

Vorlesung "Anwendungssysteme"

Benutzbarkeit

Prof. Dr. Lutz Prechelt

Freie Universität Berlin, Institut für Informatik

<http://www.inf.fu-berlin.de/inst/ag-se/>

- Begriff "Benutzbarkeit"
- Entwicklungsprinzipien
 - Direkter Kontakt zu Benutzern
 - Benutzbarkeitstesten
 - Iterativer Entwurf
 - Integrierter Entwurf
- Fallstudie zur Verwendung der Prinzipien
- Ratschläge an Organisationen
- Ergebnisse von Benutzbarkeitsprüfungen

- Technische Zweckmäßigkeit:
 - extern sichtbar: Plattformanforderungen, Installierbarkeit, Laufzeiteffizienz, Speichereffizienz, Funktionalität, Flexibilität
 - intern: geringer Konstruktionsaufwand, gute Wartbarkeit, Wiederverwendbarkeit, etc.
- Wünschbarkeit:
 - Gewährleistung allgemeiner passiver Anforderungen
 - z.B. Sicherheit, Verlässlichkeit, evtl. Schutz der Privatsphäre
 - Gestaltung gemäß spezifischer Wirkungs-Wünsche
 - z.B. Vermeidung negativer Wirkungen wie soziale Isolation, Monotonisierung von Arbeit, Erzeugung von Stress etc.
- Die Forderung nach guter Benutzbarkeit liegt am Berührungspunkt dieser zwei Bereiche

- *Benutzbarkeit* bedeutet, dass eine interaktive Software ihre Benutzer/innen gut dabei unterstützt, die gewünschten Arbeitsgänge zu erledigen
 - Verständlichkeit
 - Erlernbarkeit
 - Bedienbarkeit
 - Bequemlichkeit
 - Geschwindigkeit
 - Fehlervermeidung
- Benutzbarkeit ist ein Teilaspekt von *Gebrauchstauglichkeit*, die nach ISO 9241-11 umfasst:
 - Effektivität zur Lösung einer Aufgabe (insbes. passende Funktionalität)
 - Effizienz der Handhabung des Systems
 - Zufriedenheit der Nutzer einer Software

- Gute Benutzbarkeit kann nur erzielt werden, wenn ein geeigneter Software-Entwicklungsprozess verfolgt wird
 - Insbesondere im Bereich der Bedienschnittstelle (user interface, UI)
- Das Vorgehen dafür ist im Prinzip bekannt, wird aber in der Praxis verblüffend selten eingehalten
- Wir betrachten nun zuerst eine Fallstudie über diesen Effekt
 - "Warum wird so selten ein geeigneter Prozess durchgeführt?"
- und betrachten anschließend einige Elemente dieses Prozesses genauer

- 1. Direkter Kontakt zu Benutzer/inne/n**
- 2. Frühes und fortlaufendes Benutzbarkeitstesten**
- 3. Iterativer Entwurf**
- 4. Integrierter Entwurf**

- Diese Liste stammt von Gould und Lewis
 - J.D. Gould, C.H. Lewis: *"Designing for usability: Key principles and what designers think"*, Proc. CHI'83 Conf. on Human Factors in Computing Systems, pp.50–83, ACM, 1983
 - Es gibt aber diverse ähnliche andere
- Sehen wir uns kurz die Beschreibungen und Begründungen an:

1. Direkter Kontakt zu Benutzern

Beschreibung:

- Entwerfer brauchen von Anfang an direkten Kontakt zu Endbenutzer/inne/n
 - z.B. mittels Interviews, Beobachtung, Umfragen, direkter Partizipation im Arbeitsprozess

Begründung:

- Nur so können sie deren relevante **Eigenarten verstehen**:
 - Denkgewohnheiten und -fertigkeiten, Verhalten, Einstellungen
- sowie die Eigenschaften der **Aufgaben**, die die Benutzer/innen erledigen sollen



2. Frühes und fortlaufendes Benutzbarkeitstesten

Beschreibung:

- Echte Benutzer/innen verwenden die Software
 - oder Prototypen der SW
- Die Entwickler/innen beobachten dies und holen Feedback ein
- Beide entwickeln Verbesserungsvorschläge evtl. gemeinsam

Begründung:

1. Liefert die besten **Einsichten** in Probleme und Lösungen
 2. Liefert gute **Motivation**, die Software nochmal umzubauen
- Kein Ansatz zum Entwurf gut benutzbarer Software funktioniert zufrieden stellend ohne Benutzbarkeitstests



3. Iterativer Entwurf

Beschreibung:

- Die Entwicklung durchläuft viele **Zyklen** von Entwicklung, Benutzbarkeitstest, Bewertung, Entwurfsänderung

Begründung:

1. Die Benutzbarkeitsprobleme kommen nur **nach und nach** zum Vorschein
 - ein Problem *verdeckt* oft viele andere, bis es beseitigt wird
2. Korrekturen können **neue** Probleme hervorrufen
 - und tun dies oft auch



4. Integrierter Entwurf

Beschreibung:

- Alle Benutzbarkeits-Aspekte müssen zeitlich parallel und inhaltlich gemeinsam (in einer Hand) entwickelt werden
 - Benutzungsschnittstelle, Lehreinheiten, Hilfesystem/Dokumentation

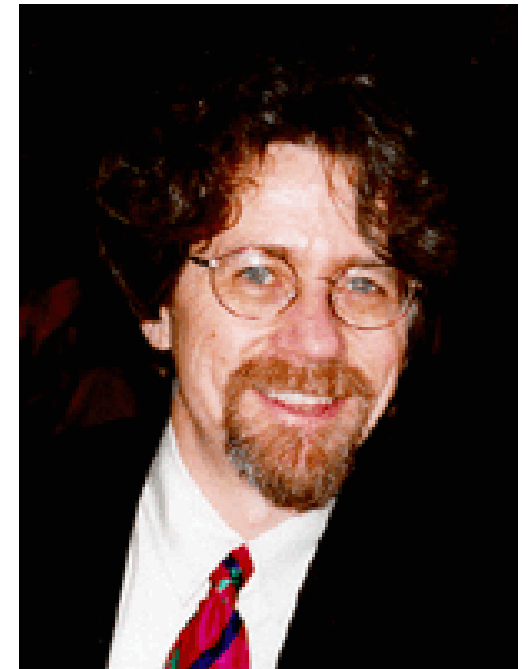
Begründung:

1. Andernfalls würden viele Lösungen den Endbenutzern nie wirklich zugute kommen
 - weil sie **erläuterungsbedürftig** sind (→ Dokumentation)
2. Andernfalls werden viele **Probleme übersehen**
 - z.B. bei der Erlernbarkeit
3. Andernfalls können viele Probleme nicht adäquat gelöst werden
 - z.B. kann ein Benutzbarkeitsproblem evtl. nicht in der SW gelöst werden, aber mit einem geeigneten Training abgemildert
 - dafür muss die Erkenntnis aus Benutzbarkeitstests zu den Autor/inn/en der Trainings übermittelt werden.

- Auf Befragen stufen erfahrene Entwickler/innen diese Prinzipien meist als "offensichtlich" ein
- Tatsächlich werden sie aber nur sehr unvollständig angewandt:
 - Wenig direkte Einbindung von Benutzer/inne/n
 - Selten werden mehrere Iterationen vorgesehen

- Entwickler/innen glauben, Benutzer/innen wüssten nicht was sie brauchen
 - Stimmt oft anfangs,
 - löst sich aber mit einem Prototyp schnell
- Entwickler/innen glauben, dass sich Entwurfsprobleme durch Überlegen lösen lassen
 - Kann gelegentlich gelingen, aber meist übersieht man etwas
- Entwickler/innen glauben, die richtige Lösung auf Anhieb finden zu können
 - Kann gelegentlich gelingen, aber meist übersieht man etwas
- Entwickler/innen verweisen auf Halsstarrigkeit von Benutzer/inne/n
 - Stimmt manchmal anfangs,
 - wird aber durch echte Zusammenarbeit meist aufgebrochen

- Steven E. Poltrock, Jonathan Grudin:
*"Organisational obstacles to interface design and development:
Two participant observer studies"*,
ACM Transactions on Computer-Human Interaction 1(1):52–80,
March 1994
 - reprinted in *Computers and Controversy* (2nd ed.) as
*"Interface development in a large organization:
An observational study"*



Jonathan Grudin

- Große Softwarefirma
- Entwicklungseinheit mit 70 Personen
- Baut neue Version eines wichtigen, international vertriebenen Produkts
- Forscher wurde für 1 Monat Mitglied einer UI-Gruppe
 - nahm an der Entwicklung teil
 - interviewte 25 Personen innerhalb und außerhalb der Gruppe

User Interface
(Bedienschnittstelle,
Benutzungsschnittstelle)

- Ein CAD-Programm
 - Ursprünglich eine Hochschulentwicklung
 - Seit 5 Jahren erfolgreich am Markt
- Fortentwicklung wurde allmählich immer schwieriger
 - vor längerem war schon einmal ein Projekt zur kompletten Neuentwicklung begonnen worden, wurde aber nach 1 Jahr abgebrochen
 - 2 Jahre Schwanken zwischen Neuentwickeln und Weiterpflegen
 - Dringlichkeit wurde deutlich, Personalausstattung erhöht
 - einige Verbesserungen erzielt, aber keine langfristige Lösung
 - Hoher Grad von Entscheidungsunfähigkeit:
 - Rücknahme von Entscheidungen aus Marketinggründen,
 - mehrfache Reorganisationen und Personalkürzungen

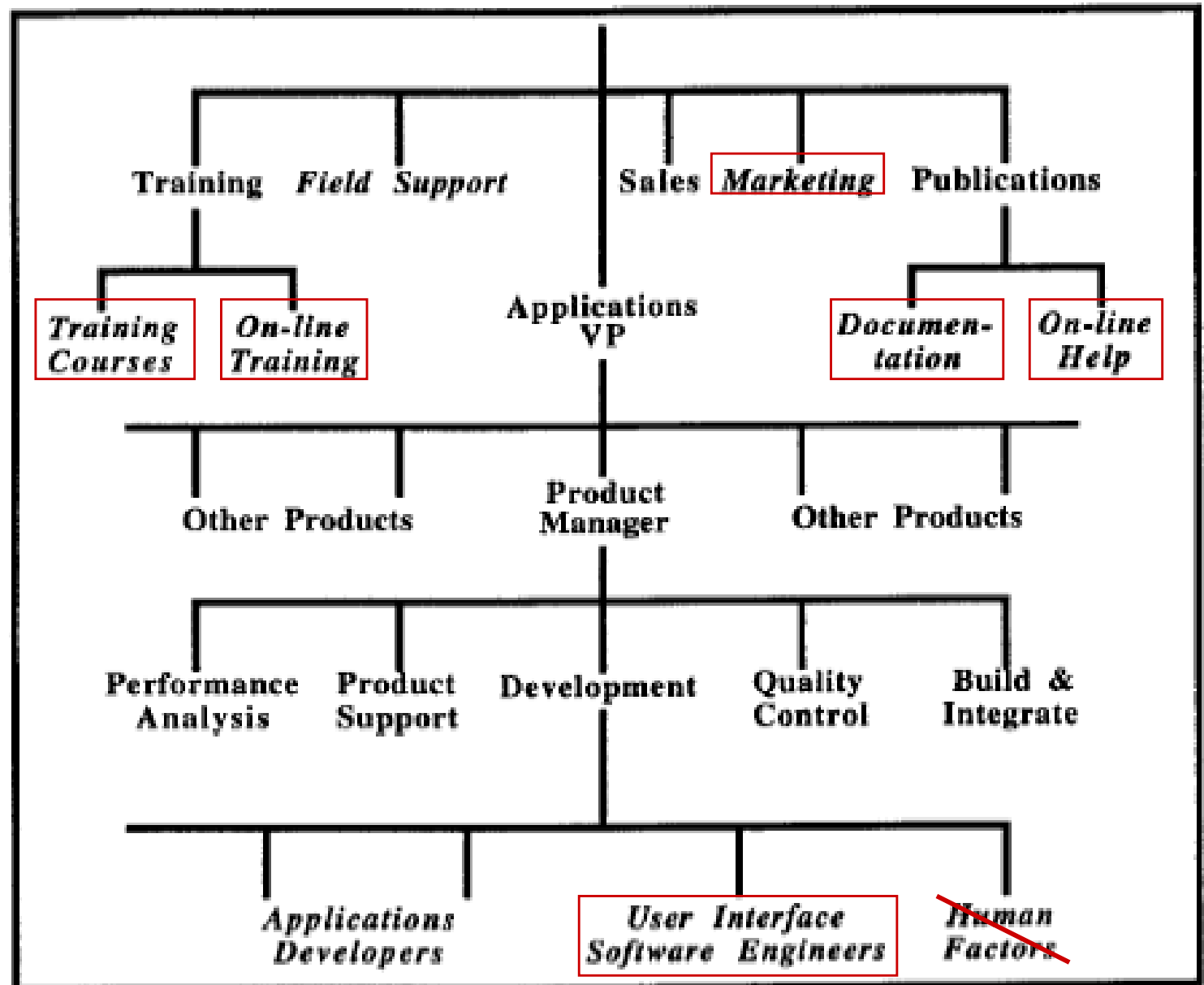


- Dann wurde ein "Superentwerfer" ins Team gebracht:
 - aus der Supportgruppe eines anderen Standorts
 - hervorragende technische Fähigkeiten
 - hatte Erfahrung mit der Anpassung des Produkts für einen großen Kunden; viel Kontakt mit Manager/inne/n und Endbenutzer/inne/n
- Er überzeugte einen Vizepräsidenten davon, binnen 1 Jahr eine sehr gute neue Version bauen zu können
 - hatte eine klare Vision
 - begann, sich einfach Mitarbeiter/innen zu suchen und ohne Zustimmung ihrer Manager/innen Aufgaben zu geben
 - täglicher Abgleich von Fortschritt und Problemen
 - volle Rückendeckung vom Vizepräsidenten half gegen die Beschwerden
- Ergebnis: Sehr erfolgreiche neue Version entwickelt

- Dieses Verfahren wurde jedoch nicht aufrecht erhalten
- Zustand bei Beginn der Studie:
 - Der Vizepräsident war weg
 - Der Superentwerfer ging gerade weg
 - Neueingestellte Manager hatten Fokus auf Standard-Entwicklungspraktiken mit maximaler Überwachung des Entwurfs und viel Koordination
 - bewerteten Zuverlässigkeits- und Planungsziele höher als Innovation
- Neues Team zur Entwicklung der Benutzungsschnittstelle
 - Keine Produktbenutzer/innen unter den Entwickler/inne/n
 - Auch keine Erfahrung in der Zusammenarbeit mit Kund/inn/en
 - Keine Möglichkeit zur Einbeziehung von Endbenutzer/inne/n
 - Keine Benutzbarkeitstests waren geplant

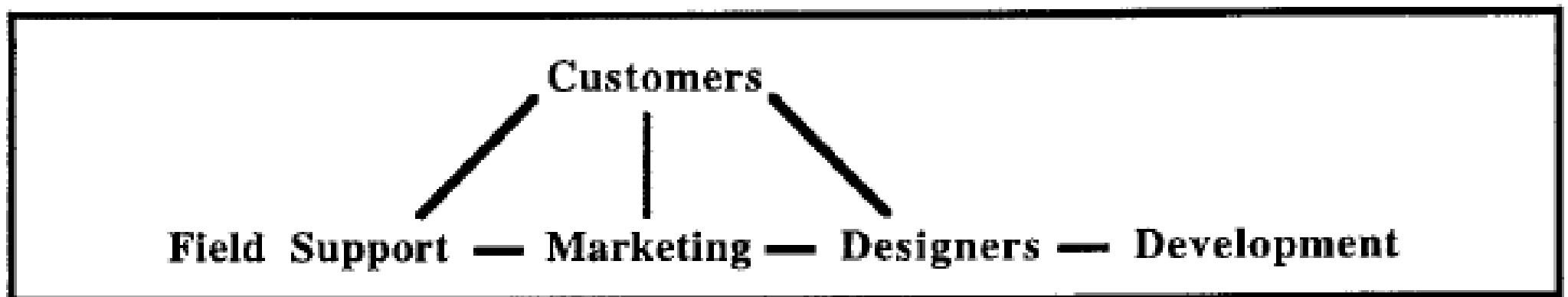
Organisationsstruktur

- ungefähr so vorgefunden
 - mit gewissen Abweichungen
- Prinzip 4 (integrierte Entwicklung) ist direkt verletzt!



Theorie der UI-Entwicklung dieser Organisation

- Marketingabteilung arbeitet mit Vertriebsabteilung ("sales") und Kunden, um die Marktanforderungen zu verstehen.
 - Außerdem mit Kundendienst ("field support")
- Entwurfsabteilung ("designers") empfängt diese und arbeitet mit Kunden, um die technischen Details der Anforderungen zu verstehen
- Entwicklung ("development") bekommt eine komplette, aber abstrakte Funktionsspezifikation von Design
 - entwickelt konkrete UI-Spezifikation (dann: Begutachtung, Bau)
 - muss nicht selbst mit Endbenutzer/inne/n arbeiten



- *Wahrnehmung der Insassen: "The general consensus is that our Marketing group is next to worthless"*
- Aussage eines Endbenutzers
 - ein Entwickler traf ihn auf einer Messe
 - *"Wie ihr über Produktfeatures entscheidet ist wirklich eine Schande: Ich kann nur mit Marketing reden. Marketing versteht nicht, was ich will. Design versteht nicht, was Marketing aufgeschrieben hat. Dann kriegt es Entwicklung, weiß nicht, was es bedeuten soll, weiß nicht, wo es herkommt, und kann mich nicht kontaktieren, um zu fragen."*
- Die Entwickler/innen versuchen das Problem zu umgehen:
 - Sie reden mit dem Kundendienst (Entwurfsdurchsichten etc.)
 - Leider wurde der dann in die Marketing-Abteilung verlagert und die sperrte diese Interaktionen

- Die Aufgabe der Marketingabteilung ist es, zu definieren, was gebraucht wird, um das Produkt zu verkaufen
 - Dazu reden Sie *unter anderem* mit Kunden
 - Aber meist mit den Entscheider/inne/n, nicht mit den Benutzer/inne/n
- Das ist jedoch weitaus zu grob, um eine im Detail nützliche Systemgestaltung vorzugeben
 - von gut benutzbar ganz zu schweigen
 - zumal weder das Personal von Marketing noch die Entscheider/innen bei den Kunden viel von UI-Gestaltung verstehen
- Genaue Funktionsdefinition erfordert also separate Kommunikationskanäle
 - Das gilt erst recht für die UI-Definition

Problem 2: Kein Kontakt zwischen Entwicklung und Benutzern

- Marketingabteilung über Entwicklungsabteilung:
 - *"Die Entwicklung kann keine benutzerfreundliche Software bauen, denn die wissen gar nicht, was ein Benutzer ist."*
- Entwicklungsabteilung über Designabteilung:
 - *"Sie geben uns eine Lösung vor und sagen nicht, was eigentlich das Problem ist."*
- Entwicklungsabteilung über Kontakte zu Kundendienst:
 - *"Gerade vorhin habe ich mit einem über ein Feature gesprochen; was er denkt, wie das aussehen sollte. ' Jetzt weiß ich, dass ich alles wegwerfen und von vorn anfangen muss."*
- Lösungsversuche der Entwicklungsabteilung:
 - Unterhielt informelle Kontakte zu Kunden
 - aber meist wiederum nur zu Entscheider/inne/n

Folgen für die Anwendung der 4 Prinzipien

1. Direkter Kontakt zu Benutzern
 - Fehlt. Das ist das Ausgangsproblem
 2. Frühes und fortlaufendes Benutzbarkeitstesten
 - Ist folglich allenfalls mit unechten Benutzer/inne/n möglich
 3. Iterativer Entwurf
 - Ist nur mäßig nützlich, da man kein qualifiziertes Feedback hat
 4. Integrierter Entwurf
 - Wird von der Organisationsstruktur extrem erschwert
- Diskussion siehe nächste Folien

- Entwicklungsabteilung präsentierte ihre Benutzungsschnittstelle vor Vertreter/inne/n der anderen Abteilungen
 - Problem: *"Unqualifizierte Leute machen einen Haufen Kommentare über alles mögliche, das ihnen gerade dazu einfällt."*
 - *"Wenn man ein Architekturmodell vorstellt, hat kaum jemand dazu was zu sagen. Aber wenn man Bildschirmwürfe zeigt, bekommt man so viele Meinungen wie Leute im Raum sind."*
- Am nützlichsten war noch, als zwei Mitarbeiter/innen des Kundendienstes das Produkt so ausprobierten, wie sie glaubten, dass Endbenutzer/innen es verwenden würden
 - (Sie fanden dabei außerdem mehr Defekte als die Qualitätssicherung es tat!)
 - Dies war aber kein Bestandteil des regulären Entwicklungsprozesses

- Diese Organisation hatte eine lange Tradition der getrennten Entwicklung von HW, SW, Dokumentation und Training
- Die ganze Kommunikation sollte eigentlich über Spezifikationen stattfinden
 - Das ist für Benutzungsschnittstellen jedoch sehr schwierig:
"Man kann nicht in Worten ausdrücken, was man vorhat, wenn man ein bestimmtes UI entwirft. Man kann zwar eine Spezifikation davon aufschreiben, aber keiner weiß, ob die richtig ist, bevor er sie benutzt hat."
- Diese Organisationsform erzeugt auch Widerstände gegen Iterationen
 - oder führt dazu, dass Dokumentation und Lehrmaterial erst sehr spät erstellt werden

- Rolle von Marketing:
 - *"Marketing sollte eine Richtung vorgeben; Gebiete die zu verbessern sind. Und dann ein paar Kunden nennen, mit denen wir direkt zusammen arbeiten können."*
- Art und Rolle von Benutzerkontakt:
 - *"Es wäre gut, wenn jeder Entwickler ein paar Stunden pro Jahr direkt zugucken könnte, wie Endbenutzer die SW verwenden. Nur zugucken. Sehen, was der Benutzer tut, und hören, wo und was er grummelt."*
 - *"Man muss zuerst mal verstehen, was Benutzer überhaupt erreichen wollen. Wie sie es angehen, ihren Tag strukturieren. Mit einem solchen Überblick bekommt man auch Ideen, wie man ihnen am besten helfen kann."*

1. Trage den Mitarbeitern möglichst nur Dinge auf, die sie gut können
 - Viele Tätigkeiten können nur von einem *Team* mit verschiedenen Rollen bewältigt werden
 - Sie einer Person zuzuweisen führt zu schlechten Ergebnissen und zu Reibungsverlusten
2. Sorge für kurze Kommunikationspfade
3. Sorge für eine zentrale Entscheidungsinstanz
4. Hindere die Mitarbeiter/innen nicht, das zu tun, was nötig ist!

Ende der Fallstudie

Fallen überwiegend in vier Kategorien:

- **Defekte:**
 - Eine Funktion tut eindeutig etwas anderes als vorgesehen
- **Funktionslücken:**
 - Eine eindeutig benötigte Funktionalität ist gar nicht vorhanden
- **Verständnisprobleme:**
 - Lernproblem: Das Konzept einer Funktion oder Bedienweise ist schwierig zu erfassen
 - Benutzungsproblem: Eine Darstellung ist auch für geübte Benutzer oft mehrdeutig und provoziert Fehler; eine Eingabeweise provoziert Fehler
- **Bedienschwierigkeiten:**
 - Eine Funktion ist zu umständlich zu benutzen
 - Eine Bedienweise provoziert Fehler
- (Die Zuordnung kann mehrdeutig sein)

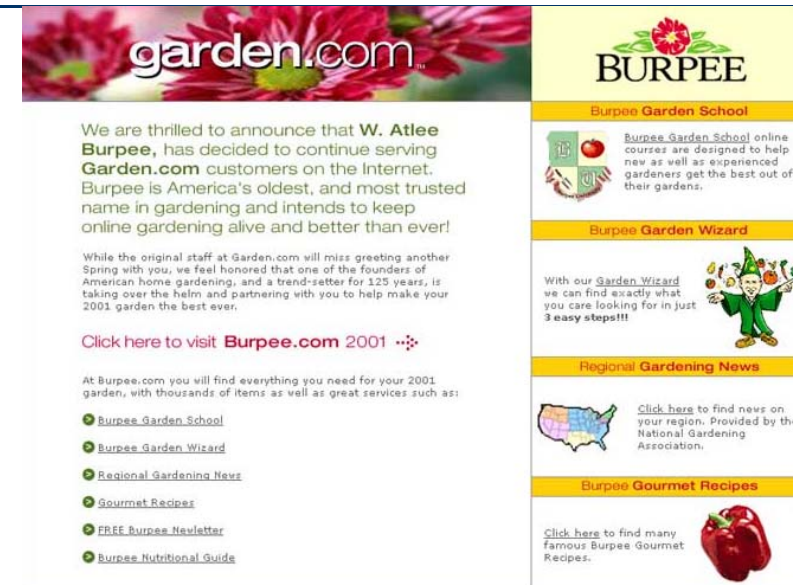
Ergebnisse: Beispiel

Benutzbarkeitsprüfung eines Web-Ladens für Gartenartikel

<http://ucs.ist.psu.edu>,
Case study "Garden-com"

Bereich: Produktkatalog

- Defekte, Funktionslücken: --
- Verständnisprobleme (Erlernen):
 - Mehrdeutigkeit zwischen "Community" und "Meine Sachen"
 - Benutzer verwenden nicht Link "Varianten" wenn ein angezeigtes Produkt ihnen nicht direkt passt
- Verständnisprobleme (Benutzen):
 - Man sucht oft im falschen Teil des Produktbaums
 - Kategorienamen im Baum sind anders als dann angezeigt
 - Sinnarme Kategorienamen (z.B. "Geschenke")
- Bedienschwierigkeiten: --



Beispiel-Ergebnisse: Suchfunktion

Bereich: Suchfunktion

- Defekte: --
- Funktionslücken: --
- Verständnisprobleme (Erlernen):
 - Wenn ein Suchfilter angewendet wurde, wird das nicht direkt angezeigt
- Verständnisprobleme (Benutzen):
 - Es ist unklar, was die Suchfunktion alles durchsucht
 - Suche verlangt oft einen ganz bestimmten Suchbegriff; zu unflexibel
- Bedienschwierigkeiten:
 - Benutzer geben oft falsch geschriebene Suchbegriffe ein (und merken dann evtl. nicht, dass das der Grund für leere Suchergebnisse ist)

Bereich: Einkaufskorb-Funktion ("Schubkarre")

- Defekte:
 - Anfänglich ist als Menge für einen Artikel immer "0" gewählt
 - Benutzer können Artikeleigenschaften (z.B. Größe) auf unmögliche Werte ändern
- Funktionslücken: --
- Verständnisprobleme (Erlernen):
 - Benutzer finden die Einkaufskorb-Funktion oft nicht
 - Benutzer finden die Funktion "Entferne Artikel" nicht und löschen statt dessen die ganze Bestellung
- Verständnisprobleme (Benutzen):
 - Die Auswahl eines Stoffes wird immer mit "Kissen-Bezugsstoff" abgefragt, auch wenn der Artikel gar keine Kissen hat
- Bedienschwierigkeiten: --

Beispiel-Ergebnisse: Bezahlen

Bereich: Bezahl-Funktion

- Defekte: --
- Funktionslücken:
 - Wenn ein Benutzer versehentlich seine Bestellung löscht, gibt es keinen Weg, sie zurück zu bekommen
- Verständnisprobleme (Erlernen):
 - Benutzer klicken oft ohne Wirkung auf Kreditkarten-Symbol
- Verständnisprobleme (Benutzen): --
- Bedienschwierigkeiten: --

Beispiel-Ergebnisse: Hilfesystem

Bereich: Online-Hilfesystem

- Defekte: --
- Funktionslücken:
 - Es gibt kein Hilfesystem, obwohl Benutzer eines erwarten und es auch brauchen.
- Verständnisprobleme (Erlernen): --
- Verständnisprobleme (Benutzen): --
- Bedienschwierigkeiten: --

Beispiel-Ergebnisse: Sonstiges

Bereich: Sonstiges

- Defekte: --
- Funktionslücken: --
- Verständnisprobleme (Erlernen): --
- Verständnisprobleme (Benutzen):
 - Der Farbunterschied zwischen neuen und bereits benutzten Links ist zu gering
- Bedienschwierigkeiten:
 - Es ist umständlich und schwierig, einen Artikel zu finden/zu bestellen, der außerhalb des Einkaufsbereichs erwähnt wird

- Software mit guter Benutzbarkeit zu bauen verlangt
 - direkten Kontakt zu Benutzern sowie
 - iterative Entwicklung mit fortlaufenden Benutzbarkeitsprüfungen
- Leider ist das in konkreten SW-Organisationen oft schwer zu gewährleisten
- Die Ergebnisse von Benutzbarkeitsprüfungen fallen in diverse Kategorien und geben gute Hinweise für Verbesserungen



V+Ü User-Centered Design: Benutzerzentrierte Softwareentwicklung

Ziele:

Prof. Claudia Müller-Birn

Julia Schenk

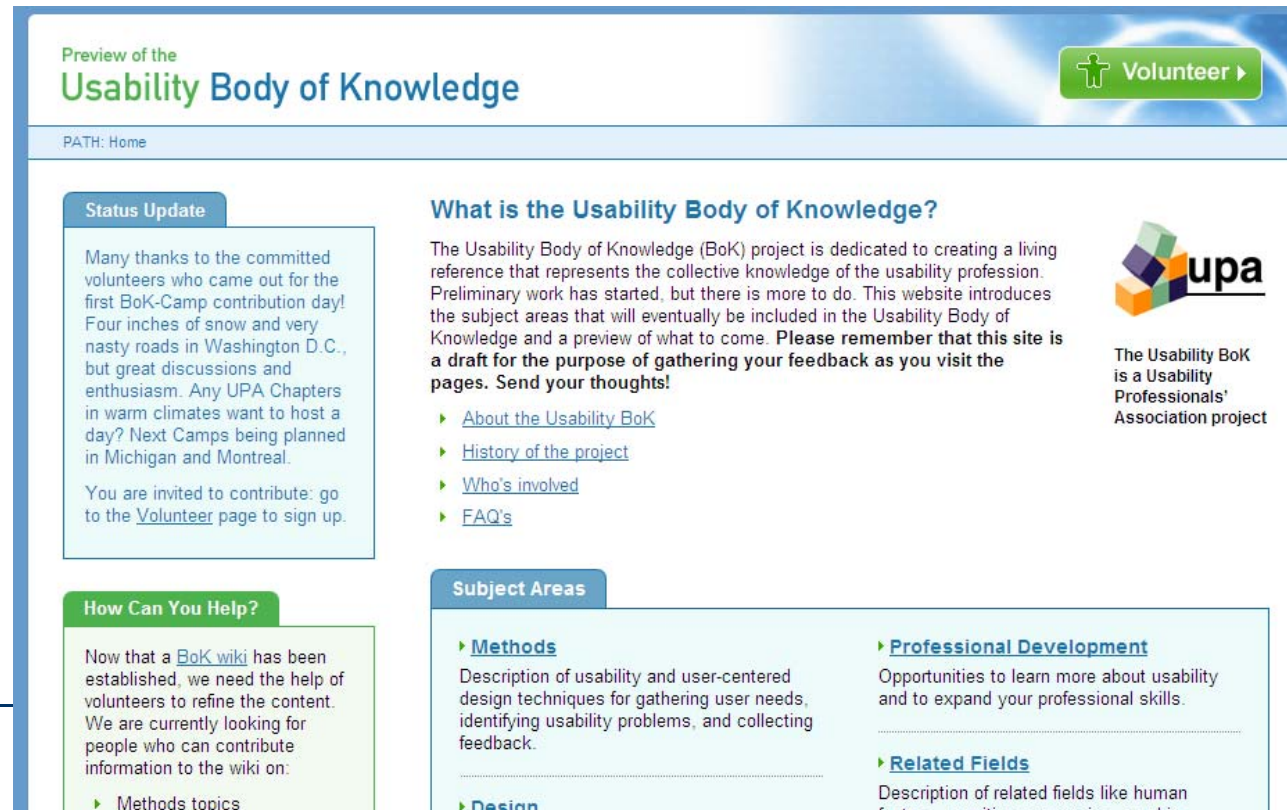
- Wissen, was ein positives Benutzungserlebnis ausmacht.
- Wissen wie man Software mit guter Usability entwickelt.
- Usability bestehender Software bewerten können.

Themen

- Den **Benutzer** und dessen Ziele **verstehen**: User Research
- Den **Faktor Mensch** berücksichtigen: Wahrnehmung und Kognition und deren **Relevanz für Interface Design und Interaction Design**
- **Konkrete Anwendungsgestaltung**: Interface Elemente, Interface Design Patterns und deren Einsatzbereiche
- Usability-Prüfung mit Benutzern: **Usability Tests**
- **Usability-Prüfung durch "Experten"**: z.B. Usability Heuristiken

Jakob Nielsen. "Usability Engineering". Boston: Academic Press, Inc. 1993.

- Usability Body of Knowledge
 - <http://www.usabilitybok.org>
 - Gute und übersichtliche Erläuterung der wichtigen Konzepte:
 - Erhebungs- und Evaluationsmethoden
 - Entwurfsprinzipien
 - Organisations- und Fortbildungsfragen
 - verwandte Wissensgebiete, Institutionen
 - Enthält weitere Literaturverweise



The screenshot shows the homepage of the Usability Body of Knowledge (BoK) project. The page has a blue header with the title "Preview of the Usability Body of Knowledge" and a "Volunteer" button. Below the header, there is a "Status Update" section with a text box containing a message of thanks to volunteers and an invitation to contribute. To the right, there is a "What is the Usability Body of Knowledge?" section with a paragraph of introductory text and a list of links: "About the Usability BoK", "History of the project", "Who's involved", and "FAQ's". On the far right, there is a logo for "upa" (Usability Professionals' Association) and a short description of the project. At the bottom, there is a "Subject Areas" section with two columns of links: "Methods" (with a description of usability and user-centered design techniques) and "Professional Development" (with a description of opportunities to learn more about usability). A "How Can You Help?" section is also visible at the bottom left, with a text box asking for volunteers to refine the content of a BoK wiki.

Danke!