Programmentwurf und Programmdokumentation

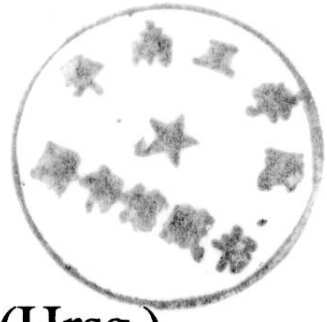
Methoden und Techniken bei der Prozeßdatenverarbeitung

VDI-Verlag

Digitaltechnik
Herausgegeben von Prof. Dr.-Ing. Wolfgang Weber

Programmentwurf und Programmdokumentation

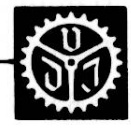
Methoden und Techniken bei der Prozeßdatenverarbeitung



Verein Deutscher Ingenieure
VDI/VDE-Gesellschaft
Meß- und Regelungstechnik (Hrsg.)

Dipl.-Ing. Helmut Nettesheim (Bearbeiter)

VDI-Verlag GmbH
Verlag des Vereins Deutscher Ingenieure · Düsseldorf



CIP-Kurztitelaufnahme der Deutschen Bibliothek

Programmentwurf und Programmdokumentation:
Methoden u. Techniken bei d. Prozeßdatenverarb./
Verein Dt. Ingenieure; VDI/VDE-Ges. Meß u.
Regelungstechnik (Hrsg.). Helmut Nettesheim
(Bearb.). – Düsseldorf: VDI-Verlag, 1982.
(Digitaltechnik)
ISBN 3-18-400524-0

NE: Nettesheim, Helmut [Bearb.]; Verein Deutscher
Ingenieure

Printed in Germany

ISBN 3-18-400524-0

Programmentwurf und Programmdokumentation

Vorwort

In allen Bereichen der Prozeßdatenverarbeitung erhält das ingenieurmäßige Planen und Erstellen von Prozeßrechnerprogrammen – häufig als Software-Engineering bezeichnet – zunehmende Bedeutung. Gleichzeitig besteht der Wunsch nach besserer und sicherer Software.

Der wachsende Umfang und die Komplexität der Programme sowie die ständig steigenden Kosten bei der Software-Produktion erfordern geeignete Hilfsmittel und Werkzeuge, um die Software-Qualität zu verbessern und den Herstellungsaufwand zu verringern. Die Bedeutung dieser Hilfsmittel zur Software-Erstellung ist unbestritten.

Eine Forderung seitens des Software-Engineering sind die Anwendung von Software-Entwurfsmethoden und die Erstellung einer projektbegleitenden Programmdokumentation. Es ist notwendig, den gesamten Ablauf der Programmerstellung – beginnend bei dem ersten Vorentwurf bis zur Inbetriebnahme der fertigen Programme, einschließlich deren Wartung und Pflege mit solchen Methoden und Techniken – aktiv zu unterstützen. Diesbezüglich gibt es eine Reihe guter Ansätze: Eine Anzahl von Programmentwurfs- und Programmdokumentations-Methoden mit unterschiedlichen Eignungsprofilen und Anwendungszielen ist bereits vorhanden. Die Praxis zeigt jedoch, daß diese Methoden und Techniken häufig nur unzureichend und nicht konsequent genug im gesamten Ablauf der Software-Erstellung angewendet werden.

Um einen Anreiz zu schaffen, diesen Mangel zu beseitigen, wurden praxiserprobte Programmentwurfs- und Dokumentations-Methoden ermittelt. Aus dieser Zusammenstellung resultiert die vorliegende vergleichende Darstellung von zehn ausgewählten Methoden, die in den Aufgabenbereichen der Prozeßrechnerprogrammierung einsetzbar sind.

Diese unmittelbar auf die Praxis zugeschnittenen Methoden können auch dem kaufmännisch orientierten EDV-Fachmann von Nutzen sein.

Die (in besonderen Anwendungsfällen der Prozeßdatenverarbeitung auch untereinander kombinierbaren) methodischen Hilfsmittel lassen sich in

- Entwurfsmethoden,
- geschlossene Methoden und
- kombinierte Methoden

einteilen. Alle Einzelbeschreibungen sind einheitlich gegliedert, d.h. jede Dokumentationsmethode entspricht einem geschlossenen Abschnitt, der im großen und ganzen jeweils aus folgenden Unterabschnitten aufgebaut ist:

- Überblick: Phasen der Projektabwicklung/Dokumente,
- Verzeichnis verwendeter Spezialbegriffe mit Erläuterungen,
- 1. Einleitung, Gesamtüberblick, Einsatzerfahrungen,
- 2. Phasenabdeckung, Dokumente je Phase,
- 3. Darstellungselemente,
- 4. Regeln und Richtlinien,
- 5. Hilfsmittel und Automatisierbarkeit,
- 6. Anwendungsbeispiel.

Charakteristisch für jede beschriebene Methode ist die Projektphasenübersicht: Dem Praktiker gibt diese Darstellung sofort Auskunft z. B. darüber, ob eine Methode durchgängig in allen Phasen der Projektabwicklung anwendbar ist, für welche Phasen der Projektabwicklung die Methode Dokumente liefert, oder ob eine bestimmte Methode bevorzugt in den Anfangsphasen der Systemplanung Anwendung findet.

Das Anwendungsbeispiel, das jeweils den Abschluß eines Hauptabschnittes bildet, hat die Form einer gleichlautenden Aufgabenstellung „Labormeßwerterfassung". Dadurch wird ein wesentlicher Beitrag zur Übersichtsinformation, besonders bei vergleichender Betrachtungsweise, vermittelt.

Das Buch soll eine Hilfestellung geben, die dem Management Entscheidungen erleichtert, dem Projektmanager das Wesentliche dieser Methoden vermittelt und den Programmierer über den Aufbau und die Eigenheiten der Methoden informiert.

Den Mitarbeitern aus dem Ausschuß Programmdokumentation des Bereiches Technik der Prozeßrechner in der VDI/VDE-Gesellschaft Meß- und Regelungstechnik (GMR), den Herren

R. Anger, Nürnberg,

R. Emge, Frankfurt,

W. Fischer, Frankfurt,

H. W. Hampe, Frankfurt-Höchst,

K. Maurer, Ludwigshafen,

H. Nettesheim, Dormagen,

H. Steinbacher, München,

E. Traue, Erlangen,

K. Zürcher, Basel,

gilt unser Dank für die Abfassung der einzelnen Abschnitte. Besonders zu danken ist dem Obmann dieses Ausschusses, Herrn Dipl.-Ing. *Helmut Nettesheim,* der mit viel Sorgfalt und Mühe die redaktionelle Bearbeitung und die einheitliche Darstellung der verschiedenen Abschnitte übernommen hat.

Düsseldorf, Juni 1982

Verein Deutscher Ingenieure
VDI/VDE-Gesellschaft
Meß- und Regelungstechnik

Inhalt

1. Einführung

1.1. Bedeutung der Software

Die ersten elektronischen Datenverarbeitungsmaschinen wurden in der Bundesrepublik Deutschland Anfang der sechziger Jahre in der Industrie erfolgreich eingesetzt. Man erkannte, daß sich besondere elektronische Datenverarbeitungsanlagen, die Prozeßrechner, als hervorragende Automatisierungsmittel z. B. im Bereich der Verfahrenstechnik bei der Erfüllung von meß-, steuer- und regelungstechnischen Aufgaben anbieten und ausgezeichnet eignen.

Solange die zu lösenden Aufgaben und Probleme anzahlmäßig gering waren, oder der Einsatz eines Computers zunächst als Prestigesymbol galt, waren die Erfassung und Rechtfertigung der Kosten und Aufwendungen für solche Anlagen von geringem Interesse. Erst mit dem starken Vordringen der Rechner in nahezu alle Bereiche der Forschung und Verwaltung sowie in die Industrie, die zunehmend rechnergestützte Automatisierungskonzepte einführte, nahm die Forderung nach einem kostengünstigen Einsatz der Prozeßrechner an Bedeutung zu. Eine spürbare Kostenminderung als Folge von technologischen Durchbrüchen bei der Herstellung von Bauteilen für Rechner kam jedoch oft nicht in dem gewünschten Maß zur Geltung. Hohe Kosten für die Software-Erstellung, die Fehlerhaftigkeit der Programme sowie Verzögerungen in der Projektabwicklung durch unzuverlässige Terminangaben waren das Ergebnis einer überforderten Projektorganisation. Somit ist die Erstellung zuverlässiger Software nicht nur zu einem technischen, sondern auch zu einem organisatorischen Problem geworden.

Oft sind die Gründe für organisatorische Versagen darin zu suchen, daß die traditionellen Werkzeuge des Managements nur mit Schwierigkeiten bei Software-Projekten eingesetzt werden können. Zum andern haben die rasch fortschreitende Ausweitung des Computereinsatzes und die stetige Zunahme der Komplexität der Probleme dazu geführt, daß die im Laufe der Zeit entwickelte „Kunst" der Programmierer ihre Grenze erreicht hat.

Man ist zu der Erkenntnis gekommen, daß kostengünstige Hardware, große Speicherkapazitäten und schnelle Verarbeitungszeiten allein noch keine

Garantie für leistungsfähige und zuverlässige Rechnersysteme geben. Auf Grund dieser Tatsache bekam der wachsende Problemkreis Software einen neuen Stellenwert: Unter dem Aspekt des betriebswirtschaftlichen Nutzens sind Computer nur so gut wie ihre Software; ohne Software ist ein Digitalrechner wertlos. Berücksichtigt man ferner noch die indirekten Kosten für die Fehlerbehebung mangelhafter Programme und die dadurch bedingten Stillstandzeiten der Anlagen, so wird die Bedeutung einer präzise erstellten und zuverlässigen Software noch unterstrichen.

1.2. Verbesserung der Software-Qualität

Bei der Lösung der vorgenannten Probleme ist die Suche nach einem Weg zu Verbesserung der Software-Produktivität und -Qualität am dringlichsten. Damit verbunden ist die Forderung nach verstärkten Forschungs- und Entwicklungsbemühungen zur Schaffung neuer Hilfsmittel oder zur Verbesserung der heute verfügbaren Hilfsmittel und Methoden der Software-Erstellung. Dazu gehört auch die Normung der Grundlagen. Nicht die Klärung von Begriffen, wie z. B. „Multilevel-Prioritäts-Interrupt", unterstützt dieses Bestreben, sondern man muß Voraussetzungen schaffen, so daß sich ein Programmentwickler im Wirrwarr der Möglichkeiten zurechtfindet, und die für seine zu lösende Aufgabe am zweckmäßigsten erscheinende Methode auswählen kann.

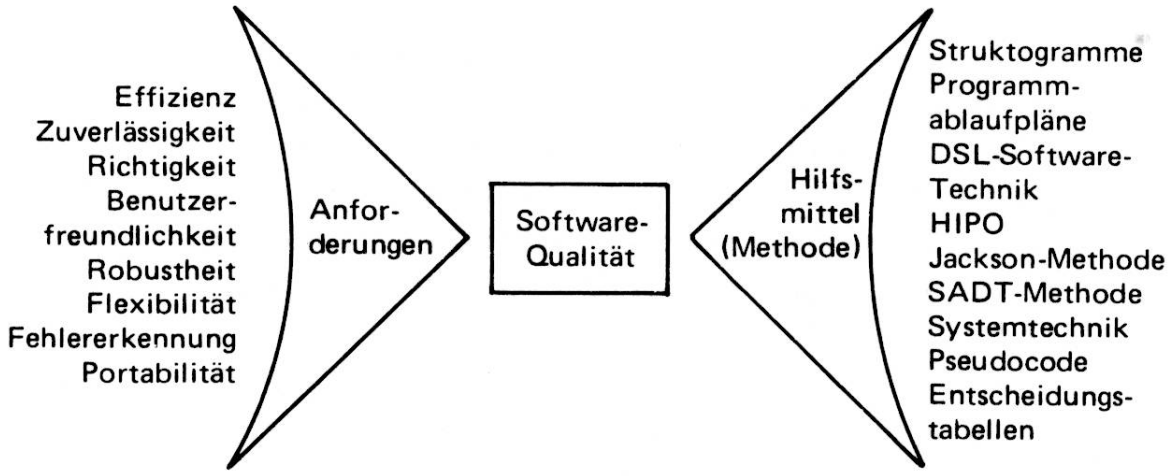


Bild 1-1. Software-Qualität: Anforderungen und Hilfsmittel.

Software-Qualität muß höchsten Ansprüchen genügen. Bild 1-1 zeigt die grundsätzlichen Zusammenhänge. Wichtige Anforderungen, die an die Software gestellt werden, sind Effizienz, Zuverlässigkeit, Richtigkeit, Benutzerfreundlichkeit, Robustheit, Flexibilität, Fehlererkennung, Portabilität.

Die Software-Qualität läßt sich mit Hilfe von Entwurfs- und Dokumentationsmethoden deutlich verbessern. Zu diesen Methoden gehören

- Entwurfsmethoden wie
  - Structured Analysis and Design Technique (SADT),
  - datenorientierte Software-Entwurfsmethode nach *Jackson,*
  - DSL-Software-Technik,
- geschlossene Methoden wie
  - Struktogramme,
  - Pseudocode,
  - Programmablaufpläne, Datenflußpläne,
  - Entscheidungstabellen,
- kombinierte Methoden wie
  - Hierarchy plus Input-Process-Output (HIPO),
  - modulare Projekt- und Programmstrukturierung und projektbegleitende Dokumentation,
  - Systemtechnik.

Für die Ausführung eines Projekts ist ein Plan unentbehrlich. Aus diesem soll einerseits hervorgehen, wie die Abwicklung des Vorhabens in organisatorischer Hinsicht zu erfolgen hat. Andererseits soll er ein Dokument sein, das die Zusammenhänge und Funktionen möglichst vollständig und widerspruchsfrei beschreibt. Der Plan dient somit dem Ziel, die Struktur und die wesentlichen Merkmale des Projekts in einer allgemein verständlichen Form – also auch lesbar für andere – darzustellen.

Für das Verständnis der Komplexität ist eine Untergliederung in überblickbare Teile unumgänglich. Bei dieser Gliederung werden bereits die Weichen gestellt, die richtungsweisend den einzuschlagenden Weg für die Programmerstellung angeben. Eine positive Beeinflussung auf die Programmerstellung ist bereits an dieser Stelle möglich. Wenn es gelingt, schon in der Entwurfsphase eines Projekts eine Methodik anzuwenden, die mit den vielfältigen Randbedingungen des Umfeldes in Einklang gebracht werden kann, so ist dieser Weg die wirkungsvollste Alternative zur Unterstützung der Software-Erstellung.

Dies erfordert jedoch unter Umständen ein Umdenken in der Projektorganisation; denn oftmals ist das Erstellen einer Programmdokumentation eine unliebsame und zuweilen auch unnütze Tätigkeit, solange die Programmierer gezwungen werden, Aktenordner mit Papier zu füllen, um

gerade der Pflicht des Dokumentierens genüge zu leisten. Mit Recht ist eine solche unübersichtliche Ansammlung von Papier kein Beitrag zur Problemlösung.

Diese unbefriedigende Situation kann durch die Anwendung einer geeigneten Software-Dokumentationstechnik gemeistert werden. Dennoch sollte man berücksichtigen, daß weder die Erfahrung noch die Kreativität eines Technikers durch irgendeine Methode oder ein Hilfsmittel ersetzt werden können. Man kann jedoch erwarten, daß die schöpferische Arbeit der Programmentwicklung durch diese unterstützt und gefördert wird. Die Anwendung von starren Vorschriften und Formalismen würde dem notwendigen Spielraum der Praktikabilität zuwider laufen. Um eine möglichst große Abdeckung aller Projektphasen zu erreichen, kann auch eine zweckmäßige Kombination von Hilfsmitteln und Methoden zum Ziel führen.

1.3. Projektbegleitende Programmdokumentation

Aus den nachfolgenden Abschnitten sollen die Wirksamkeit und der Nutzen einer projektbegleitenden Programmdokumentation hervorgehen. Mit Dokumentieren sei eine Tätigkeit verstanden, die den Informationsgewinn während der gesamten Projektabwicklung in einer wiederverwendbaren Form beschreibt und darstellt. Dabei wird vorausgesetzt, daß sich die erstellten Unterlagen an den Anforderungen des jeweiligen Adressaten orientieren. Darüber hinaus wird von diesen Dokumenten verlangt, daß sie die Kommunikation im Kreis der Benutzer untereinander unterstützen und fördern. Deshalb ist an dieser Stelle nicht von einer bestimmten Dokumentation die Rede, sondern von einer Anzahl von Dokumenten, deren Aussagekraft und Terminologie für eine Zielgruppe von Anwendern bestimmt ist.

Da es sich um eine Auswahl einer Anzahl von Entwurfsmethoden und Dokumentationstechniken handelt, wird nicht die Meinung vertreten, daß das Wissen über Software-Methodik als eine reine Lehre vermittelt werden kann. Auch kann eine bestehende Verunsicherung nicht beseitigt werden, wenn der Anwender glaubt, von der Disziplin Software-Engineering zu erfahren, welche Methode die Probleme seiner spezifischen Aufgabenstellung vollständig abdecken kann. Der Erfolg des jeweiligen Ansatzes im Programmentwicklungsprozeß ist unterschiedlich und muß entsprechend den oben erwähnten Randbedingungen relativiert werden. Somit findet der suchende Anwender im folgenden keine Darstellung einer neuen Allzweckmethode, weder eine Richtlinie darüber, was im einzelnen zu dokumentie-

ren ist, noch eine Bewertung heute bekannter und praktizierter Methoden. Es kann ihm jedoch eine Hilfestellung geboten werden, eine Antwort zu finden auf Fragestellungen, wie z. B. „Was steckt hinter den Schlagworten HIPO, DSL, SADT?" oder „Wie kann man im Durcheinander der postulierten Methoden und Hilfsmittel Ordnung schaffen?"

1.4. Phasen der Projektabwicklung

Ein Programmsystem wird in mehreren Stufen entwickelt. Diese Entwicklung läßt sich in einzelne Phasen der Projektabwicklung einteilen. In jeder Phase entstehen Informationen, die gesammelt, aufbereitet und geordnet die Gesamtheit der Programmdokumentation darstellen.

Die Erstellung und die Handhabung einer projektbegleitenden Dokumentation können erleichtert werden, wenn ein funktioneller Zusammenhang der Dokumente zu den einzelnen Phasen einer Projektabwicklung sichtbar gemacht wird. Für diesen Vorgang bildet zunächst ein allgemeines Phasenmodell gemäß Tabelle 1-1 die Grundlage, das für Entwurfs- und Dokumentationsmethoden insgesamt oder abschnittweise gilt. Im konkreten Fall kann dem Wunsch, den gebotenen Spielraum bezüglich Freizügigkeit und Varianz zu erweitern, stets entsprochen werden.

Im starren Phasenmodell sind die fortschreitenden Projektstufen Projektgenehmigung, Projektanalyse, Spezifikation, Grobentwurf, Feinentwurf, Codierung/Test, Integration/Freigabe/Übergabe, Betrieb/Produktpflege/Wartung/Erweiterung definiert und den dabei abschnittweise anfallenden Dokumenten gegenübergestellt. Dies ist die Sollvorstellung für alle Phasen der Projektabwicklung; daraus läßt sich die Gesamtentwicklung innerhalb eines Projektes ablesen. Die Entwurfs- und Dokumentationsmethoden decken in der Regel nicht alle Phasen eines Projektes ab. Dies wird bei den angepaßten Phasenmodellen in den einzelnen Abschnitten näher erläutert.

1.5. Wie sind Programme zu dokumentieren?

In Normen und Richtlinien ist festgelegt, was zu dokumentieren ist (Umfang und Inhalt der Software-Dokumentation). Es bedarf jedoch zusätzlich einer Anleitung, die beschreibt, wie zu dokumentieren ist, d. h. wie die vielfältig verfügbaren Methoden und Techniken der Software-Dokumentation angewendet werden sollen. Jeder Methode und jeder Technik ist eine eigene Philosophie zugeordnet. Daraus entsteht die Gefahr, sich in bezug auf ein konkretes Projekt für die ungeeignete („falsche") Methode bzw. für das „falsche" Dokumentationsverfahren zu entscheiden. Die Praxis zeigt in

Tabelle 1-1. Allgemeines Phasenmodell und Dokumente.

Allgemeines Phasenmodell	Dokumente je Projektphase (Inhalt)
• Projektgemehmigung	
• Projektanalyse Erhebung u. Bewertung aller Fakten, die für Inhalt u. Auslegung eines Verfahrens notwendig sind.	– Daten Istaufnahme – Fachl. Sollkonzept/ Zielsetzung – Beschreibung Produktionsablauf – Realisierungsvorstellung – Optimierung/ Restriktion
• Spezifikation Der Plan für d. Projekt wird erstellt u. d. techn. Problem d. Aufgabenstellung wird definiert. Es wird geklärt, was im Projekt durchgeführt werden soll, nicht wie es zu realisieren ist. Planer, Entwickler u. Anwender stimmen in der Definition des techn. Problems überein und überein und überprüfen die Durchführbarkeit.	– Definition der Aufgabenstellung, Aufgabenbeschreibung – Art/Menge Daten (Signale) – DV-Lösungskonzept (Prüfung der Realisierbarkeit) – Schnittstellen: Prozeß-Mensch „System"-Mensch – HW-Konzepte (Periferie)
• Grobentwurf Lösungsentwürfe der durch die Spezifikation beschriebenen Aufgaben werden erarbeitet. Eine oder mehrere annehmbare Lösungen d. Probleme werden diskutiert. Der endgültige Lösungsentwurf enthält d. Hardware- u. Software-Konfiguration, Modulaufteilung, Taskstrukturierung, Schnittstellendefinitionen.	– DV-Lösungsverfahren, Ansatz • Programmstruktur, Aufbau, Ablauf • Datenstruktur, Ordnungskriterien – HW-Konfiguration
• Feinentwurf Die Codiervorlagen werden endgültig ausgearbeitet. Sie enthalten die Ablauf-, Aufruf- u. Datenstrukturen. Die Bedingungen f. d. Programmtests werden spezifiziert.	– Verfeinertes Grobkonzept bis Codierfähigkeit – Teststrategie

Tabelle 1-1 (Fortsetzung).

Allgemeines Phasenmodell	Dokumente je Projektphase (Inhalt)
• Codierung/Test Ablauffähige Programme werden erstellt u. einer Serie von Tests unter simulierten Betriebsbedingungen unterzogen.	– Listings der Programme – Testergebnisse
• Integration/Freigabe/Übergabe Das Gesamtsystem (Hardware u. Software) wird zusammengestellt u. eine Serie von Tests unter praxisnahen Betriebsbedingungen wird durchgeführt. Das System wird d. Anwender z. Abnahme freigegeben. Danach erfolgt d. Übergabe a. d. Anwender u. U. mit Test in d. echten Betriebsumgebung.	– Berichte – Beschreibungen
• Betrieb/Produktpflege/Wartung/Erweiterung Die abgenommenen Programme laufen in ihrer eigentlichen Betriebsumgebung. Notwendige Produktpflege, Wartung u. Erweiterungen werden durchgeführt.	– Wartungshandbuch

der Vergangenheit leider häufig, daß Entwurfs- und Dokumentationsverfahren auf halbem Wege verworfen werden und das Projekt nach „Bastlerart" fortgesetzt wird.

Angeregt durch diese Eindrücke beschloß im Jahre 1977 ein VDI/GMR-Arbeitskreis, die Probleme beim Entwurf und bei der Dokumentation von Software in der Prozeßdatenverarbeitung zu beleuchten. Er stellte sich die Aufgabe, die vorhandenen zahlreichen Programmentwurfs- und Dokumentationsmethoden zu beurteilen, zu analysieren, zu ordnen, ein repräsentatives Spektrum auszuwählen und in geeigneter Form darzustellen. Aus den bekannten Programmentwurfs- und Dokumentationsmethoden mit unterschiedlichen Eignungsprofilen und Anwendungszielen wurden nach sorgfältiger kritischer Diskussion zehn Methoden ausgewählt.

Außer den vielen Bewertungsmerkmalen, die jede Methode inhaltlich auf verschiedenste Weise erfüllt, wurden die Auswahlkriterien auf drei Aspekte konzentriert, um die Methoden einzustufen: