

Auf einen Blick

1	Einführung	25
2	Softwarearchitektur und wichtige Designfragen	37
3	Softwaredesign	139
4	.NET für Fortgeschrittene	263
5	Professionell codieren	423
6	Windows Communication Foundation	525
7	Datenbank und Datenzugriff	717
8	Workflow Foundation	809
9	Windows-Apps und WinRT	989
10	Softwaretests	1051
11	Softwarepflege und Projektmanagement	1141
12	Zum Schluss	1189

Inhalt

Vorwort zur dritten Auflage	21
Vorwort zur ersten Auflage	23

1 Einführung 25

1.1 Lehre und Praxis – der Unterschied	25
1.1.1 Gute Software, schlechte Software	26
1.1.2 Wege zur Lösung	27
1.2 Das Fallbeispiel	29
1.3 Die einzelnen Kapitel	30

2 Softwarearchitektur und wichtige Designfragen 37

2.1 Einführung	38
2.1.1 Das Problem	38
2.1.2 Gute Softwarearchitektur, schlechte Softwarearchitektur	39
2.1.3 Aufgaben	40
2.1.4 Anwendungstypen	41
2.1.5 Der Architekt	43
2.2 Anforderungen	44
2.2.1 Arten von Anforderungen	45
2.2.2 Anatomie einer Anforderung	48
2.2.3 Das richtige Maß	49
2.3 Komponenten	50
2.3.1 Komponenten identifizieren	51
2.3.2 Beziehungen	53
2.4 Prozesse	55
2.4.1 Was ist ein Prozess?	55
2.4.2 Geschäftsprozessmodellierung	56
2.4.3 Auswirkungen auf die Architektur	57
2.5 Layer (Schichten)	59
2.5.1 Grundlagen	59
2.5.2 Layer vs. Tier	62

2.5.3	Die Fassade	63
2.5.4	Presentation Layer	64
2.5.5	Business Layer	66
2.5.6	Data Layer	70
2.6	Tier und verteilte Software	75
2.6.1	Gründe für oder gegen Verteilung	76
2.6.2	Designmerkmale verteilter Architekturen	77
2.6.3	Ebenen für die Verteilung	78
2.6.4	Die wichtigsten Fragen für Architekten	79
2.6.5	Die »8 Fallacies of Distributed Computing«	81
2.7	Designmerkmale	83
2.7.1	Kopplung	83
2.7.2	Ausfallsicherheit	85
2.7.3	Performance	88
2.7.4	Sicherheit	93
2.7.5	Validierung	96
2.7.6	Internationalisierung/Lokalisierung	98
2.7.7	Statusinformationen	99
2.7.8	Interoperabilität und Integration	100
2.7.9	Die Admin-Sichtweise	104
2.7.10	Transaktionen und Gleichzeitigkeit (Concurrency)	108
2.7.11	Fehlerbehandlung	113
2.8	Architekturmodelle	116
2.8.1	Monolithische Anwendungen	116
2.8.2	Client-Server-Architektur	117
2.8.3	Mehrschichtige Anwendungen	117
2.8.4	Serviceorientierte Architekturen (SOA)	117
2.8.5	Weitere Architekturmodelle	121
2.9	Mobile Anwendungen	121
2.9.1	Definition und Einschränkungen	122
2.9.2	Entwicklungsansätze	122
2.9.3	Herausforderungen in der Architektur mobiler Anwendungen	125
2.10	Vorgehensweise	127
2.10.1	Schritt 1: Architekturziele definieren	128
2.10.2	Schritt 2: Umfeld analysieren	128
2.10.3	Schritt 3: Entwurf, Review und Iteration	129
2.10.4	Schritt 4: Technologieentscheidung	129
2.10.5	Schritt 5: Dokumentation	129
2.11	Dokumentation	129

2.11.1	Was eine gute Dokumentation auszeichnet	130
2.11.2	Modelle	131
2.11.3	Inhalt	134
2.12	Was noch zu sagen wäre	137

3 Softwaredesign 139

3.1	Grundlegende Designziele	140
3.1.1	Erfüllung der Funktion	140
3.1.2	Zuverlässigkeit und Robustheit	141
3.1.3	Wartbarkeit	141
3.1.4	Erweiterbarkeit	142
3.1.5	Die Bedürfnispyramide in der Softwareentwicklung	143
3.2	Objektorientierte Analyse und Design	144
3.2.1	Gründe und Aufgaben der OOA/OOD	145
3.2.2	Das Fallbeispiel	146
3.2.3	Abstraktion und Hierarchie	146
3.2.4	Objekte und Klassen	149
3.2.5	Beziehungen	159
3.2.6	War es das?	167
3.3	Designentscheidungen	168
3.3.1	Gutes Design, schlechtes Design	168
3.3.2	Exception-Handling	174
3.3.3	Logging	176
3.3.4	Datenmodell	176
3.4	Schnittstellen und Integration	177
3.4.1	Integration	179
3.4.2	Unterscheidungsmöglichkeiten	180
3.4.3	Mindeststandards	184
3.5	Benutzeroberflächen	187
3.5.1	Die richtige Technologie	189
3.5.2	Anforderungen an eine gute grafische Benutzeroberfläche	191
3.5.3	Gestaltungsregeln	201
3.5.4	Fallstudie: Entwicklung des Kalimba.KeyAccount Systems	217
3.5.5	Der »Rest«	220
3.6	Konfiguration	221
3.6.1	Grundlegendes zu Konfigurationen	222

3.6.2	.NET-Konfigurationsdateien	225
3.6.3	Datenbankkonfiguration	234
3.7	Vom Umgang mit der Zeit in Anwendungen	235
3.7.1	Dauer und Wahrnehmung	237
3.7.2	Anwenderfreundliche Informationsdialoge	243
3.7.3	Sonstige Tipps und Tricks	245
3.8	Tutorial: Die Enterprise Library	247
3.8.1	Der Logging Application Block	248
3.8.2	Der Exception Handling Application Block	256

4 .NET für Fortgeschrittene 263

4.1	Neues in C# 6	264
4.1.1	Zeichenfolgeninterpolation	264
4.1.2	Bedingter NULL-Operator	265
4.1.3	nameof	267
4.1.4	Verbesserte Eigenschaften	268
4.1.5	Expression Bodys in Methoden und Eigenschaften	270
4.1.6	Using static	271
4.1.7	Index-Initialisierer	271
4.1.8	Exception-Filter	272
4.1.9	Collection-Initialisierer, die über eine statische Add-Erweiterungsmethode verfügen	272
4.1.10	Await in catch- und finally-Blöcken	273
4.2	Parallele Verarbeitung	273
4.2.1	Wann lohnt sich parallele Verarbeitung überhaupt?	274
4.2.2	Parallelität in der Praxis	276
4.2.3	Was sind Threads?	279
4.2.4	Multithreading in .NET	280
4.2.5	Klassische Threads	281
4.2.6	Thread Pools	288
4.2.7	Timer	289
4.2.8	Task Parallel Library	290
4.2.9	Async und Await	304
4.2.10	Thread-Synchronisierung	312
4.3	Fehlerbehandlung	322
4.3.1	Was ist eine Exception?	323
4.3.2	Der Status	324

4.3.3	Try	324
4.3.4	Catch	325
4.3.5	Finally	334
4.3.6	System.Exception	335
4.3.7	Eigene Exception-Klassen	336
4.3.8	Zum Schluss	337
4.4	Reguläre Ausdrücke	339
4.4.1	Reguläre Ausdrücke in .NET	339
4.4.2	Alternative und Gruppierung	342
4.4.3	Nach reservierten Zeichen suchen	342
4.4.4	Zeichenauswahl	342
4.4.5	Quantifizierer	344
4.4.6	Kontextsensitive Bedingungen	345
4.4.7	Suchoptionen	346
4.4.8	Gruppen	347
4.5	Lambda-Ausdrücke	349
4.5.1	Delegaten	349
4.5.2	Lambda-Ausdrücke	350
4.5.3	Func- und Action-Delegaten	352
4.5.4	Lambda-Ausdrücke vs. anonyme Methoden	353
4.5.5	Expression Tree	353
4.5.6	Babylon revisited	355
4.6	Transaktionen	359
4.6.1	Wozu Transaktionen?	359
4.6.2	Merkmale von Transaktionen	359
4.6.3	TransactionScope	361
4.6.4	Committable Transaction	364
4.6.5	Tracing	366
4.7	Erweiterungsmethoden	367
4.7.1	Erweiterungsmethoden schreiben	367
4.7.2	Der Compiler	369
4.7.3	Vorteile, Nachteile, Empfehlungen	369
4.8	Serialisierung	371
4.8.1	Ein Beispiel	372
4.8.2	Serialisierer	374
4.8.3	BinaryFormatter	374
4.8.4	XmlSerializer	377
4.8.5	JSON	379
4.8.6	Sonstiges	380

4.9	Automatische Speicherverwaltung	380
4.9.1	Speicherzuteilung	381
4.9.2	Garbage Collection	381
4.9.3	Finalisierung	388
4.9.4	Monitoring	391
4.10	Besonderheiten und typische Fehler in C#	393
4.10.1	Gleitkommazahlen	393
4.10.2	Strings	395
4.10.3	NULL	395
4.10.4	Virtual	396
4.10.5	Java != C#	397
4.10.6	Teure Exceptions	398
4.10.7	Eigenschaften, (fast) überall	399
4.10.8	As und is	399
4.10.9	const vs. readonly	400
4.10.10	Parameter	401
4.11	Tutorial: Roslyn	402
4.11.1	Scripting	403
4.11.2	Fallbeispiel	404
4.11.3	1. Vorbereitungen	405
4.11.4	2. Projekte anlegen	406
4.11.5	3. Visual Studio starten und Debugee-Projekt erzeugen	407
4.11.6	4. Beispielcode erstellen	408
4.11.7	5a. Analyzer erstellen – Vorbereitungen	410
4.11.8	5b. Analyzer erstellen – syntaktische Analyse	412
4.11.9	5c. Analyzer erstellen – semantische Analyse	414
4.11.10	5d. Analyzer erstellen – Schließen der Verbindung suchen	414
4.11.11	6. Code-Fix erstellen	417
4.11.12	7. Code-Fix testen	420
5	Professionell codieren	423
<hr/>		
5.1	Was ist sauber und strukturiert?	424
5.1.1	Die grundlegenden Probleme	425
5.1.2	Was zeichnet guten Code aus?	427
5.2	Code-Styleguides	429
5.3	Gut benennen	430
5.3.1	Deutsch oder Englisch	431
5.3.2	Aussagekräftig	431

5.3.3	Einzahl oder Mehrzahl	434
5.3.4	camelCase	435
5.3.5	Leerwörter, Weasle-Words und reservierte Wörter	435
5.3.6	Feste Namenskonventionen	436
5.3.7	C#-Konventionen	436
5.4	Sauber formatieren	439
5.4.1	Struktur	439
5.4.2	Formatierung	443
5.5	Sinnvoll kommentieren	449
5.5.1	Selbstverständliches	451
5.5.2	Kürze und Prägnanz	451
5.5.3	// vs. /* */	451
5.5.4	//todo	452
5.5.5	Kommentare in Visual Studio	452
5.5.6	Ort	454
5.5.7	Die richtige Zeit	454
5.5.8	Aktualität	454
5.6	Klassen und Klassenhierarchien	454
5.6.1	Klasse oder Schnittstelle?	454
5.6.2	Klasse oder struct?	456
5.6.3	Klassengröße	457
5.6.4	Zuständigkeit	457
5.6.5	Erweiterbarkeit	458
5.6.6	Abstrakte Klassen	461
5.6.7	Statische Klassen	461
5.7	Funktionen	461
5.7.1	Funktionsgröße	461
5.7.2	Zuständigkeit	462
5.7.3	Konstruktoren	462
5.7.4	Eigenschaft oder Funktion?	463
5.7.5	Parameter	465
5.7.6	Erweiterungsmethoden	466
5.8	Schnittstellen	467
5.9	Enums	468
5.10	Eigenschaften	469
5.11	Exceptions	470
5.11.1	Wann?	470
5.11.2	Wo?	470
5.11.3	Wie?	471

5.12 Refactoring	474
5.12.1 Gründe	474
5.12.2 Code-Smells	475
5.12.3 Der Prozess	476
5.12.4 Tools	477
5.12.5 Refactoring-Muster	477
5.13 Visual Studio 2015: Tipps & Tricks für mehr Produktivität	483
5.13.1 Editionen	484
5.13.2 Voraussetzungen	484
5.13.3 Tools	486
5.13.4 Schneller starten	488
5.13.5 Visual Studio personalisieren	493
5.13.6 Fenster und Layouts	494
5.13.7 Der Editor	496
5.13.8 Navigieren	506
5.13.9 Debuggen	506
5.13.10 Projektmappen-Explorer	508
5.13.11 IntelliSense	510
5.14 Aus der Praxis: Codeanalyse in Visual Studio 2015	512
5.15 Tutorial: Snippet Designer	517

6 Windows Communication Foundation 525

6.1 Services im Überblick	526
6.2 Der erste WCF-Service	532
6.2.1 Das Projekt	532
6.2.2 Der Service	534
6.2.3 Die Konfigurationsdatei web.config	537
6.2.4 Der Client	538
6.3 Anatomie eines WCF-Service	543
6.3.1 Endpunkte	543
6.3.2 Adressen	551
6.3.3 Binding	552
6.3.4 Contracts	555
6.3.5 Konfiguration	556
6.3.6 Transportsitzung	561
6.4 Hosting	563
6.4.1 Merkmale eines gutes Hosts	564

6.4.2	Visual Studio 2015	565
6.4.3	Selfhosting	566
6.4.4	NT-Services	569
6.4.5	IIS	574
6.4.6	WAS	581
6.4.7	AppFabric	586
6.4.8	Entscheidungshilfe	589
6.5	Clients	589
6.5.1	Proxy erzeugen	590
6.5.2	Details zum erzeugten Proxy	594
6.5.3	Proxys verwenden	598
6.5.4	ChannelFactory	607
6.6	Services im Detail	608
6.6.1	Service Contracts	608
6.6.2	Data Contracts	615
6.6.3	Kommunikationsmuster	630
6.7	Fehlerbehandlung	637
6.7.1	Grundlagen	637
6.7.2	FaultException	640
6.7.3	SOAP Faults	642
6.8	Transaktionen	646
6.8.1	Verteilte Transaktionen	646
6.8.2	Transaktionen in WCF	651
6.8.3	Wegweiser	657
6.8.4	Ressourcenkonflikte	659
6.9	Instanzen	660
6.9.1	Instanziierungsmodi	661
6.9.2	Lastbegrenzung	670
6.10	Sicherheit	672
6.10.1	Einführung	673
6.10.2	Transportweg und Nachricht sichern	677
6.10.3	Detaillkonfiguration	679
6.10.4	Identität	683
6.10.5	Autorisierung	686
6.11	Aus der Praxis: WCF erweitern	689
6.11.1	Schritt 1: Das Projekt einrichten	690
6.11.2	Schritt 2: Das Transferobjekt	690
6.11.3	Schritt 3: Anhängen des Transferobjekts an eine Nachricht	693
6.11.4	Schritt 4: Der Client	695

6.11.5	Schritt 5: Der Service	696
6.11.6	Schritt 6: Konfiguration	696
6.11.7	Schritt 7: Test	697
6.12	Tutorial: Message Queuing	697
6.12.1	Einführung	697
6.12.2	Schritt 1: MSMQ installieren	699
6.12.3	Schritt 2: Queues anlegen	700
6.12.4	Schritt 3: Projekte einrichten	702
6.12.5	Schritt 4: CustomerPortalService	702
6.12.6	Schritt 5: SalesPortalService	703
6.12.7	Schritt 6: Implementierungen	704
6.12.8	Schritt 7: Hosting	707
6.12.9	Schritt 8: Konfiguration	709
6.12.10	Schritt 9: Tests	712
6.13	Web API	714
7	Datenbank und Datenzugriff	717
<hr/>		
7.1	.NET im SQL Server	719
7.1.1	Vorbereitungen	719
7.1.2	Benutzerdefinierte Datentypen	720
7.1.3	Sicherheit	727
7.1.4	Stored Procedures	728
7.1.5	Benutzerdefinierte Funktionen	730
7.1.6	Trigger	731
7.1.7	Benutzerdefinierte Aggregatfunktionen	734
7.1.8	Einsatz in der Praxis	736
7.2	XML in der Datenbank	739
7.2.1	Tabelle mit XML-Daten erzeugen	740
7.2.2	Daten hinzufügen	740
7.2.3	Daten auslesen	741
7.2.4	Indizes anlegen	742
7.2.5	Daten abfragen	743
7.2.6	Daten modifizieren	746
7.3	Volltextsuche	747
7.3.1	Installation	750
7.3.2	Volltextkatalog anlegen	751
7.3.3	Daten abfragen	754

7.4	Filestream	759
7.4.1	Filestream installieren	760
7.4.2	Filestream aktivieren	761
7.4.3	Datenbank für Filestream einrichten	762
7.4.4	Tabellen um Filestream-Spalten erweitern	763
7.4.5	Dateien ablegen mit SqlFileStream	766
7.4.6	Die Verwaltung der Filestream-Dateien	767
7.4.7	Dateien abrufen	768
7.4.8	Volltext und Filestream	769
7.4.9	Aus der Praxis	770
7.5	Das ADO.NET Entity Framework	771
7.5.1	Einführung	776
7.5.2	Projekt einrichten	781
7.5.3	Das Modell erweitern	783
7.5.4	Daten abfragen	790
7.5.5	Daten hinzufügen und ändern	793
7.5.6	SaveChanges und Gleichzeitigkeit	796
7.5.7	Was noch zu sagen wäre	796
7.6	LINQ to XML	797
7.6.1	Statische XML-Dateien erstellen	800
7.6.2	XML-Dateien aus vorhandenen Strukturen erstellen	801
7.6.3	Erweiterungsmethoden	802
7.6.4	XML-Dateien laden	802
7.6.5	Abfragen	805
7.6.6	XML-Daten verändern	806
7.6.7	Anwendung in der Praxis	806
7.7	Was noch zu sagen wäre	806

8 Workflow Foundation 809

8.1	Einführung	810
8.1.1	Warum Workflows?	816
8.1.2	Der Workflow	820
8.1.3	Workflow Designer	825
8.1.4	Windows Workflow Foundation im Überblick	828
8.2	Fallbeispiel	829
8.3	Der erste sequenzielle Workflow	830
8.3.1	Das Projekt einrichten	830

8.3.2	Den Workflow gestalten	831
8.3.3	Der weitere Ausbau	839
8.4	Der erste Flowchart-Workflow	840
8.4.1	Wareneingang reloaded	841
8.4.2	Den Wareneingangs-Workflow umbauen	842
8.5	Workflows laden und ausführen	846
8.5.1	Workflows in XAML ausführen	847
8.5.2	Workflows in Code ausführen	847
8.5.3	WorkflowApplication	848
8.6	Eingebaute Aktivitäten verwenden	850
8.6.1	Auflistung	850
8.6.2	Parallele Verarbeitung	856
8.6.3	Fehlerbehandlung	861
8.6.4	Ausführungssteuerung	864
8.6.5	Ereignissteuerung	871
8.6.6	TerminateWorkflow	877
8.6.7	Sonstige Aktivitäten	878
8.7	Eigene Aktivitäten entwickeln	878
8.7.1	Aktivitäten im Überblick	879
8.7.2	Lebenszyklus	883
8.7.3	CodeActivity	884
8.7.4	CodeActivity mit Rückgabewert	887
8.7.5	CodeActivity mit Validierung	889
8.7.6	NativeActivity	892
8.7.7	ActivityDesigner	897
8.7.8	Bookmarks	903
8.7.9	Was noch zu sagen wäre	904
8.8	Transaktionen	904
8.8.1	TransactionScope	906
8.8.2	Kompensationen	908
8.9	Persistenz	916
8.9.1	InstanceStore	916
8.9.2	SQL Server einrichten	917
8.9.3	Änderungen an der Workflow-Anwendung	917
8.9.4	Speichern im Code	919
8.9.5	Persistenzschutz	920
8.10	Tracking und Tracing	920
8.10.1	Tracking-Grundlagen	921
8.10.2	Tracking-Objekte	923

8.10.3	Fallbeispiel	924
8.10.4	Tracing	931
8.11	Workflow Services	934
8.11.1	Grundlagen	934
8.11.2	Aktivitäten	940
8.11.3	Fallbeispiel – Teil 1: Der Laborservice	942
8.11.4	Fallbeispiel – Teil 2: WF ruft WCF	950
8.11.5	Fallbeispiel – Teil 3: Der Laborclient	953
8.11.6	Fallbeispiel – Teil 4: WCF ruft WF	954
8.11.7	Fallbeispiel – Teil 5: Persistence	964
8.11.8	Correlation	966
8.11.9	Zum Schluss	969
8.12	State Machine Workflows	969
8.12.1	Anfangszustand	971
8.12.2	Endzustand	973
8.12.3	Zustände dazwischen	974
8.12.4	Zustandsübergänge	975
8.12.5	Übungsempfehlung	979
8.13	Designer Rehosting	980
8.13.1	Fallbeispiel	980
8.13.2	Den Designer einbinden	981
8.13.3	Die Toolbox bestücken	985

9 Windows-Apps und WinRT 989

9.1	Einführung	991
9.1.1	Laufzeitvoraussetzungen	992
9.1.2	Das Windows-8-Design	993
9.1.3	Deployment und der Windows App Store	1000
9.1.4	Prozesse in WinRT und das Windows Application Model	1000
9.2	Fallbeispiel	1002
9.3	Projekt einrichten	1003
9.3.1	Voraussetzungen	1003
9.3.2	Templates	1003
9.3.3	Projekt anlegen und einrichten	1007
9.4	Seiten hinzufügen	1010
9.4.1	Das Navigationskonzept	1010
9.4.2	Seiten hinzufügen	1011

9.4.3	Startseite festlegen	1013
9.4.4	Anwendung starten	1013
9.5	Daten hinzufügen	1014
9.5.1	Klassenmodell	1014
9.5.2	Von XML in Klassenhierarchie laden	1017
9.5.3	Ressource hinzufügen	1018
9.5.4	Daten beim Aufruf der App laden	1019
9.5.5	Daten an Steuerelement binden	1019
9.6	Die Lexikonsseite	1022
9.6.1	Allgemeines zur Navigation	1023
9.6.2	Navigation zur Lexikonsseite	1023
9.6.3	Lexikonsseite: Produkte anzeigen	1024
9.6.4	Lexikonsseite: Lexikoneintrag anzeigen	1027
9.7	Die Bestellseite und die App Bar	1032
9.7.1	App Bars in eigenen Anwendungen	1032
9.7.2	Eine App Bar hinzufügen	1033
9.8	Die Warenkorbseite	1034
9.8.1	Die Anzeige	1035
9.8.2	Die App Bar	1036
9.8.3	Änderungen am Datenmodell	1038
9.9	Die Bestellbestätigungsseite	1039
9.10	Lebenszyklus- und Zustandsmanagement	1041
9.10.1	Einführung	1041
9.10.2	Anwendungsdaten	1042
9.10.3	Sitzungsdaten – Framenavigation	1042
9.10.4	Sitzungsdaten – Zustand der Seiten	1048
9.11	Was noch zu sagen wäre	1049

10 Softwaretests 1051

10.1	Grundlagen	1053
10.1.1	Ziele und Aufgaben	1053
10.1.2	Übersicht und Einteilung der Tests	1056
10.1.3	Vom richtigen Zeitpunkt	1060
10.1.4	Der Tester und sein Team	1063
10.1.5	Der Testablauf	1067
10.1.6	Kleine Fehlerkunde	1075

10.2	Testplanung und -organisation	1080
10.2.1	Release-Management	1080
10.2.2	Das Testteam	1082
10.2.3	Testfälle	1086
10.2.4	Werkzeuge	1090
10.3	Testumgebung	1091
10.3.1	Voraussetzungen	1092
10.3.2	Die zu testende Software	1093
10.3.3	Daten	1093
10.3.4	Rechner und Betriebssystem	1096
10.3.5	Server- und Zusatzkomponenten	1097
10.3.6	Tools	1098
10.4	Testverfahren und -werkzeuge	1098
10.4.1	Exploratives Testen	1098
10.4.2	Test-to-pass vs. test-to-fail	1098
10.4.3	Äquivalenzklassenbildung	1099
10.4.4	Grenzwerte	1101
10.4.5	Sinnlose Daten	1103
10.4.6	Programmzustände	1103
10.4.7	Entscheidungstabellen	1104
10.4.8	Ablaufpläne	1105
10.4.9	Geschäftsprozessmodelle	1107
10.4.10	Continuous Delivery	1107
10.5	Testarten	1111
10.5.1	Test der Spezifikation	1111
10.5.2	Unit-Test	1114
10.5.3	Komponententest	1117
10.5.4	Usability-Test	1118
10.5.5	Systemtest	1121
10.5.6	Feldtest	1122
10.5.7	Abnahmetest	1124
10.5.8	Codereview	1125
10.5.9	Der Rest	1128
10.6	Workshop: Unit-Tests mit Visual Studio	1130
10.6.1	Anlegen eines Testprojekts	1130
10.6.2	Hinzufügen der Unit-Tests	1132
10.6.3	Codeabdeckung	1135
10.6.4	Praktische Empfehlungen	1137

11 Softwarepflege und Projektmanagement 1141

11.1 Release Management	1142
11.1.1 Begriffe	1142
11.1.2 Der Release-Prozess	1143
11.2 Anforderungen	1153
11.2.1 Einführung	1153
11.2.2 Die verschiedenen Sichtweisen	1155
11.2.3 Anforderungen an eine Anforderung	1157
11.3 Zeitschätzung	1162
11.3.1 Was ist eine Zeitschätzung?	1162
11.3.2 Herausforderungen einer Zeitschätzung	1164
11.3.3 Die lernende Organisation	1171
11.3.4 Woher kommen Zeitüberschreitungen?	1174
11.3.5 Methoden der Zeitschätzung	1177

12 Zum Schluss 1189

Index	1191
-------------	------