

# HCI-11-Klassische Software-Ergonomie

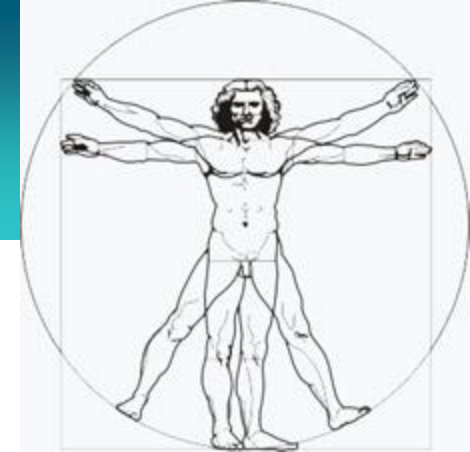
## ➤ Die "WIMP"-Regeln



# Das WIMP-Paradigma

- klassische direkte Manipulation benötigt ein "Zeigegerät" (Pointing Device)
- Daher
  - W windows
  - I icons
  - M menus
  - P pointing device
- Heute steht "P" nicht nur für "Maus"...
- Bedienelemente hießen "Widgets" – window gadgets. Heute stärker spezialisiert:
  - Prozesse mit Miniaturansicht

# Effizienz von Eingaben



- **Fitt's Law**
  - Aufwand der **Mauspositionierung** hängt ab von **Mausweg** und **Größe des Ziels**
- **Hick's Law**
  - Aufwand einer **Auswahl** hängt von der Zahl der **Entscheidungsschritte** und der Zahl der **Alternativen**
  - Krug argumentiert, dass Entscheidungen **ohne kognitive Leistung** den Aufwand nicht erhöhen.
- **GOMS-Aufwandsschätzung**
  - *GOMS -Model*: Goals, Operators, Methods, and Selection Rules → vgl. 6 Ebenen

Abkürzung	Zeit / Sekunden
K (=Keyboard, Tastatureingabe, Mausklick)	0,2
P (=Pointing; Mauspositionierung)	1,1
H (Homeing; Wechsel Maus-Tastatur)	0,4
M (Mentally prepare; Mentale Reaktionszeit)	1,35

Tabelle 15 : GOMS-Modell, Zeiten für einzelne Aktionen.

# Klassische Interaktionselemente

- Fenster
  - Haupt, Unter- und Dialogfenster
- Menüs
  - Drop-down und Pop-up-Menüs
- Widgets
  - Knöpfe ("Schaltflächen" lt. MS-Terminologie)
  - Auswahllisten
  - Deiktische Werteingaben
- Meldungen und Warnungen
- About-Boxen und Splash Screens
- Werkzeugleisten
- Hilfemenüs

# "Klassische Benutzerschnittstellen"

## ■ Klassische GUIs

- GUIs mit klassischen Interaktionselementen
- WIMP-Paradigma
- *Wie setzt man die Interaktionselemente ergonomisch ein?*
- "Sonderform" Browser-GUIs

## ■ Kommandosprachen

- Sprachen für die vollständige Kommandosteuerung
- Sprachen für die Steuerung von Teilaspekten
- Sprachen als Alternativen / Shortcuts zu Zeigehandlungen
- Aktuelles Seitenthema: DSLs (Domain Specific Languages)
- *Wie entwirft man eine benutzerfreundliche Kommandosprache?*

# Anwendungsfenster

## ■ Haupt- und Unterfenster

- Unterfenster fest in Hauptfenster integriert (Single Document Interface **SDI**)
- Unterfenster selbständige Einheit, Koexistenz von Unterfenstern (Multiple Document Interface **MDI**)
- Unterfenster als Registerkarten des Hauptfensters (Tabbed Document Interface, **TDI**)
- Fenster kann aus Teilfenstern (**Panes**) bestehen, mit festen oder verschieblichen Teilern (Multipaned Window)
- Unterfenster werden **als Folge von Interaktionen geöffnet**. Im TDI kann der Benutzer frei zwischen geöffneten Fenstern navigieren. Im SDI kann der Benutzer zusätzlich einzelne Fenster schließen.

## ■ Dialogfenster

- modal und nicht-modal
- Schließen explizit durch Interaktion
- bei modalen Fenstern **keine Navigation** möglich

# Webbrowser-Fenster

- Deutlich mehr graphische / multimediale Gestaltungsmöglichkeiten
- Ursprünglich völlig andere Struktur
  - jeweils ein Fenster
  - Unterfenster entweder als Teilbereich (Frame)
  - oder als zusätzliches Browserfenster
  - Modalität nicht möglich
  - Interaktion fensterweise (OK-Knopf)
  - Navigation zwischen Fenstern einer Anwendung direkt unterstützt (Navigationsleiste, Seitemap)
  - Menüs und Tabs wie Anwendungsfenster – aber nur zur Navigation
  - Toolbars weniger üblich
  - Individualisierbarkeit nur ausnahmsweise
- Heute weniger scharfe Unterscheidung
  - Interaktionsform "aufgeweicht" durch Ajax
  - aber andere "Kultur"
  - ggf. Einbettung einer "klassischen GUI" über Applet oder Active-X

# Widgets

- Knöpfe
  - Aktions- und Wechselknöpfe (Toggles)
  - Knopffelder
- Radio-Buttons und Checkboxes
- Auswahllisten
  - statisch oder dynamisch
  - mit / ohne Eingabemöglichkeit
- Deiktische Werteingaben
  - Slider (auch Scrollbars)
  - Kalender
  - Up-Downs mit / ohne Eingabemöglichkeit
- Texteingabe-Widgets
  - Einzeilig / mehrzeilig
  - mit / ohne Formatieroption
  - mit / ohne Autocomplete

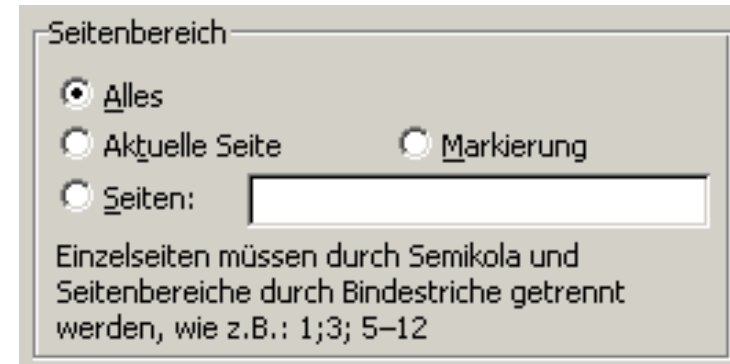


# Anordnung der Bedienelemente (auch Controls oder Widgets)

- Grundsatz der Strukturiertheit
- Grundregeln aus ISO 9241-12:
  - Klarheit
  - Unterscheidbarkeit
  - Kompaktheit
  - Konsistenz
  - Erkennbarkeit
  - Lesbarkeit
  - Verständlichkeit
- Einige spezielle Regeln für bestimmte Bedienelemente:

# Radiobuttons und Checkboxes

- Text hinter dem Element
  - kein Hyperlink!
- Anordnung vertikal
  - ggf. als Tabelle
- Vorauswahl
  - Radiobutton: immer
  - Checkbox: wo sinnvoll



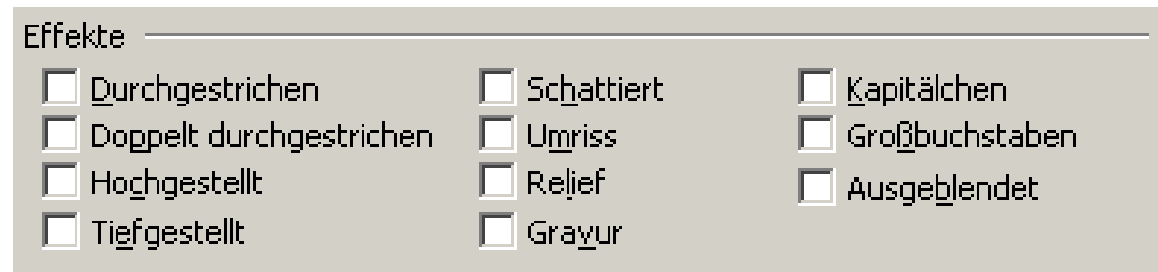
Seitenbereich

Alles

Aktuelle Seite  Markierung

Seiten:

Einzelseiten müssen durch Semikola und  
Seitenbereiche durch Bindestriche getrennt  
werden, wie z.B.: 1;3; 5-12

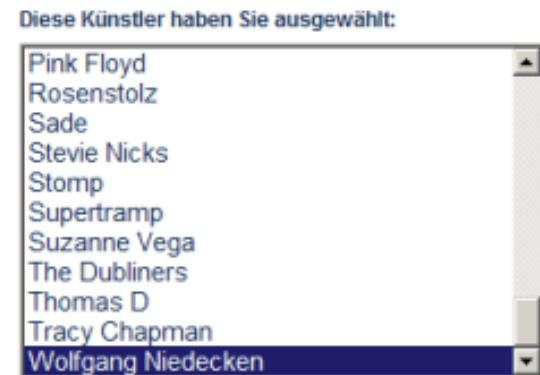
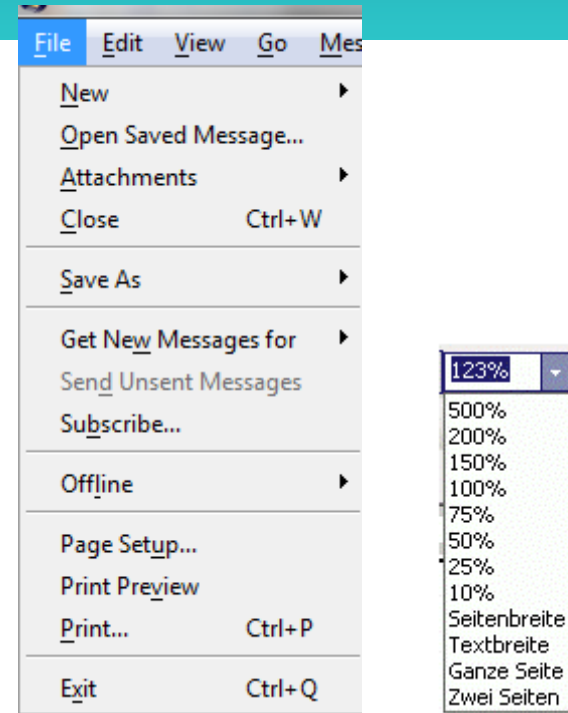


Effekte

<input type="checkbox"/> <u>D</u> urchgestrichen	<input type="checkbox"/> <u>S</u> chattiert	<input type="checkbox"/> <u>K</u> apitälchen
<input type="checkbox"/> <u>D</u> oppelt durchgestrichen	<input type="checkbox"/> <u>U</u> mriss	<input type="checkbox"/> <u>G</u> roßbuchstaben
<input type="checkbox"/> <u>H</u> ochgestellt	<input type="checkbox"/> <u>R</u> elief	<input type="checkbox"/> <u>A</u> sgeblendet
<input type="checkbox"/> <u>T</u> iefgestellt	<input type="checkbox"/> <u>G</u> rayur	

# Dropdownlisten und Auswahllisten

- **Anzahl der Elemente**
  - Drop-down-Liste: max 15
  - Auswahlliste: scrollbar, flexibel
- **Vorauswahl**
  - entweder Vorauswahl
  - oder Eingabefeld (v.a. Auswahllisten)
- **Mehrfachauswahl**
  - vermeiden!
- **Verwendung**
  - Drop-down eher Befehle oder Navigation
  - Auswahlliste eher Datenauswahl



# Eingabefelder

- Linksbündig
- Beschriftung Linksbündig über dem Feld
- lang genug!
- mehrzeilige Eingabe  
-felder immer groß genug!

Ihre Nachricht an uns:

Ihr Name:

Ihre E-Mail Adresse:

Betreff:

Ihre Mitteilung:

Senden

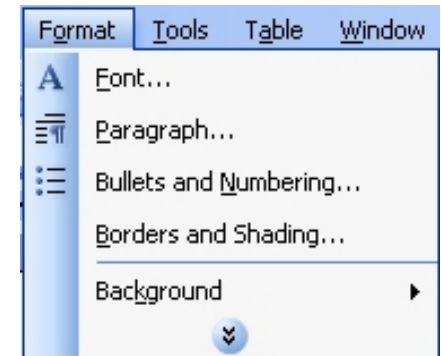
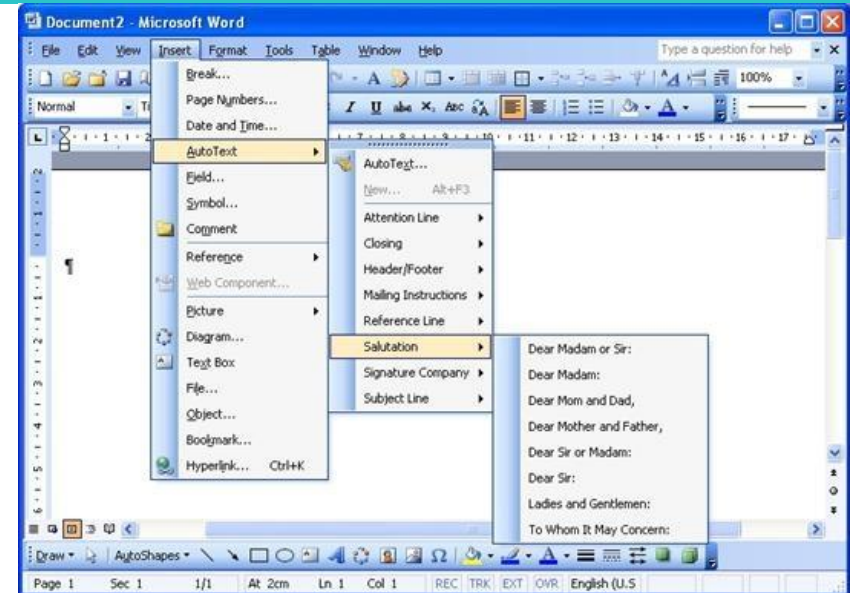
# Knöpfe ("Schaltflächen")

- Möglichst vorbelegt
- wiederkehrende Knöpfe an gleichbleibender Position
- Feedback der Interaktion "drücken"
- Wechselknöpfe (Toggle Buttons) eindeutig beschriftet!



# Menüs

- Drop-down-Menüs
  - mit Anker in der Menüleiste
- Pop-up-Menüs (Kontextmenüs)
- Gestaltungsalternativen
  - flache oder kaskadierte Menüs (max. 2 Untermenüs!)
  - gruppierte Menüoptionen
  - Icons als Menüoptionen
  - Selektion durch mnemonische Tastenkürzel
  - Selektion durch Tastenkombinationen
  - stufenweises Ausklappen

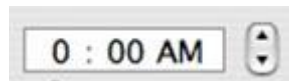
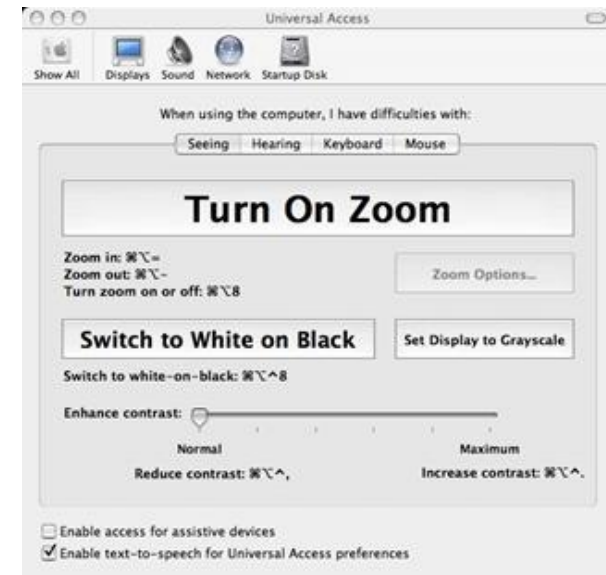


# Ergonomie-Regeln für Menüs

- Kaskadierung vermeiden
- Gruppierung nach Chunking-Regeln
- Alle Shortcut-Möglichkeiten anbieten
  - Tastenkombinationen und Shortcuts möglichst ähnlich
- wichtige Menüoptionen zusätzlich als Werkzeuge anbieten
  - als Icons im Menü darstellen
- Nicht verfügbare Optionen ausgrauen
- Individualisierung *rücksetzbar* ermöglichen:
  - Definition von Shortcuts
  - Auslagerung in die Werkzeugleiste

# Weitere Ergonomie-Regeln für Widgets

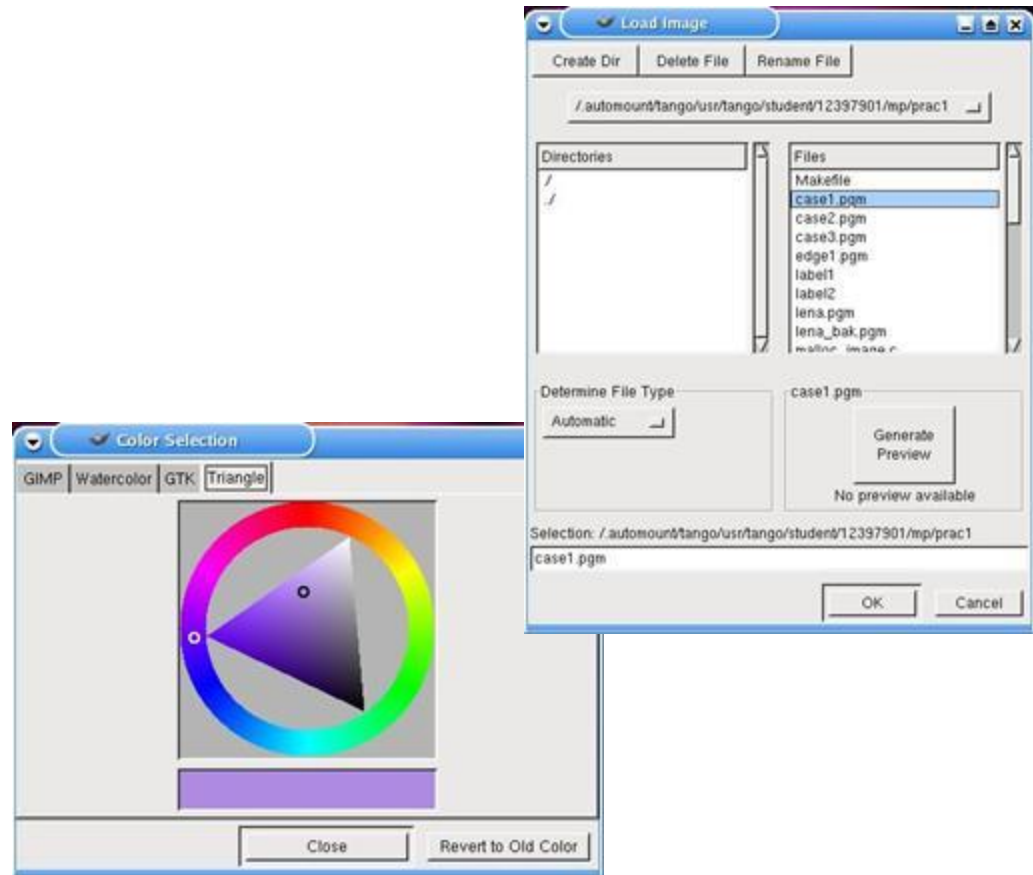
- Allgemein
  - Gruppierung, Chunking
  - Ausrichtung
- Toggles:
  - Beschriftung muss Zustand und Folgezustand ausweisen,
  - Ersatz durch Radio-Buttons, wo möglich (Platz)
- Werteingaben:
  - Affordanz!
- Texteingaben (auch in Auswahllisten):
  - Autocomplete wo möglich!





# Meldungen und Warnungen

- als Dialogfenster
  - zumeist modal
  - Kaskadierung möglich
  - Platzierung
  - verschiebbar oder fix



# Ergonomie-Regeln für Dialogfenster

- Einheitliche Gestaltung
- Minimaler, unmittelbar verständlicher Text (Stresssituation)
- Kaskadierung vermeiden
  - zumeist ein Designfehler: zu komplizierte Befehlsstruktur
- verschiebbar
  - damit Kontext lesbar bleibt
- möglichst nur applikations-modal
  - nicht system-modal!

# Ergonomie-Regeln für Warnungs- und Meldungsfenster

- Warnungs- und Meldungsfenster unterbrechen den Arbeitsablauf.

- **Stress!**

- Text kurz und verständlich
- Text informativ
- Ggf. Hinweis auf weitere Infos

- **Vermeidbar?**

- durchaus für Erfolgs- oder neutrale Meldungen
- modale Meldung unterbricht, selten gerechtfertigt!



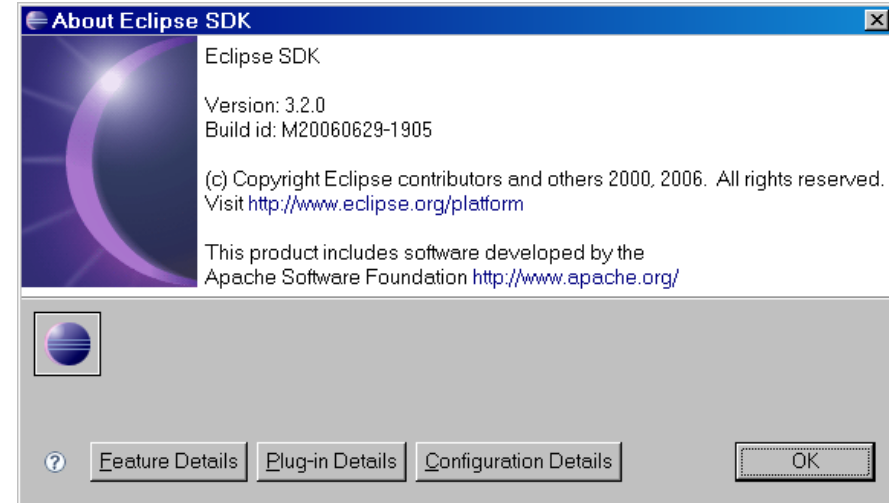
# Ergonomie-Regeln für Warnungs- und Meldungsfenster

- Text in Meldungsfenstern:
  - kurze, informative Meldungen
  - Problem klar benennen, Lösung anbieten  
*"Nicht genügend Speicherplatz auf C:/. Unter F:/C\_Extension ablegen?"*
  - Eindeutige Fragen stellen, nicht:  
*"Die vorhandene Datei wird überschrieben. Abbrechen?" – OK*
  - Respektvolle Meldungen  
*"Error 722" ist nutzlos und einschüchternd*  
*Technischen Insider-Jargon vermeiden*
  - Standardformulierungen und –Icons erleichtern die Bewertung
  - Immer einen vernünftigen Ausweg anbieten  
*"Abbruch", "Programm beenden", "Wert ignorieren" o.dgl.*

# Splash Screen und About-Box

## ■ About Box

- übliche und erwartete Informationsquelle über Programm, Version, Lizenzbedingungen und Support.



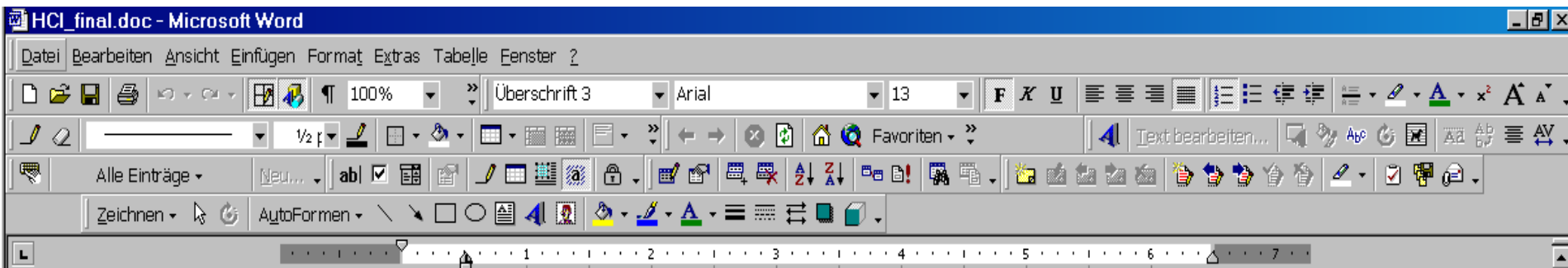
## ■ Splash Screen



- übliches Dialogfenster während der Ladezeit einer Anwendung
- enthält minimale Produktinformation
- nützliche Anzeige, dass das Programm gestartet wird
- **muss ohne Konsequenzen abschalt- oder schließbar sein.**

# Werkzeuggeste

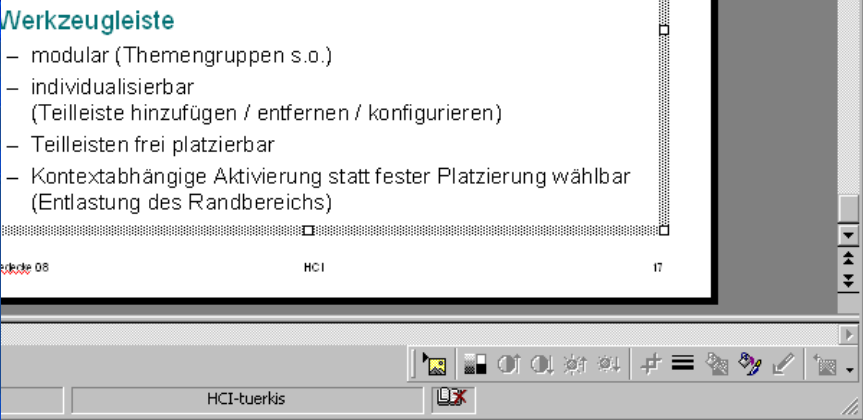
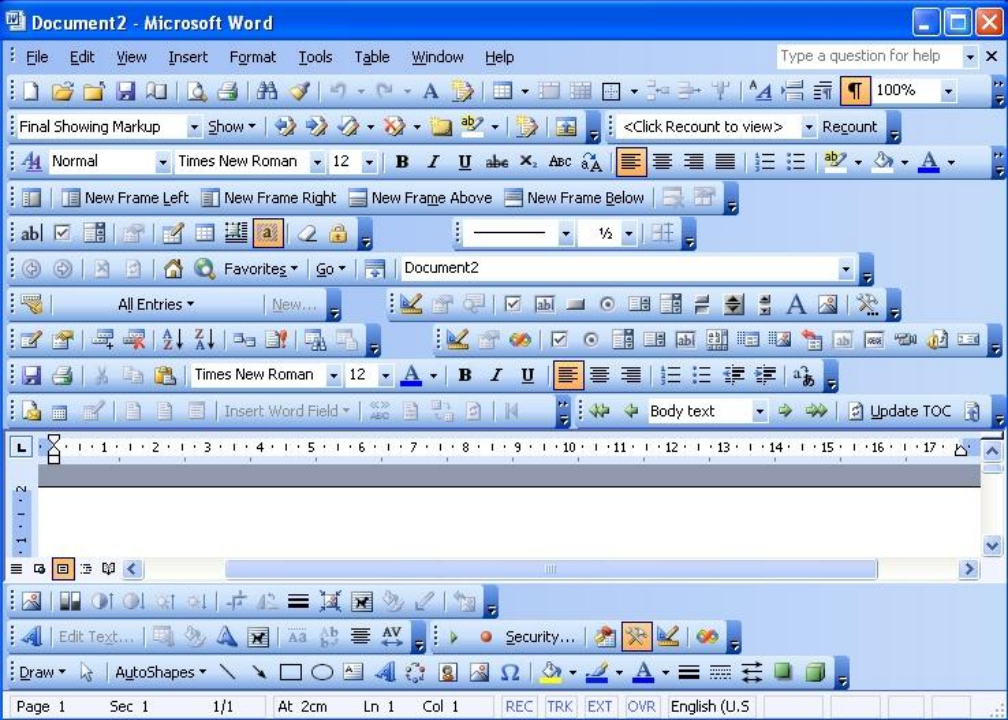
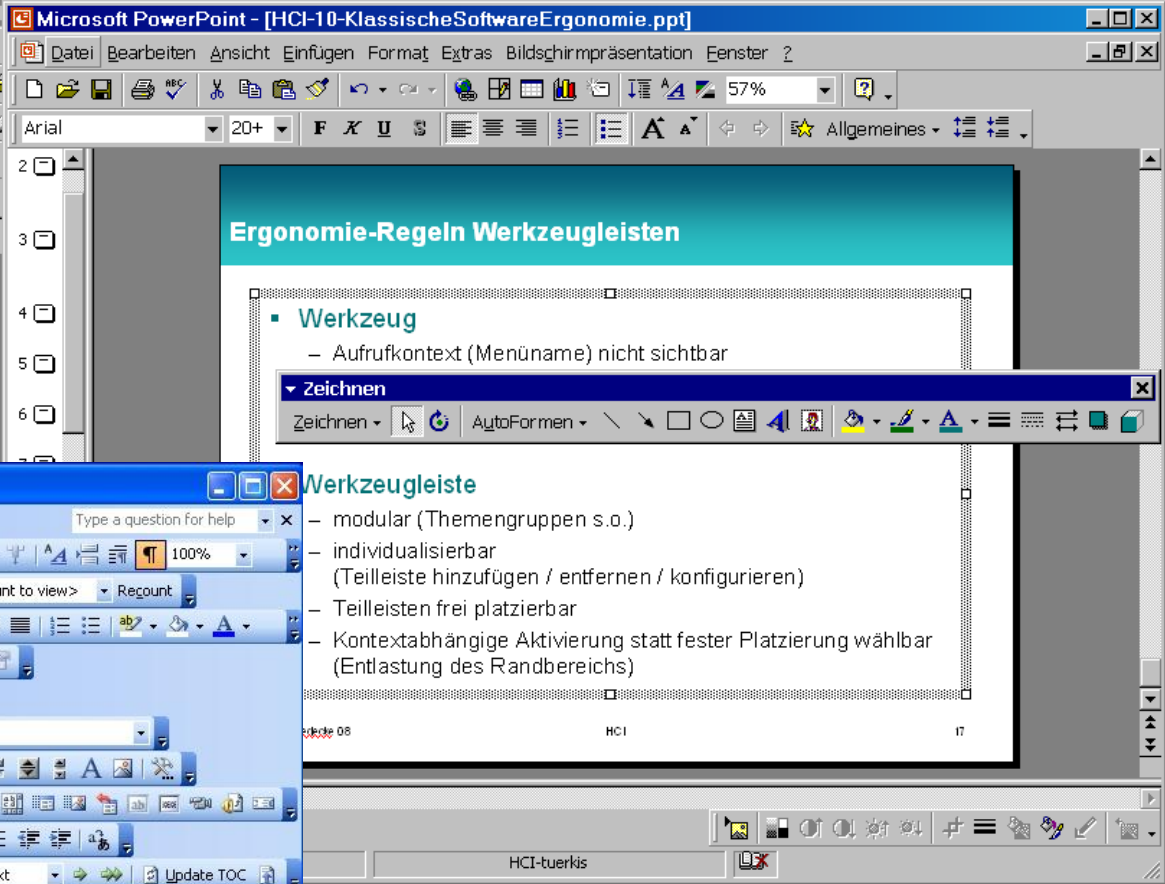
- **Werkzeug**
  - Kommando oder nichtmodaler Dialog
  - Aufrufbar über Werkzeug-Icon
- **Werkzeuggeste**
  - Auswahl wichtiger Funktionen aus Menüs (Menü: systematische, vollständige Funktionsliste)
  - dauerhaft am Anwendungsrand platziert



# Ergonomie-Regeln Werkzeugleisten

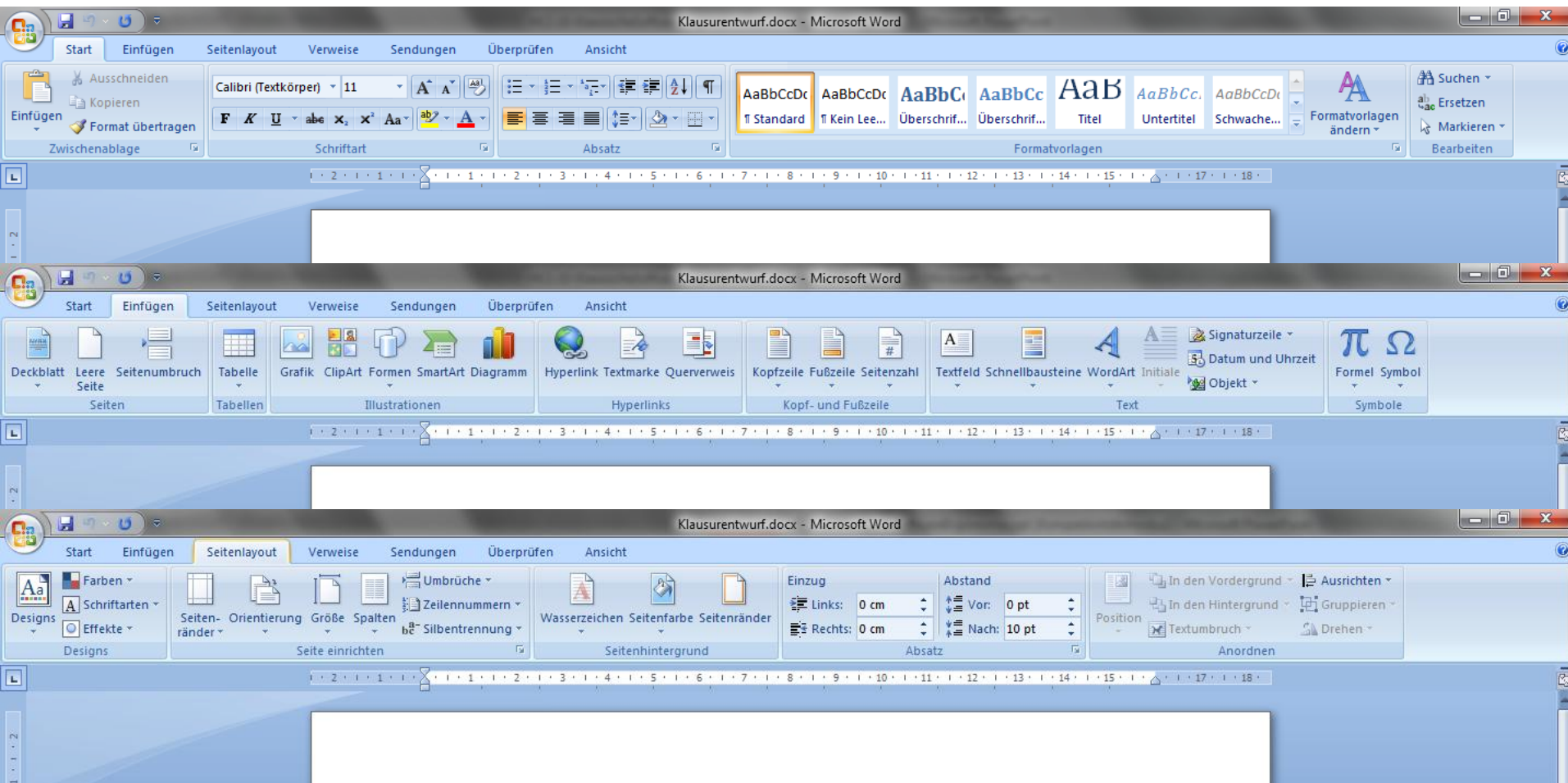
- **Werkzeug**
  - Aufrufkontext (Menüname) nicht sichtbar
  - → ausgegraut, falls nicht anwendbar
  - → gruppiert in Themengruppen, am besten an Menüs orientiert
  
- **Werkzeuggeste**
  - modular (Themengruppen s.o.)
  - individualisierbar  
(Teilleiste hinzufügen / entfernen / konfigurieren)
  - Teilleisten frei platzierbar
  - Kontextabhängige Aktivierung statt fester Platzierung wählbar  
(Entlastung des Randbereichs)

# Ergonomie-Regeln Werkzeugleisten





# Neue Werkzeugleistengestaltung in MS Office 2007



- Werkzeugleiste durch Register gruppiert
- Innerhalb des Registers Untergruppen von Werkzeugen

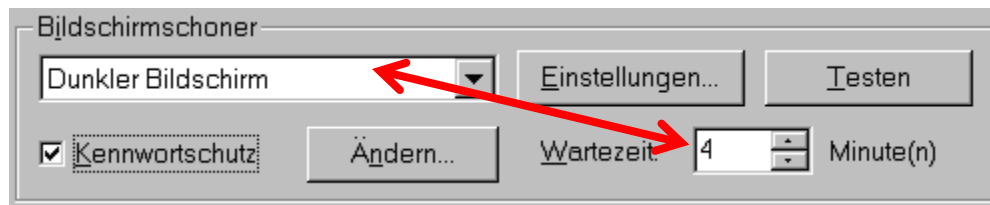
# Gruppierung: Strukturblöcke

- Inhaltlich zusammengehörige Bereiche einer GUI
- **Optisch begrenzt und hervorgehoben**
- Wiederkehrende Hauptstrukturierung

→ vgl. "Portlets"

# Gruppierung: Anordnung von Informationen

- Gesetz der Nähe für inhaltliche Zusammenhänge nutzen



# Hilfe-Systeme

- **Hilfemenü**
  - Handbuchsatz
  - Index
  - Suche
  - Themenliste
  - Verweis auf weitere Hilfen
- **Kontextsensitive Hilfe**
  - Auf den Kontext bezogene Auswahl aus den Hilfethemen
  - Per Pop-up-Menü aktivierbar
- **Assistent**
  - Initiative (selbst-aktivierte) kontextsensitive Hilfe
- **Tutorial**
  - Anwendungsunabhängige allgemeine Einführung
  - meist mit Beispielanwendung

- **Hilfethemen: Hypertext!**
  - kleine Einheiten, ggf. verkettet
  - Bezeichnung im Index jederzeit sichtbar
  - gut lesbar, ansprechend (Benutzung in Stresssituation)
  - jede Einheit für sich verständlich
  - Fachbegriffe und Hilfethemen verlinkt
  - klare Sprache
  - Screenshots von Bedienung und/oder Ergebnis
  - Wo möglich Kontextbezug
  - Navigationshilfen innerhalb der Hilfe (Vor / Zurück / Weiter...)
  - Hinweis auf weiterführende Hilfen (Tutorialkapitel, Website)

# Ergonomie-Regeln für weitere Hilfen

## ■ Index, Suche

- Wildcard-Suche
- Synonyme und ähnliche Begriffe berücksichtigen
- möglichst als lernendes System

## ■ Suchfragen

- Je nach Güte der Analyse nützlich
- **Immer Verweis auf Index**  
da Fragen oft nicht oder falsch verstanden werden
- Suche durch Beispiele

## ■ Tutorial

- Erste Schritte – Durcharbeitung unter 1 Stunde
- Ausführliches Tutorial mit allen Features
- immer mit Index zwecks "Seiteneinstieg"

# **Soweit der Überblick über die "WIMP-Ergonomie"**



**Ehe wir über die Integration von HCI  
in den Software-Entwicklungsprozess sprechen,  
müssen wir uns noch mit  
Gesetzen und Normen befassen.**