



# Benutzer-Handbücher

Gregor Snelting, Andreas Zeller und Holger Cleve

Lehrstuhl Softwaretechnik  
Universität des Saarlandes, Saarbrücken





# ***Gestaltung von Benutzer-Handbüchern***

Das *Benutzer-Handbuch* ist in der Regel der erste Kontakt eines Benutzers mit einem neuen Software-System.

Das Handbuch ist sozusagen die Visitenkarte des Systems; entsprechend sorgfältig sollte es gestaltet sein.

Das Benutzer-Handbuch soll die Handhabung und das Verhalten des Produkts möglichst vollständig und fehlerfrei beschreiben.

Es muß möglich sein, das Produkt nur anhand des Handbuchs zu bedienen.





# ***Benutzer-Kategorien***

---

Anordnung und Aufbau eines Handbuchs werden im wesentlichen durch die *Adressaten* bestimmt. Man unterscheidet drei Kategorien:

- *Anfänger* haben keine oder wenig Erfahrungen mit Computersystemen im allgemeinen und dem Produkt im speziellen.
- *Experten* haben viel Erfahrungen mit Computersystemen im allgemeinen, aber wenig Erfahrungen mit dem speziellen Produkt.
- *Fortgeschrittene* liegen irgendwo zwischen Anfängern und Experten; sie haben vielleicht schon ähnliche Produkte benutzt.





# Handbuchtypen

---

Für jede Benutzer-Kategorie gibt es eigene Handbuch-Typen:

- Trainings-Handbuch (*Tutorial*)
- Referenz-Handbuch
- Referenzkarte
- Benutzer-Leitfaden (*user guide*)





# *Trainings-Handbuch* (Tutorial) \_\_\_\_\_

Zielgruppe: *Anfänger*

Ein Trainings-Handbuch

- ist als Kurs oder Trainingsprogramm organisiert
- ist nach *Aufgaben* gegliedert:
  - Es beschreibt *Wege zur Erreichung von Arbeitszielen*.  
Beispiel: *So buchen Sie eine Anmeldung*
  - Die nötigen Arbeitsabläufe bestimmen die Gliederung.
- muß von Anfang bis Ende vollständig durchgearbeitet werden
- erfordert direkte Arbeit mit dem Produkt, d.h. der Leser führt die Anweisungen des Trainings-Handbuchs aus
- vermittelt rasch Erfolgserlebnisse
- ist für fortgeschrittene Benutzer und Experten zu elementar





# Referenz-Handbuch

---

Zielgruppe: *Experten*

Ein Referenz-Handbuch

- ermöglicht schnellen Zugriff auf spezifische Information
- ist nach *Produktaufbau* gegliedert:
  - Alle Funktionen und Bestandteile werden in der Reihenfolge beschrieben, die sich aus der Struktur des Produktes ergibt.  
Beispiel: *Die Funktion „Anmeldungen bearbeiten“*
  - Es wird beschrieben, was das Produkt kann. . .
  - . . . und nicht, wie man Arbeitsziele mit dem Produkt erreicht
- bietet vollständige Informationen zu allen Produktfunktionen
- unterstellt, daß der Benutzer mehr weiß, als er tun will





# Referenzkarte

---

Zielgruppe: Ebenfalls *Experten*

Eine Referenzkarte

- faßt die wesentlichen Informationen der wichtigsten oder häufigsten Funktionen zusammen.
- ist möglichst kompakt (nicht mehr als eine A4-Seite)

Abwandlung: die *Referenztasche*.





# ***Benutzer-Leitfaden (user guide)***

---

Zielgruppe: *fortgeschrittene Benutzer*

Ein Leitfaden

- ist eine Mischform aus Trainings- und Referenzhandbuch
- muß simultan die Bedürfnisse beider Benutzergruppen, Anfänger und Experten, erfüllen.
- besteht typischerweise aus zwei Teilen:
  - Die aufgabenorientierte *Arbeitsanleitung* dient zum Einarbeiten.
    - \* nicht vollständig
    - \* erlaubt das Arbeiten mit Grundfunktionen.
  - Der *Referenzteil* dient zum späteren Nachschlagen.
    - \* erlaubt es, sich die weiteren Funktionen zu erschließen.





# *Benutzer-Leitfaden (II)*

---

## Ein Leitfaden

- erlaubt es, bereits bekannte Informationen zu überschlagen
- darf aber nicht davon ausgehen, daß der Leser Systemdetails bereits kennt
- ist bei einfachen Systemen das einzige Handbuch

Bei umfangreichen Systemen gibt es oft alle vier Handbuch-Typen!





# ***Aufbau eines Benutzer-Handbuchs***

---

Für Benutzer-Handbücher läßt sich kein generelles Gliederungsschema wie etwa Pflichtenhefte vorgeben. Neben dem sachlichen Inhalt gehören aber bestimmte Teile in jedes Handbuch:

## **Vorwort**

Enthält Adressatenkreis, Anwendungsbereich, Änderungen gegenüber Vorversionen.

Ziel: der Leser muß nach den ersten Seiten wissen, ob Produkt und Handbuch für ihn und seine Anwendung bestimmt sind.

Manche Leser überblättern das Vorwort; deshalb gehören unverzichtbare Informationen in die *Einleitung*.

## **Inhaltsverzeichnis**





# Handbuch-Aufbau (II)

---

## Einführung

Dient in der Regel als *Gebrauchsanleitung*, die über den Aufbau und den Umgang mit dem Handbuch informiert.

Beschreibt oft auch

- die Konfiguration, die für das Produkt erforderlich ist (wenn nicht in eigenem Kapitel).
- Zielgruppe
- Vorkenntnisse zum Verstehen des Handbuchs

Insgesamt soll die Einführung kurz sein.

## Installation

Beschreibt, was der Benutzer tun muß, um das Produkt in Betrieb zu nehmen.

Wird nur selten gelesen, kann deshalb in den Anhang (dann muß aber in der Einführung darauf verwiesen werden).





# *Handbuch-Aufbau (III)*

---

## **Benutzungsoberfläche**

Verdeutlicht den prinzipiellen Aufbau der Benutzungsoberfläche.

Insbesondere

- Tasten- und Mausbelegung
- grundsätzlicher Bildschirmaufbau (z.B. Fenstertypen)
- Dialogstrategie (z.B. erst Objekt, dann Funktion auswählen)

## **Produktstruktur**

Gibt einen Überblick über die einzelnen Bestandteile des Produkts.





# Handbuch-Aufbau (IV)

---

## Trainingsteil

(im Benutzer-Leitfaden oder Trainings-Handbuch)

Beschreibt *typische Arbeitsabläufe* mit Beispielen und Übungen, mit besonderem Wert auf *Routinearbeiten*.

Anordnung:

- häufigste Arbeitsabläufe und Routinearbeiten
- seltene Arbeitsabläufe
- Initialisieren und Löschen von Systemen

*Beispiele* müssen aus dem geplanten Anwendungsbereich stammen

*Abstraktionen* (also *var-1*, *var-2* statt *zins* und *kapital*) sind zu vermeiden

*Übungen* sind so zu gestalten, daß der Benutzer mit dem Produkt arbeitet – und dabei Erfolgserlebnisse hat.





# ***Handbuch-Aufbau (V)***

---

## **Referenzteil**

(im Benutzer-Leitfaden oder Referenz-Handbuch)

Enthält eine vollständige Beschreibung der einzelnen Objekte und Funktionen.

Die Anordnung orientiert sich meistens an der Menüstruktur.

*Kommandos* werden alphabetisch angeordnet.

## **Behandlung von Problemen**

Enthält eine Liste von Fehlermeldungen einschließlich detaillierter Erklärungen und Vorschläge

## **Literaturverzeichnis**

## **Abkürzungsverzeichnis**

## **Glossar**





# ***Handbuch-Aufbau (VI)***

---

## **Stichwortverzeichnis / Index / Register**

Wichtigstes Hilfsmittel für einen schnellen Zugriff

Enthält Benutzerziele (z.B. *Anmeldung vornehmen*) und Funktionsnamen (z.B. *Buchung erfassen*)

*Erstellt der Handbuch-Autor ein gutes Handbuch, dann sparen hunderte oder tausende Benutzer eine Menge Zeit und Nerven!*





# Hilfesysteme

---

Ein *Hilfesystem* unterstützt den Benutzer während der Benutzung durch explizite Erklärungen und Auskünfte.

Schnelle und vom Kontext abhängige Ergänzung zum Benutzer-Handbuch

Hilfesysteme erläutern typischerweise folgende Punkte:

- Aktuell wählbare Objekte, Funktionen, und Kommandos (und deren Optionen)
- Bedeutung von Funktionstasten und Mausknöpfen
- Hinweise zu Eingaben
- Erläuterungen zu Funktionsergebnissen
- Spezifische Fehlererklärungen (einschließlich Hilfen zur Korrektur)





## *Vorteile gegenüber dem Handbuch:*

- + Texte in Hilfesystemen sind schneller und leichter zu aktualisieren.
- + Die Information in Hilfesystemen kann nicht verlorengehen.

## *Nachteile gegenüber dem Handbuch:*

- Der Text auf dem Bildschirm wird *langsamer gelesen* als in einem Handbuch (kürzer fassen!)
- Notizen und Markierungen wie auf Papier sind nicht möglich.

Es gibt auch *Mischformen* aus Hilfesystemen und Handbüchern:

- Es gibt Programme, die *ausschließlich per Hilfesystem* dokumentiert sind; lediglich die Installationsanleitung kommt in gedruckter Form.
- Bei manchen Programmen (z.B. *Netscape*) ist die gesamte Dokumentation als WWW-Hypertext organisiert, auf den über das Netz zugegriffen wird.





# *Dynamische Hilfesysteme*

---

Ein *statisches Hilfesystem* liefert stets dieselbe Information, unabhängig vom Kontext, z.B.:

in einem Fenster wird immer die gleiche Erklärung gegeben.

der UNIX-Editor ed gibt bei Fehlern aller Art die Meldung “?” aus.

Ein *dynamisches Hilfesystem* berücksichtigt den Kontext zum Zeitpunkt der Hilfeanforderung, z.B.:

in jedem Eingabefeld eines Fensters wird eine spezifische Hilfe gegeben.

eine Fehlermeldung lautet *Der eingegebene Wert 0.005 ist zu klein*

Dynamische Hilfesysteme sind stets vorzuziehen.





# *Individuelle Hilfesysteme*

---

Ein *uniformes Hilfesystem* gibt jedem Benutzer dieselbe Hilfe unabhängig von seinem Kenntnisstand.

Ein *individuelles Hilfesystem* unterscheidet nach verschiedenen Benutzergruppen (Anfänger, Experte).

Die Einstufung kann auch automatisch erfolgen.

Beispiel: Nachdem Sachbearbeiter Müller nur noch selten das Hilfesystem aufruft, wird er als vom Hilfesystem als *Experte* eingestuft und erhält nur noch eine Kurzform der Erklärungstexte.

Individuelle Hilfesysteme sind stets vorzuziehen.





# *Passive und Aktive Hilfesysteme*

---

Nur 40% der Funktionalität komplexer Systeme werden benutzt.

Manche Funktionen werden nur selten benutzt, da der Benutzer

- die Funktionen nicht im Detail kennt
- sich unsicher über ihre Details und Wirkungen ist

Das ist der Einsatzbereich von *passiven Hilfesystemen*.

Ein passives Hilfesystem erwartet, daß der Benutzer von sich aus eine Hilfeleistung anfordert, z.B.

- durch Drücken einer Taste (*F1* oder *Help*)
- durch Eingabe eines Kommandonamens ("*man ls*")
- durch Anfragen in natürlicher Sprache („*Warum kann hier kein Text eingefügt werden?*“)
- durch Anwahl eines Bedienelements (*balloon help*)





## *Passive und Aktive Hilfesysteme (II)*

---

Problem: Oft sind Konzepte realisiert, die der Benutzer nicht vermutet.

Ein *aktives Hilfesystem* beobachtet das Benutzungsverhalten und wird von sich aus aktiv, um dem Benutzer eine Hilfe zu geben.

Beispiel: Um sich von einem Wort zum anderen zu bewegen, wurden einzeln Pfeiltasten benutzt. Der Word-Assistent erkennt diesen Vorgang und weist den Benutzer darauf hin, daß mit Ctrl+Pfeiltasten Wörter übersprungen werden können.

In der Regel werden heute passive und uniforme Hilfssysteme eingesetzt, die statische und dynamische Hilfeleistungen mischen.





# ***Checkliste: Benutzer-Handbuch***

---

## **Kann man leicht durch das Handbuch navigieren?**

Sind im Benutzer-Leitfaden und Trainings-Handbuch Kapitel und Abschnitte nach Benutzerzielen organisiert und benannt?

Ist das Inhaltsverzeichnis gut gegliedert?

Enthält das Stichwortverzeichnis Benutzerziele und Funktionsnamen?

## **Unterstützt das Handbuch leichtes Erlernen?**

Richtet sich die Sprache an den Endanwender?

Wird die Information in kleinen Einheiten dargeboten?

Wird die Information in einer logischen Sequenz dargeboten?

Gibt es Beispiele? Zeichnungen? Grafiken?

Werden visuelle Kennzeichen (Typographie) konsistent verwendet?

Wird eine „Vermenschlichung“ des Computers vermieden?





## **Ist das Handbuch gut lesbar?**

Lesbarkeit wird insbesondere erreicht

- durch großzügigen Gebrauch von Leerraum
- durch Vermeiden von unnötigem Jargon

## **Sind die Pflichtteile enthalten?**

### **Gibt es geeignete Szenarien?**

Benutzer-Leitfaden und Trainings-Handbuch müssen typische Szenarien im Umgang mit dem System enthalten, und beschreiben, wie sich das System verhält.

Diese Szenarien bilden die Grundlage für spätere Tests (und sollten deshalb einzeln gekennzeichnet sein).

### **Ist eine Hilfefunktion vorhanden?**

Das System sollte zumindest eine statische, passive, uniforme Hilfe anbieten.

