

Einführung in Software Engineering

von

Reinhold Kimm

Wilfried Koch

Werner Simonsmeier

Friedrich Tontsch



Walter de Gruyter · Berlin · New York 1979

Inhaltsverzeichnis

1. <u>Gegenstandsbestimmung</u>	11
1.1. Historische Entwicklung	12
1.2. Was ist Software Engineering ?	15
1.3. Die Phasen der Softwareentwicklung und -verwendung	18
1.3.1. Problemanalyse	20
1.3.2. Entwurf	21
1.3.3. Implementierung	22
1.3.4. Funktions- und Leistungsüberprüfung	24
1.3.5. Installation und Abnahme	25
1.3.6. "Wartung"	25
1.3.7. Rückgriffe im Entwicklungsprozeß	26
1.3.8. Projektorganisation und Dokumentation	27
1.4. Zusammenfassung	28
2. <u>Zielbestimmung</u>	29
2.1. Softwareproduktion wozu ?	29
2.2. Anforderungen an den Softwareherstellungsprozeß	31
2.3. Anforderungen an das Produkt	32
2.3.1. Zuverlässigkeit	33
2.3.2. Benutzungsfreundlichkeit	36
2.3.3. Flexibilität	39
2.3.4. Lesbarkeit	43
2.3.5. Effizienz	43
2.4. Zusammenhang der Qualitätsanforderungen	44
2.5. Zusammenfassung	47
3. <u>Problemanalyse</u>	48
3.1. Istanalyse	49
3.2. Erarbeitung eines Sollkonzepts	52
3.3. Darstellungsmethoden	54
3.3.1. Gebräuchliche Darstellungsmittel	55
3.3.2. Die SADT-Methode	59
3.3.3. Das ISDOS-Projekt	64
3.4. Durchführbarkeitsstudie	68

3.5. Projektplanung	70
3.5.1. Umfang des Softwareprodukts	70
3.5.2. Die Arbeitsproduktivität der Programmierer	73
3.5.3. Zeitplan und Personalverteilung	76
3.5.4. Kostenschätzung und Sachmittel	79
3.5.5. Automatische Unterstützung	81
3.6. Zusammenfassung	83
4. Dokumentation	85
4.1. Benutzerdokumentation	85
4.1.1. Inhalt der Benutzerdokumentation	86
4.1.2. Sprachmittel der Benutzerdokumentation	87
4.1.3. Form der Benutzerdokumentation	88
4.2. Entwicklungsdokumentation	90
4.2.1. Inhalt und Aufbau der Entwicklungsdokumentation	90
4.2.2. Gestaltung der Entwicklungsdokumentation	93
4.3. Technische Dokumentation	94
4.4. Zusammenfassung	95
5. Entwurf	96
5.1. Ein Zerlegungsbeispiel	101
5.2. Typen von Modulen	109
5.3. Beziehungen zwischen Modulen	112
5.4. Kriterien der Zerlegung	117
5.4.1. Einfachheit der Schnittstellen	117
5.4.2. Getrennte Übersetzbarkeit	119
5.4.3. Geringe Modulgröße	120
5.5. Entwurfsmethodik	121
5.5.1. Die Top-down- oder Outside-in-Methode	126
5.5.2. Die Bottom-up- oder Inside-out-Methode	128
5.5.3. Vorschläge für praktikable Entwurfsmethoden	129
5.6. Zusammenfassung	136
6. Spezifikation	137
6.1. HIPO-Diagramme	139
6.2. Informelle Inhaltsverzeichnisse	143

6.3. Parnassche Spezifikation	145
6.4. Algebraische Spezifikation	150
6.4.1. Eine Beispieldspezifikation	151
6.4.2. Die Vorteile der algebraischen Spezifikation	154
6.4.3. Datentypgeneratoren	156
6.4.4. Das Problem der versteckten Funktionen	157
6.4.5. Fehlerbehandlung	159
6.4.6. Erweiterung auf andere Modulararten	161
6.4.7. Schrittweise Implementierung und Verifikation	164
6.4.8. Notwendige Weiterentwicklung der algebraischen Spezifikationsmethode	169
6.5. Petri-Netze	169
6.6. Zusammenfassung	174
 <u>7. Realisierung von Modulkonzepten in Programmiersprachen</u>	 175
7.1. Modulare Zerlegung und algorithmische Sprache	176
7.2. Beurteilung von Zerlegungskonzepten in Programmier- sprachen	176
7.3. Assemblersprachen	177
7.4. Klassische Programmiersprachen	179
7.4.1. Klassische Sprachen ohne Blockstruktur	179
7.4.2. Klassische Sprachen mit Blockstruktur	181
7.4.3. Weiterentwicklung klassischer Sprachen	182
7.5. Sprachen mit abstrakten Datentypen	186
7.6. Trennung von Typ- und Modulbegriff	193
7.7. ALPHARD: von der Spezifikation zur Implementierung	201
7.8. CDL2: eine Sprache mit mehrstufigem Zerlegungskonzept	206
7.9. Zusammenfassung	210
 <u>8. Programmierung im Kleinen</u>	 211
8.1. Elemente der Programmierung im Kleinen	214
8.1.1. Schrittweise Verfeinerung und Verbalisierung	214
8.1.2. Beschreibung des Steuerflusses	220
8.1.3. Datenkonstrukte	228
8.1.4. Prozeduren	234
8.2. Graphische Darstellungen der Programmstruktur	234
8.3. Programmentwicklung und algorithmische Sprachen	237

8.4. Programmverifikation	238
8.4.1. Verifikation mit Hilfe von Zusicherungen	239
8.4.2. Zusicherungen als Konstruktionshilfsmittel	247
8.5. Zusammenfassung	253
9. <u>Testen</u>	255
9.1. Fehlerarten und -ursachen	255
9.2. Testphasen und Teststrategie	258
9.3. Durchführung des Testens	260
9.3.1. Kriterien zur Auswahl der Testdaten	260
9.3.2. Automatische Unterstützung	263
9.4. Symbolische Programmausführung	265
9.4.1. Grundlagen	265
9.4.2. Anwendung der symbolischen Programmausführung	270
9.4.3. Praktische Bedeutung	271
9.5. Finden und Beheben von Fehlern	272
9.6. Hilfsmittel zum Auffinden von Fehlern	273
9.7. Effizienztest und Optimierung	275
9.8. Zusammenfassung	277
10. <u>Projektmanagement</u>	278
10.1. Einbettung der Softwareherstellung in die Betriebsorganisation	278
10.2. "Klassische" Projektmodelle	281
10.3. Das Chef-Programmierer-Team	284
10.4. Probleme der Team-Organisation	286
10.5. Zusammenfassung	289
Literaturverzeichnis	291
Stichwortverzeichnis	301