

# Software Anleitungen



**verständlich schreiben**

Konzepte, Strukturen,  
Beispiele

**Dietrich Juhl**

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation  
in der Deutschen Nationalbibliografie; detaillierte bibliografische  
Daten sind im Internet über <http://dnb.dnb.de> abrufbar.

© 2018 Dietrich Juhl

Herstellung und Verlag: BoD – Books on Demand, Norderstedt

ISBN 978-3752815528

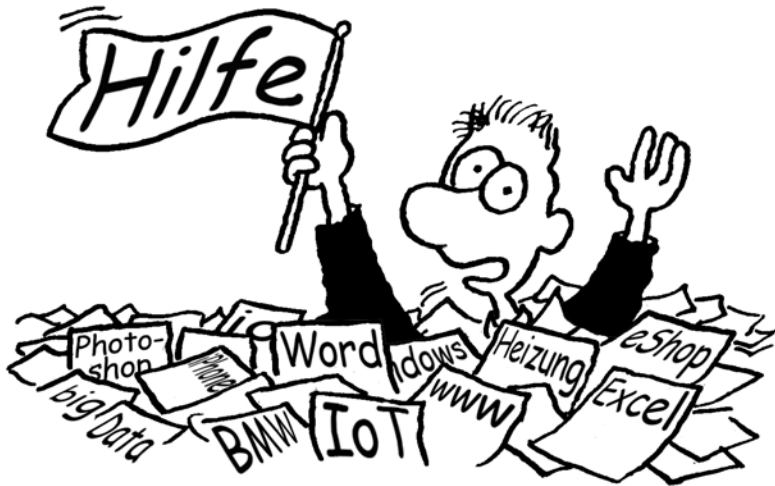
# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort.....</b>	<b>9</b>
1.1	Konzept: Software-Anleitungen zielorientiert schreiben	10
1.2	Zielgruppe des Buchs	14
1.3	Anmerkung zu den Beispielen	15
1.4	Errata und aktuelle Ergänzungen	16
<b>2</b>	<b>Vorüberlegungen .....</b>	<b>17</b>
2.1	Wozu soll die Anleitung dienen?	18
2.2	Use Cases	19
2.3	Was-macht-wer-Matrix	25
2.4	Zielgruppe	26
2.5	Anwendungshäufigkeit	30
2.6	Grundbedienung / optionale Features	32
2.7	Didaktik	33
2.8	Gliederungen	34
<b>3</b>	<b>Grobgliederung einer Software-Anleitung.....</b>	<b>37</b>
3.1	Leistungsbeschreibung	39
3.2	Fachliche Grundlagen	42
3.3	Bedienoberfläche	45
3.4	Einführung in die Bedienung	49
3.5	Tätigkeiten	55
3.6	Funktionen	56
3.7	Ganze Anleitungen differenzierter sehen	58

<b>4 Handlungen.....</b>	<b>59</b>
4.1 Einleitung: Anleiten zum Handeln	60
4.2 Schritt für Schritt	63
4.3 Ausprobieren lassen	66
4.4 Die „erweiterte“ Handlungsanweisung bei SW	69
4.5 Elemente der erweiterten Handlungsanweisung	75
4.6 Rückblick: „erweiterte Handlungsanweisung“	123
4.7 Umfangreiche Handlungen didaktisch aufteilen	124
4.8 Komplexe Handlungen fallweise einzeln anleiten	126
4.9 Einzelhandlungen und Prozesse	127
4.10 Anordnen der Handlungsanleitungen	129
4.11 Exkurs: Screenshots	134
4.12 Exkurs: Suchworte	141
<b>5 Funktionsbeschreibungen.....</b>	<b>143</b>
5.1 Vorüberlegungen	145
5.2 Der Blick auf Funktionen	147
5.3 Die Funktionsbeschreibung	152
5.4 Die Elemente der Funktionsbeschreibung	154
5.5 Ganze Funktionsbeschreibungen	178
5.6 Nur Zweck und Handlungsmöglichkeiten	181
5.7 Anordnen von Funktionsbeschreibungen	182
<b>6 Ganze Anleitungen.....</b>	<b>185</b>
6.1 Normale Software-Anleitung	187
6.2 Nur Tätigkeiten	189
6.3 Nur Funktionen	189

6.4	Anleitung für kleine Software	190
6.5	Anleitung mit funktionsweiser Anordnung	191
6.6	Wissen didaktisch aufbauen	194
6.7	Kontextsensitive Hilfe	196
6.8	Tools	199
6.9	Apps	200
6.10	Displaygeführte Software	202
6.11	Software für Anwender ohne Fachkenntnisse	209
6.12	Software mit individuellen Benutzereinstellungen	211
6.13	Nur Nachschlagen(?)	212
6.14	Nur Topics(?)	213
6.15	Aufteilung der Anleitung	214
6.16	Unterteilung der Anleitung	215
6.17	Sichtbarkeit der Unterteilung/Aufteilung	216
6.18	Anleitung zur Anleitung	217
<b>Anhang .....</b>		<b>219</b>
<b>7</b>	<b>Andere Ansätze.....</b>	<b>221</b>
7.1	FAQ	222
7.2	Tipps	223
7.3	Tutorial	224
7.4	Videos, Screencasts	226
7.5	Weiterlernen unterstützen	228
7.6	Strategie	230
7.7	Kurzanleitung	232
7.8	Quick Guide / Schnelleinstieg	233

<b>8</b>	<b>Papier, elektronisch, KI ...</b>	<b>235</b>
8.1	Technisch-didaktische Möglichkeiten im Topic	239
<b>9</b>	<b>Bezug zur Software</b>	<b>241</b>
9.1	Nebeneinander von Software und Anleitung	243
9.2	Kontextsensitive Anbindung	246
9.3	Hilfetexte in der Bedienoberfläche	248
<b>10</b>	<b>Topics</b>	<b>255</b>
10.1	EPPO (Every Page is Page One)	257
10.2	Top-down / Bottom-up	258
10.3	Topics nutzbar machen	260
<b>11</b>	<b>Literatur, Bilder</b>	<b>263</b>
	<b>Stichwortverzeichnis</b>	<b>266</b>



# 1 Vorwort

*Alles ist Software*

*Software ist anders!*

*Software braucht andere Anleitungen*

**Auszug aus dem Original**

## 1.1 Konzept: Software-Anleitungen zielorientiert schreiben

Mit diesem Buch lege ich ein Konzept für Technische Redakteure vor, um Anleitungen für Software zu strukturieren und verständlich zu schreiben. Dabei benutze ich vor allem das zielorientierte Schreiben und didaktische Überlegungen, wie Anwender durch schriftliche Unterlagen befähigt werden können.

### Aus der Praxis



Das Konzept habe ich aus meiner Tätigkeit als Technischer Redakteur und Trainer in vielen Schritten entwickelt:

- Eigene Schreibpraxis
  - Optimieren des eignen Stils bei vielfältigen Beschreibungsaufgaben
  - Didaktische Überlegungen dabei
  - Überlegungen zur sinnvollen Gliederung der Inhalte
- Anregungen aus anderen Anleitungen
- Operationalisieren von sinnvollen Schreibstilen, Gliederungsprinzipien (Herausfinden von Gemeinsamkeiten, Benennung, Einordnung)
- Systematisieren der Theorien
- Optimieren
- Beschreiben der Theorien für dieses Buch
- Beispiele entwickeln für dieses Buch
- Abrunden zu einem Gesamtkonzept

Das Konzept beruht in hohem Maße auf meiner eigenen Schreibpraxis und dem Bemühen, eine gute Schreibpraxis für andere vermittelbar zu machen.



## **Zielorientiert schreiben**

Das zielorientierte Schreiben habe ich in meinem ersten Buch (Juhl, 2015; Erstauflage 2002) entwickelt und eingeführt. Es beruht auf der Idee, dass jeder Text einer Anleitung ein bestimmtes Ziel verfolgt (z.B. zur Handlung anleiten). Wenn man als Autor dieses Ziel kennt und „anzieht“, kann man Strukturen und Texte darauf ausrichten und optimieren.

Dabei ergab sich z.B. die Juhl'sche Handlungsanweisung, die neben der Schritt-für-Schritt-Folge vor allem das Handlungsziel, das Handlungsergebnis und den Ausblick beinhaltet.

Dieses Prinzip hat sich in der Praxis sehr bewährt, weil es eine gute Grundlage für das Schreiben bietet und weil Strukturen und Formulierungshilfen im Laufe der Zeit optimiert und an neuen Bedarf oder auch den Zeitgeist angepasst werden können.

Diese Idee führe ich hier fort.

## **Software ist überall**

Unser Leben wird immer mehr von Technik und vor allem Software durchdrungen. Kaum noch ein Arbeitsplatz kommt ohne PC aus, Internet und Smartphone ermöglichen uns, auf Informationen zuzugreifen, sie zu bearbeiten und Prozesse oder Maschinen zu steuern. Viele Aufgaben, die früher von Fachkräften oder als Dienstleistung ausgeführt wurden, können wir, dank Software, heute leicht selbst erledigen.

## **Anwender sind geübter geworden**

Durch den täglichen Umgang mit Software sowie gute und intuitive Bedienoberflächen gehen Anwender sehr viel unbefangener und mutiger an neue Software als noch vor zehn Jahren. Das sind manchmal nur kleine Anwendungen, die gar nicht als Software wahrgenommen werden, wie das Einkaufen im Internet oder das Suchen in Google Maps.

Manchmal ist es eine neue App oder ein Tool und seltener eine umfangreiche Software, wie Word oder Excel.

Immer mehr steigen Anwender ohne Anleitung in neue Anwendungen ein, probieren aus, lassen sich von Freunden oder Kollegen etwas zeigen und kommen in vielen Fällen ohne Anleitung zurecht. Das ist gut so!

## **Brauchen wir noch Anleitungen?**

Den meisten Anwendern ist es lieber, neue Anwendungen sich selbst zu erschließen, ohne vorher lesen zu müssen („Ich lese nicht, ich probiere aus ...“). Das geht auch in vielen Fällen gut: einfache Prozesse, wie das Einkaufen, können so intuitiv sein, dass keine Anleitung notwendig ist. Auch komplexere Anwendungen, wie Word, lassen sich in vielerlei

Beziehung intuitiv bedienen. (Dass das nicht immer ausreicht, wissen Sie vermutlich, darauf gehe ich gleich noch ein.)

Ganz sicher gibt es aber komplexe Anwendungen, bei denen es nicht ohne Anleitung geht. Die Hürden für den Anwender sind dabei sehr unterschiedlich:

- Manchmal braucht der Anwender nur einen Einstieg in die Software-Idee (z.B. bei einer Software, die doppelte Dateien erkennt und Dubletten löscht).
- Manchmal ist ihm die Idee sofort klar, aber für komplexere Funktionen braucht er Hilfe (z.B. Photoshop ist in der Grundanwendung leicht bedienbar, für fortgeschrittene Funktionen, wie Ebenen oder Pfade, sind Anleitungen notwendig).
- Bei vielen Software-Produkten wäre es sinnvoll, etwas zu lernen, um die besonderen Angebote zu nutzen (z.B. Formatvorlagen in Word).
- Viele Anwendungen sind nach wie vor sehr erklärungsbedürftig, weil eine komplexe Materie behandelt wird oder die Bedienoberfläche noch nicht so intuitiv ist, dass alle Anwender sofort damit klar kommen (z.B. Messprogramme, Buchhaltungssoftware, Maschinensteuerung).

### **Software-Anleitungen sind anders**

Aber warum sollen Software-Anleitungen anders sein als solche für Geräte oder Maschinen? Gelten bei Software nicht die gleichen Prinzipien der Technischen Dokumentation? Richtig anleiten, Schritt für Schritt ...

Vor allem durch den intuitiven Zugang und das direkte Feedback beim Bedienen (zumindest häufig), kann Software in weiten Teilen ohne oder mit wenig Anleitung genutzt werden. Die Bedienung selbst scheint vielfach unproblematisch zu sein, so dass Schritt-für-Schritt-Anleitungen häufig überflüssig sind.

Bei Software sind die Hürden für den Anwender häufig ganz anders als das eigentliche Tun:

- Bei vielen Features muss vor allem das Angebot erklärt werden:
  - Ein Anwender muss die Möglichkeiten der bedingten Formatierung in Excel verstehen, bevor er sie anwenden kann.
  - Druckformate in Word lassen sich nur benutzen, wenn man das Prinzip verstanden hat.
  - Überblendungen in PowerPoint muss man kennen, um sie zu nutzen.
- Bei anderen Software-Anwendungen muss vor allem die Idee der Software vermittelt werden:
  - Bei Excel wäre das die Idee, Zahlen in einer Tabelle darzustellen und Zellen mitei-

inander zu verknüpfen.

- Bei einer Videoschnittsoftware wäre es die Idee, Videos in Kanäle zu legen und auf dem Zeitstrahl festzulegen, welcher Kanal gerade gezeigt wird.

## **Wenig Schritt für Schritt**

Vor allem brauchen wir in Software-Anleitungen meistens wenige Schritt-für-Schritt-Anleitungen, weil das Tun durch intuitive Bedienoberflächen und Standardbedienelemente klar ist! (Verstehen Sie mich nicht falsch: Die Schritt-für-Schritt-Anleitung bleibt auch in Software-Anleitungen wichtig! Es gibt aber viele Alternativen, die sinnvoll eingesetzt werden können.)

## **Neue Beschreibungsansätze, mehr Systematik**

Im Buch werden neue Beschreibungsansätze entwickelt, um Software-Ideen, Software-Konzepte und Handlungsmöglichkeiten besser zu vermitteln. Sicher schreiben Sie vielfach schon so oder ähnlich und machen vieles intuitiv bereits richtig.

Mein Anspruch ist es aber, diese Methoden zu benennen und zu systematisieren. Das gibt Ihnen mehr Möglichkeiten, Methoden bewusst einzusetzen und zu optimieren.

## **Mehr Flexibilität und Mut für neue Anleitungen**

Insgesamt plädiere ich für flexiblere Strukturen, weil der Informationsbedarf bei unterschiedlicher Software und unterschiedlichen Handlungen sehr unterschiedlich ist. Und ich plädiere für neue systematische Beschreibungsansätze (wie z.B. das Beschreiben der Handlungsmöglichkeiten), die das Schreiben für uns leichter und vor allem das Lesen angenehmer machen.

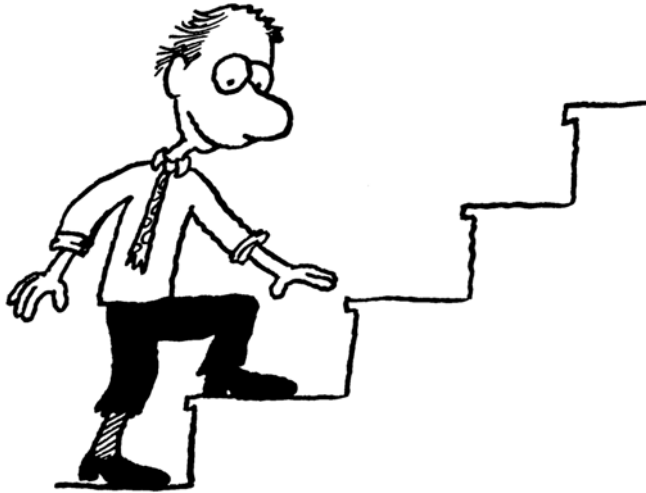


## 1.2 Zielgruppe des Buchs



Dieses Buch wendet sich an alle, die verständliche Anleitungen für Software schreiben wollen.

- Am meisten profitieren Sie, wenn Sie schon Anleitungen geschrieben haben, weil Sie dann viele angesprochene Probleme bereits aus der Praxis kennen und neue Vorschläge leicht in Ihre Praxis „eindenken“ können.
- Auch als Student oder Anfänger finden Sie hier systematische Überlegungen und Konzepte, die Ihnen konkret helfen, Ihren Schreibstil zu entwickeln.
- Ich wende mich ausdrücklich auch an „gestandene“ Technische Redakteure, die schon seit Jahren schreiben und hoffe, dass Sie noch manche Anregung finden und viele Methoden systematisieren können.
- Vielleicht geht es Ihnen auch wie vielen meiner Workshop-Teilnehmer, die sich bestätigt und ermutigt fühlen:
  - mehr Mut für flexible Strukturen und damit mehr Mut zu Inkonsistenz (im Sinne angepasster Beschreibungen) zuzulassen,
  - mehr Mut zum Weglassen aufzubringen,
  - Mut zu haben, auf Schritt für Schritt zu verzichten
  - und Funktionsbeschreibungen als Beschreibungsmethode einzusetzen.



## 2 Vorüberlegungen

*Wozu soll die Anleitung dienen?*

*Wer ist der Anwender?*

*Wann nimmt der Anwender die Anleitung zur Hand?*

*Wie sollen wir gliedern?*

Die hier dargestellten Gedanken sind wichtige Vorüberlegungen, die man als Technischer Redakteur anstellt, um die neue Anleitung auf die Anwender in ihren Situationen auszurichten.

### 2.1 Wozu soll die Anleitung dienen?

Die Frage klingt trivial und ist doch eine Betrachtung wert: Wozu soll die Anleitung dienen?

Antworten wie „Zu jeder Software muss doch eine Anleitung geliefert werden“ oder „Das ist vorgeschrieben“ bringen uns nicht weiter, weil wir daraus keine Kriterien ableiten können.

Die Anleitung muss mehrere Zwecke erfüllen. Der Anwender

- soll installieren können,
- soll sich selber einarbeiten können,
- soll das Prinzip verstehen (Prinzip = Funktionsweise?),
- soll Grundfunktionen ausführen können,
- soll seine Aufgaben mit der SW erledigen können,
- soll bei Bedarf nachschlagen können, sein Problem finden, identifizieren und lösen,
- soll sein SW- und Handlungswissen erweitern können.

Aber auch:

- Wir haben unsere Informationspflicht erfüllt.
- Der Anwender kommt alleine klar
- und belästigt unsere Hotline nicht unnötig.
- Der Anwender schätzt die Software!

## 2.2 Use Cases



Anleitungen sollen dem Anwender helfen, seine Aufgaben oder Probleme mit dem Produkt zu lösen. Dafür ist es sinnvoll, die Use Cases der Anleitung zu betrachten, also die Fälle, in denen ein Anwender die Anleitung zur Hand nimmt (oder sinnvollerweise zur Hand nehmen sollte).

Standardmäßig können wir folgende Use Cases unterscheiden, die man ggf. noch weiter unterteilen kann:

Use Cases
Auswahl
Installation
Konfiguration
Erste Einarbeitung
Normalnutzung
Spezielle Fragen lösen
Weiterlernen

Diese Standard Use Cases möchte ich im Folgenden näher betrachten und die Konsequenzen für den Technischen Redakteur darstellen.

## 2.8 Gliederungen



Normalerweise gliedern wir unsere Inhalte so, dass ein Anwender sich systematisch einarbeiten kann und für alle Use Cases eine Ordnung vorfindet, in der er sein Anliegen einsortieren kann und dadurch schnell eine Antwort findet.

### 2.8.1 Curriculare Gliederung

Die meisten Anleitungen müssen irgendwie geeignet sein, sich systematisch (Top Down) einzuarbeiten. Hierfür können wir die Inhalte lernlogisch sortieren und den Anwender von Kapitel zu Kapitel führen.

Eine Anleitung, die darauf spezialisiert ist, nennen wir „Lernanleitung“.

Mögliche Ordnungen sind:

- vom Normalen zum Speziellen
- vom Häufigen zum Seltenen
- vom Einfachen zum Komplizierten

### 2.8.2 Nachschlage-Gliederung

Auf der anderen Seite gibt es reine „Nachschlageanleitungen“, deren Inhalte nur einzeln angesehen werden, wie ein Lexikon, das man ja nicht von vorne nach hinten liest, z.B. Befehle einer Programmiersprache oder alle Funktionen einer Software.

Zur Ordnung bieten sich folgende Ansätze an:

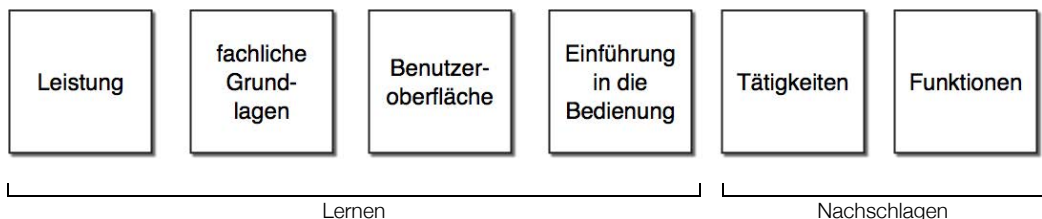
- funktionsweise,  
systematisch in der Hierarchie des Menüs  
Diese Ordnung kennt der Anwender, weil er sie stets vor Augen hat.
- alphabetisch  
Die alphabetische Ordnung ist die schnellste Möglichkeit, einen Fachbegriff zu finden, wenn er bekannt ist und der Anwender das ABC beherrscht.

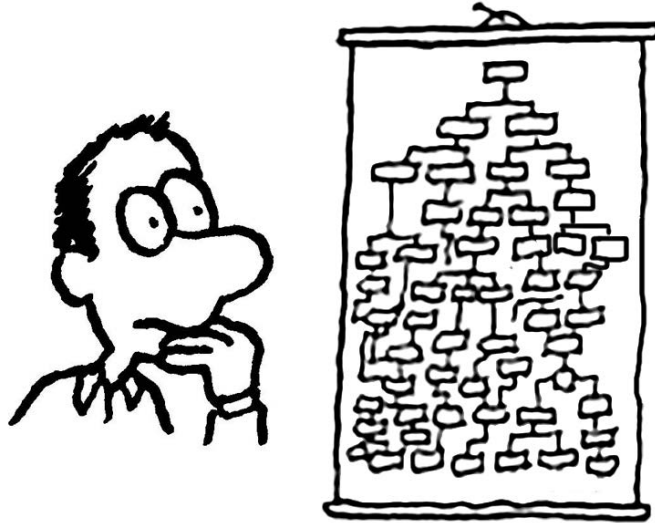


- sachlogisch  
Inhalte lassen sich immer irgendwie zu Oberbegriffen/Kategorien zusammenfassen und so hierarchisch ordnen. Solche Zuordnungen und vor allem die Reihenfolge in der Hierarchie sind aber nicht eindeutig, so dass eine gewählte Ordnung nicht unbedingt für alle Leser logisch ist (auch wenn es für den Autor super logisch erscheint!). (Vergleiche Baker, 2013)  
Zum Beispiel lassen sich PKWs nach Hersteller, Typ, Bauart, Hubraum, PS, Farben usw. ordnen. Die Reihenfolge in einer Hierarchie ist aber beliebig, z.B. Hersteller, Typ, Hubraum ... Jemandem, der nach einem Cabrio sucht, wäre vielleicht die Ordnung Bauart, PS, ... lieber.
- prozessorientiert  
Wenn eine Software einen oder mehrere Prozesse abbildet (z.B. Bearbeitung einer Schadensmeldung), kann diese Ordnung auch in der Nachschlageanleitung übernommen werden, z.B. als Hierarchie aus Prozessen und Prozessschritten.
- Facettenklassifikation  
Elektronisch ist es möglich, unterschiedliche Ordnungen anzubieten, wobei der Anwender die Auswahl und die Reihenfolge der Kategorien bestimmen kann, z.B. Bauart: Cabrios, 2 Türen, sortiert nach PS.  
Ein solches Angebot ist bei umfangreicher Dokumentation sinnvoll, um schnell nur relevante Topics anzuzeigen, z.B. bei einer Programmiersprache: alle Befehle, die einen String als Eingangsvariable und einen logischen Ausdruck als Ausgangsvariable haben.  
Ein solches System setzt voraus, dass die Topics mit Metadaten gekennzeichnet sind.

### 2.8.3 Kombination Lern-/Nachschlage-Anleitung

Anleitungen für Software sind häufig eine Kombination einer Lernanleitung zur systematischen Einarbeitung und einer Nachschlageanleitung, die je nachdem, Funktionen und/oder Tätigkeiten oder auch Parameter geordnet darstellt.





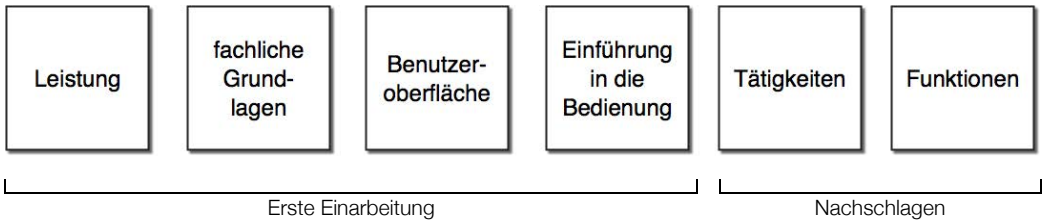
### 3 Grobgliederung einer Software-Anleitung

*lernlogisch – sachlogisch*

*als Einstieg*

*und zum Nachschlagen*

## Grobgliederung



Die im Bild gezeigte Grobgliederung eignet sich für die meisten Software-Anleitungen.

- Das Bild zeigt die inhaltstragenden Kapitel einer Software-Anleitung.
- Weitere Elemente können dazu kommen, z.B.:
  - Orientierungselemente wie Titelblatt, Impressum ...
  - andere Kapitel wie Versionshistorie, Produktbeschreibung, Fehlersuche ...

Meines Erachtens können wir uns aber zunächst auf die oben gezeigten Elemente beschränken und diese vertiefen.

### **Anmerkung: Hier herkömmliche Gliederungen**

Im Anhang habe ich die Ideen von Topics, EPPO und einem Content Delivery dargestellt. Diesen Überlegungen schließe ich mich durchaus an, vor allem:

- Topics als einzeln stehende unabhängige Einheiten zu sehen,
- Topics in einem CMS zu halten und mit Metadaten vielfach verwertbar zu machen.

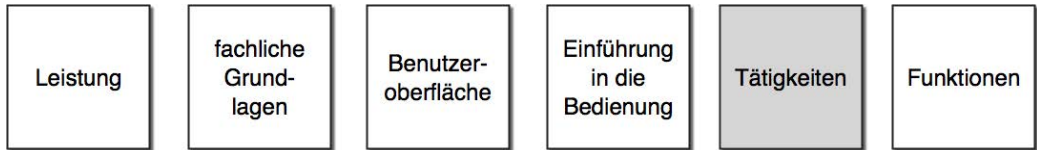
Die Überlegungen, nur noch Topics zu schreiben, diese vielfach zu verlinken und über unterschiedliche Suchen und Ordnungen zugreifbar zu machen, gehen mir im Moment noch zu weit, da immer ein geeignetes CMS vorhanden sein muss und für mich die Theorien noch nicht greifbar genug sind.

Deswegen arbeite ich hier im Buch mit herkömmlichen Gliederungen, so wie das oben dargestellt ist und wie ich das auch hergeleitet habe.

Ich denke, mit dieser Gliederung sind wir heute arbeitsfähig und trotzdem offen für neue Ordnungen.

EPPO: Siehe auch Kapitel 10 Topics, Seite 255.

## 3.5 Tätigkeiten



In diesem Bereich leiten Sie zu den wichtigsten Handlungen an.

- Jede Handlung in einem eigenen Kapitel.
- Jede Handlung als Schritt-für-Schritt-Anleitung mit Überschrift und Schritt für Schritt
- oder als erweiterte Handlungsanweisung (siehe weiter hinten, Seite 69).

### Welche Handlungen?

Meistens ist es nicht sinnvoll, hier zu allen Handlungen anzuleiten, weil es häufig sehr viele, oft auch selbsterklärende Handlungen und Handlungsmöglichkeiten gibt.

- Daher ist es sinnvoll nur die wesentlichen Handlungen hier einzeln zu beschreiben, so dass ein Anfänger hier alle Handlungen findet, die er am Anfang braucht.
- Wenn die Anzahl an möglichen Handlungen überschaubar ist, können Sie hier auch alle Handlungen in einzelnen Kapiteln abhandeln.
- Möglicherweise ist es auch sinnvoll, einzelne Handlungen nur anzureißen (z.B. Photoshops: Verfremden mit Filtern). Sie geben eine Handlungsanweisung zu der Funktion, erklären aber nicht alle Einstellmöglichkeiten, sondern verweisen auf die systematische Beschreibung im Bereich „Funktionen“.

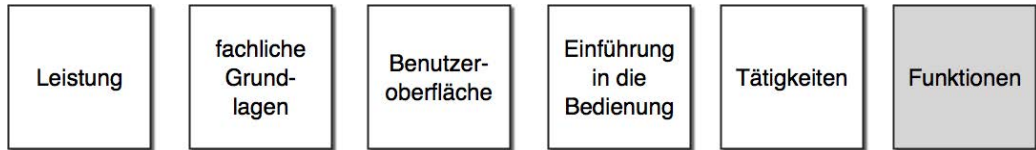
### Wie zur Handlung anleiten?

Es gibt mehrere Möglichkeiten zur Handlung anzuleiten. Die einfachste und bekannteste ist die Schritt-für-Schritt-Anleitung. Andere Möglichkeiten beschreibe ich später.



Kapitel 4 Handlungen, Seite 59

## 3.6 Funktionen



Neben der Anleitung zum Handeln, kann man als TR auch Funktionen der Software systematisch erklären.

- Wenn ein Anwender die Grundbedienung beherrscht, ist vielleicht diese Beschreibungsmöglichkeit besser und kürzer als die Anleitung zu einzelnen Handlungen.
- Als TR müssen wir uns aber immer fragen, ob der Anwender mit dieser Beschreibung handeln kann. (Wie Sie später sehen werden, gibt es hier gute Beschreibungstechniken für Funktionen, wenn die Bedienoberfläche intuitiv ist.)

### **Ist das ein Widerspruch zur herrschenden Lehrmeinung?**

Bisher haben wir in der Technischen Dokumentation die Meinung vertreten, dass nur die handlungsorientierte Beschreibung (z.B. Schritt für Schritt) richtig ist. Eine „geräteorientierte“ Beschreibung, wie sie auch die funktionsweise Beschreibung darstellt, war nicht gewünscht.

Ich sehe das (heute) etwas anders:

- Prinzipiell ist es richtig, dass die Anleitung zum Handeln im Vordergrund steht und Schritt-für-Schritt-Anleitungen sinnvoll sind.
- Ein erfahrener Anwender (z.B. im Lernfortschritt der Anleitung) kann möglicherweise auch handeln, wenn er die Handlungsmöglichkeiten und die Bedienelemente kennt.
- Eine solche Beschreibung ist möglicherweise kürzer und angenehmer zu lesen.
- Deswegen müssen wir als TR diese Beschreibungstechnik kennen und ggf. gezielt einsetzen.

### **Müssen alle Funktionen erklärt werden?**

Zunächst erscheint es sinnvoll, alle Funktionen im Sinne einer Dokumentation zu erklären. Der Blick auf ein praktisches Beispiel zeigt aber schnell, dass das meist nicht der Fall ist, z.B.:

- Klare Funktionen, wie „Speichern unter ...“, „Kopieren“ oder „Beenden“ brauchen für heutige Anwender nicht beschrieben werden. Ich plädiere dafür sie wegzulassen.

- Manche Funktionen können besser zusammen beschrieben werden, wie z.B. die Filter in Photoshop, so dass man auf Einzelheiten nicht eingehen muss.
- Manche Funktionen lassen sich gut ausprobieren und so erfahrbar machen, so dass wir als TR auch hier wenig beschreiben müssen.
- Insgesamt ist die Anzahl an Funktionen in vielen Fällen so groß, dass wir zumindest überlegen müssen, ob die systematische Beschreibung aller Funktionen sinnvoll ist.
- Trotzdem gibt es ganz sicher Funktionen, die stark erklärungsbedürftig sind.

Daraus folgt, dass wir nur einen Teil der Funktionen beschreiben und auch in der Detaillierung variieren müssen.

### Wie können Funktionen richtig erklärt werden?

Es gibt mehrere Möglichkeiten, Funktionen systematisch zu erklären. Hier haben sich vor allem der „Zweck der Funktion“ und die „Handlungsmöglichkeiten“ bewährt.

Alle Möglichkeiten, eine Funktion systematisch zu erklären, beschreibe ich später:



Kapitel 5.3 Die Funktionsbeschreibung, Seite 152



## 4 Handlungen

*Zum Handeln anleiten*

*Hauptzweck einer Anleitung*

*Welche Information braucht der Anwender?*

## 4.4 Die „erweiterte“ Handlungsanweisung bei SW



Die hier vorgestellte „erweiterte“ Handlungsanweisung enthält unterschiedliche Beschreibungsansätze in einer sinnvollen Reihenfolge.



Je nach Software und zu beschreibender Handlung können Elemente eingesetzt oder weggelassen werden, was ich mit dem Equalizer zum Ausdruck bringen möchte. Als Technischer Redakteur müssen Sie die einzelnen Elemente beherrschen, um sie für einzelne Tätigkeiten sinnvoll einzusetzen.



## Unterschiedliche Bedarfe

Die unterschiedlichen Bedarfe ergeben sich vor allem aus der Software:

- Bei einer Standardhandlung (z.B. „Datei speichern unter...“) muss das Ziel nicht ausführlich erklärt werden.
- Bei einem besonderen Feature (z.B. „bedingte Formatierung“ in Excel) ist das Ziel besonders wichtig.
- Bei einer aufgeräumten Bedienoberfläche mit Standardbedienelementen braucht man weder „Step by Step“ noch eine „Erklärung der Parameter“.

Aber auch die anvisierte Zielgruppe muss berücksichtigt werden:

- Ein Fachmann fühlt sich mit Regeln wohl.
- Ein Anfänger bevorzugt eine gute Schritt-für-Schritt-Anleitung.

## Reihenfolge

Bei der „erweiterten“ Handlungsanweisung ist die gezeigte Reihenfolge der Elemente meistens die sinnvolle Reihenfolge. Aus meiner Sicht kann die Reihenfolge aber je nach Sache auch leicht variiert werden, z.B.:

- Das Software-Konzept kann eventuell im Ziel der Handlung abgehandelt werden.
- Die Regeln können möglicherweise vor Step by Step genannt werden („Beachten Sie bei den folgenden Schritten besonders diese Regeln: ...“)

## Welche Informationen braucht der Anwender?

Bei jeder Handlungsanweisung, die wir schreiben, müssen wir neu überlegen, welche Informationen der Anwender braucht, um handeln zu können.

## Beispiele auf den folgenden Seiten

Auf den folgenden Seiten finden Sie ein paar Beispiele für unterschiedliche Bedarfe und unterschiedliche Zusammenstellungen der Elemente.

## Die „normale“ Handlungsanweisung

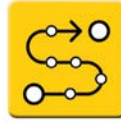
Teilweise ist die klassische Handlungsanweisung richtig:



Überschrift



Ziel der Handlung



Step by Step

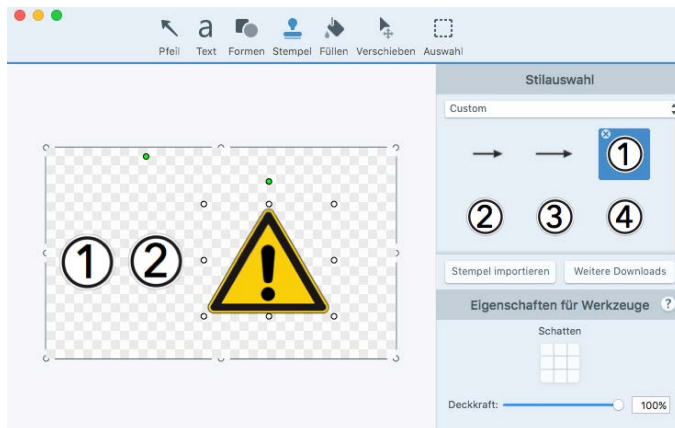


Handlungsergebnis

### Beispiel

#### Eigene Stempel anlegen (SnagIt)

Sie können in SnagIt eigene Stempel anlegen. Das ist praktisch, um eigene Symbole oder Piktogramme einfach einsetzen zu können.



Überschrift

Ziel

Beispiel

- Wählen Sie im Menü **Datei > Importieren > Stempel importieren**.
- Es erscheint ein Dateiauswahlfenster.
- Markieren Sie ein oder mehrere Dateien, die Sie importieren wollen.
- Bestätigen Sie mit **OK**.
- Die Dateien werden importiert und stehen in den Stempeln im Ordner **Custom** zur Verfügung.

Step by Step

Ergebnis



## 5 Funktionsbeschreibungen

*Die Sicht auf Funktionen*

*Funktionen sind keine Handlungen*

*Funktionen sind meistens gut gebündelt*

*Funktionen haben einen Zweck  
und viele Handlungsmöglichkeiten*

## 5.3 Die Funktionsbeschreibung

Ich möchte Ihnen hier eine Systematik anbieten, in der Sie alle Funktionen beschreiben können. Wie bei der erweiterten Handlungsanweisung haben wir wieder diverse Elemente, die je nach Funktion sinnvoll sein können (siehe Kasten unten).

- Sie sehen sicher gleich, dass die Strukturen sich ähneln.
- Sie erkennen Elemente, die zumindest gleich lauten, wie in der „erweiterten“ Handlungsanweisung.
- Wie vorher können die Elemente je nach Bedarf verwendet werden.
- Wie vorher kann die Reihenfolge auch leicht verändert werden.

Funktionsbeschreibung
Überschrift
Zweck der Funktion
Anzeigen und Bedienelemente
SW-Konzept
Voraussetzungen
Aufruf der Funktion
Handlungsmöglichkeiten
Handlungsprinzipien
(Step by Step + Ergebnis)
Schalter/Parameter/Funktionen
Regeln
Ausblick und Bezüge

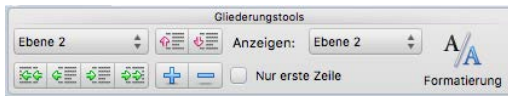
Ziel soll es sein, Funktionen so zu beschreiben, dass der Anwender den Zweck und die Handlungsmöglichkeiten kennt, die Bedienelemente versteht und selber handeln kann. Nur in wenigen Fällen müssen wir ihm dann noch einzelne Step-by-Step-Anleitungen geben.

## Beispiel: Funktionsbeschreibung

### Gliederungsfunktion

In der Gliederungsfunktion können Sie eine Gliederung erstellen oder ein bestehendes Dokument als Gliederung ansehen und bearbeiten.

- Wählen Sie im Menü **Ansicht > Gliederung**.



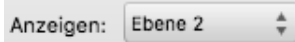
- + Zu dieser Anleitung
- + Planung
  - + Planung der Dimmprofile
  - + Planung der Elektroinstallation
  - + Planung des Servers
- + Installation
- + Bedienung
- + Anhang

Überschrift

Zweck

Aufruf

Anzeigen und Bedienelemente



Sie können festlegen, wie viele Gliederungsebenen angezeigt werden.



Sie können einzelne Überschriften weiter aufklappen (mehr Ebenen) oder weiter zuklappen.



Sie können für jedes Element die Ebene festlegen.

Tab / Shift+Tab

Sie können die markierte Ebene ein- oder ausrücken.

Sie können ...

Handlungsmöglichkeiten

### Gliederung erstellen

### Gliederung bearbeiten

Handlungsanleitungen

Die Gliederungsfunktion ist hier als Funktionsbeschreibung realisiert. Es wird nicht zur Handlung angeleitet. Ergänzend könnten aber kleine Handlungsanleitungen folgen, wie angedeutet.

Vergleichen Sie mit der Handlungsanweisung „Gliederung ansehen“ auf Seite 86.