

Geodätisches Institut

Univ.-Prof. Dr.-Ing. Wilhelm Benning
Templergraben 55
52062 Aachen

Tel.: +049-241-80-95300

Fax: +049-241-8092-142

Email: info@gia.rwth-aachen.de

WWW: <http://www.gia.rwth-aachen.de/>

Materialen zur MA-Ausbildung

Richtlinien zur Softwaredokumentation

Version 0.0

Jürgen A. Lamers

e-mail: jaloma@gia.rwth-aachen.de

17. Februar 2002

Geodätisches Institut der RWTH Aachen – 2002

Richtlinien zur Softwaredokumentation

Dieses Skript ist nur für den internen Gebrauch zur Ausbildung Mathematisch-Technische(r) Assistent(in) genehmigt.

©2002 Geodätisches Institut der RWTH Aachen, Templergraben 55, 52062 Aachen, Germany.

Zusammenfassung

Entwicklung einer Richtlinie zur Dokumentation größerer Software-Projekte.

Inhaltsverzeichnis

1	Einleitung	5
2	Anwendungsbereich	5
3	Inhalt der Richtlinie	6
4	Funktionen und Arten der Softwaredokumentation	6
5	Allgemeine Empfehlungen zur Entwicklung einer Softwaredokumentation	7
5.1	Entwicklungsbegleitende Dokumentation	7
5.2	Benutzerorientierung der Dokumentation	7
5.3	Programmkommentare	8
5.4	Grafische Darstellungsmittel	8
5.5	Dokumentationshilfsmittel	8
5.6	Hinweise zur Gestaltung der Dokumentation	8
6	Inhalt der Programmdokumentation	8
6.1	Inhalt der Programmdokumentation	8
7	Checkliste zur Programmdokumentation	15
7.1	Checkliste zur Programmdokumentation	15

1 Einleitung

Obwohl Computerprogramme und die zugehörige Dokumentation untrennbare Softwarebestandteile sind, wird die Dokumentation im Softwareentwicklungsprozess häufig vernachlässigt. In vielen Fällen fehlt sie vollkommen. Dabei ist eine vollständige, übersichtlich gegliederte, für den jeweiligen Benutzerkreis verständliche und mit der Funktionalität des Programms übereinstimmende Dokumentation eine Grundvoraussetzung für die reibungslose Anwendung und Pflege (Änderung, Erweiterung) von Software. Zu den in der Praxis oft auftretenden Dokumentationsmängeln zählen die folgenden:

- Dokumentationen sind in Fachsprachen verfasst (schwer verständlich, schwer lesbar).
- Dokumentationen halten nicht mit dem Stand der Softwareänderungen mit (fehlende Aktualität, Vollständigkeit und Widerspruchsfreiheit).
- Dokumentationen sind nicht übersichtlich gestaltet (schlechte Gliederung, fehlendes Stichwortverzeichnis).

Die aufgezeigten Dokumentationsmängel können folgenschwere Auswirkungen nach sich ziehen. Dazu gehören z. B.:

- Einarbeitungsschwierigkeiten bei Personalwechsel,
- unnötige Arbeitswiederholungen in Projektfolgephasen,
- Mehraufwand bei der Softwarepflege oder gar ihre Nichtdurchführbarkeit,
- juristische Folgen bei Softwareproblemen (Produkthaftung).

Um die genannten Dokumentationsmängel und ihre Folgen zu vermeiden, ist es notwendig, die Ziele und die Funktion der Softwaredokumentation herauszustellen (siehe Abschnitt 4) und dies in eine Richtlinie für die Erstellung von Programmdokumentationen umzusetzen. Eine Dokumentationsrichtlinie ermöglicht durch die einheitliche Vorgabe von Gliederungen und eine einheitliche Begriffsverwendung die Vollständigkeit und Vergleichbarkeit der Dokumentation.

2 Anwendungsbereich

Die Richtlinie dient der Unterstützung der Qualitätssicherung bei der Softwareentwicklung.

Die Anwendung der Richtlinie bei jeder weiteren Softwareentwicklung empfohlen, wenn

- deren Umfang 200 Quelltextzeilen (ohne Kommentar) überschreitet oder
- deren vorgesehene Anwendungsdauer mindestens 1 Jahr beträgt.

3 Inhalt der Richtlinie

Die Anwendung einer Richtlinie für die Softwaredokumentation trägt zur Vollständigkeit und Vergleichbarkeit der für die Benutzung und Pflege wichtigen Angaben über Programme bei. Erreicht wird dies insbesondere durch die Vorgabe einer Gliederung und einer Checkliste, durch die einheitliche Verwendung von Begriffen sowie durch beispielhafte Darstellungen von Dokumentationsteilen.

Die Gestaltung einer Softwaredokumentation wird durch folgende inhaltliche Schwerpunkte unterstützt:

- Allgemeine Empfehlungen zur Entwicklung einer Softwaredokumentation (Abschnitt 5),
- Inhalt der Programmdokumentation (Abschnitt 6),
- Checkliste zur Auswahl und Wichtung von Dokumentationselementen (Abschnitt 7),

Die angegebenen Gliederungspunkte der Programmdokumentation basieren auf der Norm DIN 66230 (Informationsverarbeitung; Programmdokumentation). Die Dokumentationspunkte wurden für die Dokumentierung wissenschaftlich-technischer Software (z. B. Messwertverarbeitungsprogramme, Simulationsprogramme) angepasst. Die Dokumentation von Betriebssystemen oder Datenbanksystemsoftware ist somit nicht Gegenstand dieser Richtlinie.

4 Funktionen und Arten der Softwaredokumentation

Im wesentlichen soll die Softwaredokumentation die drei folgenden Hauptfunktionen erfüllen:

1. Anleitungsfunktion

- Grundlage für Kauf- und Einsatzentscheidung,
- Bedienungsanleitung,
- Arbeitsanleitung (z. B. zur Fehlerbehandlung, Unterbrechungsbehandlung, Datenarchivierung),
- Arbeitsgrundlage zur Vermeidung von Mehrfachentwicklungen und zur Softwarepflege (Änderungen, Erweiterungen, Fehlersuche),
- kurze Einarbeitung neuer Mitarbeiter, einfache Schulungen.

2. Kontroll- und Nachweisfunktion

- gegenüber internen und externen Prüfern (Produkthaftung, ISO-9001-Zertifizierung),
- Projektfortschrittskontrollen.

3. Kommunikationsfunktion

- einheitliche Kommunikationsbasis
 - für alle an der Softwareentwicklung Beteiligten,
 - für Auftragnehmer/Auftraggeber,
 - für Käufer/Verkäufer,

- Steigerung der Wiederverwendbarkeit der Software,
- Steigerung der Transparenz/Vergleichbarkeit der Programme auf dem Softwaremarkt.

Die unterschiedlichen Dokumentationsfunktionen müssen bei der inhaltlichen Gestaltung der Softwaredokumentation Berücksichtigung finden. Je nach beabsichtigter Verwendung der Dokumentation (und demnach unterschiedlichem Leserkreis) können Dokumentationsteile bzw. einzelne Gliederungspunkte unterschiedlich gewichtet werden. Während die Programmdokumentation technisch orientierte Angaben dokumentiert, die für die Installation, den Betrieb und die Pflege (Änderungen und Erweiterungen) eines Programms notwendig sind, wird in der Anwenderdokumentation auf verständliche Art und Weise die sinnvolle Anwendung des Programms beschrieben.

Zur Softwaredokumentation im weiteren Sinne werden auch die Projektmanagementdokumentation (sie dokumentiert den gesamten Projektablauf inklusive Zeit- und Kostenaspekte) sowie die Programmentwicklungsdokumentation gezählt. Die letztere beschreibt die Ergebnisse der einzelnen Programmentwicklungsschritte im Softwarelebenszyklus (Anforderungsliste, Fachkonzepte, Entwurfsspezifikationen, Programme, Testergebnisse). Die beiden obengenannten prozessorientierten Dokumentarten werden in der Richtlinie nicht betrachtet. Die Richtlinie konzentriert sich auf die Dokumentation des Produkts Software.

5 Allgemeine Empfehlungen zur Entwicklung einer Softwaredokumentation

Im folgenden werden einige Prinzipien der zweckmäßigen und zeitsparenden Dokumentierung erläutert.

5.1 Entwicklungsbegleitende Dokumentation

Jeder Softwareentwicklungs- bzw. Softwarewartungsprozess muss zugleich als Dokumentationsprozess verstanden werden. Durch entwicklungsbegleitende (projektbegleitende) Dokumentation soll arbeitsaufwendige Nachdokumentierung im Anschluss an die Programmerstellung vermieden werden. Basis dafür bildet die Anwendung von Vorgehensmodellen (Phasenmodellen) für die Softwareentwicklung. Dokumentationsteile, die Ergebnisse bzw. Teilergebnisse einzelner Phasen der Programmentwicklung repräsentieren (vgl. Abschnitt 4, Programmentwicklungsdokumentation), werden bzgl. der vorgegebenen Gliederung zusammengestellt und gegebenenfalls um produktspezifische Aspekte ergänzt.

Für Einrichtungen der öffentlichen Verwaltung ist als Phasenmodell der Softwareentwicklung das V(orgehens)-Modell vorgeschrieben. Das 1997 aktualisierte V-Modell ist Bestandteil des für Behörden verbindlichen Entwicklungsstandards für IT-Systeme des Bundes (EStdIT, Allgemeiner Umdruck Nr. 250, 1997).

5.2 Benutzerorientierung der Dokumentation

Die Softwaredokumentation muss zweckgebunden gestaltet und damit jeweils auf verschiedene Benutzerkreise zugeschnitten sein. Eine zweckgebundene Aufteilung der Softwaredokumentation kann gemäß Abschnitt 4 erfolgen. Dort wurde zwischen Programmdokumentation und

Anwenderdokumentation unterschieden. Die im Abschnitt 6 aufgeführten Gliederungspunkte für die Dokumentation müssen entsprechend zugeordnet werden.

5.3 Programmkommentare

Neben der vollständigen Programmdokumentation ist unbedingt eine aussagekräftige Kommentierung im Quelltext des Programms vorzusehen.

5.4 Grafische Darstellungsmittel

Bei der Erstellung der Programmdokumentation ist bei jeder sich bietenden Möglichkeit einer grafischen Darstellung der Vorzug zu geben. Aufrufbäume, Programmablaufpläne bzw. Struktogramme, Datenflusspläne und Dialogablaufpläne erhöhen die Übersichtlichkeit und Verständlichkeit der Dokumentation. Die logische Struktur von Datenbanken lässt sich sinnvoll durch Entity-Relationship-Diagramme abbilden.

5.5 Dokumentationshilfsmittel

Zur Reduzierung des Zeit- und Denkaufwandes wird die Anwendung von Formularblättern eines Textverarbeitungsprogramms, von Literate-Programming-Systemen wie WEB oder Erweiterungen des Programmeditors empfohlen.

5.6 Hinweise zur Gestaltung der Dokumentation

Neben einer geeigneten Gliederung sollte die Dokumentation unbedingt ein Stichwortverzeichnis sowie Querverweise enthalten. Zusammenfassungen und Beispiele erhöhen die Verständlichkeit.

6 Inhalt der Programmdokumentation

Basierend auf DIN 66230 wird eine Gliederung für die Dokumentation wissenschaftlich-technischer Software angegeben. Die einzelnen Spalten der Tabelle enthalten Informationen zur Gliederungsnummer (Nr.), zur Bezeichnung des Gliederungspunktes (Titel) und zur näheren Erläuterung des Gliederungspunktes (Beschreibungsinhalte). Je nach Zweck und Benutzerkreis der zu entwickelnden Dokumentation können Gliederungspunkte weggelassen oder stärker ausgebaut werden. Zum Beispiel muss bei Programmen mit Datenbank Anwendungen der Gliederungspunkt Datenorganisation detaillierter behandelt werden als bei Programmen mit einfachen Ein- bzw. Ausgabedateien.

Gemäß Abschnitt 4 und Punkt 5.2 ist eine Unterscheidung von Programmdokumentation und Anwenderdokumentation vorzunehmen. Eine Anwenderdokumentation lässt sich beispielsweise aus einer Programmdokumentation gewinnen durch Streichen der die Interna des Programms betreffenden Gliederungspunkte Programmaufbau, Programmablauf und Test sowie durch Hinzufügen des Gliederungspunktes Anwendungsbeispiel.

6.1 Inhalt der Programmdokumentation

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 <ul style="list-style-type: none"> • fachlichen Voraussetzungen oder Einschränkungen, • technischen Anwendungsgrenzen (z. B. maximale Antwortzeit), • vertraglichen Informationen (Copyright, Lizenzen, Nutzungsbedingungen, Garantiefragen).

Nächste Seite geht es weiter

Nr.	Titel	Beschreibungsinhalte
1.4.	Gerätebedarf	Angabe der für den Programmablauf nötigen Geräte (Prozessoren, Datenperipherie, Prozessperipherie), Geräteanforderungen, Gerätere striktionen.
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	Genaue 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 Dokumentations- teile (Programmdokumentation, Benutzerhand- buch, Entwicklungsdokumentation, Projektma- nagementdokumentation).
1.9.	Zuständigkeiten	Genaue Angaben zum Programmautor, Verant- wortlichkeiten für Programmpflege, Fehlerbesei- tigung, Vertrieb.
2.	Programmfunktion	
2.1.	Aufgabenstellung	
2.1.1.	Aufgabenbeschreibung	Detaillierte Beschreibung der zu lösenden Aufga- be Überblick, Zusammenhänge, Was soll gelöst werden?)
2.1.2.	Theoretische Grundlagen / Rand- bedingungen / Literaturhinweise	Grobe Beschreibung der angewendeten Model- le / Verfahren (Auf welcher Grundlage wird die Aufgabe gelöst?)
Nächste Seite geht es weiter		

Nr.	Titel	Beschreibungsinhalte
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).
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).

Nächste Seite geht es weiter

Nr.	Titel	Beschreibungsinhalte
4.2.	Datenflussbeschreibung	Darstellung des Datenflusses nach DIN 66001, Methode der Strukturierten Analyse (SA).
4.3.	Dialogablauf	Einfache grafische Darstellung der Folge der Bildschirmdialoge in ihrer Verknüpfung (anwendungsbezogen). Programmablaufplanähnliche genaue Darstellung des Dialogablaufs (Darstellung der Bedingungen, die zu Dialogelementen führen).
4.3.1.	Dialogfolgeplan	
4.3.2.	Dialogablaufplan	
4.4.	Dialogbeschreibung	Detaillierte Beschreibung der einzelnen Bildschirmlayouts.
5.	Datenorganisation	Beschreibung der Datenstrukturen <ul style="list-style-type: none"> • bereits gespeicherter Daten (Dateien mit Datenfeldern, Tabellen mit Attributen, Klassen mit Instanzenvariablen), • zu erfassender Daten (über Tastatur, Eingabegeräte). • Angaben zu Zugriffsverfahren, Standard-Initialisierungen, zulässigen Eingabebereichen und maximal zu verarbeitenden Datenmengen.
5.1.	Eingabedaten	
		Nächste Seite geht es weiter

Nr.	Titel	Beschreibungsinhalte
5.2.	Ausgabedaten	Beschreibung von Datenbankeinträgen, Drucker- ausgaben, Displayausgaben, Gerätesteuern, Ausgabedaten in Dateien. Angaben zu Daten- mengen.
5.3.	Datenbankanwendung	Beschreibung der logischen Datenbankstruktur (Entity-Relationship-Modell, objektorientiertes Datenmodell, Relationenschemata: Tabellen mit Attributen), Name und Version des verwen- deten Datenbankbetriebssystems, Zusammenstel- lung der Datenbankrecherchen und -einträge.
5.4.	Temporäre Dateien	Kurze Beschreibung, falls temporäre Dateien vorhanden sind.
5.5.	Datenschutz	Beschreibung der Zugriffsregelungen (Zugriffsbe- rechtigung, Passwortschutz). Maßnahmen zum Manipulationsschutz.
5.6.	Datensicherung	Angaben zur Datenarchivierung, zu Sicherungs- verfahren (Sperrfristen, Verfallsdaten, Siche- rungszyklen, Datenrekonstruktion, Wiederan- laufverfahren).
6.	Programmtest	
6.1.	Testziele	Durchführung von z. B. Modultest, Integrations- test, Akzeptanztest oder Test auf Normenkon- formität. Geforderter Testabdeckungsgrad.
6.2.	Testverfahren	Beschreibung der verwendeten Testmethoden und Testwerkzeuge (Testprogramme, Testumge- bung).
6.3.	Testfälle/Testresultate	Angabe der Testdaten, der erwarteten und erziel- ten Testergebnisse. Erreichte Abdeckungsgrade. Benchmarks.

Nächste Seite geht es weiter

Nr.	Titel	Beschreibungsinhalte
7.	Programminstallation	
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"> • Vorbereitung peripherer Geräte, • Programmsteuerung, • Programmüberwachung, • Datensicherung, Archivierung, Passwortschutz.
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.

Nächste Seite geht es weiter

Nr.	Titel	Beschreibungsinhalte
-----	-------	----------------------

7 Checkliste zur Programmdokumentation

Die vorgeschlagene Checkliste dient

- der Auswahl und Wichtung von Gliederungspunkten der Programmdokumentation,
- der Überprüfung der Fertigstellung von einzelnen Gliederungspunkten,
- dem Festhalten des Fertigstellungstermins,
- der Angabe von Bemerkungen und Hinweisen zum Dokumentationsprozess sowie
- der Angabe von Verweisen z. B. auf schon fertiggestellte Dokumentationsteile.

Die ersten beiden Spalten der nachfolgenden Tabelle stellen sämtliche Gliederungspunkte der Programmdokumentation gemäß Abschnitt 6 zusammen (Gliederungsnummer, Dokumentationspunkt). Die restlichen drei Spalten enthalten

- Angaben zur Auswahl und Wichtung von Gliederungspunkten,
- den Fertigstellungstermin des entsprechenden Dokumentationspunkts sowie
- Bemerkungen und Verweise.

7.1 Checkliste zur Programmdokumentation

Nr.	Dokumentationspunkt	Erforderlich? Wichtung	Datum der Fertigstellung	Bemerkungen Verweise
1.	Programmkenndaten			
1.1.	Programmidentifizierung			
1.1.1.	Programmname			
1.1.2.	Systemzuordnung			
1.1.3.	Variantenbezeichnung			
1.1.4.	Aktuelle Version			
1.1.5.	Freigabedatum			
1.2.	Deskriptoren			
1.3.	Programmkurzbeschreibung			
1.3.1.	Programmaufgabe			
1.3.2.	Programminhalt			
1.3.3.	Vorschriften			
1.3.4.	Besonderheiten			
1.4.	Gerätebedarf			

Read the next side

Nr.	Dokumentationspunkt	Erforderlich? Wichtung	Datum der Fertigstellung	Bemerkungen Verweise
1.5.	Programmgröße			
1.6.	Programmbedarf			
1.6.1.	Betriebssysteme			
1.6.2.	Programmiersprachen/ Compiler			
1.6.3.	Sonstige Programme			
1.7.	Datenorganisation			
1.8.	Dokumentationsumfang			
1.9.	Zuständigkeiten			
2.	Programmfunktion			
2.1.	Aufgabenstellung			
2.1.1.	Aufgabenbeschreibung			
2.1.2.	Theoretische Grundlagen			
2.1.3.	Maßeinheiten			
2.1.4.	Vorschriften			
2.2.	Aufgabenlösung			
2.2.1.	Funktionshierarchie			
2.2.2.	Methoden/Algorithmen			
2.2.3.	Fehlerbehandlung			
3.	Programmaufbau			
3.1.	Programmbausteine			
3.2.	Programmstruktur			
3.3.	Quelltext			
3.4.	Kompilieren und Linken			
4.	Programmablauf			
4.1.	Programmablaufbeschrei- bung			
4.2.	Datenflussbeschreibung			
4.3.	Dialogablauf			
4.3.1.	Dialogfolgeplan			
4.3.2.	Dialogablaufplan			
4.4.	Dialogbeschreibung			
5.	Datenorganisation			
5.1.	Eingabedaten			
5.2.	Ausgabedaten			
5.3.	Datenbankanwendung			
5.4.	Temporäre Dateien			

Read the next side

Nr.	Dokumentationspunkt	Erforderlich? Wichtung	Datum der Fertigstellung	Bemerkungen Verweise
5.5.	Datenschutz			
5.6.	Datensicherung			
6.	Programmtest			
6.1.	Testziele			
6.2.	Testverfahren			
6.3.	Testfälle/Testresultate			
7.	Programminstallation			
7.1.	Gerätebedarf, Hard- und Softwarebedarf			
7.2.	Installationsanweisung			
7.3.	Besonderheiten			
8.	Programmbetrieb			
8.1.	Bedienungsanweisungen			
8.2.	Programmmeldungen			
8.3.	Unterbrechungsbehandlung			
9.	Anwendungsbeispiel			