



Testbild

fos @ GOR

# Experimentelle Untersuchungen zur Gebrauchstauglichkeit von Web-Applikationen auf Open-Source-Basis

Dr. Herbert A. Meyer

ARTOP

Institut an der Humboldt-Universität zu Berlin

[meyer@artop.de](mailto:meyer@artop.de)

# Überblick

- Einleitung: Open Source & Usability
- Problemstellung
- Methodisches Vorgehen
- Experimentelle Untersuchung 1
- Experimentelle Untersuchung 2
- Ausblick: „Remote Usability Testing“

# Hintergrund: Freie Online Systeme

*Langzeitprojekt zurzeit in einer Ruhephase*

- **Zielsetzungen des Projekts**

1. Schaufenster für Open-Source-Software
2. Thematischer Produktkatalog
3. Zugang zur Erprobung der Systeme
4. Status Quo der Gebrauchstauglichkeit

- **Maßnahmen**

1. Bestandsaufnahme >100
2. Kategoriensystem 12
3. Installationen 62
4. Benutzungstests 32/350

# Open Source & Usability

*Neuland*

Es gibt kaum systematische Untersuchungen zur Gebrauchstauglichkeit von Open-Source-Software!

Ausnahme:

Nichols, D. M.; Twidale, M. B. (2003). The Usability of Open Source Software. *First Monday*, Vol. 8, Nr. 1.

# Open Source & Usability

*Warum ist der Zusammenhang bislang nicht bearbeitet worden?*

- Open-Source-Lösungen werden heute vor allem in den Bereichen **Betriebssysteme, Serveranwendungen, Datenbanken** und **Entwicklungsumgebungen** eingesetzt (z.B. "Linux", "Apache", "MySQL" und "PHP").
- **Software von Experten für Experten:** Es ist hinreichend, wenn die Software nützlich ist und funktioniert!
- Benutzerzufriedenstellung im Sinne einer leichten Erlernbarkeit und einfachen Bedienung steht nicht im Vordergrund der Anwendung, vielmehr ihre **Effizienz**.

# Open Source & Usability

*Warum sollte der Zusammenhang bearbeitet werden?*

- Open-Source-Lösungen, insbesondere Web-Applikationen, zielen immer mehr auf **breite Nutzerschichten**.
- Berücksichtigung der „**Usability-Trias**“ (Effektivität, Effizienz und Benutzerzufriedenstellung) ist angemessen, wenn Software nicht nur von **Experten**, sondern auch von **Anfängern** und **Fortgeschrittenen** benutzt werden soll.
- Gebrauchstauglichkeit ist ein sinnvolles **Auswahlkriterium** bei Open-Source-Lösungen mit vergleichbarer Funktionalität.

# Open Source & Usability

## *Die Kernprobleme*

- **Auch** Open-Source-Entwickler sind keine typischen End-Benutzer.
- **Scratching someone else's itch?** Es geht den Entwicklern in erster Linie darum, eigene Probleme zu lösen, eigenen Juckreiz zu stillen.
- Belohnungen wirken besser auf die Verbesserung der **Funktionalität** als auf die **Gebrauchstauglichkeit**.
- Open-Source ist i.a.R. **nicht-kommerziell** ausgerichtet und für Gebrauchstauglichkeits-Untersuchungen fehlen die finanziellen Mittel.

# Problemstellung

- **Handlungsempfehlungen** zugunsten von gebrauchstauglichen Web-Applikationen über eine **summative Evaluation**. Also: Vergleich von Softwareprodukten innerhalb einer Produktklasse.
- **Ökonomische Verfahrensweise**: Erste Vorarbeiten zu einem Ansatz, der den Aufwand an finanziellen Mitteln möglichst weitgehend reduziert.

# Methodischer Ansatz

- Aufgabenorientierte Benutzungstests unter Einsatz experimenteller Kontrolltechniken
- Weitgehend automatisierter Untersuchungsablauf
  - Jedoch noch Absicherung des Verfahrens durch einen Versuchsleiter vor Ort
  - Also noch kein vollständiger Verzicht auf Hilfestellung und Drop-Out-Verhinderung durch soziale Kontrolle

# Methodisches Vorgehen - UV's

- **Unabhängige Variable:**  
Softwareprodukte innerhalb einer Produktklasse (auf dem Prüfstand)
- **Weitere Kontrollfaktoren:**  
Alter, Geschlecht und Expertise
- = **Quasi-Experiment im „within“-Design**  
mit Ausbalancierung der UV-Bedingungen zur Kontrolle von Carry-Over-Effekten

# Methodisches Vorgehen - AV's

## Abhängige Variable 1

Bearbeitungszeiten von Kernaufgaben

## Abhängige Variable 2

Subjektive Software-Bewertung nach  
DIN EN ISO 9241-10

# Methodisches Vorgehen - AV 1

## Abhängige Variable 1

### Bearbeitungszeiten von Kernaufgaben

#### *Vorarbeiten*

- **Anforderungsanalyse** der Tätigkeiten mit Softwareprodukten einer Produktklasse
- Formulierung und Überprüfung von **Kernaufgaben**
- Implementierung eines **Trackingverfahrens**
- **Online-Auswertung** der Aufgabenbearbeitung als potenzielle Schutzmaßnahme vor „Lurkern“ und zur direkten Rückmeldung der Ergebnisse

# Methodisches Vorgehen - AV 2

## Abhängige Variable 2

Subjektive Software-Bewertung nach  
DIN EN ISO 9241-10

### *Vorarbeiten*

- Auswahl des Benutzerfragebogens **ErgoNorm** (28 Items, DATech e.V.)
- **Digitalisierung und Implementierung** des Fragebogens für eine Web-basierte Anwendung
- **Online-Auswertung** für eine unmittelbare Rückmeldung der Ergebnisse (optional)

# Methodisches Vorgehen - AV 2

## *Sieben Skalen der Norm ISO 9241-10*

- *Aufgabenangemessenheit (9 Items)*
- *Selbstbeschreibungsfähigkeit (6 Items)*
- *Steuerbarkeit (5 Items)*
- *Erwartungskonformität (3 Items)*
- *Fehlerrobustheit (3 Items)*
- *Individualisierbarkeit (1 Item)*
- *Erlernbarkeit (1 Item)*

# Methodisches Vorgehen - AV 2

## *Item 1 Selbstbeschreibungsfähigkeit*

1. Sind die Informationen, die zur Erledigung der Aufgabe notwendig sind, auf dem Bildschirm übersichtlich verfügbar?

ja       nein       Frage trifft nicht zu

**wenn "nein":**

Nennen Sie bitte die Informationen, die Sie benötigen, aber nicht "auf einen Blick" zur Verfügung stehen.

---

---

---

Ich empfinde dies als sehr störend

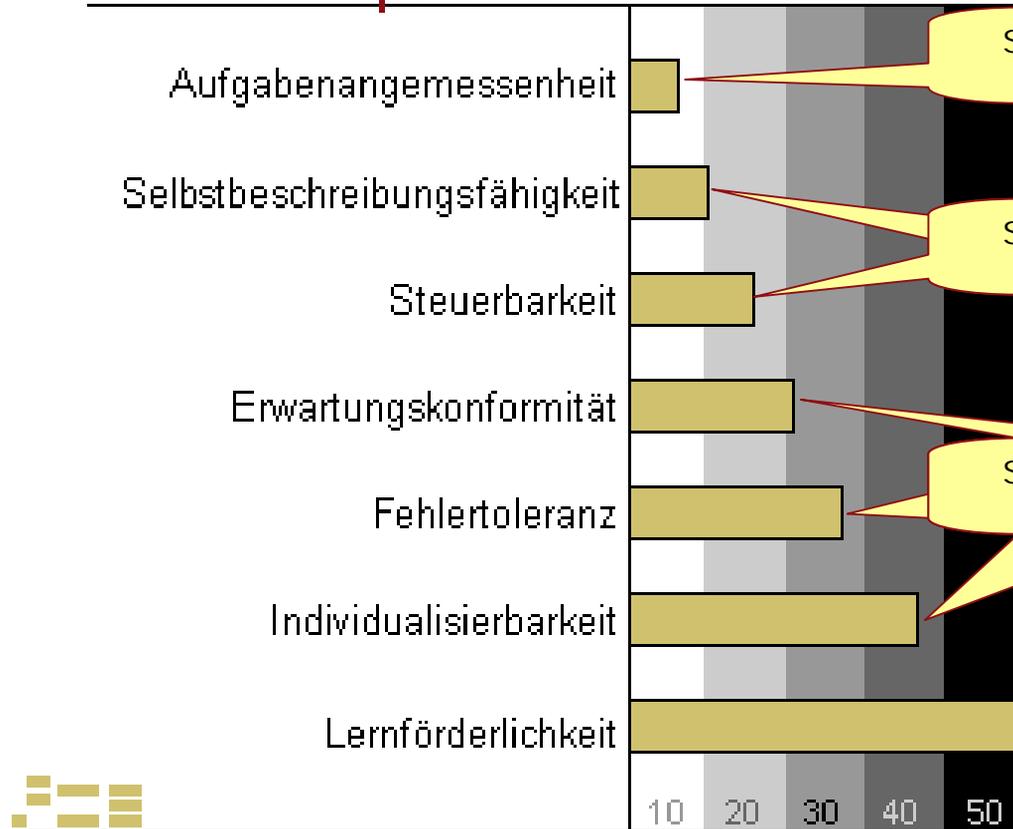
# Methodisches Vorgehen: Profil

Subjektive Software-Bewertung  
nach DIN EN ISO 9241-10

Name des untersuchten  
Version des untersuchten  
Softwareprodukts

## Softwareprodukt

Version



Subjektive Beschwerden  
sind unbedenklich!

Subjektive Beschwerden  
sind tolerierbar!

Subjektive Beschwerden  
sind sehr bedenklich!

Anzahl der von Benutzern im Schnitt berichteten Probleme (in Prozent)  
Subjektive Software-Bewertung nach DIN EN ISO 9241-10

# Methodisches Vorgehen

## *Durchführung*

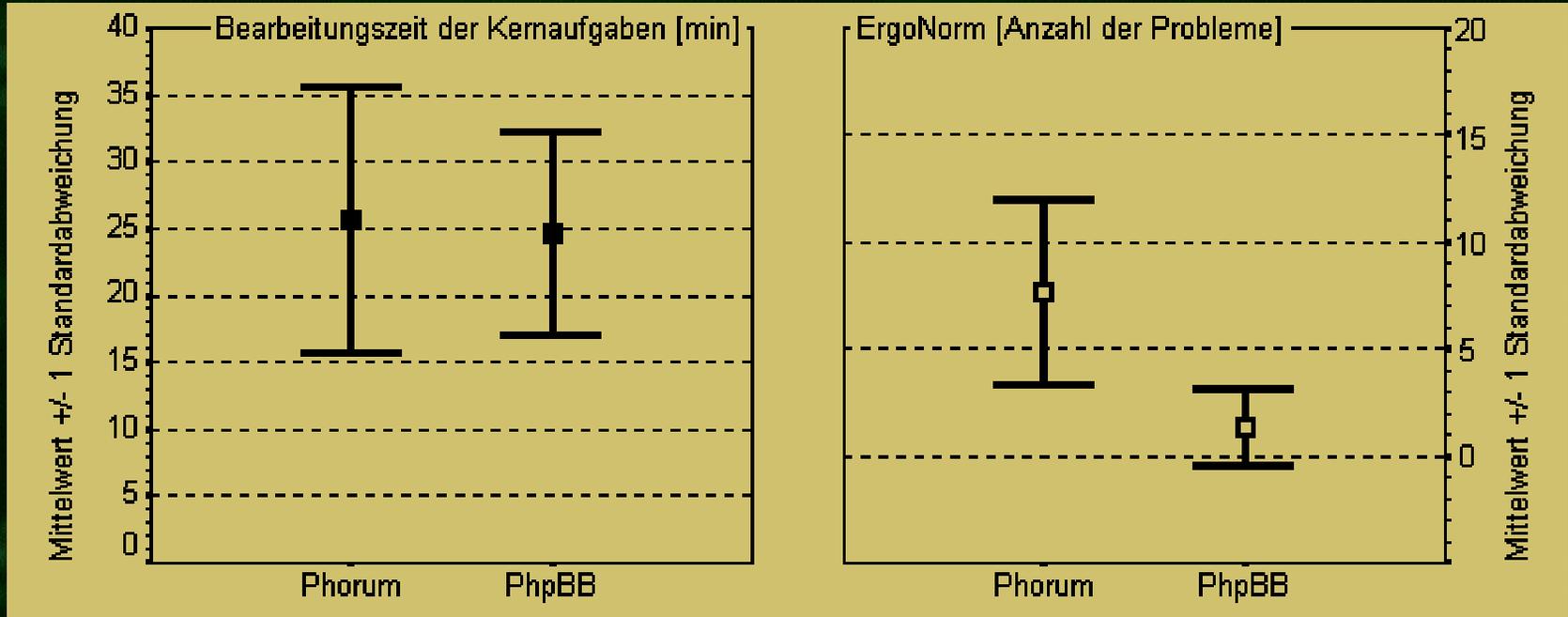
- Qualitatives Interview
- 10-15 Min. Exploration
- 20-40 Min. Aufgabenbearbeitung
- 10-30 Min. Software-Bewertung
- Abschlussbesprechung

# Methodisches Vorgehen

## *Teilnehmende*

- 12 Probanden je Softwareprodukt (6 Frauen und 6 Männer)
- 29 Jahre (Range 19-41)
- 7 Jahre Web-Nutzung, 6 Jahre Produktion von Web-Angeboten, 50% Vorerfahrung mit Administration, 66% Programmiersprachenkenntnisse
- Aufwandsentschädigung 12 €/Std.

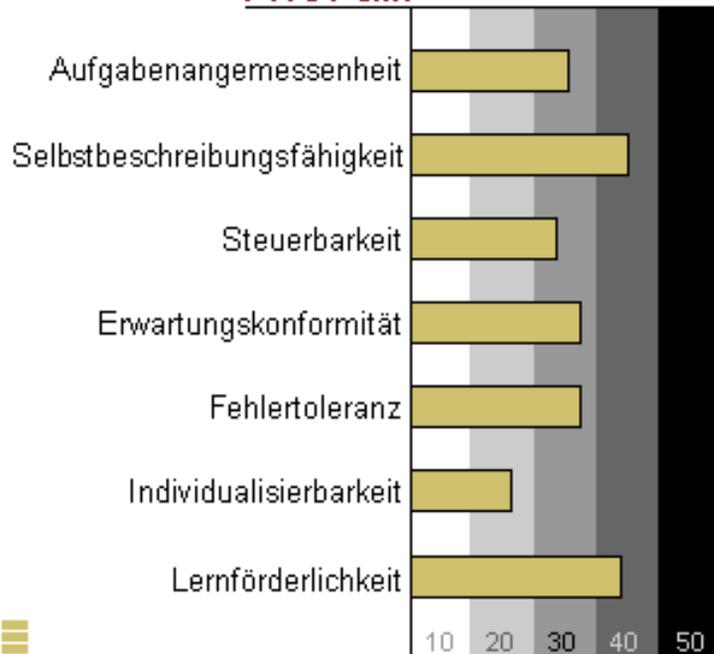
# Ergebnisse Foren



# Ergonorm Foren

## Phorum

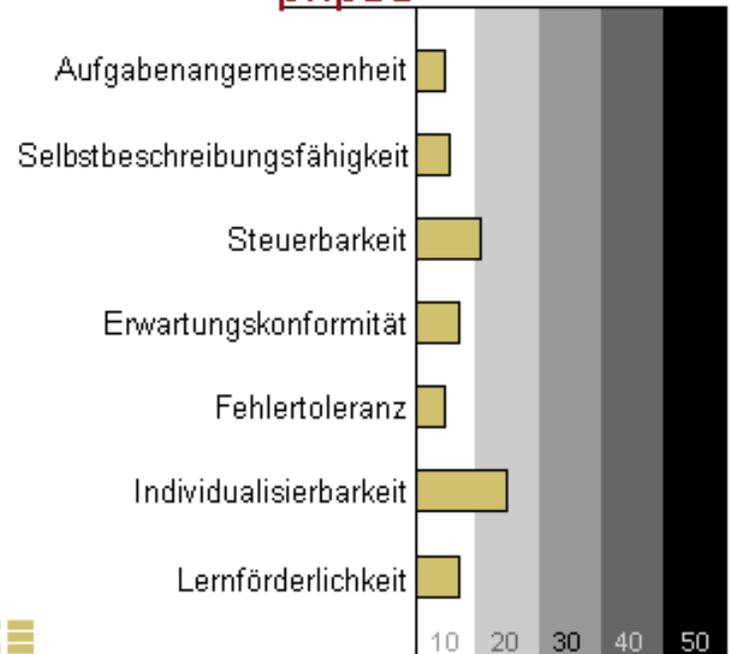
Version 3.3.2a



Anzahl der von Benutzern im Schnitt berichteten Probleme (in Prozent)  
Subjektive Software-Bewertung nach DIN EN ISO 9241-10

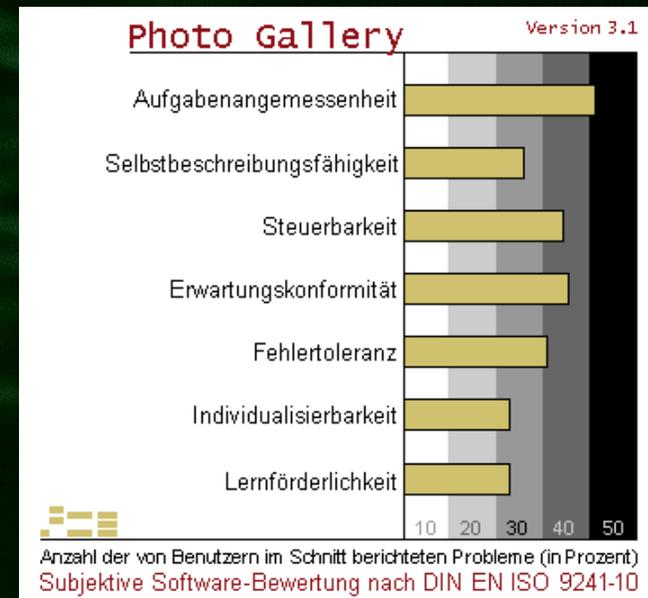
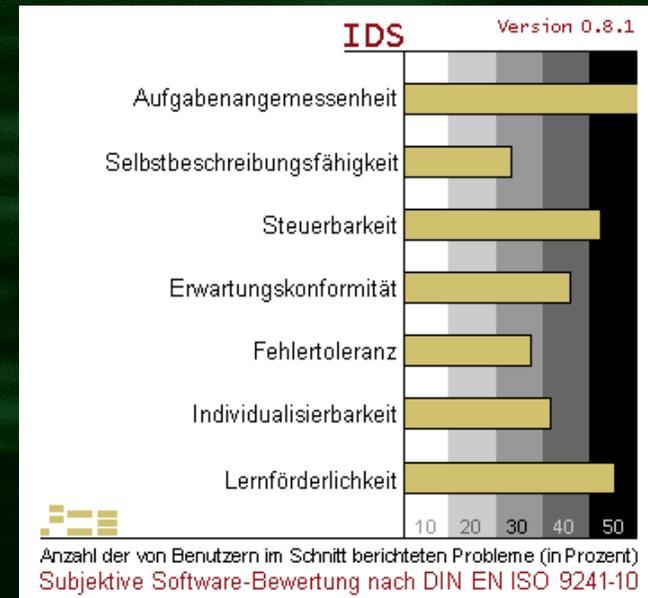
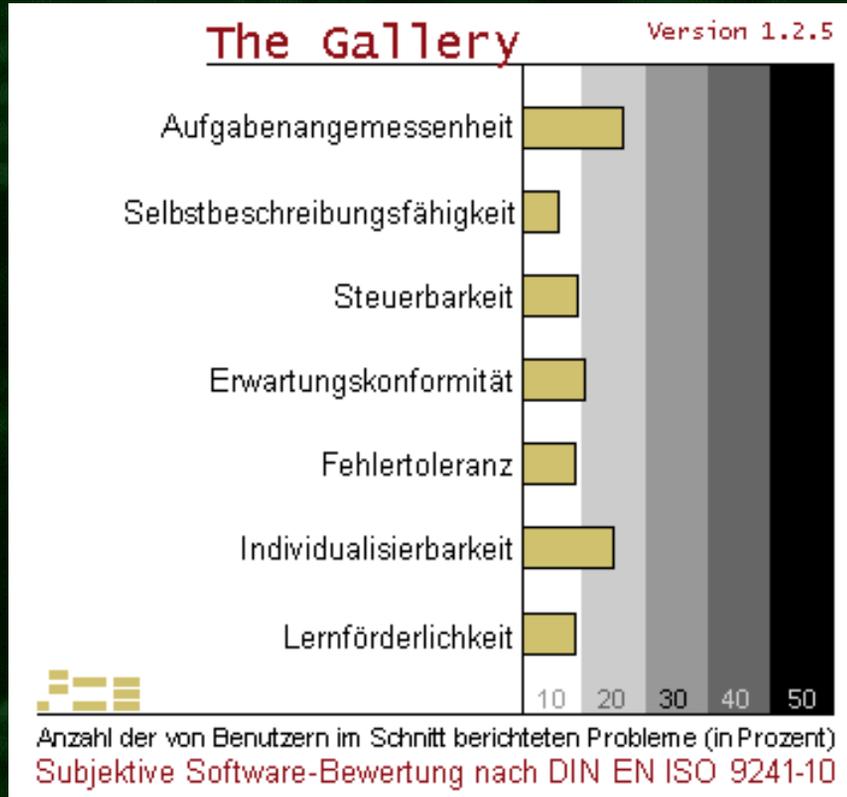
## phpBB

Version 2.01

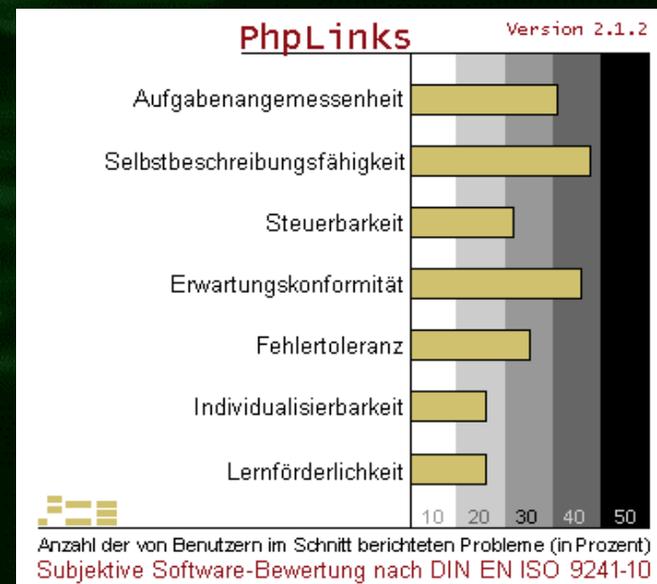
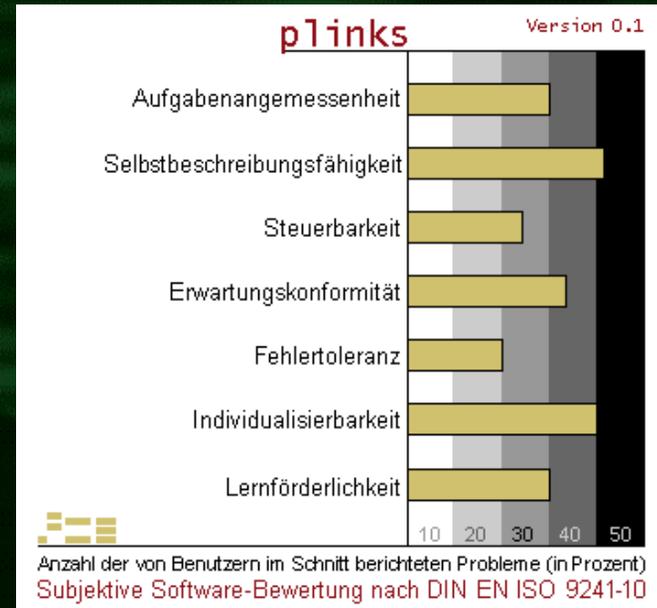
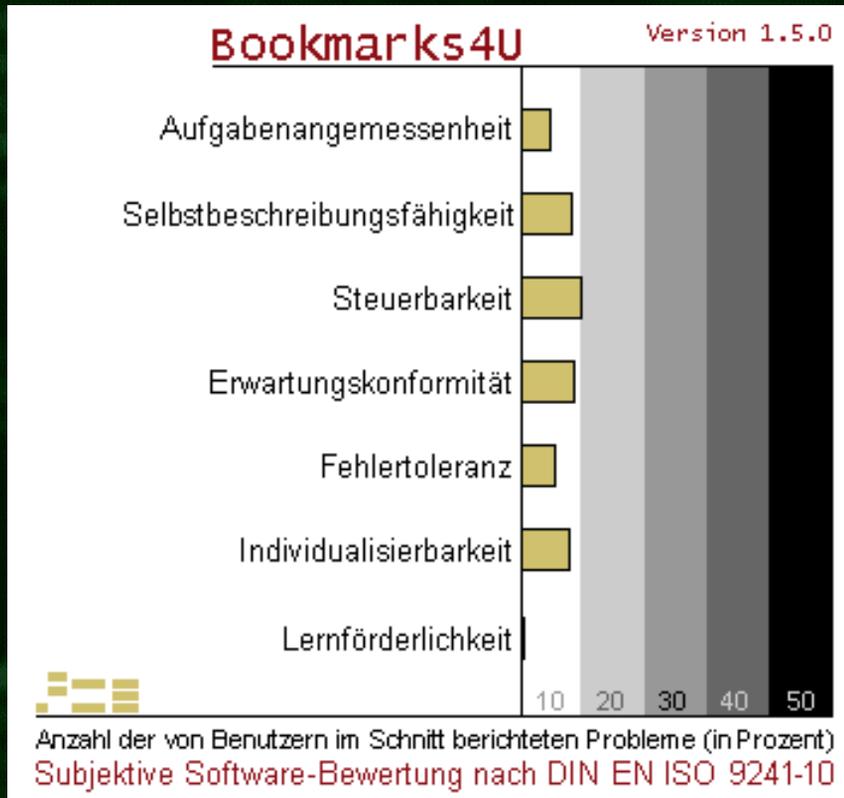


Anzahl der von Benutzern im Schnitt berichteten Probleme (in Prozent)  
Subjektive Software-Bewertung nach DIN EN ISO 9241-10

# Ergonorm Web-Alben



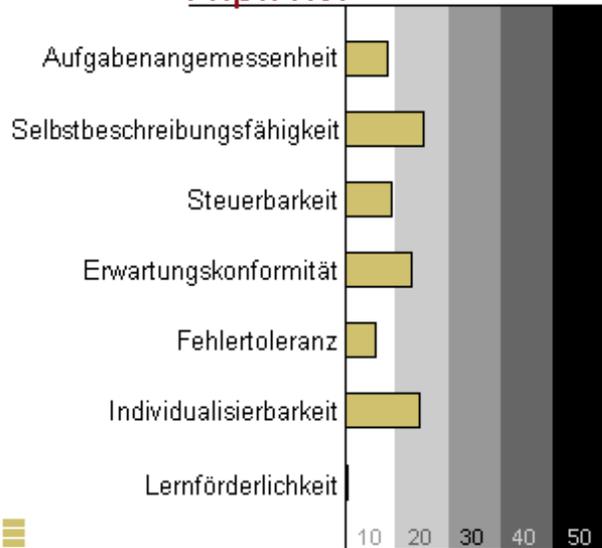
# Ergonorm Bookmark



# Ergonom Wiki-Wiki

## Phpwiki

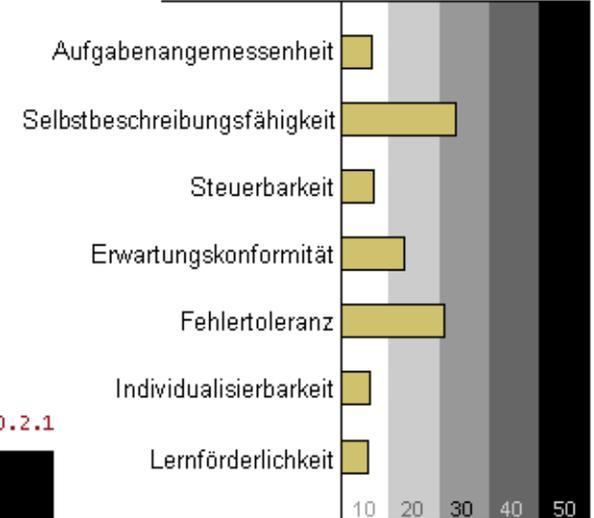
Version 1.03



Anzahl der von Benutzern im Schnitt berichteten Probleme (in Prozent)  
Subjektive Software-Bewertung nach DIN EN ISO 9241-10

## MoinMoin

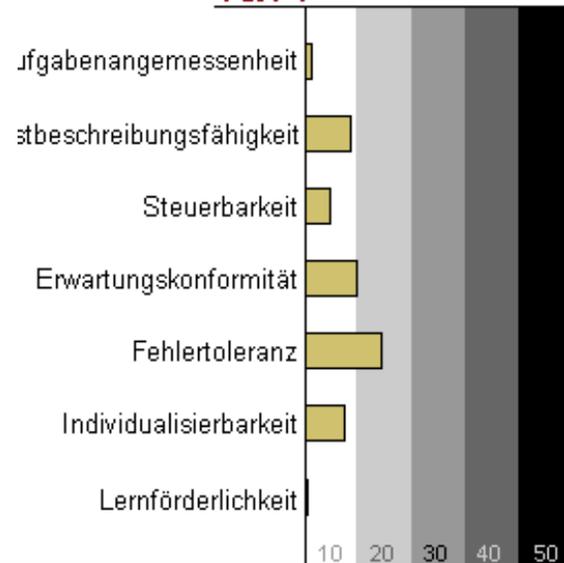
Version 0.11



Anzahl der von Benutzern im Schnitt berichteten Probleme (in Prozent)  
Software-Bewertung nach DIN EN ISO 9241-10

## Tavi

Version 0.2.1



Anzahl der von Benutzern im Schnitt berichteten Probleme (in Prozent)  
Subjektive Software-Bewertung nach DIN EN ISO 9241-10

# Diskussion und Ausblick I

- ! Handlungsempfehlungen möglich - auch ohne aufwändige „Reinigung“ der Daten (bei einer summativen Evaluation im within-Design mit  $N=12$ )
- ! Fehlende Normkonformität kann natürlich nur in Erhärtungsprüfungen festgestellt werden. Ob eine solche notwendig wäre, wird jedoch angezeigt
- ! Für Geschlecht- und Alter wurden keine experimentellen Effekte registriert
- ! Expertisegrad spielt hingegen eine Rolle (z.B. Haupteffekte bei Bearbeitungszeiten)

# Diskussion und Ausblick II

- ! Problem der Referenzaufgaben („Usability in Kontext“)
- ! Problem des Verzichts auf Rückmeldeschleifen für das Instruktionsverständnis muss gezielt bearbeitet werden
- ! Hohe Verfahrensökonomie ist unbedingt erstrebenswert, um mit den schnellen Entwicklungszyklen von Open-Source-Software mithalten zu können
- ! Grundproblem der Gratifikation für die Teilnahme an einem „Remote Usability Testing“ ist ungelöst
- ! Chance hier: sehr aufschlussreiche Mängellisten



**Vielen Dank  
für Ihre  
Aufmerksamkeit !**

BTW:  
Kontaktaufnahme erwünscht;)