

Usability Engineering – Phasen, Methoden und Fallbeispiele für die benutzerzentrierte Softwareentwicklung



Usability Engineering

Begriff wurde Anfang der 80er bei DEC geprägt.

“concepts and techniques for planning, achieving, and verifying objectives for system usability. The key idea is that measurable usability goals must be defined early ... and then assessed repeatedly ... to ensure that they are achieved” (Benett et al., 1984)

“developers should track changes in performance times, errors, and user attitudes for specific tasks ... to measure the impact of particular design changes on usability.” (Carroll & Rosson, 1985).

Usability Engineering vs. Softwareergonomie

Softwareergonomie

- Beschreibt primär wünschenswerte Eigenschaften interaktiver Systeme (aufgabenangemessen, selbsterklärungsfähig, individualisierbar, fehlertolerant, ...)
- Besonders geeignet für die Analyse

Usability Engineering

- Beinhaltet einen ingenieurmäßigen Prozess, der zur Erstellung gut benutzbarer Systeme führt
- Orientiert sich (auch) am Software Engineering
- Klare Entwicklungsziele in Bezug auf Usability
- Basiert auf Szenarien, Prototypen, Styleguides und Evaluierungen

Was tut ein Usability Engineer?

- Koordiniert und strukturiert Diskussionen mit Anwendern/Kunden in Bezug auf die Arbeitsweise
- Benutzerzentrierte Aufgabenanalyse, Styleguideentwicklung
- Identifiziert wesentliche Usability-Faktoren und konkrete (auch quantitative) Usability-Ziele
- Leitet den benutzerzentrierten Entwurf (Fokussierung auf Szenarien, die aus Sicht der Anwender wesentlich sind).
- Leitet die Prototypentwicklung, bereitet Tests vor, wertet Tests aus ,... .
- Zusammenarbeit mit User Interface Developer und User Interface Designer

Phasen des Usability Engineerings

- Startphase
 - Sammlung von Informationen,
 - Studium von konkurrierenden Systemen
 - Abschätzung des zeitlichen, personellen und finanziellen Aufwandes
 - Wesentliches Ergebnis: “Situation of concern”
- Initiale Entwicklungsphase
 - Aufgabenanalyse, Kontakt zu Benutzern
 - Definition von Entwicklungszielen und Meilensteinen

Phasen des Usability Engineerings

- Iterative Designphase
 - Zeitlich längste Phase
 - Zyklen aus Analyse, Implementierung und Evaluierung
 - Höhepunkte: umfangreiche Feldtests
- Installationsphase
 - Aufbau von Schulungs- und Supportmaßnahmen
 - Weiterentwicklung von Schulungsunterlagen

Prinzipien des Usability Engineerings

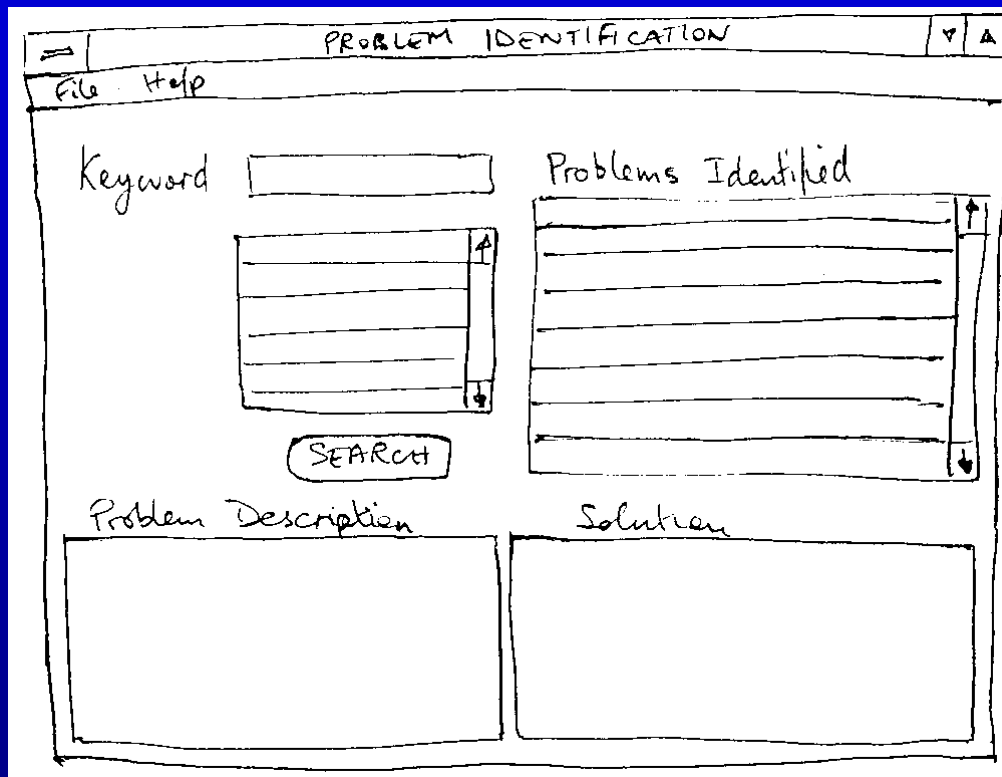
- Frühzeitige und kontinuierliche Konzentration auf Benutzer
 - Direkter Kontakt durch Interviews und Beobachtungen
- Frühzeitiges und kontinuierliches Testen
 - Skizzen und Prototypen für schnelle Tests
- Iteratives Design
 - Zyklen aus Test, Evaluierung, Implementierung
- Integriertes Design
 - Parallele Entwicklung aller Teile

Prinzipien des Usability Engineerings

- Frühzeitige und kontinuierliche Konzentration auf Benutzer
 - Fragebögen, Umfragen, Sammlung von Informationen
 - Direkter Kontakt zu Benutzern, intensive Kommunikation
 - Studium der Arbeitsplätze, Beobachten von Benutzern bei der Arbeit
 - Homogenität der Benutzer “herausfinden” und daraus nötige Anpassungen ableiten

Prinzipien des Usability Engineerings

- Frühzeitiges und kontinuierliches Testen



Quelle: Virzi et al. [1996]

Prinzipien des Usability Engineerings

- Frühzeitiges und kontinuierliches Testen
 - Entwurf und Diskussion von Metaphern
 - Nutzung von Skizzen zur Diskussion statischer Aspekte (Layout, Beschriftungen)
 - Nutzung von Flussdiagrammen und Zustandsübergangdiagrammen zur Diskussion dynamischer Aspekte
 - Erstellen von Videos (Nutzung von Graphikeditoren)
 - Interne Diskussion durch Nutzung des Intranets

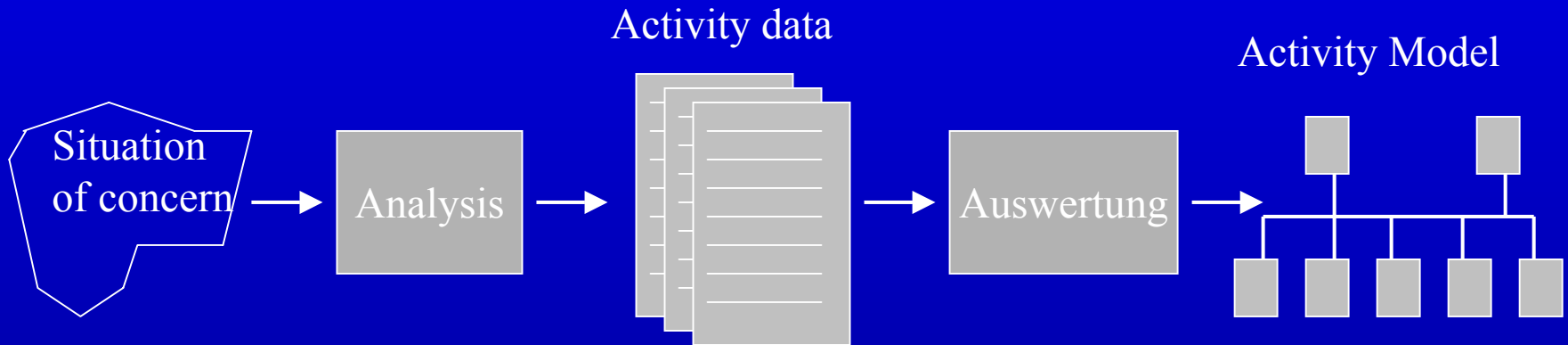
Prinzipien des Usability Engineerings

- Iteratives Design
- Voraussetzung:
 - Arbeitsumgebung und Entwicklungswerkzeuge, die Veränderungen erleichtern
- Methoden:
 - Angemessene Berücksichtigung von Iterationszyklen in der Planung
 - Sorgfältige und kreative Auswertung der Tests
 - Klare Trennung von Applikation und Schnittstelle

Prinzipien des Usability Engineerings

- Integriertes Design
 - Voraussetzung: Koordinierung durch eine Persönlichkeit (den Usability Engineer)
 - Parallele Entwicklung von Benutzungsschnittstelle, Hilfesystem, Schulungs- und Installationsunterlagen
 - Nutzung von Werkzeugen, die die Konsistenz (z.B. von Begriffen) sichern

Analyse von Aufgaben und Benutzern



Nach: Newman und Lamming [1995]

Aufgabenanalyse: Motivation

- Detailliertes Verständnis der unterstützten Vorgänge ist Voraussetzung
 - für die Erstellung quantitativer Ziele
 - für die Gestaltung von Evaluierungen.
- Positiver Kontakt zwischen Entwicklungsabteilung und Kundenabteilung (bzw. Kunden) entsteht.
- Relativ hohen Kosten steht gegenüber, dass ein verlässliches Modell der Aktivitäten der Benutzer entsteht, das
 - über einen mittleren Zeitraum (etwa 5 Jahre) für die Entwicklung weiterer Versionen und
 - Verwandter Produkte verwendet werden kann.

Aufgabenanalyse

- Identifikation der Motivation für eine Neuentwicklung (*Situation of concern*) und der wesentlichen Erfolgsfaktoren (vgl. Newmann und Lamming [1996])
- Bisherige Lösung der Aufgabe
- Kontext der Lösung
- Vor- und Nachteile der bisherigen Lösung

Methoden

- Besuch vor Ort
- Studium von Aufgabenbeschreibungen
- Persönliche Interviews
- Fragebögen

Aufgabenanalyse

“Situation of concern”

- Fehlende oder unzureichende Integration einzelner Vorgänge (Bsp. eine integrierte Entwicklungsumgebung soll einzelne Werkzeuge zur Programmierung ablösen)
- Verbesserung des Services (Bsp. ein webbasierter Help-Desk, an dem Fragen und Probleme klassifiziert und strukturiert werden, damit geeignete Ansprechpartner gefunden werden)
- Hohe Kosten (Bsp. Digitale Befundung in der Radiologie, um die Kosten der Filmentwicklung und die Kosten der Suche nach Bildern zu sparen)
- Starke Abhängigkeit von beschränkten Ressourcen (anatomische oder chirurgische Lehrsysteme auf Basis von 3d-Modellen, reduziert Abhängigkeit von Lehrkräften, Tierpräparaten)

Aufgabenanalyse

“Situation of concern”

- Reduktion langer Warteschlangen (Bsp. Fahrkartenautomaten)
- Hohe Personalkosten (Bsp. Geldautomaten, Electronic Banking)
- Hohe Anzahl von Tierversuchen (Bsp. Entwicklung von gut parametrisierbaren Simulationsmodellen)
- Vereinfachung von Vorgängen (Bsp. Start eines Autos, früher: Benzinhahn an, Chock ziehen, Maschine ankurbeln,...)

Aufgabenanalyse

Kritische Erfolgsfaktoren: usefulness und usability

- Auflösung der “situation of concern” (usefulness)
- Beibehaltung der Qualität in anderen Aspekten (z.B. Qualität und Effizienz der digitalen Befundung im Vergleich zur konventionellen) (usefulness)
- Usability
- Preis und Zeit der Entwicklung

Aufgaben und Prozessanalyse

Aufgaben	Prozesse (Geschäftsprozesse)
Klares Ziel (Bsp. Fahrkartenkauf, Platzreservierung)	Haupt- und Nebenziele (Reisevorbereitung, Operationsplanung)
Überschaubare Anzahl an <ul style="list-style-type: none">• Teilaufgaben,• Abhängigkeiten,• Variationsmöglichkeiten	Viele optionale Teilaufgaben, komplexe, teilweise schlecht formalisierbare Abhängigkeiten
Analyse ist einfacher und präziser.	Analyse unterstützt das Verständnis für die Bedeutung einzelner Aufgaben in ihrem Kontext.

Methoden der Aufgabenanalyse

Methoden	Varianten	Vor- und Nachteile
Beobachtung	“Über die Schulter schauen”, Videoaufzeichnung, akustische Aufzeichnung	Persönlicher Kontakt, wenig Vorbereitung, Performance-Aspekte können direkt erfasst werden, Ergebnisse sofort verfügbar
Interview	Mehr oder weniger stark strukturiert, Interview von mehreren einzelnen Personen oder einer Gruppe	Persönlicher Kontakt, mittlerer Vorbereitungsaufwand, organisatorisch und zeitlich aufwändig, Am besten geeignet, um Non-Routine-Aspects zu erfassen.
Fragebogen	Integration von relativ allgemeinen Fragen	Hoher Vorbereitungsaufwand, statistische Auswertung möglich, Daten erst nach langer Zeit verfügbar, aufwändige Auswertung

Methoden der Aufgabenanalyse: Beobachtung

- Aufzeichnung planen (Video oder verbale Kommentare des Benutzers)
- Videoaufzeichnung von Bildschirmanzeigen kaum erkennbar \Rightarrow Video durch verbale Kommentare ergänzen
- Transcript aus Videos erstellen relativ aufwändig.
- Varianten:
 - Passive Beobachtung: keine Aktion des Benutzers
 - Action Research: Analyst führt die Aktivität selbst aus (bzw. nimmt daran teil)

Methoden der Aufgabenanalyse: Fragebögen

Gestaltung muss sich an der geplanten Auswertung orientieren.

- Zeit- und Konzentrationsbedarf für das Ausfüllen minimieren
- Zweck der Befragung benennen
- Umfang (max. eine Doppelseite)
- Neutrale Formulierung der Fragen
- Geeignete (und beschriftete!) Skalen
- Typische Antworten sollten angekreuzt werden können. (Dabei deutlich machen, ob eine oder mehrere Kreuze möglich sind!)
- Platzbedarf bei Antworten durchdenken
- Kontaktperson angeben
- Fragebogen vor einer Verschickung “testen” (intern mit Kollegen und mit 1-2 Anwendern) und überarbeiten

Methoden der Aufgabenanalyse: Fragebögen

Varianten von Fragebögen:

- Befragung mittels e-Mail.
 - Befragte Person kann nicht anonym bleiben.
 - Geringere Flexibilität des Layouts.
 - Evtl. arbeitet nur ein Teil der Benutzer mit e-Mail, so dass ein Bias entsteht.
 - Anwendung: Bevorzugt kürzere Fragebögen (< 10 Fragen)
- Befragung auf Web-Sites
Befragte Personen werden nicht ausgewählt.

Beide Varianten sind sehr kostengünstig!

Methoden der Aufgabenanalyse: Fragebögen

Rücklaufquote:

- Bei Benutzern in der gleichen Firma (z.B. Banken): ~33%
- Bei Benutzern außerhalb der eigenen Firma: ~10%

Rücklaufquote hängt ab, von

- Unterstützung durch das eigene Management bzw. die Leitung der Kundenorganisation (Amtsleiter) signalisiert durch das Anschreiben
- Inhalt und Formulierung des Anschreibens (Zweck, Hinweise zum Ausfüllen, Ansprechpartner, Kenntnis der Ergebnisse)
- Leichtigkeit, den Fragebogen zurückzuschicken (adressierte und frankierte Briefumschläge beilegen)

Befragung von Chirurgen

Persönliche Daten

Damit wir die Antworten besser strukturieren können, bitten wir um einige persönliche Daten. Diese Daten werden selbstverständlich vertraulich behandelt und nur in anonymisierter Form verwendet.

Alter:

Geschlecht:

Erfahrungen in der Tumorchirurgie (in Jahren):
.....

Gebietsbezeichnung

- ohne
- Chirurgie
- Sonstige

Spezialgebiete:
.....

Durchschnittliche Anzahl Operationen pro Jahr:
.....

Erfahrungen mit OP-Planung auf der Basis von 3D-Visualisierung:

- Ja Nein
- Wenn ja, welche Erfahrungen haben Sie dabei gemacht?
.....
.....

OP-Vorbereitung

Frage 1: Welche Informationen und Hilfsmittel werden zur OP-Planung genutzt?

- Bilder
- eigene Skizzen
- Operationslehren
- andere

Frage 2: Wie wird der OP-Plan dokumentiert?

- gar nicht
- grob informell
- ausführlich anhand eines Formulars
- mit Computerunterstützung

Frage 3: Welche Maße beeinflussen die Entscheidung für einen Eingriff bei einem Leber- bzw. Pankreaskarzinom?

- Abstände zwischen Tumor und wichtigen Gefäßen
- Ausdehnung des Tumors
- Volumen
- Durchmesser wichtiger Gefäße
- andere

Kommunikation bei der Planung

Frage 4: Welche Informationen bekommen Sie von der Radiologie?

.....
.....
.....
.....
.....

Frage 5: Welche Aspekte der OP-Vorbereitung werden mit Kollegen diskutiert?

- spezielle OP-Techniken
- Komplikationen
- andere

Frage 6: Wie ausführlich werden Ihr(e) Assistent(en) bzw. andere Teilnehmer an der OP über die geplante Operation informiert?

.....
.....

Frage 7: Welche Rolle spielen CT- und MRT-Bilder bei der Kommunikation?

- keine
- zur Groborientierung
- zur detaillierten Diskussion der Lageverhältnisse

Unterschiede zwischen der OP-Vorbereitung und Durchführung

Frage 8: Welche Entscheidungen werden oft intraoperativ getroffen?

.....
.....
.....

Frage 9: Welche Entscheidungen können nur intraoperativ getroffen werden?

.....
.....
.....

Frage 10: Welche Situationen führen dazu, daß ein OP-Plan nicht umgesetzt werden kann?

.....
.....
.....

Frage 11: Wie genau werden die Resektionsgebiete begrenzt?

- sehr grob
- grob
- detailliert
- sehr genau

Methoden der Aufgabenanalyse: Interviews

Planung von Interviews

- Aktivitäten festlegen, die studiert werden sollen
- Länge festlegen (30 .. 60 Minuten, bei längeren Interviews Pausen)
- Mehr Zeit für die ersten Interviews einplanen
- Keine zu starre Struktur bei Interviews in der Anfangsphase der Analyse
- Benutzer und deren Ängste berücksichtigen (Benutzer fragen sich, ob schwierige Anforderungen auf sie zukommen, ob ihr Arbeitsplatz “wegrationalisiert” werden soll).
- Fokus auch auf Ausnahmesituationen, Fehler und Probleme
- Erlaubnis für die akustische Aufzeichnung einholen (auch anbieten, dass bestimmte Antworten nicht aufgezeichnet werden).

Auswertung der Aufgabenanalyse

- Ergebnisse der Aufgabenanalyse sind in der Regel zu komplex, um direkt für den Entwurf genutzt zu werden.
- Aufgabenanalyse ist ein Sammelprozess, bei dem eine Fülle von Rohdaten anfällt. → Daten müssen gefiltert, strukturiert, validiert und interpretiert werden.
- Typische Ergebnisse:
 - Videos (Benutzer bei der Arbeit)
 - Ausgefüllte Fragebögen
 - Mitschriften von Interviews
 - Logging-Protokolle, die die Arbeit an einem interaktiven System auf Low-Level-Ebene beschreiben

Auswertung der Aufgabenanalyse

Ziele:

- Identifikation von Schwachstellen
- Identifikation der wichtigen Aspekte
- Entwurf alternativer Lösungen
- Integration von Ergebnissen der Analyse mit ursprünglichen Zielen
- Fokus der Entwicklung auf geringen Lernaufwand oder effiziente Benutzung oder Fehlerfreiheit bei der Interaktion

Ergebnisse der Aufgabenanalyse:

- Tabellarische Aufstellung von Aufgaben, Teilaufgaben und Parametern
- Activity Model und Performance Model
- Problembeschreibung

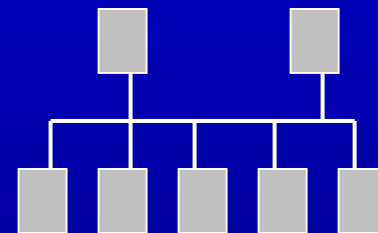
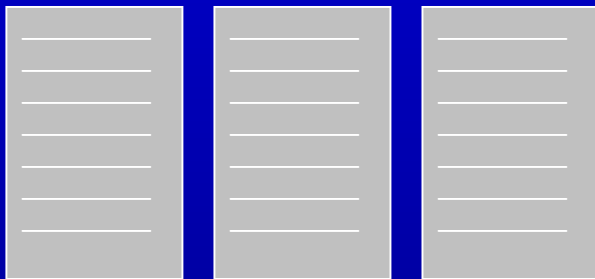
Auswertung der Aufgabenanalyse

Identifikation von Aufgaben
und Abhängigkeiten

Activity Data



Activity Model



Beispiel Fahrkartenautomat:

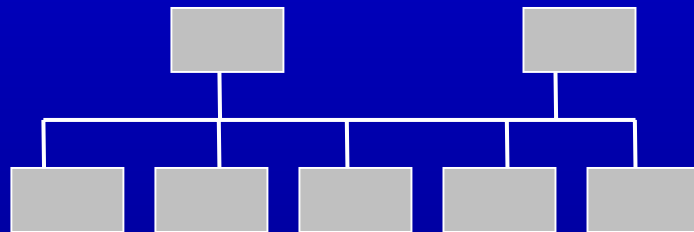
Fahrgeldbestimmung: Angabe Ziel, Angabe Zugtyp (z.B. ICE),
Ausgabe Fahrgeld

Fahrkartenausgabe: Geldeinzahlen, Ticket- und
Wechselgeldausgabe

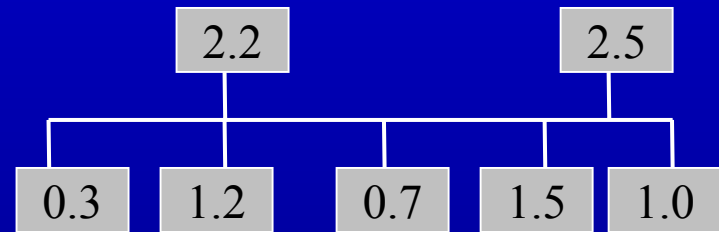
Auswertung der Aufgabenanalyse

Erfassen des durchschnittlichen Zeitbedarfs ggf. der Standardabweichung

Activity Model



Performance Model



Deskriptive Performance Model betont, dass das Modell auf Beobachtungen basiert. Normative Performance Modelle beziehen sich auf Normen, die die Dauer von Aktivitäten festlegen (Newman).

Auswertung der Aufgabenanalyse

- Problembeschreibung: Kurze Charakterisierung der Lösungsform, der Art der Untertstützung, der unterstützten Aktivität und der Benutzer
- Beispiel:



Auswertung der Aufgabenanalyse (II)

Wichtige Aspekte

- Reihenfolge von Aufgaben
- Dauer und Häufigkeit von Aufgaben
- Hervorhebung kritischer Aufgaben/Situationen
- Wichtige Probleme Ausnahmesituationen und ihre Handhabung (Workarounds)
- Möglichkeiten der (Vor)konfiguration
- Optionale und obligatorische Eingaben

Auswertung der Aufgabenanalyse (III)

- Beispiel:
 - Analyse zeigt, dass Benutzer in einer lauten Umgebung arbeiten, oft unterbrochen und abgelenkt werden (z.B. Mediziner)
- Konsequenzen:
 - Wichtige Informationen müssen besonders hervorgehoben werden.
 - Umfangreiche Kontextinformationen müssen dargestellt werden, um den Einstieg nach einer Unterbrechung zu erleichtern.

Analyse von Benutzern

- Unterscheidung zwischen Kunde (Auftraggeber) und Endanwender!!

Wichtige Aspekte:

- Alter
- Ausbildung, spezielle Fähigkeiten (z.B. *typing skills*)
- Computervertrautheit
- Sprachkenntnisse, Lesefähigkeit
- Motivation, Aufgeschlossenheit gegenüber einem Computereinsatz
- Verantwortungsbereich

Methoden:

- Direkter Kontakt, Studium von Unterlagen, Arbeitsergebnissen, Aufgabenbeschreibungen

Analyse von Benutzern (II)

Ergebnisse:

- Identifikation von Benutzergruppen mit ihren Bedürfnissen, tabellarische Aufstellung (Bsp. Webbasierter Studienplan)
- Orientierung für den Systementwurf (Was sollte genau gemacht werden? Welche Prioritäten sollten gesetzt werden?)
- Schulungsaufwand (Welche Voraussetzungen sind bei bestimmten Benutzergruppen vorhanden?)

Analyse von Benutzern (II)

Fragen zur Analyse von Benutzern (nach Mayhew 1999)

Question: Feel about computers

Answers: {don't like| neutral| like}

Question: Computers affected your job

Answers: {more difficult| neutral| easier}

Question: Interested in computers

Answers: {not interested| only as a means| interested}

Question: Enjoy learning applications

Answers: {no| sometimes| yes}

Partizipatives Design

- Integration eines Anwenders in das Entwicklerteam
- Gemeinsame Erarbeitung und Diskussion von Entwürfen
- Verbreitung in Skandinavien (Mitbestimmungsaspekt)
- Bsp:
 - Softwareentwicklung in einer Firma
 - Informationssystem für olympische Spiele (Boies, 1985)

Voraussetzung: gegenseitiges Verständnis von Anwendern und Entwicklern

Probleme: Entwicklung wird oft teurer und dauert länger

Quantitative Ziele der Entwicklung

Motivation:

Ingenieurmäßige Prozesse sind auf das Erreichen messbarer Ziele ausgerichtet. Entwickler und Auftraggeber einigen sich frühzeitig auf diese Ziele.

Prototypen werden getestet und iterativ weiterentwickelt, um die Ziele zu erreichen.

Dem Erreichen messbarer Ziele wird auch in der Softwareentwicklung höhere Priorität (Geld, Zeit) eingeräumt als dem Erreichen vager Ziele.

Quantitative Ziele zur Fokussierung der User Interface Entwicklung.

Quantitative Ziele der Entwicklung

Ziele:

- Einsparmöglichkeiten/Verbesserungen durch Benutzerfreundlichkeit messbar machen
- Fortschritte bei der Verfeinerung der Benutzungsschnittstelle erkennbar machen

Beispiele:

- **Lernaufwand:** 75 % der Benutzer erlernen fehlerfreie Erledigung einer Aufgabe in 20 Minuten.
- **Behaltensleistung:** Anteil der Benutzer, die nach 1 Woche bzw. 4 Wochen eine Aufgabe korrekt erledigen.
- **Effektivität:** 75 % der Benutzer können eine Aufgabe in 30 Minuten erledigen.
- **Fehlerrate:** Zahl der erkannten und unerkannten Fehler, kritische Fehler
- **Konsultationsrate:** Benutzung von Online-Hilfe, Manuals, Anrufe bei der Hotline

Analyse im Rahmen des Usability Engineerings vs. Herkömmliche Systemanalyse

Systemanalyse fokussiert auf:

- DV-Aspekte (wo fallen Daten an?, wo werden sie verarbeitet? Gespeichert?)
- Analyse in techn. Sinn (techn. Sprachgebrauch)
- **Ergebnis:** Datenflussdiagramme
- **Ziel:** Integrität, Effizienz und Zuverlässigkeit der resultierenden DV-Software
- Unterstützung für den softwaretechnischen Entwurf

Aufgaben- und Benutzeranalyse fokussiert auf:

- Menschliche Aktivitäten, Usability-Faktoren
- Akzeptanz bei Benutzern, Orientierung für den Entwurf der Benutzungsschnittstelle

Zusammenfassung

Usability Engineering beginnt mit einer Analysephase, in deren Ergebnis

- die zu erledigenden Aufgaben und der Kontext präzisiert sind,
- die wesentlichen Erfolgsfaktoren erfasst sind,
- typische Benutzergruppen mit ihren Eigenschaften identifiziert sind,
- Rahmenbedingungen festgelegt werden und
- quantitative Ziele hinsichtl. Benutzbarkeit gesetzt werden.

Iterative Entwicklung interaktiver Systeme

- Richtlinien,
- Prototypen,
- Gestaltung und Auswertung von Tests

Richtlinien im Entwicklungsprozess

Richtlinien

- Geben Orientierung
- Tragen zur Konsistenz in einer Anwendung und zwischen mehreren Anwendungen bei
- Dienen der Diskussion
- Dokumentieren Erfahrungen
- Tragen zur Strukturierung von Tests bei
- Erleichtern die Zertifizierung von Software
- Erleichtern die Erstellung von Handbüchern und Hilfetexten

Kriterien für “gute” Richtlinien

- Nachvollziehbarkeit (Beispiele und Gegenbeispiele)
- Begründungen
- Angemessene Beschreibung des Gültigkeitsbereiches
- Konsistenz und Kohärenz einer Menge von Richtlinien
- Richtlinien sollen Orientierung geben, aber dennoch Freiräume lassen. → Nicht zu viele “starre” Richtlinien (z.B. “Animationen müssen abschaltbar sein.”)

Inhalte von Richtlinien

Beispielhafte Gliederung

- Einführung
 - Zweck, Inhalt, Wie der Styleguide genutzt werden soll?
 - Wesentliche Inhalte der Aufgaben- und Benutzeranalyse
 - Möglichkeiten und Einschränkung durch die Entwicklungsplattform
 - Entwicklungsziele (quantitativ und qualitativ)
- Anwendung von Interaktionsstilen
- Bezeichnung wichtiger Begriffe (in Menüs, Dialogen, etc.)
- Eingabegeräte
 - Verwendung von Maustasten und Modifiziertasten, Doppelclicks
 - Tastatur, Einsatz von Funktionstasten und Cursorstasten
- Nutzung bestimmter (Nonstandard-) Dialogelemente
- Fensterplatzierungsstrategien

Inhalte von Richtlinien

Beispielhafte Gliederung (Fortsetzung)

- Hilfesystem
 - Aufbau, Suchmöglichkeiten, Zugang zur Hilfe
- Layout
 - Groß-/Kleinschreibung, Nutzung von Kapitälchen, Fett- oder Kursivschreibung, Fonts
 - Generelles Layout (z.B. Position von Iconleisten, Statuszeilen, ...)
 - Layout von Dialogboxen, Messageboxen, Warnungen
- Systemausgaben
 - Verwendung von Farben, Klängen
 - Formulierung von Systemnachrichten, Hilfetexte, Tutorials, Schulungsunterlagen
- Verweise auf übergeordnete Richtlinien (z.B. für eine Produktfamilie oder alle Produkte der Firma oder auf das Betriebssystem)

Vermittlung von Richtlinien

- Schriftliche Dokumentation

Überzeugungsarbeit durch

- Vorträge und Diskussionen
- Live-Präsentationen
- Beteiligung der (anderer) Entwickler an der Weiterentwicklung

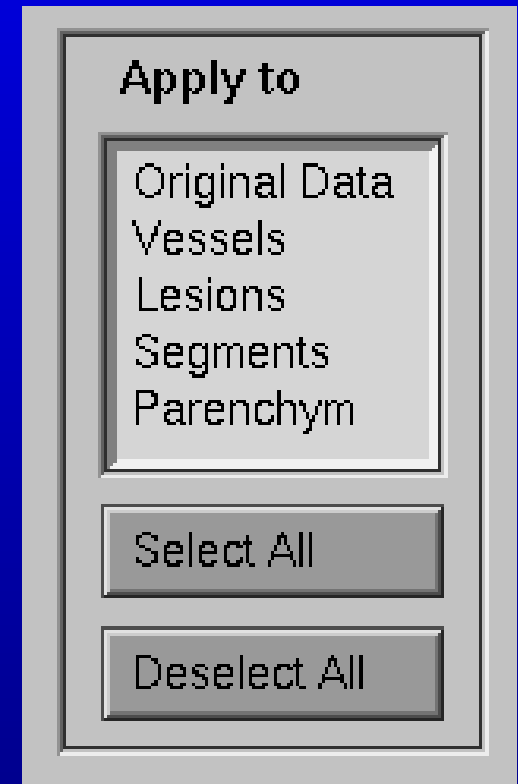
Beispiele von Richtlinien

- Allgemeine Richtlinien
- Plattformspezifische Richtlinien
 - Apple Styleguide, 1987
 - IBM SAA/CUA, 1991
 - Java Styleguide (JavaDoc)
- Projektspezifische (interne) Richtlinien

Beispiele von Richtlinien

Listen, die eine Mehrfachauswahl ermöglichen, sollten mit den Buttons **“Select All”** und **“Deselect All”**, kombiniert werden, um die Selektion/Deselektion aller Einträge zu erleichtern.

Alle drei Eingabeelemente sollten in einem Rahmen zusammengefasst sein, damit der Zusammenhang deutlich wird.



Prototypen

- **Ziel:** Test wichtiger Eigenschaften eines Systems anhand eines Modells. Prototypen sind Testgegenstände.
- **Aspekte:** Beurteilung äußerer Eigenschaften (Design) und Beurteilung des Verhaltens
- **Vorbild:** Prototypen in der Fahrzeugentwicklung,
 - Test der Konstruktion, der Aerodynamik, Risikoabschätzungen

Kriterien für “gute” Prototypen

- Prototyp ist auf die Testziele zugeschnitten.
- Prototypentwicklung ist schnell und kostengünstig.
- Prototyp kann weiterentwickelt werden.
- Tradeoff zwischen den Zielen 2 und 3. Oft wird ein Prototyp ohne Programmierung erstellt, der nicht weiterentwickelt wird.

Prototypen: Werkzeuge

Anforderungen an Prototyping-Werkzeuge (nach Newman und Lamming):

- Einfache Erstellung und Modifikation von Screen-Layouts
- Einfache Verbindung von Screens (Links)
- Unterstützung verschiedener Eingabemöglichkeiten (Buttons, Listen, ...)
- Einfache Scripting-Möglichkeiten
- Aufruf externer Programme

Beispiele: Hyper-Card (Apple-Programm), Visual Basic

Prototypen: Werkzeuge

Hyper-Card and Hyper-Talk

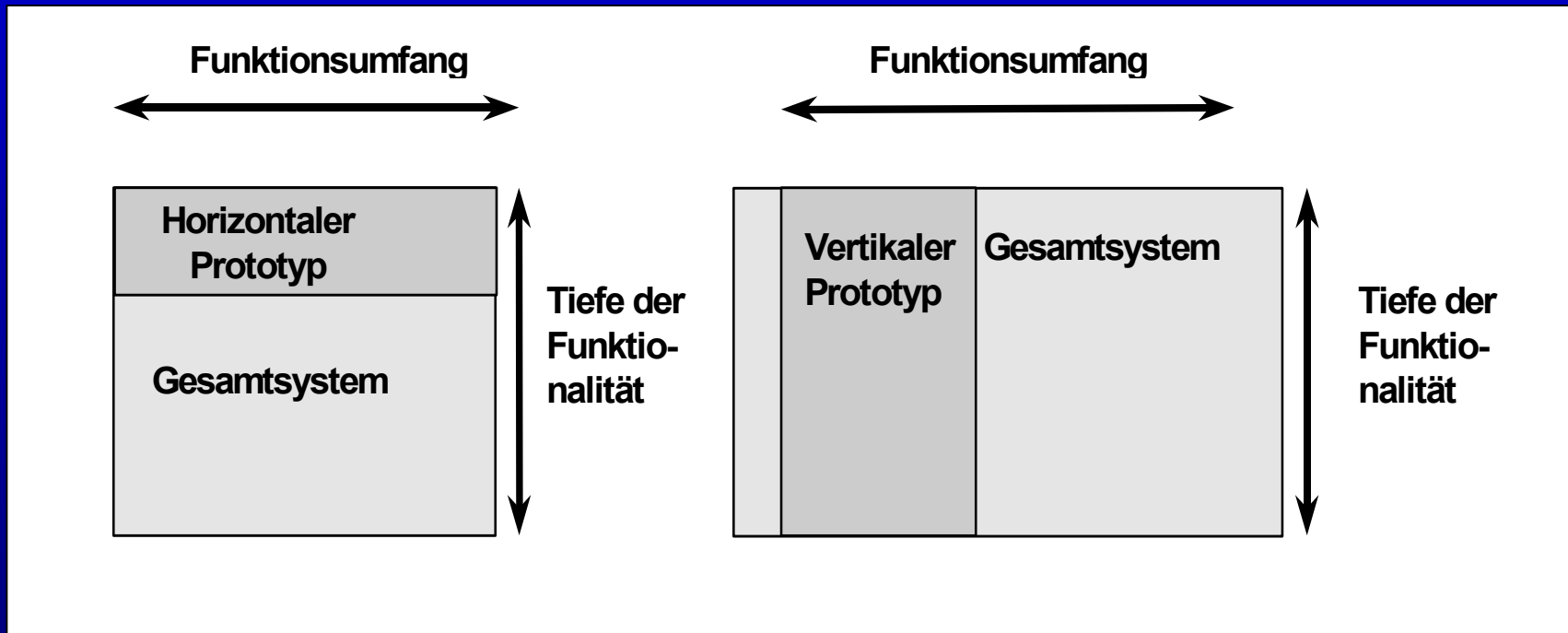
- Erstellung von Screen-Layouts auf Cards mit Boxen, Buttons, ...
- “Link-To” and “Link-From”-Angaben verbinden mehrere Screens, wobei der eine durch eine Eingabe aus dem anderen hervorgegangen ist und in der Regel ein leicht verändertes Layout hat.
- Prototyping ohne Programmiererfahrung möglich.

Hyper-Talk:

- einfache Skriptsprache (on mouseUp ..., on mouseOver ...)
- Vorläufer von Visual Basic und Java Script

Prototypen

- Horizontale Prototypen (I)
- Vertikale Prototypen (II)
- Kombinationen aus (I) und (II)



Aspekte der Prototypentwicklung

- Vernachlässigung von Optimierung
- Ignorieren von Sonderfällen
- Verwendung leistungsfähiger Hardware
- Einsatz von Pseudodaten
- Verwendung von Platzhaltern
- Kombination mit anderen Programmen (anstelle einer beabsichtigten Eigenentwicklung)

Ausgereifte und weniger ausgereifte Prototypen

- **Ausgereifte Prototypen:** Software-Prototypen, die Aussehen und Verhalten der Schnittstelle andeuten.
- **Weniger ausgereifte Prototypen:** Papierskizzen, Storyboards, Prototypen auf der Basis von Präsentationsprogrammen (erfordern keine Programmierung)

Prototypen: Kriterien für die zu testenden Merkmale

- Merkmale, die
 - Aufgrund der Aufgabenanalyse wesentlich erscheinen
 - Entsprechend der quantitativen Ziele wesentlich sind.
 - Kontroverse Diskussion unter den Entwicklern hervorgerufen haben (Vergleich von Alternativen)

Gestaltung und Auswertung von Tests

Was wird getestet?

- Leistungsparameter,
- Benutzungsschnittstelle (Dialoge, Icons, Menüstruktur, Systemnachrichten)
- Schulungs- und Hilfsprogramme,
- Installation

Wer nimmt teil?

- Potenzielle Endanwender,
- Systemadministratoren,
- Ausbilder,
- Vertriebspersonal

Gestaltung und Auswertung von Tests

Ziele der Evaluierung:

- Effizienz und Zuverlässigkeit der Bedienung
- Erlernbarkeit, Behaltensleistung
- Akzeptanz und Zufriedenheit
- Vergleich mit anderen Systemen (konkurrierende Systeme)
- Identifikation von Schwachstellen (*design flaws*) und von sehr gut gelösten Problemen
- Schwachstellen werden identifiziert, um sie zu beseitigen. Besonders positive Aspekte werden dokumentiert, um ihre spätere Modifikation zu verhindern.

Heuristische Evaluierung

- Heuristisch: sich an erlernten Zusammenhängen und “Faustregeln” orientieren (Newman und Lamming [1996])
- Evaluierung durch Experten
- Varianten:
 - Kalte Einschätzung
 - Warme Einschätzung
 - Heiße Einschätzung

Heuristische Evaluierung: Cognitive Walkthrough

Ein Usability-Experte analysiert einen Prototypen (evtl. einen nicht-implementierten Entwurf) daraufhin, ob in bestimmten Situationen erkennbar ist,

- Was zu tun ist?
- Wie nach einer benötigten Funktion gesucht werden kann?
- Ob es leicht zu versehentlichen Aktivierungen (z.B. Verwechslungen) kommen kann?
- Wie eine versehentlich aktivierte Funktion rückgängig gemacht werden kann?

Vorgehensweise: Experte analysiert das Layout, die sichtbaren bzw. leicht erreichbaren Interaktionselemente und versucht, von seinem Vorwissen über ähnliche Systeme/übliche Interaktionsmöglichkeiten zu abstrahieren.

Heuristische Evaluierung: Cognitive Walkthrough

Beispiele konkreter Fragen für einen kognitiven Walkthrough eines Fahrkartenautomaten:

- Verstehen Benutzer
 - die Reihenfolge, in der sie vorgehen können/sollten?
 - wie sie Geld eingeben können?
 - wie sie Geld zurückbekommen?
 - Wieviel Geld noch eingegeben werden muss?

Anwendbarkeit: Systeme, in denen kein großes Anwendungswissen erforderlich ist. Test der Benutzbarkeit durch Anfänger!

Heuristische Evaluierung: Usability Inspection

- Von Mack und Nielsen eingeführte Variante
- Parallelen zur Code-Inspection in der (klassischen) Softwareentwicklung
- Unterschiede zu Cognitive Walkthroughs:
 - Benutzer generell keine Anfänger,
 - Test durch mehrere Usability-Experten
- **Ziele:**
 - Identifikation, Klassifikation und Priorisierung der Probleme einer Benutzungsschnittstelle,
 - Diskussion von Alternativen,
 - Grobe Aufwandsabschätzung

Heuristische Evaluierung: Usability Inspection

Vorbereitung

- Auswahl der Testpersonen und Szenarien
- Testpersonen generell keine Anfänger

Kriterien

- Einhaltung von Richtlinien und Konventionen
- Vermeiden von Fehlern in der Bedienung
- Verringerung von Lern- und Interaktionsaufwand

Durchführung

- Individueller Test,
- Gemeinsame Auswertung

Heuristische Evaluierung: Diskussion

- Bei Begutachtung durch 3-4 Experten wird der “Löwenanteil” potenzieller Probleme erkannt.
- Heuristische Evaluierung vor einer empirischen Evaluierung zum “Aufräumen” (Kantner)
- **Vorteile:**
 - Experten können Probleme zusammenfassen und priorisieren.
 - Lerneffekt für die Tester
- **Nachteile:**
 - Anwendungsabhängige Probleme bleiben oft unerkannt.
 - In der Regel wird im Ergebnis ein Entwurf oft “nur” verfeinert und nicht ein grundsätzlich anderer Entwurf gewählt.
 - Statistische Aussagen sind nicht möglich.

Empirische Evaluierung

- Empirisch: erfahrungsbezogen
- Empirische Evaluierung: Test mit Endanwendern
- **Einordnung:** oft in späten Phasen (z.B. Feldtest)
- **Varianten:** Test des Gesamteindrucks, Test von Teilsystemen (z.B. Dokumentation)

Empirische Evaluierung: Voraussetzungen

- Testkriterien (objektive und subjektive), Priorisierung entsprechend der Aufgabenanalyse
- Geeigneter Prototyp als Testgegenstand
- Tester, die den Test vorbereiten, organisieren und auswerten (nicht die Entwickler!)
- Testablauf festlegen (was soll wie lange gemacht werden? Einarbeitung berücksichtigen)
- Geeignete (repräsentative) Testpersonen
- Testumgebung (beim Anwender, auf einer Messe, in einem Usability Lab)

Empirische Evaluierung: Durchführung

Davor:

- Aufklärung der Testpersonen über Ziele und konkrete Durchführung (Auswahl und Erläuterung der im Test zu lösenden Aufgaben)

Dabei:

- “Lautes Denken”
- Protokollierung der Interaktion in Logdateien

Danach:

- Befragung in Form von Interviews
- Ausfüllen von Fragebögen

Fragestellungen bei der empirischen Evaluierung

- Subjektive Kriterien
 - Wurde die Arbeit als {interessant| erfreulich | vorhersagbar} empfunden ? War die Arbeit phasenweise {langweilig|stupide}?
- Objektive Kriterien
 - **Effizienz:** Zeit zur Erledigung bestimmter Aufgaben
 - **Fehler:** Wobei traten wie häufig welche Fehler auf ?
 - **Erlernbarkeit/Vergessen:** Wie lange hat das Erlernen von Bedienhandlungen gedauert ?

Empirische Evaluierung subjektiver Kriterien

- Interviews
- Gruppeninterviews (günstig: 6-8 Benutzer)

Probleme von Befragungen:

Antworten konzentrieren sich auf Aspekte, die leicht verbalisiert werden können.

Zeitspanne zwischen Bedienhandlung und Befragung

Alternativen:

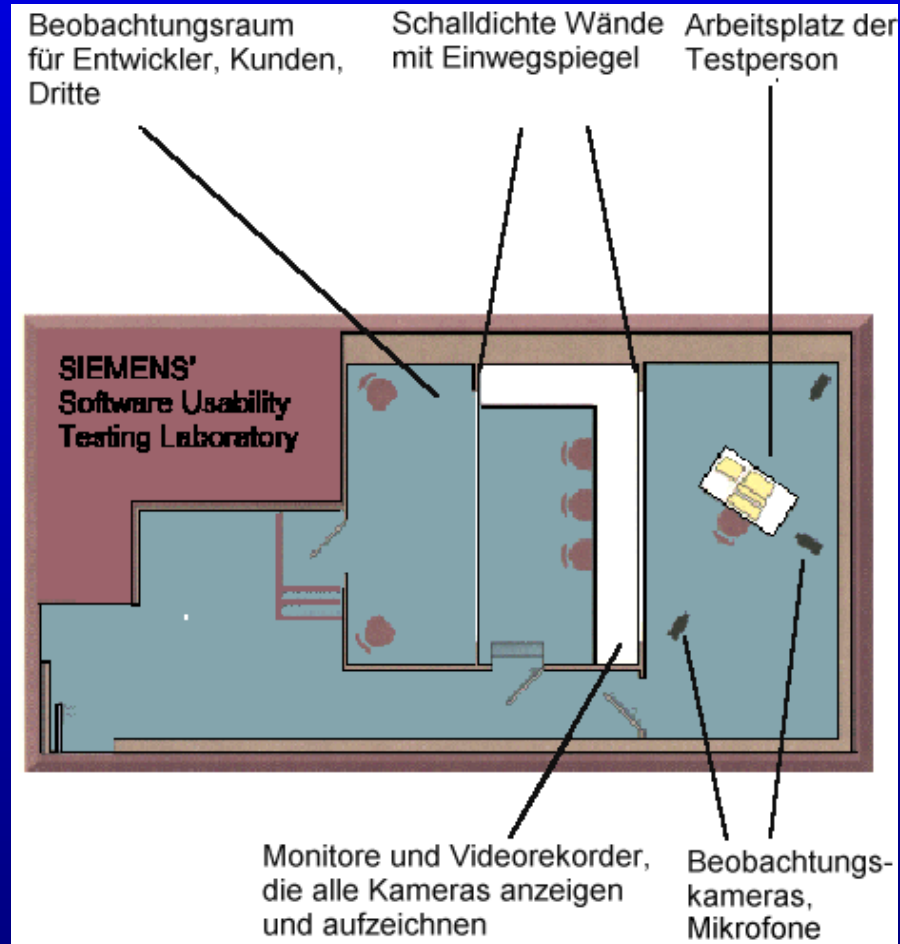
Beobachtung und “Lautes” Denken (in Kombination mit Interviews),

Tagebuchmethode bei langfristiger Nutzung (Feldtest)

Empirische Evaluierung: Usability Labs

- Spezielle, genormte Testräume in großen Firmen bzw. Organisationen
- Gründung einer Human-Factors-Gruppe zur Planung und Durchführung von Tests
- Ausgangspunkt: Usability Lab von IBM (1980)
- Bestandteile:
 - Testraum mit Testarbeitsplatz, Kameras und Mikrofonen
 - Kontrollraum für die Tester (schalldichte Wände)
 - Beobachtungsraum, z.B. für Entwickler

Empirische Evaluierung: Aufbau eines Usability-Labors



Empirische Evaluierung: Usability Labs



Blick vom Beobachtungsraum

Empirische Evaluierung über das Internet

- Nutzung von Fragebögen im WWW ermöglicht Test mit großer Zahl von (heterogenen) Benutzern
- **Ziel:** Identifikation von Problemen, Einschätzung subjektiver Kriterien
- Weitergehende Aussagen solcher Tests durch Nutzung von Videos und durch Logdateien, die beim Benutzer erzeugt werden.

Empirische Evaluierung: Qualitätskriterien

- Glaubwürdigkeit
 - Klare Definition von Zielen, Zielgruppen und Schwerpunkten
 - Plausible Auswahl repräsentativer Probanden
 - Beim Vergleich von Systemen verfügen die Probanden über gleiche Vorkenntnisse der Systeme.
 - Ausschalten von Bias-Faktoren

Empirische Evaluierung: Qualitätskriterien (II)

- Übertragbarkeit
 - Testergebnisse lassen sich auf andere Umgebungen und Aufgaben übertragen.
 - Bei Tests in Usability Labs eher gering.
- Zuverlässigkeit
 - Art, Durchführung und Auswertung des Tests sowie Auswahl der Probanden lassen erwarten, dass die mit der Stichprobe erzielten Ergebnisse für die gesamte Zielgruppe zutreffend sind.
 - Bei Tests in Usability Labs hoch.

Evaluierung: Auswertung

- Identifikation, Klassifikation und Priorisierung aufgetretener Probleme
- Entscheidung, welche Probleme mit vertretbarem Aufwand gelöst werden *können* bzw. welche Probleme aufgrund ihrer Priorität gelöst werden *müssen*.
- Identifikation von Konflikten zwischen wünschenswerten Veränderungen und positiven Aspekten, die beibehalten werden sollen.
- Dokumentation von Erkenntnissen, die evtl. auf zukünftige Entwicklungen Einfluss haben.

“Preiswerte” Testverfahren

- “Usability on the Cheap” (Tognazzini [1991]),
- “Discount Usability Engineering (Nielsen [1989])

Ziel: Möglichkeiten der Evaluierung,

- die aussagekräftig sind und
- die auch bei kurz- und mittelfristigen Projekten mit kleinerem Budget praktikabel sind

Methoden für preiswerte Tests

- *Testziel:* sorgfältige Einschränkung auf Schwerpunkte
- *Prototyperstellung:*
schnelle Prototypen ohne Programmieraufwand
- *Testumgebung:*
“normale” Arbeitsplätze, “lautes Denken”, Beobachten
- *Anzahl und Auswahl der Testpersonen:*
Beschränkung auf wenige Testpersonen (85 % der Probleme, die von 10 Personen identifiziert werden, werden von 4 Personen identifiziert),

Nutzung von Mitarbeitern (Kollegen), Studenten

Vielen Dank!

- Wenn Sie Interesse
 - an diesem Vortrag haben: preim@mevis.de
 - an dem Buch und weiteren Informationen haben:
<http://www.mevis.de/~bernhard/book>
 - am German Chapter der ACM haben:
<http://www.informatik.org>