
Inhalt

Vorwort der Herausgeber	9
Vorwort des Autors	11
1. Einführung	13
1.1 Programmieren als Ingenieurtätigkeit	13
1.2 Grundbegriffe des Software-Entwurfs	14
1.3 Chronologische Entwicklung des computerunterstützten Software-Entwurfs	17
2. Prinzipien des Software-Entwurfes	21
2.1 Das Prinzip der strukturierten Programmierung	21
2.2 Das Prinzip der schrittweisen Verfeinerung (Abstraktionsstufenmodell)	28
2.3 Weitere Strukturierungsprinzipien	31
2.4 Funktioneller Entwurf und Funktionsbaum	34
2.5 Geheimnisprinzip und Programmmodularisierung	44
2.6 Entity-Relationship-Modell als Entwurfsprinzip	51
2.7 Die Bedeutung der Software-Ergonomie für den Software-Entwurfsprozeß	54
3. Software-Entwurfsmethoden	59
3.1 Die SADT-Methode	59
3.2 Die Constantine-Methode	63
3.2.1 Der Modulbegriff bei Constantine	63
3.2.2 Die Entwurfsmethode	72
3.2.3 Der Entwurf von Dialogprogrammen	77
3.2.4 Zusammenfassung, Vorteile und Nachteile der Constantine-Methode	80
3.3 Die Weiterentwicklung der Constantine-Methode	82
3.3.1 Structured Analysis (SA)	82
3.3.2 Die Entwurfsmethoden LITOS und ANIMOS	88
3.4 Die HIPO-Methode	91

3.5	Die Jackson-Methoden	99
3.5.1	Die JSP-Methode	99
3.5.1.1	Der elementare Teil	99
3.5.1.2	Strukturkonflikte und Programminversion	108
3.5.1.3	Die Backtracking-Technik	117
3.5.1.4	Der Entwurf von Dialogprogrammen in JSP	122
3.5.1.5	Zusammenfassung, Vorteile und Nachteile der JSP-Methode	124
3.5.2	Die JSD-Methode	125
3.5.2.1	Der Ansatz	125
3.5.2.2	Ein Beispiel	127
3.5.2.3	Zusammenfassung, Vorteile und Nachteile der JSD-Methode	136
3.6	Die LCP-Methode von Warnier und der Einsatz von Entscheidungstabellen im Software-Entwurf	137
3.6.1	Der Ansatz von Warnier	137
3.6.2	Entscheidungstabellen als Hilfsmittel des Software-Entwurfs und der LCP-Methode	141
3.6.3	Zusammenfassung, Vorteile und Nachteile der LCP-Methode	147
3.7	Der Einsatz von Petri-Netzen im Software-Entwurf, dargestellt am Beispiel der ISAC-Methode	149
3.7.1	Grundlagen der Petri-Netze	149
3.7.2	Die ISAC-Methode	153
3.7.3	Zusammenfassung, Vorteile und Nachteile der ISAC-Methode	158
3.8	Weitere Methoden	160
3.8.1	Stand der Methodenentwicklung	160
3.8.2	Attributierte Grammatiken	160
3.8.3	Das Bertelsmann-Modell Software Engineering (BAG-SE)	161
3.8.4	Endliche Automaten (finite state machines: FSM)	163
3.8.5	Normierte Programmierung	163
3.8.6	Das Salzgitter-Software-Engineering-Modell	166
3.8.7	SEBIS (System zum Entwurf betrieblicher Informationssysteme)	168
3.8.8	Der IBM Software Engineering Workshop (SEW)	169
3.9	Der Einfluß von Sprachen der 4. Generation auf den Software-Entwurfsprozeß	170

4.	Software Engineering-Werkzeuge (SE-Werkzeuge)	177
4.1	Klassifizierung und Anforderungen	177
4.2	Typen von SE-Werkzeugen	180
4.3	CAS-Gesamtsysteme und ihre Architektur	181
5.	Vergleich und Bewertung von Software-Entwurfsverfahren	189
5.1	Methodenvergleich	189
5.2	Werkzeugvergleich	195
6.	Ausblick	201

Anhang

1.	Literatur	205
2.	Register	223