

Usability Engineering

von Thomas Tigges & Jens Willmer

Usability Engineering

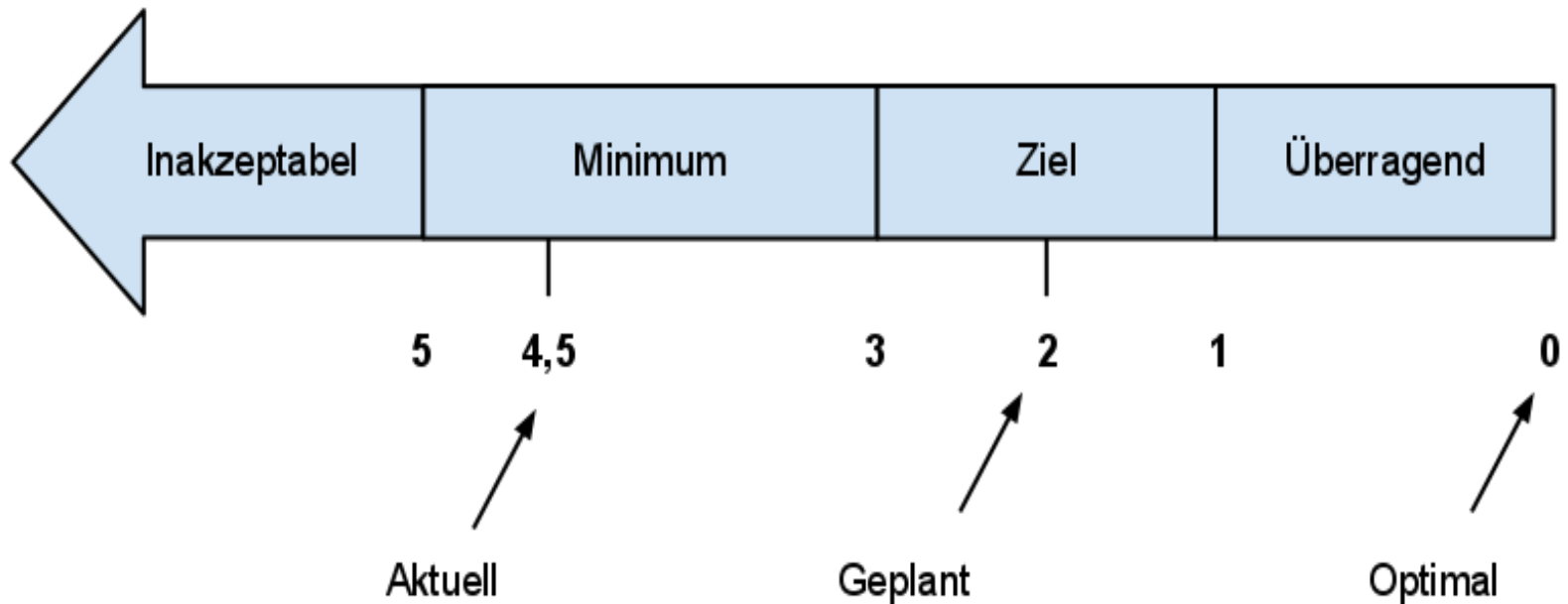
Usability Engineering bezeichnet ein Gebiet, in dem es darum geht eine optimale Mensch-Maschinen-Schnittstelle zu entwickeln. Bei einer solchen Schnittstelle liegt das Hauptaugenmerk auf der Benutzerfreundlichkeit. Die entwickelte Schnittstelle soll möglichst effizient für die vorgesehenen Arbeitsaufgaben gestaltet sein und zudem ein ansprechendes, gewohntes Aussehen haben, damit der Benutzer gerne damit arbeitet und sich schnell zu Recht findet. Des Weiteren muss die Schnittstelle vorgegebene Standards und Guidelines einhalten.

Kenne den Anwender

- Definieren des Anwenderkreises
 - Kontakt zwischen Anwender und Entwickler
 - Oberfläche an Kenntnisse der Anwender anpassen
- Evolution des Anwenders
 - Aus Laien werden Profis
- Anwender werden Programm verändern
 - Anwender finden neue Anwendungsmöglichkeiten
 - Programm verändert Arbeitsablauf

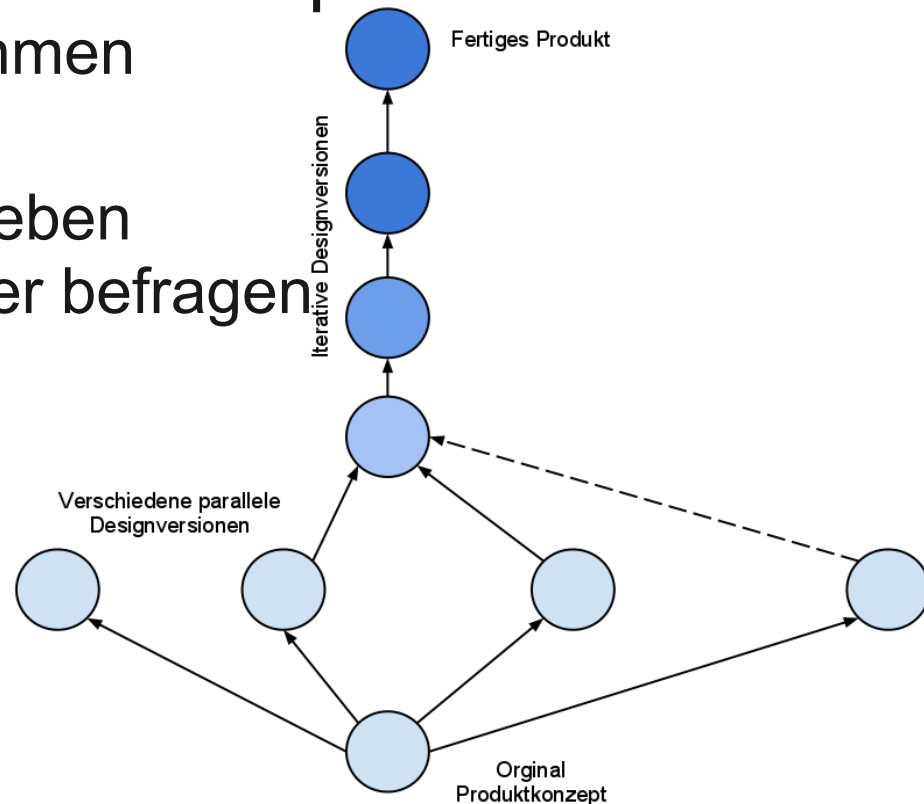
Zielsetzung für die Bedienbarkeit

Anwenderfehler pro Stunde welche das System benutzen:



Designphase

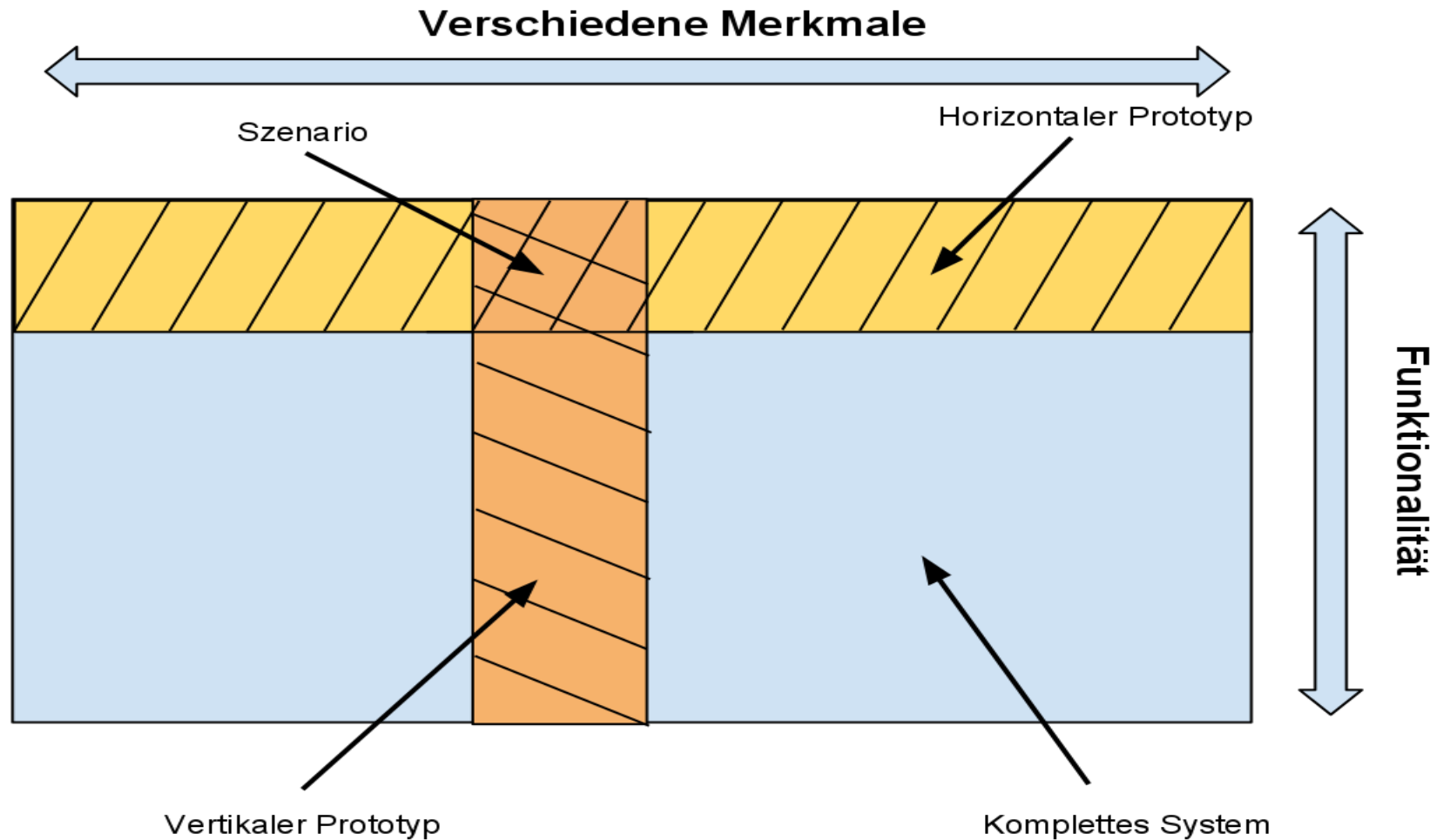
- Designer arbeiten unabhängig an Konzepten
- Zusammenführung der Konzepte
 - Beste Ansätze übernehmen
- Zugriff auf Anwender
 - Aktuelle Probleme beheben
 - Verschiedene Anwender befragen



Guidelines & Standards

- **Standards**
 - Wiedererkennungswert
 - Wiederverwendbarkeit von Anleitungen
 - Wiederverwendbarkeit von Beispielen
- **Guidelines**
 - Farbschema
 - Genaueres Aussehen, Größe und Form

Prototyp



Oberflächen Evaluation

		Wieviel Anwender werden auf das selbe Problem stoßen?	
		Wenige	Viele
Behinderung für den Anwender der das Problem entdeckt hat	Klein	Wenig Gewichtung	Mittlere Gewichtung
	Groß	Mittlere Gewichtung	Hohe Gewichtung

Usability Testing

- Anwendertests
 - Zuverlässigkeit
 - Selbe Ergebnisse bei Wiederholung des Tests
 - Aussagekraft
 - Resultat betrifft die Problemstellung

Testmethoden

- **Laut denken**
 - Aufdecken von Missverständnissen
 - Erkennung des gedanklichen Arbeitsablaufs
- **Konstruktive Interaktion**
 - Zwei Anwender arbeiten zusammen
 - Bevorzugt bei Kindern
- **Retrospektives testen**
 - Videoanalyse
 - Qualitativ hochwertige Antworten
- **Coaching Methode**
 - Schnelles verstehen der Anwendung
 - Beschleunigt den Wandel zum Profi

Gedächtnis

- **Sensorisches Kurzzeitgedächtnis**
 - Puffert die durch die Sinne wahrgenommenen Informationen
 - Kapazität etwa 1 Sekunde
- **Kurzzeitgedächtnis**
 - "Arbeitsspeicher" des Gehirns
- **Langzeitgedächtnis**
 - Speicher alle während des Lebens gelernten Informationen

Kurzzeitgedächtnis

- Unser Bewusstsein
- Aktueller Kontext
 - Neue Informationen werden innerhalb des Kontextes interpretiert
 - Funktionen und Rückmeldungen einer Anwendung müssen zum Kontext (dem Ziel) des Anwenders passen [Google Docs](#)
- Kapazität von 7 ± 2 Chunks
 - Gruppierung kann Informationsgehalt pro Ebene reduzieren [DHBW FFW Mühlingen](#)

Langzeitgedächtnis

- Enthält unser Wissen
- Speichert
 - Fakten, Daten und Konzepte
 - Bilder und Vergleiche
 - Zusammenhänge und Schlussfolgerungen
 - Abläufe
- Gespeichertes Wissen bzw. die Erfahrung müssen beim Entwurf von Anwendungen beachtet werden
 - Öffnen, Speichern, Drucken immer oben links
 - Links sind blau und unterstrichen

Assoziationen

- **Mentale Modelle**
 - Abstraktes Modell dessen wie etwas funktioniert oder zusammenhängt
 - Lernen kann durch Verknüpfungen zu bereits Bekanntem beschleunigt werden
 - Umblättern auf einem Tablett durch Wischen
- **Metaphern**
 - "bildliche Übertragung"
 - Symbole in Funktionsleisten orientieren sich an realen Vorbildern
 - Ziehen einer Datei auf den Papierkorb

Erfahrung

- Unterschiedliches Vorwissen der Anwender
- Zwei Extreme
 - Anfänger
 - Wenig Wissen und Fertigkeiten
 - Aufmerksamkeit muss oft bewusst auf die Bedienung gerichtet werden
 - Experten
 - Ausgeprägtes Wissen und Fähigkeiten
 - Bedienung funktioniert Unterbewusst, dadurch Konzentration auf die eigentliche Aufgabe

Aufmerksamkeit

- Nur bewusst auf jeweils eine Aktivität
- Dank Unterbewusstsein ist Parallelisierung möglich
- Beispiel Autofahren
 - Als Anfänger häufige Konzentration auf Kuppeln, Schalten, usw. notwendig
 - Als Erfahrener Fahrer funktioniert dies Automatisch. Konzentration darauf nichts zu überfahren.
- Entspricht eine Benutzeroberfläche den Erwartungen des Benutzers, so kann diese unterbewusst bedient werden

Intellektuelle Fehler

- Denkfehler
 - Können fast nicht verhindert werden
- Fehler im mentalen Modell
 - Anwendung sollte dem mentalen Modell des Anwenders entsprechen
 - Innerhalb der Zielgruppe kann die Erfahrung sehr stark variieren
 - Funktionen sollten in der Dokumentation beschrieben werden

Fehler in flexiblen Handlungsmustern

- Ausnahmen werden vom Anwender ignoriert
 - Schuld oft beim Anwendungsentwickler selbst!
- Abstumpfung durch zu viele und irrelevante Meldungen
- Unterschiedliche Meldungstypen sollten gestalterisch sofort erkennbar sein
- Rein informative Meldungen auf ein Minimum reduzieren und in Status-/Seitenleiste anzeigen
- Nutzer kann selbst die Meldungen bestimmen

Sensomotorische Fehler

- Mangelnde Koordination zwischen Sensoren und Motorik
- Durch zu kleine Elemente wird an falsche Stelle geklickt
- Schlimmstenfalls muss die Aufmerksamkeit auf das Klicken eines Knopfes gerichtet werden
- Interaktionselemente benötigen angemessene Größe

Antwortzeiten

- Wichtig für die "User Experience"
- Lange Antwortzeiten kosten den Anwender Zeit
- Lange Antwortzeiten durch geschätzte verbleibende Zeit überbrücken
- Pro 0,1 Sekunden entgehen Amazon 1% Umsatz

bis 1 Sekunde	unmittelbar
bis 5 Sekunden	verzögert
bis 10 Sekunden	stark verzögert
ab 10 Sekunden	antwort wird nicht mehr erwartet

Layoutgestaltung

- Klares und einheitliches Layout wichtig für Übersichtlichkeit
- Ein Layout sollte über mehrere Seiten gleich bleiben
 - Navigation, Inhalt, usw. sind immer an der selben Stelle zu finden
- Textbreite sollte 80 Zeichen nicht übersteigen
- Spalten sollten einen Abstand haben

Rasterlayout

- Seite wird in einheitliche Blöcke aufgeteilt
- Ermöglicht strukturierte Anordnung von Elementen
- Positionierung Anhand des Rasters, dadurch geringer Aufwand zur Positionierung von Elementen [Bootstrap YAML](#)

Farben

- Farbwirkung stark durch Kultur und Erfahrung geprägt
- Für harmonische Kontraste bieten sich die beiden Nachbarkomplementärfarben einer Grundfarbe an [Color Scheme Designer](#)

Erhöhte Lesbarkeit durch starken Kontrast vom Text zum Hintergrund

Typographie

- Text sollte maximal 80 Zeichen breit sein
- Kopf sollte beim Lesen nicht mitgedreht werden müssen
- Absätze und angemessene Zeilenabstände erleichtern den Augen die Orientierung [25 Secrets](#)
- Normaler Fließtext sollte 8-12pt groß sein, Überschriften und Schauschrift 13-48pt

Probleme für Usability in Open Source

- Entwickler sind meist Programmierer
- Entwickler sind Poweruser
 - Deren Erfahrung und Erwartung != Zielgruppe
- Entwicklung meist Codezentriert
- Kein Ansporn für Usability
 - Schlecht messbar im Gegensatz zu Bugreports
 - Keine Preise zu gewinnen

Integration von Usability in den Entwicklungsprozess

- Bewusstsein für Usability schaffen
- Gestaltungsrichtlinien erstellen
- Testverfahren wie Remote Testing auch im Open Source Umfeld einsetzbar
- Frühes hinzuziehen von Usability-Experten und Gründen einer Usability-Gruppe mit klaren Entscheidungsstrukturen
- Der Benutzergruppe bewusst werden

Open Source Usability am Beispiel Wordpress

- Seit v2.7 standardmäßige Tests der Module
- Tests mit kleiner Nutzergruppe führten zu unscharfen Ergebnissen
- Entwicklung eines Systems in das Testfälle eingestellt werden und von Benutzern auf der ganzen Welt durchgeführt werden können

Vielen Dank!

Fragen.