

SOFTWARE-DOKUMENTATION  
MIT XML – STAND UND NEUE  
ANSÄTZE

**D I P L O M A R B E I T**

IM FACHBEREICH INFORMATIK  
UND KOMMUNIKATIONSSYSTEME  
DER HOCHSCHULE MERSEBURG

vorgelegt von  
Timo Schöpf  
KTD 04-E  
aus Helmbrechts

---

Erster Betreuer:  
Zweiter Betreuer:

Herr Dr. Thomas Meinike  
Herr Dipl.-Phys. Bernd Letz

Eingereicht am:

26.09.2008

---

Ich danke allen Personen, die mich bei der Anfertigung dieser Diplomarbeit unterstützt haben. Ein besonderer Dank gilt meinen Betreuern Dr. Thomas Meinike und Bernd Letz. Weiterhin möchte ich Herrn Siegfried Siegel für die fachliche Beratung recht herzlich danken.

---

---

## Inhaltsverzeichnis

<b>1 Einleitung</b> .....	<b>1</b>
1.1 mediaTEXT Jena GmbH .....	1
1.2 Aufgabenstellung .....	2
1.3 Themenabgrenzung .....	2
<b>2 Single-Source-Prinzip</b> .....	<b>3</b>
2.1 XML .....	3
2.1.1 XML-Codierung .....	3
2.1.2 Dokumententyp-Deklaration (DTD) .....	4
2.1.3 XML-Verarbeitung .....	4
2.2 Single-Source-Publishing .....	5
2.2.1 Desktop-Publishing .....	5
2.2.2 Cross-Media-Publishing .....	6
2.3 Strukturierungsmöglichkeiten .....	7
2.3.1 Topics .....	7
2.3.2 Topicorientierte Strukturierung .....	7
2.3.3 Information Mapping® (IMAP) .....	9
2.3.4 Kombinierung IMAP – topicorientierte Strukturierung .....	10
2.3.5 Modularisierung .....	11
<b>3 Software</b> .....	<b>12</b>
3.1 Was ist Software? .....	12
3.2 Einordnung von Software .....	12
3.2.1 Standardsoftware .....	13
3.2.2 Branchenübergreifende Software .....	13
3.2.3 Branchenspezifische Software .....	13
3.2.4 Eingebettete Software .....	14
<b>4 Software-Dokumentation</b> .....	<b>15</b>
4.1 Einführung in die Technische Dokumentation .....	15
4.1.1 Definitionen .....	15
4.1.2 Anforderungen an Technische Dokumentation .....	16
4.1.3 Dokumentationstätigkeiten .....	17
4.2 Dokumentation in der Software-Entwicklung .....	18
4.3 Besonderheiten der Software-Dokumentation .....	18
4.4 Normen und Richtlinien für die Software-Dokumentation .....	20

---

4.5 Unterschiede zur Hardware-Dokumentation .....	21
4.5.1 Makrostruktur .....	21
4.5.2 Lesetechniken .....	22
4.5.3 Hypertextbasierte Dokumentation .....	23
4.6 Barrierefreiheit .....	23
<b>5 Recherchen zum Thema Software-Dokumentation .....</b>	<b>24</b>
5.1 Fachverband tekcom .....	24
5.2 Ergebnisse der Untersuchung .....	25
<b>6 Arten der Software-Dokumentation .....</b>	<b>27</b>
6.1 Online-Dokumentation .....	28
6.1.1 Online-Hilfen .....	31
6.1.2 Elektronische Handbücher .....	32
6.1.3 Dialoghilfe .....	33
6.1.4 Direkthilfe .....	33
6.1.5 Hilfeagenten (Wizards) .....	34
6.1.6 Tipps .....	34
6.1.7 Computer Based Training .....	35
6.1.8 Lernprogramme .....	35
6.1.9 Auswahlkriterien .....	36
6.2 Gedruckte Handbücher .....	36
6.3 Vergleich Online-Print .....	37
6.3.1 Kontextsensitivität .....	39
<b>7 Dokumentenformate in der Software-Dokumentation .....</b>	<b>41</b>
7.1 DocBook .....	41
7.2 DITA .....	43
7.2.1 Definition .....	43
7.2.2 Topic-Grundtypen .....	44
7.2.3 DITA-Map .....	46
7.2.4 Programme zur Bearbeitung von DITA .....	47
7.2.5 DITA Open Toolkit .....	47
7.3 Information Mapping .....	49
7.4 Formatierungs- und Transformationssprachen .....	50
7.4.1 XSLT .....	50
7.4.2 XSL-FO .....	52

---

---

<b>8 Erstellungsprogramme für Software-Dokumentation</b> .....	<b>53</b>
8.1 Adobe FrameMaker .....	53
8.2 Microsoft HTML Help .....	53
8.3 Microsoft Vista AP Help .....	55
8.4 Microsoft Visual Studio .....	58
8.5 Weitere Autoren- und Erstellungswerkzeuge .....	60
<b>9 Erstellung einer Dokumentationsvorlage</b> .....	<b>61</b>
9.1 Ansatz mit Eclipse .....	61
9.1.1 Konfiguration .....	61
9.1.2 Erstellung von DITA-Dateien .....	62
9.1.3 Ausgabe .....	63
9.2 Ansatz mit FrameMaker 8 .....	64
9.2.1 Strukturierte Anwendung .....	64
9.2.2 Erstellung von DITA-Dateien .....	66
9.2.3 Ausgabe erzeugen .....	68
9.3 Vergleich FrameMaker – Eclipse .....	69
9.3.1 Einlesen/Erstellen .....	69
9.3.2 Bearbeitung .....	70
9.3.3 Ausgabe .....	71
9.4 Erarbeitung eigenständiger Transformationsdateien .....	71
<b>10 Zusammenfassung</b> .....	<b>73</b>
<b>Abkürzungsverzeichnis</b> .....	<b>75</b>
<b>Abbildungsverzeichnis</b> .....	<b>76</b>
<b>Quelltextauszüge</b> .....	<b>77</b>
<b>Quellennachweise</b> .....	<b>78</b>
<b>Eidesstattliche Erklärung</b> .....	<b>84</b>
<b>Anhang</b> .....	<b>A1</b>

# 1 Einleitung

Die Anforderungen an Software werden immer höher. Dadurch muss auch die Dokumentation erweitert und verbessert werden. Oft ist es so, dass Softwareentwickler auch für die Dokumentation verantwortlich sind. Aber viele Entwickler sehen diese mehr als lästiges Übel an. Doch die Benutzerfreundlichkeit und letztendlich auch der kommerzielle Erfolg von Softwareprodukten hängen auch von der Dokumentation mit ab. Aufgrund steigender Komplexität von Softwareprodukten wird auch in Zukunft die Software-Dokumentation eine wichtige Rolle im Software-Engineering einnehmen.

Um Software-Dokumentation zu erstellen, gibt es verschiedene Ansätze und Hilfsmittel. Diese Arbeit soll die gängigen Techniken, aber auch neue Ansätze behandeln. Dafür wurden sowohl Informationen aus der Praxis von Firmenbesuchen als auch theoretische Informationsquellen aus Büchern, Zeitschriften und dem Internet herangezogen.

## 1.1 mediaTEXT Jena GmbH

Das Unternehmen wurde 1995 in Jena gegründet und versteht sich als Dienstleistungsunternehmen mit starker technologischer Ausprägung. Der Firmenname ergibt sich aus der Konzentration auf textorientierte Datenaufbereitung für Publikationen in unterschiedlichsten Medien mit Einbettung von Grafiken, Tabellen, Formeln usw.

Das Leistungsspektrum umfasst die Optimierung des Gesamtkonzeptes für eine durchgängige Publikationstechnologie beim Kunden mit medienneutralen Datenbeständen ebenso wie die Entwicklung einzelner Komponenten und das Erbringen von Leistungen in den Segmenten Mengendatensatz, Aufbau von CD-ROM-Applikationen und die Einbindung der Publikationsdaten in Web-Systeme.

Dienstleistungen rund um das Publizieren aus medienneutralen Daten in XML/SGML sind der Schwerpunkt des Unternehmens. Dabei beginnen die Leistungen oft bei der Erstaufnahme der Altdatenbestände oder dem Aufbau einer Produktionskette zur Erzeugung strukturierter Daten. Weiterhin entwickelt die Firma mediaTEXT auch Softwareanwendungen und -werkzeuge. (vgl. mediaTEXT 2008)

## **1.2 Aufgabenstellung**

Nachdem immer mehr Anfragen nach einer Vorlage für Software-Dokumentation bei der Firma mediaTEXT eingingen, wollte man, neben einem bestehenden Werkzeug für herkömmliche Hardware-Dokumentation, ein weiteres Tool, das speziell für Software-Dokumentation konzipiert wurde, bereitstellen.

Dazu sollte das ganze Thema zunächst theoretisch beleuchtet, die besonderen Anforderungen von Software-Dokumentationen erkannt werden und eine Möglichkeit zur Entwicklung einer Vorlage auf XML-Basis bestimmt werden. Dieser neue Ansatz wurde mit dem Programm FrameMaker und der XML-Technologie DITA vollzogen.

## **1.3 Themenabgrenzung**

Diese Arbeit beleuchtet die theoretischen Hintergründe und praktische Vorgehensmöglichkeiten der Entwicklung von Software-Dokumentation. Dabei wird nicht auf Inhalte, sprachliche Besonderheiten oder die didaktisch korrekte Gestaltung der Dokumentationen eingegangen. Es wird nur der technische Aspekt dieser Dokumentationsarten betrachtet.

## 2 Single-Source-Prinzip

Es ist anzustreben, dass aus einer Datenquelle verschiedene Ausgabeformate hergestellt werden können. Dies erspart Zeit- und Kostenaufwand und ist in der modernen Informationstechnologie nicht mehr wegzudenken. Um das Single-Source-Prinzip umzusetzen, sind verschiedene Techniken und Programme nötig.

### 2.1 XML

XML (Extensible Markup Language) ist eine erweiterbare Auszeichnungssprache. Sie dient zum hierarchischen Strukturieren elektronischer Dokumente. XML trennt Inhalt, Struktur und Layout, dieses vereinfacht die Informationsverarbeitung erheblich. Dadurch eignen sich XML-Dateien sehr gut für Single-Source-Publishing. XML ist eine Ableitung, das heißt eine vereinfachte Form, von SGML (Standard Generalized Markup Language), welche durch den ISO-Standard ISO 8879:1986 normiert ist. Während SGML wegen der hohen Komplexität keine große Beliebtheit in der Praxis fand, hat sich XML mittlerweile durchgesetzt.

XML wurde 1996 von einer Arbeitsgruppe entwickelt, die unter der Schirmherrschaft des World Wide Web Consortium (W3C) gegründet wurde. XML ist eine W3C-Empfehlung und gilt daher als Web-Standard. (vgl. Mintert 2003)

#### 2.1.1 XML-Codierung

Der mit XML ausgezeichnete Text unterscheidet sich vom normalen Text durch die Anfangs- und Endtags. Durch diese wird es für die Maschine möglich, die Informationen zuzuordnen und weiterzuverarbeiten. Jedes einzelne Element kann angesprochen werden. Dazu wird oft die XPath-Language verwendet.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Handlungsanweisung SYSTEM "beispiel.dtd">
<Handlungsanweisung>
  <Schritt Nr="1">
    <Aufgabe>Neues Absatzformat gestalten</Aufgabe>
    <Handlung>Wählen Sie Befehle &gt; Neues Format.</Handlung>
    <Handlung>Stellen Sie im Absatzgestaltungskatalog die Gewünschten
    Formate ein.</Handlung>
    <Resultat>Das neue Absatzformat erscheint im Absatzformatkatalog.
    </Resultat>
  </Schritt>
</Handlungsanweisung>
```

Beispiel: 2-1: Quelltext eines Handlungsschrittes mit XML-Auszeichnung

## 2.1.2 Dokumententyp-Deklaration (DTD)

Zur XML-Datei sollte eine DTD (Document Type Definition) vorhanden sein. Diese besteht aus Elementtypen, Attributen von Elementen, Entitäten und Notationen und gibt die Verschachtelung der Elemente, sowie die Art des Inhalts von Attributen vor. Damit bestimmt sie die Struktur eines XML-Dokumentes.

```
<!ELEMENT Handlungsanweisung (Schritt+)>
<!ELEMENT Schritt (Aufgabe, Handlung*, Resultat*)>
<!ELEMENT Aufgabe (#PCDATA)>
<!ELEMENT Handlung (#PCDATA)>
<!ELEMENT Resultat (#PCDATA)>
<!ATTLIST Schritt Nr CDATA #REQUIRED>
```

Beispiel: 2-2: DTD zum strukturiertem Handlungsschritt aus Beispiel 2-1

## 2.1.3 XML-Verarbeitung

Zur Verarbeitung der XML-Daten stehen verschiedenen Technologien zur Verfügung. Damit können die Daten formatiert und in andere Ausgabeformate konvertiert werden. Diese Techniken sind (siehe auch Kapitel 7.4):

- XSLT (Extensible Stylesheet Language – Transformation)
- XSL-FO (Extensible Stylesheet Language – Formatting Objects)

Um ein Ausgabeformat zu erzeugen, empfiehlt es sich, einen Parser zu verwenden. Diese sind je nach Transformationstechnologie unterschiedlich und sind teilweise frei im Internet downloadbar.

Weiterhin werden verschiedene Dokumentenformate auf XML-Basis verwendet. Die für die Technische Dokumentation am relevantesten sind:

- DITA (Darwin Information Typing Architecture)
- DocBook

## **2.2 Single-Source-Publishing**

Single-Source-Publishing bezeichnet das Publizieren aus einer medienneutralen Quelle. Diese ist ein Einzeldatei oder eine Ansammlung von XML- oder SGML-Dateien. Bekannte Programme zur Umsetzung von Single-Source-Publishing sind FrameMaker, InDesign oder QuarkXPress.

Zentrale Aspekte von Single-Source-Publishing sind eine möglichst flexible und weit reichende Verwendung der erstellten Inhalte (vgl. Kapitel 2.2.2). Für Single-Source-Publishing kommt eine Strukturierungsmethode zum Einsatz, die Inhalte in wieder verwendbare Module (Topics) zerlegt. Aus denen lassen sich nach dem Baukastenprinzip passende Dokumente zusammensetzen (vgl. Closs 2007, S. 27). Diesen Arbeitsablauf nennt man topicorientierte Strukturierung (vgl. Kapitel 2.3.2).

### **2.2.1 Desktop-Publishing**

Desktop-Publishing ist die soft- und hardwaremäßige Unterstützung bei der Erstellung von Druckschriften auf dem Computer. Es umfasst die Aufbereitung und Zusammenführung von Texten, Grafiken und Bildern bis zur Gestaltung des Layouts. Moderne Systeme bieten die komplette Software-Unterstützung, sie arbeiten mit eigener Bilderfassung mit einem Scanner und erstellen die Farbauszüge

nach entsprechenden Farbmodellen. Ferner haben sie Tools für die Typografie, Bildbearbeitung und Layout-Erstellung und können Dateien ausgeben, die zur unmittelbaren Belichtung der Druckplatten benutzt werden. (vgl. Datacom 2008)

DTP hatte seinen Ursprung in den achtziger Jahren und war die erste Möglichkeit, Single-Source-Publishing umzusetzen. Damals fanden Programme wie Interleaf, QuarkXPress und FrameMaker regen Zuspruch. (vgl. Closs 2007, S. 18)

Nachdem FrameMaker ab der Version 5.0 SGML unterstützte, kann das Programm ab der Version 7.0 auch XML verarbeiten. Die neueste Version 8.0 unterstützt zudem DITA. Heute ist FrameMaker ein beliebtes Programm für die Erstellung von Dokumentationen und findet häufig in der Praxis Anwendung. Die Firma mediaTEXT verwendet das Programm für den Satz, Siegfried Siegel von CARSTENS + PARTNER GmbH, München, bezeichnet FrameMaker als eines der besten Produkte für die Technische Dokumentation. (Siegel 2008a)

### **2.2.2 Cross-Media-Publishing**

Die Begriffe Single-Source-Publishing und Cross-Media-Publishing sind verwandt. Manchmal werden beide Begriffe synonymisiert, was jedoch nicht ganz korrekt ist, da Cross-Media das Publikationsverfahren und Single-Source die Quelle beschreibt.

Nach Heinig (2006) wurde Cross-Media-Publishing ursprünglich in der Agenturumgebung geprägt und bezeichnet den parallelen Einsatz verschiedener Werbemedien. Dabei sollten die verschiedenen Werbepublikationen im besten Falle so aufeinander aufbauen, dass sie den Werbeerfolg gegenüber unabhängigen Einzel-Publikationen steigern. Cross-Media-Publishing im IT-Bereich bezeichnet die parallele Publikation in mehrere unterschiedliche Medien. Dabei dient eine medienneutrale Datei als Quelle (Single-Source).

Erfolgreich wurde Cross-Media-Publishing umgesetzt, wenn aus einer Quelle beispielsweise eine Online-Hilfe, eine Druckversion im PDF-Format und möglicherweise eine Anwendung für mobile Endgeräte erzeugt werden kann. Für die Software-Dokumentation ist Cross-Media-Publishing prädestiniert, da aus einem Format meist eine Druckversion und eine Online-Hilfe entstehen soll.

## **2.3 Strukturierungsmöglichkeiten**

Um den Single-Source-Gedanken zweckmäßig umzusetzen, ist eine sinnvolle Strukturierung der Informationen nötig. Dazu gibt es verschiedene Ansätze, die im Zusammenhang mit dem Single-Source-Publizieren zu nennen sind.

### **2.3.1 Topics**

*„Ein Topic ist ein in sich abgeschlossener, möglichst kontextunabhängiger Inhaltsblock, der einer vorgegebenen Systematik genügt und eine Kernaussage enthält.“ (Closs 2007, S. 37)*

Zwischen Topics können Beziehungen in Form von Verweisen (Links) bestehen. Werden mehrere Topics zusammengefasst, nennt man dieses neue Dokument Informationsgebilde. Topics sind in Klassen einteilbar, die daraus entstehende Topic-Klasse ist eine Menge von Topics der gleichen Art. Sie dienen sowohl als Modelle für die Erstellung und Aufteilung von Inhalten als auch als Basiseinheiten für die Standardisierung. Eine Topic-Klasse enthält einen Namen und kann grafisch dargestellt werden. (vgl. Closs 2007, S. 39)

### **2.3.2 Topicorientierte Strukturierung**

Die topicorientierte Strukturierung hat das Ziel, jedes Thema isoliert und kontextunabhängig in einem Baustein zu beschreiben. Der Baustein wird Topic genannt.

Diese Strukturierungsmöglichkeit ist für Single-Source-Publishing die beste Wahl. (vgl. Closs 2007, S. 29)

Topicorientierte Strukturierung ist keine neue Technik. Sie kann genauso gut auf dem Papier produziert werden. Der Begriff Topic wurde mit Aufkommen der Online-Hilfen eingeführt. Auch DITA arbeitet mit Topics und ist daher eine gute Wahl zur Umsetzung von topicorientierter Strukturierung. (vgl. Closs 2007, S. 36)

Folgende Vorteile ergeben sich durch die topicorientierte Strukturierung (nach Closs 2007, S. 32):

- Redundanzfreiheit  
Jede Begriffserklärung kommt nur einmal vor.
  
- Systematik (Standardisierung)  
Jede Begriffserklärung ist nach demselben Muster geschrieben. Aufbau, Beschreibungsstil und Darstellung sind für alle Begriffserklärungen einheitlich.
  
- Mehrautorenfähigkeit  
Viele Autoren können Begriffserklärungen unabhängig voneinander liefern.
  
- Wiederverwendbarkeit  
Eine Begriffserklärung kann nicht nur in vielen verschiedenen Begriffssammlungen vorkommen, sondern auch an anderen Stellen, an denen der Begriff erklärt werden soll.
  
- Vielfalt (Varianz)  
Aus einem Bestand von Begriffserklärungen können für unterschiedliche Zwecke unterschiedliche Arten von Begriffssammlungen zusammengestellt werden.
  
- Skalierbarkeit  
Eine einmal bestehende Begriffssammlung kann beliebig wachsen, ohne dass vorhandene Inhalt umstrukturiert oder neu erstellt werden müssen.

- **Effektivität**  
Der Zugang ist durch die alphabetische Anordnung der Begriffserklärungen schnell und effektiv.

### **2.3.3 Information Mapping® (IMAP)**

Die Information Mapping®-Methode wurde 1967 von Prof. Robert E. Horn entwickelt und sollte eine lesergerechte Aufbereitung der Informationen erzielen. Darin finden sich Grundlagen der Lernpsychologie und Wahrnehmungsforschung wieder. Die Methode ist mit der Topic-Einteilung verwandt. Allerdings sind bei Information Mapping die beiden Informationseinheiten Map und Block für die Strukturierung der Inhalte vorgesehen. (vgl. Closs 2007, S. 111)

Wie auch die topicorientierte Strukturierung ist auch Information Mapping ohne technische Hilfsmittel möglich. Die Methode kann auch mit Stift und Papier umgesetzt werden.

Information Mapping basiert nach Siegel (2008) auf vier Hauptbestandteilen:

- **sieben Prinzipien**
  - Betitelungssprinzip
  - Gliederungssprinzip
  - Relevanzprinzip
  - Einheitlichkeitsprinzip
  - Auswahl der optimalen Darstellung
  - Verfügbarkeit von Einzelheiten
  - Systematische Gliederung
- **zwei Informationseinheiten**
  - Map
  - Block

- **sieben Informationsarten mit Darstellungsempfehlungen**

- Anleitung
- Prozess
- Struktur
- Begriff
- Prinzip
- Fakt
- Klassifikation

- **lesergerechte Analyse**

Bei der IMAP-Methode gibt es mit Map und Block zwei Informationseinheiten mit klaren Regeln. Ein Block besteht aus nur einer für den Leser relevanten Informationseinheit, die eine konkrete Aussage enthält. Weiterhin besitzt ein Block auch einen Titel, der klar den Inhalt beschreibt. Eine Map ist die Zusammenstellung von mehreren Blöcken. (vgl. Böhler 2008)

Die Methode ist durch Warenzeichen und Handelsmarken geschützt, deren Rechte bei Information Mapping, Inc. liegen. Die Methode wird durch zertifizierte Partner in Seminaren geschult. Bei Erlernen der Methode sind Lizenzgebühren fällig, die Anwendung später ist jedoch frei. Die Firma CARSTENS + PARTNER GmbH, München, ist seit 1996 zertifizierter Betrieb für die Information-Mapping-Methode und setzt diese auch in den Projekten um. (vgl. Siegel 2008) Ein IMAP-Beispiel von der Firma CARSTENS + PARTNER GmbH, München, ist im Anhang dieser Diplomarbeit zu finden.

### **2.3.4 Kombinierung IMAP – topicorientierte Strukturierung**

Die Einteilung der Informationen in Informationseinheiten ist der Topic-Einteilung nahe stehend, wobei ein Topic weniger strikt festgelegt ist. Die Informationsarten sind vergleichbar mit Topic-Klassen. Das Vorgehen bei der Erstellung einer Informationsart ist nahezu analog zur Definition einer Topic-Klasse. Allerdings ist die feste Vorgabe von den sieben Informationsarten bei IMAP für den praktischen Gebrauch zu unflexibel. (vgl. Closs 2007, S. 111)

### **2.3.5 Modularisierung**

Modularisieren ist das Zerlegen der Dokumentation in einzelne Bausteine (Module). Der Inhalt der Dokumentation wird dabei in Module aufgeteilt, welche je nach Zielgruppe oder Produkt ausgetauscht oder ersetzt werden können.

Die Prinzipien der Information-Mapping-Methode bilden eine gute Grundlage für die Modularisierung. Durch die Informationseinheiten Map und Block werden die Informationen modularisiert. Auch DITA bildet durch die topic-orientierte Strukturierung modulare Informationseinheiten. Eine weitere Modularisierungstechnik stellt die Funktionsdesign®-Technik dar, auf die in dieser Diplomarbeit nicht näher eingegangen wird.

## 3 Software

Das Wort „Software“ existiert seit dem Jahr 1957. Es wurde von John W. Tukey als Kunstwort dem schon existierenden Wort „Hardware“ gegenübergestellt. Hardware bezeichnete früher alle Gegenstände, die aus Metall hergestellt wurden. Später stand Hardware für alle materiellen Teile an Rechenmaschinen. (vgl. Ludewig/Lichter 2007, S. 34)

### 3.1 Was ist Software?

*„Software ist, was nicht Hardware ist.“* (Ludewig/Lichter 2007, S. 34)

Doch dieses reicht als Definition nicht aus. Die Bedeutung ist bis heute umstritten. Eine Definition der IEE Computer Society lautet:

*„**software** – Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.“* (IEEE 610.12/1990)

Software umfasst demnach Programme, Abläufe, Regeln, Dokumentation und Daten, die mit dem Betrieb eines Rechners zu tun haben. Software ist ein technisches Produkt, das systematisch von Ingenieuren entwickelt werden kann und durch feststellbare Eigenschaften wie Funktionalität oder Qualität gekennzeichnet ist. (vgl. Ludewig/Lichter 2007, S. 34)

### 3.2 Einordnung von Software

Software wird im Allgemeinen in zwei Gruppen eingeteilt: Systemsoftware und Anwendungssoftware. Systemsoftware enthält die Basisfunktion des Computers.

Anwendersoftware ist zur Unterstützung der verschiedenen Aufgaben von Anwendern bestimmt. In dieser Arbeit soll der Schwerpunkt auf der Dokumentation von Anwendersoftware liegen.

Zur Anwendersoftware gehören:

- Standardsoftware
- branchenübergreifende Software
- branchenspezifische Software
- eingebettete Software

### **3.2.1 Standardsoftware**

Diese Software richtet sich an einen großen Anwenderkreis und besitzt hohen Bekanntheitsgrad (Textverarbeitungsprogramme usw.). Menü und Werkzeugleisten sind nach ergonomischen Gesichtspunkten gestaltet. (vgl. Grünwied 2007, S. 2)

### **3.2.2 Branchenübergreifende Software**

Das sind Programme, die in kaufmännischen Abteilungen von Industriebetrieben aus den verschiedensten Branchen eingesetzt wird (Finanzbuchhaltungsprogramme, Lagerverwaltung usw.). Der Software liegt in der Regel eine Datenbank zugrunde. Diese Programme sind mitunter sehr komplex und es ist eine Schulung der Mitarbeiter erforderlich. (vgl. Grünwied 2007, S. 3)

### **3.2.3 Branchenspezifische Software**

Es handelt sich dabei um Software für den spezifischen Einsatz in einer Branche. Für Technische Redakteure wären beispielsweise Content Management Systeme oder Autorensysteme hier anzuführen. Die Software wird für branchenübliche Aufgaben verwendet. Auch hier müssen Mitarbeiter geschult werden. Je besser die Software zu den Abläufen in der Branche passt, umso schneller finden sich die Anwender darin zurecht. (vgl. Grünwied 2007, S. 4)

### **3.2.4 Eingebettete Software**

Diese Art von Software findet man in eingebetteten Systemen, die wiederum Bestandteile von größeren mechanischen oder mechatronischen Produkten sind. Beispiele hierfür sind Fahrkartenautomaten, Navigationssysteme in Fahrzeugen oder moderne Waschmaschinen.

Ein eingebettetes System kommt häufig von einem anderen Hersteller und besteht aus Software- und Hardwarekomponenten. Die Software muss nicht explizit installiert werden, sie ist schon im System integriert. Für die Dokumentation dieser Software gelten besondere Anforderungen. Sie ist ein Teil der gesamten Produktdokumentation und sollte darin integriert werden. Oft muss sich die Software selbst erklären und die Bedienung soll möglichst einfach gehalten werden. Von daher ist eine ergonomische Benutzerführung schon von Programmierseite aus erforderlich. Wird zudem eine Dokumentation erstellt, ist eine Online-Hilfe ins System zu integrieren.

## 4 Software-Dokumentation

*„Dokumentation gilt als ewiges Sorgenkind der Software-Entwicklung.“ (Ludewig/Lichter 2007, S. 241)*

### 4.1 Einführung in die Technische Dokumentation

*„Die traditionelle Auffassung von Dokumentation war lange Zeit vom statischen Gesichtspunkt des Sammelns, Ordnen und Speicherns geprägt. Erst mit der wachsenden Bedeutung des Informationsbegriffs bzw. der Information setzte sich die Erkenntnis durch, dass Dokumentation nur in Verbindung mit der Wahrnehmung von Informationsaufgaben verstanden werden kann. Dokumentation wird daher zweckorientiert definiert als Sammeln, Erfassen, Ordnen und Aufschließen von Dokumenten sowie deren Aufbereitung für Zwecke der Information.“ (Lehner 1994, S. 2)*

Software-Dokumentation gehört zur Technischen Dokumentation. Diese umfasst alle erforderlichen Informationen zum Produkt. Dazu gehören sowohl Bedienungs- und Gebrauchsanleitungen sowie auch Werbematerialien und Prospekte, die das Produkt näher beschreiben.

#### 4.1.1 Definitionen

Nach DIN 6789 lässt sich Technische Dokumentation wie folgt definieren:

*„Technische Dokumentationen umfassen sowohl direkt oder mit optischen Hilfsmitteln lesbare Unterlagen als auch solche Dokumente, die nicht den Kriterien für Unterlagen genügen (z.B. Magnetbänder).“*

*Ein Dokument ist eine als Einheit gehandhabte Zusammenfassung oder Zusammenstellung informationstragender Daten.*

*Eine Dokumentation ist die für einen bestimmten Zweck vollständige Zusammenstellung von Dokumenten.“*

(DIN 6789 Dokumentationssystematik, Aufbau Technischer Erzeugnis-Dokumentationen, Entwurf Feb. 1986)

In der Technischen Dokumentation wird nach DIN 4500 zwischen interner und externer Dokumentation unterschieden. Die interne Dokumentation umfasst z. B. Fertigungsunterlagen, Anweisungen an die Mitarbeiter oder Qualitätssicherungsdokumente. Zur externen Dokumentation zählen sämtliche Benutzerinformationen, die direkt für den Kunden bestimmt sind, wie Betriebsanleitung, Gebrauchsanweisung oder Prospekte.

Technische Dokumentation erfüllt eine kommunikative Funktion und steht zwischen dem Hersteller, dem Produkt und dem Anwender. Sie muss für jede Anwendergruppe verständlich und zielgruppengerecht aufbereitet werden. Der Technische Redakteur gilt dadurch als Mittler zwischen dem Laien und dem Experten und muss selbst komplizierte Zusammenhänge verständlich erklären.

#### **4.1.2 Anforderungen an Technische Dokumentation**

Damit Technische Dokumentation ihren Zweck der korrekten und vollständigen Instruktion erfüllt, muss sie bestimmten Anforderungen genügen. Die folgenden Punkte sollte die Technische Dokumentation nach Edelmann (2007) aufweisen:

- effektive und sichere Nutzung des Produktes
- sachliche Richtigkeit der Inhalte
- mediengerechte Gestaltung der Dokumentation
- Lesbarkeit und Verständlichkeit
- Handlungsorientiertheit
- kostengünstige Erstellung und Aktualisierung
- Einhaltung der gesetzlichen Bestimmungen

Die DIN 66270 stellt folgende spezielle Anforderungen an die Software-Dokumentation.

- Korrektheit
- Vollständigkeit und Relevanz
- Konsistenz und Redundanz
- Verständlichkeit und Lesbarkeit
- Eindeutigkeit und Präzision
- Navigierbarkeit
- Verfolgbarkeit
- Änderbarkeit
- Verwaltbarkeit
- Vorgabenkonformität

#### **4.1.3 Dokumentationstätigkeiten**

Es können nach Krauß (2007, S. 33) vier grundlegende Tätigkeiten unterschieden werden:

- Erstellen eines Dokuments  
Ein Autor oder ein Generator erstellt ein Dokument, indem er Aussagen über beobachtete oder erdachte Sachverhalte dokumentiert, also Aussagen darüber in einem Dokument ablegt.
- Verändern eines Dokuments  
Der Unterschied zur Erstellung besteht darin, dass der Autor oder Generator in einem vorhandenen Dokument enthaltene Aussagen durch neue Aussagen ersetzt oder die ersatzlos streicht.
- Lesen eines Dokuments  
Beim Lesen entnimmt ein Leser oder Interpret Informationen aus einem dargestellten Dokument. Für die Lesetätigkeit selbst ist es unerheblich, was der Leser oder Interpret mit den gewonnenen Informationen macht. Darum fällt auch die Ausführung eines Programms unter die Tätigkeit Lesen.

- Verwalten eines Dokumentes

Bei der Dokumentenverwaltung veranlasst ein Verwalter verschiedene Tätigkeiten in der Umgebung, die das Dokument betreffen. Das kann in der Folge zu Veränderungen von anderen Teilen der Dokumentation, aber auch des Dokuments selbst führen.

## **4.2 Dokumentation in der Software-Entwicklung**

Schon beim Software-Engineering wird eine systematische Dokumentation gefordert. Software-Entwickler haben in ihrer Ausbildung das Programmieren gelernt, nicht aber das Dokumentieren. (vgl. Ludewig/Lichter 2007, S. 248) Demzufolge wird Software-Dokumentation als notwendiges Übel betrachtet und nur ungern ausgeübt. Das führt zu einer verhältnismäßig schlechten Dokumentation des Softwareproduktes.

## **4.3 Besonderheiten der Software-Dokumentation**

Software wird immer umfangreicher und komplexer. Dadurch werden auch höhere Ansprüche an die Dokumentation dieser Programme gestellt. Oft reicht dabei ein Handbuch nicht mehr aus, es gibt beispielsweise Anfänger-, Fortgeschritten- und Referenzhandbücher.

Software-Dokumentation gliedert sich, wie die Technische Dokumentation im Allgemeinen, in die interne Entwicklungs- und die externe Anwendungsdokumentation. Die Anwendungsdokumentation enthält die Handlungsanweisungen für die Software. Der Trend geht dabei verstärkt in Richtung Online-Dokumentation, in der Praxis wird aber oft immer noch zusätzlich eine Print-Version geliefert.

Dokumentieren ist eine Daueraufgabe. Weit verbreitet ist es, nach Ende der Codierung zu dokumentieren. Dies führt jedoch zu einer schlechten Dokumentation, da manche Informationen schlichtweg vergessen werden. (vgl. Ludewig/Lichter

2007, S. 241) Deshalb sollte die Dokumentation immer gleichzeitig mit dem Produkt entstehen. Dabei ist es unabhängig, ob Software oder Hardware dokumentiert wird. Meist ist es der Fall, dass die Software weiterentwickelt wird und neue Releases folgen. Dabei muss auch die vorhandene Dokumentation immer wieder angepasst und vervollständigt werden.

In der Software-Dokumentation wird unterschieden in integrierte Dokumentation und separate Dokumentation. Integriert bedeutet in der Codierung eingebunden. Ein Beispiel hierfür wären Kommentare oder die Bezeichner der Variablen. Unter separater Dokumentation versteht man den Teil der Software, der nicht in den Programmen enthalten ist. (vgl. Ludewig/Lichter 2007, S. 241)

Dirnfeldner (1996) unterscheidet je nach Kenntnisstand der Anwender verschiedene Dokumentationsarten. Dieses Bild verdeutlicht am Beispiel der Programmier-Software für speicherprogrammierbare Steuerung (SPS), wie Software-Dokumentation nach Aufgaben gegliedert sein kann.

Dokumentationsart	Zielgruppe	Aufgabe	Inhalt
Lernanleitung	Einsteiger ohne Vorkenntnisse	Grundfunktionen und Handhabung vermitteln	Lernziele, Funktionen, grundsätzliche Handhabung, Beispiele
Schnellanleitung	Umsteiger mit Vorkenntnissen	Schnellen Einstieg in die neue Software ermöglichen	Hauptfunktionen anhand eines praktischen Beispiels
Nachschlageanleitung	Anwender	Fachgerechte Nutzung aller Funktionen ermöglichen	Elemente, Funktionen, Fehlermeldungen, Stichwortregister, Glossar
Taschendokumentation	geübter Anwender	"Gedankenstütze"	Bedienoberfläche, Beschleunigertasten, Befehlsübersicht, Funktionsbausteine

Abb. 4-1: Mögliche Gliederung einer Software-Dokumentation (Quelle: www.tekom.de)

Weiterhin gibt es bei der Arbeitsweise leichte Abweichungen zur Hardware-Dokumentation. Bei der Arbeit am Computer müssen leicht veränderte Arbeitsschritte ausgeführt werden. Norman (1987) formulierte sieben Schritte, die bei der Bearbeitung einer Aufgabe am Computer vollzogen werden:

1. Formulierung des Ziels
2. Formulierung der Absicht
3. Spezifizieren der Handlung
4. Ausführen der Handlung
5. Wahrnehmen des Systemzustandes
6. Interpretieren des Zustandes
7. Evaluierung des Ergebnisses

#### **4.4 Normen und Richtlinien für die Software-Dokumentation**

Für die Software-Dokumentation existieren Normen zur Gestaltung etc. Weiterhin finden viele Normen für die allgemeine Technische Dokumentation Anwendung. Zudem werden von Firmen oder Instituten interne Richtlinien für die Gestaltung von Software-Dokumentation herausgegeben. Ein Beispiel ist die Richtlinie von der Physikalisch-Technischen Bundesanstalt in Braunschweig 2005 für deren Softwareentwicklung (siehe Anhang).

Normen haben im Gegensatz zur nationalen Gesetzgebung und den EU-Richtlinien keine juristische Bedeutung. Sie geben Empfehlungen und Mindestanforderungen vor und verkörpern Praxisbezogenheit. In den letzten Jahren wurden viele Normen grundlegend überarbeitet. So befasste sich beispielsweise die Norm IEEE 1063 in ihrer ursprünglichen Form ausschließlich mit gedruckter Software-Dokumentation. Mittlerweile berücksichtigt die Auflage von Dezember 2001 auch die elektronischen Medien. (vgl. Grünwied 2007, S. 53)

## 4.5 Unterschiede zur Hardware-Dokumentation

Der große Vorteil an der Software-Dokumentation ist die ständige Verfügbarkeit eines Bildschirms. Deshalb sollte dieses Ausgabemedium bestmöglich eingebunden werden. Somit kann in der Software-Dokumentation oft mit elektronischen Handbüchern, Online-Hilfen und Tutorials direkt am Bildschirm gearbeitet werden. Zudem sollte noch eine gedruckte oder druckbare Version existieren.

Hardware-Dokumentation hingegen besteht oft ausschließlich aus einer gedruckten Version. In manchen Fällen liegt diese auch als elektronisches Handbuch vor, das am Bildschirm gelesen werden kann. Ist dies der Fall, sollte aber unbedingt eine Funktion zum Drucken angeboten werden. Jedoch kann auch bei Gebrauchsgütern Online-Dokumentation zum Einsatz kommen, durch die Zugabe einer animierten Gebrauchsanweisung oder eines Videos für die Inbetriebnahme kann insbesondere bei komplizierten Geräten ein deutlicher Mehrwert der Dokumentation erreicht werden.

### 4.5.1 Makrostruktur

Auch die Makrostruktur bei Handbüchern in der Software-Dokumentation unterscheidet sich in einigen Dingen von der Hardware-Makrostruktur. So sind einige Kapitel wie Aufbau, Reinigung oder Entsorgung nicht vorhanden. Dafür sind Änderungsprotokolle oder Austauschchanweisungen bei Bedarf möglich. Hoffmann/Schlummer (1990) gaben eine mögliche Makrostruktur für Softwaredokumentationen vor:

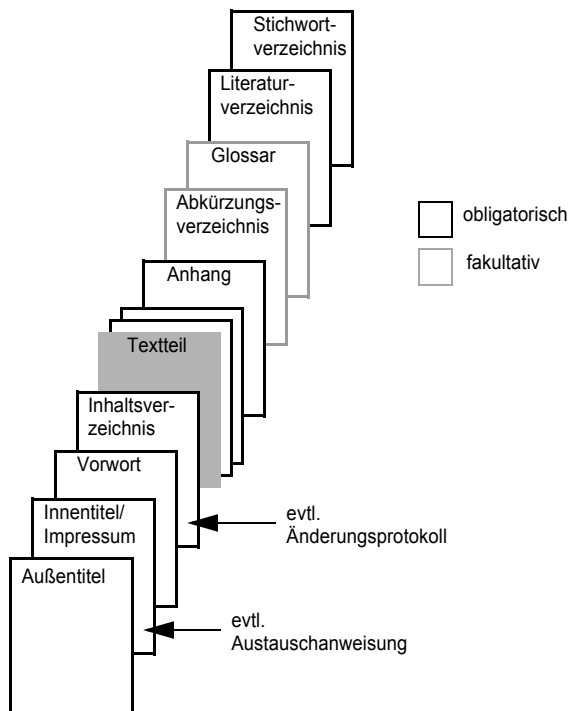


Abb. 4-2: Makrostruktur von Softwarehandbüchern (nach Hoffmann/Schlummer 1990)

#### 4.5.2 Lesetechniken

Die Lesetechniken unterscheiden sich je nach Ausgabemedium. Ein Handbuch, auch elektronische Handbücher als PDF, wird linear gelesen. Das erfordert einen klaren Aufbau und sollte beispielsweise mit der Installation beginnen. Auch eine Seiten- und Kapitelnummerierung ist nötig. Der Benutzer muss durch das Programm geführt werden. Online-Hilfen werden punktuell gelesen und in den Kapiteln wird „gesprungen“. Dazu sind Verlinkungen möglich. Die Online-Hilfe wird nicht komplett gelesen, sondern soll bei Problemen oder Unklarheiten in der Bedienung der Software zur Verfügung stehen. Seiten oder Kapitel müssen nicht nummeriert werden.

### 4.5.3 Hypertextbasierte Dokumentation

Der Begriff „Hypertext“ wurde im Jahre 1965 von Ted Nelson geschaffen. (vgl. W3C 2008) Hypertext bezeichnet einen Text, der nicht zum linearen Lesen konstruiert wurde, sondern aus vernetzten Texteinheiten besteht. Weiterhin enthält dieser Text Hyperlinks, die auf andere Texte oder Dateien verweisen. (vgl. Münz 2007)

Diese spezielle Strukturierung macht das Lesen von Bildschirmtexten einfacher. Dadurch haben diese Texte einen Mehrwert im Vergleich zu herkömmlichen Buchtexten, die meist linear gelesen werden müssen.

Werden multimediale Inhalte wie Bilder, Töne oder Videos eingebunden, spricht man von „Hypermedia“. Das Präfix „Hyper-“ steht also für eine bestimmte Art von Organisation der einzelnen Text- oder Multimediaeinheiten. (vgl. Münz 2007)

### 4.6 Barrierefreiheit

Da Software-Dokumentation häufig auf dem Bildschirm gelesen wird, sollte diese auch barrierefrei gestaltet werden. Unter Barrierefreiheit versteht man die behindertengerechte Gestaltung. Darstellungen auf dem Bildschirm sollen auch für Menschen mit Behinderung verfügbar gemacht werden. Dazu gibt es eine Reihe von Vorgaben, die im amerikanischen Gesetz „Section 508“ festgelegt sind. Manche Autorensysteme ermöglichen es, diese Vorgaben während des Kompilierungsvorgangs für Web-basierte Hilfen zu konfigurieren. (vgl. Grünwied 2007, S. 89)

## 5 Recherchen zum Thema Software-Dokumentation

Zum ersten Mal tauchte Software-Dokumentation im großen Rahmen für die Windows-Betriebssysteme auf. Damals wurden noch größere Handbücher zum Softwareprodukt mitgeliefert. Diese waren hilfreich und enthielten gute Anweisungen, jedoch war das Einlesen in ein solches Buch nicht immer einfach. Recherchen zum Thema Software-Dokumentation

In den folgenden Jahren wurde die Dokumentation für Softwareprodukte immer wichtiger. Der Erfolg einer Software hängt auch von der Güte der Dokumentation ab und es wurde mehr und mehr Wert darauf gelegt. Das spiegelt sich auch in den Publikationen über dieses Thema wieder.

Die Diplomarbeit soll deshalb auch beleuchten, wie oft das Thema in der Vergangenheit in Fachzeitschriften und auf Tagungen angesprochen wurde. Dafür wurde eine Untersuchung angestellt, wie oft Software-Dokumentation in den vergangenen Jahren vorkam. Um dieses aufzuschlüsseln, wurde die vom Verband „tekomp“ herausgegebene Fachzeitschrift „Technische Kommunikation“ bis ins Jahr 2000 analysiert, die Internetrecherche nach Artikeln geht bis 1992 zurück. Ferner wurden auch die Vortragsthemen bei den tekomp-Tagungen ab dem Jahr 2003 durchleuchtet. Bei der Recherche wurde auch das Vorkommen von Artikeln über Online-Dokumentation und -Hilfen mit berücksichtigt. Diese Themen stehen oft im unmittelbaren Zusammenhang mit Software-Dokumentation.

### 5.1 Fachverband tekomp

„tekomp“ bezeichnet die Gesellschaft für Technische Kommunikation e. V. mit dem Sitz in Stuttgart. Die Gesellschaft ist der größte Fachverband für Technische Kommunikation in Europa. Sie wurde 1978 gegründet. (vgl. tekomp 2008)  
Ein großes Anliegen des Fachverbandes ist die Aus- und Weiterbildung und die Professionalisierung der Mitglieder. Derzeit hat die tekomp knapp 6000 Mitglieder

aus allen Wirtschaftsbranchen europaweit. Eine Firmenmitgliedschaft halten 441 Unternehmen. In den vergangenen Jahren betrug der Zuwachs an Neumitgliedern 10 bis 15 Prozent pro Jahr. Die Gesellschaft veranstaltet zweimal pro Jahr Tagungen, auf denen hochrangige und renommierte Referenten die neuesten Entwicklungen in der Technischen Kommunikation präsentieren. (vgl. tekomp 2008)

In Deutschland ist die tekomp in 17 Regionalgruppen unterteilt, die eigene Tagungen mit zum Teil externen Referenten abhalten. Die Regionalgruppen erarbeiten auch eigene Publikationen zu ausgewählten Themen.

## **5.2 Ergebnisse der Untersuchung**

Es wurde zur Demonstration des Vorkommens des Themas Software-Dokumentation in Fachartikeln die Fachzeitschrift „Technische Kommunikation“, welche die tekomp herausgibt, herangezogen. Zudem sind die Themen der tekomp-Tagungen in den vergangenen Jahren beleuchtet worden.

Bei der Analyse der tekomp-Fachzeitschriften zeigt sich, dass das Thema im Jahr 1992 zum ersten Mal im größeren Rahmen auftauchte. Damals wurden vor allem Software-Handbücher und Strukturen der Dokumentationen unter die Lupe genommen. Danach erschienen in den Neunzigern nur noch vereinzelte Artikel über das Thema.

Mit Aufkommen von XML wurde der Fokus immer mehr auf Online-Dokumentation gelegt. Damit befassten sich viele Artikel, in denen auch Software-Dokumentation nebenbei angesprochen wurde. Von dem her ist es nicht klar abzugrenzen, welcher Artikel sich eindeutig mit Software-Dokumentation befasst.

In jedem Fall wurde in den Jahren 2001 und 2004 vermehrt das Thema behandelt. In diesen Jahren wurde bei jeweils einer Ausgabe Software-Dokumentation

als Schwerpunktthema behandelt und dementsprechend erschienen viele Artikel darüber. In den letzten beiden Jahren ist ein Rückgang der Artikel zu verzeichnen, da hier Software-Dokumentation immer wieder in Artikeln über DITA, Single-Source-Publishing oder XML auftaucht, jedoch sehr wenige Artikel über Software-Dokumentation explizit veröffentlicht wurden.

Bei den tekcom-Vortragsthemen auf Jahrestagungen seit 2003 sieht es anders aus. Hier stand das Thema Software-Dokumentation so gut wie nie im Vordergrund, einzig bei der Regionalgruppe Rhein-Main wurde es 2007 als Schwerpunktthema behandelt. Die Regionalgruppe München bildete von 2001 bis 2006 eine Arbeitsgruppe, um eine Broschüre zur Gestaltung von Online-Informationen herauszugeben. XML hingegen wurde bei nahezu jeder Tagung und auch in jedem Heft der tekcom behandelt. DITA kommt seit dem Jahr 2006 sehr stark zur Geltung. Dieses Dokumentenformat passt sehr gut für diese Zwecke und steht daher oft im Zusammenhang mit Software-Dokumentation.

In Anbetracht dessen, dass durchschnittlich acht Artikel in einer tekcom-Zeitschrift abgedruckt sind, beträgt der Anteil der Artikel, die sich ausführlich mit dem Thema Software-Dokumentation befassen, seit dem Jahr 2000 rund 5%. Eine Übersicht der relevanten Artikel ist im Anhang zu finden.

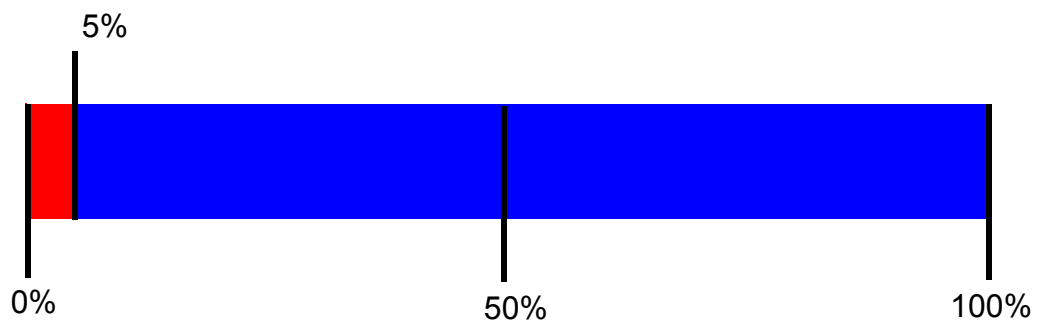


Abb. 5-1: Anteil der Artikel über Software-Dokumentation (roter Balken) ab dem Jahr 2000

## 6 Arten der Software-Dokumentation

Grundsätzlich wird Software-Dokumentation in zwei Bereiche eingeteilt, Anwenderdokumentation und Entwicklerdokumentation. Die Anwenderdokumentation richtet sich direkt an den Benutzer und liegt dem Softwareprodukt in einer Druckversion oder auf einem elektronischen Medium bei. Sie beschreibt alle relevanten Vorgänge zur vollständigen Nutzung des Produktes. Entwicklerdokumentation ist für die Erstellung und Wartung der Software bestimmt und wird von Fachleuten gelesen. Sie ist meist kürzer, verwendet Fachausdrücke und beschreibt spezielle Vorgänge, die nicht vom Benutzer ausgeführt werden können. Die Diplomarbeit konzentriert sich im Weiteren auf die Anwenderdokumentation.

Software-Dokumentation enthält alle Informationen, die ein Benutzer zur Bedienung eines Softwareproduktes benötigt. Diese werden in der Regel auf zwei verschiedene Arten bereitgestellt, als Printversion und Online-Dokumentation. Die Unterscheidung wird aufgrund des Präsentationsmedium getroffen. Abbildung 6-1 zeigt die verschiedenen Medientypen, die von Print- und Online-Dokumentation genutzt werden. (vgl. Grünwied 2007, S. 22)

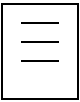
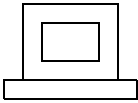
	Speicher-Medien	Präsentations-Medien	Wahrnehmungs-Medien	
<b>Print-Dokumentation</b> 	Papier CD	Papier Folie	Text Bild	Visuell
<b>Online-Dokumentation</b> 	Festplatte CD Server Internet	Bildschirm Display Lautsprecher	Text Bild Animation  Text Geräusch Melodie	Visuell  Auditiv

Abb. 6-1: Verschiedene Medien der Print- und Online-Dokumentation (nach Grünwied 2007)

Die Wahl des Speichermedium ist von der Notwendigkeit der Aktualisierung abhängig. Unterliegt die Dokumentation häufig Aktualisierungen, ist das Speichermedium Internet sinnvoll, da hier schnellere und kostengünstigere Aktualisierungen vorgenommen werden, als es bei gedruckten Dokumenten der Fall ist.

Das Präsentationsmedium ist für die Darstellung der Dokumentation zuständig. Dabei ist bei Online-Publikationen zu beachten, dass die Bildschirmgröße und Auflösung der Benutzer variieren. Auch auf barrierefreie Darstellung sollte geachtet werden.

Das Wahrnehmungsmedium wird größtenteils durch das Präsentationsmedium vorgegeben. Es bestimmt, wie Inhalte visuell oder auditiv aufgenommen werden.

## 6.1 Online-Dokumentation

Der Begriff Online-Dokumentation führt oft zu Missverständnissen, da „online“ genau genommen bedeutet, dass eine Online-Verbindung zum Internet besteht. Die korrekte Bezeichnung für diese Form der Darbietung wäre „onscreen“, für die Darstellung am Bildschirm. Der Begriff „Online-Dokumentation“ hat sich mittlerweile eingebürgert und wird in der Praxis häufig verwendet. (vgl. Thiemann 2001)

Wenn Online-Dokumentation in irgendeiner Form eingesetzt werden soll, müssen das Produkt, die Zielgruppe und die Umgebungsumstände berücksichtigt werden. Bei Software bietet sich immer Online-Dokumentation an.

Allgemein wird unter Online-Dokumentation jegliche Form der Dokumentation verstanden, die auf Computersystemen verfügbar gemacht wurde. Folgende Merkmale zeichnen nach Forst (1996) die Online-Dokumentation aus:

- große Mengen elektronisch verfügbarer Informationen
- verschiedene Zugangsmethoden, mit denen die Benutzer die Informationen finden und anzeigen
- ein Bildschirm als Ausgabemedium für die Informationen

Der Arbeitskreis für Online-Dokumentation innerhalb des Fachverbandes tekom erarbeitete zudem folgende Unterscheidungsmerkmale:

### **Produktkopplung**

Online-Dokumentation ist produktgekoppelt, wenn die Benutzer Zugang zu den Informationen haben, ohne ihr Arbeitsgebiet verlassen zu müssen.

Online-Dokumentation ist nicht produktgekoppelt, wenn die Informationen nicht im System/Produkt integriert sind. (Forst, 1996)

### **Kontextbezug**

Kontextbezug hat zwei Anwendungen, Kontextabhängigkeit und Kontextsensitivität. Kontextabhängige Programme zeigen im Fehlerfall sehr allgemeine Fehlermeldungen, kontextsensitive Programme zeigen im gleichen Fehlerfall eine genauere Fehlermeldung an, die die Angaben des Benutzers stärker berücksichtigt. (Forst, 1996)

### **Interaktivität**

In interaktiver Online-Dokumentation können die Benutzer aktiv über verschiedene Suchkriterien Informationen abrufen. Bei nicht interaktiver Online-Dokumentation werden Informationen vom System direkt auf dem Bildschirm angezeigt, ohne dass die Benutzer Einfluss nehmen können. (Forst, 1996)

Online-Dokumentation kann der Benutzer während der Softwarebedienung am Bildschirm lesen und wird deshalb als fensterorientierte Dokumentation bezeichnet. Online-Dokumentation sollte aber dennoch eine Funktion zum Drucken aufweisen. Die DIN EN ISO 13407 „Benutzerorientierte Gestaltung interaktiver Systeme“ legt die Vorgehensweise für die Entwicklung einer Online-Hilfe oder Software-Oberfläche fest.

Der Trend geht in Zukunft eindeutig in Richtung Online-Dokumentation. Dies ist besonders im Softwarebereich zu beobachten. Einige Softwarehersteller ersetzen das gedruckte Handbuch durch eine PDF-Datei und stellen diese entweder im Internet oder auf CD-ROM zur Verfügung. Software-Dokumentation ist

prädestiniert für Online-Dokumentation, da ohnehin auf dem Bildschirm gearbeitet wird.

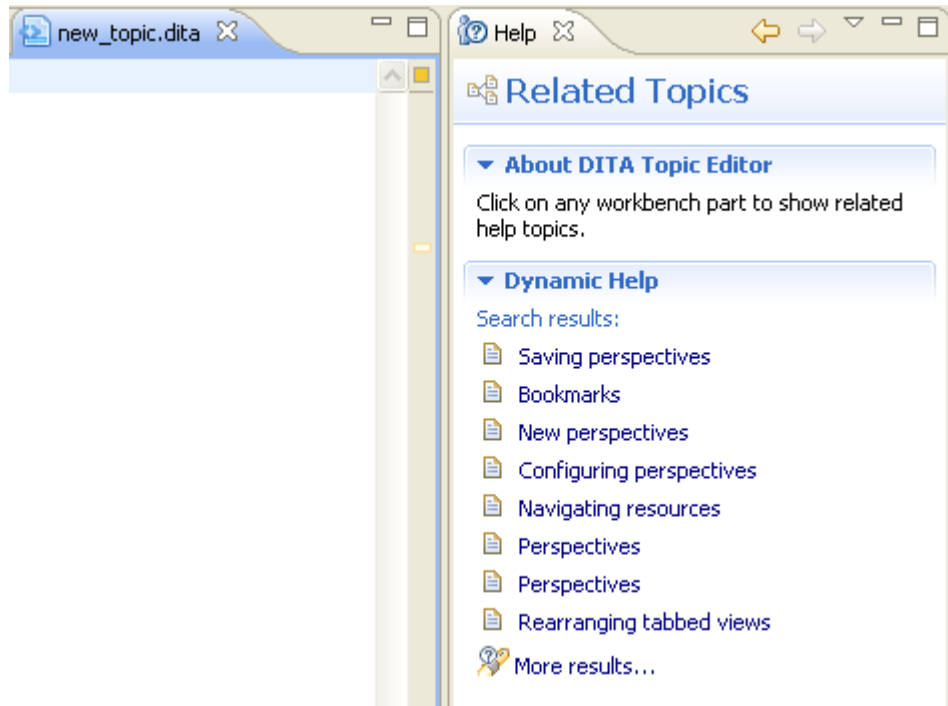


Abb. 6-2: Online-Hilfe der Software Eclipse, Kontextbezug zum aktuellen Projekt (new\_topic.dita)

Zur Online-Dokumentation gehören:

- Online-Hilfen
- elektronische Handbücher
- Dialoghilfe
- Tipps
- Direkthilfe
- Hilfeagenten
- Computer Based Training
- Lernprogramme

Die verschiedenen Arten der Online-Dokumentation werden im Folgenden näher beschrieben.

### 6.1.1 Online-Hilfen

Online-Hilfen sind in jedem Programm enthalten und mit der Taste F1 oder über die Menüleiste aufrufbar. Sie bestehen aus HTML- oder Textdateien, die mit einem Programm in ein Hilfeformat kompiliert wurden. Das gebräuchlichste Format ist hierbei HTML Help von Microsoft. Andere Formate wären z. B. JavaHelp, FlashHelp oder WebHelp.

Der Bildschirm ist bei der Online-Hilfe in drei Bereiche eingeteilt. Oben befindet sich die Symbolleiste mit verschiedenen Optionen. Links ist die Navigation innerhalb der Hilfe mittels einer Baumstruktur dargestellt. Diese ist aufklappbar und anklickbar. Darüber sind in Karteireitern Suchfunktionen angebracht. Den größten Bereich nimmt bei der Online-Hilfe der Themenbereich ein. Das ist der Anzeigebereich der kompilierten HTML- bzw. Textdatei. Im Text sind weiterhin Verlinkungen zu anderen Seiten möglich. Dadurch werden Online-Hilfen in der Regel nicht linear gelesen.

Die Online-Hilfe ist normalerweise kontextunabhängig und nicht produktgekoppelt. Sie sind jederzeit als separates Fenster aufrufbar.

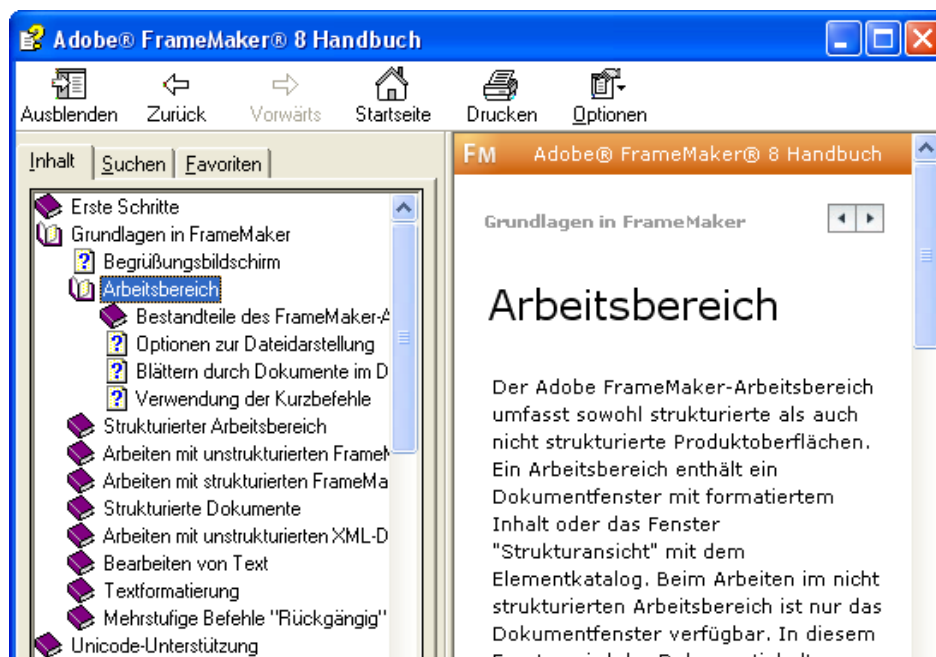


Abb. 6-3: Online-Hilfe der Software Adobe FrameMaker in Microsoft Help Viewer

## 6.1.2 Elektronische Handbücher

Elektronische Handbücher besitzen den gleichen Inhalt wie das eventuell vorhandene gedruckte Handbuch. Sie liegen meist im PDF-Format vor und können bei Bedarf jederzeit ausgedruckt werden. Elektronische Handbücher werden meist auf CD-ROM mit der Software mitgeliefert.

Viele Unternehmen stellen seit Verbreitung der CD-ROM und des Internets ihre Handbücher nur noch elektronisch als Acrobat PDF zur Verfügung. Für den Hersteller senkt diese Methode die Druck- und Distributionskosten. Das Ausdrucken der elektronischen Handbücher verlagert sich auf den Benutzer. (Grünwied 2007 S. 30)

Die Inhalte werden nicht mediengerecht aufbereitet, es ist lineares Lesen nötig. Jedoch sind meist Lesezeichen integriert, die eine schnelle Navigation und bessere Suchmöglichkeiten ermöglichen. Wie die Online-Hilfe ist auch das elektronische Handbuch nicht kontextabhängig und nicht produktgekoppelt.

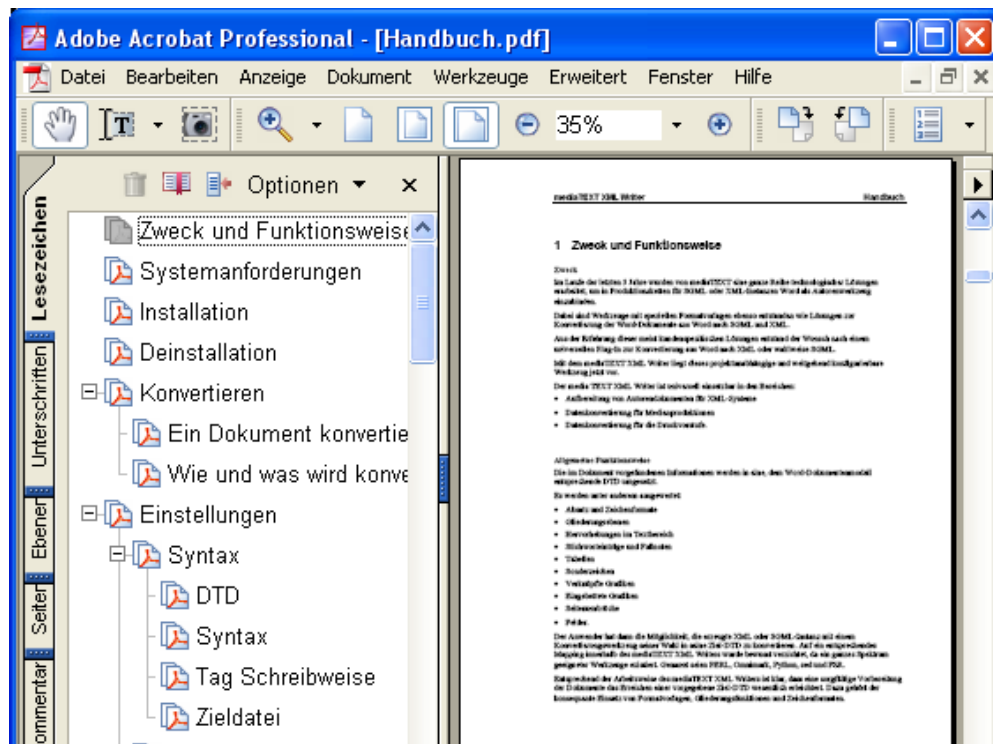


Abb. 6-4: Elektronisches Handbuch für die Software XML-Writer der Firma mediaTEXT

### 6.1.3 Dialoghilfe

Bei der Dialoghilfe handelt es sich um eine kontextsensitive Hilfeart. Sie ist im Dialogfenster zu finden und bietet dazu die jeweilige Hilfestellung. Um die Dialoghilfe zu aktivieren, ist eine Schaltfläche vorhanden. Diese öffnet die Online-Hilfe der Software und springt direkt zum relevanten Thema.

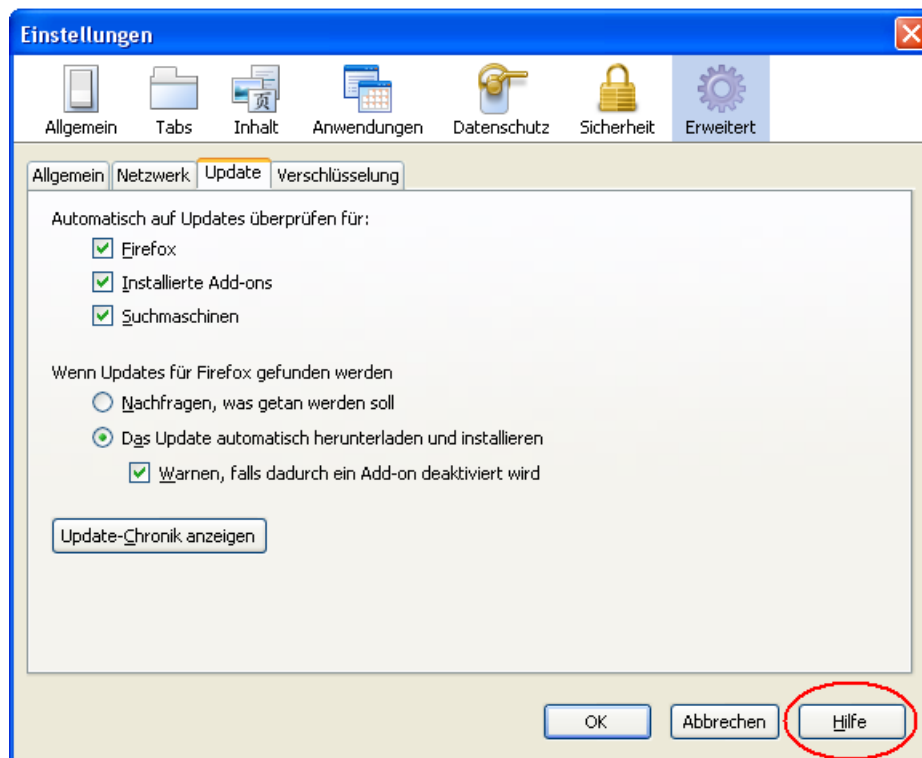


Abb. 6-5: Dialoghilfe im Dialogfenster „Einstellungen“ der Software „Mozilla Firefox 3.0“

### 6.1.4 Direkthilfe

Diese Hilfeart zählt zu den kontextintensivsten Online-Hilfen. Ähnlich wie der Tooltip, der beim Überfahren mit der Maus Informationen zu Objekten anzeigt, zeigt die Direkthilfe zu einem einzelnen Objekt (Menübefehl, Symbol usw.) den passenden Hilfetext an. Die Hilfe wird mit der Menüleiste oder der Tastenkombination Umschalttaste+F1-Taste (Windows) aktiviert. Ist sie angeschaltet, erscheint ein Fragezeichen neben den Cursor. Wird nun ein Objekt angeklickt, erscheint ein kurzer Hilfetext in einem Fenster.

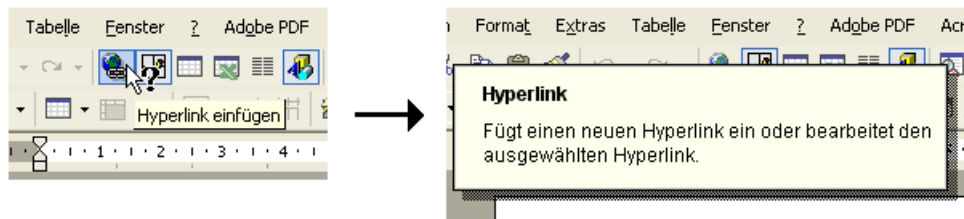


Abb. 6-6: aktivierte Direkthilfe (erkennbar am Fragezeichen) und angeklicktes Symbol (MS Word)

### 6.1.5 Hilfeagenten (Wizards)

Hilfeagenten sind dynamisch und kontextsensitiv. Sie erscheinen in Problemsituationen. Dafür ist es notwendig, das Benutzerverhalten zu analysieren und darauf zu reagieren. (vgl. Grünwied 2007, S. 43)

Hilfeagenten sind oft in Form einer Figur sichtbar. Dies soll einen beruhigenden und witzigen Effekt erzeugen, kann aber auch schnell störend wirken. Bekannt sind die Agenten vom Microsoft Office-Produkten. Ein Vorteil von Hilfeagenten ist die Möglichkeit, ganze Fragen zu formulieren und einzugeben. Daraufhin werden Antwortmöglichkeiten angeboten, die anklickbar sind und wiederum auf die Online-Hilfe verweisen. Der Hilfeagent kann über die Menüleiste ein- und ausgeblendet werden.

### 6.1.6 Tipps

Tipps sind nützliche Hinweise für Benutzer, die mit der Software bereits vertraut sind. Bekannt ist der „Tipp des Tages“, welcher sich beim Programmstart selbstständig öffnet. Tipps weisen auf besondere Funktionen hin und sollen den Arbeitsablauf beschleunigen. Benutzer können die Anzeige des Tipp-Fensters beim Programmstart deaktivieren oder den Tipp mittels der Menüleiste aufrufen. Der nächste Tipp kann durch Klicken einer Schaltfläche im Tipp-Fenster angezeigt werden. (vgl. Grünwied 2007, S. 46)

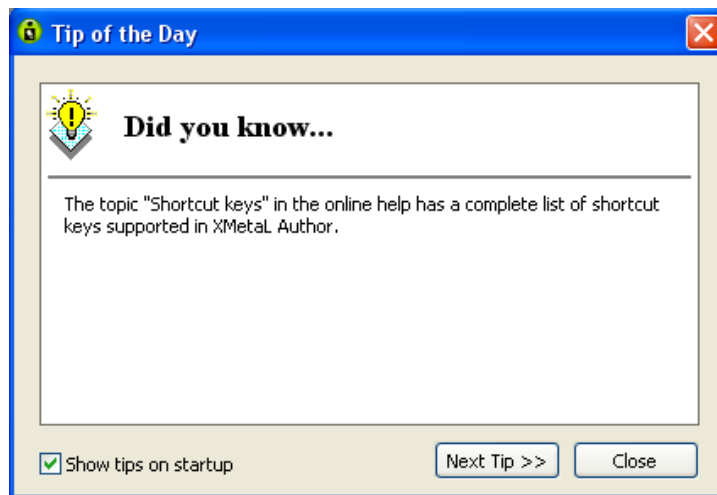


Abb. 6-7: Tipp nach dem Öffnen der Software „XMetaL Author“

### 6.1.7 Computer Based Training

Computer Based Training (Rechnerunterstützter Unterricht) bezeichnet eine Art des computerunterstützten Lernens. Die Lernenden durchschreiten Programme und damit verschiedene Lernschritte auf dem Computer (meistens über den Einsatz von CD-ROM). Bisher wird CBT vorwiegend zum Erlernen von Computer-Anwendungen, Sprachen, in Kinderlernprogrammen oder der betrieblichen Weiterbildung eingesetzt. (e-teaching 2006)

### 6.1.8 Lernprogramme

Bei Lernprogrammen unterscheidet man zwischen zwei Konzepten: Das erste Konzept unterstützt den Anwender beim Erlernen des eigentlichen Anwenderprogramms. Solche Lernprogramme werden mit dem Anwenderprogramm geliefert und zeigen Lösungswege an Hand konkreter Beispiele. Im zweiten Konzept steht die Weiterbildung im Vordergrund. Bei diesen Programmen geht es um Wissensvermittlung. Diese Lernprogramme werden für das Erlernen von Fremdsprachen, für die Vermittlung mathematischer, physikalischer, geographischer oder naturwissenschaftlicher Kenntnisse eingesetzt. (vgl. Datacom 2008a)

### **6.1.9 Auswahlkriterien**

Komplexe Software, die auch für gewerbliche Zwecke eingesetzt wird, sollte bereits vor der Installation beschrieben werden. Dazu ist eine Installationsanleitung im gedruckten Handbuch eine mögliche Lösung. Punktuelle Informationen, die während des Betriebes der Software benötigt werden, sollten mittels Online-Hilfen eingebunden werden. Ideal wäre die Bereitstellung beider Varianten. Papier für die Kerninformationen und Online-Hilfe für die Benutzerdokumentation. (vgl. Grünwied 2007, S. 47)

Bei der Auswahl der Online-Hilfeformate sind auch die Nutzergruppen zu berücksichtigen. Ist das Softwareprodukt für alle Nutzergruppen (Anfänger, Fortgeschrittenen und Experten) bestimmt, bedarf es mehr Aufwand für die Hilfestellungen. In diesem Fall sind auch mehrere Varianten gleichzeitig sinnvoll, inklusive der Implementierung eines Hilfeagenten.

Weiterhin sollte die Verteilung und Aktualisierung der Dokumentation berücksichtigt werden. Ist häufig eine Aktualisierung nötig, empfiehlt sich die Bereitsstellung der Dokumentation im Internet oder auf CD-ROM. Eine gedruckte Version wäre in diesem Fall sehr kostenintensiv. Möglich ist auch, schon von Entwicklerseite aus die Software mit einer automatischen Update-Funktion auszustatten.

## **6.2 Gedruckte Handbücher**

Print-Dokumentationen werden auf Papier gedruckt und dem Benutzer zur Verfügung gestellt. Trotz des Trends zur Online-Dokumentation im Softwarebereich gibt es Gründe, Informationen auf Papier zu drucken. Wenn z. B. das System nicht mehr benutzbar ist, kann auch nicht mehr auf die Online-Hilfe zurückgegriffen werden. Das gedruckte Handbuch kann in diesem Fall mit Problemlösungen Abhilfe schaffen. Zudem fordern die aktuelle Rechtsprechung sowie einschlägige Normen folgende Inhalte in gedruckter Form (nach Grünwied 2007, S. 27):

- Produkt- und Versionsangaben zur eindeutigen Bestimmung der Software
- Anforderungen an Hardware und Betriebssystem
- Installationsanleitungen
- Erste Schritte mit dem System
- Anleitungen zum Recovery und der Fehlerbehandlung
- Anleitungen zum Starten der Software
- Anleitungen zum Zugriff auf sämtliche Online-Dokumentationen
- Kontaktadressen zum Service

### **6.3 Vergleich Online-Print**

Zwischen Online-Dokumentation und gedruckter Version bestehen Unterschiede, aber es gibt auch Gemeinsamkeiten.

Ein wesentliches Merkmal bei den meisten Varianten der Online-Dokumentation ist die Kontextsensitivität (siehe Kapitel 6.3.1). Die Hilfe reagiert auf Tätigkeiten des Benutzers und spricht immer den aktuellen Bezug an. Dadurch sind direkte, punktuelle Hilfen möglich. Online-Dokumentation wird zudem parallel zum Programm verwendet und muss nicht linear gelesen werden.

Printprodukte oder auch Handbücher auf dem Bildschirm beinhalten die komplette Handlungsanweisung, ausgehend von der Installation. Diese können auch vor der eigentlichen Benutzung der Software zur Hand genommen werden. Vorteile von Druckprodukten sind die gute Lesbarkeit und die Möglichkeit, Randnotizen und Anstreichungen im Text anzubringen. Druckprodukte werden in der Regel linear gelesen.

Befragungen des Technischen Leiters der Firma CARSTENS + PARTNER GmbH, München, zu diesem Thema ergaben, dass in der Praxis oft beide Varianten zur Verfügung gestellt werden. Dabei wird bei der Online-Variante auf Kontextsensitivität geachtet und zwischen den Textkapiteln verlinkt. Verschiedene Bildschirmmasken und Felder werden nur in der Online-Hilfe beschrieben.

Screenshots werden hingegen nur in der Druckvariante eingebunden. Es wird zuerst die Printversion, dann die Online-Hilfe erzeugt. (Siegel 2008a) Inhalte des Interviews sind im Anhang zu finden.

Die Frage ist nun, welches Hilfeformat zu welchem Zweck sich eignet. Beide Varianten haben unterschiedliche Eigenschaften. Achtelig (2002) hat diese aufgelistet und bewertet.

<b>Eigenschaft</b>	<b>Print</b>	<b>Online</b>
Verfügbarkeit	eingeschränkt <ul style="list-style-type: none"> <li>• nicht an Produkt gekoppelt</li> <li>• mehr Nutzer als Handbücher</li> </ul>	uneingeschränkt <ul style="list-style-type: none"> <li>• mit dem Produkt verfügbar</li> </ul>
Aktualität	schwer aktualisierbar <ul style="list-style-type: none"> <li>• Ergänzungsbände</li> <li>• Loseblattsammlungen</li> </ul>	einfach und automatisch aktualisierbar <ul style="list-style-type: none"> <li>• Web-Aktualisierung</li> </ul>
Zugriff	eher ungezielt <ul style="list-style-type: none"> <li>• Inhaltsverzeichnis oder Index</li> <li>• Blättern erforderlich</li> </ul>	gezielt <ul style="list-style-type: none"> <li>• kontextsensitiv</li> <li>• Volltextsuche</li> </ul>
Orientierung	gut <ul style="list-style-type: none"> <li>• feste Seitenfolge</li> </ul>	eingeschränkt <ul style="list-style-type: none"> <li>• isolierter Blick auf einzelne Topics</li> <li>• keine feste Reihenfolge der Topics</li> </ul>
Navigation	schwierig <ul style="list-style-type: none"> <li>• Folgen von Querverweisen erfordert Blättern</li> </ul>	einfach <ul style="list-style-type: none"> <li>• Anklicken von Links führt zum Ziel</li> </ul>
Lesbarkeit	gut <ul style="list-style-type: none"> <li>• flimmerfrei</li> </ul>	eingeschränkt <ul style="list-style-type: none"> <li>• ungewohnt, anstrengend</li> <li>• Scrollen bei längeren Texten</li> </ul>
besondere Möglichkeiten	<ul style="list-style-type: none"> <li>• handschriftlich Notizen</li> <li>• Anstreichungen</li> </ul>	<ul style="list-style-type: none"> <li>• Animationen</li> <li>• Interaktion</li> <li>• Zugriffsschutz</li> </ul>

Abb. 6-8: Optimale Medien für bestimmte Zwecke (nach Achtelig 04/2002)

Ein gedrucktes Handbuch und eine Online-Hilfe können sich gut ergänzen. Das Handbuch ist sehr gut zur Einarbeitung an der Software geeignet, gerade bei wenig erfahrenen Benutzern. Dadurch beschränkt sich die Zahl der geöffneten Fenster am Bildschirm. Online-Hilfen dienen mehr den schon mit dem System vertrauten Anwender eine gezielte und schnelle Recherche. Ideal wäre also das Herstellen beider Formate. (Achtelig, 2002)

Je nach Situation ist entweder eine Online- oder eine Printversion besser geeignet. Achtelig (2002) hat in seinem Artikel diese Situationen betrachtet und aufgelistet.

Situation	Optimales Medium
Produkt in Betrieb nehmen	Print
In das Produkt und seine Bedienung einarbeiten	Print
Konkrete Fragen während der Arbeit lösen	Online
Technische Daten nachschlagen	Online
Der Anwender ist mit dem Produkt unterwegs	Online
Handbücher stehen im Schrank des Kollegen	Online
Produkt ohne Software-Oberfläche	Print
Produkt mit Internetzugang	Online
Modulares Produkt	Online
Geringes Vorwissen	Print
Schulung zum Produkt	Online
Seltene Nutzung mit wenig Zeit	Online
Häufige Nutzung mit mehr Zeit	Print
Eher technisch orientierte Anwender	Online
Eher technisch unversierte Anwender mit wenig PC-Erfahrung	Print

Abb. 6-9: Vergleich von Print und Online (nach Achtelig 2002)

### 6.3.1 Kontextsensitivität

„Kontext“ ist eine Zustandsmenge. Zustände bestimmen den aktuell benutzten Zeichenmodus, das Quellmaterial oder das Zeichenziel. Diese Zustände werden von der Anwendung ständig kontrolliert. (vgl. Neuwirt/Forster 2007)

Kontextsensitivität bedeutet, das Kontextmenü oder der Hilfetext ist abhängig von der jeweiligen Bearbeitungssituation. Das einfachste Kontextmenü wird bei Betätigung der rechten Maustaste eingeblendet. Es gibt die an der jeweiligen Stelle möglichen Optionen vor.

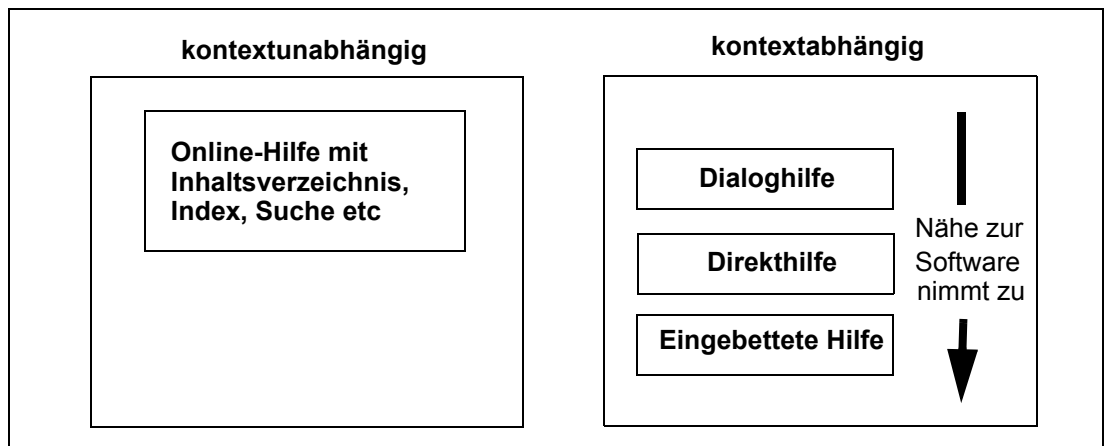


Abb. 6-10: Übersicht Kontextsensitivität (nach Grünwied, 2008)

Hervorragend wird kontextsensitive Hilfe im Programm Visual Studio 2008 umgesetzt. Beim Erstellen von Quelltexten wird, bei der Eingabe des ersten Buchstaben, ein Dialogfenster mit den an dieser Stelle möglichen und gültigen Listeneinträgen mit dem eingegebenen Buchstaben eingeblendet. Die Einträge werden durch einen Tooltip noch näher beschrieben. Die folgende Abbildung zeigt das Hilfsmittel „IntelliSense“ von Microsoft, das in den Produkten dieser Firma eingesetzt wird.

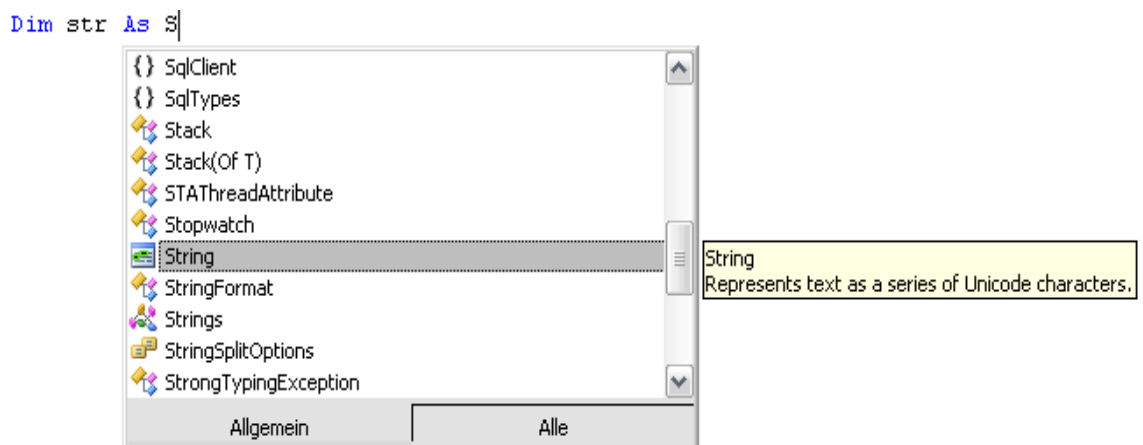


Abb. 6-11: Kontextsensitive Hilfe „IntelliSense“ (Programm Visual Studio 2008)

## 7 Dokumentenformate in der Software-Dokumentation

Zur Erstellung von Online-Hilfen existieren standardisierte Dokumentenformate auf XML-Basis, die zum Strukturieren der Informationen und für die Ausgabe benötigt werden.

### 7.1 DocBook

DocBook ist ein verbreitetes Dokumentenformat auf XML-Basis, das speziell für die Technische Dokumentation entwickelt wurde. Es ist standardisiert und ohne Lizenzkosten verwendbar. DocBook existiert bereits seit 1991 und wurde ursprünglich für SGML entwickelt. Es existiert mittlerweile auch eine XML-Version. (vgl. Walsh/Muellner 1999)

Seit 2002 ist eine vereinfachte Version, das sogenannte „Simplified DocBook“ verfügbar. Mit dieser Version sind nur einzelne Dokumente, also Artikel, erstellbar.

DocBook erzeugt verschiedene Ausgabeformate und ist von daher auch für Single-Source-Publishing geeignet. DocBook ist eher für größere Projekte und Bücher vorgesehen. Es ist nicht topicorientiert, sondern strukturiert das Dokument in Kapiteln. DocBook geht von einem hierarchischen Aufbau der Software-Dokumentation aus und lehnt sich in der Struktur eng an den Aufbau von traditionellen Büchern an. Eine Dokumentation auf Basis von DocBook besteht daher aus einer Menge von Büchern, die wiederum selbst wieder in Kapitel und Unterkapitel unterteilt sind.

DocBook unterscheidet folgende grundlegende Dokumentenklassen:

- article  
Ein article ist geeignet für Artikel, Technische Notizen, FAQs aber auch Zeitschriftenartikel.

- **book**  
Book wird für größere Dokumente (Bücher, Bedienungsanleitungen) verwendet.
- **set**  
Die Zusammenfassung und Sammlung mehrerer Dokumente kann mit dem Element set vollzogen werden.

Verschiedene Editoren verarbeiten DocBook und bieten beim Erstellen spezielle Vorlagen an. Ein Beispiel für ein Buch sieht folgendermaßen aus:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
"http://www.oasis-open.org/docbook/xml/4.4/docbookx.dtd">
<book lang="de">
<bookinfo>
<title>Title des Buches</title>
<author><firstname>Vorname des Autors</firstname><surname>Nachname
des Autors</surname></author>
<publisher><publishername>Verlagname</publishername></publisher>
<isbn>ISBN</isbn><copyright><year>Jahr</year><holder>Name</holder></
copyright></bookinfo>
<preface>
<title>Vorwort</title>
<para>Inhalt des Vorworts</para>
</preface>
<part>
<title>Erstes Kapitel</title>
<chapter>
<title>Kapiteltitel</title>
<section>
<title>Erster Abschnitt</title>
<para>Erster Absatz</para>
<para>Zweiter Absatz</para><para>...</para>
</section>
</chapter></part>
<appendix>
<title>Anhang</title>
<para>Inhalt des Absatzes</para>
</appendix>
<index>
<title>Index</title>
<indexentry><primaryie>Text, Seitenzahl</primaryie></indexentry>
</index>
</book>
```

Beispiel: 7-1: Docbook-Quelltextbeispiel

## 7.2 DITA

DITA ist eine Informationsarchitektur auf XML-Basis, die speziell für die Technische Dokumentation konzipiert wurde. Der Topic-orientierte Ansatz und die flexiblen Spezialisierungsmöglichkeiten sind Merkmale für diese Technologie. (vgl. Closs 2007, S. 153)

Topics sind in sich geschlossene, möglichst kontextunabhängige Inhaltsblöcke. Die Topic-Strukturierung eignet sich gegenüber einer starren Kapitelstruktur, wie DocBook sie vorschreibt, besser zur Wiederverwendung, vorausgesetzt die Topic-Einteilung wird richtig gemacht. Weiterhin bietet DITA die Möglichkeit, auf Basis der vorhandenen neue Domänen und Grundtypen zu definieren. Dies bezeichnet man als Spezialisierung. (vgl. Closs 2007, S. 163)

DITA wurde 2005 in der Version DITA 1.0 als OASIS-Standard verabschiedet. OASIS (Organization for the Advancement of Structured Information Standards) ist eine nicht gewinnorientierte Organisation, die sich mit der Verwaltung von E-Business-Standards beschäftigt. Die Nachfolgeversion DITA 1.1 wurde 2007 standardisiert. Von daher gilt DITA als sehr modernes und noch nicht weit verarbeitetes Dokumentenformat.

### 7.2.1 Definition

Der Name DITA steht für Darwin Information Typing Architecture. Dabei steht

- „Darwin“ für Anpassung und Vererbung
- „Information Typing“ für die Klassifizierung und Einteilung der Topics in Topic-Klassen
- „Architecture“ für die Definition und Vorgabe von Informationsgebilden

Für den DITA-Standard sind Sprachdefinitionen in Form von DTDs und Schemas, die Dokumentation und Beispiele kostenlos im Internet erhältlich. Weiterhin ist das ebenfalls kostenlose DITA Open Toolkit downloadbar. Dieses enthält eine

Vielzahl von XSL-Transformationen und Skripte, die aus den DITA-Dateien verschiedene Ausgabeformate wie HTML oder PDF erzeugen. Diese Skripte können je nach Bedarf verändert oder modifiziert werden. (vgl. Closs 2007, S. 153)

## 7.2.2 Topic-Grundtypen

In der DITA-Sprache gibt es folgende Topic-Typen:

- **generisches Topic**

Zur Darstellung allgemeiner Informationen. Der Inhalt besteht dabei nur aus dem body-Element, das andere Elemente umfasst. Die Elemente für den generischen Topic-Typ gelten auch für die anderen Topic-Typen, sofern für diese keine spezifischen Elemente vorhanden sind.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD DITA Topic//EN" "topic.dtd">
<topic id="Demo" xml:lang="de-DE">
<title>Einführung</title>
<prolog>
<author>TS</author>
</prolog>
<body>
<p>Hier kann beispielsweise eine kurze Einführung oder Produktbeschreibung stehen.</p></body>
</topic>
```

- **Concept**

Zur Beschreibung von Hintergrund- und Überblicksinformationen.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="test" xml:lang="de-DE">
<title>Informationen zum Produkt</title>
<prolog>
<author>TS</author>
<metadata>
<keywords>
<keyword>Hardware</keyword>
<keyword>Maschine</keyword>
<keyword>Gerät</keyword>
</keywords>
```

```

</metadata>
</prolog>
<conbody><p>Das Produkt ist die neueste Erfindung auf dem Markt.</p>
</conbody>
</concept>

```

Beispiel: 7-2: DITA-Concept

### • Reference

Zur Beschreibung stark strukturierter Inhalte, wie z. B. Kommandobeschreibungen oder Besonderheiten des Produktes.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE reference PUBLIC "-//OASIS//DTD DITA Reference//EN" "reference.dtd" >
<reference id="features" xml:lang="de-DE">
<title>Features</title>
<prolog>
<author>TS</author>
<metadata>
<keywords>
<keyword>Features</keyword>
<keyword>Neuigkeiten</keyword>
<keyword>Besonderheiten</keyword>
</keywords>
</metadata>
</prolog>
<refbody>
<properties>
<property>
<proptype>Feature 1</proptype>
<propvalue>Beschreibung</propvalue>
</property>
<property>
<proptype>Feature 2</proptype>
<propvalue>Beschreibung</propvalue>
</property>
<property>
<proptype>Feature 3</proptype>
<propvalue>Beschreibung</propvalue>
</property>
</properties>
</refbody>
</reference>

```

Beispiel: 7-3: DITA-Referenz

- **Task**

Zur Beschreibung von Schritt-für-Schritt-Anleitungen.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN" "task.dtd" >
<task id="anweisung" xml:lang="de-DE">
<title>Handlungsanweisung</title>
<prolog>
<author>TS</author>
<metadata>
<keywords>
<keyword>Schritte</keyword>
<keyword>Anweisung</keyword>
<keyword>Tätigkeiten</keyword>
</keywords>
</metadata>
</prolog>
<taskbody>
<context>Das sollten Sie tun.</context>
<steps>
<step><cmd>Schritt...</cmd></step>
<step><cmd>Schritt...</cmd></step>
<step><cmd>Schritt...</cmd></step>
</steps>
<result>Sie haben Ihr Ziel erreicht!</result>
</taskbody>
</task>
```

Beispiel: 7-4: DITA-Task

### 7.2.3 DITA-Map

Eine Map ermöglicht, die verschiedene Topics miteinander zu kombinieren. Sie ähnelt einem Inhaltsverzeichnis und stellt ein Gerüst für ein komplettes Informationsgebilde dar. Die DITA-Map legt fest, welche Topics zum Inhaltsgebilde gehören. Eine Ausgabe mittels DITA Open Toolkit wird immer über die DITA-MAP erzeugt, das heißt, die in der Map eingefügten Topics werden in andere Ausgabeformate umgewandelt.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "map.dtd" >
<map xml:lang="de-DE">
<title>Inhalt</title>
<topicmeta>
```

```
<author>TS</author>
</topicmeta>
<topicref navtitle="Einführung" href="diplom_concept.xml" type="concept"/>
<topicref navtitle="Anweisung" href="diplom_task.xml" type="task"/>
<topicref navtitle="Besonderheiten" href="diplom_reference.xml" type="reference"/>
</map>
```

Beispiel: 7-5: DITA-Map

## 7.2.4 Programme zur Bearbeitung von DITA

Programme, die in der Technischen Dokumentation typischerweise zum Einsatz kommen, bieten in den neuesten Versionen auch die Verarbeitung von DITA an. FrameMaker hat seit der Version 7.2 eine DITA-Unterstützung. XMeta1, XMLSpy und Eclipse erlauben mittlerweile auch, DITA zu editieren. Generell muss aber keine DITA-Unterstützung vom Programm vorliegen, zur Erstellung von DITA-Dateien genügt ein einfacher Texteditor.

## 7.2.5 DITA Open Toolkit

Es gibt für die Weiterverarbeitung von DITA-Dateien eine frei erhältliche Zusammenstellung von Tools, DTDs und XSL-Dateien, das DITA Open Toolkit von OASIS. Dieses ist besonders für den Einstieg gut geeignet. Die XSL-Vorlagen können aber auch nach Belieben verändert und individuell angepasst werden. Mit Hilfe des DITA Open Toolkits können HTML, XHTML, PDF, HTMLHelp, Java-Help, EclipseHelp, DocBook, TROFF und RTF-Dateien erstellt werden. Deshalb ist es ein gutes Werkzeug für das Single-Source-Publishing.

Im ganzen Paket sind verschiedene Prozessoren enthalten (u. a. ANT, Xerces, FOP). Weiterhin kann auch der XSLT-Formatter von Antenna House durch Einbringen der von Antenna House herausgegebenen Stylesheets eingebunden werden. Dabei werden die vorhandenen Stylesheets einfach durch die heruntergeladenen XSL-Dateien ersetzt und einige zusätzliche Dateien eingebunden.

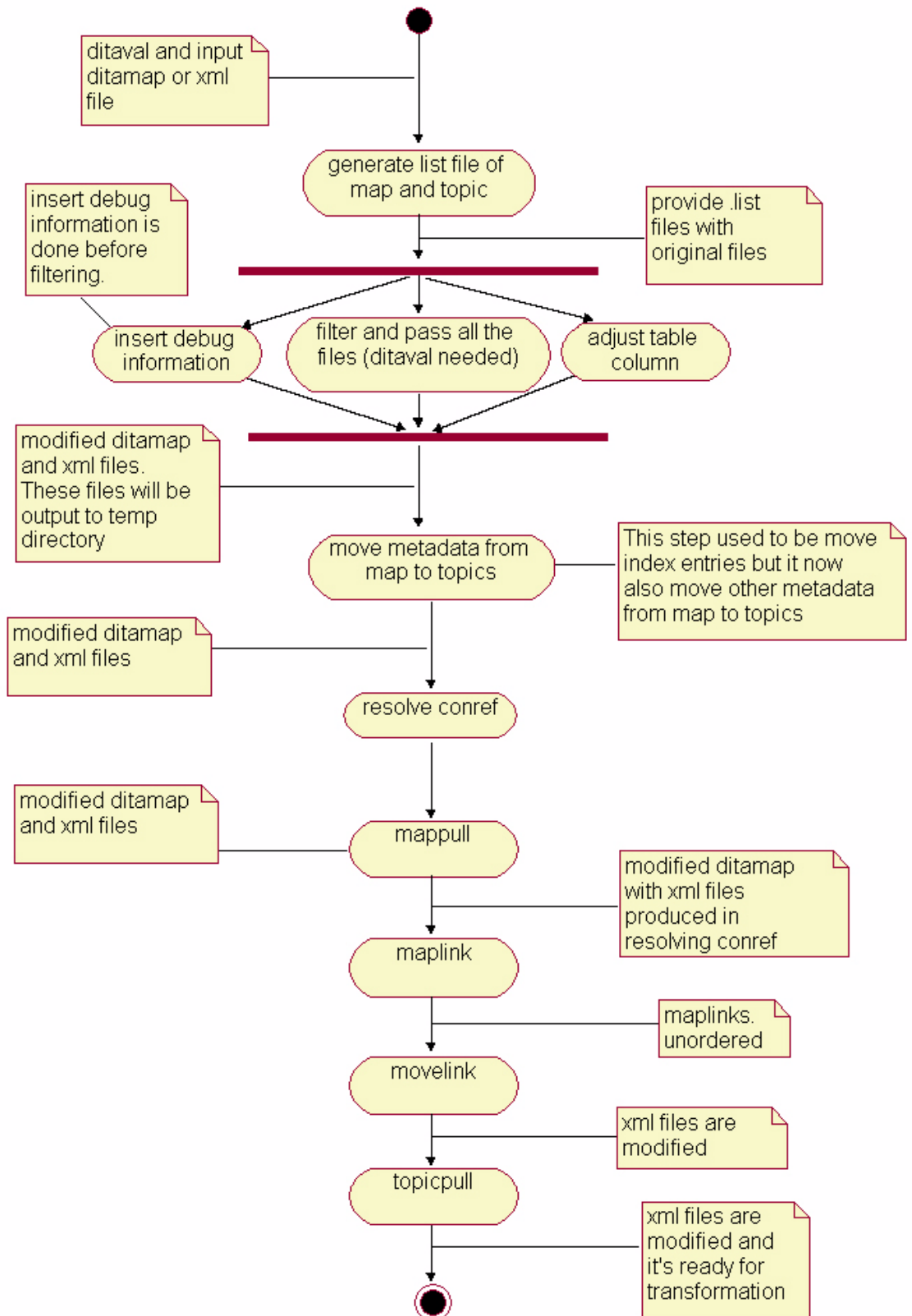


Abb. 7-1: Preprocess des DITA Open Toolkit (Quelle: <http://dita-ot.sourceforge.net>)

Das DITA Open Toolkit kann kostenlos heruntergeladen werden. Aktuell liegt es in Version 1.4.2.1 vor, diese Version ist auch für die Diplomarbeit verwendet worden. Es lässt sich sowohl an die Eclipse-Plattform, als auch an FrameMaker anbinden. Erst dadurch können verschiedene Ausgabeformate erzeugt werden. Das Toolkit kann aber auch ohne Anbindung benutzt werden. DITA-Dateien können in jedem Editor erstellt werden und mittels dem Open Toolkit transformiert werden. Dazu ist ein Aufruf des Prozessors über die Kommandozeile nötig.

### **7.3 Information Mapping**

Wie schon im Kapitel 2.3.3 beschrieben ist die IMAP-Methode eine gute Möglichkeit der Strukturierung und Einteilung von Texten. Information Mapping basiert jedoch nicht auf einem XML-Hintergrund und kann in der Single-Source-Dokumentation nur in Verbindung mit anderer Technologien angewendet werden.

Sehr gut lassen sich IMAP und DITA kombinieren. Beide Methoden weisen verwandte Strukturierungsmethoden auf. Der Vorteil beider Techniken ist die Erstellung modularer Information (siehe Kapitel 2.3.5). Information Mapping gliedert die Informationen in Blöcken, die wiederum in Maps zusammengefasst werden. IMAP-Maps sind jedoch nicht mit einer DITA-Map gleichzustellen. (vgl. Siegel 2008)

Information Mapping wird in einem Topic umgesetzt. Dabei ist ein Topic mit einer IMAP-Map vergleichbar. IMAP-Blöcke werden als Task oder Concept eingefügt. Siegel (2008) hat folgende Liste von IMAP-DITA-Entsprechungen in seinem Artikel im Heft „Technische Kommunikation“, Ausgabe 03/08, aufgestellt:

IMAP	DITA-Elemente
Anleitung	Task + <i>Steps</i>
Prozess	Concept + <i>table(Stufen)</i>
Struktur (Grafik)	Concept + <i>fig(img + ol)</i>
Begriff	Concept + <i>p (paragraph)</i>
Prinzip	Concept + <i>note (z. B. Danger)</i>
Fakt	Concept + <i>table oder ul</i>
Klassifikation	Concept + <i>table oder ul</i>

Abb. 7-2: IMAP-Elemente und entsprechende DITA-Elemente

## 7.4 Formatierungs- und Transformationssprachen

Wie schon im Kapitel 2.1.3 beschrieben, gibt es für die Ausgabe von XML-Dateien in andere Formate spezielle Technologien.

### 7.4.1 XSLT

XSLT ist eine Transformationssprache für XML-Dokumente. Mit ihr können Inhalte von XML-Elementen in andere Dateiformate übertragen werden. XSLT wird in der Praxis häufig zum Erstellen von HTML-Seiten für Online-Hilfen verwendet.

Die Ansprache der Elemente erfolgt dabei häufig über die Pfadbeschreibungssprache XPath. Diese dient der Navigation innerhalb von Dokumentenbäumen. Zur Transformation ist ein XSLT-Prozessor nötig. Mit diesem kann aus der XML-Datei und dem XSLT-Stylesheet beispielsweise ein HTML-Dokument erzeugt werden.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html"/>
<xsl:template match="/">
```

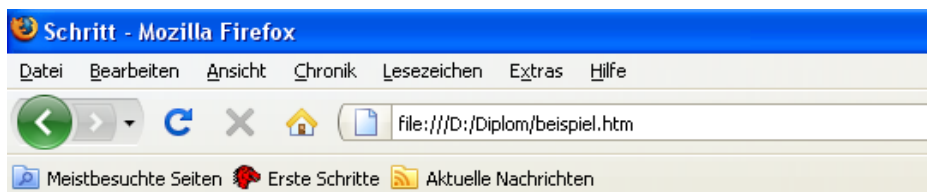
```

<html>
<head>
<title>Schritt</title>
<style type="text/css">
body
{font-family:arial;}
h1
{font-size:12pt;
font-weight:bold;}
</style>
</head>
<body>
<xsl:for-each select="Handlungsanweisung/Schritt">
<h1><xsl:value-of select="Aufgabe"/></h1>
<ol><xsl:for-each select="Handlung">
<li><xsl:value-of select="."/></li>
</xsl:for-each></ol>
<i><xsl:value-of select="Resultat"/></i>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Beispiel: 7-6: XSLT-Stylesheet

Transformiert man dieses XSLT-Stylesheet mit der XML-Datei aus Beispiel 2-1, entsteht folgende Ausgabe:



### Neues Absatzformat gestalten

1. Wählen Sie Befehle > Neues Format.
2. Stellen Sie im Absatzgestaltungskatalog die gewünschten Formate ein.

*Das neue Absatzformat erscheint im Absatzformatkatalog.*

Abb. 7-3: Ausgabe der Transformation im Browser Mozilla Firefox 3

## 7.4.2 XSL-FO

XSL-FO steht für Extensible Stylesheet Language – Formatting Objects und ist eine XML-basierte Formatierungssprache für die Ausgabe von XML-Dateien. XSL-FO ist eine W3C-Recommendation, die Version 1.0 wurde im Jahr 2001, die Version 1.1 im Jahr 2006 verabschiedet. XSL-FO kann auch in Verbindung mit XSLT angewendet werden.

Der Einsatzbereich für XSL-FO liegt ausschließlich in der automatisierten Aufbereitung und Verarbeitung von XML-Dokumenten. XSL-FO stellt die dafür benötigten Stylesheets zur Verfügung. Die Formatierung beruht auf der XML-Struktur der Eingangsdokumente und wird in allgemein gültigen Regeln im Stylesheet eindeutig formuliert. (vgl. Krüger/Welsch 2007, S. 11)

Die Verarbeitung von XML-Daten mit XSL-FO ist ein mehrstufiger Prozess ohne Eingriffsmöglichkeit innerhalb der Prozessschritte und ohne Schleifenbildung. Die XML-Datei und das XSL-FO-Stylesheet werden vom Prozessor eingelesen. Dieser bildet aus beiden Dateien ein XSL-FO-Dokument, welches wiederum von einem FO-Formatierer verarbeitet wird und meist ein PDF-Dokumentenformat ausgibt. (vgl. Krüger/Welsch 2007, S. 15)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="A4">
      <fo:region-body />
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="A4">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hier steht der Text</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

Beispiel: 7-7: XSL-FO Stylesheet

## **8 Erstellungsprogramme für Software-Dokumentation**

Für die Erstellung der Dokumentation, besonders von Online-Hilfen, stehen einige Programme zur Verfügung. Die wichtigsten werden im Folgenden näher erläutert.

### **8.1 Adobe FrameMaker**

Adobe FrameMaker ist ein professionelles Authoring- und Desktop-Publishing-System. Seit der Version 7.0 unterstützt FrameMaker XML, seit der Version 7.2 kann auch DITA verarbeitet werden. Die aktuelle Version ist FrameMaker 8.

Eine wichtige Funktion des Programmes ist die Bucherzeugung. Dadurch können verschiedene einzelne Dateien, die auch im XML-Format vorliegen können, zu einem Buch zusammengefasst werden und im PDF-Format gespeichert werden. FrameMaker kann dabei auch Inhaltsverzeichnisse und andere Verzeichnisse automatisch generieren.

FrameMaker ist beliebt in der Technischen Dokumentation. Sowohl die Firma mediaTEXT als auch CARSTENS + PARTNER GmbH, München, arbeitet mit der Software. FrameMaker ist für Online-Handbücher oder für Druckwerke gedacht. Es können keine kontextsensitiven Online-Hilfen damit erstellt werden.

### **8.2 Microsoft HTML Help**

Microsoft HTML Help ist ein kostenloses Programm zur Erstellung von HTML-basierten Hilfeformaten. Das Programm erzeugt Ausgabedateien im Microsoftspezifischen chm-Format. Diese Datei wird dann mit dem in Windows vorhandenen HTML-Help-Viewer angezeigt. (vgl. Kapitel 6.1.1)

Microsoft HTML Help stellt zur Navigation verschiedene Funktionen zur Verfügung, z. B. Inhaltsverzeichnis, Index oder Volltextsuche. Das Programm wird meist zur Erstellung von Online-Hilfen, Internet- und Intranetseiten verwendet. (vgl. Grünwied 2007, S. 81) Dazu werden vorher angefertigte HTML-Seiten im Programm kompiliert. Die HTML-Dateien müssen vorher in einem anderen Programm erstellt werden.

Da das Programm von Microsoft entwickelt wurde, ist zur Anzeige der kompilierten HTML-Dateien der Internet Explorer integriert. Verschiedene Hersteller haben mittlerweile spezielle Help-Viewer entwickelt, die CHM-Dateien auch unter Linux und Unix anzeigen.

Das Programm, das durch die neue Vista-Hilfe nicht mehr weiterentwickelt wird, ist in der Praxis weit verbreitet. Auch das DITA Open Toolkit erzeugt hhc- und hhp-Dateien, die in HTML Help weiterverarbeitet können.

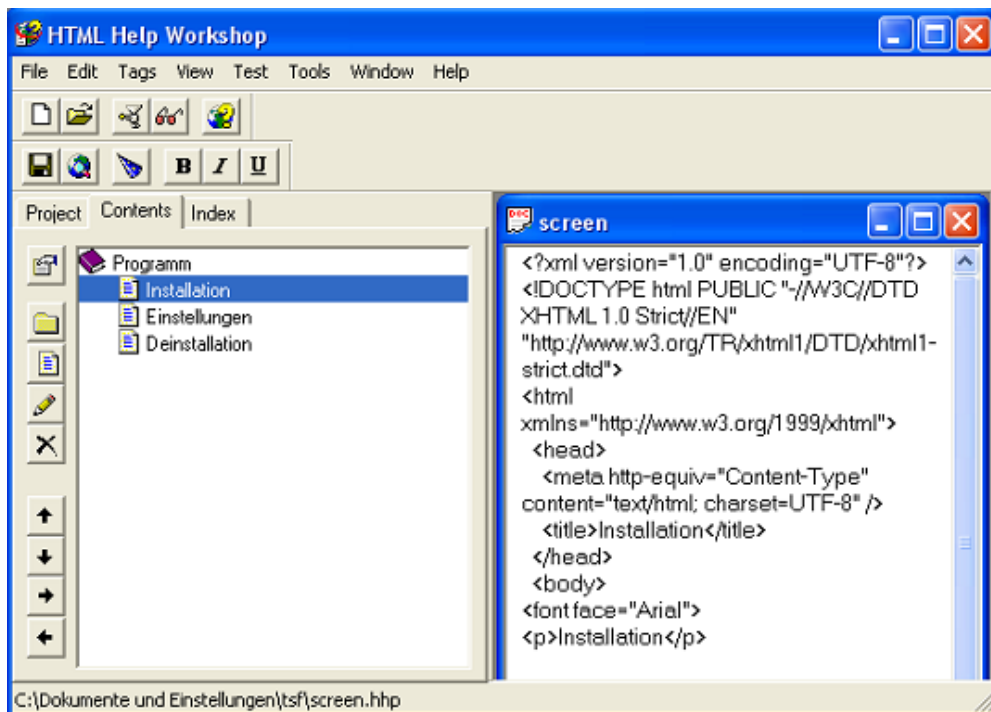


Abb. 8-1: Bedienoberfläche HTML Help Workshop

## 8.3 Microsoft Vista AP Help

Mit dem neuem Betriebssystem Microsoft Vista kam 2007 auch ein neues Hilfeformat auf dem Markt. Früher mit dem Codenamen „Longhorn“ bezeichnet, ist es nun in Vista Assistance Platform (AP) Help oder einfach Vista-Help umbenannt worden.

Dieses Hilfesystem ist XML-basiert, Inhalt und Layout sind streng getrennt. Die Hilfeinhalte werden im XML-Format erstellt. Ferner unterstützt es einen ansteigenden Pfad der Benutzerhilfe. Dieser beginnt mit einer gut gestalteten Oberfläche und führt über die Direkthilfe, das Hilfe-Center und die Benutzer-Community bis zum Endpunkt, dem Produkt-Support-Center. (vgl. Kulle 2006)



Abb. 8-2: Hilfefpad in der Vista AP Hilfe (Quelle: [www.help-info.de](http://www.help-info.de))

Zum Editieren und Entwickeln der Vista AP Hilfe hat Microsoft eine spezielle Markup-Sprache auf XML-Basis herausgegeben, Microsoft Assistance Markup Language (MAML). Mit MAML wird der Fokus auf Semantik liegen, beim Vorgängerformat HTML Help lag dieser noch auf Präsentation. (vgl. Kulle 2006)

Eine weiteres Feature dieser Hilfe ist der „Aktive Inhalt“. Dieser zeigt sich durch Shortcuts, Kontextsensibilität und den Active Content Wizard. Der Aktive Inhalt erlaubt mehr Interaktivität und ist mehr auf die Bedürfnisse der Anwender eingestellt. (vgl. Schicht/Grünwied 2006)

Die ursprüngliche Planung, die Version 2.0 kurz nach Auslieferung des Vista Betriebssystems herauszugeben, wurde von Microsoft wieder verworfen. Es ist zum

heutigen Zeitpunkt unklar, wann die zweite Version, die auch einen Help-Workshop und einen Hilfecompiler beinhalten soll, freigegeben wird. Ob AP Help mit Microsoft AML öffentlich verwendbar wird, ist ebenfalls unsicher. (vgl. Kulle 2006)



Abb. 8-3: Microsoft Vista AP Help

Herkömmliche Online-Hilfen im CHM-Format werden auch unter Windows Vista mit dem Help-Viewer geöffnet.

### JavaHelp

Java Help ist ein Hilfesystem von Sun Microsystems, das in der Programmiersprache Java entwickelt wurde. Es ist plattform- und browserunabhängig. JavaHelp benötigt das Java Runtime Environment. Die produzierte Hilfe kann in einer einzigen JAR-Datei verpackt werden. Mit JavaHelp können Online-Hilfen für alle Arten von Java- oder XML-basierten Anwendungen erstellt werden. (vgl. Grünwied 2007, S. 90)

JavaHelp liegt aktuell in der Server-basierten Version 2.0 vor, die im Jahr 2003 veröffentlicht wurde. Jedoch basiert diese Version auf dem veralteten Standard HTML 3.2 und unterstützt kaum Funktionen von HTML 4.0. Dadurch fehlt eine DHTML- und JavaScript-Unterstützung. Um JavaHelp entwickeln zu können, ist

das Java Development Kit nötig, das kostenlos im Internet verfügbar ist. (vgl. Grünwied 2007, S. 90)

Die JavaHelp API ist so konzipiert, dass die Oberfläche der Hilfe flexibel erweiterbar ist. Gängig sind Anpassung der Navigation und der Suche. Die API ermöglicht zudem kontextsensitive Hilfeaufrufe aus der Softwareanwendung heraus. (vgl. Grünwied 2007, S. 90)

Für die Anzeige von JavaHelp ist der JavaHelp Viewer nötig. Dieser ist ähnlich dem HTML Help-Viewer und stellt ein dreigeteiltes Fenster dar, mit der Navigation links und dem Anzeigebereich rechts.

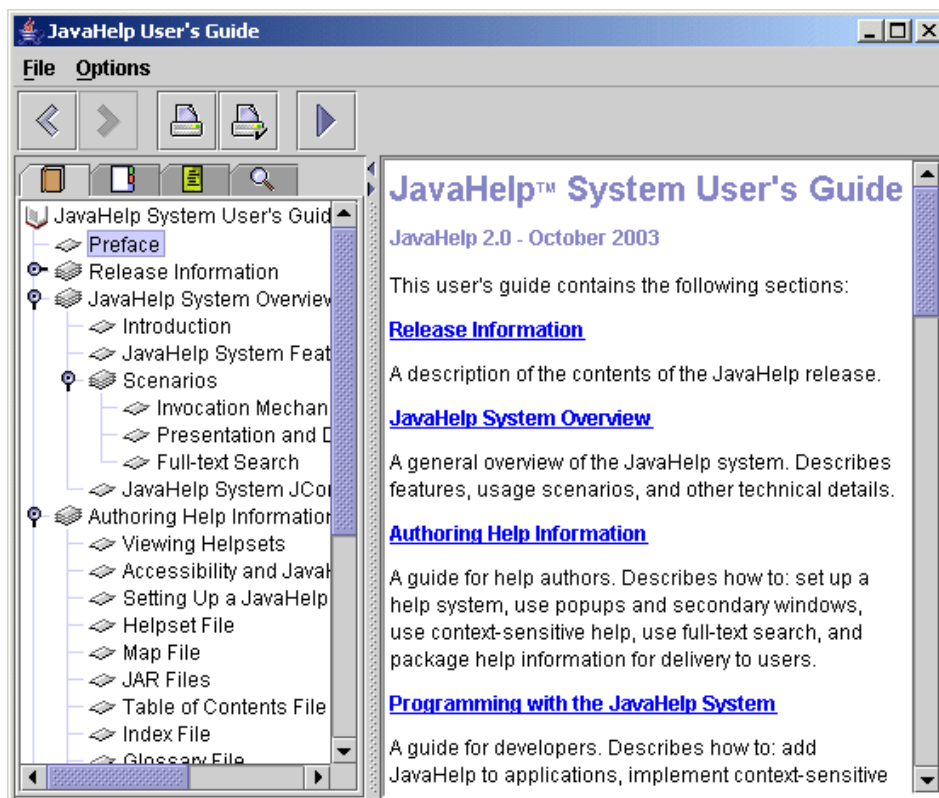


Abb. 8-4: JavaHelp-Hilfeformat (Quelle: www.sun.com)

## 8.4 Microsoft Visual Studio

Microsoft Visual Studio ist eine Entwicklungsumgebung, die alle Bereiche der Software-Entwicklung abdeckt. Das Programm ist unter Umständen auch für die XML-basierte Software-Dokumentation geeignet. Zudem sind für Visual Studio eine Vielzahl von Add-ins erhältlich, die auch den Dokumentationsprozess erleichtern. Die aktuelle Version ist Visual Studio 2008.

Das Besondere daran ist, dass der produzierte Quellcode dokumentiert werden kann. Das geschieht in Form von Kommentierungen, die im XML-Format erscheinen und editiert werden können. Visual Studio ist jedoch in erster Linie kein Editor für Technische Redakteure, sondern vielmehr ein Werkzeug für Programmierer. Auch die Quelltextdokumentation ist Entwicklersache.

```
using System;
using System.Collections.Generic;
using System.Text;
using mT.CSVPlus;

namespace ConsoleApplication1
{
    class Program
    {
        /// <summary>
        /// Addiert zwei Zahlen zusammen.
        /// </summary>
        /// <param name="a">Wert 1</param>
        /// <param name="b">Wert 2</param>
        /// <returns>Ergebnis</returns>
        private static int Addiere_Zahlen(int a, int b)
        {
            return a + b; }
        /// <summary>
        /// Subtrahiert zwei Zahlen.
        /// </summary>
        /// <param name="a">Wert 1</param>
        /// <param name="b">Wert 2</param>
        /// <returns>Ergebnis</returns>
        private static int Subtrahiere_Zahlen(int a, int b)
        {
            return a - b;
        }
    }
}
```

```
static void Main(string[] args)
{
    Console.WriteLine("Addiere...");
    Console.WriteLine("Ergebnis: " + Addiere_Zahlen(1, 1));
    Console.WriteLine("Subtrahiere...");
    Console.WriteLine("Ergebnis: " + Subtrahiere_Zahlen(2, 1));
    Console.ReadLine();
}
}
```

Beispiel: 8-1: Programmiercode mit Quelltextdokumentation

Beispiel 8-1 zeigt ein einfaches Programm in der Sprache C# zum Addieren und Subtrahieren von Zahlen. Die Quelltextdokumentation erkennt man an den XML-Tags. Daraus kann beim Kompilieren eine XML-Datei erzeugt werden:

```
<?xml version="1.0"?>
<doc>
<assembly>
<name>ConsoleApplication1</name>
</assembly>
<members>
<member name="M:ConsoleApplication1.Program.Addiere_Zahlen(System.Int32, System.Int32)">
<summary>Addiert zwei Zahlen zusammen.</summary>
<param name="a">Wert 1</param>
<param name="b">Wert 2</param>
<returns>Ergebnis</returns>
</member>
<member name="M:ConsoleApplication1.Program.Subtrahiere_Zahlen(System.Int32, System.Int32)">
<summary>Subtrahiert zwei Zahlen.</summary>
<param name="a">Wert 1</param>
<param name="b">Wert 2</param>
<returns>Ergebnis</returns>
</member>
</members>
</doc>
```

Abb. 8-5: XML-Datei nach der Kompilierung

Es existieren auch spezielle Dokumentations-Tools (Sprachcompiler) wie Sandcastle von Microsoft oder NDoc, mit denen die XML-Dateien direkt in HTML Help-Files oder Webseiten konvertiert werden können.

## 8.5 Weitere Autoren- und Erstellungswerkzeuge

Es gibt eine Reihe von Autorensystemen auf dem Markt. Diese eignen sich besonders für die Erstellung von Online-Hilfen und bedingt auch für Single-Source-Publishing. Im Folgenden werden drei weit verbreitete Systeme kurz vorgestellt.

### **RoboHelp**

RoboHelp ist ein Autorensystem zur Erstellung von Online-Hilfen in vielen Formaten. Aktuell liegt das Programm in der Version 7.0 vor und ist auch in der Adobe Technical Communication Suite enthalten. Vorteile sind die weite Verbreitung und der große Funktionsumfang inklusive XML-Unterstützung, Nachteile die manchmal zu hohe Komplexität und der vergleichsweise hohe Preis. (vgl. Achtelig 2006)

### **Flare**

Flare ist ein XML-basiertes Content Management System zur Erstellung von Online-Dokumentationen und Handbüchern. Es wurde von der US-Firma Madcap im Jahr 2006 auf dem Markt gebracht. (vgl. Ertelt 2008)

Flare unterstützt Single-Source-Publishing und kann somit verschiedene Ausgabeformate erzeugen. Vorteile bringt der leistungsstarke Editor, der jedoch auch sehr komplex ist. Ein weiterer Pluspunkt ist die XML-Unterstützung mit Sicht auf die XML-Struktur. (vgl. Achtelig 2006)

### **Author-it**

Author-it erschien unter der Flagge von der Author-it Software Corporation. Author-it ist eine Mischung aus Hilfe-Autorenwerkzeug und datenbankgestütztem Content Management System. Das System ist durch den objektorientierten Ansatz sehr effizient und unterstützt neben XML auch DITA, allerdings entfaltet Author-it seine Stärken erst ab einer bestimmten Projektgröße und ist je nach Anzahl der Autoren und Fremdsprachen sehr kostenintensiv. (vgl. Achtelig 2006)

## 9 Erstellung einer Dokumentationsvorlage

Zum praktischen Teil der Diplomarbeit, das Erstellen einer Vorlage zur Software-Dokumentation, wurde der DITA-Standard verwendet, da dieser für die Software-Dokumentation gut geeignet ist. Zur Bearbeitung wurden zwei Programme eingesetzt, die Open-Source-Entwicklerplattform Eclipse und FrameMaker 8 von Adobe.

### 9.1 Ansatz mit Eclipse

Eclipse ist eine Open-Source-Entwicklungsumgebung von der Firma Sun zur Erstellung von Software-Anwendungen. Durch die DITA-Unterstützung seit der Version 3.3 ist Eclipse auch zur Erstellung von Software-Dokumentationen einsetzbar. Die von Claude Vedovini 2008 herausgegebene DITA Open Platform enthält die aktuelle Eclipse-Version. (Vedovini 2008)

#### 9.1.1 Konfiguration

Bevor mit Eclipse DITA verarbeitet werden kann, muss das DITA Open Toolkit noch angebunden werden. Dieses enthält alle DTDs und nötigen Stylesheets für die Erzeugung von verschiedenen Ausgabeformaten. Durch die Anbindung können sehr schnell und einfach eine Printversion im PDF-Format, ein komplettes HTML Help-Hilfeformat oder auch einzelne XHTML-Dateien erzeugt werden. Weiterhin ist auch die Ausgabe von RTF-Dateien, JavaHelp- oder EclipseHelp-Files möglich (vgl. Kapitel 7.2.5). Die Ausgabe über das angebundene Toolkit muss noch konfiguriert werden, dabei sollte für jedes Ausgabeformat eine eigene Konfiguration erstellt werden. Darin muss das Format, die DITA-Map und der Zielpfad angegeben werden.

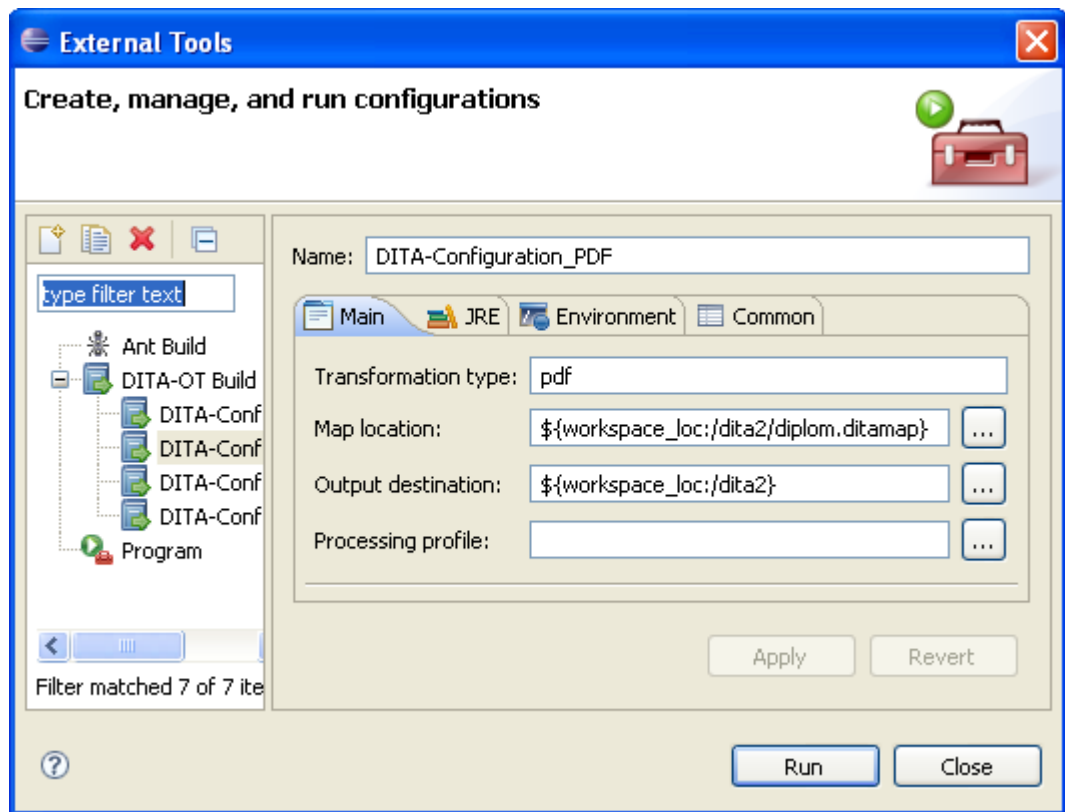


Abb. 9-1: Konfigurationsfenster für die DITA-Anbindung der Software Eclipse

### 9.1.2 Erstellung von DITA-Dateien

Um DITA-Dateien zu erstellen, muss ein neues Projekt erstellt werden. Dazu ist „File > New > Project > DITA-Project“ zu wählen und der „workspace“ anzugeben. Im Projektnavigator können dann die einzelnen DITA-Dateien mit „File > New > Other > DITA-...“ hinzugefügt werden. Bei Auswahl einer neuen DITA-Datei wird die jeweilige Grundstruktur bereits angezeigt.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN" "task.dtd" >
<task id="id20080918085252" xml:lang="de-DE">
<title></title>
<prolog>
<author>tsf</author>
</prolog>
<taskbody></taskbody>
</task>
```

Beispiel: 9-1: Anzeige bei Neuerstellung einer DITA-Task

Eclipse eignet sich sehr gut für die Bearbeitung von DITA-Projekten. Nachdem das Open Toolkit angebunden wurde, können verschiedene Konfigurationen für die Generierung erstellt werden.

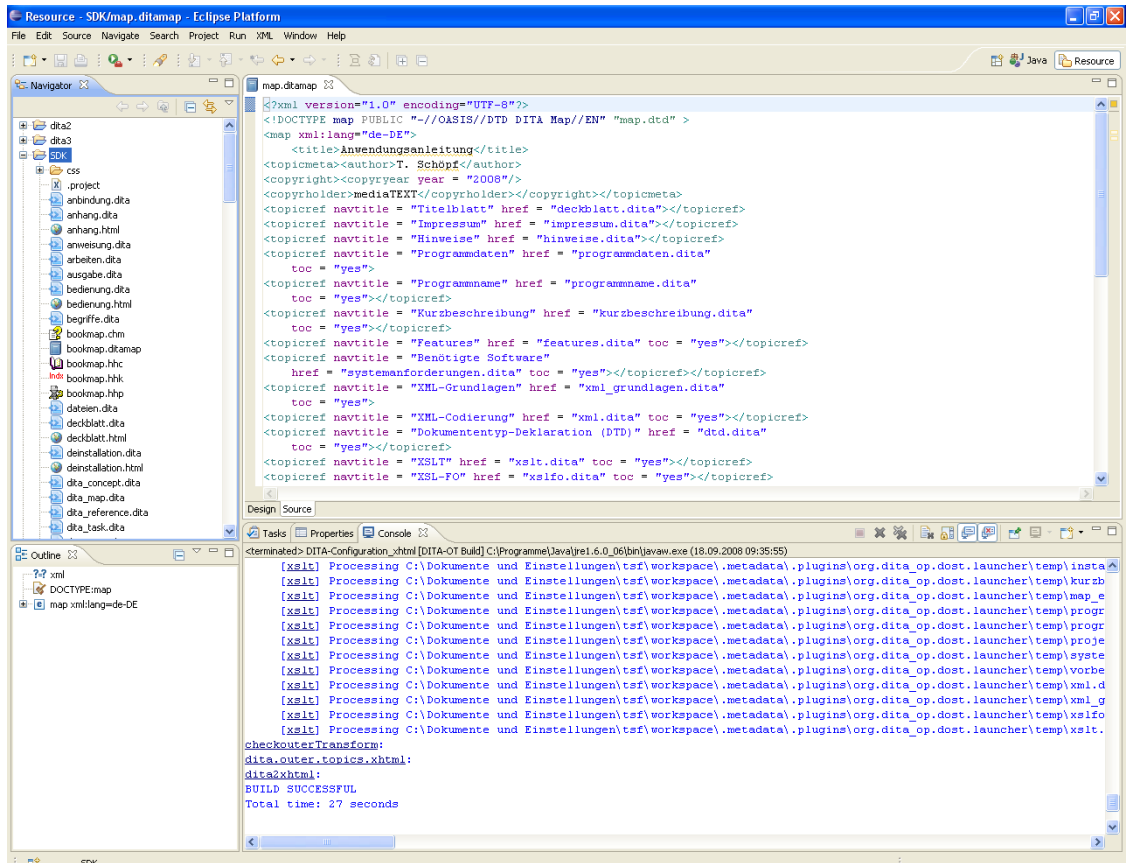


Abb. 9-2: Eclipse-Oberfläche mit DITA-Map

### 9.1.3 Ausgabe

Mit Hilfe von Eclipse und dem DITA Open Toolkit kann aus einer Datenquelle (der DITA-Datei) ein Printerzeugnis und eine Online-Hilfe erstellt werden. Das angestrebte Ziel aus Kapitel 2.2.2 für Software-Dokumentationen kann somit erreicht werden.

Die ausgegebenen Dateien werden ausschließlich durch die Prozessoren im DITA Open Toolkit erzeugt. Wie schon im Kapitel 9.1.2 beschrieben, ist eine Konfiguration für die Ausgabe mit dem Open Toolkit notwendig. Wurde diese durch-

geführt, kann durch Auswahl der jeweiligen Konfiguration oder durch Mausklick auf das Run-Icon in der Symbolleiste das dementsprechende Ausgabeformat erzeugt werden.

Die ausgegebenen Dateien erscheinen standardmäßig im Workspace-Verzeichnis, welches vorher angelegt werden muss. Wenn eine Online-Hilfe im HTML Help-Format erzeugt werden soll, werden eine HHP-Datei, eine HHK-Datei und eine TOC-Datei, sowie die HTML-Files ausgegeben. Diese müssen noch anschließend im HTML Help Workshop kompiliert werden.

## **9.2 Ansatz mit FrameMaker 8**

FrameMaker von Adobe unterstützt seit der Version 7.2 DITA. Für die aktuelle Version 8 wurde die DITA-Verarbeitung noch einmal verbessert.

### **9.2.1 Strukturierte Anwendung**

Die Verarbeitung von XML erfolgt in einer sogenannten „Strukturierten Anwendung“. Diese muss bereits beim Import der XML- oder DITA-Datei ausgewählt werden. In der Anwendung sind Regeln und Definitionen festgelegt, nach denen FrameMaker das externe Dokument während des Einlesevorgangs parst. Diese Anwendungen beinhalten beispielsweise den Pfad zur DTD oder einem Template. Sie sind in der FrameMaker-Datei „structapps.fm“ verzeichnet, welche im Installationsverzeichnis hinterlegt ist. (vgl. Krüger 2008, S. 236)

**Application name:** DITA-Map-FM  
**Template:** \$STRUCTDIR\xml\dita\app\DITA-Map-FM\dita-map.template.fm  
**DTD:** \$STRUCTDIR\xml\dita\app\dtd\map.dtd  
**Read/write rules:** \$STRUCTDIR\xml\dita\app\DITA-Map-FM\dita-map.rules.txt  
**DOCTYPE:** map  
ditamap

**Entity locations**  
**Public ID:** -//OASIS//DTD DITA Map//EN  
**Filename:** \$STRUCTDIR\xml\dita\app\dtd\map.dtd  
**Public ID:** -//IBM//DTD DITA Map//EN  
**Filename:** \$STRUCTDIR\xml\dita\app\dtd\map.dtd

**Conditional Text:**  
**Output Text PI:** OutputAllTextWithPIs  
**Use API client:** ditafm\_app

Beispiel: 9-2: Strukturierte Anwendung einer DITA-Map

Soll ein XML-Dokument im FrameMaker bearbeitet werden, ist immer eine strukturierte Anweisung nötig. Ferner muss eine DTD, ein Template und eine RW-Datei (Schreib-/Leseregeln) vorhanden sein. Diese Dateien können alle verändert werden, um individuelle Anpassungen vorzunehmen. Soll eine eigene DTD verwendet werden, muss in der structapps-Datei der Pfad angegeben werden.

Ein weiterer Teil der strukturierten Anwendung ist das Element Definitions Document (EDD). Dieses ist für die Formatierung einer XML-Datei zuständig. Zudem sind Strukturinformationen aus der DTD darin enthalten. Die EDD wird aus der DTD generiert. (vgl. Krüger 2008, S. 237)

**Element (Container):** author

**Valid as the highest-level element.**

**General rule:** (<TEXT> | keyword)\*

**Attribute list**

- |          |                |                       |          |
|----------|----------------|-----------------------|----------|
| 1. Name: | href           | String                | Optional |
| 2. Name: | keyref         | String                | Optional |
| 3. Name: | type           | Choice                | Optional |
|          | Choices:       | creator   contributor |          |
| 4. Name: | xtrc           | String                | Optional |
| 5. Name: | xtrf           | String                | Optional |
| 6. Name: | class          | String                | Optional |
|          | Default:       | - topic/author        |          |
|          | Control flags: | Read-only             |          |

Beispiel: 9-3: Auszug aus der DITA Map-EDD

FrameMaker öffnet XML- oder DITA-Dateien im Template, das in der strukturierten Anweisung genannt wird und greift nicht direkt auf das EDD zu, da die dort enthaltenen Struktur- und Formatanweisungen bereits Bestandteil des Templates sind. Damit sich Änderungen in der EDD auf das Dokument auswirken, müssen diese vorab in das Template importiert werden. (vgl. Krüger, S. 240)

## 9.2.2 Erstellung von DITA-Dateien

Um eine DITA-Datei zu erstellen, ist in der Menüleiste der Eintrag „DITA > neue DITA-Datei“ auszuwählen. Dann kann zwischen den Dateitypen gewählt werden.

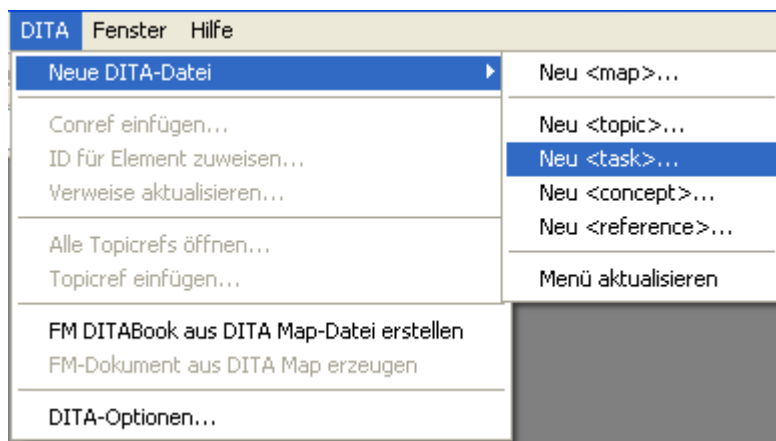


Abb. 9-3: Neue DITA-Task erstellen (FrameMaker)

Nach der Aufforderung, die Datei zu speichern, folgt ein leeres Dokument mit der DITA-Task-Struktur und den dazugehörigen Elementen. Die Bearbeitung von DITA-Dateien funktioniert nach dem WYSIWYG-Prinzip. Der Autor fügt in der Strukturansicht ein Element ein und kann im Dokumentfenster die Dateien, wie von Textprogrammen gewohnt, bearbeiten. Dazu sind keine weiteren XML-Kenntnisse nötig, da keine Berührung mit dem Quellcode vorgesehen ist.

Die Darstellung der Dokumente ist gewöhnungsbedürftig, da FrameMaker die Metadaten sowie manche Elementnamen standardmäßig mit ausgibt. Das ist bei der Bearbeitung hilfreich, wird aber in der daraus erzeugten PDF-Datei auch mit gedruckt. Um dieses zu ändern, muss die EDD angepasst werden.

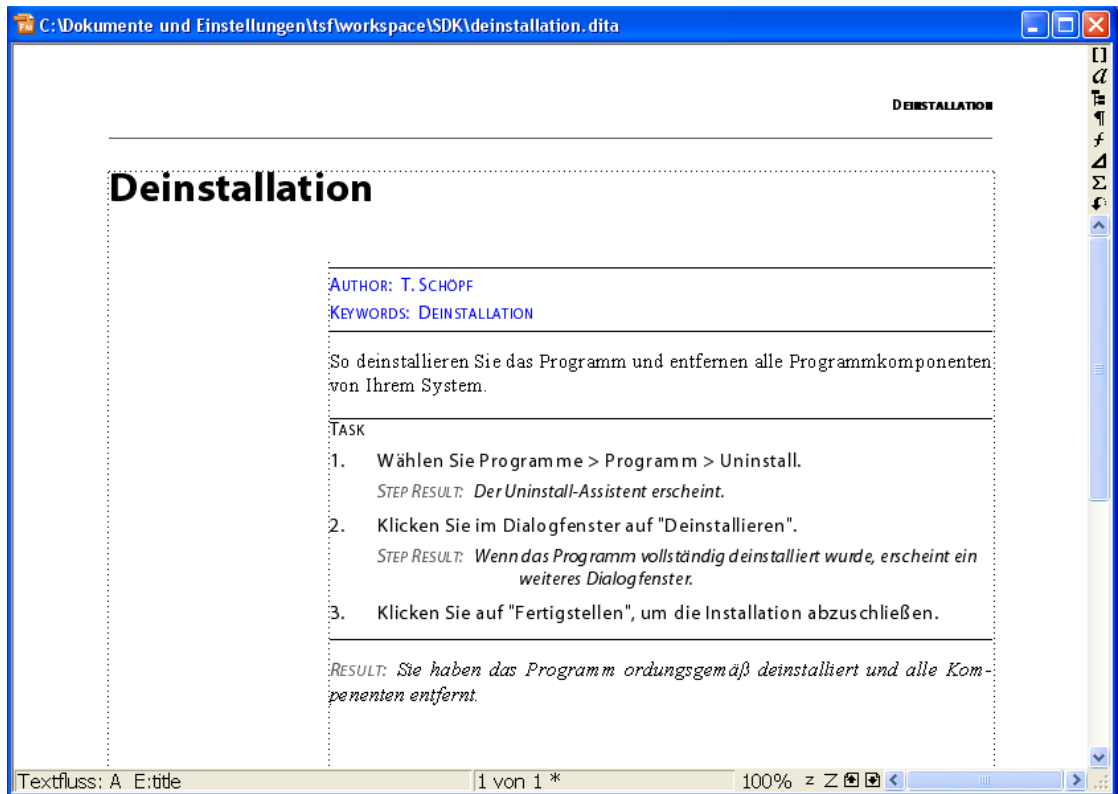


Abb. 9-4: Bearbeitung einer DITA-Task im Dokumentfenster (FrameMaker)

Die erstellten Elemente können in der Strukturansicht, verschoben und editiert werden. Hier können auch neue Elemente eingefügt werden. FrameMaker erkennt dabei, ob das gewählte Element an der jeweiligen Stelle gültig ist.

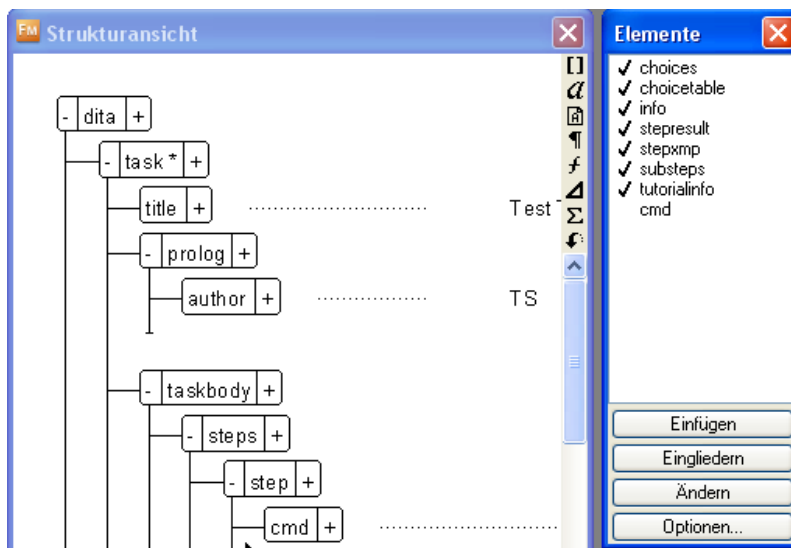


Abb. 9-5: Strukturansicht und Elementkatalog einer DITA-Task

Auch im FrameMaker können alle Topic-Grundtypen aus Kapitel 7.2 erstellt werden. Diese können dann in einer Map zusammengefasst werden. Die Map kann in ein normales (strukturiertes) FrameMaker-Dokument oder in eine Buch-Datei umgewandelt werden.

### **9.2.3 Ausgabe erzeugen**

FrameMaker 8 kann aus den DITA-Dateien eine FrameMaker-Datei oder ein Buch erzeugen, das anschließend als PDF-Dokument ausgegeben werden kann. Ohne Zusatztools sind jedoch nur PDF- und (nicht zufriedenstellende) HTML-Ausgaben möglich. Erst durch die Anbindung des DITA Open Toolkit können andere Ausgabeformate zur Erzeugung einer Online-Hilfe erstellt werden. Dazu ist auf der Adobe-Webseite ein Plug-in zum freien Download erhältlich. Dieses einzubinden gestaltet sich jedoch schwierig, da nur eine Dokumentation in englischer Sprache vorliegt und diese nicht vollständig und korrekt ist. Demzufolge war es nötig, weitere Recherchen im Internet zu diesem Thema durchzuführen. Erst dann wurde klar, dass die Reihenfolge der Schritte in der Dokumentation nicht immer richtig war. Weiterhin müssen einige Systemvariablen gesetzt werden, damit eine Ausgabe erzeugt werden kann. Eclipse fordert diese nicht, da sämtliche JAR-Dateien bereits als Plug-in integriert sind.

Das Plug-in erhält eine DLL-Datei und verschiedene XML-Dateien, werden diese korrekt eingebunden, erscheint ein neuer Eintrag im FrameMaker-Menüpunkt „DITA“. Bei Aufruf erscheint ein Dialogfenster mit den Einstellungen zur Ausgabe und den Pfad zum DITA Open Toolkit. Bei vollständig konfiguriertem Toolkit und richtiger Einbindung des Plug-ins erzeugt FrameMaker das gewünschte Ausgabeformat, welches identisch zu dem von Eclipse ist.

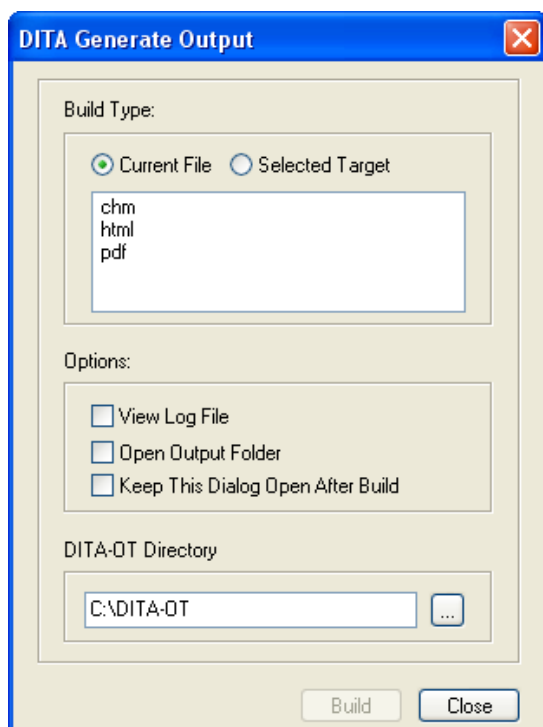


Abb. 9-6: FrameMaker-Dialogfenster zur Steuerung der Ausgabe mit DITA Open Toolkit

## 9.3 Vergleich FrameMaker – Eclipse

### 9.3.1 Einlesen/Erstellen

Die in Eclipse erstellten Maps oder Bookmaps sind nicht fehlerfrei in FrameMaker einlesbar. Der Grund hierfür ist, dass in der Standardversion von FrameMaker nicht das DITA Open Toolkit, sondern ein FrameMaker eigenes DITA-Anwendungspaket integriert ist. Dieses beinhaltet eigene DTDs und Stylesheets, auf die FrameMaker zurückgreift. Die DTDs unterscheiden sich leicht von denen des Open Toolkit, dadurch sind manche mit dem Open Toolkit erstellten Dateien nicht in FrameMaker valide. Für Bookmaps ist keine DTD in FrameMaker vorhanden, dieser Dateityp kann nicht eingelesen werden.

Ein Problem erschien, als die zuvor in Eclipse erstellte DITA-Map in FrameMaker eingelesen werden sollte. Im Open Toolkit ist beim Map-Dateityp das Element

„Title“ erlaubt, in der FrameMaker-DTD ist „Title“ aber nur als Attribut des Elementes „Map“ gültig.



Abb. 9-7: Fehlermeldung beim Laden der Map-Datei in FrameMaker

### 9.3.2 Bearbeitung

Beide Programme haben eine grundverschiedene Funktionsweise. Während in Eclipse reiner Quellcode produziert wird, legt FrameMaker Wert auf das Editieren von Dateiinhalten in der WYSIWYG-Ansicht. Besitzt der Redakteur XML-Kenntnisse, dürfte es für ihn kein Problem sein, mit Eclipse zu arbeiten. Auch das Editieren der DTDs und XSL-Dateien kann in Eclipse vollzogen werden.

Für die Bearbeitung im FrameMaker sind nicht unbedingt XML-Kenntnisse vorhanden, allerdings sollte das Vorgehen mit der strukturierten Version bekannt

sein. Die Arbeit mit dem Elementkatalog und der Strukturansicht erleichtert das Erstellen, da an jeder Stelle die jeweilig gültigen Symbole angezeigt werden.

### **9.3.3 Ausgabe**

Das DITA Open Toolkit ist eine wertvolle Hilfe und Erleichterung für das Erstellen von Hilfeformaten, allerdings sollte es immer den jeweiligen Zweck angepasst werden. Das kann durch Editieren der einzelnen XSL/FO-Dateien, der DTDs oder der CSS-Dateien geschehen. Mindestens sollte das kaskadierende Stylesheet verändert werden, sodass eine ansehnliche Ausgabe entsteht. Aufgrund der Lizenzfreiheit ist ein Editieren der Dateien beliebig möglich. Das Verändern von XSL-Stylesheets gestaltet sich jedoch als schwierig, denn viele XSL-Dateien greifen auf andere Dateien oder Templates zurück. Das Suchen aller notwendiger Stylesheets kann damit längere Zeit in Anspruch nehmen. Zudem sind zum Ändern der Templates XML-Kenntnisse erforderlich, da oft einzelne Dinge wie zum Beispiel die automatische Nummerierung in der Bildunterschrift geändert werden müssen. Auch in FrameMaker können die EDDs editiert werden.

Mit Hilfe des Open Toolkit kann Single-Source-Publishing vollzogen werden. Durch die vielen Stylesheets können aus einer DITA-Datei verschiedene Ausgabeformate und für Online-Hilfen nötige Dateiformate wie HHP-Dateien erzeugt werden. FrameMaker produziert ohne Anbindung des Toolkits nur PDF- oder HTML-Formate. DITA-Maps müssen dazu vorher in FrameMaker-Dateien oder -Bücher umgewandelt werden.

## **9.4 Erarbeitung eigenständiger Transformationsdateien**

Im Grunde kann ein DITA-Projekt mit jedem beliebigen Editor durchgeführt werden. Neben den DITA-Dateien ist es auch möglich, die kompletten DTDs und Stylesheets selbst zu erzeugen.

Das DITA Open Toolkit enthält eine Vielzahl von DTDs, Stylesheets und ANT-Skripte, die aufeinander abgestimmt sind. Eigene Dateien zur Ausgabe von DITA-basierter Dokumentation zu erarbeiten, würde eine Menge Zeit und Aufwand in Anspruch nehmen, da hier XML- sowie Programmierkenntnisse erforderlich sind, die der Technische Redakteur nicht unbedingt mitbringt. Von daher ist es besser, die vorhandenen Dateien im DITA Open Toolkit den eigenen Bedürfnissen anzupassen um eine optimale Ausgabe zu erzielen.

Soll nicht mit dem DITA-Standard dokumentiert werden, kann FrameMaker (strukturiert) eingesetzt werden. Hier müssten auch eigene DTDs bzw. EDDs erstellt werden. Danach können die XML-Dateien in FrameMaker erstellt werden. Vorteil ist dabei, dass die Dokumentation komplett an die eigenen Bedürfnisse angepasst werden kann, Nachteile gibt es bei der Ausgabe, da wie beschrieben FrameMaker nur ein ordentliches Ausgabeformat (PDF) erzeugt.

## 10 Zusammenfassung

Software-Dokumentation ist ein Teilgebiet der Technischen Dokumentation, für das spezielle Bedingungen gelten. Die Dokumentation von Software existiert nicht mehr als klassisches Handbuch, wie es früher der Fall war, es werden fast ausschließlich Online-Hilfen und elektronische Handbücher erzeugt. Dazu sollte möglichst eine Datenquelle im medienneutralen XML-Format eingesetzt werden.

Software wird oft weiterentwickelt und bei jedem Upgrade bedarf auch die Dokumentation einer Aktualisierung, somit ist auch der Weg der Distribution entscheidend. Die Dokumentation von Software sollte immer von einer externen Person erstellt werden, nie vom Entwickler selbst, da dieser oft zu viel Wissen voraussetzt oder Dinge beschreibt, die für die Nutzung der Software nicht unbedingt von Bedeutung sind.

Software-Dokumentation kann in unterschiedlicher Form vorliegen. Früher wurde lediglich zwischen elektronischer und gedruckter Form unterschieden, heute wird in kontextsensitiv und kontextunabhängig eingeteilt. Handbücher existieren kaum mehr. Dennoch sollte, neben einer Online-Hilfe, eine druckfähige PDF-Version der Dokumentation existieren.

Zur Entwicklung von Software-Dokumentation gibt es eine Vielzahl von Hilfsmitteln und Programmen. Einige wurden in dieser Arbeit kurz beschrieben. In die Tiefe eines jeden Programms zu gehen, würde den Rahmen der Diplomarbeit sprengen. Deshalb wurde das Hauptaugenmerk auf die Eclipse-Plattform und auf Adobe FrameMaker gelegt. Beide Programme unterstützen das DITA-Format.

Der XML-Standard DITA eignet sich sehr gut für die Software-Dokumentation. Durch den topicorientierten Ansatz werden die Informationen klar strukturiert und in informierende und anweisende Einheiten unterschieden. Jede Anweisung ist ein in sich geschlossener Informationsblock, jedes Thema wird isoliert und kontextunabhängig in einem eigenständigen Baustein beschrieben. Die verschiede-

nen Topics können dann in einer DITA-Map zusammengefasst und ausgegeben werden. Hier liegt auch der nächste Vorteil vom DITA-Standard. Gerade bei der Software-Dokumentation ist es notwendig, aus einer Datenquelle unterschiedliche Ausgabeformate erzeugen zu können. Eine Druckversion (PDF-Datei) und eine Online-Hilfe sind obligatorisch. Diese Formate kann mit Hilfe des frei erhältlichen DITA Open Toolkit, aber auch (mit Einschränkungen) durch Adobe FrameMaker erreicht werden. Alternativ kann in der XML-basierten Dokumentation auch eine eigene XML-Umgebung samt DTDs und Stylesheets entwickelt werden oder in FrameMaker (strukturiert) die Dokumentation aufgrund einer eigenen XML-Struktur erstellt werden.

Beide im Kapitel 9 beschriebenen Programme eignen sich gut für die DITA-basierte Software-Dokumentation. Eine Verwendung der Eclipse-Plattform kann jedoch nur mit XML-Kenntnissen erfolgen, da hier reiner Quelltext geschrieben werden muss. Dafür ist die Ausgabevielfalt größer als bei FrameMaker in der Standardversion. Jedoch sollte man immer beachten, dass diese Hilfsmittel oder Programme eine vorgegebene XML-Umgebung mitbringen. Es ist nötig, diese für die individuellen Bedürfnisse anzupassen. Dazu können DTDs (EDDs), XSL-Dateien und kaskadierende Stylesheets modifiziert werden. Durch diese individuelle Anpassung ist durch wenig Zeitaufwand eine ordentliche Dokumentation möglich ohne eine komplett neue XML-Umgebung zu schaffen.

Deshalb lautet die Empfehlung, Software-Dokumentationen mit DITA und dem DITA Open Toolkit zu erzeugen. Dazu kann das Programm der Wahl eingesetzt werden. Die Verwendung des DITA-Standards erspart eine Menge an Entwicklerarbeit. Der Technische Redakteur sollte in der Lage sein, mit dem DITA Open Toolkit alleine zu arbeiten und die nötigen Konfigurationen und Anpassungen durchzuführen. Daher ist DITA ein kostengünstiger, zeitsparender und zukunftsorientierter XML-Standard.

## Abkürzungsverzeichnis

API	Application Programm Interface
CSS	Cascading Stylesheet
DITA	Darwin Information Typing Architecture
DTD	Document Type Definition
DTP	Desktop Publishing
EDD	Element Definitions Document
HTML	Hypertext Markup Language
IMAP	Information Mapping ®
OASIS	Organization for the Advancement of Structured Information Standards
PDF	Portable Document Format
SPS	Speicherprogrammierbare Steuerung
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language – Transformations
XSL-FO	Extensible Stylesheet Language – Formatting Objects
WYSIWYG	What You See Is What You Get

---

## Abbildungsverzeichnis

Abb. 4-1: Mögliche Gliederung einer Software-Dokumentation (Quelle: <a href="http://www.tekom.de">www.tekom.de</a> ) . . . . .	19
Abb. 4-2: Makrostruktur von Softwarehandbüchern (nach Hoffmann/Schlummer 1990) . . . . .	22
Abb. 5-1: Anteil der Artikel über Software-Dokumentation (roter Balken) ab dem Jahr 2000 . . . . .	26
Abb. 6-1: Verschiedene Medien der Print- und Online-Dokumentation (nach Grünwied 2007) . . . . .	27
Abb. 6-2: Online-Hilfe der Software Eclipse, Kontextbezug zum aktuellen Projekt. . . . .	30
Abb. 6-3: Online-Hilfe der Software Adobe FrameMaker in Microsoft Help Viewer . . . . .	31
Abb. 6-4: Elektronisches Handbuch für die Software XML-Writer der Firma mediaTEXT . . . . .	32
Abb. 6-5: Dialoghilfe im Dialogfenster „Einstellungen“ der Software „Mozilla Firefox 3.0“ . . . . .	33
Abb. 6-6: aktivierte Direkthilfe (erkennbar am Fragezeichen) und angeklicktes Symbol . . . . .	34
Abb. 6-7: Tipp nach dem Öffnen der Software „XMetal Author“ . . . . .	35
Abb. 6-8: Optimale Medien für bestimmte Zwecke (nach Achtelig 04/2002). . . . .	38
Abb. 6-9: Vergleich von Print und Online (nach Achtelig 2002) . . . . .	39
Abb. 6-10: Übersicht Kontextsensitivität (nach Grünwied, 2008) . . . . .	40
Abb. 6-11: Kontextsensitive Hilfe „IntelliSense“ (Programm Visual Studio 2008) . . . . .	40
Abb. 7-1: Preprocess des DITA Open Toolkit (Quelle: <a href="http://dita-ot.sourceforge.net">http://dita-ot.sourceforge.net</a> ). . . . .	48
Abb. 7-2: IMAP-Elemente und entsprechende DITA-Elemente . . . . .	50
Abb. 7-3: Ausgabe der Transformation im Browser Mozilla Firefox 3 . . . . .	51
Abb. 8-1: Bedienoberfläche HTML Help Workshop. . . . .	54
Abb. 8-2: Hilfpfad in der Vista AP Hilfe (Quelle: <a href="http://www.help-info.de">www.help-info.de</a> ) . . . . .	55
Abb. 8-3: Microsoft Vista AP Help . . . . .	56
Abb. 8-4: JavaHelp-Hilfeformat (Quelle: <a href="http://www.sun.com">www.sun.com</a> ) . . . . .	57
Abb. 8-5: XML-Datei nach der Kompilierung. . . . .	59
Abb. 9-1: Konfigurationsfenster für die DITA-Anbindung der Software Eclipse . . . . .	62
Abb. 9-2: Eclipse-Oberfläche mit DITA-Map . . . . .	63
Abb. 9-3: Neue DITA-Task erstellen (FrameMaker) . . . . .	66
Abb. 9-4: Bearbeitung einer DITA-Task im Dokumentfenster (FrameMaker) . . . . .	67
Abb. 9-5: Strukturansicht und Elementkatalog einer DITA-Task. . . . .	67
Abb. 9-6: FrameMaker-Dialogfenster zur Steuerung der Ausgabe mit DITA Open Toolkit. . . . .	69
Abb. 9-7: Fehlermeldung beim Laden der Map-Datei in FrameMaker . . . . .	70

---

## Quelltextauszüge

Beispiel: 2-1: Quelltext eines Handlungsschrittes mit XML-Auszeichnung . . . . .	4
Beispiel: 2-2: DTD zum strukturiertem Handlungsschritt aus Beispiel 2-1 . . . . .	4
Beispiel: 7-1: Docbook-Quelltextbeispiel . . . . .	42
Beispiel: 7-2: DITA-Concept . . . . .	45
Beispiel: 7-3: DITA-Referenz . . . . .	45
Beispiel: 7-4: DITA-Task . . . . .	46
Beispiel: 7-5: DITA-Map . . . . .	47
Beispiel: 7-6: XSLT-Stylesheet . . . . .	51
Beispiel: 7-7: XSL-FO Stylesheet . . . . .	52
Beispiel: 8-1: Programmiercode mit Quelltextdokumentation . . . . .	59
Beispiel: 9-1: Anzeige bei Neuerstellung einer DITA-Task . . . . .	62
Beispiel: 9-2: Strukturierte Anwendung einer DITA-Map . . . . .	65
Beispiel: 9-3: Auszug aus der DITA Map-EDD . . . . .	65

## Quellennachweise

### Literaturquellen

Achtelig, 2002

Achtelig, Marc: Weg vom Papier? In: Technische Kommunikation – Fachzeitschrift für Technische Dokumentation und Informationsmanagement, Verlag Schmidt Römhild, Lübeck. Heft 4/2002

Achtelig, 2006

Achtelig, Marc: Von AuthorIT bis XDK. In: Technische Kommunikation – Fachzeitschrift für Technische Dokumentation und Informationsmanagement, Verlag Schmidt Römhild, Lübeck. Heft 5/2006

Closs, 2007

Closs, Sissi: Single Source Publishing – Topicorientierte Strukturierung und DITA, Software & Support Verlag, Frankfurt am Main, 2007

Edelmann, 2003

Edelmann, Anja: Hypertextbasierte Softwaredokumentation – tekomp Hochschulschriften 7; Verlag Schmidt-Römhild, Lübeck, 2003

Forst, 1996

Forst, Annyse: Online-Dokumentation. In: Krings, Hans Peter: Wissenschaftliche Grundlagen der Technischen Kommunikation. G. Narr Verlag, Tübingen, 1996

Grünwied, 2007

Grünwied, Gertrud: Software-Dokumentation, 2. Aufl., expert Verlag, Renningen, 2007

Hoffmann/Schlummer, 1990

Hoffmann, Werner; Schlummer, Werner: Erfolgreich beschreiben – Praxis des Technischen Redakteurs. MC&D-Verlag, München/Berlin, 1990

Krauß, 2007

Krauß, Stefan: Verfahren der Software-Dokumentation, Shaker Verlag, Aachen, 2007

Krüger, 2008

Krüger, Klaus: Adobe FrameMaker 8, Springer-Verlag, Heidelberg, 2008

Krüger/Welsch, 2007

Krüger, Manfred, Welsch, Ursula: XSL-FO Praxis. Software & Support Verlag, Frankfurt am Main, 2007

Ludewig/Lichter, 2007

Ludewig, Jochen; Lichter, Horst: Software Engineering: Grundlagen, Menschen, Prozesse, Techniken. dpunkt.verlag, Heidelberg, 2007

Schicht, Grünwied 2008

Schicht, Gabriele; Grünwied, Gertrud: Zehn Fragen zur Vista-Hilfe. In: Technische Kommunikation - Fachzeitschrift für Technische Dokumentation und Informationsmanagement. Verlag Schmidt Römhild, Lübeck. Heft 2/2006

Siegel, 2008

Siegel, Siegfried: Ein Regelwerk für DITA. In: Technische Kommunikation – Fachzeitschrift für Technische Dokumentation und Informationsmanagement. Verlag Schmidt Römhild, Lübeck. Heft 3/2008

Thiemann, 2001

Thiemann, Petra: Online oder Offline? In: Technische Kommunikation – Fachzeitschrift für Technische Dokumentation und Informationsmanagement. Verlag Schmidt Römhild, Lübeck. Heft 2/2001

## **Sekundärliteratur**

Norman D. A. (1987): Cognitive Engineering - Cognitive Science. In: Carroll, J. M. (Hrsg.): Interfacing thought. Cambridge, MA: MIT Press S. 325-336

Postigo, M. L. (1997): Linguistische Eigenschaften und Verständlichkeit von Software-Dokumentationen. Eine anwendungsorientierte Untersuchung. Frankfurt/Main, 1997 S. 180

## **Internetquellen**

Böhler, 2008

Böhler, Klaus: Einheitliche Strukturierung von Information mit Information Mapping. URL: <http://carstens-techdok.de/de/inhalt/pdf/ImapStrukturierung.pdf> (10.09.2008)

Datacom, 2008

DATAKOM Buchverlag GmbH (Hrsg.): DTP (desktop publishing) URL: <http://www.itwissen.info/definition/lexikon/desktop-publishing-DTP.html> (31.07.2008)

Datacom, 2008a

DATAKOM Buchverlag GmbH (Hrsg.): Lernprogramm URL: <http://www.itwissen.info/definition/lexikon/Lernprogramm-learning-program.html> (23.07.2008)

Dirnfeldner, 1996

Dirnfeldner, Martin: Struktur der Technischen Dokumentation. URL: [http://www.tekom.de/index\\_neu.jsp?url=/servlet/ControllerGUI?action=voll&id=519](http://www.tekom.de/index_neu.jsp?url=/servlet/ControllerGUI?action=voll&id=519) (04.08.2008)

Ertelt, 2008

Ertelt, Jörg: Flare. URL: <http://www.helpdesign.eu/software/madcap/flare/index.html> (08.08.2008)

e-teaching, 2006

e-teaching.org (Hrsg.): CBT (Computer Based Training) URL: <http://www.e-teaching.org/glossar/cbt> (23.07.2008)

Heinig, 2006

Heinig, Anja: Cross Media Publishing – ein Begriff und eine Idee mit vielen Facetten. URL: <http://www.marketing-boerse.de/Fachartikel/details/Cross-Media-Publishing-%96-ein-Begriff-und-eine-Idee-mit-vielen-Facetten/1345> (31.07.2008)

Holzmann, 2000

Holzmann, Martin: Information Mapping: Vorteile durch effizientes Strukturieren URL: <http://www.doculine.com/news/2000/Januar/infomap.htm> (31.07.2008)

Kulle, 2008

Kulle, Ulrich: AP Help – Überblick. URL: [http://help-info.de/de/Help\\_Info\\_AP\\_Help/ap\\_help\\_overview.htm](http://help-info.de/de/Help_Info_AP_Help/ap_help_overview.htm) (07.08.2008)

mediaTEXT, 2008

mediaText Jena GmbH (Hrsg.) URL: <http://www.mediatext.de/Dienstleistungen/> (21.07.2008)

Mintert, 2008

Stefan Mintert: Extensible Markup Language (XML) 1.1 – Deutsche Übersetzung URL: <http://edition-w3c.de/TR/2004/REC-xml11-20040204/> (22.07.2008)

Münz, 2007

Münz Stefan: SELFHTML, Version 8.1.2 vom 01.03.2007 URL: <http://de.selfhtml.org/intro/hypertext/definitionen.htm> (30.07.2008)

Neuwirt/Forster 2007

Neuwirt, Oleg; Forster, Thomas: Master Seminar - Kontextsensitivität, MS I / HS Mannheim, 19.04.2007 URL: [http://gazoo.informatik.hs-mannheim.de/master-events/pub\\_events/MSI-07/07-04-19-Kontextsensitivitaet.pdf](http://gazoo.informatik.hs-mannheim.de/master-events/pub_events/MSI-07/07-04-19-Kontextsensitivitaet.pdf) (24.07.2008)

tekom, 2008

tekom, Fachverband für Technische Kommunikation (Hrsg.) URL: [http://www.tekom.de/index\\_neu.jsp?18-D-de\\_%DCber\\_uns](http://www.tekom.de/index_neu.jsp?18-D-de_%DCber_uns) (22.07.2008)

Trieloff 2002

Trieloff, Lars: DocBook - XML. URL: <http://www.goshaky.com/docbook-tutorial/index.html> (05.08.2008)

Vedovini, 2008

Vedovini, Claude: DITA Open Plattform. URL: <http://www.dita-op.org> (19.08.2008)

W3C, 2008

World wide web consortium (Hrsg.): What is HyperText? URL: <http://www.w3.org/WhatIs.html> (30.07.2008)

Walsh/Muellner, 1999

Walsh, Norman; Muellner, Leonard: DocBook: The Definitive Guide. URL: <http://docbook.org/tdg/en/html/docbook.html> (05.08.2008)

## **Softwarequellen**

DITA Open Toolkit 1.4.2.1:

<http://sourceforge.net/projects/dita-ot/>

DITA Open Platform:

[http://sourceforge.net/project/showfiles.php?group\\_id=228491&package\\_id=276725](http://sourceforge.net/project/showfiles.php?group_id=228491&package_id=276725)

DITA Open Toolkit Plug-in für FrameMaker:

[http://www.adobe.com/devnet/framemaker/fm8\\_opentoolkit.html](http://www.adobe.com/devnet/framemaker/fm8_opentoolkit.html)

## **Bildquellen**

Abb. 4-1: Mögliche Gliederung einer Software-Dokumentation

[http://www.tekom.de/index\\_neu.jsp?url=/servlet/ControllerGUI?action=voll&id=519](http://www.tekom.de/index_neu.jsp?url=/servlet/ControllerGUI?action=voll&id=519) (04.08.2008)

Abb. 7-1: Preprocess des DITA Open Toolkit

<http://dita-ot.sourceforge.net/SourceForgeFiles/doc/DITA-OTPreprocess.html>  
(17.09.2008)

Abb. 8-2: Hilfefad in der Vista AP Hilfe

[http://www.help-info.de/de/Help\\_Info\\_AP\\_Help/ap\\_help\\_overview.htm](http://www.help-info.de/de/Help_Info_AP_Help/ap_help_overview.htm)  
(07.08.2008)

Abb. 8-4: JavaHelp-Hilfeformat

<http://java.sun.com/developer/technicalArticles/J2SE/Desktop/javahelp/>  
(08.08.2008)

## **Weitere Quellen**

Siegel, 2008a

nach einem Gespräch mit Siegfried Siegel, Technischer Leiter der CARSTENS + PARTNER GmbH, München, vom 15.07.2008

## **Eidesstattliche Erklärung**

Ich versichere an Eides statt, dass ich die Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, habe ich als solche kenntlich gemacht. Ich weiß, dass bei Abgabe einer falschen Versicherung die Prüfung als nicht bestanden zu gelten hat.

Datum, Ort

Unterschrift

---

## Anhang

I DITA-Dokumentationsvorlage .....	A2
II Dokumentation der Vorlage .....	A6
III Relevante Normen für die Software-Dokumentation .....	A17
IV Richtlinie der Technisch-Physikalischen Bundesanstalt .....	A23
V Einblick in ein Praxisprojekt .....	A28
VI IMAP-Beispiel .....	A30
VII Rechercheergebnisse „tekom“-Artikel .....	A32

---

## I DITA-Dokumentationsvorlage

### Beispiel-Reference:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE reference PUBLIC "-//OASIS//DTD DITA Reference//EN" "reference.dtd" >
<reference id="Systemanforderungen" xml:lang="de-DE">
<title>Systemanforderungen</title>
<prolog>
<author></author>
<metadata>
<keywords>
<keyword>systemanforderungen</keyword>
</keywords>
</metadata>
</prolog>
<refbody>
<properties>
<property>
<proptype>Bauteil</proptype>
<propvalue>Mindestwert</propvalue>
</property>
</properties>
</refbody>
</reference>
```

Beispiel A-1: DITA-Reference der Vorlage: Systemanforderungen

### Beispiel-Concept:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="Features" xml:lang="de-DE">
<title>Features</title>
<prolog>
<author></author>
<metadata>
<keywords>
<keyword>Features</keyword>
<keyword>Neuheiten</keyword>
<keyword>Besonderheiten</keyword></keywords>
</metadata>
```

---

```

</prolog>
<conbody>
<section>
<p><b>Besonderheiten der Software:</b></p>
<ul><li>...</li>
<li>...</li>
<li>...</li>
</ul>
</section></conbody>
</concept>

```

Beispiel A-2: DITA-Concept der Vorlage zur Featurebeschreibung

### Beispiel-Task:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN" "task.dtd" >
<task id="installation" xml:lang="de-DE">
<title>Installation</title>
<prolog>
<author></author>
<metadata><keywords><keyword>Installation</keyword></keywords></me-
tadata></prolog>
<taskbody>
<context>Für die Installation der Software gehen Sie folgendermaßen
vor:</context>
<steps><step><cmd>erster Schritt ...</cmd></step>
<step><cmd>zweiter Schritt ...</cmd></step>
<step><cmd>dritter Schritt ...</cmd>
<stepresult><p><i>Damit haben Sie das Programm installiert.</i></p></
stepresult>
</step>
</steps></taskbody></task>

```

Beispiel A-3: DITA-Task: Anweisungen zur Installation

### Vorlage-Map:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "map.dtd" >
<map xml:lang="de-DE">
<title>Anwendungsanleitung_Map</title>
<topicmeta><author></author>
<copyright><copyryear year = "2008"/>
<copyrholder></copyrholder></copyright></topicmeta>
<topicref navtitle = "Titelblatt" href = "deckblatt.dita"></topicref>

```

```
<topicref navtitle = "Impressum" href = "impressum.dita"></topicref>
<topicref navtitle = "Programmdaten" href = "programmdaten.dita"
  toc = "yes">
<topicref navtitle = "Programmname" href = "programmname.dita"
  toc = "yes"></topicref>
<topicref navtitle = "Kurzbeschreibung" href = "kurzbeschreibung.dita"
  toc = "yes"></topicref>
<topicref navtitle = "Features" href = "features.dita" toc = "yes"></
topicref>
<topicref navtitle = "Benötigte Software"
  href = "systemanforderungen.dita" toc = "yes"></topicref></topi-
cref>
<topicref navtitle = "Bedienung" href = "bedienung.dita" toc = "yes">
<topicref navtitle = "Vorbereitung" href = "vorbereitung.dita"
  toc = "yes">
<topicref navtitle = "Installation" href = "installation.dita"
  toc = "yes"></topicref>
</topicref>
<topicref navtitle = "Erste Schritte" href = "erste_schritte.dita"
  toc = "yes"></topicref>
<topicref navtitle = "Handlungsanweisung" href = "anweisung.dita"
  toc = "yes"></topicref>
<topicref navtitle = "Handlungsanweisung" href = "anweisung2.dita"
  toc = "yes"></topicref>
<topicref navtitle = "Deinstallieren" href = "deinstallation.dita"
  toc = "yes"></topicref></topicref>
<topicref navtitle = "FAQ" href = "faq.dita" toc = "yes"></topicref>
<topicref navtitle = "Anhang" href = "anhang.dita" toc = "yes"></to-
picref>
</map>
```

Beispiel A-4: DITA-Map der Vorlage

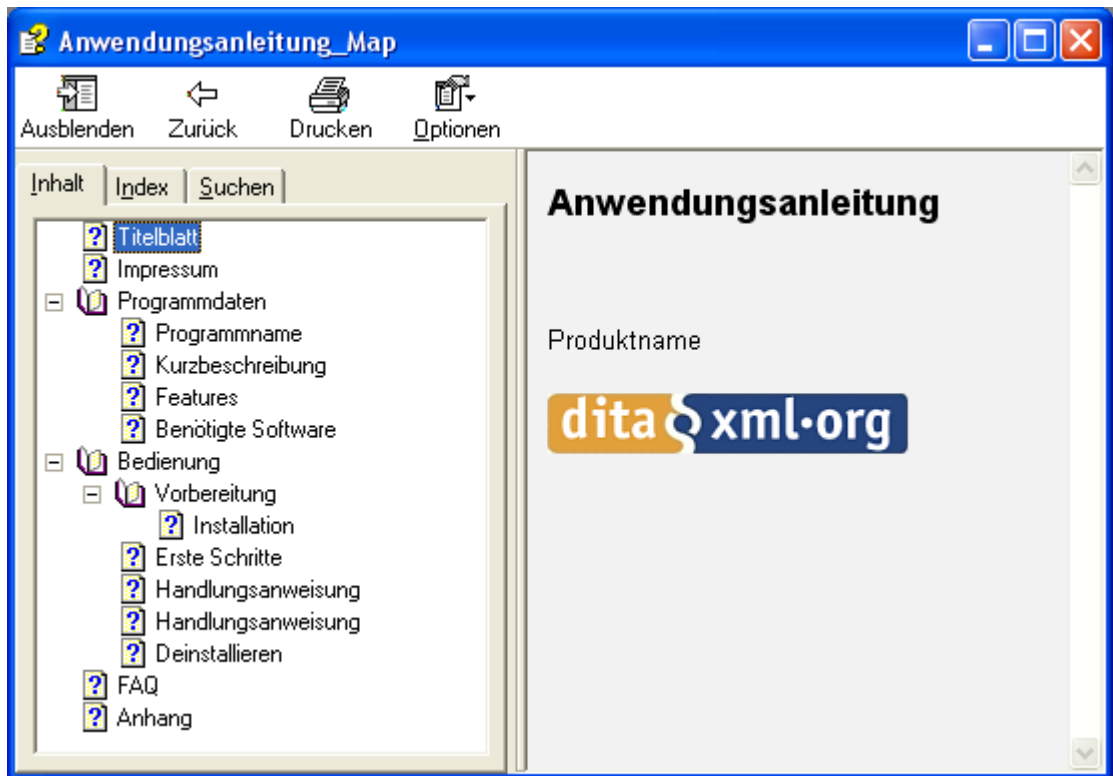
**Ausgabe: HTML Help**

Abb. A-1: Startseite mit Strukturvorschlag der kompilierten Online-Hilfe

---

## II Dokumentation der Vorlage

### Map der Dokumentation

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "map.dtd" >
<map xml:lang="de-DE">
<title>Anwendungsanleitung_Map</title>
<topicmeta><author>T. Schoepf</author>
<copyright><copyryear year = "2008"/>
<copyrholder>mediaTEXT</copyrholder></copyright></topicmeta>
<topicref navtitle = "Titelblatt" href = "deckblatt.dita"></topicref>
<topicref navtitle = "Impressum" href = "impressum.dita"></topicref>
<topicref navtitle = "Hinweise" href = "hinweise.dita"></topicref>
<topicref navtitle = "Programmdaten" href = "programmdaten.dita"
    toc = "yes">
<topicref navtitle = "Programmname" href = "programmname.dita"
    toc = "yes"></topicref>
<topicref navtitle = "Kurzbeschreibung" href = "kurzbeschreibung.dita"
    toc = "yes"></topicref>
<topicref navtitle = "Features" href = "features.dita" toc = "yes"></
topicref>
<topicref navtitle = "Benötigte Software"
    href = "systemanforderungen.dita" toc = "yes"></topi-
cref>
<topicref navtitle = "XML-Grundlagen" href = "xml_grundlagen.dita"
    toc = "yes">
<topicref navtitle = "XML-Codierung" href = "xml.dita" toc = "yes"></
topicref>
<topicref navtitle = "Dokumententyp-Deklaration (DTD)" href = "dtd.di-
ta"
    toc = "yes"></topicref>
<topicref navtitle = "XSLT" href = "xslt.dita" toc = "yes"></topicref>
<topicref navtitle = "XSL-FO" href = "xslfo.dita" toc = "yes"></topi-
cref>
<topicref navtitle = "Wichtige Begriffe" href = "begriffe.dita"
    toc = "yes"></topicref></topicref>
<topicref navtitle = "DITA" href = "dita.dita" toc = "yes">
<topicref navtitle = "Topic" href = "dita_topic.dita" toc = "yes"></
topicref>
<topicref navtitle = "Concept" href = "dita_concept.dita" toc =
"yes"></topicref>
<topicref navtitle = "Reference" href = "dita_reference.dita" toc =
```

```

"yes"></topicref>
<topicref navtitle = "Task" href = "dita_task.dita" toc = "yes"></to-
picref>
<topicref navtitle = "Map" href = "dita_map.dita" toc = "yes"></topi-
cref></topicref>
<topicref navtitle = "Bedienung" href = "bedienung.dita" toc = "yes">
<topicref navtitle = "Vorbereitung" href = "vorbereitung.dita"
    toc = "yes">
<topicref navtitle = "Installation" href = "installation.dita"
    toc = "yes"></topicref>
<topicref navtitle = "Anbindung des DITA-OT" href = "anbindung.dita"
    toc = "yes"></topicref></topicref>
<topicref navtitle = "Arbeiten mit eclipse" href = "arbeiten.dita"
    toc = "yes">
<topicref navtitle = "Erste Schritte" href = "erste_schritte.dita"
    toc = "yes"></topicref>
<topicref navtitle = "Pojekte erstellen" href = "projekt.dita"
    toc = "yes"></topicref>
<topicref navtitle = "Dateien einfügen" href = "dateien.dita"
    toc = "yes"></topicref>
<topicref navtitle = "Map zusammenstellen" href = "map_erstellen.dita"
    toc = "yes"></topicref>
<topicref navtitle = "Ausgabe konfigurieren" href = "ausgabe.dita"
    toc = "yes"></topicref></topicref></topicref>
<topicref navtitle = "FAQ" href = "faq.dita" toc = "yes"></topicref>
<topicref navtitle = "Anhang" href = "anhang.dita" toc = "yes"></to-
picref>
</map>

```

Beispiel A-5: DITA-Map der Dokumentation

## angepasstes Stylesheet „commonltr.css“

```

/*
| This file is part of the DITA Open Toolkit project hosted on
| Sourceforge.net. See the accompanying license.txt file for
| applicable licenses.
*/
/*
| (c) Copyright IBM Corp. 2004, 2005 All Rights Reserved.
*/

/*Eigene Änderungen*/

```

---

```
body
{background-color:#F2F2F2;
font-family: Arial;
font-size:10pt;
}
a:link{
color:#0000CC;
text-decoration: none
}
a:visited{
color:#0000CC;
text-decoration: none
}
a:hover{color:#5C5CFF;
text-decoration: none
}
a:active{color:#ff0000;
text-decoration: none
}
table
{background-color:#ffffff;}
li
{margin-bottom:10px;}

image
{border:none;}

/*Ende*/

.unresolved { background-color: skyblue; }
.noTemplate { background-color: yellow; }

.base { background-color: #F2F2F2; }

/* Add space for top level topics */
.nested0 { margin-top : 1em;}

/* div with class=p is used for paragraphs that contain blocks, to keep
the XHTML valid */
.p {margin-top: 1em}

/* Default of italics to set apart figure captions */
.figcap { font-style: italic }
```

---

---

```
.figdesc { font-style: normal }

/* Use @frame to create frames on figures */
.figborder { border-style: solid; padding-left : 3px; border-width :
2px; padding-right : 3px; margin-top: 1em; border-color : Silver;}
.figsides { border-left : 2px solid; padding-left : 3px; border-right
: 2px solid; padding-right : 3px; margin-top: 1em; border-color : Sil-
ver;}
.figtop { border-top : 2px solid; margin-top: 1em; border-color : Sil-
ver;}
.figbottom { border-bottom : 2px solid; border-color : Silver;}
.figtopbot { border-top : 2px solid; border-bottom : 2px solid; margin-
top: 1em; border-color : Silver;}

/* Most link groups are created with <div>. Ensure they have space
before and after. */
.ullinks { list-style-type: none }
.ulchildlink { margin-top: 1em; margin-bottom: 1em }
.olchildlink { margin-top: 1em; margin-bottom: 1em }
.linklist { margin-bottom: 1em }
.linklistwithchild { margin-left: 1.5em; margin-bottom: 1em }
.sublinklist { margin-left: 1.5em; margin-bottom: 1em }
.relconcepts { margin-top: 1em; margin-bottom: 1em }
.reltasks { margin-top: 1em; margin-bottom: 1em }
.relref { margin-top: 1em; margin-bottom: 1em }
.relinfo { margin-top: 1em; margin-bottom: 1em }
.breadcrumb { font-size : smaller; margin-bottom: 1em }
.prereq { margin-left : 20px;}

/* Set heading sizes, getting smaller for deeper nesting (Wurde ange-
passt) */
.topictitle1 { margin-top: 0pc; margin-bottom: 50px; font-size:
1.34em; }
.topictitle2 { margin-top: 1pc; margin-bottom: .45em; font-size:
1.17em; }
.topictitle3 { margin-top: 1pc; margin-bottom: .17em; font-size:
1.17em; font-weight: bold; }
.topictitle4 { margin-top: .83em; font-size: 1.17em; font-weight:
bold; }
.topictitle5 { font-size: 1.17em; font-weight: bold; }
.topictitle6 { font-size: 1.17em; font-style: italic; }
.sectiontitle { margin-top: 1em; margin-bottom: 0em; color: black;
font-size: 1.17em; font-weight: bold;}
.section { margin-top: 1em; margin-bottom: 1em }
```

---

---

```
.example { margin-top: 1em; margin-bottom: 1em }

/* All note formats have the same default presentation */
.note { margin-top: 1em; margin-bottom : 1em;}
.notetitle { font-weight: bold }
.notelisttitle { font-weight: bold }
.tip { margin-top: 1em; margin-bottom : 1em;}
.tiptitle { font-weight: bold }
.fastpath { margin-top: 1em; margin-bottom : 1em;}
.fastpathtitle { font-weight: bold }
.important { margin-top: 1em; margin-bottom : 1em;}
.importanttitle { font-weight: bold }
.remember { margin-top: 1em; margin-bottom : 1em;}
.remembertitle { font-weight: bold }
.restriction { margin-top: 1em; margin-bottom : 1em;}
.restrictiontitle { font-weight: bold }
.attention { margin-top: 1em; margin-bottom : 1em;}
.attentiontitle { font-weight: bold }
.dangertitle { font-weight: bold }
.danger { margin-top: 1em; margin-bottom : 1em;}
.cautiontitle { font-weight: bold }
.caution { font-weight: bold; margin-bottom : 1em; }
.familylinks {margin-top: 40px;}
/* Simple lists do not get a bullet */
ul.simple { list-style-type: none }

/* Used on the first column of a table, when rowheader="firstcol" is
used */
.firstcol { font-weight : bold;}

/* Various basic phrase styles */
.bold { font-weight: bold; }
.boldItalic { font-weight: bold; font-style: italic; }
.italic { font-style: italic; }
.underlined { text-decoration: underline; }
.uicontrol { font-weight: bold; }
.parmname { font-weight: bold; }
.kwd { font-weight: bold; }
.defkwd { font-weight: bold; text-decoration: underline; }
.var { font-style : italic;}
.shortcut { text-decoration: underline; }

/* Default of bold for definition list terms */
.dlterm { font-weight: bold; }
```

---

---

```
/* Use CSS to expand lists with @compact="no" */
.dltermexpand { font-weight: bold; margin-top: 1em; }
*[compact="yes"]>li { margin-top: 0em;}
*[compact="no"]>li { margin-top: .53em;}
.liexpand { margin-top: 1em; margin-bottom: 1em }
.sliexpand { margin-top: 1em; margin-bottom: 1em }
.dlexpand { margin-top: 1em; margin-bottom: 1em }
.ddexpand { margin-top: 1em; margin-bottom: 1em }
.stepexpand { margin-top: 1em; margin-bottom: 1em }
.substepexpand { margin-top: 1em; margin-bottom: 1em }

/* Align images based on @align on topic/image */
div.imageleft { text-align: left }
div.imagecenter { text-align: left }
div.imageright { text-align: right }
div.imagejustify { text-align: justify }

/* The cell border can be turned on with
   {border-right:solid}
   This value creates a very thick border in Firefox (does not match
   other tables)

   Firefox works with
   {border-right:solid 1pt}
   but this causes a barely visible line in IE */
.cellrowborder { border-left:none; border-top:none; border-right:solid 1px; border-bottom:solid 1px }
.row-nocellborder { border-left:none; border-right:none; border-top:none; border-right: hidden; border-bottom:solid 1px}
.cell-norowborder { border-top:none; border-bottom:none; border-left:none; border-bottom: hidden; border-right:solid 1px}
.nocellnorowborder { border:none; border-right: hidden;border-bottom: hidden }

pre.screen { padding: 5px 5px 5px 5px; border: outset; background-color: #CCCCCC; margin-top: 2px; margin-bottom : 2px; white-space: pre}
```

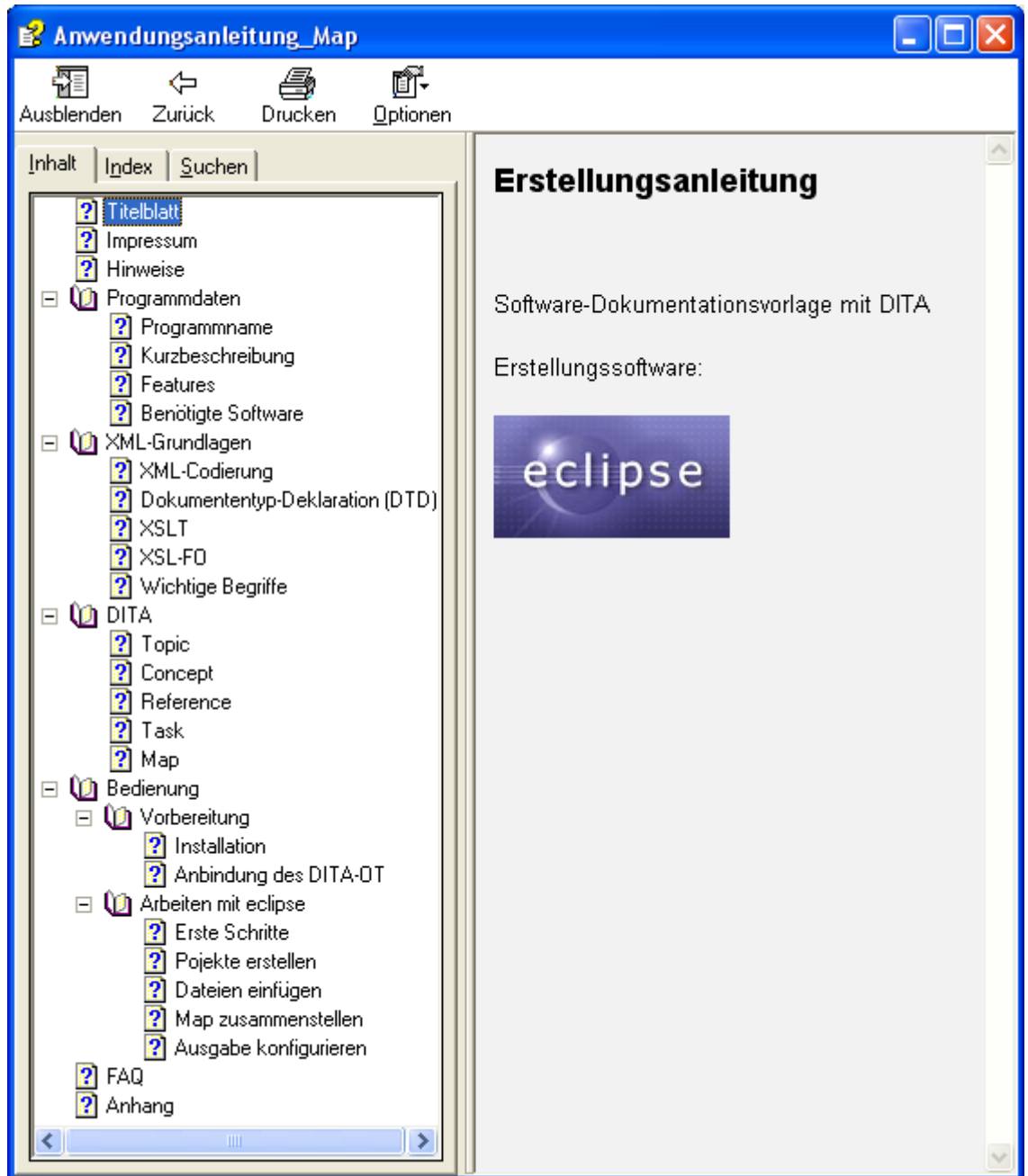
**Ausgabe: HTML Help**

Abb. A-2: Startseite mit Struktur der Dokumentation im HTML Help-Format

---

## Ausgabe: PDF (Auszug)

Anwendungsanleitung

### Bedienung

### Vorbereitung

#### Installation

Im SDK-Paket befinden sich die aktuelle Version der Eclipse-Plattform und des DITA Open Toolkit. Installieren/Entpacken Sie diese Dateien. Gehen Sie dazu folgendermaßen vor:

1. Entpacken Sie den ZIP-Ordner "doped-1.0.0.M1-20080818-win32.zip".
2. Führen Sie die Datei "eclipse.exe" aus.
3. Folgen Sie den weiteren Installationsanweisungen auf dem Bildschirm.

*Damit haben Sie die eclipse-Entwicklungsumgebung installiert.*

4. Entpacken Sie den ZIP-Ordner "DITA-OT1.5\_full\_easy\_install\_M4\_bin".

*Das DITA-OT muss nicht installiert, nur angebunden werden. Das wird im [nächsten Schritt](#) erklärt.*

#### Anbindung des DITA Open Toolkit

Um die gewünschten Ausgaben zu erzielen, muss das DITA Open Toolkit an eclipse angebunden werden. Dieses haben Sie zuvor schon entpackt.

1. Öffnen Sie eclipse und bestimmen Sie den Ablageort des Ordners "workspace".

**Hinweis:** In diesem Ordner werden Ihre erstellten Dateien zukünftig gespeichert

2. Wählen Sie in der Menüleiste "Window > Preferences"
3. In der Navigation auf der linken Seite erscheint der Eintrag "DITA Open Toolkit". Klicken Sie diesen an.
4. Klicken Sie "Browse" und stellen Sie im Dialogfenster den Pfad zum entpackten Ordner mit dem DITA Open Toolkit ein.
5. Klicken Sie auf "OK" im Dialogfenster.

*Es erscheint der Pfad im "Preferences"-Fenster*

6. Beenden Sie Ihre Einstellungen durch "OK".

## Anwendungsanleitung

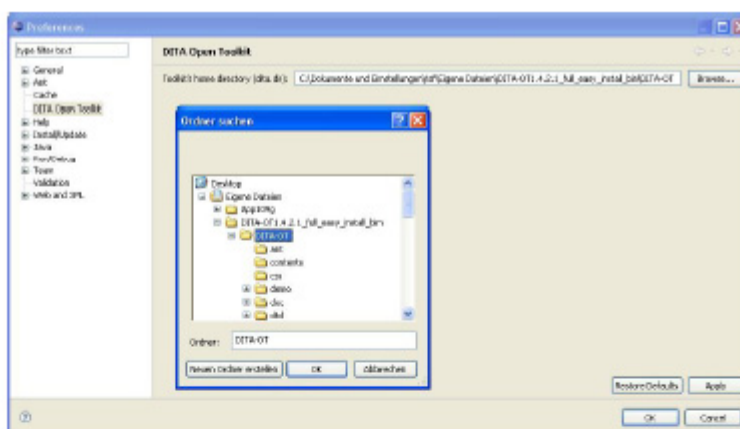


Abbildung 10: Korrekte Anbindung des DITA-OT

Damit haben Sie die das DITA Open Toolkit angebunden.

## Arbeiten mit eclipse

### Erste Schritte

Um eine Dokumentation mit DITA in eclipse zu erstellen, gehen Sie wie folgt vor:

1. Erstellen Sie ein [Projekt](#).
2. Erstellen Sie die [DITA-Dateien](#).
3. Fassen Sie diese in einer [DITA-Map](#) zusammen.
4. Konfigurieren Sie die [Ausgabe](#).

### Projekterstellung

Zuerst müssen Sie ein neues DITA-Projekt erstellen. Hier erfahren Sie, was Sie genau machen müssen.

1. Wählen Sie "New > Projekt".
2. Wählen Sie im folgenden Dialogfenster im Ordner DITA "DITA Projekt" und klicken Sie "Next".
3. Geben Sie den Projektnamen ein und klicken Sie auf "Finish".

Sie haben ein Projekt erstellt. [Fügen Sie nun DITA-Dateien ein.](#)

### Erstellen von DITA-Dateien

Hier erfahren Sie, wie Sie ihrem Projekt References, Concepts oder Tasks hinzufügen.

1. Klicken Sie mit der rechten Maustaste auf Ihr DITA-Projekt in der Navigator-Ansicht.
2. Wählen Sie "New > Other".

Anwendungsanleitung

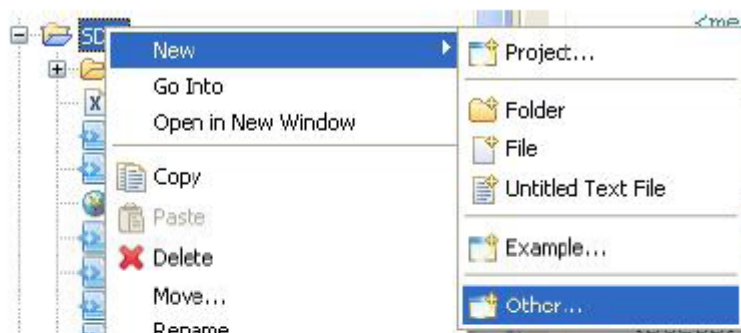


Abbildung 11: Neue DITA-Datei erstellen

3. Wählen Sie im folgendem Dialogfenster Ihren gewünschten Dateityp aus dem DITA-Ordner.
4. Benennen Sie Ihre Datei und geben Sie das Zielverzeichnis an.
5. Schließen Sie die Erstellung mit "Finish" ab.

### Erstellen von DITA-Maps

Wie Sie Ihre DITA-Dateien zu einer Map zusammenführen, erfahren Sie hier.

1. Klicken Sie mit der rechten Maustaste auf Ihr Projekt.
2. Wählen Sie "New > Other".
3. Wählen Sie im folgendem Dialogfenster "DITA Map" aus dem DITA-Ordner.
4. Fügen Sie mit dem Element "topicref" Ihre Dateien in die Map ein.

```
<topicref navtitle="XML-Grundlagen" href="xml_grundlagen.dita" toc="yes"/>
<topicref navtitle="XML-Codierung" href="xml.dita" toc="yes"/>
  <topicref navtitle="Dokumententyp-Deklaration (DTD)" href="dtd.dita" toc="yes"/>
  <topicref navtitle="XSLT" href="xslt.dita" toc="yes"/>
  <topicref navtitle="XML-FO" href="xalfo.dita" toc="yes"/>
  <topicref navtitle="Wichtige Begriffe" href="begriffe.dita" toc="yes"/>
</topicref>
```

Abbildung 12: Beispiel für eine DITA-Map

### Ausgabe konfigurieren

So erzeugen Sie aus Ihrer Map verschiedene Ausgabeformate:

1. Wählen Sie "Run > External Tool > External Tools Dialog"

Das Dialogfenster "External Tools" erscheint.

2. Klicken Sie mit der rechten Maustaste auf "DITA-OT Build" und wählen Sie "New".
3. Geben Sie in den Textfeldern rechts den Namen Ihrer Konfiguration an.
4. Geben Sie im Feld "Transformation type" Ihr gewünschtes Ausgabeformat an.

**Hinweis:** Es können die Ausgabeformate xhtml, eclipsehelp, eclipsecontent, javahelp, htmlhelp, pdf, pdf2, troff, docbook und wordrtf erzeugt werden.

5. Geben Sie den Pfad zu Ihrer DITA-Map an.
6. Wählen Sie den Ablageort Ihrer erzeugten Dateien.
7. Optional können Sie noch den Pfad eines ditaval processing profils und weitere processing arguments hinzufügen.

Anwendungsanleitung

8. Klicken Sie auf "Run", um den Ausgabeprozess zu starten.

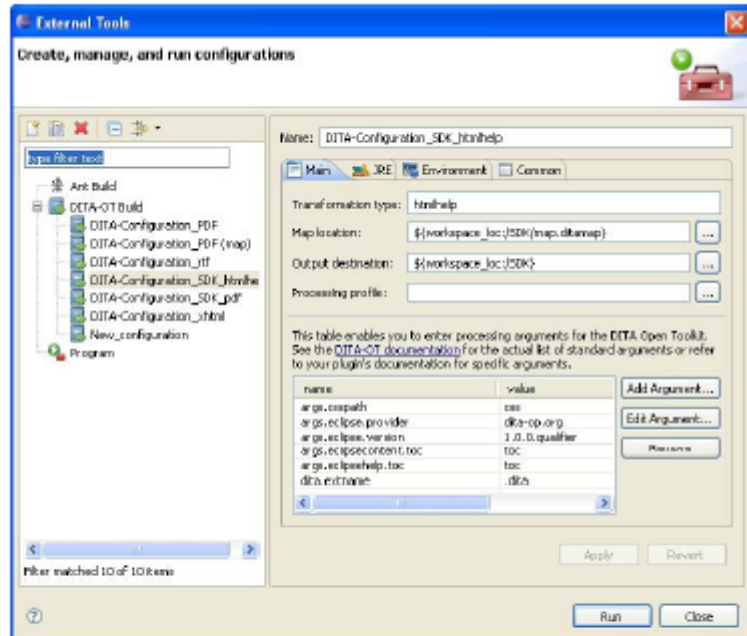


Abbildung 13: Einstellungen für ein HTML-Help-Ausgabeformat


**Hinweis:** Wenn Sie einmal ein Ausgabeformat konfiguriert haben, können Sie den Ausgabeprozess mit dem Symbol  in der Symbolleiste starten.

Abb. A-3: Auszug aus PDF der Anleitung

---

## III Relevante Normen für die Software-Dokumentation

Normen zur Informationstechnik, Software und Software-Dokumentation.

Zusammengestellt von Carl-Heinz Gabriel.

Erschienen in „Technische Kommunikation“, Heft 2/2001

**DIN V 32734**, Ausgabe: 1992-04-00, Digitale Automation für die Technische Gebäudeausrüstung; Allgemeine Anforderungen für die Planung und Ausführung (Digitale Gebäudeautomation)

(Norm-Entwurf) **DIN 33880**, Ausgabe: 1999-09, Informationstechnik – Programmierschnittstelle für Sprachsteuerung (VOC-API), Originalsprache: Deutsch

**DIN 44300-4**, Ausgabe: 1988-11, Informationsverarbeitung – Begriffe – Programmierung, Originalsprache: Deutsch

**DIN 66230**, Ausgabe: 1981-01, Informationsverarbeitung; Programmdokumentation, Originalsprache: Deutsch

**DIN 66230 Beiblatt 1**, Ausgabe: 1981-01, Informationsverarbeitung; Programmdokumentation mit fester Gliederung, Originalsprache: Deutsch

**DIN 66231**, Ausgabe: 1982-10, Informationsverarbeitung; Programmentwicklungsdokumentation, Originalsprache: Deutsch

**DIN 66270**, Ausgabe: 1998-01-00, Informationstechnik – Bewerten von Softwaredokumenten – Qualitätsmerkmale

**DIN 66272**, Ausgabe: 1994-10-00, Informationstechnik – Bewerten von Softwareprodukten – Qualitätsmerkmale und Leitfaden zu ihrer Verwendung; Identisch mit ISO/IEC 9126:1991

**DIN 66285**, Ausgabe: 1994-02-00, Informationstechnik; Software-Pakete; Qualitätsanforderungen und Prüfbestimmungen (ISO/IEC 12119:1993)

**DIN EN 50128 VDE 0831 Teil 128**, Ausgabe: 1996-06-00, Bahnanwendungen – Software für Eisenbahnsteuerungs- und -überwachungssysteme; Deutsche Fassung prEN 50128:1995

**DIN EN ISO 9241-1**, Ausgabe: 1997-09, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 1: Allgemeine Einführung (ISO 9241-1:1997); Deutsche Fassung EN ISO 9241-1:1997, Originalsprache: Deutsch

---

**DIN EN ISO 9241-5**, Ausgabe: 1999-08, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 5: Anforderungen an Arbeitsplatzgestaltung und Körperhaltung (ISO 9241-5:1998); Deutsche Fassung ISO 9241-5:1999, Originalsprache: Deutsch

(Norm-Entwurf) **DIN EN ISO 9241-6**, Ausgabe: 1998-07, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 6: Leitsätze für die Arbeitsumgebung (ISO/DIS 9241-6:1998); Deutsche Fassung prEN ISO 9241-6:1998, Originalsprache: Deutsch

**DIN EN ISO 9241-7**, Ausgabe: 1998-12, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 7: Anforderungen an visuelle Anzeigen bezüglich Reflexionen (ISO 9241-7:1998); Deutsche Fassung EN ISO 9241-7:1998, Originalsprache: Deutsch

**DIN EN ISO 9241-8**, Ausgabe: 1998-04, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 8: Anforderungen an Farbdarstellungen (ISO 9241-8:1997); Deutsche Fassung EN ISO 9241-8:1997, Originalsprache: Deutsch

(Norm-Entwurf) **DIN EN ISO 9241-9**, Ausgabe: 1998-10, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 9: Anforderungen an Eingabemittel, ausgenommen Tastaturen (ISO/DIS 9241-9:1998); Deutsche Fassung prEN ISO 9241-9:1998, Originalsprache: Deutsch

**DIN EN ISO 9241-10**, Ausgabe: 1996-07, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 10: Grundsätze der Dialoggestaltung (ISO 9241-10:1996); Deutsche Fassung EN ISO 9241-10:1996, Originalsprache: Deutsch

**DIN EN ISO 9241-11**, Ausgabe: 1999-01, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 11: Anforderungen an die Gebrauchstauglichkeit; Leitsätze (ISO 9241-11:1998); Deutsche Fassung EN ISO 9241-11:1998, Originalsprache: Deutsch

**DIN EN ISO 9241-12**, Ausgabe: 2000-08, Ergonomische Anforderungen für Bürotätigkeit mit Bildschirmgeräten – Teil 12: Informationsdarstellung (ISO 9241-12:1998); Deutsche Fassung EN ISO 9241-12:1998, Originalsprache: Deutsch

**DIN EN ISO 9241-13**, Ausgabe: 2000-08, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 13: Benutzerführung (ISO 9241-13:1998); Deutsche Fassung EN ISO 9241-13:1998, Originalsprache: Deutsch

**DIN EN ISO 9241-14**, Ausgabe: 2000-12, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 14: Dialogführung mittels Menüs (ISO 9241-14:1997); Deutsche Fassung EN ISO 9241-14:1999, Originalsprache: Deutsch

---

**DIN EN ISO 9241-15**, Ausgabe: 1999-08, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 15: Dialogführung mittels Kommandosprachen (ISO 9241-15:1997); Deutsche Fassung EN ISO 9241-15:1997, Originalsprache: Deutsch

**DIN EN ISO 9241-16**, Ausgabe: 2000-03, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 16: Dialogführung mittels direkter Manipulation (ISO 9241-16:1999); Deutsche Fassung EN ISO 9241-16:1999, Originalsprache: Deutsch

**DIN EN ISO 9241-17**, Ausgabe: 2000-04, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 17: Dialogführung mittels Bildschirmformularen (ISO 9241-17:1998); Deutsche Fassung EN ISO 9241-17:1998, Originalsprache: Deutsch

**DIN EN ISO 13407**, Ausgabe: 2000-11, Benutzer-orientierte Gestaltung interaktiver Systeme (ISO 13407:1999); Deutsche Fassung EN ISO 13407:1999, Originalsprache: Deutsch

(Norm-Entwurf) **DIN IEC 65C/220/CD**, Ausgabe: 2000-08, Funktionsbausteine für Leitsysteme in der Verfahrenstechnik – Teil 1: Allgemeine Anforderungen (IEC 65C/220/CD:1999, Text Englisch), Originalsprache: Englisch

**DIN IEC 60880**, Ausgabe: 1987-08-00, Software für Rechner im Sicherheitssystem von Kernkraftwerken; Identisch mit IEC 60880, Ausgabe 1986

**DIN IEC 60880-1**, Ausgabe: 1999-11-00, Software für Rechner in Sicherheitssystemen von Kernkraftwerken – Ergänzung 1: Vorkehrungen gegen Versagen aufgrund von Software-Fehlern – Verwendung von Software-Werkzeugen und vorgefertigter Software (IEC 45A/357/CDV:1999)

**DIN ISO/IEC 12119**, Ausgabe: 1995-08 Informationstechnik – Software-Erzeugnisse – Qualitätsanforderungen und Prüfbestimmungen; Identisch mit ISO/IEC 12119:1994

**prEN 50128**, Ausgabe: 2000-05-00, Bahnanwendungen – Software für Eisenbahnsteuerungs- und Überwachungssysteme

**IEC 61506, CEI 61506**, Ausgabe: 1997-02-00, Leittechnik für industrielle Prozesse – Dokumentation der Anwendersoftware

(Norm-Entwurf) **ISO 9241-3 FDAM 1**, Ausgabe: 2000-09, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 3: Anforderungen an visuelle Anzeigen; Änderung 1, Originalsprache: Englisch

**ISO/IEC 6592**, Ausgabe: 2000-03-00, Informationstechnik – Leitfaden für die Dokumentation von computergestützten Anwendungssystemen

**ISO/IEC 15910**, Ausgabe: 1999-12-00, Informationstechnik – Prozessmodell für die Erstellung einer Software-Benutzerdokumentation

**VDI/VDE 3559**, Ausgabe: 1981-02-00, Umfang der Dokumentation von Hardware und Software für Prozessrechensysteme

**VDI/VDE 3542 Blatt 4**, Ausgabe: 2000-10, Sicherheitstechnische Begriffe für Automatisierungssysteme – Zuverlässigkeit und Sicherheit komplexer Systeme (Begriffe), Originalsprache: Deutsch, Englisch

**AWV PC-Finanzbuchführung** ; Ausgabe: 1989-00-00, PC-Finanzbuchführung, Anforderungen an Hardware, Software, Dokumentation und Prüfbarkeit

**BB 22443**, Ausgabe: 1988-04-00, Geräte und Anlagen der Fernwirktechnik; Dokumentation von Softwaresystemen

**DGQ-ITG 12-52**, Ausgabe: 1992-00-00, Methoden und Verfahren der Software-Qualitätssicherung

**CWA 14040**, Ausgabe: 2000-10, A Standardization Work Programme for „Learning and Training Technologies & Educational Multimedia Software“, Originalsprache: Englisch

**IEEE-Standards:**

**730-1998** IEEE Standard for Software Quality Assurance Plans

**730.1-1995** IEEE Guide for Software Quality Assurance Planning (Revision and redesignation of IEEE Std 983-1986)

**828-1998** IEEE Standard for Software Configuration Management Plans

**830-1998** IEEE Recommended Practice for Software Requirements Specifications

**982.1-1988** IEEE Standard Dictionary of Measures to Produce Reliable Software

**982.2-1988** IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software

**1008-1987 (R1993)** IEEE Standard for Software Unit Testing

---

**1012-1998** IEEE Standard for Software Verification and Validation

**1012a-1998** IEEE Standard for Software Verification and Validation — Content Map to

**1016-1998** IEEE Recommended Practice for Software Design Descriptions

**1028-1997** IEEE Standard for Software Reviews

**1042-1987** (R1993) IEEE Guide to Software Configuration Management (This standard has been withdrawn.)

**1044-1993** IEEE Standard Classification for Software Anomalies (Includes and shipped with IEEE Std 1044.1-1995)

**1044.1-1995** IEEE Guide to Classification for Software Anomalies

**1045-1992** IEEE Standard for Software Productivity Metrics

**1058-1998** IEEE Standard for Software Project Management Plans

**1059-1993** IEEE Guide for Software Verification and Validation Plans (This standard has been withdrawn.)

**1061-1998** IEEE Standard for a Software Quality Metrics Methodology

**1063-1987** (R1993) IEEE Standard for Software User Documentation

**1074-1997** IEEE Standard for Developing Software Life Cycle Processes

**IEEE-Standards zum Thema „Software-Engineering“:**

**1219-1998** IEEE Standard for Software Maintenance

**1320.1-1998** IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0

**1320.2-1998** IEEE Standard Conceptual Modeling Language — Syntax and Semantics for IDEF1F1X97 (IDEFobject)

**1348-1995** IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools

---

**1420.1-1995** IEEE Standard for Information Technology — Software Reuse — Data Model for Reuse Library Interoperability: Basic Interoperability Data Model (BIDM)

**1420.1a-1996** IEEE Guide for Information Technology — Software Reuse-Data Model for Reuse Library Interoperability: Asset Certification Framework

**1420.1b-1999** IEEE Trial-Use Supplement to IEEE Standard for Information

**1430-1996** IEEE Guide for Information Technology — Software Reuse-Concept of Operations for Interoperating Reuse Libraries

**1462-1998** IEEE Standard Adoption of ISO/IEC 14102: 1995 — Guidelines for the Evaluation and Selection of CASE Tools

**1471-2000** IEEE Recommended Practice for Architectural Description of Software-Intensive Systems

**1490-1998** IEEE Guide to the Project Management Body of Knowledge, Adoption of PMI Standard

**Andere „Software-Engineering“ Standards:**

**ANSI/ISO/ASQ Q9000-3-1997**, Quality Management and Quality Assurance Standards – Part 3: Guidelines for the Application of ANSI/ISO/ASQC Q9001-1994 to the Development, Supply, Installation and Maintenance of Computer Software

**J-STD-016-1995** EIA/IEEE Interim Standard for Information Technology Software Life Cycle Processes Software Development Acquirer-Supplier Agreement (Issued for Trial Use)

## IV Richtlinie der Technisch-Physikalischen Bundesanstalt

Nr.	Titel	Beschreibungsinhalte
1	Programmkenndaten	Kurzer Programmüberblick.
1.1	Programmidentifizierung	
1.1.1	Programmname	Bezeichnung zur Identifizierung des Programms.
1.1.2	Systemzuordnung des Programms	Angabe des übergeordneten Programms im Falle einer untergeordneten Programmeinheit.
1.1.3	Variantenbezeichnung	Zusatz zum Programmnamen, um mehrere inhaltlich unterschiedliche Programmvarianten zu identifizieren.
1.1.4	Aktuelle Programmversion	Zusatz zum Programmnamen oder zur Variantenbezeichnung, um die zeitlich aktuelle Programmversion zu benennen (z.B. Nummer, Datum).
1.1.5	Freigabedatum des Programms	Datum der Freigabe der ersten und aktuellen Programmversion.
1.2	Deskriptoren	Angabe von Schlagwörtern/Suchbegriffen für das Programm.
1.3	Programmkurzbeschreibung	
1.3.1	Aufgabe des Programms	Kurze anwendungsbezogene Beschreibung der Aufgabe, die mit dem Programm gelöst wird.
1.3.2	Programminhalt	Kurze Beschreibung der zur Lösung der Aufgabe verwendeten Methoden, Theorien und Berechnungsverfahren.
1.3.3	Vorschriften	Hinweise auf wichtige Gesetze, Normen, Richtlinien, die beachtet werden müssen.
1.3.4	Besonderheiten	Angaben zu - fachlichen Voraussetzungen oder Einschränkungen, - technischen Anwendungsgrenzen (z.B. maximale Antwortzeit), - vertraglichen Informationen (Copyright, Lizenzen, Nutzungsbedingungen, Garantiefragen).
1.4	Gerätebedarf	Angabe der für den Programmablauf nötigen Geräte (Prozessoren, Datenperipherie, Prozessperipherie), Geräteanforderungen, Geräterestrictionen.

Nr.	Titel	Beschreibungsinhalte
1.5	Programmgröße	Maximaler Speicherbedarf (Angabe in kBytes).
1.6	Programmbedarf	
1.6.1	Betriebssysteme	Angabe der Betriebssysteme, unter dessen Steuerung das Programm läuft.
1.6.2	Programmiersprachen/ Compiler	Genauere Angaben zur Sprach- und Compilerversion.
1.6.3	Sonstige Programme	Auflistung der benötigten Hilfsprogramme bzw. Programmbibliotheken.
1.7	Datenorganisation	Auflistung aller verwendeten Dateien (Name, Zweck, Zugriffsart, Speicherbedarf, Anwendungsgrenzen). Angaben zur Verwendung eines Datenbankbetriebssystems.
1.8	Dokumentationsumfang	Verzeichnis der vorhandenen Dokumentationsteile (Systemdokumentation, Benutzerdokumentation, Programmentwicklungsdokumentation, Projektmanagementdokumentation).
1.9	Zuständigkeiten	Genauere Angaben zum Programmautor, Verantwortlichkeiten für Programmpflege, Fehlerbeseitigung, Vertrieb.
2	Programmfunktion	
2.1	Aufgabenstellung	
2.1.1	Aufgabenbeschreibung	Detaillierte Beschreibung der zu lösenden Aufgabe (Überblick, Zusammenhänge, Was soll gelöst werden?)
2.1.2	Theoretische Grundlagen/ Randbedingungen/ Literaturhinweise	Grobe Beschreibung der angewendeten Modelle/ Verfahren (Auf welcher Grundlage wird die Aufgabe gelöst?)
2.1.3	Maßeinheiten	Auflistung der benutzten physikalischen Größen mit entsprechenden Maßeinheiten.
2.1.4	Vorschriften	Vollständige Auflistung der zu beachtenden Gesetze, Normen, Richtlinien mit Quellenangaben. Abweichungen sind zu vermerken. Informationstechnische und anwendungsbezogene Verarbeitungsbestimmungen.
2.2	Aufgabenlösung	
2.2.1	Funktionshierarchie	Hierarchische Struktur von Aufgaben und Unteraufgaben (Bezug zum Programmaufbau).

Nr.	Titel	Beschreibungsinhalte
2.2.2	Methoden/Algorithmen	Exakte Beschreibung der Lösungsverfahren mit Anwendungsgrenzen (Wie wird die Aufgabe gelöst?). Beschreibung von Plausibilitätskontrollen.
2.2.3	Fehlerbehandlung	Angabe der vorgesehenen Fehlermeldungen und der daraus resultierenden Maßnahmen.
3	Programmaufbau	
3.1	Programmbausteine	Komplette Liste aller Unterprogramme, Prozeduren, Moduln, Klassen, Methoden, Ereignisse, Regeln, globalen Variablen und Konstanten.
3.2	Programmstruktur	Strukturübersicht mit Schnittstellendarstellung, z.B. grafisch als Aufrufhierarchie oder Jackson-Diagramm bzw. Klassenhierarchie.
3.3	Quelltext des Programms	
3.4	Kompilieren und Linken	Liste aller Schritte zur Erzeugung eines ablauffähigen Programms (z.B. Makefile).
4	Programmablauf	
4.1	Programmablaufbeschreibung	Darstellung des internen Programmablaufs mit Querverweisen auf den Quelltext (Programmablaufplan nach DIN 66001, Struktogramm nach DIN 66261).
4.2	Nutzungsanalyse	Präsentation der Außenansicht auf das System, Verwendung von UML-Diagrammen (Unified Modeling Language)
4.3	Dialogablauf	Programmablaufplanähnliche genaue Darstellung des Dialogablaufs (Darstellung der Bedingungen, die zu Dialogelementen führen).
4.4	Dialogbeschreibung	Detaillierte Beschreibung der einzelnen Bildschirmlayouts.
4.5	Datenflussbeschreibung	Darstellung des Datenflusses nach DIN 66001, Methode der Strukturierten Analyse (SA).
5	Datenorganisation	

Nr.	Titel	Beschreibungsinhalte
5.1	Eingabedaten	Beschreibung der Datenstrukturen - bereits gespeicherter Daten (Dateien mit Datenfeldern, Tabellen mit Attributen, Klassen mit Instanzvariablen), - zu erfassender Daten (über Tastatur, Eingabegeräte). Angaben zu Zugriffsverfahren, Standard-Initialisierungen, zulässigen Eingabebereichen und maximal zu verarbeitenden Datenmengen.
5.2	Ausgabedaten	Beschreibung von Datenbankeinträgen, Drucker- ausgaben, Displayausgaben, Gerätesteuerungen, Ausgabedaten in Dateien. Angaben zu Datenmengen.
5.3	Datenbankanwendung	Beschreibung der logischen Datenbankstruktur (Entity-Relationship-Modell, objektorientiertes Datenmodell, Relationenschemata: Tabellen mit Attributen), Name und Version des verwendeten Datenbankbetriebssystems, Zusammenstellung der Datenbankrecherchen und -einträge.
5.4	Temporäre Dateien	Kurze Beschreibung, falls temporäre Dateien vorhanden sind.
5.5	Datenschutz	Beschreibung der Zugriffsregelungen (Zugriffsberechtigung, Passwortschutz). Maßnahmen zum Manipulationsschutz.
5.6	Datensicherung	Angaben zur Datenarchivierung, zu Sicherungsverfahren (Sperrfristen, Verfallsdaten, Sicherungszyklen, Datenrekonstruktion, Wiederanlaufverfahren).
6	Programmtest	
6.1	Testziele	Durchführung von z.B. Modultest, Integrationstest, Akzeptanztest oder Test auf Normenkonformität. Geforderter Testabdeckungsgrad.
6.2	Testverfahren	Beschreibung der verwendeten Testmethoden und Testwerkzeuge (Testprogramme, Testumgebung).
6.3	Testfälle/Testresultate	Angabe der Testdaten, der erwarteten und erzielten Testergebnisse. Erreichte Abdeckungsgrade. Benchmarks.
7	Programminstallation	

Nr.	Titel	Beschreibungsinhalte
7.1	Gerätebedarf, Hard- und Softwarebedarf	Genaue Beschreibung von Prozessoren, Daten- und Prozessperipherie, Rechnerkopplung, Datenübertragung. Angaben zu Geräteanforderungen, Softwarebedarf.
7.2	Installationsanweisung	
7.3	Besonderheiten	Angaben zu technischen Einschränkungen, Datensicherung, Konfigurierung. Einrichtungs- und Anpassungshinweise.
8	Programmbetrieb	
8.1	Bedienungsanweisungen	Angaben zur <ul style="list-style-type: none"> <li>- Vorbereitung peripherer Geräte,</li> <li>- Programmsteuerung,</li> <li>- Programmüberwachung,</li> <li>- Datensicherung, Archivierung, Passwortschutz.</li> </ul>
8.2	Programmmeldungen	Ursachen und Maßnahmen.
8.3	Unterbrechungsbehandlung	Anweisungen zur geregelten Unterbrechung sowie zur Unterbrechung in Notfällen. Maßnahmen bei Absturz, zur Datensicherung und zum Wiederanlauf.
9	Anwendungsbeispiel	Erläuterung eines Anwendungsfalls mit Beschreibung sämtlicher Programmeingaben/ausgaben.

### Richtlinie für die Softwaredokumentation

Herausgeber: Physikalisch-Technische Bundesanstalt, Braunschweig

Datum: 2005-09-12

Basierend auf:

**ISO/IEC 6592:** 2000, Information Technology - Guidelines for the Documentation of Computer-based Application Systems.

**ISO/IEC FCD 9127:** 2005, Software Engineering – User Documentation and Cover Information for Consumer Software Packages.

**ISO/IEC 18019:** 2004, Software and System Engineering – Guidelines for the Design and Preparation of User Documentation for Application Software.

---

## V Einblick in ein Praxisprojekt

Da die Firma mediaTEXT keine eigenen Projekte im Bereich Software-Dokumentation vorweist, mussten praktische Vorgehensweisen in diesem Bereich an anderer Stelle eingeholt werden. Die Firma CARSTENS + PARTNER GmbH, München, bot die Möglichkeit, ein Gespräch über die praktische Umsetzung der Software-Dokumentation zu führen. Hierzu stand der Technische Leiter Herr Siegfried Siegel zur Verfügung. Ferner wurde die Möglichkeit geboten, in ein laufendes Praxisprojekt Einsicht zu nehmen.

Das XML-basierte Projekt wurde mit FrameMaker 7 umgesetzt. Die Transformation erfolgte mit XSLT. Dabei wurden sämtliche Transformationen und Formatierungen vom Auftraggeber vorgegeben, diese waren nicht ersichtlich. Auch die Buchfunktion im FrameMaker wird nicht benötigt, das wird ebenfalls vom Vorgabeprodukt des Auftraggebers gesteuert. Es werden aus einer XML-Datenquelle eine PDF-Datei und eine nicht kontextsensitive Online-Hilfe erstellt.

Allgemein existiert in der Firma CARSTENS + PARTNER GmbH, München, momentan noch kein DITA-Projekt, da die Konvertierung bestehender Daten meist aufwändig ist. Bei Neuprojekten wird jedoch zu DITA geraten. Im Moment wird gut die Hälfte aller Projekte mit Microsoft Word und WebWorks Publisher umgesetzt.

Alle Projekte werden mit dem Information Mapping-Verfahren umgesetzt. Dazu schult der Betrieb seine Mitarbeiter selbst, CARSTENS + PARTNER GmbH, München, ist dafür zertifiziert. Weiterhin steht eine stark topicorientierte Textproduktion im Vordergrund. Bei den meisten Projekten wird ein Content Management System eingesetzt.

---

## **Gespräch mit Siegfried Siegel, Technischer Leiter der Firma CARSTENS + PARTNER GmbH, München**

Welche Besonderheiten/Unterschiede gibt es bei der Software-Dokumentation?

- keine Feldbeschreibung im Handbuch
- Kontextsensivität der Online-Hilfe
- Masken und Router müssen beschrieben werden
- Verlinkungen zwischen den Texten
- Screenshots nur in der Druckversion

Wie findet Single-Source-Publishing statt?

- PDF und Online-Hilfe werden mit FrameMaker realisiert

Welche Technologien werden eingesetzt?

- XML, XSLT
- DocBook
- oft Information Mapping (bei Neukunden immer), da zertifizierter Betrieb
- Content Management System
- alle Projekte sind Topic-orientiert

Wird DITA verwendet?

- aktuell kein DITA, da Konvertierung bestehender Daten in XML oder DITA aufwändig und kostenintensiv
- bei komplett neuen Projekten würde zu DITA in Verbindung mit IMAP geraten werden

Sonstiges:

- gut über die Hälfte wird mit Word gemacht + WebWorks-Publisher
- Es werden meist Online-Hilfe und Print (PDF) erstellt
- zuerst wird Print-Dokument erzeugt, dann Online-Hilfe
- viele technische Dinge sind bereits vom Kunden vorgegeben

Aussagen (Siegfried Siegel):

Wenn FrameMaker verschwindet, wird nicht InDesign das Ersatzprodukt werden, es wird vielmehr ein anderer XML-Editor den Platz einnehmen.

DITA wird sich durchsetzen.

Praktisch werden die tekcom-Vorträge erst später umgesetzt.

---

## VI IMAP-Beispiel

---

Beispiel „Vorher“: Eine typische geschäftliche Mitteilung

---

### Rundschreiben

Datum: 10. Januar 2007  
An: Alle Angestellten  
Von: Peter Mapper

**Betrifft: NEUORGANISATION**

Wie Sie ohne Zweifel bereits wissen, haben wir letztes Jahr die Auswirkungen verschiedener wirtschaftlicher Veränderungen besonders hart zu spüren bekommen, so unter anderem die Zunahme der Einfuhren aus dem Ausland, Schwierigkeiten mit den Angestelltenverbänden und Probleme im Zusammenhang mit der Entwicklung neuer Baugruppen. Diese Gründe haben dazu geführt, dass unsere Geschäftsleitung eine eingehende Untersuchung der gegenwärtigen Situation in Auftrag gegeben hat mit dem Ziel, mögliche Alternativen zur Verbesserung unserer Gewinnsituation und zur längerfristigen Erhöhung unserer Produktivität aufzuzeigen. Einzelne Abteilungen haben in der letzten Zeit abwechslungsweise mit Gewinn, dann mit Verlust gearbeitet, ohne dass wir uns kritisch mit den tieferen Ursachen auseinandergesetzt haben. Bisher haben wir nur da und dort jeweils Korrekturen vorgenommen, um soweit wie möglich die negativen Auswirkungen abzuschwächen und um Wiederholungen zu vermeiden.

Am 1. April werden zwei neue Abteilungen ins Leben gerufen: Produktionsentwicklung und Betrieb. Im Zusammenhang damit wird auch die Geschäftsleitung neu geordnet. Am 22. März wird die Geschäftsleitung alle betroffenen Mitarbeiter in einer gemeinsamen Betriebsversammlung über die Aufgaben der neuen Abteilungen informieren. Bei dieser Gelegenheit werden auch Einzelheiten zur Neuordnung der Geschäftsleitung bekanntgegeben. Nachfolgend geben wir Ihnen die wichtigsten Änderungen bekannt, damit die verantwortlichen Abteilungsleiter die Vorbereitungsarbeiten an die Hand nehmen können.

Willi Fröhlich verläßt seine jetzige Stelle als Geschäftsführer des Einkaufs und wird Direktor der neuen Betriebsabteilung. Monika Dreher, vormals Vizepräsidentin für Forschung wird Vizepräsidentin der neuen Betriebsabteilung. Walter Fässler wird neu Vizepräsident der Produktentwicklung und verläßt seine Stelle als Direktionsassistent der Forschung. Diese Neuordnung der Geschäftsleitung tritt ebenfalls am 1. April in Kraft.

Heiner Blässer wird zum Direktionsassistenten der Forschung ernannt. Irmgard Huber zur Direktionsassistentin der Betriebsabteilung. In der Regel werden die direkten Mitarbeiter der obengenannten Vorgesetzten weiterhin diesen unterstellt bleiben. Die Geschäftsleitung wird die vom Umzug betroffenen Mitarbeiter am 15. März informieren.

Abb. A-4: IMAP-Beispiel vorher (Quelle: CARSTENS+PARTNER GmbH, München)

## Beispiel „Nachher“: mit der Information Mapping Methode aufbereitet

### Rundschreiben

Datum: 10. Januar 2007  
 An: Alle Angestellten  
 Von: Peter Mapper

#### Schaffung zweier neuer Abteilungen und Neuordnung der Geschäftsleitung

##### Ausgangslage

Wie Sie bereits wissen, haben wir letztes Jahr die Auswirkungen verschiedener wirtschaftlicher Veränderungen besonders hart zu spüren bekommen, so unter anderem

- die Zunahme der Einfuhren aus den Ausland.
- Schwierigkeiten mit den Angestelltenverbänden und
- Probleme im Zusammenhang mit der Entwicklung neuer Baugruppen.

Aus diesen Gründen hat die Geschäftsleitung eine eingehende Untersuchung in Auftrag gegeben, mit dem Ziel, mögliche Alternativen zur Verbesserung unserer Gewinnsituation und zur Erhöhung unserer Produktivität aufzuzeigen. Diese werden nachfolgend erläutert.

##### Zwei neue Abteilungen

Am 1. April werden zwei neue Abteilungen ins Leben gerufen:

- Produktentwicklung
- Betrieb

##### Neuordnung der Geschäftsleitung

Im Zusammenhang damit wird auch die Geschäftsleitung neu geordnet.

Die folgende Neuordnung der Geschäftsleitung tritt ebenfalls am 1. April in Kraft:

Name	Bisherige Position	Neue Funktion
Willi Fröhlich	Abteilungsleiter Einkauf	Direktor Betrieb
Monika Dreher	Vizepräsidentin Forschung	Vizepräsidentin Betrieb
Walter Fässler	Direktionsassistent Forschung	Vizepräsident Produktentwicklung
Heiner Blässer	Direktionsassistent Finanzen	Direktionsassistent Forschung
Irmgard Huber	Abteilungsleiterin Buchhaltung	Direktionsassistentin Betrieb

##### Wichtige Termine

Nachfolgend die wichtigsten Daten im Zusammenhang mit diesen Änderungen:

- 15. März Information der vom Umzug betroffenen Mitarbeiter
- 22. März Bekanntgabe der Einzelheiten der neuen Organisation auf einer Betriebsversammlung
- 1. April Inkrafttreten der neuen Organisation (Neuordnung der Geschäftsleitung)

Abb. A-5: IMAP-Beispiel nachher (Quelle: CARSTENS+PARTNER GmbH, München)

---

## VII Rechercheergebnisse „tekomp“-Artikel

Heft	Artikel
2/1992	Einsatz eines Hypermedia-Tools
2/1992	Struktur der Software-Dokumentation
4/1992	Software-Dokumentation und die Forschung
2/1996	Struktur der Technischen Dokumentation
1/2000	Methoden des Cross Media Publishing
5/2000	Online-Dokumentation strukturiert
2/2001	Online oder Offline
2/2001	Multilinguale Software-Dokumentation
4/2001	Pfadfinder im Datenschlingel
4/2001	Hilfe zur Hilfe
6/2001	Online-Hilfen im Vergleich
4/2002	Vergleich Online-Hilfen
2/2003	Werkzeuge zur Erstellung im Vergleich
3/2003	Online-Hilfe mit FrameMaker
3/2004	Online-Hilfe mit LaTeX
5/2004	Mac-Help
5/2004	Linux-Hilfe
2/2005	Java-Help
4/2005	Wege zur effizienten Software-Dokumentation
2/2006	Vista-Hilfe
5/2006	Internationale Hilfe
2/2008	Single-Source-Publishing
3/2008	Ein Regelwerk für DITA