

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	50
3.4	Lehrbücher und andere Basisliteratur	53

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
5.3	Qualitätsmodelle	71
Teil II	Menschen und Prozesse	75
6	Menschen im Software Engineering	77
6.1	Software-Leute und Klienten	77
6.2	Rollen und Verantwortlichkeiten	78
6.3	Die Produktivität des Projekts	80
6.4	Motivation und Qualifikation	83
6.5	The Personal Software Process	87
6.6	Moralische und ethische Aspekte	88
7	Das Software-Projekt – Begriffe und Organisation	91
7.1	Begriffsbildung	91
7.2	Software-Projekte	94
7.3	Projekttypen	95
7.4	Formen der Teamorganisation	97
7.5	Die interne Organisation der Software-Hersteller	102
8	Projektleitung und Projektleiter	107
8.1	Ziele und Schwerpunkte des Projektmanagements	107
8.2	Das Vorprojekt	108
8.3	Start des Projekts und Projektplanung	111
8.4	Aufwands- und Kostenschätzung	118
8.5	Einige Verfahren zur Aufwandsschätzung	123
8.6	Schätzung in agilen Projekten	133
8.7	Risikomanagement	138
8.8	Projektkontrolle und -steuerung	143

8.9	Der Projektabschluss	152
8.10	Projektmanagement als Führungsaufgabe	154
9	Vorgehensmodelle	161
9.1	Code and Fix und der Software Life Cycle	161
9.2	Schwierigkeiten mit dem Wasserfallmodell	166
9.3	Die Klassifikation der Programme nach Lehman	169
9.4	Prototyping	171
9.5	Nichtlineare Vorgehensmodelle	177
9.6	Das Spiralmodell	186
10	Prozessmodelle	189
10.1	Begriffe und Definitionen	190
10.2	Das Phasenmodell	192
10.3	Das V-Modell	198
10.4	Der Unified Process	210
10.5	Cleanroom Development	219
10.6	Agile Prozesse	225
11	Bewertung und Verbesserung des Software-Prozesses	243
11.1	Voraussetzungen hoher Software-Qualität	243
11.2	Reifegradmodelle für die Prozessbewertung	244
11.3	Die CMM/CMMI-Reifegradmodelle	245
11.4	Fazit	254
11.5	Verbesserung des Software-Prozesses	255
Teil III	Daueraufgaben im Software-Projekt	259
12	Dokumentation in der Software-Entwicklung	261
12.1	Begriff und Einordnung	261
12.2	Ziele und Wirtschaftlichkeit der Dokumentation	262
12.3	Taxonomie der Dokumente	264
12.4	Die Benutzungsdokumentation	266
12.5	Die Qualität der Dokumente	267
12.6	Vorlagen und Normen für Dokumente	268
12.7	Dokumentation in der Praxis	269
12.8	Die gefälschte Entstehungsgeschichte	270

13	Software-Qualitätssicherung und -Prüfung	271
13.1	Software-Qualitätssicherung	271
13.2	Prüfungen	275
13.3	Mängel und Fehler	276
13.4	Prüfungen im Überblick	278
13.5	Reviews	284
13.6	Varianten der Software-Inspektion	294
14	Metriken und Bewertungen	297
14.1	Metriken, Begriff und Taxonomie	298
14.2	Objektive Metriken – Messungen	303
14.3	Subjektive Metriken – Beurteilungen	307
14.4	Pseudometriken	313
14.5	Die Suche nach der geeigneten Metrik	323
14.6	Ein Beispiel für die Entwicklung einer Metrik	327
14.7	Hinweise für die praktische Arbeit	331
Teil IV	Techniken der Software-Bearbeitung	335
15	Analyse und Spezifikation	337
15.1	Die Bedeutung der Spezifikation im Entwicklungsprozess	337
15.2	Die Analyse	341
15.3	Begriffslexikon und Begriffsmodell	348
15.4	Anforderungen	351
15.5	Die Spezifikation im Überblick	359
15.6	Die Darstellung der Spezifikation	362
15.7	Konzepte und Komponenten der Spezifikation	369
15.8	Muster und Normen für die Spezifikation	382
15.9	Regeln für Analyse und Spezifikation	384
16	Entwurf	387
16.1	Ziele und Bedeutung des Entwurfs	388
16.2	Begriffe	392
16.3	Prinzipien des Architekturentwurfs	399
16.4	Architekturmuster	412
16.5	Entwurfsmuster	426
16.6	Weitere Arten der Wiederwendung von Architekturen	435

16.7	Der Entwurf von Anwendungssoftware	442
16.8	Die Qualität der Architektur	455
17	Codierung	463
17.1	Programmiersprachen als Werkstoffe	464
17.2	Regeln für die Codierung	466
17.3	Die Dokumentation des Codes	469
17.4	Realisierungen des Information Hiding	473
17.5	Robuste Programme	480
17.6	Das Vertragsmodell	481
17.7	Werkzeuge zur Codierung	488
18	Programmtest	489
18.1	Begriffe und Grundlagen des Tests	489
18.2	Einige spezielle Testbegriffe	499
18.3	Die Testdurchführung	503
18.4	Die Auswahl der Testfälle	508
18.5	Der Black-Box-Test	514
18.6	Der Glass-Box-Test	528
18.7	Testen mit Zufallsdaten	538
18.8	Beispiele zum Test	539
18.9	Werkzeuge für den Test	553
18.10	Ausblick	554
19	Integration	555
19.1	Einbettung der Integration in die Software-Entwicklung	555
19.2	Integrationsstrategien	556
19.3	Probleme der Integration	560
19.4	Planung und Dokumentation der Integration	561
19.5	Grundsätze für die Integration	562
Teil V	Verwaltung und Erhaltung von Software	565
20	Konfigurationsverwaltung	567
20.1	Grundlagen der Konfigurationsverwaltung	567
20.2	Die Aufgaben der Konfigurationsverwaltung	574
20.3	Benennung und Identifikation von Software-Einheiten	575

20.4	Arbeitsumgebungen für die Software-Bearbeitung	578
20.5	Automatisierte Software-Auslieferung	581
21	Software-Wartung	587
21.1	Begriff und Taxonomie der Software-Wartung	587
21.2	Inhalt und Ablauf der Wartung	592
21.3	Risiken, Probleme und Grundsätze der Wartung	595
21.4	Die Wartungsorganisation	598
22	Reengineering	607
22.1	Software-Evolution	607
22.2	Reengineering	610
22.3	Refactoring	615
22.4	Erblasten, Legacy Software	619
22.5	Technische Schulden	621
23	Wiederverwendung	627
23.1	Die alltägliche Wiederverwendung	627
23.2	Terminologie und Taxonomie der Wiederverwendung	629
23.3	Kosten und Nutzen der Wiederverwendung	633
23.4	Chancen und Probleme der Wiederverwendung	635
23.5	Rahmenbedingungen für die Wiederverwendung	636
23.6	Entwicklungstechniken für die Wiederverwendung	638
23.7	Von der Codierung zur Komposition	640
Teil VI	Software Engineering lehren	643
24	Software-Engineering-Ausbildungs- und -Studiengänge	645
24.1	Der Studiengang Software Engineering in Stuttgart	645
24.2	Weitere Software-Engineering-Ausbildungs- und -Studiengänge	647
Teil VII	Literatur und Index	649
25	Literaturangaben	651
25.1	Hinweise zu den Literaturangaben	651
25.2	Literaturangaben, nach Verfassern geordnet	652
25.3	Verzeichnis der Normen und Standards	677
26	Index	681