

Jochen Ludewig • Horst Lichter

Software Engineering

Grundlagen, Menschen, Prozesse, Techniken

fijj dpunkt.verlag

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.	53

4	Software-Nutzen und -Kosten	57
4.1	Ziele und Nutzen eines Software-Projekts.	57
4.2	Kostenfaktoren eines Software-Projekts.	59
4.3	Der Aufwand in den einzelnen Phasen des Software-Projekts und in der Wartung.	59
4.4	Risiken durch Qualitätsmängel.	60
4.5	Die Beziehung zwischen Fehlerentstehung und -entdeckung.	61
5	Software-Qualität	63
5.1	Qualität.	63
5.2	Taxonomie der Software-Qualitäten.	64
Teil II	Menschen und Prozesse	69
6	Menschen im Software Engineering	71
6.1	Software-Leute und Klienten.	71
6.2	Rollen und Verantwortlichkeiten.	72
6.3	Die Produktivität des Projekts.	74
6.4	Motivation und Qualifikation.	78
6.5	The Personal Software Process.	82
6.6	Moralische und ethische Aspekte.	84
7	Das Software-Projekt - Begriffe und Organisation	87
7.1	Begriffsbildung.	87
7.2	Software-Projekte.	90
7.3	Projekttypen.	91
7.4	Formen der Teamorganisation.	93
7.5	Die interne Organisation der Software-Hersteller.	97
8	Projektleitung und Projektleiter	101
8.1	Ziele und Schwerpunkte des Projektmanagements.	101
8.2	Die Initialisierung des Projekts.	102
8.3	Aufwands- und Kostenschätzung.	114
8.4	Projektkontrolle und -Steuerung.	129
8.5	Termindrift-Diagramme (Meilenstein-Trend-Analyse).	133
8.6	Der Projektabschluss.	135

Inhaltsverzeichnis

8.7	Die Fähigkeiten des Projektleiters	136
8.8	Gängige Führungsprobleme.	139
8.9	Regeln für das Projektmanagement	140
9	Vorgehensmodelle	143
9.1	Code and Fix und der Software Life Cycle.	143
9.2	Schwierigkeiten mit dem Wasserfallmodell	148
9.3	Die Klassifikation der Programme nach Lehman.	151
9.4	Prototyping	153
9.5	Nichtlineare Vorgehensmodelle.	159
9.6	Das Spiralmodell	168
10	Prozessmodelle	171
10.1	Begriffe und Definitionen.	172
10.2	Das Phasenmodell.	174
10.3	Das V-Modell.	178
10.4	Der Unified Process.	190
10.5	Cleanroom Development	200
10.6	Extreme Programming und andere agile Prozesse.	207
11	Bewertung und Verbesserung des Software-Prozesses	219
11.1	Voraussetzungen hoher Software-Qualität	219
11.2	CMMI, das Reifegradmodell für Software-Prozesse.	220
11.3	SPICE / ISO 15504.	233
11.4	Prozessverbesserung	236
ggMHII	Daueraufgaben im Software-Projekt	239
12	Dokumentation in der Software-Entwicklung	241
12.1	Begriff und Einordnung	241
12.2	Ziele und Wirtschaftlichkeit der Dokumentation.	242
12.3	Die Form der Dokumente, Normen	244
12.4	Taxonomie der Dokumente.	245
12.5	Die Qualität der Dokumente.	246
12.6	Dokumentation in der Praxis.	247
12.7	Die gefälschte Entstehungsgeschichte.	249

13	Software-Qualitätssicherung und-Prüfung	251
13.1	Software-Qualitätssicherung	251
13.2	Prüfungen	255
13.3	Mängel und Fehler	256
13.4	Software Engineering und Software-Prüfung	258
13.5	Prüfungen im Überblick	258
13.6	Reviews	262
13.7	Varianten der Software-Inspektion	273
14	Metriken und Bewertungen	277
14.1	Metriken, Begriff und Taxonomie	278
14.2	Objektive Metriken, Messung	283
14.3	Subjektive Metriken, Beurteilung	287
14.4	Pseudometriken	293
14.5	Die Suche nach der geeigneten Metrik	303
14.6	Ein Beispiel für die Entwicklung einer Metrik	306
14.7	Hinweise für die praktische Arbeit	310
15	Werkzeuge und Entwicklungsumgebungen	313
15.1	Bewertung von Methoden und Werkzeugen	313
15.2	Computer-Aided Software Engineering	315
15.3	Offene integrierte Software-Engineering-Umgebungen	317
15.4	Code-Generierung aus Modellen	322
15.5	Die Auswahl eines Werkzeugs	327
15.6	Ein Blick in die Praxis	330
Teil IV	Techniken der Software-Bearbeitung	333
16	Analyse und Spezifikation	335
16.1	Die Bedeutung der Spezifikation im Entwicklungsprozess	335
16.2	Die Analyse	339
16.3	Begriffslexikon und Begriffsmodell	346
16.4	Anforderungen	348
16.5	Die Spezifikation	353
16.6	Grafische, formale und natürlichsprachliche Spezifikation	364

Inhaltsverzeichnis

16.7	Spezifikation und Entwurf - Probleme der Abgrenzung	368
16.8	Normen und Vorlagen	369
16.9	Regeln für Analyse und Spezifikation	370
17	Entwurf	373
17.1	Die Bedeutung des Entwurfs im Entwicklungsprozess	374
17.2	Begriffe	377
17.3	Prinzipien des Architekturentwurfs	384
17.4	Konzepte des objektorientierten Entwurfs	393
17.5	Architekturmuster	399
17.6	Entwurfsmuster	407
17.7	Bibliotheken und Rahmenwerke	414
17.8	Qualität und Dokumentation der Architektur	418
18	Codierung	421
18.1	Programmiersprachen als Werkstoffe	422
18.2	Regeln für die Codierung	424
18.3	Die Dokumentation des Codes	427
18.4	Realisierungen des Information Hiding	431
18.5	Robuste Programme	437
18.6	Das Vertragsmodell	438
18.7	Werkzeuge zur Codierung	443
19	Programmtest	445
19.1	Begriffe und Grundlagen des Tests	445
19.2	Einige spezielle Testbegriffe	455
19.3	Testdurchführung	459
19.4	Die Auswahl der Testfälle	464
19.5	Der Black-Box-Test	470
19.6	Der Glass-Box-Test	480
19.7	Testen mit Zufallsdaten	487
19.8	Beispiele zum Test	488
19.9	Ausblick	503

20	Integration	505
20.1	Einbettung der Integration in die Software-Entwicklung	505
20.2	Integrationsstrategien	506
20.3	Probleme der Integration	510
20.4	Planung und Dokumentation der Integration	511
20.5	Grundsätze für die Integration	512
Teil V	Verwaltung und Erhaltung der Software	515
21	Konfigurationsverwaltung	517
21.1	Grundlagen der Konfigurationsverwaltung	517
21.2	Die Aufgaben der Konfigurationsverwaltung	522
21.3	Identifikation und Benennung von Software-Einheiten	523
21.4	Arbeitsbereiche für die Software-Verwaltung	526
22	Software-Wartung	529
22.1	Begriff und Taxonomie der Software-Wartung	529
22.2	Inhalt und Ablauf der Wartung	534
22.3	Risiken, Probleme und Grundsätze der Wartung	537
22.4	Die Wartungsorganisation	540
23	Reengineering	547
23.1	Software-Evolution	547
23.2	Reengineering	550
23.3	Refactoring	555
23.4	Erblasten, Legacy Software	559
24	Wiederverwendung	563
24.1	Die alltägliche Wiederverwendung	564
24.2	Terminologie und Taxonomie der Wiederverwendung	565
24.3	Kosten und Nutzen der Wiederverwendung	569
24.4	Chancen und Probleme der Wiederverwendung	571
24.5	Rahmenbedingungen für die Wiederverwendung	572
24.6	Entwicklungstechniken für die Wiederverwendung	574
24.7	Von der Codierung zur Komposition	576

Teil VI	Nachwort, Literatur und Index	579
25	Nachwort: Die Schule der Software-Ingenieure	581
25.1	Software Engineering in der Praxis.	581
25.2	Stand der Technik und Stand der Praxis.	582
25.3	Der Studiengang Softwaretechnik.	583
25.4	Nachfrage und Angebot auf dem Ausbildungsmarkt.	585
26	Literaturangaben	587
26.1	Hinweise zu den Literaturangaben.	587
26.2	Literaturangaben, nach Verfassern geordnet.	588
26.3	Verzeichnis der Normen und Standards.	608
	Index	611