

Gestaltung ergonomischer und intuitiver Benutzeroberflächen unter Windows - mit MS Access

GRUNDLAGEN.....	2
BEGRIFFE	2
REGELN	2
HISTORISCHE ENTWICKLUNG DES GUI-DESIGNS.....	2
IST DESIGN NÖTIG?.....	3
GUTES DESIGN	3
SCHLECHTES DESIGN	3
PRINZIPIEN FÜR BENUTZEROBERFLÄCHEN LT. DIN.....	4
AUFGABENANGEMESSENHEIT.....	4
SELBSTBESCHREIBUNGSFÄHIGKEIT	6
STEUERBARKEIT.....	8
ERWARTUNGSKONFORMITÄT	8
FEHLERTOLERANZ.....	8
INDIVIDUALISIERBARKEIT	9
LERNFÖRDERLICHKEIT	9
WICHTIGE GESTALTUNGSASPEKTE.....	10
GLIEDERUNG - STRUKTURIERUNG	10
GRAFISCHE GESTALTUNG – ÄSTHETIK	10
DIREKTHEIT.....	11
LERNENDE FUNKTIONEN.....	12
ANWENDER VERSTEHEN/BE(OB)ACHTEN	12
KONSISTENZ.....	12
FENSTER	13
DIALOGFENSTER	13
MELDUNGSFENSTER.....	13
REGISTER	13
MENÜS	14
KONTEXTMENÜS	15
SYMBOLLEISTEN	15
STATUSLEISTE	15
BENUTZERDEFINIERTER MENÜS	15
MENÜ-TEST.....	15
DER WINDOWS-LEITFADEN VON MS	16
GRUNDLEGENDE INTERAKTIONSTECHNIKEN	16
FENSTERTYPEN.....	16
PRIMÄRFENSTER.....	17
SEKUNDÄRFENSTER.....	17
SDI UND MDI	18
WEITERE RICHTLINIEN	18
ENTWURFSZYKLUS	18
INSTALLATION UND REGISTRIERUNG	19
EINGABEHILFEN	19
INTERNATIONALISIERUNG	19
LITERATUR, LINKS	20

Grundlagen

Begriffe

Ergonomie = Zweig der Arbeitswissenschaft, der sich mit der Anpassung der Technik an den Menschen (zur Erleichterung der Arbeit) befasst. (Störig)

Intuition = unmittelbares Erkennen von Vorgängen und Zusammenhänge vom Gefühl her (Störig)

GUI = Graphical User Interface --> Grafische Benutzeroberfläche

GUI-Design = Gestaltung der Schnittstelle zw. Anwender und Computer

WIMP = Windows, Icons, Menus, Pointing (a. Windows, Icons, Mouse, Pulldowns)
Kürzel für die Merkmale und Gemeinsamkeiten sämtlicher graphischen SW-Benutzeroberflächen

Regeln

- Design-Grundregeln
gelten für die Gestaltung sämtlicher Produkte
Grundlagen kommen z.B. aus der Gestaltpsychologie (v.a. Max Wertheimer), Kommunikationswissenschaften (v.a. Leseforschung), Kunst und Kunsthandwerk (Gestaltungsprinzipien aus Architektur, Malerei, Gebrauchsdesign etc.) bzw. Kunstwissenschaften
- besondere Anforderungen für Software-Design
DIN-Normen v.a. DIN EN ISO 9241 - Grundsätze der Dialoggestaltung
Benutzeroberfläche, Zeichenanordnung, Farben, Menüs, Masken und Dialoge
- Windows-Richtlinien von Microsoft (s. MSPress 1995)

Weitere Faktoren sind Möglichkeiten und Beschränkungen durch die jeweilige Entwicklungsumgebung und den konkreten Auftrag/Auftraggeber.

Unterschiede zw. Gestaltung v. Datenbanken, Spielsoftware, Internet, Steuerungs-SW für Maschinen etc. aber auch sehr viele Gemeinsamkeiten und Regeln, die für alle gelten.

historische Entwicklung des GUI-Designs

- Fenster und Maus wurden in den 1960ern von Douglas Engelbart erfunden
- längere Tradition bei Apple und and. meist daran angelehnten GUIs (DOS-Shells, Amiga)
- im Intel-Bereich v.a. ab Win 3.0 und OS2
- stark steigender Bedarf durch die rel. einfach zu bedienenden graphischen Entwicklungswerkzeuge wie VB oder Access, Motto: jeder sein eigener Entwickler, SW-Entwicklung wurde zum Breitensport

Ist Design nötig?

Design (Oberflächengestaltung) findet immer, bei jeder Applikation, statt d.h. es gibt nur gutes oder schlechtes Design, durchdachtes oder missachtetes, aber **nie kein Design**.

Design ist mehr als das bloße Behübschen von Oberflächen. Es betrifft - neben der ästhetischen Gestaltung der Oberfläche - die Strukturierung des Programmes, der Abläufe, der ganzen Interaktion mit dem Benutzer.

Wichtig ist das Bewusstsein, dass die Oberfläche der einzige Bestandteil einer Anwendung ist, mit dem der Normalanwender in Berührung kommt.

Gutes Design

ermöglicht dem Anwender

- leichtere Einarbeitung
- schnelleres, effizienteres Arbeiten
- Übertragbarkeit von Wissen (v.a. durch Standards)
- weniger Bedienungsfehler
- Konsistenz (unternehmensweite CI etc.)

→ spart effektiv Kosten beim Kunden

→ bringt größere Akzeptanz und Zufriedenheit beim Benutzer

Längerfristig d.h. bei Vorliegen von Erfahrung, erprobten Konzepten und Vorlagen spart überlegtes Design auch beim Entwickler Zeit und Kosten, weil nicht mehr mühsam herumprobiert werden muss, um eine Aufgabenstellung designmäßig umzusetzen.

Schlechtes Design

kann darüber hinaus

- gute Programmierung oder Funktionalität zudecken
die tollste Funktion nützt nichts, wenn der Anwender sie nicht findet oder nicht (einfach, effizient) bedienen kann
- ein Wettbewerbsnachteil sein
ein von der technischen Funktionalität her weniger leistungsfähiges Konkurrenzprodukt wird gewählt, weil die Oberfläche "schöner" oder besser zu bedienen ist

Gründe für schlechtes Design

- Wichtigkeit wird unterschätzt d.h. mangelndes Bewusstsein
- Zeit- und Geldmangel ("Hauptsache, es läuft")
- fehlende Ausbildung (DB-Stricker) und fehlende Leitfäden

Prinzipien für Benutzeroberflächen lt. DIN

festgehalten in der Norm EN ISO 9241 Teil 10

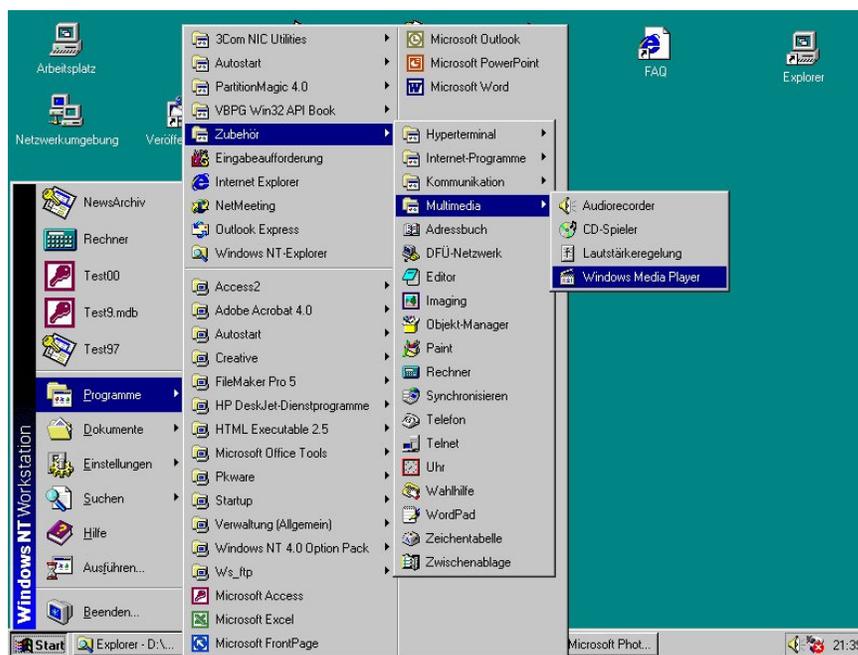
- Aufgabenangemessenheit
- Selbstbeschreibungsfähigkeit
- Steuerbarkeit
- Erwartungskonformität
- Fehlertoleranz
- Individualisierbarkeit
- Lernförderlichkeit

Aufgabenangemessenheit

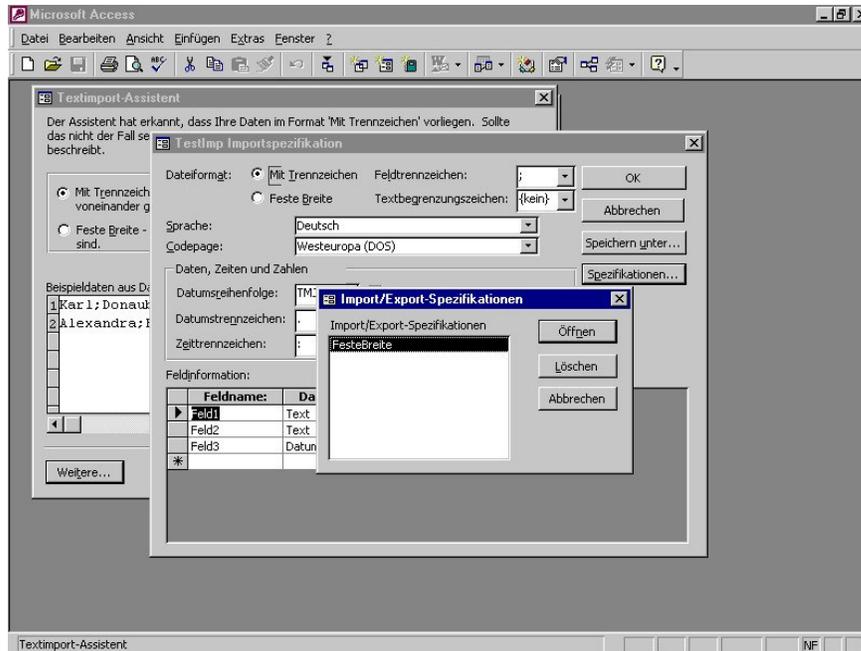
DIN: Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen.

- Hierarchisches System beim Informationsfluß d.h. Information nach Priorität gliedern
- Information erst aufgrund der passenden Situation oder auf Wunsch
- mit geringem Aufwand erreichbar machen = kurze Wege, schnelle und einfache Navigation

Test durch Zählen der nötigen Mausklicks oder Tastatureingaben oder Bildschirme, um eine Aufgabe durchzuführen oder an eine best. Information zu gelangen



WinNT 4, suboptimale Menüerschachtelung --> verlangt viel Feinmotorik



Access 2000, Importspezifikation, tiefe Verschachtelung, viele Fenster, Versteckspiel

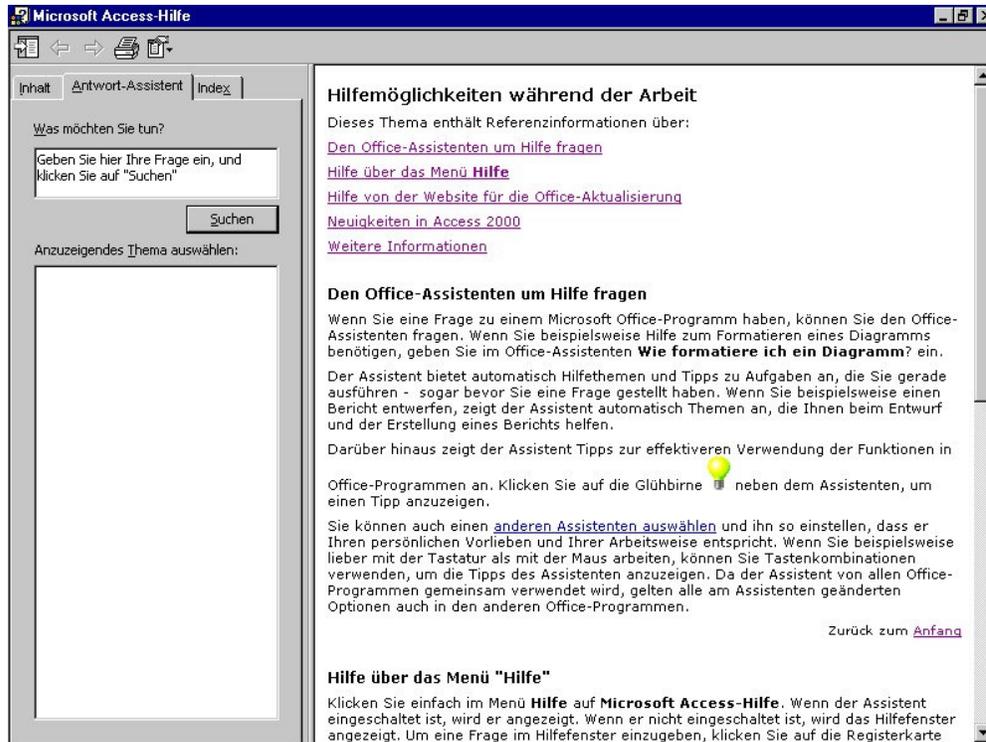
- Automatisierung

keine unnötigen Arbeitsschritte z.B.

Fokus beim Öffnen von Masken auf wichtige und passende Steuerelemente setzen



Outlook Express 5.0, Suchfenster öffnet sich mit Cursor (Kreis) in nicht-editierbarem Feld



A00, Online-Hilfe: Kein erkennbarer Fokus – sollte auf Text im Suchfeld sein

- Vorgabe von sinnvollen Standardwerten
in Access zuletzt gewählte Werte beim nächsten DS, wenn Wahrscheinlichkeit hoch ist
- Leichte Erreichbarkeit von sehr oft benutzten Funktionen
Selten benötigte Funktionen nicht zu prominent gestalten. (betrifft z.B. Vorkommen und Stellung in Symbolleisten)
- Berücksichtigung der Anwenderkenntnisse und -gruppen
Windows-Einsteiger haben oft Probleme mit Mausbedienung (Ziehen, Doppelklicken, linke/rechte Maustaste), Fensterverwaltung, Dateiverwaltung → Vereinfachen, Verallgemeinern der Bedienung
Wege für Anfänger (Menü, Symbol) und "Poweruser" schaffen (Tastenkombinationen etc.)
Bedienung per Maus und Tastatur ermöglichen
- Einfachheit d.h. nicht zu viel unnötiger, ablenkender Schnickschnack
aber geschickt angewandtes, geringes Ausmaß an Schnickschnack kann positive, emotionale Bindung stärken z.B. etwas Humor und Originalität bei der Icon-Wahl, Grafiken, Namensgebung etc. (s. Plüschtiere und Grünpflanzen am Monitor)

Selbstbeschreibungsfähigkeit

DIN: Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschritt durch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklärt wird.

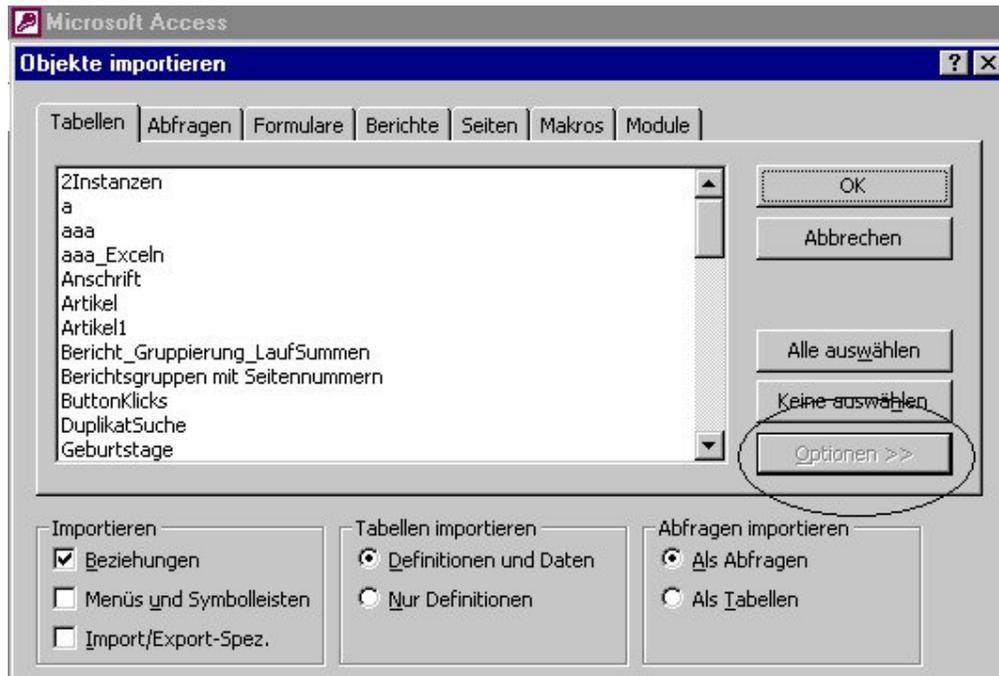
Intuitiv heißt im SW-Bereich v.a. der Benutzer soll sich auf den ersten Blick zurechtfinden, ohne große Erklärungen. d.h. logische Abfolgen, die sich selbst erklären.

Anwender muss zur Bedienung keine neuen besonderen Fertigkeiten lernen oder sich - nur zur Bedienung der Software - neue Kenntnisse aneignen.

dabei gibt es fließende Übergänge:

manche Dinge sind dem durchschnittlichen Anwender oder erfahrenen Windows-Anwender bekannt → quasi intuitiv; für Neulinge nicht (best. Symbole, Tastenkombinationen etc.)

- aktuelle Möglichkeiten bzw. aktuellen Zustand anzeigen
d.h. der Anwender soll jederzeit orientiert sein
- Deutlichkeit bei den Möglichkeiten der Bedienelemente
rechter Mausklick oder Doppelklick sind nicht sehr intuitiv
flacher Webstil bei Schaltflächen ist in dieser Hinsicht bedenklich
- kein Versteckspielen
d.h. keine "geheimen", hinter nichtssagenden Menüpunkten oder Schaltflächen versteckten Funktionen, nicht sichtbaren Optionen, übertriebenen Tastenkürzel etc. (Bsp. <F8> beim Windows-Start, "Extras" oder "Weitere" als Beschriftung oder Menünamen)



Access 2000, Import-Dialog mit versteckten, wichtigen Optionen

- Symbole, Abläufe etc. aus der Realität auf den Computer übertragen
Kalender nachbilden, Telefon-Tastatur etc.
- sprechende Metaphern verwenden
bei Icons, Bildern etc. falls nicht möglich, dann Beschriftungen
- Feedback liefern
Rückmeldung möglichst nahe an der Aktion
keine toten Bildschirme sondern Meldungsfenster, Fortschrittsbalken, Sanduhr etc. bei Durchführung längerer Operationen
bei Abschluss einer Operation Meldung: Die Daten wurden gespeichert, Vorgang xyz (konkret nennen!) wurde erfolgreich ausgeführt etc.

Steuerbarkeit

DIN: *Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.*

SW soll als Werkzeug dienen d.h. der Anwender den Eindruck haben, dass er Herr der Lage ist. Er kontrolliert die SW, nicht umgekehrt. Er setzt die Aktionen, steuert das Geschehen und die SW reagiert entsprechend.

- Abläufe durchschaubar halten - User startet und beendet
- SW unterstützt nur bei Fehlern oder Konflikten
- Meldungen und Dialoge entsprechend gestalten

Erwartungskonformität

DIN: *Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z.B. seinen Kenntnissen aus dem Arbeitsgebiet, seiner Ausbildung und seiner Erfahrung sowie allgemein anerkannten Konventionen.*

Der Anwender profitiert von der Einhaltung von Konventionen, weil er mit Windows gemachte Erfahrungen übertragen kann und nicht neu lernen muss. Durch nicht konforme Vorgänge oder Einstellungen wird er hingegen leicht irritiert. Er soll sich aber auf seine eigentlichen, inhaltlichen Aufgaben konzentrieren können, nicht auf das Erlernen neuer Konventionen oder Kämpfen mit einzelnen Abweichungen.

Benutzer erwartet, dass die SW gewisses Maß an Windows-Standardverhalten zeigt z.B.

- Aussehen und Bedienung der Fenster
- Standard-Dialoge
- Standard-Verhalten von Steuerelementen
- Standardsymbole und Menüs
- Standard-Tastenkürzel wie <F1> für Hilfe, <Strg>+ <C> etc. zur Bedienung der Zwischenablage - nicht anders belegen
- Navigation auch mit Pfeiltasten und Return-Taste - nicht nur mit Tab

MS ist da selber nicht immer konsistent z.B.

im Access-Formularentwurf Umschalttaste statt <Strg> zur erweiterten Auswahl von Elementen <Strg>+<Alt>+<Entf> zur Anmeldung in NT, während das sonst die Tastenkombination zum Neustart etc. war (zudem unsinnig kompliziert für eine Anmeldung)

Fehlertoleranz

DIN: *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.*

Vieles lässt sich durch gutes Design vermeiden, aber unerwartete und falsche Aktionen und Eingaben sind nie ganz zu verhindern und auch nie zur Gänze vorhersehbar. Anwender sind in dieser Hinsicht sehr kreativ.

- Rückgängig- und Abbrechen-Funktionen anbieten
- Präventive Maßnahmen zur Fehlervermeidung setzen
z.B. in einer best. Situation nicht anwählbare Menüpunkte oder Steuerelemente abblenden
→ erspart Fehler(meldung)
Rückfragen bei Löschooperationen oder großen Änderungen
- Eingabevalidierung
"sanfte" Korrekturen oder Nachfragen bei Eingabefehlern, mit verständlichen Meldungen durchführen
Prüfung möglichst frühzeitig, ideal bereits bei der Eingabe in das betreffende Feld
bei ungültiger Eingabe Ablehnung oder Rückmeldung, Prävention durch einschränkende Steuerelemente (Listenfelder, Optionsgruppen etc.)

bei späterer Prüfung Fenster geöffnet lassen, Meldung, Fokus auf das betroffene Steuerelement

Individualisierbarkeit

DIN: *Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe sowie an die individuellen Fähigkeiten und Vorlieben des Benutzers zulässt.*

benutzerdefinierbare Einstellungen ermöglichen

Wenn der Anwender selbst etwas festlegen oder verändern kann, gibt das neben der größeren Anpassungsfähigkeit der SW v.a. auch den Eindruck von Flexibilität und mehr Sicherheit und Selbstbewusstsein. z.B.

- zusätzliche (vorbereitete) Felder selber definieren
- Meldungen und Nachfragen abschaltbar halten
- Standardwerte belegen lassen etc.
- Möglichkeiten vorsehen, Funktionen auf versch. Arten zu steuern
d.h. Anwender kann sich eigene Wege der Bedienung suchen z.B. mit Maus oder Tastatur

Lernförderlichkeit

DIN: *Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.*

- Online-Hilfe, Tipp-Texte
- Abläufe, Bilder, Begriffe aus dem Alltag und konkretem Einsatzgebiet der SW
- Regeln, Konventionen und Konsistenz dabei

Wichtige Gestaltungsaspekte

- Gliederung
- grafische Gestaltung
- Direktheit
- lernende Funktionen
- Anwender verstehen/be(ob)achten
- Konsistenz

Gliederung - Strukturierung

Ziele: Klarheit, Deutlichkeit, einfache Lesbarkeit
gilt für Anwendung, einzelnes Fenster, bis zum letzten Schriftzug

Bei manchen Typen von Applikationen ist Innovation, Abwechslung, "Thrill" gut (Spiele etc.).
Bei Datenbankapplikationen ist da große Vorsicht geboten.
Im Zweifelsfall Funktionalität und Deutlichkeit vor Ästhetik stellen (z.B. flacher Webstil bei Schaltflächen und Vertiefung erst bei Mausbewegung ist nicht bes. deutlich).

- Struktur und ausgewogene Aufteilung
Eine tragende Struktur ist eines der wichtigsten Prinzipien der grafischen Gestaltung d.h. Linien, denen der Anwender folgen kann und Schwerpunkte zur Konzentration, ausgewogene Verteilung der Elemente und effiziente Raumausnutzung
- Navigation
in westlichen Ländern von links nach rechts und von oben nach unten
→ primäres Steuerelement, mit dem der Benutzer interagiert oben links anbringen
Steuerelemente so anordnen, dass sich der Benutzer in einer logischen Folge durch das Fenster bewegen kann, Befehlschaltflächen, die auf das ganze Fenster wirken befinden sich am Ende der Sequenz
- Hierarchie
Hervorhebung der wichtigen Informationen durch Stellung (exponierte Lage auf Blickachse), besondere Formatierung, Fokussetzen
Proportionen gezielt zur Betonung von Wichtigkeit einsetzen

Grafische Gestaltung – Ästhetik

"Auch die durchdachtste Funktionalität einer Anwendung leidet unter einer schlechten optischen Präsentation." (MSPress 1995, S 355)

Eine ruhige, harmonische Oberfläche erleichtert die Orientierung, stresst bei längerer Arbeit weniger als eine nicht nach ästhetischen Grundprinzipien gestaltete. Darüber hinaus sind diese psychologischen Wirkungen sehr wichtig für die Zufriedenheit des Anwenders mit der SW.

- Ausrichten von Elementen aneinander und an Rastern bzw. Linien
horizontale, vertikale und diagonale Ausrichtung der Elemente beachten

z.B. Haupt-Befehlsschaltflächen in einer Zeile des unteren Randes, wichtigste Schaltfläche

als erste im Satz

"Ok" und "Abbrechen" immer zusammen gruppieren

besondere Anordnungen bei natürlicher Beziehung z.B. Himmelsrichtungen kompassartig

- Abstände harmonisieren

Elemente zueinander, zu Fensterrändern, zu Applikationsrändern

Es gibt exakte Richtlinien für Größen und Abstände

Im Windows-Styleguide und anderen Werken gibt es genaue Bemaßungsregeln

Wichtiger als das exakte Einhalten dieser Richtlinien ist Konsistenz d.h. gleiche Abstände innerhalb des Fensters und durch die Anwendung hindurch

- Gleichmäßigkeit bzw. Proportionalität in Höhe und Breite aller Elemente
- Blicklinien und Blickachsen
gelesen wird - auch eine Bildschirmseite - in unserer Kultur von links nach rechts und von oben nach unten → bei Anordnung der Elemente beachten
- Beziehungen zwischen den Elementen
sollen auf den ersten Blick klar erkennbar sein durch z.B.
räumliche Nähe, ähnliche Gestaltung, umfassende Rechtecke, Linien etc.
d.h. Schaltflächen, die ein anderes Steuerelement verändern, in der Nähe desselben platzieren; verwandte Komponenten in Gruppen zusammenfassen
- Oberfläche nicht überfrachten
mit zu viele Elemente oder Funktionen
Extrembeispiel war früher die Optionen-Einstellung bei Win3.x
- Farbe
sparsam einsetzen, gezielt zur Hervorhebung, Kennzeichnung, Markierung
mit gleicher Farbe kann man z.B. Beziehungen zwischen Elementen betonen
emotionale Aspekte von Farben beachten (warm, kühl)

Achtung: 9% aller männl. Erwachsenen haben Probleme bei der Farbwahrnehmung ("Farbenblindheit") d.h. nicht als einziges Mittel oder primäre Form der Information verwenden

- Schrift
sparsam einsetzen, Standardschrift in Windows ist MS Sans Serif 8-Punkte
serifenlose Schriften sind i.d.R. am Bildschirm besser lesbar
Schriftarten und Schriftformatierungen sparsam einsetzen
(Prinzipien aus dem Druckerhandwerk, keine oder sehr wenige exotische Fonts)
- Klänge
nur als sekundäres Mittel einsetzen
(kann abgeschaltet werden, Hörbehinderung)
- 3-Dimensionalität nicht übertreiben
Elemente sind sonst schwer auszumachen, unruhige Wirkung
- Mauszeiger selten und gezielt verändern
wenn er verändert wird, dann auf Konsistenz achten, auf klar erkennbaren Hotspot achten

Direktheit

durch direkte Manipulation, Anzeige, Feedback

- Elemente verwenden, die bei Bedienung unmittelbare vorhersehbare Auswirkungen zeigen
v.a. Schaltflächen (drücken - vertiefen - Vorgang starten)

- Drag & Drop mit markierten Elementen
- Mauszeiger sprechend anpassen ("+" bei Kopierstatus, Pfeile bei Größen- oder Positionsänderungen)
- Menüs, die immer wieder abgeklappert werden müssten, vermeiden; besser direkt anzeigen über Symbole oder Buttons

lernende Funktionen

auch SW kann/sollte lernende d.h. unterstützende Funktionen bieten z.B.

- neue Stammdaten, die beim nächsten Mal angeboten werden (PLZ, Anreden etc.)
- häufig genutzte Inhalte vorschlagen (Standardwert)
- letzte Einstellungen merken und anbieten (zuletzt bearbeiteter Datensatz etc.)
"Dialog künftig nicht mehr anzeigen"-Funktionen

Anwender verstehen/be(ob)achten

- Anwendung nicht für sich sondern für möglichst konkrete Benutzer schreiben

Entwickler ist ein sehr erfahrener Windows-Anwender. Standards und Konventionen sind internalisiert. Sie fallen nur mehr auf, wenn sie grob verletzt werden, d.h. eine Sache nicht funktioniert, eine Funktionalität nicht auffindbar ist usw.

Als Entwickler sollte man sich dieser Standards und Konventionen aber bewusst sein, um sie nicht aus Gedankenlosigkeit oder doch schlichter Unkenntnis zu verletzen.

- Wird die SW i.d.R. auf erfahrene Enthusiasten oder evtl. ängstliche Anfänger treffen?
- Wie ist die Einstellung des konkreten Anwenders zum Computer, zur künftigen SW?
- Ist er technisch versiert, interessiert? Kennt er die Computer-Fachtermini?
Wie sicher bewegt er sich in Windows? (Tastenkürzel, Markierungsvarianten, Kontextmenüs
Symbol- und Menüleistenverwendung etc.)
- Nicht zuviel voraussetzen
z.B. Feinmotorik (verschachtelte Menüs, kleine Elemente, Doppelclicks)
- Gelegenheit geben für Anmerkungen, Verbesserungsvorschläge, Fehlerbeschreibung
evtl. eigenes Formular und Bericht mit Emailmöglichkeit

Konsistenz

wichtige Forderung an den Entwickler

gilt für alle genannten Prinzipien und Richtlinien

innerhalb des Produkts: einheitliches Design durch die ganze Anwendung hindurch

innerhalb der Umgebung: an Windows-Konventionen angelehnt

Fenster

Dialogfenster

- nicht unnötig groß gestalten
aber mit Bedacht eingesetzter Leerraum ist wichtiges Organisationsmittel, ausgewogenes Maß zwischen den Elementen, abhängig von Elementanzahl
- Position von Child-Fenstern im Hauptfenster
nie willkürlich sondern überlegt, zentriert oder gezielt choreografieren
- unterschiedliche Färbung von Fensterhintergründen ist bei Windows nicht nötig (evtl. anders in DOS-Zeiten), denn Fenster haben klare Begrenzungen
- wenige Schriftarten und -formatierungen
- Anzahl der geöffneten Fenster begrenzen → Übersichtlichkeit
- wird ein Dialog- (oder Meldungs-) Fenster aus einem anderen aufgerufen, sollte Startpunkt klar sein (entweder offenhalten oder z.B. in der Titelleiste Bezug nehmen)
- Nicht zu viele Elemente in ein Fenster packen, aber auch nicht zuviel Verteilung d.h. Notwendigkeit viele Fenster offen zu halten → durchdachte Kompromisse

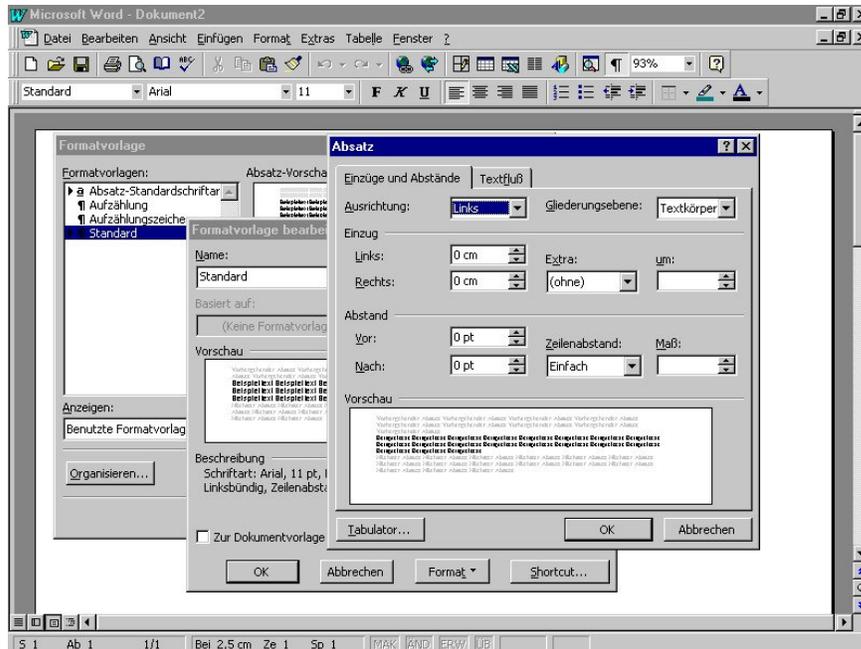
Meldungsfenster

- Texte konkretisieren
z.B. statt "Kunde löschen?", "Datei schließen?" den Namen nennen
- wichtige Teile z.B. mit Zeilenumbruch hervorheben - sparsam mit fett, kursiv, versch. Font umgehen
- am häufigsten absehbare Schaltfläche als Standard vorbelegen
Ausnahme: bei kritischen d.h. folgenreichen Buttons ist der harmloseste vorzubelegen

Register

natürlich keine Fenster, aber ebenfalls Container-Element, deshalb viele Gemeinsamkeiten

- keine zusätzlichen Dialog-Fenster über Buttons öffnen sondern zusätzliche Register (evtl. abblenden, wenn nicht aktuell)
- allg. gültige Elemente nur einmal außerhalb des Registers (z.B. Schließen-Schaltfläche)
- Position der Elemente auf den Registerseiten vereinheitlichen v.a. Position des ersten Elementes links oben



Word 97, Formatvorlagen, Tabulator-Schaltfläche statt weitere Registerseite

Menüs

dazu zählen Menüleisten, Symbolleisten, Kontextmenüs, Statusleiste, Taskleiste

Nachteil von Menüs: relativ schwierige und motorisch mühsame Bedienung

Vorteil von Menüs: geringer Platzbedarf

- nicht zu viele Menüs und Menüitems - 7 +/-2 Regel von George Miller (Wessel S170)
- Untermenüs: nicht zu tief verschachteln - Zugriffszeit, Feinmotorik, schwer auffindbar
- Gruppierungen: inhaltlich sinnvoll, mit Trennlinie
- Reihenfolge: nach Wichtigkeit von li nach re, nach inhaltlichen Kriterien, nach Häufigkeit
- Beschriftung: möglichst nur ein Wort, möglichst klar ("Sonstiges" nur wenn es sein muss), möglichst eindeutig, Anfangsbuchstabe groß, keine Abkürzungen
- Hotkeys: Anfangsbuchstaben bevorzugen, auf Windows-Standards achten, nicht neben Unterlängen (Unterstrich schlecht sichtbar)
- keine Geheimfunktionen und -tasten (wie z.B. "Öffnen mit" für Dateien nur bei Shift-Taste)
- aktuell nicht sinnvolle Items abblenden - nicht ausblenden, sonst Positionsveränderung, Auswahl wird nicht nur nach Inhalt vorgenommen sondern auch nach räumlichem Gedächtnis (dynamische Menüs in Office 2000 widersprechen diesem Prinzip)
- bei zu vielen (bes. thematisch zusammengehörigen) Items besser Dialogfenster oder Register als Alternative, Schaltflächen als Alternative sind deutlicher, benötigen aber ständig Platz
- berücksichtigen der Taskleiste
Höhe der Taskleiste bei Fenstergröße beachten oder Taskleiste in den Hintergrund als Vorgabe (bes. bei 800*600)

Kontextmenüs

Vorteile eben Kontext und räumliche Nähe, Nachteil rechte Maustaste nicht intuitiv - unbekanntes Wesen für viele Anwender, motorisch schwierig (abrutschen)

Symbolleisten

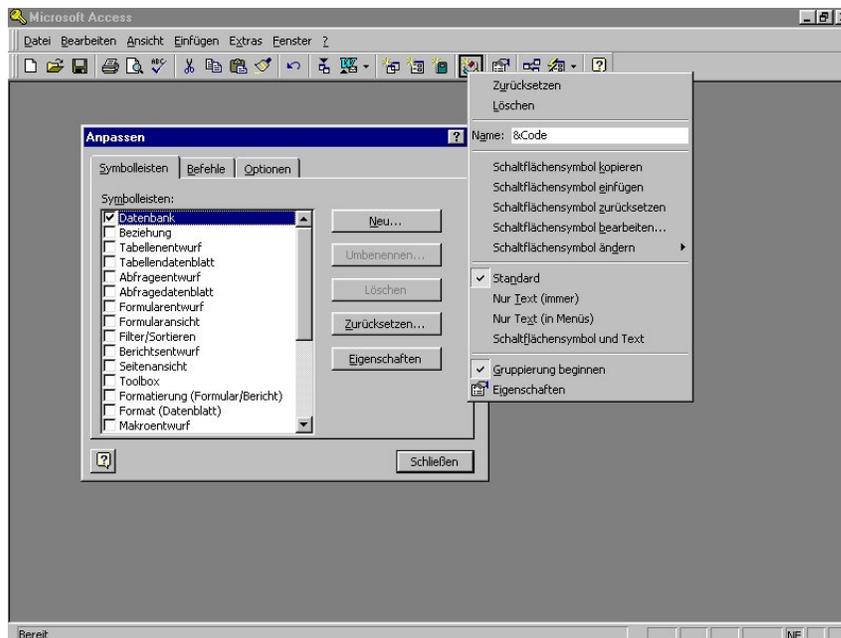
gehören wie Menüleisten zum (oft vom Anwender erwarteten) Standard in Windows sprechende Symbole verwenden, Tooltips können helfen, aber nicht Aussagekraft ersetzen, gleiche Mengen- und Aufteilungsregeln wie bei Menüs d.h. nicht mehrere Leisten, nicht zu viele Elemente, sinnvolle Gruppierung

Statusleiste

theoretisch sehr flexibel, in der Praxis kaum beachtet

Benutzerdefinierte Menüs

scheitern meist an der komplizierten und nicht intuitiven Gestaltung des "Anpassen"-Dialogs



Access 97, komplizierte, unübersichtliche Möglichkeiten, Menüs anzupassen

Menü-Test

möglichst nahe am Zielpublikum - im Idealfall der konkrete Anwender, jedenfalls nicht der Entwickler selbst

- Zählen der Häufigkeit
- Beobachtung des Anwenders
- Ratespielchen ("In welchem Menü wird sich wohl Option x befinden?").

Wenn der Anwender hilflos herumsucht, stimmt etwas am Aufbau nicht.

z.B. Formatierungsvarianten in Word: Seitengestaltung unter "Datei", Fußnoten unter "Ansicht", Absatz unter "Format", dazu noch in "Extras" Register Ansicht etc.)

Der Windows-Leitfaden von MS

MS veröffentlicht Leitfäden für die Softwaregestaltung unter Windows (s. MSPress 1995). Hier noch ein paar Auszüge aus diesem Styleguide.

MS versucht die Datenorientierung zu fördern d.h. nicht die Anwendung steht im Mittelpunkt sondern die Daten, die es zu manipulieren gilt. Stichwort: Dokument als quasi anwendungsunabhängiger Begriff

Windows-Konventionen beziehen sich auf

- Aktionen
Mausaktionen (Zeigen, Klicken, Doppelklicken, Drücken, Ziehen)
Tastaturaktionen (Drücken, Halten, Tippen, Verwendung von Zugriffstasten, Modustasten, Tastenkombinationen)
- Objekte
Desktop, Task-Leiste, Symbole, Fenster, Eingabegeräte etc.

Grundlegende Interaktionstechniken

(mit Maus, Tastatur, Stift etc.)

- Navigation
- Markieren (Kennzeichnen der Markierung, Anpassung der Markierung)
- Transferoperationen (Verschieben, Kopieren, Verknüpfen, Varianten dafür bei OLE, Drag&Drop)
- Standardoperation (Öffnen für Ordner, Klicken für Schaltflächen etc.)

Konventionen sind hier extrem wichtig:

- Wie erreicht man Öffnen, Schließen, Eigenschaften, Hilfe?
- Wie führt man Verschieben, Kopieren etc. aus?
- Was bedeutet ein Doppelklick auf einen Text oder ein Steuerelement?
- Wie erreicht man den Abbruch einer Aktion? (Esc-Taste, Abbrechen-Kontextmenüeintrag, Abbrechen-Schaltfläche)
- Wie erstelle ich ein neues Objekt? (Kopieren, Menü "Neu", von Vorlage)

Mischung von Standardoperationen und Schnellverfahren
direkte Manipulation (z.B. Drag&Drop) gegenüber indirekter Manipulation (Menüs etc.)

Fenstertypen

Die wichtigsten Konventionen betreffen natürlich Windows also Fenster und die Komponenten in und um Fenster herum.

Primärfenster: "vollwertige" Fenster mit allen üblichen Komponenten

Sekundärfenster: in der Ausstattung reduziert

Primärfenster

Konventionen gibt es z.B. für

- Symbole, Beschriftungen (Titeltext) und Schaltflächen der Titelleiste (Schließen, Minimieren etc.), Kompatibilität zu Win 3.x.
- Grundlegende Fensteroperationen
- Aktivieren und Deaktivieren
- Öffnen und Schließen
- Verschieben und Größe ändern
- Bildlauf und Bildlaufleisten
z.B. Abweichung in Access: bei mehrzeiligen Textfeldern werden ohne Fokus keine Bildlaufleisten angezeigt, was eigentlich Konvention wäre: "Die durchgängige Anzeige von Bildlaufleisten liefert eine stabilere Umgebung"
- Fenster teilen (z.B. Win-Explorer, Steuerelement Teilungsfeld = Splitter Control wie im Codefenster)
- Menüs und Symbolleisten
z.B. Abweichung in Access: im Bearbeiten-Menü vorhandene Befehle sollten auch im Kontextmenü des Objektes erscheinen - Duplizieren im Formular- und Berichts Entwurf fehlt

Dropdown und Kontextmenüs, Gestaltung von Menütiteln und Menüelementen, Symbole, überlappende Menüs (Untermenüs), Tastenkombinationen, Navigation, Anzeigen des Status etc.

Großschreibung des ersten Wortes (<> engl. Buchtitel-Prinzip d.h. jeder Wortanfang groß)

- Steuerelemente
z.B. unterschiedliche Verwendung von Optionsfeldern (für abhängige und ausschließende Auswahl = Optionsgruppe) und Kontrollkästchen (für unabhängige und nichtausschließende Auswahl).
- Standardbilder für Schaltflächen
Neu, Öffnen, Speichern, Drucken, Kopieren, Eigenschaften etc. sollten nur für diese Funktionen verwendet werden, um den Benutzer nicht zu verwirren.

Sekundärfenster

Info-, Dialog-, Eigenschaftenfenster, Standarddialoge für Öffnen, Speichern, Drucken etc.

Dienen zur Abfrage oder Anzeige zusätzlicher Informationen zu Objekten im Primärfenster, erscheinen nicht in der Taskleiste. z.B. Eigenschaftenfenster und Toolbox in Access;

Umschalten zwischen Sekundärfenstern mit Alt + F6

Gestaltung interaktiv oder modal möglich (bezogen auf Anwendung oder System)

Meldungsfenster

ist ein Sekundärfenster mit Information zu einer bestimmten Situation oder Bedingung. Am besten solche Situationen von vornherein vermeiden durch vorherige Prüfung und entsprechende Reaktion im Programm. Wichtig ist die verständliche Formulierung des Meldungstextes.

MS wagt es zu schreiben: "Der Text in einem Meldungsfeld sollte klar und knapp sein und Begriffe enthalten, die der Benutzer ohne weiteres versteht. Verzichten Sie daher auf Fachbegriffe und systemorientierte Informationen" (MSPress 1995, S 224)

SDI und MDI

SDI = Single Document Interface

Programm besteht aus einem einzigen Hauptfenster, dem evtl. einige Unterfenster (Sekundärfenster) zugeordnet sind. Eintrag in die Task-Leiste.

Vorteil: einfache Handhabung bez. Öffnen, Schließen, Speichern, Überblick etc.

MDI = Multiple Document Interface

besteht aus einem übergeordneten Anwendungsfenster (Parent-Fenster) und mehreren Dokumentenfestern (Child-Fenster), die diesem untergeordnet sind. Child-Fenster verwenden versch. Elemente des Parent-Fensters z.B. Menü-, Werkzeug- und Statusleiste.

abweichende Konventionen gibt's bei Child-Fenstern z.B. für

- das Systemmenü (enthält "Schließen" statt "Beenden" etc.)
- Verschieben (nur innerhalb des Parent-Fensters)
- Vergrößern und Verkleinern
- Vollbildmodus oder Wiederherstellen gilt immer für alle Child-Fenster, Verkleinern nicht auf Symbolgröße sondern mit Titelzeile. (z.B. Formularfenster in Access 97)

SDI und MDI sind keine absolut vorgegebenen Stati sondern Richtlinien, aus denen man Mischformen bilden kann:

- **Arbeitsumgebungen** bilden Alltagssituationen ab z.B. Schreibtischoberfläche oder Büro, ähnlich MDI aber z.B. lassen sich Objekte in Ordnern abbilden
- **Arbeitsmappen** bilden Objekte nicht als einzelne Child-Fenster ab, sondern als Register im Hauptfenster der Arbeitsmappe (Excel)
- **Projekte** Verbindung mehrerer Objekte und deren Fenster, ohne diese Fenster zu enthalten, jedes Objekt erhält einen Eintrag in die Taskleiste

Kombinationen von Fenstermodellen sind zulässig solange sie konsistent und verständlich sind.

Die Auswahl des Fenstermodells sollte sich an den den Bedürfnissen, Kenntnissen und v.a. der Computererfahrung der Benutzer ebenso orientieren wie an den Erfordernissen der Anwendung (effektive Bildschirmausnutzung, datenorientiertes Design etc.)

Weitere Richtlinien

Entwurfszyklus

lt. Windows-Styleguide: Entwurf – Prototyp - Test

Wiederholung der 3 Phasen mit schrittweiser Verfeinerung
In der Theorie schön - in der Praxis für kleinere Firmen oder einzelne Entwickler kaum durchführbar.

Installation und Registrierung

Es gibt Richtlinien für die Installation von Anwendungen

Interessante Auszüge:

"Vermeiden Sie, bei der Installation Ihrer Software Dateien in das Windows-Verzeichnis oder in das Unterverzeichnis "System" zu kopieren, denn damit können Sie in Null Komma nichts Ihr System verbauen, was schließlich zu Leistungseinbußen oder gar zu unerklärlichen Abstürzen von Windows 95 führen kann." (MSPress 1995, S 264)

Übrigens gibt es auch eine Richtlinie bez. Deinstallation, Eintrag dafür in die Registry etc. (spaßig in Hinsicht auf die Access-Runtime)

Ebenso gibt es Richtlinien für Einträge von Applikationen in die Registry. Statusinfo (Version, RunOnce bei Absturz etc.), Suchpfad der Anwendung, Dateinamenerweiterung, Symbole, Befehle usw.

Eingabehilfen

Darauf achten, ob evtl. Behinderte (temporär, permanent) damit arbeiten müssen. Abgesehen davon einige Normen und Grundregeln einhalten, um sie grundsätzlich zu fördern. Kann auch dem "Normanwender" nutzen z.B. Deutlichkeit, mehrkanalige Hinweise

Sehbehinderungen

Aufpassen bei Farbkombinationen, deutlicher Fokus etc. spezielle Regeln für schwer Sehbehinderte (Tauglichkeit für Bildschirmlesegeräte)

Hörbehinderungen, Behinderungen der Bewegungsfähigkeit, Sprachstörungen, kognitive Beeinträchtigungen etc.

Grundsätzlich gilt, möglichst viele Windows-Systemkonforme Einstellungen zu übernehmen, weil Windows in der Hinsicht einiges bietet, und Tests mit Betroffenen während der Software-Entwicklung.

Internationalisierung

auch "Lokalisierung" = Anpassung an ein anderes Land

Bereits in frühen Phasen der Entwicklung vorbereiten - im Nachhinein nur aufwendig zu machen. Betrifft Bildschirmausgabe, Dokumentation, Hilfedateien, Vorlagen, Beispiele etc.

Sprachanpassung, versch. Tastaturlayouts, Zeichensätze und Formate (Datum, Maßeinheiten, Telefonnummern etc.), kulturelle Unterschiede (z.B. Symbole)

Texte

verschiedene Längen in versch. Sprachen (z.B. englisch ca. 30% kürzer als deutsch)

klare hochsprachliche Ausdrücke

Texte nicht in Bilder einfügen

Literatur, Links

MS Press

Die Windows-Oberfläche

Leitfaden zur Softwaregestaltung. Für Windows 95 und Windows NT.
MS Press, 1995, ISBN 3-86063-226-4 (DM 69,-)

amerikan. Originalausgabe:

MS Press

The Windows Interface Guidelines for Software Design

MS Press, 1995, ISBN 1-55615-679-0

Ivo Wessel

GUI-Design

Richtlinien zur Gestaltung ergonomischer Windows-Applikationen
Hanser, 1998, ISBN 3-446-19389-8 (DM 69,-)

Alan Cooper

About Face

The Essentials of User Interface Design
IDG Books 1995, ISBN 1-56884-322-4

Rebecca M. Riordan

Designing Relational Database Systems

MS Press, 1999, ISBN 0-7356-0634-X

DIN-Normen bei

<http://www.din.de>

und

<http://www.beuth.de/beuth.html?normenrecherche>

umfangreiche Webseite über alle Aspekte des GUI-Designs
(Gesellschaft Arbeit und Ergonomie – online e.V.)

http://www.sozialnetz-hessen.de/ergo-online/ergo_frame1.htm

Hall of Fame, Hall of Shame

<http://www.iarchitect.com>