

Techniken der Projektentwicklung

Interface Design

Ingo Lütkebohle

Termin 12

Einordnung

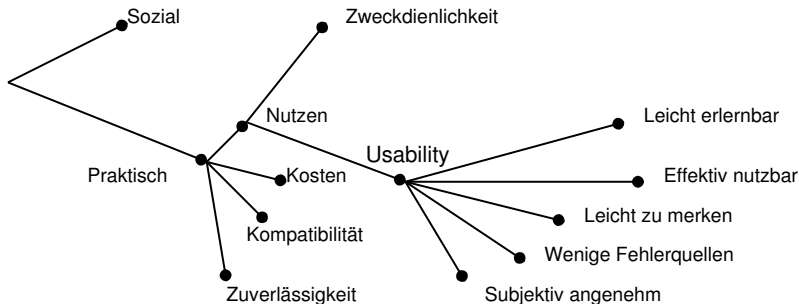
Was bisher war

- GUI-Architektur
- Katalog von UI-Elementen
- Use Cases mit festgelegten Zielen

Themen heute

- Anforderungen an *benutzbare* GUIs (*Usability*)
- Tips um Usability zu erreichen
- Gestaltung von Oberflächen
- Empirisches Testen von GUIs (*Usability Engineering*)

Kriterien für akzeptable Software



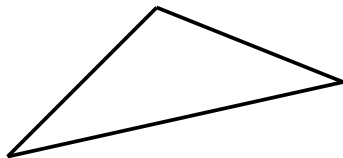
(Nielsen 1993, S. 25)

Was ist Usability?

- Leicht erlernbar
Wie schnell können Neulinge produktiv arbeiten?
- Effektiv nutzbar
Wie lange dauert die Erledigung von Aufgaben?
- Leicht zu merken
Ist der Wiedererkennungseffekt der Oberfläche hoch?
- Wenige Fehlerquellen
Wie viele Fehler kommen vor und wie leicht lassen sie sich rückgängig machen?
- Subjektiv angenehm
Arbeiten die Anwender gerne mit dem System?

Interdisziplinarität von Interaction Design

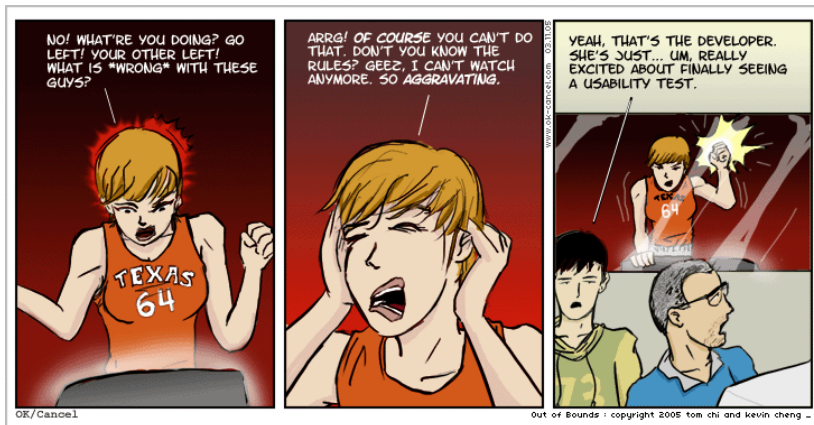
Human-Computer-Interaction
Wissen, Kognition, Interaktion



Kunst
Ästhetik,
Affekt,
Emotion

Interface Design
Ein-/Ausgabe, User Interface

Wieso nicht einfach auf Entwickler-Know-How vertrauen?



- Handlungspotential der Benutzer ist unbegrenzt
- *Annahmen* der Entwickler sind nicht gut genug!

Aus <http://www.ok-cancel.com/comic/78.html>

Ein möglicher Designprozeß

- Verstehen
das *wirkliche* Problem erfassen, User Tests
- Abstrahieren
Hauptelemente, was ist wichtig
- Strukturieren
Beziehungen, Ordnungen, Prioritäten
- Repräsentieren
bildlich/klanglich, UI-Skizze, Prototyping
- Details ausarbeiten
Farben, Fonts, Icon-Stile, Übergänge

Iterativ wiederholen!

Heuristiken

Ein paar allgemeine Hinweise zu benutzbaren Programmen.

- Kontext des Benutzers mit einbeziehen:
 - ① Aufgabenorientiertes Design (use Use Cases!)
 - ② Performanceerwartungen
- Oberflächen konsistent (nicht unbedingt exakt gleich)
- unmittelbares Feedback wichtig, gerade bei längeren Tasks
- Fehlerquellen identifizieren und *entfernen*

Sprache des Anwenders sprechen

- Begriffe der Domäne verwenden
- Texte aus der Sicht des Anwenders schreiben
- Keine doppelten Verneinungen
- Weniger ist mehr
- Entwickler-Jargon ersetzen
 - 1 mögliche Ersetzungen aufstellen
 - 2 Anwender abstimmen lassen

Feedback durch das Programm

- Muß alle nötige Info enthalten
z.B. Name der zu löschenden Datei
- Aktionsbeginn signalisieren
z.B. Button wird „eingedrückt“
- Verlaufsbenachrichtigung für längere Aktionen
z.B. Fortschrittsbalken
- Persistenz
 - niedrig verschwindet von selbst
 - mittel verlangt Bestätigung
 - hoch Verbleibt in der UI

Obergrenzen für interaktives Verhalten

- 0.1s Unmittelbar
- 1.0s Merkleich aber Arbeitsfluß bleibt erhalten
- 10s Aufmerksamkeit wandert ab

Fehler und ihre Behandlung

Vermeidung

- Ungültige Aktionen nicht verwendbar
- Wo möglich Auswahl statt freier Eingabe
- Eingaben unmittelbar validieren und Feedback einblenden

Behandlung

- Klare, kurze, höfliche Meldungen
- Lösungshinweise anbieten
- Undo durchgängig anbieten

Design der Benutzerschnittstelle

Die Benutzerschnittstelle...

- ... *ist* für den Anwender das Programm
- ... *vermittelt* konzeptionelles Modell des Programms
- ... *repräsentiert* Systemzustand nach Aussen

Ihre Gestaltung unterscheidet *funktionierende* von *benutzbaren* Programmen.

Gestaltbare Bereiche

- **Verwendete physische Geräte**
(Tastatur, Touchscreen, Soundkarte, Projektor, ...)
- **Visuelle Darstellung von Information**
(grafisch, textuell, farbig, bewegt, ...)
- **Eingabeelemente**
(Anordnung, Auswahl)
- **Konzeptionelle Analogien/Abstraktionen**
(Datei, Papierkorb, Hyperlink, Blog, ...)

Gestaltbare Bereiche

- **Verwendete physische Geräte**
(Tastatur, Touchscreen, Soundkarte, Projektor, ...)
- **Visuelle Darstellung von Information**
(grafisch, textuell, farbig, bewegt, ...)
- **Eingabeelemente**
(Anordnung, Auswahl)
- **Konzeptionelle Analogien/Abstraktionen**
(Datei, Papierkorb, Hyperlink, Blog, ...)

Grafikdesign

- Gutes Design essentiell
- Grafikdesigner hinzuziehen
- Heuristiken:
 - An vorhandenes UI-Wissen anknüpfen
 - Gestaltregeln zur Anordnung
 - Weniger ist mehr

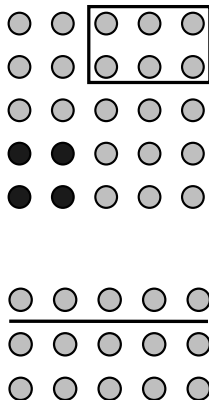
Anordnung von Elementen

- Leserichtung folgen
im westlichen Kulturraum: links→rechts, oben→unten
- **Platz** lassen
Gruppierung, Absetzung
- Standards folgen
Menüposition, Buttonposition in Dialogen, etc.
- Mit skizzenhaften Prototypen testen

Gestaltregeln

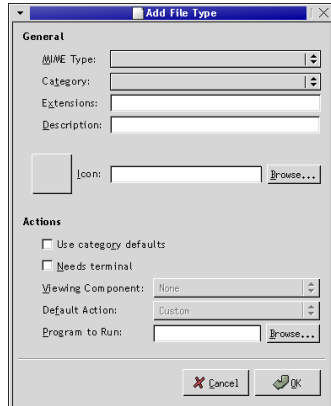
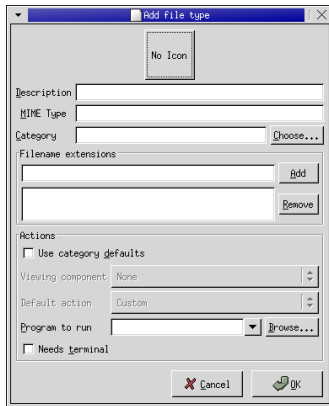
Zusammengehörigkeit wird gefördert durch

- Räumliche Nähe
- Umschlossen von Linien / Rahmen
- Gleiche Bewegung / Änderung
- Gleiche Farbe, Form, Größe, Schrift



Heuristiken zur Anordnung von Information

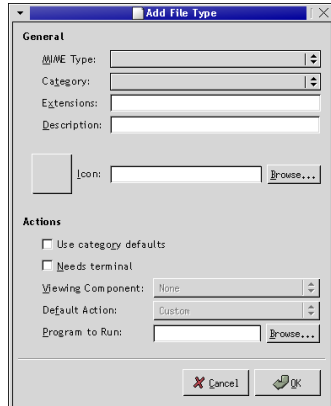
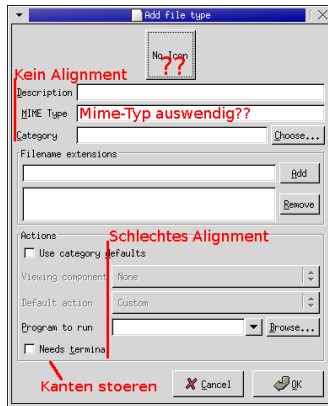
- Überblick durch Ausrichtung
- Logische zu räumlicher Gruppierung



Beide Dialoge sind überladen!

Heuristiken zur Anordnung von Information

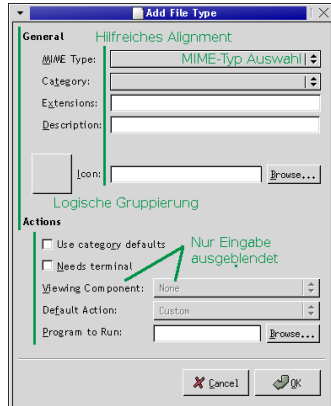
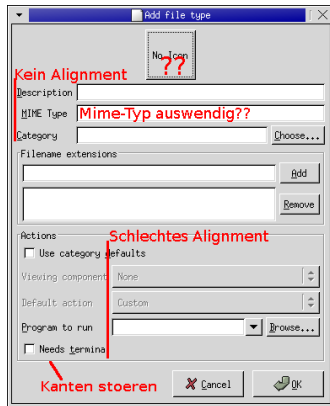
- Überblick durch Ausrichtung
- Logische zu räumlicher Gruppierung



Beide Dialoge sind überladen!

Heuristiken zur Anordnung von Information

- Überblick durch Ausrichtung
- Logische zu räumlicher Gruppierung



Beide Dialoge sind überladen!

Erklärungen in Dialogen

Configure SingleValue plugin 'SingleValueDisplay-26G'

A simple plugin displaying a single double value as bar graph in a coordinate system.


Lower and upperlimit specify the coordinate (and value) limits. The Color is adjustable.




Lower limit:

Upper limit:

Opacity: %

 Color ...

 Update

 Close

- Erläuterung maximal nah
- Visuell unaufdringlich getrennt

Farbe

- Erreicht viel Aufmerksamkeit
- Sparsam einsetzen
Inhalt hervorheben, nicht „bunt sein“
- Geringe Kontraste sind oft ausreichend
z.B. zur visuellen Abtrennung
- Konsistent einsetzen

Usability Engineering

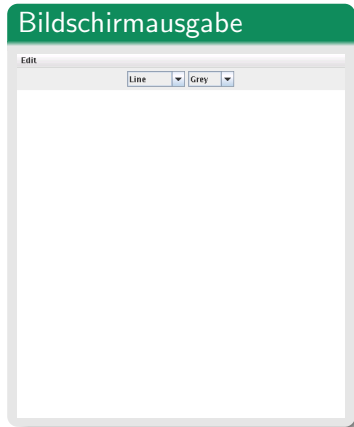
- Nur Rückmeldungen von Benutzern führen zu guten Programmen
- Wichtigstes Instrument: **Benutzerstudien**
- **Beobachtung** von Testpersonen, insbesondere unter Verwendung von:
 - **Scenarios** Beobachtung bei Ausführung eines bestimmten Use Case
 - **Laut denken** Benutzer beschreibt, was er/sie tut
 - **Zeitmessung** Zeit ermitteln, die Ausführung einer bestimmten Aufgabe dauert
 - **Meta-Notizen** Experimentator beobachtet und interpretiert Ablauf schriftlich

Beispielaufgabe I: Rechteck zeichnen

- Programmprinzip bekannt
- Konkretes Programm unbekannt
- „Studie“

Aufgabe Vorgegebenes Bild zeichnen

Zweck Intuitivität der Oberfläche testen

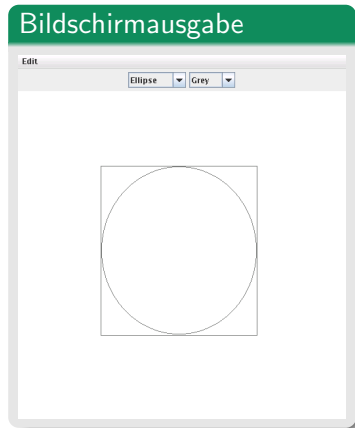


Beispielaufgabe I: Rechteck zeichnen

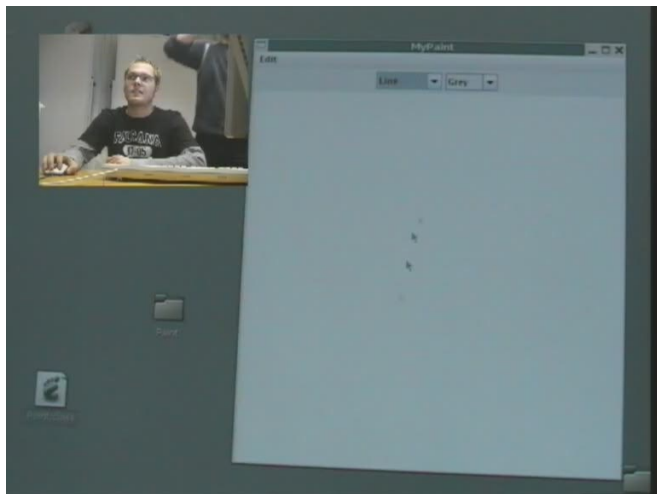
- Programmprinzip bekannt
- Konkretes Programm unbekannt
- „Studie“

Aufgabe Vorgegebenes Bild zeichnen

Zweck Intuitivität der Oberfläche testen



Video zu Beispiel I



`/vol/tdpe/share/material/session12/draw.avi`

Beispielaufgabe II: Information auf Website finden

Aufgabe Finde die Anzahl Leistungspunkte für das Bestehen des Moduls „Werkzeuge und Programmierung“



Technische Fakultät

Startseite

Studium

Forschung

Personen

Willkommen auf den Seiten der Technischen Fakultät der Universität Bielefeld

Mit ihren Schwerpunkten **Bioinformatik**, **Biotechnologie** und **Kognitive Informatik/Intelligente Systeme** bietet die Technische Fakultät innovative Studiengänge und

Biotechnologie

- Programm bekannt (Web-Browser Firefox)
- Sitezweck bekannt (Techfak), Struktur nicht
- **Zweck** Erfäßbarkeit der Seiteninhalte testen

Beispielaufgabe II: Information auf Website finden

Aufgabe Finde die Anzahl Leistungspunkte für das Bestehen des Moduls „Werkzeuge und Programmierung“

Scheinerwerb

Für diese Veranstaltung können je nach Modul bis zu 2,5 LP erworben werden:

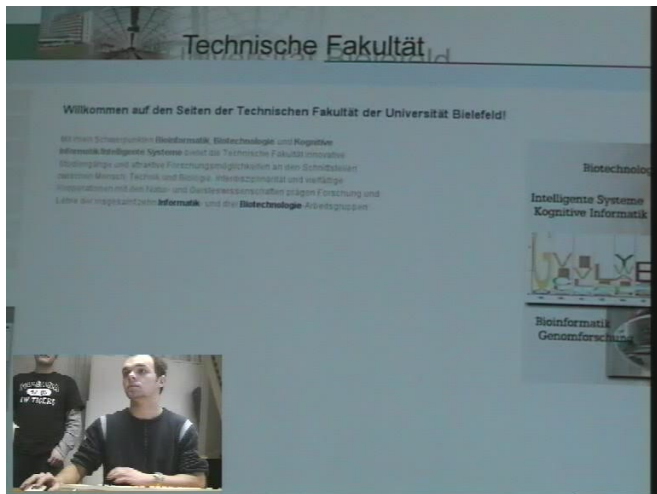
Modul	Studiengänge	LP
Werkzeuge und Programmierung	BIG , NWI , KOI , MIG	2
Werkzeuge	INF	2,5
Informatik MBT	MBT	1

Die Kriterien sind:

- regelmäßige, aktive Teilnahme an den Übungsgruppen (maximal eine nicht geeignet)

- Programm bekannt (Web-Browser Firefox)
- Sitezweck bekannt (Techfak), Struktur nicht
- **Zweck** Erfassbarkeit der Seiteninhalte testen

Video zu Beispiel II



/vol/tdpe/share/material/session12/findlp.avi

Benutzerstudien

- 1 Studienzweck festlegen
 - gestaltend z.B. zur Verbesserung der Schnittstelle im Rahmen iterativer Entwicklung
 - evaluierend z.B. zur Messung der Qualität, Vergleich von Ansätzen
- 2 Studienplan aufstellen (vor der Studie)
- 3 *Pilottest* mit wenigen Versuchspersonen
„den Test testen“
- 4 Versuchspersonen ermitteln (möglichst repräsentativ)
- 5 Durchführung und Analyse der Ergebnisse

Experimentplanung

Prinzipiell gelten die Regeln des Experiment-Designs. Kurzgefaßt:

Gültigkeit

- Ergebnisse aussagekräftig?
- Variation nur in Gegenstand der Studie
- Realistisches Szenario
- Repräsentative Benutzergruppe

Wiederholbarkeit

- Ergebnisse überprüfbar?
- **Grundproblem:** Benutzer differieren stark^a
- Vor- und Nachbedingungen exakt festlegen
- vollständiger Studienplan!

^atypische Standardabweichung
Geschwindigkeit: 33% bis 50%

Auswahl der Benutzergruppe

- Oberster Maßstab: Ähnlichkeit zur Zielgruppe
- Anpassung nach Größe
 - **klein?** möglichst einheitlich repräsentativ
 - **groß?** alle relevanten Subgruppen repräsentieren
- Bedeutung von Erfahrung
 - **Neulinge** i.d.R. wichtigste Gruppe, testet Lernbarkeit
 - **Experten** fortgeschrittene Aufgaben, testet Konsistenz

Ethische Belange

Der Benutzer hat immer Recht

- Mitteilen, dass das *System*, nicht der Benutzer studiert wird
- Systeme noch in Entwicklung!
- Neutrale Aufgaben stellen
- Motivation aufrechterhalten, nicht „leiden“ lassen
- Privatsphäre beachten, *während* und *nach* der Studie

Aufgaben

- **Vor allem:** Repräsentative Aufgaben, z.B. aus Use Cases
- Aufteilung in Bereiche, z.B. Dateneingabe, -operationen, -analyse
- Beschreibung in Schriftform (reproduzierbar, referenzierbar)
- „Witze“ vermeiden
- Erste Aufgabe besonders einfach

Fragebögen

Fragebögen dienen dem expliziten Feedback vom Benutzer und sind komplementär zur Beobachtung

- Art der Erhebung
 - **quantitativ** Auswahl aus Möglichkeiten, numerisches Ergebnis.
 - **qualitativ** Freie Antwort – hier nicht weiter betrachtet.
- Likert-Skala
 - Als Aussage formuliert
 - Antwort gibt *Grad der Übereinstimmung* an
 - Typisch: Werte von 1 (völlige Ablehnung) bis 7 (völlige Zustimmung)
- Alternativ: Einordnung zwischen zwei Extremen
- statistische Tests je nach geforderter Güte

Zusammenfassung: Elemente des Studienplans

Rahmenbedingungen

- Studienziel
- Ort, Zeit, Dauer
- Hard-/Software, Konfiguration
- reproduzierbarer Anfangszustand
- Experimentatoren
- Benutzergruppe

Durchführung

- Aufgaben
- **Erfolgskriterien**
- Benutzereinführung
- Erlaubte Hilfestellungen
- Art/Umfang der Daten
- Analyseverfahren

siehe auch (Nielsen 1993, S. 170f)

Durchführung eines Experiments

- 1 Vorbereitung
bevor die Testperson anwesend ist!
- 2 Einführung
Ziel der Studie, Vorgehen beschreiben, System *soweit nötig* einführen
- 3 Experiment durchführen
minimale Interaktion, aufzeichnen (schriftlich, auf Video)
- 4 Abschluss
evtl. Fragebögen ausfüllen, Kommentare erbitten, Report schreiben

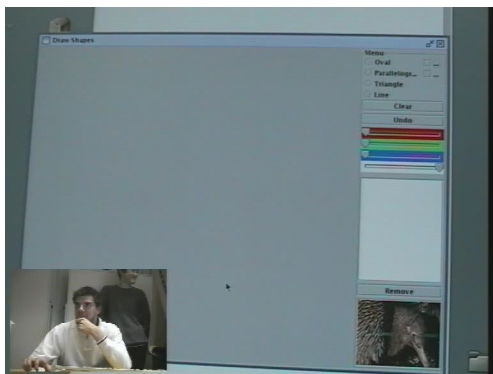
Discount Usability Studies

- „Volles Programm“ evtl. zu aufwendig
 - Lockere Benutzerstudie besser als gar keine
 - Einfach durchführbare Methoden:
 - Durchsicht auf Einhaltung der Heuristiken (1. Teil)
 - „Laut denken“
 - 1 Anwender beschreibt seine Überlegung während der Aktion
 - 2 Experimentator macht Aufzeichnungen
- Kann viele „Denkfallen“ aufdecken.

Aufgabe: Analyse des Benutzerverhaltens

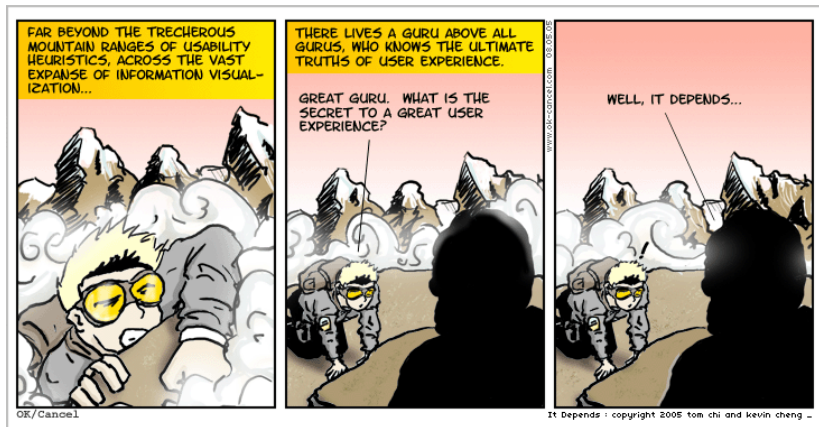
Fragen

- Was passiert?
- Treten Probleme auf?
- Vorschläge?



`/vol/tdpe/share/material/session12/draw2.avi`

The Secret to a Great User Experience



Quellen

Nielsen 1993 *Usability Engineering*, Jakob Nielsen, AP Professional, 1993.

Winograd 1996 *Bringing Design to Software*, Terry Winograd (editor), ACM Press, 1996.