

Inhaltsverzeichnis

1 Was ist Software-Engineering?	1
2 Prozessmodellierung	7
2.1 Unternehmensprozesse.....	8
2.2 Prozessmodellierung mit Aktivitätsdiagrammen.....	10
2.3 Risikomanagement	19
2.4 Risikoanalyse Prozessmodellierung.....	21
2.5 Aufgaben.....	22
3 Vorgehensmodelle	25
3.1 Phasen der Software-Entwicklung.....	27
3.2 Wasserfallmodell	28
3.3 Prototypische Entwicklung	30
3.4 Iterative Entwicklung.....	31
3.5 Iterativ-inkrementelle Entwicklung	33
3.6 Allgemeines V-Modell	34
3.7 Das V-Modell der Bundesrepublik Deutschland	35
3.8 Rational Unified Process.....	42
3.9 Agile Vorgehensmodelle	45
3.10 Scrum.....	48
3.11 Extreme Programming	50
3.12 Risikoanalyse Vorgehensmodell.....	53
3.13 Aufgaben.....	53
4 Anforderungsanalyse	55
4.1 Stakeholder und Ziele	55
4.2 Klärung der Hauptfunktionalität (Use Cases).....	63
4.3 Beschreibung typischer und alternativer Abläufe	72
4.4 Ableitung funktionaler Anforderungen	76
4.5 Nicht-funktionale Anforderungen	84
4.6 Lasten- und Pflichtenheft.....	88

4.7 Risikoanalyse Anforderungsanalyse	89
4.8 Aufgaben.....	90
5 Grobdesign.....	93
5.1 Systemarchitektur	93
5.2 Ableitung von grundlegenden Klassen	94
5.3 Ableitung von Methoden und Kontrollklassen	101
5.4 Validierung mit Sequenzdiagrammen	107
5.5 Überlegungen zur Oberflächenentwicklung.....	117
5.6 Anforderungsverfolgung.....	120
5.7 Risikoanalyse Grobdesign	122
5.8 Aufgaben.....	123
6 Vom Klassendiagramm zum Programm	127
6.1 CASE-Werkzeuge.....	127
6.2 Übersetzung einzelner Klassen	129
6.3 Übersetzung von Assoziationen	133
6.4 Spezielle Arten der Objektzugehörigkeit.....	138
6.5 Aufbau einer Software-Architektur.....	142
6.6 Weitere Schritte zum lauffähigen Programm	148
6.7 Risikoanalyse Klassendiagrammübersetzung	153
6.8 Aufgaben.....	154
7 Konkretisierungen im Feindesign.....	157
7.1 Zustandsdiagramme.....	157
7.2 Object Constraint Language	168
7.3 Risikoanalyse Feindesign.....	174
7.4 Aufgaben.....	174
8 Optimierung des Designmodells	177
8.1 Design im Kleinen.....	178
8.2 Model View Controller.....	186
8.3 Vorstellung einiger GoF-Pattern	192
8.4 Abschlussbemerkungen zu Pattern.....	214
8.5 Risikoanalyse Design-Optimierungen	217
8.6 Aufgaben.....	217

9 Implementierungsaspekte.....	223
9.1 Einfluss nicht-funktionaler Anforderungen.....	224
9.2 Verteilte Systeme	225
9.3 Grundideen von XML	230
9.4 Programmabibliotheken	232
9.5 Komponenten.....	233
9.6 Frameworks	238
9.7 Persistente Datenhaltung	244
9.8 Annotationen.....	247
9.9 Domain Specific Languages	250
9.10 Model Driven Architecture	253
9.11 Refactoring.....	255
9.12 Risikoanalyse Implementierungsaspekte	257
9.13 Aufgaben.....	258
10 Oberflächengestaltung	263
10.1 Hintergründe der Oberflächengestaltung	263
10.2 Konkretisierung des Nutzbarkeitsbegriffs	265
10.3 Berücksichtigung der Ergonomie im Software-Entwicklungsprozess.....	270
10.4 Prüfung der Nutzbarkeit	272
10.5 Risikoanalyse Oberflächengestaltung	276
10.6 Aufgaben.....	277
11 Qualitätssicherung	279
11.1 Formale Korrektheit	280
11.2 Zusicherungen.....	282
11.3 Unit-Tests	285
11.4 Testbarkeit von Systemen herstellen.....	292
11.5 Äquivalenzklassentests.....	295
11.6 Kontrollflussbezogene Testverfahren	303
11.7 Testarten und Testumfeld.....	309
11.8 Metriken	315
11.9 Konstruktive Qualitätssicherung.....	320
11.10 Manuelle Prüfverfahren.....	321
11.11 Risikoanalyse Qualitätssicherung.....	325
11.12 Aufgaben.....	326

Inhaltsverzeichnis

12 Umfeld der Software-Entwicklung	333
12.1 Versionsmanagement	334
12.2 Build-Management	338
12.3 Grundlagen der Projektplanung und -verfolgung	343
12.4 Aufwandsschätzung	353
12.5 Qualitätsmanagement	365
12.6 Der Mensch im Projekt	372
12.7 Risikoanalyse Projektumfeld	379
12.8 Aufgaben	380
A UML-Überblick	383
Literaturverzeichnis	389
Sachwortverzeichnis	397