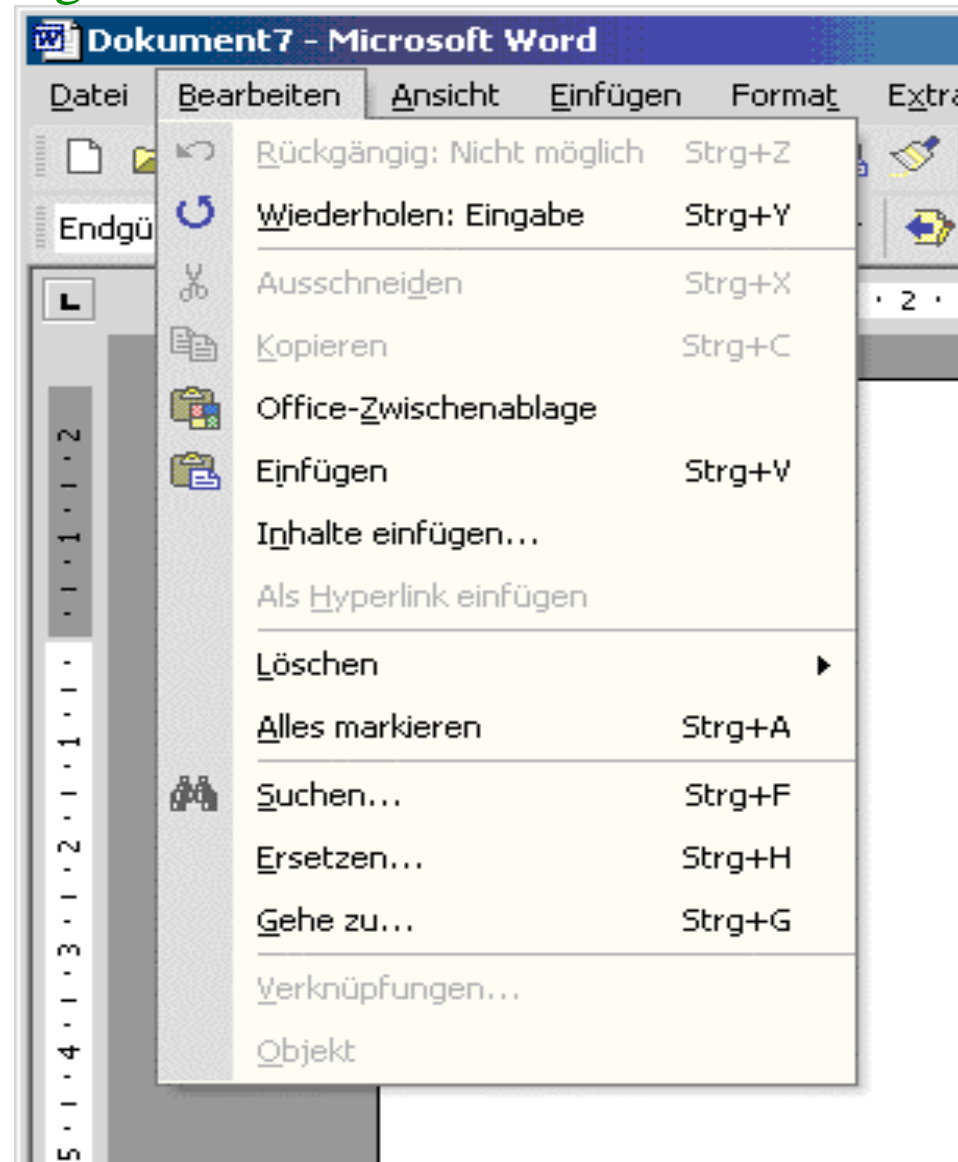
[Schalten Sie hier Ihre Anzeige >](#)

2.2.7 Lernförderlichkeit

- **Erlernen der Bedienung:** Eine Anwendung soll den Nutzer beim Erlernen der Bedienung unterstützen und anleiten.
- **Ähnlichkeit zur realen Arbeit:** Programmkonzepte werden i.d.R. möglichst nah am realen Anwendungsgebiet entwickelt (→ Metaphern). Das ermöglicht die Übertragung erworbener Kenntnisse aus dem Anwendungsgebiet durch eine nahezu isomorphe Systemstruktur, die das Erlernen des Computerprogramms erleichtert.
- **Hilfesysteme:** In vielen Fällen unterstützen Hilfesysteme oder Schulungen den Einstieg in die Nutzung eines neuen Systems (hier deckt sich die *Lernförderlichkeit* mit dem Kriterium der *Selbstbeschreibungsfähigkeit*).

Bezeichnung des Kriteriums	Lernförderlichkeit
Definition nach ISO- Norm	<i>Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.</i>
Beispiele für mögliche Anwendungen	<ul style="list-style-type: none"> • <i>Wichtige und am häufigsten benötigte Funktionen werden gut erkennbar angeordnet.</i> • <i>Funktionen werden anschaulich und in Übereinstimmung mit dem fachsprachlichen Gebrauch benannt.</i> • <i>Icons werden als symbolische Darstellung der Funktionen verwendet.</i> • <i>Wenn Funktionen auf mehreren Wegen benutzbar sind, wird auf die jeweils andere Möglichkeit hingewiesen.</i> • <i>Tutorials zur Nutzung der Anwendung werden durch das System selbst angeboten.</i> • <i>Ein gut durchdachter Katalog von FAQ („frequently asked questions“) löst häufig aufgetretene Probleme.</i> • <i>Schulungen sollten i.d.R. überflüssig sein, sind bei hochkomplexen Systemen, die viele Aufgabenbereiche abdecken müssen, manchmal aber nicht vermeidbar.</i>

Beispiele: Wenn für eine Funktion ein Short-Cut existiert, sollte dieses Short-Cut im Menü neben der Funktionsbezeichnung deutlich erkennbar angezeigt werden.



2.3 Expertenleitfaden zur Evaluation von Software: EVADIS II und seine Kriterien

Expertenleitfaden: Prüfverfahren für Experten im Rahmen einer Usability-Evaluation.

- **Ziel:** Es geht nicht um die Frage, ob das System geeignet ist, bestimmte Aufgaben zu erfüllen, sondern alleine um die **Frage, ob alle ergonomischen Gestaltungskriterien erfüllt wurden**. Ihre Erfüllung wird im Sinne einer Selbstkontrolle der Entwickler zwecks Entdeckens von Mängeln der Software überprüft.

- **Voraussetzung:** Der Prüfer muss die Software genauestens kennen und Erfahrungen in der Evaluation von Software haben; er muss aber nicht selbst Systementwickler sein und auch nicht wissenschaftlich ausgebildeter Ergonomiefachmann.
- **Verfahren:** Das Prüfverfahren besteht in einer **Liste von Fragen hinsichtlich der Gestaltung von Benutzerschnittstellen**, die entweder vom Experten allein zu beantworten sind oder von potenziellen Benutzern. Die **Erfüllung der einzelnen Bewertungskriterien bei der Bewältigung festgelegter Prüfaufgaben wird benotet** und in einem standardisierten Prüfbericht dokumentiert; so ist auch eine quantitative Auswertung und ein Vergleich verschiedener Versionen im Laufe der Systementwicklung möglich.
- **Vorteil:** Schnelle Durchführbarkeit, daher geringe Kosten im Vergleich zur Anwendung von Beobachtungsmethoden mit potenziellen Nutzern.

- **Beispiel: EVADIS II** (Oppermann et al., 1992)

Die geprüften Kriterien bestehen aus den oben beschriebenen ISO-4291-Normen sowie den im Folgenden beschriebenen Kriterien:

Verfügbarkeit: Der Benutzer wird nicht durch Störungen oder zu lange Antwortzeiten behindert.

Nützlichkeit: Nützlichkeit umfasst Aspekte der *„Aufgabengemessenheit*, die sich auf eine angemessene Funktionalität der Software für die Aufgabenerfüllung beziehen.

Komfort: Komfort umfasst Aspekte der *„Aufgabenangemessenheit*, die sich auf eine eventuell unerwünschte Mehrbelastung durch Eigenschaften des Dialogsystems beziehen.

Übersichtlichkeit: Übersichtlichkeit bezieht sich auf Darstellung und Anordnung der Informationen auf dem Bildschirm (→ Sichtbarkeit)

Kooperations- und Kommunikationsförderlichkeit: Dieses Kriterium bezieht sich auf die Frage, inwieweit die Software den Arbeitenden eine gemeinsame, kooperative Aufgabenerfüllung ermöglicht, wobei der unmittelbaren, persönlichen Kommunikation ein hoher Stellenwert zukommt.

Datenschutz/Datensicherheit: Dieses Kriterium betrifft den Schutz der eingegebenen Daten vor unbefugtem Zugriff, andererseits auch den Schutz der Arbeitenden vor elektronischer Leistungs- und Verhaltenskontrolle.

Beispiel für eine Prüffrage aus EVADIS II:

Kriterium 4: Übersichtlichkeit

Prüffrage: Wie erfolgt die Farbgestaltung am Bildschirm?

Antwortvorgaben: zutreffende Alternativen ankreuzen!

- Die Anzahl der verwendeten Farben pro Bildschirm beträgt maximal drei bis vier.
- Die Bedeutung der Farben entspricht jener im Alltag oder Beruf.
- Die Farben werden einheitlich im gesamten Anwendungssystem verwendet.
- Auf farbfehlsichtige Benutzer wird Rücksicht genommen (z.B. durch doppelte Codierung).
- Es werden nur gut verträgliche Farbkombinationen eingesetzt.

Kommentar:

Prinzipiell sollte vom Farbeinsatz nur sparsam Gebrauch gemacht werden, da eine gute Strukturierung der Information dem Farbeinsatz vorzuziehen ist. Das heißt, Farbe sollte primär als zusätzliches Codiermittel eingesetzt werden.

Bewertung:

Note: Kriteriumsgewichtung:

Begründung:

Notiz:

Kritische Punkte:

- Herkunft der Prüffragen: Basieren auf der Literatur, eigenen Erfahrungen der Autoren bis ca. 1991; danach nicht aktualisiert für moderne Anwendungsfelder (z.B. WWW, Virtual Reality).
- Vergleichbarkeit: Da die Benotungen subjektiven Urteilen entsprechen, sind Vergleiche nur dann einigermaßen verlässlich, wenn die zu vergleichenden Systeme oder Versionen vom selben Beurteiler evaluiert wurden.
- Wissenschaftliche Gütekriterien des Verfahrens: Reliabilität und Validität wurden nicht empirisch ermittelt, sind aber wahrscheinlich geringer als bei direkter Verhaltensbeobachtung von Benutzern.
- Expertise der Beurteiler: Bei komplexen Aufgaben mit einem komplexen System kann auch Experten mancher Designfehler eines Systems entgehen; bei einfachen Aufgaben und einfachen Systemen hat sich die Methode bewährt.

2.4 Usability-Evaluation von Software

2.4.1 Heuristische Evaluation

- **Heuristische Evaluation:**

- Explorative, qualitative Methode der Usability-Evaluation;
- bedient sich einer Menge von Usability-Kriterien (s.o.) als *Heuristiken* zur Suche nach Verstößen;

Heuristisches Vorgehen: Sehr globale Strategie des Problemlösens durch systematisches Ausprobieren auf der Basis sehr abstrakter Regeln (hier: Usability-Kriterien).

- wird von Usability-Experten durchgeführt als Vertreter der späteren Benutzer, indem sie die Software zur Bewältigung von zahlreichen Aufgaben der verschiedenen Anwendungsbereiche einsetzen und die dabei auftretenden Probleme (Abweichungen von Usability-Kriterien) feststellen.

- Unterschiede zu EVADIS:
 - > keine Fragenbogenmethode,
 - > keine einheitliche Dokumentation,
 - > keine quantitativen Evaluationsergebnisse
 - > dadurch keine gute Basis für Vergleiche zwischen verschiedenen Programmversionen.

Heuristiken nach Nielsen (1994):

- Die Heuristiken entsprechen inhaltlich weitgehend den bereits oben dargestellten Usability-Kriterien;
- das Verstehen und Beachten der heuristischen Regeln, obwohl sie weniger abstrakt formuliert sind als die Usability-Kriterien, setzt die Kenntnis der allgemeiner formulierten Usability-Kriterien und ihrer psychologischen Hintergründe voraus;
- die sprachliche Form besteht (fast wie in einem Katechismus) in Anweisungen („du sollst“, „du sollst nicht“) an den Systementwickler, was mit der Software zu erreichen bzw. zu vermeiden ist;
- das macht sie weniger abstrakt als die üblichen Usability-Kriterien und deshalb vielleicht leichter anwendbar;
- die Heuristiken sprechen häufig gemachte und ganz typische Fehler von Softwareentwicklern an; das fokussiert die Evaluatoren auf ganz bestimmte, häufig auftretende Probleme.

1. Sichtbarkeit des Systemstatus: Das System sollte den Nutzer immer informieren, was gerade vorgeht, durch geeignetes Feedback innerhalb angemessener Zeit.

2. Übereinstimmung zwischen System und realer Welt: Das System soll die Sprache des Nutzers sprechen, mit Wörtern, Formulierungen und Konzepten, die dem Nutzer vertraut sind, nicht mit systemorientierten Begriffen (Fachsprache der Informatiker). Befolge Konventionen aus der realen Arbeitswelt, so dass Informationen in natürlicher und logischer Reihenfolge erscheinen.

3. Benutzerkontrolle und Freiheit: Nutzer wählen Softwarefunktionen oft versehentlich. Sie brauchen einen klar markierten Notausgang, um einen unabsichtlich erreichten Zustand verlassen zu können, ohne durch einen ausgedehnten Dialog hindurch zu müssen. Unterstütze „Undo“ und „Redo“.

4. Konsistenz und Standards: Die Nutzer sollten nicht überlegen müssen, ob verschiedene Begriffe, Situationen oder Aktionen dasselbe bedeuten. Befolge Plattform-Konventionen (Konventionen, Styleguides der verschiedenen Betriebssystem-Hersteller).

5. Fehler vermeiden: Besser noch als gute Fehlermeldungen ist ein sorgfältiges Design, das verhindert, dass überhaupt erst Probleme auftreten.

6. Erkennen und Erinnern: Mache Objekte, Aktionen und Optionen sichtbar. Der Nutzer sollte nicht gezwungen werden, sich Informationen aus einem Teil eines Dialogs für einen anderen Teil zu merken. Anleitungen zur Benutzung des Systems sollten sichtbar oder leicht auffindbar sein, wann immer angemessen.

7. Flexibilität und effizienten Nutzung: Beschleunigungsmöglichkeiten, die der ungeübte Nutzer nicht sieht, können für erfahrene Nutzer die Interaktionsgeschwindigkeit erhöhen, so dass das System sowohl für Neulinge als auch für Experten geeignet ist. Erlaube Nutzern, häufige Aktionen auf ihre Bedürfnisse zuzuschneiden.

8. Ästhetisches und minimalistisches Design: Dialoge sollten keine Information enthalten, die irrelevant ist oder selten benötigt wird. Jedes Extra an Information in einem Dialog konkurriert mit relevanten Informationen und vermindert deren relative Sichtbarkeit.

9. Unterstützung beim Erkennen, Verstehen und Bearbeiten von Fehlern: Fehlermeldungen sollten in klarer Sprache (keine Kodierungen) gegeben werden. Sie sollen das Problem genau beschreiben und konstruktiv eine Lösung vorschlagen.

10. Hilfe und Dokumentation: Obwohl es besser ist, wenn das System ohne Dokumentation benutzt werden kann, kann es nötig sein, Hilfe und Information zu geben. Jede solche Information sollte leicht zu durchsuchen sein, die Aufgabe des Nutzers in den Mittelpunkt stellen und konkrete Schritte zur Ausführung nennen. Die Dokumentation sollte nicht zu umfangreich sein.

Zusätzliche Heuristiken bei Sarodnick & Brau (2006):

11. Joy of use (\approx Benutzerzufriedenheit): Arbeitsabläufe und graphische Gestaltung des Systems sollten bei notwendiger Konsistenz Monotonie vermeiden und zeitgemäß wirken. Metaphern sollten adäquat und auf den Nutzungskontext abgestimmt verwendet werden.

12. Interkulturelle Aspekte: Das System sollte auf einen definierten Nutzerkreis und dessen funktionale, organisatorische und nationale Kultur abgestimmt sein.

2.4.2 Cognitive Walkthrough

Eines der Hauptprobleme aus Sicht der Nutzer bei der Entwicklung von Software ist die Frage ihrer *Erlernbarkeit*.

Cognitive Walkthrough: Evaluationsmethode speziell hierfür, die sehr frühzeitig im Rahmen der Systementwicklung und -evaluation einsetzbar ist (Wharton et al., 1994).

- Einer Gruppe von potenziellen Nutzern oder Experten wird ein Vorschlag zur Gestaltung einer Software vorgestellt.
- Zur Verfügung steht noch keine echte Software sondern nur schriftliche Beschreibungen und graphische Entwürfe von Bildschirmhalten (fiktive Screenshots).

Ziel: Sicherstellung der Gebrauchstauglichkeit für unerfahrene Nutzer unter dem Gesichtspunkt eines **schnellen Wissenserwerbs** über die Funktionen des Systems und ihrer Anwendung im Kontext bestimmter Aufgaben.

Prinzipielles Vorgehen:

1. Für jede Aufgabe, die mit dem System bearbeitet werden soll, wird als Ideallösung eine korrekte Handlungsabfolge bestimmt.
2. Die Handlungsabfolgen werden von einer Gruppe von Usability-Experten anhand von vier Fragen analysiert, um zu ermitteln, ob spätere Nutzer exakt diese Handlungsabfolge mit dem System umsetzen können.
3. Zeigt sich, dass die Handlungsabfolge nicht oder nur schwer einzuhalten ist, werden Alternativlösungen erarbeitet.
4. Systementwickler modifizieren den Systementwurf entsprechend den Ergebnissen.

Durchführung eines Cognitive Walkthrough:

Zwei Phasen: 1. Vorbereitungsphase; 2. Analysephase

1. Vorbereitungsphase:

1.1 Festlegung der Annahmen über die potenziellen Nutzer:

Erstellung eines prototypischen Nutzerprofils (z.B. Vorkenntnisse, Bildung, Alter usw.)

1.2 Festlegung der zu analysierenden Aufgaben:

- Auswahl häufig auszuführender, zentraler Aufgaben
- detaillierte Beschreibung der Aufgaben

1.3 Festlegung der idealen Handlungsabfolge zur korrekten Aufgabenbewältigung:

- Wie verstehen Nutzer die Aufgaben: mentales Modell wird zu erstellen versucht
- Definition und Beschreibung aller notwendigen Handlungen zur Bewältigung jeder einzelnen Aufgabe, z.B. „gib Login-Namen ein“, „gibt Passwort ein“, „drücke Knopf ‚Anmelden‘“ usw.

1.4 Definition der Schnittstellen:

- Beschreibung dessen, was der Nutzer von der Schnittstelle sieht für jeden einzelnen Handlungsschritt, z.B. Symbole, Bezeichnungen, Bedienelemente, Farben, Fenster, Reaktionen des Systems usw.
- Herstellung von Mock-Up-Screens mit diesen Schnittstellenmerkmalen

2. Analysephase:

- Durcharbeiten aller Handlungsschritte bei jeder Aufgabe in chronologischer Reihenfolge durch Experten.
- Evaluatoren beantworten dabei **vier Fragen** zu jeder einzelnen Handlung:

1. Werden die Nutzer versuchen, den gewünschten Effekt zu erzielen?

Nutzer müssen erkennen können, welche Handlungen zu einem gegebenen Zeitpunkt/Systemzustand notwendig sind, um einen gewünschten Effekt (Handlungsziel) zu erreichen.

Beispiel: Nutzer will zuerst Ware aus Onlinekatalog aussuchen und dann erst Adressdaten eingeben, System verlangt aber zuerst Adressdaten; erkennt Nutzer dies nicht rechtzeitig, bricht er möglicherweise den Kauf ab.

Ein Erfolg ist möglich, wenn

- das System den Nutzer zur korrekten Handlung auffordert (bzw. mögliche Handlungsalternativen explizit zur Auswahl anbietet);
- die Handlung in der virtuellen Welt bereits Teil der ursprünglichen Aufgabe in der realen Welt ist;
- der Nutzer einschlägige Erfahrungen mit dem System oder ähnlichen, nach Normen konsistent gestalteten Systemen hat.

2. Werden die Nutzer erkennen, dass die korrekte Handlung ausgeführt werden kann?

Es reicht nicht, wenn ein Nutzer zwar erkennt, dass er zum Bearbeiten seiner Aufgabe eine bestimmte Handlung ausführen muss, er muss darüber hinaus auch erkennen, dass er sie als nächste ausführen kann und wie er sie ausführen kann.

Beispiel: Wenn der Nutzer eine Lieferanschrift eingeben möchte, erwartet er ein Eingabefeld mit entsprechender Beschriftung, das ihm die Handlungsmöglichkeit signalisiert.

3. Werden die Nutzer erkennen, dass die korrekte Handlung zum gewünschten Effekt führen wird?

Hat der Nutzer das nächste Handlungsziel erkannt und weiß er, dass er die dazu notwendigen Handlungsschritte ausführen kann, kann es immer noch Probleme bei der Umsetzung von Intention und Handlung kommen, wenn er nicht eindeutig Schritt für Schritt zu den notwendigen Teilhandlungen geführt wird.

Beispiel: Der Nutzer hat zwar erkannt, dass er Kundendaten eingeben muss, aber die dafür angebotenen Eingabefelder sind nicht oder nicht eindeutig beschriftet oder erzwingen (unnötigerweise) eine bestimmte Reihenfolge, an die sich der Nutzer zunächst nicht hält.

4. Werden die Nutzer den Fortschritt erkennen, wenn sie die korrekte Handlung ausgeführt haben?

Der Nutzer muss nach Ausführung der Handlung sofort und eindeutig erkennen können, dass der gewünschte Effekt erzielt wurde (Rückmeldungen!), um Vertrauen zu dem System und seine damit ausgeführten Handlungen zu gewinnen.

Beispiel: Der Nutzer hat zwar die Kundendaten korrekt eingegeben, sieht sie auch in den Eingabezeilen, erhält aber keine Rückmeldung darüber, dass das System die Daten gespeichert hat. Nach Ausfüllen muss z.B. ein Button zum Abschicken der Daten gedrückt werden oder System speichert automatisch und gibt explizite Rückmeldung.

Wharton et al. (1994) haben festgestellt, dass alle **Misserfolge** bei der Benutzung eines Softwaresystems auf **wenigstens eine negative Antwort** zu einer der vier Fragen zurückzuführen sind →

Lösungsansätze zur Überarbeitung des Systems

1. Der Nutzer versucht nicht, den richtigen Effekt zu erzielen.

- Entfernung der Handlung aus der Handlungssequenz: wenn möglich Automatisierung
- Hinweistext verdeutlicht die jeweils nächste notwendige Handlung.

2. Der Nutzer erkennt nicht, dass die korrekte Handlung ausgeführt werden kann.

Besser geeignete Bedienelemente einsetzen: z.B. statt Eingabefelder Auswahlmenüs.

3. Der Nutzer erkennt nicht, dass die korrekte Handlung zum gewünschten Ziel führt.

Anordnung, Gestaltung und Benennung der Auswahl- oder Eingabemöglichkeiten verbessern.

4. Der Nutzer erkennt nicht den Fortschritt, der durch die korrekte Handlung entsteht.

Feedback: sofortiges, differenziert erläuterndes Feedback darüber, was vorgeht bzw. jetzt erreicht ist.

Kritik:

Pluspunkt: sehr früh einsetzbares Verfahren

Minuspunkt: keine Beobachtung mit echten Nutzern: auch Experten können irren!

2.4.3 Usability-Tests

Usability-Tests:

- empirisches Verfahren zur Überprüfung der Gebrauchstauglichkeit von Software;
- das noch in Entwicklung befindliche System wird von potenziellen Nutzern bei der Lösung realistischer Aufgaben eingesetzt;
- die Nutzer werden von Usability-Experten dabei beobachtet (bzw. werden Video-Aufzeichnungen ausgewertet);
- aus übereinstimmenden Beobachtungen von erfolgreichen Handlungen und Fehlern, aus begleitenden oder anschließend durchgeführten Interviews sowie aus Messungen von Leistungsparametern werden Schlüsse über Probleme und Verbesserungsmöglichkeiten gewonnen.
- Ursprünglich war ein experimenteller Ansatz mit systematischer Bedingungsvariation und „harten“ Messwerten vorherrschend; inzwischen dominieren die eher „weichen“ Methoden der reinen Beobachtung und Befragung.

2.4.3.1 Induktive und deduktive Usability-Tests

induktive Tests (formative Evaluation): Eine einzelne Vorabversion (Prototyp) wird evaluiert mit dem Ziel, Schwachstellen aufzudecken und so begründete Vorschläge für eine Verbesserung der Bedienbarkeit der Software zu machen.

- **formativ:** Das Verfahren dient der Änderung der Form, in der sich das System dem Benutzer darbietet.
- **induktiv:** Aus Einzelbeobachtungen wird auf das Verhalten des Gesamtsystems geschlossen.
- **Durchführung:** meistens am realen Arbeitsplatz, um die realen Arbeitsbedingungen realistisch mit einbeziehen zu können.

deduktive Tests (summative Evaluation): Alternative Systementwürfe gleicher Funktionalität werden vergleichend untersucht oder die erreichten Verbesserungen gegenüber älteren Versionen desselben Systems ermittelt.

- **summativ:** Aussagen in Form von Differenzen zwischen verschiedenen Systemen werden angestrebt.
- **deduktiv:** Aus den Merkmalen und Unterschieden der zu vergleichenden Softwareversionen werden Hypothesen abgeleitet und experimentell überprüft.
- **Durchführung:** i.d.R. im Labor zwecks Schaffung standardisierter Versuchsbedingungen → zentral für Sicherung der Vergleichbarkeit der Daten verschiedener Vpn und Versionen; Leistungsmaße stehen im Vordergrund (Bearbeitungszeiten, Fehlerhäufigkeiten, Blickbewegungen usw.).

2.4.3.2 Gesichtspunkte der Testpersonenauswahl

→ Die Ergebnisse aller empirischen Methoden hängen nicht nur von den eingesetzten Methoden ab sondern ganz zentral auch von der Zusammensetzung der untersuchten Stichprobe!

- **Zusammensetzung der Stichprobe:**

- **Repräsentativität:** Die untersuchte Stichprobe von Testpersonen sollte die gesamte Bandbreite der späteren Nutzer angemessen widerspiegeln. Daraus ergibt sich, dass der Stichprobenumfang nicht zu klein bemessen sei darf (s.u.).

- **wichtige Variablen:** Alter, Einkommen (sozioökonomischer Status), Geschlecht, Grad der Vorerfahrung mit Software (z.B. Informatikstudenten vs. Studierende anderer, nicht häufig differenzierte Computerkenntnisse erfordernde Studiengänge).

- Unwissentlichkeit:

→ Die Testpersonen dürfen das System oder Vorversionen davon noch nicht kennen (Kontrolle der Vorerfahrung), da sonst viele Probleme aufgrund der Vorerfahrungen erst gar nicht mehr auftreten.

→ Novizen und Experten decken unterschiedliche Fehler auf.

→ Die Testpersonen dürfen nicht wissen, worin genau das Ziel der Evaluation besteht: z.B. bei summativer Evaluation dürfen sie nicht wissen, welche die neue und welche die alte Version ist, weil sie sonst den Hypothesen der Untersucher entsprechend handeln könnten (z.B. aufgrund von Erwartungseffekten).

- **Anzahl benötigter Testpersonen:**

Weit verbreitete Auffassung (z.B. Nielsen, 1993): 5 – 6 Testpersonen reichen, um 80% der Fehler in einer Software zu entdecken (**aber reicht das wirklich?**).

Neuere Untersuchungen (Ausschnitt einer Tabelle von Faulkner, 2003) aber zeigen ein differenzierteres Bild:

Anzahl der Testpersonen	Minimal gefundene Probleme (%)	Durchschnittlich gefundene Probleme (%)
5	55	85,6
10	82	94,7
20	95	98,4
30	98	99,0
50	98	100

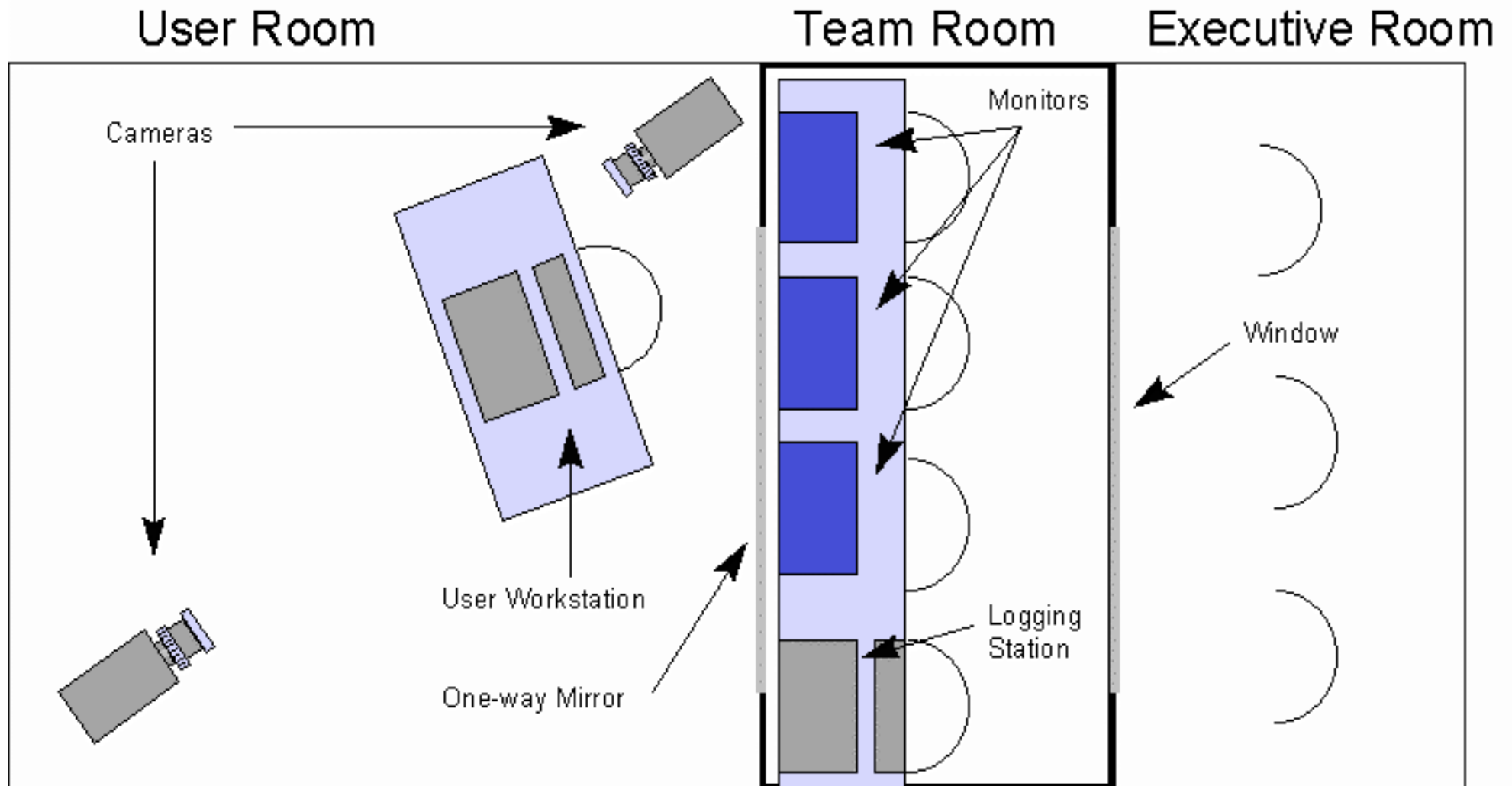
- **Artefakte aufgrund selektiver Stichprobenauswahl:**

Selektionseffekte: Bei der Zusammenstellung der Stichprobe gelangen Personen mit bestimmten Merkmalen unbeabsichtigt in größerer Anzahl in die Stichprobe als es einer repräsentativen Auswahl entsprechen würde.

→ Nutzer, die für einen solchen Test ihre Freizeit zu opfern bereit sind, unterscheiden sich deutlich von der Masse.

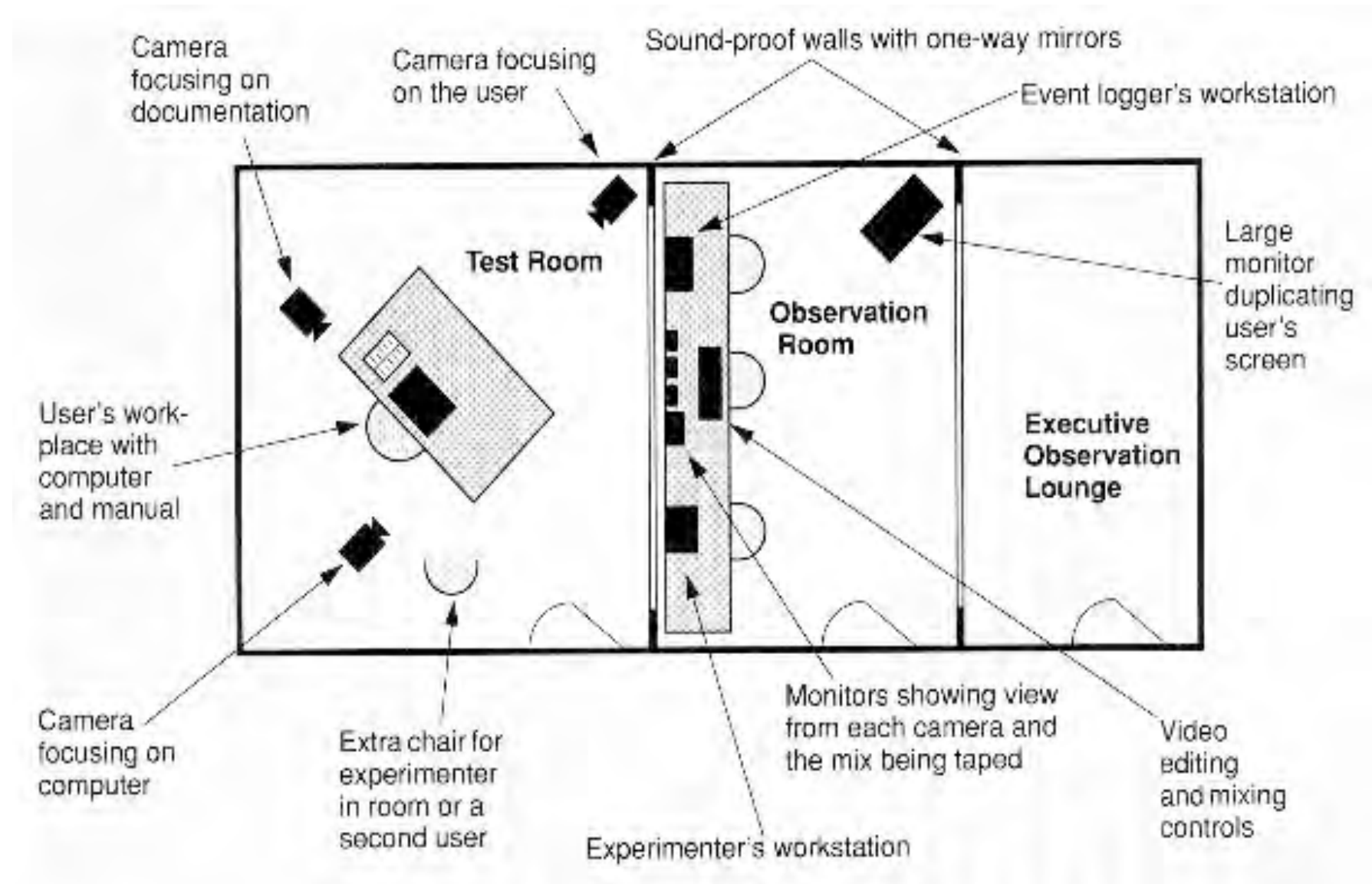
→ Studenten sind aufgrund der für sie attraktiven Bezahlung häufiger als andere Personengruppen bereit, als Testperson tätig zu werden.

2.4.3.3 Ausstattung eines typischen Usability-Labors



2.4.3.4 Methoden der Datenerhebung bei Usability-Tests

- **Videoaufzeichnungen**



- **Methode des lautend Denkens**

- Qualitative Methode zur Erfassung bewusster, handlungsbegleitender Kognitionen und Emotionen.

- Testpersonen äußern ihre Eindrücke, Überlegungen, Probleme, Bewertungen, Emotionen während des Handelns.

- Ist **nicht in Kombination mit Leistungsmaßen** anwendbar, da durch die Verbalisierungsanforderungen die Reaktionen verzögert und Entscheidungen fehleranfälliger werden (Doppeltätigkeit).

- Unterstützung durch Testleiter mit Hilfe von Fragen:

 - „Was denken Sie gerade?“

 - „Was denken Sie, was diese Meldung zu bedeuten hat?“

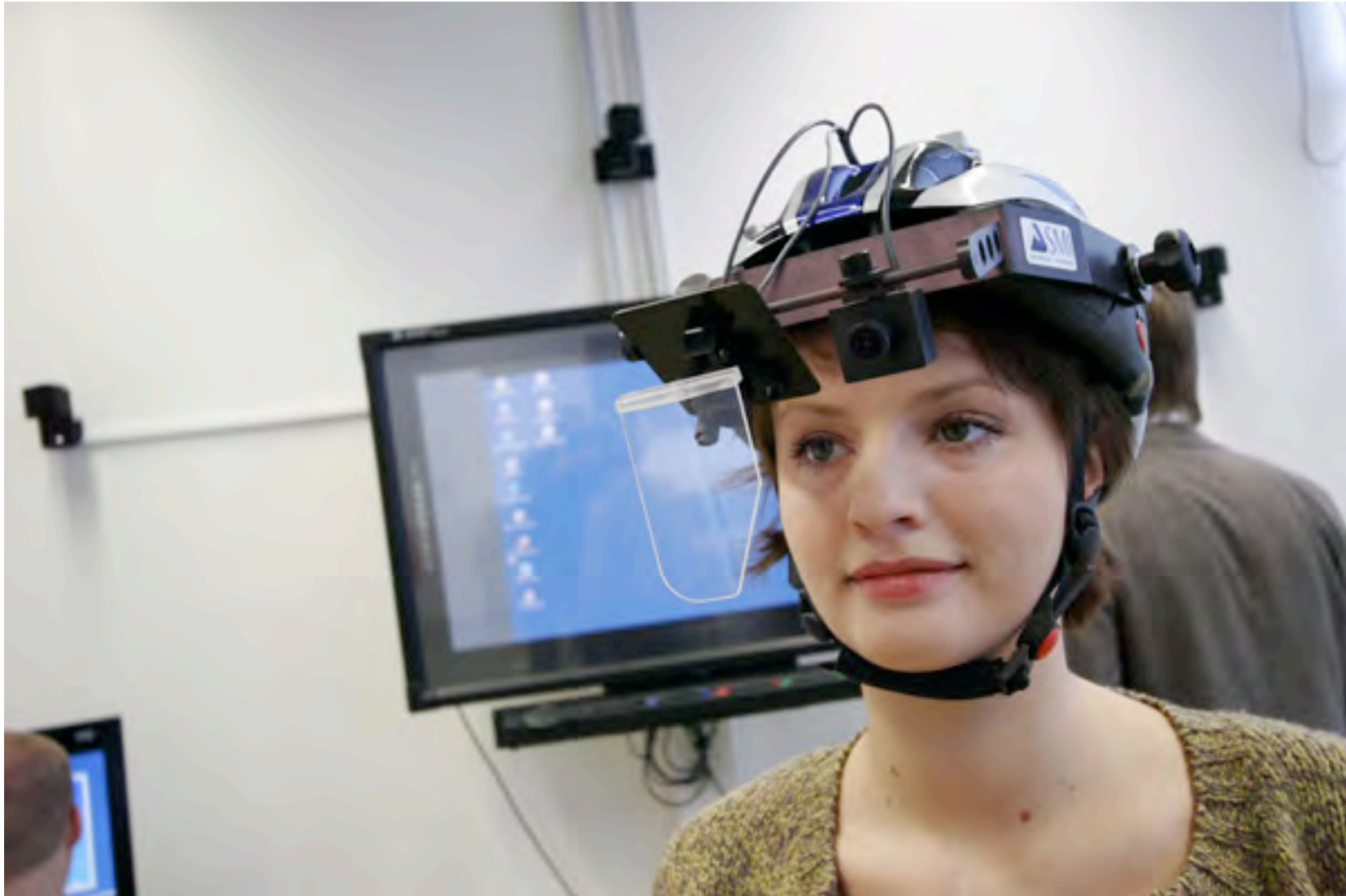
 - „Was denken Sie, was jetzt passieren wird?“

 - „Welche Reaktionen haben Sie erwartet?“

- **Leistungsdaten: Zeit- und Fehlerwerte**

- **Zeitmaße:**
 - Zeit für eine Handlung bei Bearbeitung einer Aufgabe (Reaktionszeit)
 - Zeit für eine Fehlerbehebung
 - Zeit, die nicht genutzt wird (z.B. systembedingte Wartezeiten)
 - Zeit zwischen Ende der letzten Handlung und Beginn der nächsten (z.B. Zeit zum Überlegen des nächsten Schritts)
- **Fehlermaße:**
 - Zahl der Fehler
 - Verhältnis von erfolgreichen Handlungen zu Fehlern (Fehlerprozent)
- **Häufigkeitsmaße:**
 - Anzahl der verschiedenen Funktionen eines Systems, die bei einer bestimmten Aufgabe benutzt werden
 - Verhältnis positiver zu negativer Äußerungen
 - Verhältnis der Verwendung unterschiedlicher Lösungswege bei Vorhandensein verschiedener Lösungswege

andere Leistungsmaße: z.B. Blickbewegungsregistrierung, Fokussierdauern, Registrierung beachteter Orte auf dem Bildschirm



Fragebogenmethoden: Allgemeine Gesichtspunkte

Vorteile gegenüber anderen Methoden	Nachteile gegenüber anderen Methoden
Keine Beeinflussung durch Dritte (z.B. Interviewer) → hohe Objektivität	je strukturierter das Verfahren, desto weniger Freiraum bleibt dem Nutzer
subjektive Daten können kategorisiert und statistisch analysiert werden	Qualität der Daten abhängig von der Qualität des Erhebungsinstruments
geringer Aufwand, geringe Kosten; daher in mehreren Phasen des iterativen Testens einsetzbar	Entwicklung qualitativ hochwertiger Fragebogen ist sehr aufwendig
Vielzahl standardisierter Fragebogen vorhanden	Befragte sind auf ihr Gedächtnis angewiesen → Fehlerquelle
große Stichprobe möglich durch Onlineerhebungen	auf bewusste Repräsentation von Fehlern und Problemen angewiesen

- **Spezielle Fragebogen**

QUIS: Questionnaire for User Interface Satisfaction (Shneiderman, 1987)

Ziel: Messung der subjektiven Zufriedenheit der Nutzer mit der Schnittstelle eines Softwaresystems

Langversion → 90 Items / Kurzversion → 20 Items

Itemkonstruktion: Jedes Item bezieht sich auf einen Aspekt des Systems und verlangt ein Urteil auf einer 9-stufigen Skala, deren einer Endpunkt durch ein positives Adjektiv, deren anderer durch ein negatives Adjektiv markiert ist. „Nicht bewertbar“ (n.b.) ist eine zusätzliche Antwortmöglichkeit.

Beispiele für Items:

Mitteilungen, die auf dem Bildschirm erscheinen:

verwirrend 1 2 3 4 5 6 7 8 9 klar / n.b.

Instruktionen für Befehle oder Auswahlen:

verwirrend 1 2 3 4 5 6 7 8 9 klar / n.b.

Kritik:

- Items beziehen sich zwar auf verschiedene Aspekte einer Software, tun dies aber pauschal, d.h. die jeweilige Stelle, die zu dem kritischen Urteil Veranlassung gibt, ist nicht identifizierbar → Fehler sind daher nicht eindeutig lokalisierbar!
- Kein Bezug zur ISO-Norm 9241

Positiv:

- Empirisch ermittelte recht hohe Reliabilität (Chin et al., 1988);
- computergestützte Version QUOS 7.0 sowie Online-Version liegen vor.

SUMI: Software Usability Measurement Inventory

→ International wohl der bekannteste und am häufigsten eingesetzte Fragebogen zur Benutzerzufriedenheit von Software.

Ziel: Messung von Eindrücken und Gefühlen des Nutzers beim Umgehen mit der zu evaluierenden Software.

Itemkonstruktion: Die 50 Items von SUMI sind als positive oder negative Aussagen über verschiedene Aspekte der Qualität der Arbeit mit dem Softwaresystem formuliert, denen der Beurteiler entweder zustimmt, nicht zustimmt oder ein Urteil „unentschlossen“ abgibt.

Beispiele für Items:

Item	Zustimmung	unent- schlossen	keine Zustimmung
Wenn diese Software stoppt, ist es nicht einfach, sie wieder zu starten.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Software hat eine sehr attraktive Aufmachung.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich finde Gefallen an meiner Arbeit mit dieser Software.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Art und Weise, in der das System Informationen darbietet, ist klar und verständlich.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Es ist einfach, die Software dazu zu bringen, genau das zu tun, was man will.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kritik:

- Reliabilität und Validität sind nicht bekannt.
- Items stellen Pauschalurteile dar, d.h. die jeweilige Stelle, die zu dem Urteil Veranlassung gibt, ist nicht identifizierbar → Fehler sind nicht eindeutig lokalisierbar!
- Kein Bezug zur ISO-Norm 9241

Positiv:

- leicht, schnell, bei großer Zahl von Beurteilern und preiswert anwendbar.

IsoNorm 9241/10

Ziel: Beurteilung der Gebrauchstauglichkeit von Software am Arbeitsplatz; geeignet sowohl für formative wie summative Evaluation.

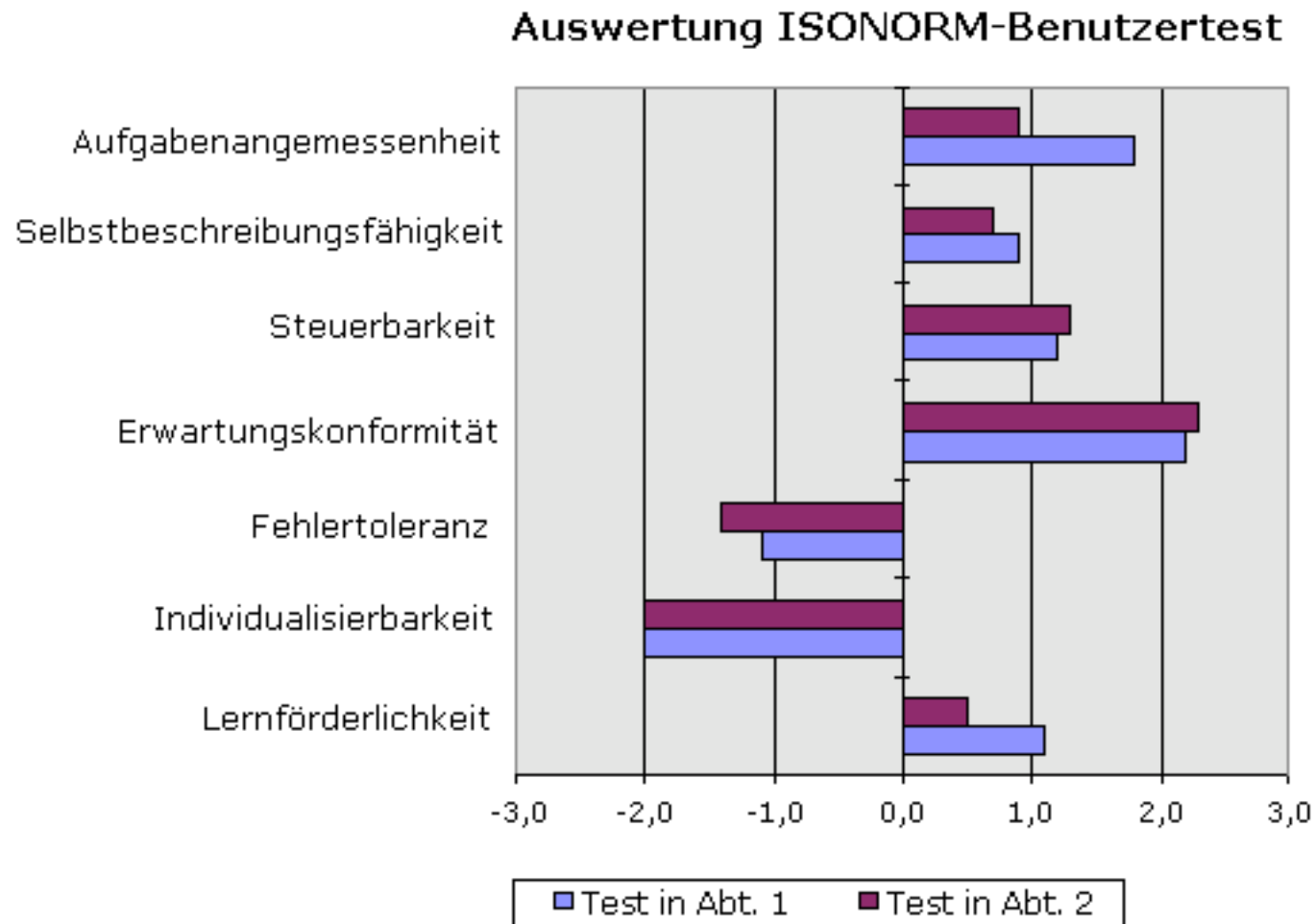
Itemkonstruktion: Dem Fragebogen liegen die 7 Gestaltungsanforderungen der DIN EN ISO 9241 zugrunde. Pro Kriterium werden 5 Fragen gestellt, die durch Skalierung auf einer siebenstufigen Skala zwischen „sehr positiv“ bis „sehr negativ“ vorgenommen wird.

Beispiele für Items:

Aufgabenangemessenheit: Unterstützt die Software die Erledigung Ihrer Arbeitsaufgaben, ohne Sie als Benutzer unnötig zu belasten?

Die Software...	--- -- - -/+ + ++ +++	Die Software...
... ist kompliziert zu bedienen.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	... ist unkompliziert zu bedienen.
... bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	... bietet alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.
... bietet schlechte Möglichkeiten, häufig sich wiederholende Bearbeitungsvorgänge zu automatisieren.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	... bietet gute Möglichkeiten, häufig sich wiederholende Bearbeitungsvorgänge zu automatisieren.

Zusammenfassung der Ergebnisse nach den ISO-Normen möglich:



Kritik:

- Keine bekannte Reliabilität.
- Wegen der pauschalen Erfassung der ISO-Kriterien keine genaue Lokalisierung der ergonomischen Fehler (wie bei systematischen Usability-Beobachtungsstudien möglich).

Positiv:

- leicht, schnell, bei großer Zahl von Beurteilern und preiswert anwendbar.
- direkter Bezug zur ISO-Norm → Validität wahrscheinlich
- aufgrund von Normierung (Prümper, 1997) unmittelbarer Vergleich der Ergebnisse mit anderen Systemen möglich.
- Online-Version liegt vor.

• **Vergleich der Evaluationsmethoden:**

Beurteilung: ● → gering / ● ● → mittel / ● ● ● → hoch /
 (●) → zusätzlich bei Verwendung von Leistungsmaßen

Kriterium	Guide- lines: ISO-Norm 9241	EVADIS : Experten- leitfaden	Cognitive Walk Through	Usability- Tests (Beo- bachtung)	Frage- bogen
Produktivität	● ● ●	●	● ●	● ● ●	●
Materieller Aufwand	●	●	● ●	● ● ●	●
Zeitaufwand	● ● ●	● ●	● ●	● ● ●	●
Qualifikation der Evaluatoren	●	● ● ●	● ●	● ● ●	●
Detaillierungsgrad	● ● ●	●	● ●	● ● ●	●
Flexibilität	● ●	● ●	● ● ●	● ● ●	●
Vorhersagekraft	●	● ●	● ●	● ● ●	● ● ●
Objektivität	●	● ●	●	● ● (●)	● ● ●
Reliabilität	● ●	● ● ●	● ●	● ● (●)	● ● ●
Validität	● ● ●	● ●	●	● ● ●	●