

Inhaltsverzeichnis

Teil I	Grundlagen	1
1	Modelle und Modellierung	3
1.1	Modelle, die uns umgeben	3
1.2	Modelltheorie	5
1.3	Ziele beim Einsatz von Modellen	6
1.4	Entwicklung und Validierung von Modellen	9
1.5	Modelle im Software Engineering	11
1.6	Theoriebildung	12
1.7	Modellierung durch Graphen und Grafiken	13
1.8	Modellierung durch Zahlen: Skalen und Skalentypen	19
1.9	Übergänge zwischen verschiedenen Skalentypen	22
2	Grundbegriffe	29
2.1	Kosten	29
2.2	Engineering und Ingenieur	31
2.3	Software	34
2.4	Arbeiten, die an Software ausgeführt werden	39
2.5	Weitere Grundbegriffe	40
3	Software Engineering	43
3.1	Fortschritte in Hardware und Software	43
3.2	Grundideen des Software Engineerings	47
3.3	Probleme und Chancen des Software Engineerings	51
3.4	Lehrbücher und andere Basisliteratur	54

4	Software-Nutzen und -Kosten	57
4.1	Die Kosten eines Software-Projekts	57
4.2	Der Aufwand in den einzelnen Phasen des Software-Projekts und in der Wartung	61
4.3	Risiken durch Qualitätsmängel	62
4.4	Die Beziehung zwischen Fehlerentstehung und -entdeckung	63
5	Software-Qualität	65
5.1	Qualität	65
5.2	Taxonomie der Software-Qualitäten	66
Teil II	Menschen und Prozesse	71
6	Menschen im Software Engineering	73
6.1	Software-Leute und Klienten	73
6.2	Rollen und Verantwortlichkeiten	74
6.3	Die Produktivität des Projekts	76
6.4	Motivation und Qualifikation	79
6.5	The Personal Software Process	83
6.6	Moralische und ethische Aspekte	85
7	Das Software-Projekt – Begriffe und Organisation	89
7.1	Begriffsbildung	89
7.2	Software-Projekte	92
7.3	Projekttypen	93
7.4	Formen der Teamorganisation	95
7.5	Die interne Organisation der Software-Hersteller	99
8	Projektleitung und Projektleiter	103
8.1	Ziele und Schwerpunkte des Projektmanagements	103
8.2	Das Vorprojekt	104
8.3	Start des Projekts, Planung	107
8.4	Aufwand, Kosten, Risiken	114
8.5	Projektkontrolle und -steuerung	137
8.6	Der Projektabschluss	146
8.7	Projektmanagement als Führungsaufgabe	148

9	Vorgehensmodelle	153
9.1	Code and Fix und der Software Life Cycle	153
9.2	Schwierigkeiten mit dem Wasserfallmodell	158
9.3	Die Klassifikation der Programme nach Lehman	161
9.4	Prototyping	163
9.5	Nichtlineare Vorgehensmodelle	169
9.6	Das Spiralmodell	177
10	Prozessmodelle	181
10.1	Begriffe und Definitionen	182
10.2	Das Phasenmodell	184
10.3	Das V-Modell	190
10.4	Der Unified Process	202
10.5	Cleanroom Development	211
10.6	Agile Prozesse	218
11	Bewertung und Verbesserung des Software-Prozesses	235
11.1	Voraussetzungen hoher Software-Qualität	235
11.2	CMMI, das Reifegradmodell für Software-Prozesse	236
11.3	SPICE / ISO 15504	250
11.4	Prozessverbesserung	253
Teil III	Daueraufgaben im Software-Projekt	257
12	Dokumentation in der Software-Entwicklung	259
12.1	Begriff und Einordnung	259
12.2	Ziele und Wirtschaftlichkeit der Dokumentation	260
12.3	Taxonomie der Dokumente	262
12.4	Die Benutzungsdokumentation	263
12.5	Die Qualität der Dokumente	265
12.6	Die Form der Dokumente, Normen	266
12.7	Dokumentation in der Praxis	266
12.8	Die gefälschte Entstehungsgeschichte	268

13	Software-Qualitätssicherung und -Prüfung	269
13.1	Software-Qualitätssicherung	269
13.2	Prüfungen	273
13.3	Mängel und Fehler	274
13.4	Prüfungen im Überblick	276
13.5	Reviews	282
13.6	Varianten der Software-Inspektion	292
14	Metriken und Bewertungen	295
14.1	Metriken, Begriff und Taxonomie	296
14.2	Objektive Metriken, Messung	301
14.3	Subjektive Metriken, Beurteilung	305
14.4	Pseudometriken	311
14.5	Die Suche nach der geeigneten Metrik	321
14.6	Ein Beispiel für die Entwicklung einer Metrik	324
14.7	Hinweise für die praktische Arbeit	328
15	Werkzeuge und Entwicklungsumgebungen	331
15.1	Bewertung von Methoden und Werkzeugen	331
15.2	Computer-Aided Software Engineering	333
15.3	Offene integrierte Software-Engineering-Umgebungen	335
15.4	Code-Generierung aus Modellen	340
15.5	Die Auswahl eines Werkzeugs	345
15.6	Ein Blick in die Praxis	348
Teil IV	Techniken der Software-Bearbeitung	351
16	Analyse und Spezifikation	353
16.1	Die Bedeutung der Spezifikation im Entwicklungsprozess	353
16.2	Die Analyse	357
16.3	Begriffslexikon und Begriffsmodell	364
16.4	Anforderungen	366
16.5	Die Spezifikation im Überblick	375
16.6	Die Darstellung der Spezifikation	378
16.7	Konzepte und Komponenten der Spezifikation	384

16.8	Muster und Normen für die Spezifikation	393
16.9	Regeln für Analyse und Spezifikation	395
17	Entwurf	399
17.1	Ziele und Bedeutung des Entwurfs	400
17.2	Begriffe	404
17.3	Prinzipien des Architekturentwurfs	410
17.4	Der objektorientierte Entwurf	421
17.5	Wiederverwendung von Architekturen	429
17.6	Die Qualität der Architektur	453
18	Codierung	455
18.1	Programmiersprachen als Werkstoffe	456
18.2	Regeln für die Codierung	458
18.3	Die Dokumentation des Codes	461
18.4	Realisierungen des Information Hiding	465
18.5	Robuste Programme	471
18.6	Das Vertragsmodell	472
18.7	Werkzeuge zur Codierung	478
19	Programmtest	479
19.1	Begriffe und Grundlagen des Tests	479
19.2	Einige spezielle Testbegriffe	489
19.3	Die Testdurchführung	493
19.4	Die Auswahl der Testfälle	498
19.5	Der Black-Box-Test	504
19.6	Der Glass-Box-Test	513
19.7	Testen mit Zufallsdaten	523
19.8	Beispiele zum Test	524
19.9	Ausblick	539

20	Integration	541
20.1	Einbettung der Integration in die Software-Entwicklung	541
20.2	Integrationsstrategien	542
20.3	Probleme der Integration	546
20.4	Planung und Dokumentation der Integration	547
20.5	Grundsätze für die Integration	548
Teil V	Verwaltung und Erhaltung von Software	551
21	Konfigurationsverwaltung	553
21.1	Grundlagen der Konfigurationsverwaltung	553
21.2	Die Aufgaben der Konfigurationsverwaltung	559
21.3	Identifikation und Benennung von Software-Einheiten	560
21.4	Arbeitsbereiche für die Software-Verwaltung	562
22	Software-Wartung	565
22.1	Begriff und Taxonomie der Software-Wartung	565
22.2	Inhalt und Ablauf der Wartung	570
22.3	Risiken, Probleme und Grundsätze der Wartung	573
22.4	Die Wartungsorganisation	576
23	Reengineering	585
23.1	Software-Evolution	585
23.2	Reengineering	588
23.3	Refactoring	593
23.4	Erblasten, Legacy Software	597
24	Wiederverwendung	601
24.1	Die alltägliche Wiederverwendung	602
24.2	Terminologie und Taxonomie der Wiederverwendung	603
24.3	Kosten und Nutzen der Wiederverwendung	607
24.4	Chancen und Probleme der Wiederverwendung	609
24.5	Rahmenbedingungen für die Wiederverwendung	610
24.6	Entwicklungstechniken für die Wiederverwendung	612
24.7	Von der Codierung zur Komposition	614

Teil VI	Nachwort, Literatur und Index	617
25	Nachwort: Die Schule der Software-Ingenieure	619
25.1	Software Engineering in der Praxis	619
25.2	Stand der Technik und Stand der Praxis	620
25.3	Der Studiengang Softwaretechnik	621
25.4	Nachfrage und Angebot auf dem Ausbildungsmarkt	623
26	Literaturangaben	625
26.1	Hinweise zu den Literaturangaben	625
26.2	Literaturangaben, nach Verfassern geordnet	626
26.3	Verzeichnis der Normen und Standards	649
	Index	653