

Veikko Krypczyk, Olena Bochkor

Handbuch für Softwareentwickler

Auf einen Blick

TEIL I Überblick

- | | | |
|---|---|-----|
| 1 | Überblick | 21 |
| 2 | Programmierung als Kern der Softwareentwicklung | 35 |
| 3 | Algorithmen und Datenstrukturen | 105 |

TEIL II Der Softwarelebenszyklus

- | | | |
|----|--|-----|
| 4 | Softwareprojekte professionell planen | 167 |
| 5 | Die Anforderungsanalyse-Startpunkt der Entwicklung | 199 |
| 6 | Der Entwurf des Softwaresystems | 239 |
| 7 | Die Implementierung schafft den Kern der Anwendung | 281 |
| 8 | Testen als Voraussetzung für fehlerarme Software | 319 |
| 9 | Distribution-das Produkt muss zum Kunden | 359 |
| 10 | Vom Altsystem zurzeitgemäßen Anwendung | 393 |

TEIL III Technologien und Methoden

- | | | |
|----|---|-----|
| 11 | Kundenzufriedenheit durch Nutzerorientierung | 419 |
| 12 | Datensicherheit und Datenschutz | 443 |
| 13 | Plattform-und geräteübergreifende Entwicklung | 463 |
| 14 | Parallelprogrammierung | 503 |
| 15 | Grundlagen der Datenhaltung | 525 |
| 16 | Werkzeugunterstützung | 555 |
| 17 | Qualitätssicherung und Clean Code Development | 581 |

TEIL IV Trends

- | | | |
|----|-----------------------------|-----|
| 18 | Enterprise Mobile Computing | 617 |
| 19 | Internet ofThings | 649 |
| 20 | Cloud-Computing | 675 |

Inhalt

Vorwort	17
---------	----

TEIL I Überblick

1 Überblick	21
1.1 Berufswunsch Softwareentwickler	21
1.1.1 Der Softwareentwickler-ein ganz besonderer Informatiker	23
1.1.2 Aktuelle Arbeitsmarktsituation und Verdienstmöglichkeiten	26
1.2 Über dieses Buch	28
1.3 Literatur und Links	34
2 Programmierung als Kern der Softwareentwicklung	35
2.1 Die Programmierung	35
2.2 Paradigmen der Softwareentwicklung	39
2.3 Objektorientierte Programmentwicklung	40
2.3.1 Objektorientierung im Überblick	41
2.3.2 Objektorientierte Konzepte im Detail	45
2.3.3 Objektorientierung und Wiederverwendung	52
2.3.4 Visualisierung: Objektorientierung und UML	56
2.4 Programmiersprachen	62
2.4.1 Historische Entwicklung und sprachliche Vielfalt	62
2.4.2 Die Systematik der Programmiersprachen	66
2.5 Essenzielle Sprachmerkmale	72
2.5.1 Kommentare	72
2.5.2 Operatoren und Vergleiche	76
2.5.3 Kontrollstrukturen	78
2.5.4 Objektorientierung	89
2.6 Fazit und Ausblick	103
2.7 Literatur und Links	104

3	Algorithmen und Datenstrukturen	105
3.1	Algorithmen als Kernelemente des Programms	106
3.1.1	Wichtige Klassen von Algorithmen	109
3.1.2	Komplexität von Algorithmen	109
3.2	Entwurf von Algorithmen	111
3.2.1	Vorgehensweise	112
3.2.2	Visualisierung des Ablaufs von Algorithmen	114
3.2.3	Fallbeispiele	119
3.3	Sortieren und Suchen als Basis-Algorithmen	123
3.3.1	Sortieren – das Wichtigste im Überblick	124
3.3.2	Such-Algorithmen im Überblick	129
3.4	Elementare Datenstrukturen	138
3.4.1	Datenobjekte	139
3.4.2	Systematik von Datenobjekten und Datentypen	142
3.5	Zusammenfassung und Ausblick	160
3.6	Literatur und Links	162

TEIL II Der Softwarelebenszyklus

4	Softwareprojekte professionell planen	167
4.1	Der Wasserfall – mehr als nur historische Bedeutung	168
4.1.1	Analyse	169
4.1.2	Entwurf	171
4.1.3	Implementierung	171
4.1.4	Test	172
4.1.5	Betrieb und Wartung	173
4.1.6	Kritische Betrachtung	173
4.2	Iterative Entwicklung – schrittweise zum Ziel	173
4.3	Das V-Modell – eine Struktur hauptsächlich für Projekte der öffentlichen Hand	175
4.4	Bessere Risikobeherrschung durch das Spiralmodell	176
4.5	Agile Ansätze – der Komplexität und Unsicherheit mit Flexibilität begegnen	177
4.5.1	Extreme Programming	180
4.5.2	Serum	181

4.6	Softwareentwicklung und die Schnittstellen zum Projektmanagement	187
4.6.1	Personen und Rollen in IT-Projekten	191
4.7	Ist Agilität heute alternativlos?	194
4.8	Literatur und Links	197
5	Die Anforderungsanalyse – Startpunkt der Entwicklung	199
5.1	Überblick und Zielstellung	199
5.2	Ausgangssituation und Notwendigkeit	202
5.3	Anforderungen ermitteln	207
5.3.1	Quellen von Anforderungen	207
5.3.2	Anforderungen erforschen	208
5.3.3	Arten von Anforderungen	209
5.3.4	Verbindlichkeit und Priorität von Anforderungen	211
5.4	Anforderungen formulieren und visualisieren	219
5.4.1	Lastenheft	221
5.4.2	Pflichtenheft	226
5.4.3	Grafische Notation als Hilfsmittel	227
5.5	Tool-Unterstützung	230
5.6	Diskussion im Kontext agilen Vorgehens	234
5.7	Fazit	237
5.8	Literatur und Links	237
6	Der Entwurf des Softwaresystems	239
6.1	Die Softwarearchitektur als Basis für Erfolg und Wartbarkeit eines Softwaresystems	240
6.1.1	Ziele und Auswahlkriterien für eine Architektur	241
6.1.2	Architekturprinzipien	243
6.2	Architekturparadigmen	246
6.2.1	Monolithisches IT-System	246
6.2.2	Verteilte Architektur	248

6.2.3	Client-Server-Architektur	250
6.2.4	Serviceorientierte Architektur	251
6.2.5	Web-Architektur	255
6.2.6	Middleware	258
6.3	Entwurfsmuster als Baupläne	259
6.3.1	Fassaden-Muster	261
6.3.2	Model-View-Controller-Muster	262
6.4	Entwurfsunterstützung durch grafische Modellierung	265
6.4.1	Das Komponentendiagramm	266
6.4.2	Das Paketdiagramm	267
6.4.3	Das Verteilungsdiagramm	268
6.5	Weitere Entwurfsentscheidungen	269
6.5.1	Softwaretyp	269
6.5.2	Technologie	269
6.6	Erfolgreiche Entwürfe mithilfe von Prototypen	270
6.6.1	Arten von Prototypen	271
6.6.2	Wege und Tools zum Prototyp	273
6.7	Zusammenfassung und Fazit	279
6.8	Literatur und Links	279
7	Die Implementierung schafft den Kern der Anwendung	281
7.1	Die Entwicklung der Benutzerschnittstelle	282
7.1.1	Benutzeroberflächen in Java	283
7.1.2	Benutzeroberflächen auf Basis der Beschreibungssprache XAML	299
7.1.3	Im Turbogang zum User Interface: Rapid Application Development	309
7.1.4	Web-Technologien	311
7.2	Die Businesslogik umsetzen	311
7.3	Die Datenwelt anbinden	313
7.3.1	Java Database Connectivity	314
7.3.2	Entity Framework	315
7.4	Zusammenfassung und Fazit	316
7.5	Literatur und Links	317

8	Testen als Voraussetzung für fehlerarme Software	319
8.1	Zur Notwendigkeit von Softwaretests	320
8.2	Testgetriebene Entwicklung	321
8.3	Ein Überblick über wichtige Testarten	323
8.4	Testmethoden	327
8.4.1	Statische Tests	327
8.4.2	Dynamische Tests	330
8.5	Testebenen	335
8.5.1	Komponententests	337
8.5.2	Integrationstests	341
8.5.3	Systemtests	342
8.5.4	Abnahme- und Nutzertests	343
8.6	Technik des Testens	353
8.6.1	Tools	353
8.6.2	Test-Automation	355
8.7	Fazit und Zusammenfassung	357
8.8	Literatur und Links	357
9	Distribution – das Produkt muss zum Kunden	359
9.1	Die Softwareverteilung im Überblick	360
9.1.1	Die Merkmale von Software und ihre Auswirkungen auf den Vertrieb	360
9.1.2	Vertragliche und lizenzrechtliche Bestimmungen	361
9.1.3	Vertriebswege	362
9.1.4	Erlösmodelle	365
9.2	Die App-Store-Revolution	370
9.2.1	Vor- und Nachteile eines App Stores	371
9.2.2	Programminterne Stores	372
9.2.3	Vermarktungsstrategien	374
9.2.4	Präsentation im Store	375
9.3	DevOps-der Kreis schließt sich	376
9.3.1	Probleme durch eine künstliche Trennung der Zuständigkeiten	377
9.3.2	Verbesserte Zusammenarbeit	378

9.4	Die Technik der Softwarebereitstellung	380
9.4.1	Software für den Desktop	380
9.4.2	Apps für mobile Systeme	387
9.4.3	Automatisierte Softwareverteilung in Unternehmen	390
9.5	Literatur und Links	392

10 Vom Altsystem zur zeitgemäßen Anwendung 393

10.1	Technologien im Wandel	394
10.2	Notwendigkeit einer Software-Migration	396
10.3	Ziele einer Software-Migration	398
10.4	Planung der Migration	398
10.5	Migrationsstrategien	401
10.6	Arten der Migration	405
10.7	Wirtschaftlichkeitsbetrachtungen	406
10.8	Eine Fallstudie	407
10.8.1	Der Status quo	408
10.8.2	Einen technischen Migrationsansatz auswählen	409
10.8.3	Web-Applikationen mit Wisej	410
10.9	Literatur und Links	416

TEIL III Technologien und Methoden

11 Kundenzufriedenheit durch Nutzerorientierung 419

11.1	Der Nutzer im Fokus	421
11.1.1	Development versus Design	421
11.1.2	Personas-was sind sie?	423
11.2	Benutzerschnittstellen im Wandel	424
11.2.1	Command-line interface	425
11.2.2	Graphical User Interface	427
11.2.3	Natural User Interface	431

11.3 Gestaltung von Benutzeroberflächen	433
11.3.1 Objektive Merkmale und subjektives Empfinden	434
11.3.2 Software-Ergonomie	437
11.3.3 Designrichtlinien als Handlungsrahmen	439
11.4 Literatur und Links	441
<u>12 Datensicherheit und Datenschutz</u>	443
12.1 Begriffe und Angrenzungen	445
12.2 Dilemma: Globale Technik und lokale Gesetzgebung	447
12.3 Betrieblicher Datenschutz	449
12.4 Datenschutz und Mobile Computing	450
12.5 Datenschutz und Cloud-Computing	453
12.6 Sichere Kommunikation durch Verschlüsselung	454
12.6.1 Kryptografie	455
12.6.2 Digitale Signatur	457
12.6.3 Digitales Zertifikat	460
12.7 Fazit	461
12.8 Literatur und Links	462
<u>13 Plattform- und geräteübergreifende Entwicklung</u>	<u>463</u>
13.1 Systemumgebungen im Einsatz	463
13.1.1 Der Desktop	464
13.1.2 Der Webserver	467
13.1.3 Apps für mobile Geräte	468
13.1.4 Gerätetechnik-vom Smartphone bis zum Desktop	469
13.2 Ansätze für die geräte- und plattformübergreifende Entwicklung	471
13.2.1 Java	471
13.2.2 Universal Windows Platform	473

13.3	Mobile Computing als Motivationstreiber	475
13.3.1	App-Arten	475
13.3.2	Native Technologien	477
13.3.3	Hybride Technologien	479
13.3.4	Mit plattformübergreifender Programmierung zur nativen App	481
13.3.5	Weitere Ansätze	498
13.4	Fazit	500
13.5	Literatur und Links	500
14	Parallelprogrammierung	503
14.1	Systemtechnische Ebene	504
14.1.1	Prozesse und Threads	507
14.1.2	Zustände von Prozessen	509
14.2	Anwendungsebene	510
14.2.1	Parallelisierung von Algorithmen	512
14.2.2	Probleme und Erfolgsbestimmung	516
14.3	Programmiertechnische Ebene	518
14.3.1	Konzepte der Task Parallel Library	519
14.3.2	Reaktionsfähigkeit mithilfe von async und await	521
14.4	Zusammenfassung	522
14.5	Literatur und Links	523
15	Grundlagen der Datenhaltung	525
15.1	Datenbank-ein Ort, um Ordnung zu halten	525
15.1.1	Anforderungen an eine Datenbank	527
15.1.2	Überblick über Datenbanksysteme	529
15.2	Phasen der Datenbankentwicklung	534
15.2.1	Das Entity-Relationship-Modell (ER-Modell)	536
15.2.2	Alles eine Frage des Schlüssels	541
15.2.3	Normalisierung der Datenbankstruktur	546
15.3	NoSQL-Datenbanken und Fazit	551
15.4	Literatur und Links	553

16	Werkzeugunterstützung	555
16.1	Die Rolle von Werkzeugen	556
16.2	Integrierte Entwicklungsumgebungen als Multifunktionswerkzeuge	557
16.2.1	Bestandteile einer integrierten Entwicklungsumgebung	558
16.2.2	Wichtige integrierte Entwicklungsumgebungen im Überblick	562
16.3	Versionsverwaltungen als Voraussetzung für eine erfolgreiche Entwicklung	564
16.3.1	Varianten der Versionsverwaltung	565
16.3.2	Vergleich zentrale und verteilte Versionsverwaltung	567
16.3.3	Git-Grundlagen	568
16.4	Weitere Softwaretools für die Entwicklung	574
16.4.1	Tools für die modellgetriebene Entwicklung	575
16.4.2	Tools für die Datenmodellierung	576
16.4.3	Einfache Texteditoren	577
16.4.4	Tools für die Koordination der Zusammenarbeit und fürs Projektmanagement	578
16.5	Fazit und Ausblick	579
16.6	Literatur und Links	580

17 Qualitätssicherung und Clean Code Development **58i**

17.1	Das Verständnis von Softwarequalität	581
17.2	Ansätze zur Bestimmung der Softwarequalität	584
17.2.1	Qualitätseigenschaften nach Boehm	584
17.2.2	Quantifizierung der Qualität nach Gilb	585
17.2.3	Goal Question Metrie	587
17.3	Softwarequalität zwischen Nutzen und Kosten	589
17.4	Qualität verbessern	591
17.4.1	Sich stetig weiterentwickeln: kontinuierliche Verbesserung	592
17.4.2	Prozessqualität über alle Entwicklungsschritte	596
17.4.3	Regeln als Handlungsleitfaden	597
17.4.4	Clean Code Development	602
17.4.5	Lässig und unverzichtbar-die Entwicklungsdokumentation	605

17.5	Eine gesunde Fehlerkultur	606
17.5.1	Der Begriff des Fehlers	608
17.5.2	Umgang mit Fehlern	610
17.5.3	Fehlermetriken	612
17.6	Fazit und Zusammenfassung	613
17.7	Literatur und Links	614

TEIL IV Trends

18 Enterprise Mobile Computing 617

18.1	Einsatzszenarien	618
18.1.1	Mobilisierung bestehender digitaler Geschäftsprozesse	619
18.1.2	Digitale Transformation	620
18.2	Bring Your Own Device	627
18.2.1	Bedeutung und Nutzen einer BYOD-Strategie	628
18.2.2	Problemlage und Anforderungen bei einer BYOD-Strategie	630
18.2.3	Mögliche Szenarien aus Unternehmenssicht	634
18.2.4	Umsetzungsstrategie	636
18.3	Technische Aspekte für Enterprise Mobile Computing	637
18.3.1	Geräteauswahl	637
18.3.2	Enterprise Mobile Device Management	638
18.3.3	Backend as a Service for Enterprise	639
18.4	Literatur und Links	647

19 Internet of Things 649

19.1	Stand heute und Zukunftsszenarien	650
19.1.1	Vom Eigenbau zum Standard	653
19.2	IoT-Architekturansätze	656
19.2.1	Netzwerktopologien	656
19.2.2	Anwendungsarchitekturen	657
19.3	Hardware auf kleinstem Raum	660

19.4 Fallbeispiel: Windows 10 auf dem Raspberry Pi	665
19.4.1 Anforderungen und Anwendungsarchitektur	665
19.4.2 Installation von Windows 10 IoT	668
19.4.3 Konfiguration aus der Ferne per Remote	670
19.4.4 Entwicklungsumgebung konfigurieren	671
19.5 Fazit und Ausblick	672
19.6 Literatur und Links	672
20 Cloud-Computing	<u>675</u>
20.1 Grundlagen des Cloud-Computings	677
20.1.1 Merkmale und Eigenschaften des Cloud-Computings	677
20.1.2 Modelle des Cloud-Computings	678
20.1.3 Nutzungsmodelle des Cloud-Computings	681
20.2 Ökonomische Gesichtspunkte	683
20.2.1 On-Premises-versus Cloud-Computing	684
20.2.2 Kostenbasierter Vergleich	685
20.3 Cloud-Computing als Werkzeug der Softwareentwicklung	688
20.3.1 Cloud-Lösungen für Kommunikation und Datenaustausch	688
20.3.2 Projekt- und Quellcodeverwaltung	689
20.3.3 Entwicklungsumgebungen, Build- und Testsysteme	690
20.3.4 Monitoring und Bugtracking	691
20.4 Fazit	692
20.5 Literatur und Links	692
Index	693