DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

USABILITY EVALUATION

Lars Briem

(briem.lars@googlemail.com)

Duale Hochschule Baden Württemberg - Standort Karlsruhe

Suche nach problematischen Stellen

- Auswirkungen für den Benutzer
  - Verwirrung beim Benutzer
  - Unnötiger Zeitaufwand
  - Datenverlust
- Wie entstehen problematische Stellen
  - Fehlende Zeit
  - Fehlendes Wissen
  - Fehlende Ressourcen

$\Rightarrow$ Reflektion der Entwicklungsarbeit

$\Rightarrow$ Potentielle Lösungen aufzeigen

Review durch Experten

- Überprüfung durch
  - Usability Experte
  - Usability und Domänen Experte
- Evtl. Einführung für Usability Experte
- Idealerweise mehrere Experten
- Formelle / Informelle Evaluation
- Prüfung basierend auf einfachen Regeln

Sichtbarkeit des Systemstatus

- Welche Informationen werden dem Benutzer angezeigt?
- Wie aktuell sind die Informationen?
- Potentielle Probleme
  - Benutzer sind immer ungeduldig
  - Aktionen sind intransparent
  - Störung der Hand Augen Koordination
- Mögliche Ursachen
  - Fehlende Spezifikation der Antwortzeiten
  - Performance $\neq$ Antwortverhalten
  - Schlechte Tool Unterstützung

Responsiveness Principles

1. Responsiveness is not the same as performance

- Langsame Software mit gutem Antwortverhalten
- Schnelle Software mit schlechtem Antwortverhalten

2. Processing resources are always limited

- Je schneller der PC, desto mehr hat er zu tun
- Kunden PC langsamer als Entwickler PC

Responsiveness Principles

3. The user interface is a real-time interface

Wesentliche Zeitschranken

0,1 s	Wichtig für Hand-Augen Koordination
1 s	Zeit bis Aktion beendet oder Zeitabschätzung
10 s	Konzentration auf eine Aufgabe

Responsiveness Principles

4. All delays are not equal

- Unterschiedlich schnelle Rückmeldung
- Komplexe Aufgaben brauchen mehr Zeit

5. Don’t do tasks in the order they appear

- Umsortieren der Aktionen kann Zeit sparen

Responsiveness Principles

6. Software need not do all tasks it was asked to do

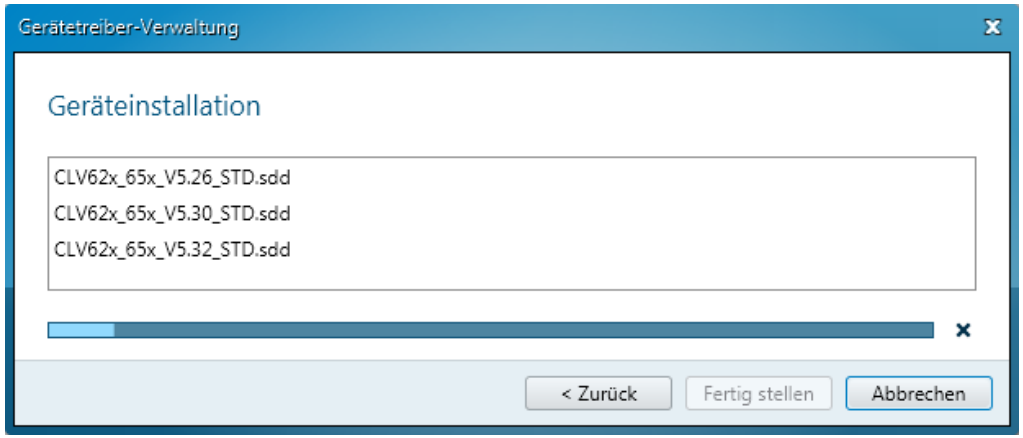
- Aktion nicht notwendig
- Zeitschranken können nicht eingehalten werden

Techniken für gutes Antwortverhalten

- Rechtzeitige Rückmeldung
- Parallele Problemlösung
- Optimierung der Aktionsreihenfolge
- Dynamisches Zeitmanagement

Rechtzeitige Rückmeldung

- Benutzereingabe direkt bestätigen
- Anzeige von „beschäftigt“ und „idle“
- Fortschrittsanzeige für lange Aktionen
- Wichtiges zu erst, komplexes hinauszögern
- Auch im Web beachten



Parallele Problemlösung

- Prioritäten vergeben
  - Hohe Priorität: Benutzereingaben
  - Mittlere Priorität: Rechtschreibprüfung
  - Niedrige Priorität: Update Check
- Vorarbeiten
  - Arbeitsschritte vorbereiten
  - Nächstes Bild / Nächste Seite laden
  - Inkrementeller Build im Hintergrund

Optimierung der Aktionsreihenfolge

- Unkritische Aufgaben verzögern / umsortieren
- Laufende Aktionen bei neuer Aktion abbrechen
- Notwendigkeit der Aufgabe überprüfen
- Unnötige Aufgaben entfernen / ignorieren

Dynamisches Zeitmanagement

- Abarbeitung der Warteschlange ändern
- Qualität / Quantität reduzieren
- Bearbeitungszeit vorausberechnen
- Einhaltung von Zeitschranken vorausberechnen

Zusammenfassung Sichtbarkeit des Systemstatus

- Extrem wichtig für den Benutzer
- Transparenz durch Rückmeldung
- Sinnvolle Verwaltung der Aufgaben
- Performance $\neq$ Antwortverhalten
- Schnellere Hardware $\Rightarrow$ Komplexere Funktionalität

Unterschiede zwischen Realität und System

- Sind die Begriffe in der Sprache des Benutzers geschrieben?
- Folgen die Aktionen der natürlichen Abfolge?
- Potentielle Probleme
  - Verwendung von Entwicklersprache oder 2er Potenzen (8, 16. . . . 1024)
  - Doppeldeutige Worte: Dialog, String, Menü
  - Sichtbarkeit interner Konzepte direkt in der GUI

Unterschiede zwischen Realität und System

HTTP Status 500 -

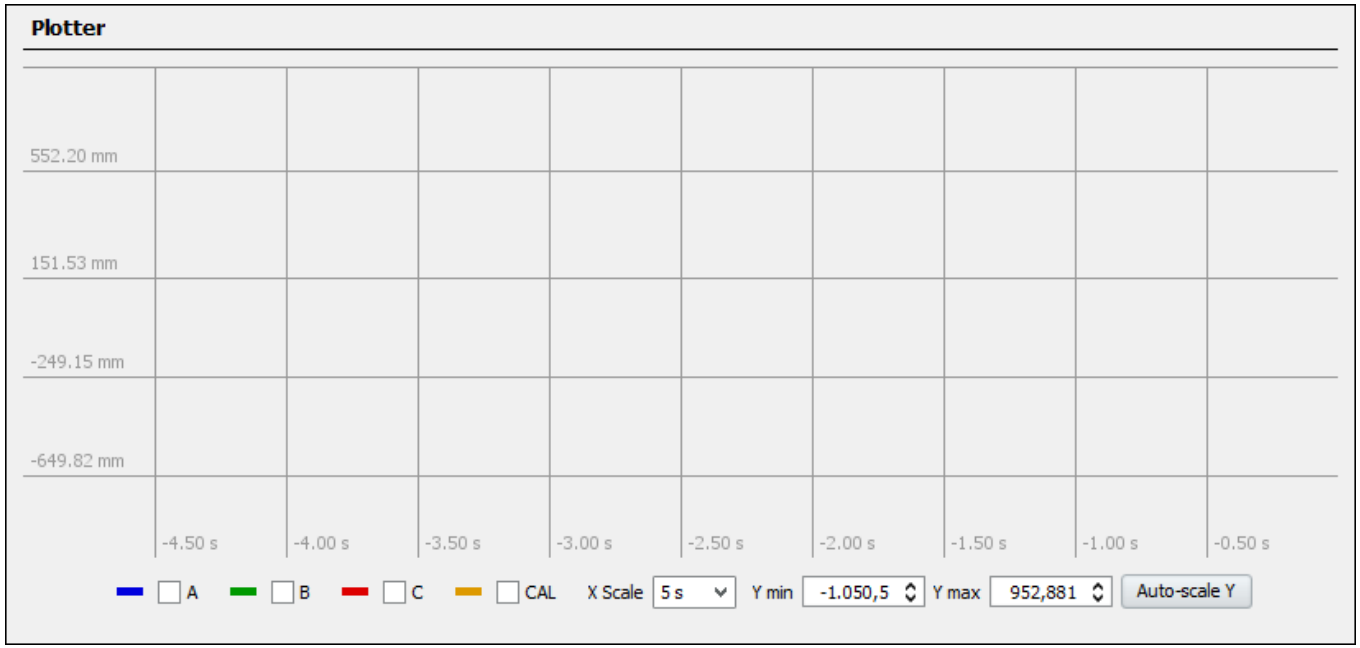
type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

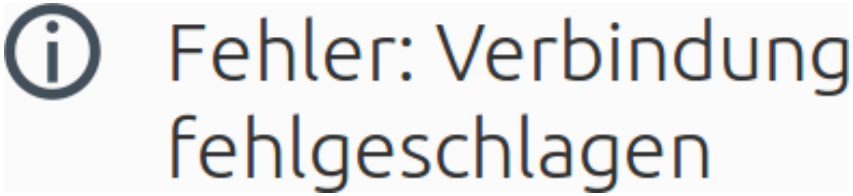
```
org.apache.jasper.JasperException: An exception occurred processing JSP page /index.jsp at line 14
```



Unterschiede zwischen Realität und System

- Mögliche Lösungsansätze
  - Sprache und Verständnis des Benutzers kennen
  - Aufgabenspezifische Begriffe verwenden
  - Konzepte auf Benutzer anpassen
  - Komplexere Informationen an anderer Stelle zugänglich machen

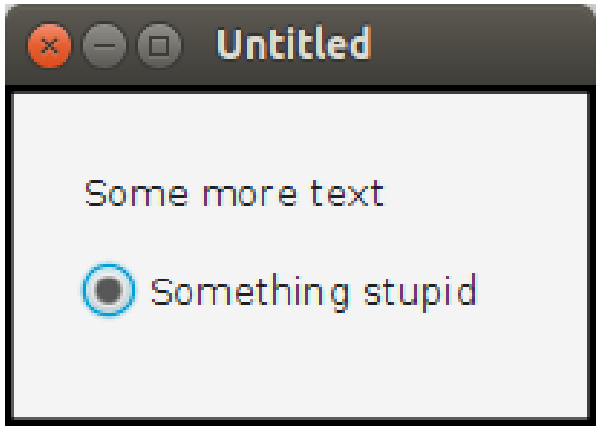
  $\Rightarrow$ Mentales Modell und Produkt Lexikon verwenden

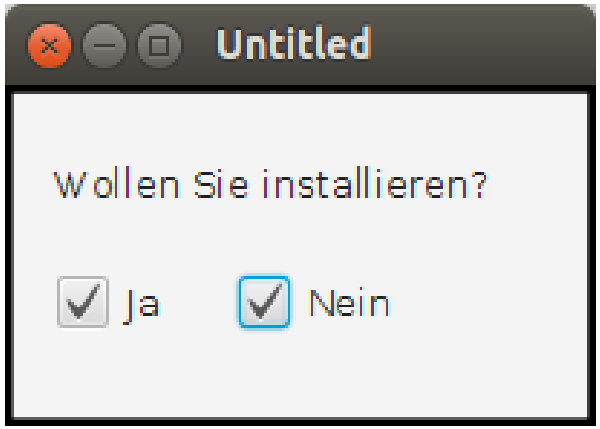


Konsistenz / Einhaltung von Standards

- Werden GUI Komponenten einheitlich verwendet?
- Werden GUI Komponenten korrekt verwendet?
- Potentielle Probleme
  - Falsche Assoziation von Aktionen zu GUI Komponenten (Affordances)
  - Checkbox vs. Radiobutton vs. Toggle vs. . . .
  - Nicht direkt editierbare Komponenten

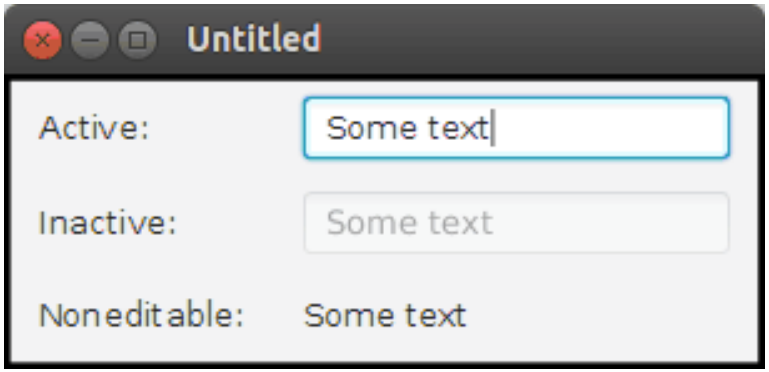
Konsistenz / Einhaltung von Standards

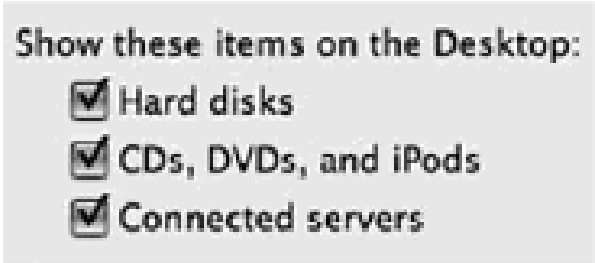




Konsistenz / Einhaltung von Standards

- Mögliche Lösungsansätze
  - Radiobutton für 1 aus N
  - Checkbox für Ja/Nein bzw. An/Aus
  - Design entsprechend anpassen





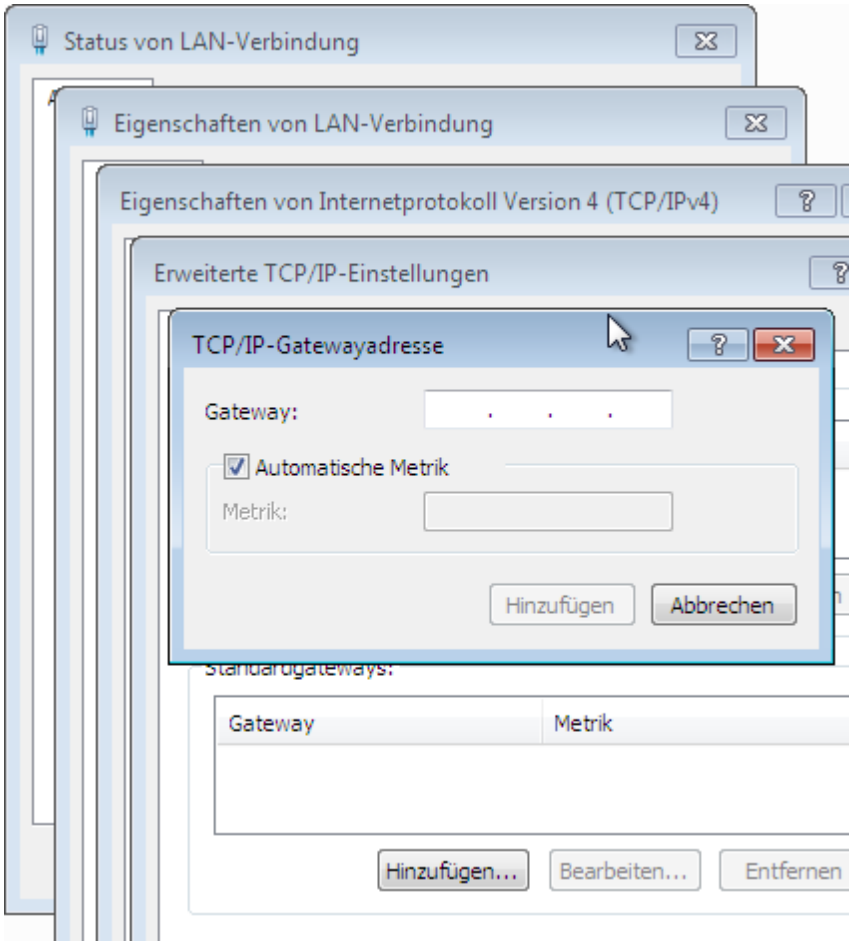
Flexibilität und Effizienz

- Welche Optimierungen können Experten vornehmen?
- Wie stark wird der Nutzer eingeschränkt?
- Potentielle Probleme
  - Zeichenbeschränkungen bei Namen und Passwörtern
  - Beschränkung der Anzahl an Eingabeelementen
- Mögliche Lösungsansätze
  - Speicherplatz zur Laufzeit vergrößern
  - Verwendung und Zuweisung von Tastenkombinationen
  - Makroerstellung

Gedächtnis des Benutzers entlasten

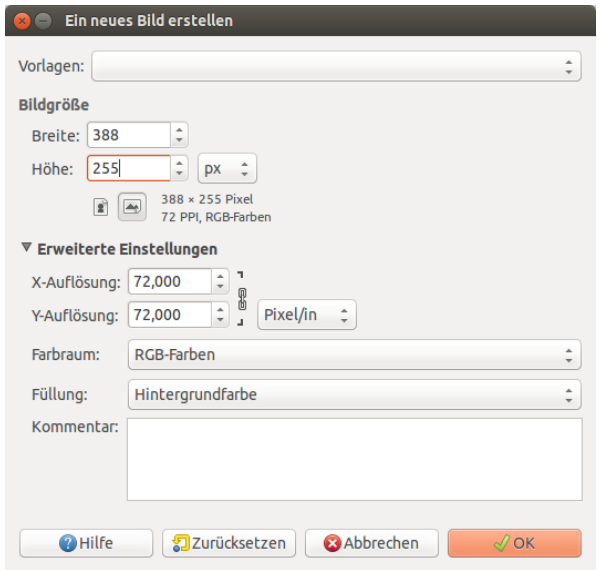
- Welche relevanten Informationen fehlen?
- Welche Informationen aus vorherigen Schritten muss der Benutzer sich merken?
- Potentielle Probleme
  - Schwer zu merkende Benutzernamen oder Passwörter
  - Lange Anleitungen, die zu schnell verschwinden
  - Zu tiefe Hierarchien

Gedächtnis des Benutzers entlasten



Gedächtnis des Benutzers entlasten

- Mögliche Lösungsansätze
  - Alle notwendigen Informationen anzeigen
  - Informationen so lange wie notwendig anzeigen
  - Maximal 2 Ebenen in der Hierarchie
  - Ein-/Ausklappbare Details statt zusätzlichem Dialog
  - Frei wählbare Namen und Passwörter



Minimalistisches Design

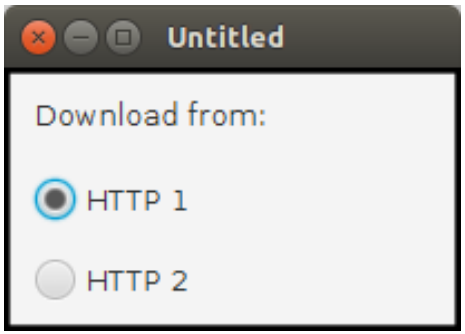
- Wie viele Informationen werden angezeigt?
- Auf welche Informationen wird der Fokus gelenkt
- Potentielle Probleme
  - Informationen zu klein oder außerhalb des Wahrnehmungsbereichs
  - Fokussierung auf unwichtiges durch falsche Größenverhältnisse oder Farben
  - Zu viel weitere Informationen (Rauschen)
- Mögliche Lösungsansätze
  - Erstellung einer visuellen Hierarchie
  - Wahrnehmungsbereich des Nutzers beachten
  - Rauschen reduzieren

Benutzer einen Ausweg lassen

- In welchen Situationen muss ein Benutzer eine Entscheidung treffen, die er nicht treffen will?
- Welche Aktionen lassen sich nicht abbrechen?
- Potentielle Probleme
  - Dialoge ohne *Abbrechen*
  - Dialoge mit unpassenden Optionen
  - Unklare Bedeutung
- Mögliche Lösungsansätze
  - Beschreibungstexte analysieren und mit Benutzer testen
  - Einzelne Wahlentscheidungen vermeiden
  - Wenn möglich immer *Abbrechen* zulassen

Fehlervermeidung

- Welche Fehler lassen sich von Beginn an vermeiden?
- Können nicht verfügbare Aktionen deaktiviert werden?
- Potentielle Probleme
  - Auswahlmöglichkeiten ohne Unterschied
  - Nicht sichtbarer Unterschied für den Benutzer
  - Falsche Auswahlmöglichkeiten

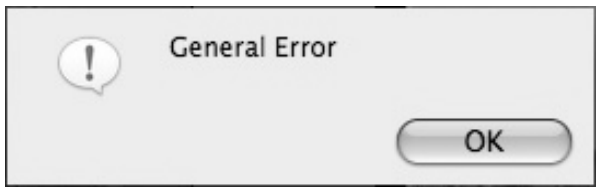


Fehlervermeidung

- Mögliche Lösungsansätze
  - Fehlerhafte Aktionen deaktivieren
  - Nur echte Auswahlmöglichkeiten anzeigen
  - Nur verständliche Fragen stellen
  - Antwortmöglichkeiten vor Anzeige überprüfen

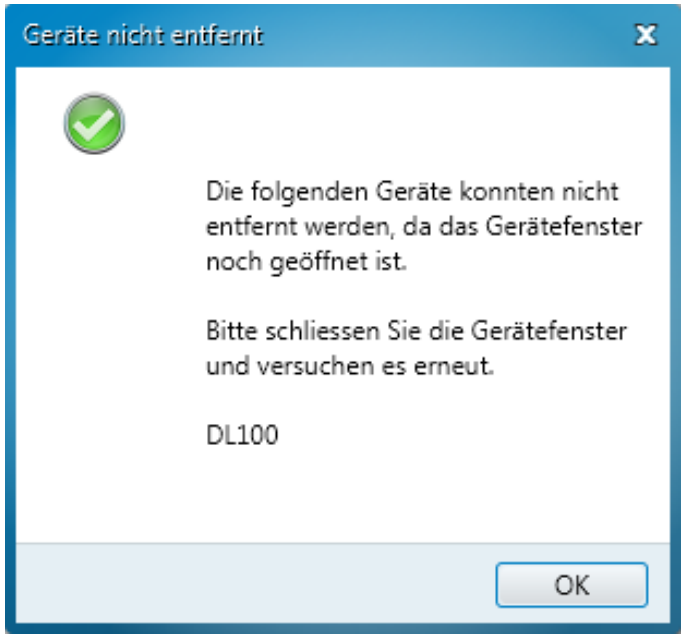
Unterstützung bei Fehlern

- Sind die Fehlermeldungen einfach verständlich?
- Wird eine Lösung für den Fehler angeboten?
- Potentielle Probleme
  - Nichtssagende Fehlermeldungen aus nicht GUI Code
  - Fehlerursache nicht bis zur GUI weitergegeben
  - Benutzer wird in die falsche Richtung geleitet



Unterstützung bei Fehlern

- Mögliche Lösungsansätze
  - Fehler passend zur Aufgabe beschreiben
  - Lösung anbieten
  - Fehlerursache bis zur GUI weitergeben
  - Adressat des Fehlers berücksichtigen
    - Benutzer
    - Administrator
    - Entwickler



Hilfe und Dokumentation

- Ist die Hilfe einfach durchsuchbar?
- Ist die Hilfe kurz verständlich und korrekt geschrieben?
- Potentielle Probleme
  - Falsche Hilfetexte
  - Schlechte Grammatik und Rechtschreibung
  - Unpr0fessi0nelles Aussehen
- Mögliche Lösungsansätze
  - Mentales Modell und Produkt Lexikon sind Grundlage für die Hilfe
  - Professionellen Schreiber bzw. Technical Writer beauftragen
  - Automatische Rechtschreib-/Grammatikprüfung

Evaluationsarten

Formell

- Jeder Experte 1 Bericht
- Zusammenfassung der Berichte durch alle Experten
- Klassifizierung der Probleme
  - Katastrophe
  - Schweres / Leichtes Problem
  - Kosmetik

Informell

- 1 Teammitglied überprüft Software
- Informelles Memo / Meeting

Usability Test

- Usability Test mit „echten“ Benutzern
- Zeit für Usability Test und Korrekturen einplanen

Ziele beim Testen:

- Information
  - Komplexe Stellen in der UI finden

  $\Rightarrow$ Ursache der Probleme finden und beheben
- Soziales Ziel
  - Entwickler sieht sich oft persönlich angegriffen

  $\Rightarrow$ Tests nicht als „Entwicklereinstufung“ gedacht

$\Rightarrow$ Teste die Software mit Benutzern und verbessere sie!

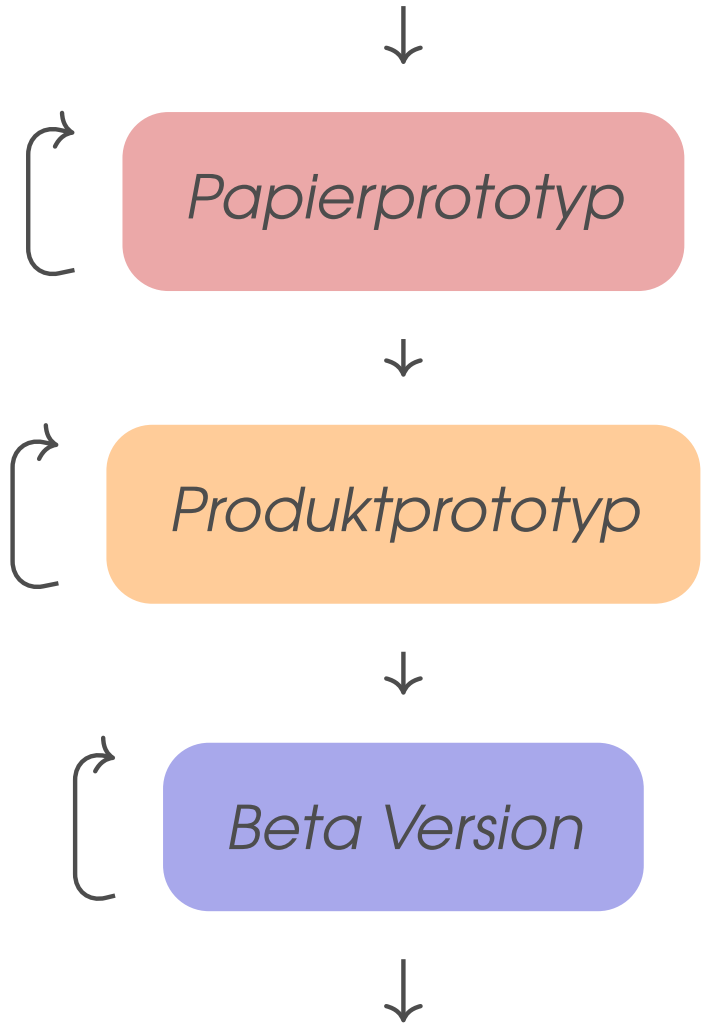
Usability Testarten

- Formativer Test
  - Während Entwicklung
  - Speziell für ein Ziel / Problem
  - Kleine Studie
  - Wiederholend / Iterativ
- Summativer Test
  - Nach der Entwicklung
  - Fertiges Produkt vor Auslieferung
  - Umfangreiche Studie

Formativer Test - Iteratives Vorgehen

- Kleine Studien
- Schnelle Rückmeldung
- Kostengünstig
- Lösung wieder testbar
- Vergleichbar mit agiler Entwicklung

Formativer Test - Iteratives Vorgehen



Vorbereitung

- Benutzerprofile erstellen / auswählen
- Szenarien und Ziele definieren / auswählen
- Umfang definieren
- Teilnehmer rekrutieren
- Zeitraum festlegen
- Testablauf für alle beteiligten beschreiben
- Labor / Unterlagen vorbereiten

Auswahl der Benutzer

- Benutzergruppe wählen
  - $\Rightarrow$ siehe Benutzerprofile
- Anzahl der Benutzer
  - Normalerweise sind 3 - 5 Personen ausreichend
  - Evtl. 1 Benutzer (wöchentlich)
- Zeitplan der Beteiligten beachten
- Reale Person unter Umständen wenig verfügbar
  - $\Rightarrow$ Evtl. „günstigere“ Vertretung

Szenarios

- Benutzer ist zielorientiert
- Beschreibt Ziel und Umfeld des Benutzers
- Bezogen auf einzelne Benutzerprofile

Benutzer sucht Informationen auf Webseite

- Registrierung für mehr Informationen
- Registrierung ist Aufgabe nicht Ziel

$\Rightarrow$ Szenario beschreibt „Suche nach Informationen“

Einführung

- Teilnehmer begrüßen
- Angenehme Atmosphäre schaffen
- Beteiligte vorstellen
- Räumlichkeiten zeigen
- Szenarien und Ziele erklären
- Verhalten während dem Test besprechen

$\Rightarrow$ Produkt wird getestet nicht Teilnehmer

Lautes Denken

- Benutzer spricht Gedanken während des Tests aus
- Gedanken werden aufgezeichnet

\+ Besseres Verständnis über
  + Benutzer
  + Absichten
  + Probleme
  + Mentales Modell des Benutzers

− Ungewohnt für viele Benutzer

− Passende Umgebung notwendig

Durchführung

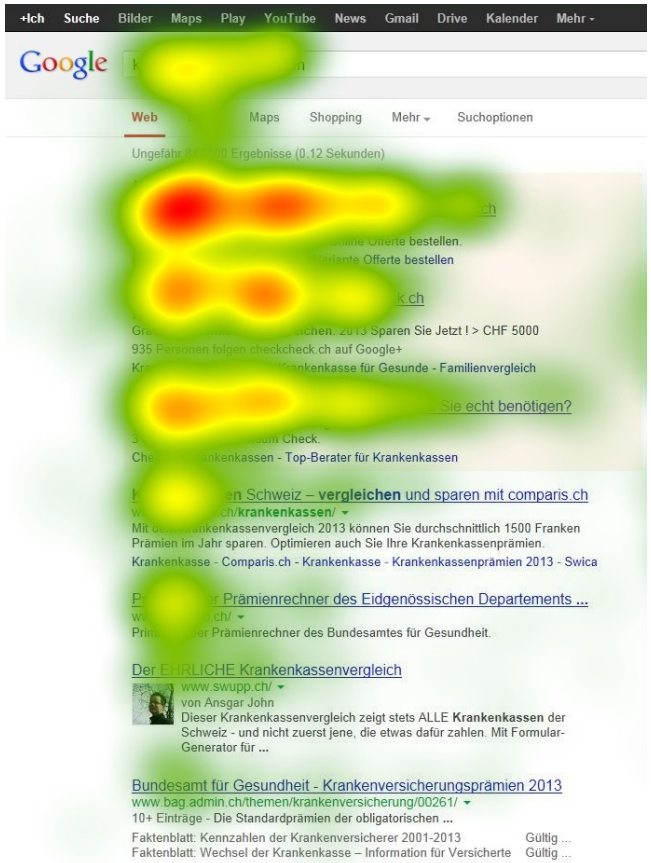
- Teilnehmer sollte Lösung selbst finden
- Teilnehmer bestimmt Tempo
- Genügend Pausen einlegen
- Klare Aufteilung zwischen
  - Moderator
  - Beobachtern
- Produktexperte für Nachfragen
- Bei Verwirrung nachfragen
- Teilnehmer verabschieden

Testumgebung

- Testutensilien
- Eigenes Labor
- Allgemein nutzbarer Raum
- Feldtest / Mobiler Test
- Remotetest

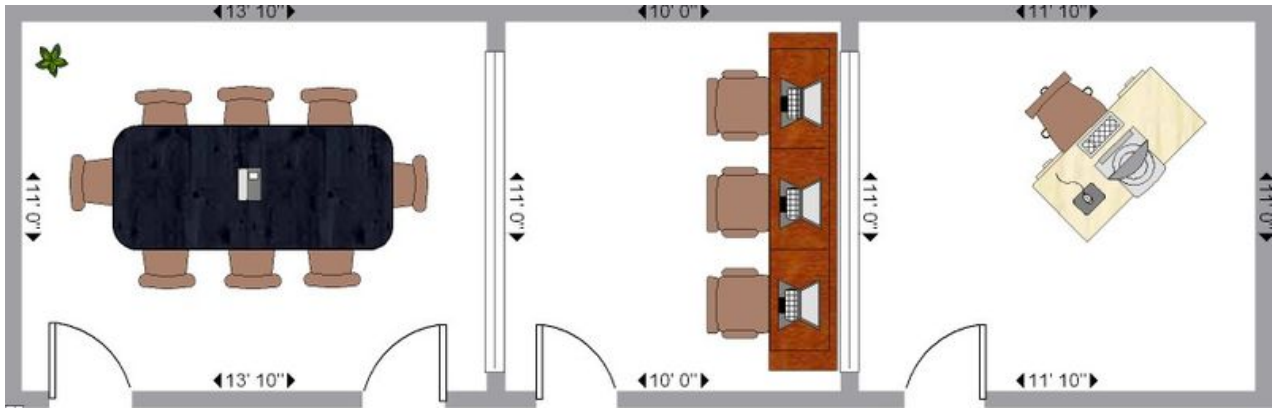
Testutensilien

- Basis Equipment
  - Raum
  - Tisch
  - Laptop / Computer
- „Nice to have“ Equipment
  - Kamera
  - Mikrofon / Headsets / Telefon
  - Logging Computer / Software
  - Generator für Umgebungsgeräusche
- Spezial Equipment
  - Eye-Tracker
  - Zusätzliches Equipment für Mobile Geräte



Eigenes Labor

- Aufbau
  - 2 / 3 Räume
  - Einwegspiegel
- Räume
  - Teilnehmer
  - Moderator
  - Management / Führungsebene



Eigenes Labor

+ Equipment immer aufgebaut
+ Basisaufwand für Test geringer
+ Gestaltung passend zu Szenario
+ Viel Equipment möglich
+ Größter Nutzen

− Teuer
− Großer Platzbedarf

Allgemein nutzbarer Raum

- Kein eigenes Usability Labor
- Besprechungsraum oder gemieteter Raum
- 1 Raum und Laptop ausreichend

\+ Günstig

\+ Wenig Platz notwendig

− Höherer Basisaufwand

Feldtest / Mobiler Test

- Potentiell überall
  - Kunde
  - Öffentliche Gebäude
  - Café / Pausenraum
- Laptop ausreichend

+ Reale Umgebung
  + Hintergrundkulisse
  + Lichtverhältnisse
  + Arbeitsplatzaufbau
+ Benutzer in gewohnter Umgebung

Feldtest / Mobiler Test

- Umgebung nicht festlegbar
- Testmoderator immer direkt beim Teilnehmer
- Ablenkungen durch Kollegen / Telefon
- Laut Denken nicht möglich / nicht gewollt
- Höhere Kosten
- Höherer Nacharbeitungsaufwand

Remotetest

- Benutzer, Moderator und weitere Beobachter örtlich getrennt
- Räumliche Zusammenführung nicht möglich
- Räumliche Zusammenführung zu teuer
- Arten
  - Synchron
  - Asynchron

Remotetest - Synchron

Moderator, Teilnehmer und Beobachter per Audio- / Videokonferenz verbunden

+ Vielfältigere Benutzergruppen günstiger erreichbar
+ Zeitersparnis für Teilnehmer
+ Größerer Testzeitraum
+ Schneller günstiger Start

Remotetest - Synchron

- Schwierigere Moderation
- Detailliertere Beschreibung notwendig
- Größerer Setup Aufwand
- Allgemeine Probleme von Remoteverbindungen

Remotetest - Asynchron

- Vordefinierte Fragen / Anweisungen
- Automatisches Aufzeichnen aller Benutzeraktionen

\+ Potentiell mehr Teilnehmer

\+ Vergleich mit Konkurrenzprodukten

\− Kein Audio- / Blickkontakt
  - Keine Begründung für einzelne Benutzeraktionen
  - Keine komplexeren Fragen

\− Teuer

Auswahlkriterien

- Idealerweise verschiedene Verfahren
- Budget
- Ressourcen
  - Räume
  - Computer
- Größe des potentiellen Teilnehmerkreises

Häufige Fehler beim Testen

- Verwendung von Wörtern aus der Benutzeroberfläche
- Beeinflussung des Testteilnehmers
  - Bewusst
  - Unbewusst
- Erzeugung von Stress
- Benutzer gibt sich die Schuld am Fehler

Evaluation

- Hauptfragen
  - Was wurde gesehen?
  - Was kann das bedeuten?
  - Wie sollte man damit umgehen?
- Evaluation durch verschiedene Personen
  - Alle im Team
  - Entwickler
  - Teilgruppe
- Einteilung der resultierenden Aktionen
  - Global vs. Lokal
  - Dringlichkeit (hoch, mittel, niedrig)

A / B Tests

Warum

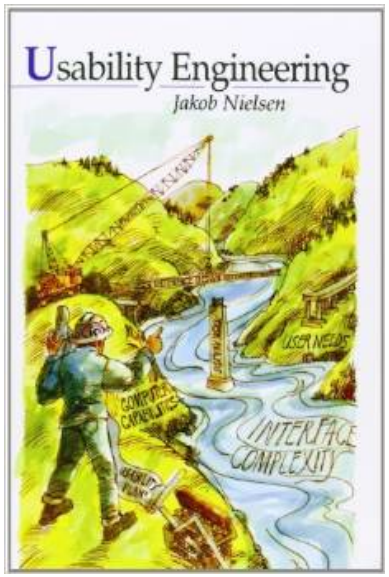
- Redesign vs. bestehende Software
- Mehrere Designvorschläge
- Unklar welcher besser
- Oft keine „beste“ Version möglich

A / B Tests

Funktionsweise

- Benutzer in Gruppen einteilen
- Jeder Gruppe eine Version geben
- Usability messen
- Statistische Signifikanz beachten

Literatur

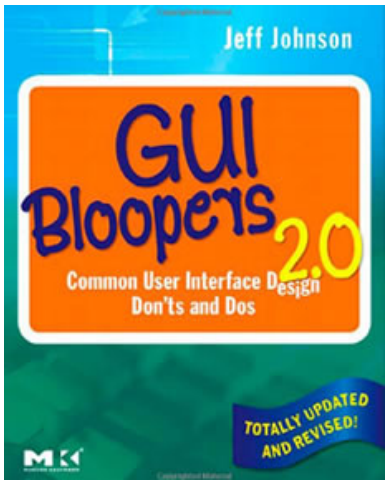


- Usability Engineering
  - Jakob Nielsen
  - Morgan Kaufmann / Elsevier
  - ISBN: 978-0125184069

- Usability Testing Essentials
  - Carol Barnum
  - Elsevier
  - ISBN: 978-0123750921

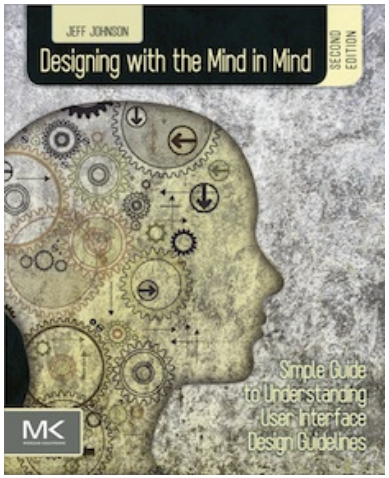
Literatur

- Designed for Use
  - Lukas Mathis
  - The Pragmatic Programmers
  - ISBN: 978-1934356753



- GUI Bloopers 2.0
  - Jeff Johnson
  - Morgan Kaufmann / Elsevier
  - ISBN: 978-0123706430

Literatur



- Designing with the Mind in Mind
  - Jeff Johnson
  - Morgan Kaufmann / Elsevier
  - ISBN: 978-0124079144

Bildquellen

- Internet
  - donrickertinventions.com
  - eyetracking.ch
  - jisc.ac.uk
  - semanticstudios.com
  - templatemonster.com
  - wqusability.com
  - zezz.co