

Auf einen Blick

1	Einführung	25
2	Softwarearchitektur	37
3	Softwaredesign	131
4	.NET für Fortgeschrittene	241
5	Professionell codieren	361
6	Windows Communication Foundation	433
7	Datenbank und Datenzugriff	625
8	Workflow Foundation	731
9	Windows 8 und WinRT	913
10	Softwaretests	987
11	Softwarepflege	1077

Inhalt

Vorwort zur zweiten 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 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	51
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	63

2.5.3	Die Fassade	63
2.5.4	Presentation Layer	64
2.5.5	Business Layer	67
2.5.6	Data Layer	71
2.6	Tier und verteilte Software	76
2.6.1	Gründe für oder gegen Verteilung	76
2.6.2	Designmerkmale verteilter Architekturen	78
2.6.3	Ebenen für die Verteilung	79
2.6.4	Die wichtigsten Fragen für Architekten	80
2.7	Designmerkmale	81
2.7.1	Kopplung	82
2.7.2	Ausfallsicherheit	83
2.7.3	Performance	86
2.7.4	Sicherheit	91
2.7.5	Validierung	94
2.7.6	Lokalisierung	97
2.7.7	Statusinformationen	98
2.7.8	Interoperabilität und Integration	98
2.7.9	Die Admin-Sichtweise	102
2.7.10	Transaktionen und Gleichzeitigkeit (Concurrency)	106
2.7.11	Fehlerbehandlung	112
2.8	Architekturmodelle	114
2.8.1	Monolithische Anwendungen	114
2.8.2	Client-Server-Architektur	115
2.8.3	Mehrschichtige Anwendungen	115
2.8.4	Serviceorientierte Architekturen (SOA)	115
2.9	Vorgehensweise	119
2.9.1	Schritt 1: Architekturziele definieren	120
2.9.2	Schritt 2: Umfeld analysieren	120
2.9.3	Schritt 3: Entwurf, Review und Iteration	121
2.9.4	Schritt 4: Technologieentscheidung	121
2.9.5	Schritt 5: Dokumentation	121
2.10	Dokumentation	121
2.10.1	Was eine gute Dokumentation auszeichnet	122
2.10.2	Modelle	123
2.10.3	Inhalt	126
2.11	Was noch zu sagen wäre	129

3.1 Grundlegende Designziele	132
3.1.1 Erfüllung der Funktion	132
3.1.2 Zuverlässigkeit und Robustheit	133
3.1.3 Wartbarkeit	133
3.1.4 Erweiterbarkeit	134
3.2 Objektorientierte Analyse und Design	135
3.2.1 Gründe und Aufgaben der OOA/OOD	136
3.2.2 Das Fallbeispiel	137
3.2.3 Abstraktion und Hierarchie	138
3.2.4 Objekte und Klassen	140
3.2.5 Beziehungen	151
3.2.6 War es das?	159
3.3 Designentscheidungen	160
3.3.1 Gutes Design, schlechtes Design	160
3.3.2 Exception-Handling	167
3.3.3 Logging	169
3.3.4 Datenmodell	169
3.4 Schnittstellen und Integration	169
3.4.1 Integration	172
3.4.2 Unterscheidungsmöglichkeiten	173
3.4.3 Mindeststandards	176
3.5 Benutzeroberflächen	180
3.5.1 Die richtige Technologie	182
3.5.2 Anforderungen an eine gute grafische Benutzeroberfläche	183
3.5.3 Fallstudie: Entwicklung des Kalimba.KeyAccount Systems	192
3.5.4 Der »Rest«	196
3.6 Konfiguration	197
3.6.1 Grundlegendes zu Konfigurationen	198
3.6.2 .NET-Konfigurationsdateien	201
3.6.3 Datenbankkonfiguration	210
3.7 Vom Umgang mit der Zeit in Anwendungen	212
3.7.1 Dauer und Wahrnehmung	214
3.7.2 Anwenderfreundliche Informationsdialoge	219
3.7.3 Sonstige Tipps und Tricks	221
3.8 Tutorial: Die Enterprise Library	224
3.8.1 Der Logging Application Block	225
3.8.2 Der Exception Handling Application Block	233

4.1 Parallele Verarbeitung	241
4.1.1 Wann lohnt sich parallele Verarbeitung überhaupt?	242
4.1.2 Parallelität in der Praxis	244
4.1.3 Was sind Threads?	247
4.1.4 Multithreading in .NET	249
4.1.5 Klassische Threads	250
4.1.6 Thread Pools	257
4.1.7 Timer	258
4.1.8 Task Parallel Library	259
4.1.9 Async und Await	272
4.1.10 Thread-Synchronisierung	281
4.2 Fehlerbehandlung	291
4.2.1 Was ist eine Exception?	292
4.2.2 Der Status	293
4.2.3 Try	293
4.2.4 Catch	294
4.2.5 Finally	302
4.2.6 System.Exception	303
4.2.7 Eigene Exception-Klassen	304
4.2.8 Zum Schluss	305
4.3 Reguläre Ausdrücke	307
4.3.1 Reguläre Ausdrücke in .NET	308
4.3.2 Alternative und Gruppierung	310
4.3.3 Nach reservierten Zeichen suchen	310
4.3.4 Zeichenauswahl	311
4.3.5 Quantifizierer	312
4.3.6 Kontextsensitive Bedingungen	313
4.3.7 Suchoptionen	315
4.3.8 Gruppen	315
4.4 Lambda-Ausdrücke	317
4.4.1 Delegaten	317
4.4.2 Lambda-Ausdrücke	319
4.4.3 Func- und Action-Delegaten	320
4.4.4 Lambda-Ausdrücke vs. anonyme Methoden	321
4.4.5 Expression Tree	322
4.4.6 Babylon revisited	323

4.5 Transaktionen	327
4.5.1 Wozu Transaktionen?	327
4.5.2 Merkmale von Transaktionen	328
4.5.3 TransactionScope	329
4.5.4 Committable Transaction	332
4.5.5 Tracing	334
4.6 Erweiterungsmethoden	335
4.6.1 Erweiterungsmethoden schreiben	336
4.6.2 Der Compiler	337
4.6.3 Vorteile, Nachteile, Empfehlungen	338
4.7 Serialisierung	339
4.7.1 Ein Beispiel	340
4.7.2 Serialisierer	342
4.7.3 BinaryFormatter	343
4.7.4 XmlSerializer	345
4.8 Automatische Speicherverwaltung	348
4.8.1 Speicherzuteilung	348
4.8.2 Garbage Collection	349
4.8.3 Finalisierung	355
4.8.4 Monitoring	358

5 Professionell codieren	361
5.1 Was ist sauber und strukturiert?	362
5.1.1 Die grundlegenden Probleme	362
5.1.2 Was zeichnet guten Code aus?	365
5.2 Code-Styleguides	367
5.3 Gut benennen	368
5.3.1 Deutsch oder Englisch	369
5.3.2 Aussagekräftig	369
5.3.3 Einzahl oder Mehrzahl	372
5.3.4 camelCase	373
5.3.5 Leerwörter, Weasle-Words und reservierte Wörter	373
5.3.6 Feste Namenskonventionen	374
5.3.7 C#-Konventionen	374
5.4 Sauber formatieren	376
5.4.1 Struktur	376

5.4.2	Formatierung	380
5.5	Sinnvoll kommentieren	386
5.5.1	Selbstverständliches	388
5.5.2	Kürze und Prägnanz	388
5.5.3	// vs. /* */	388
5.5.4	//todo	389
5.5.5	Kommentare in Visual Studio	389
5.5.6	Ort	391
5.5.7	Die richtige Zeit	391
5.5.8	Aktualität	391
5.6	Klassen und Klassenhierarchien	391
5.6.1	Klasse oder Schnittstelle?	391
5.6.2	Klasse oder struct?	393
5.6.3	Klassengröße	394
5.6.4	Zuständigkeit	394
5.6.5	Erweiterbarkeit	395
5.6.6	Abstrakte Klassen	398
5.6.7	Statische Klassen	398
5.7	Funktionen	398
5.7.1	Funktionsgröße	399
5.7.2	Zuständigkeit	399
5.7.3	Konstruktoren	399
5.7.4	Eigenschaft oder Funktion?	400
5.7.5	Parameter	402
5.7.6	Erweiterungsmethoden	403
5.8	Schnittstellen	404
5.9	Enums	405
5.10	Eigenschaften	406
5.11	Exceptions	407
5.11.1	Wann?	408
5.11.2	Wo?	408
5.11.3	Wie?	409
5.12	Refactoring	411
5.12.1	Gründe	411
5.12.2	Code-Smells	412
5.12.3	Der Prozess	413
5.12.4	Tools	414
5.12.5	Refactoring-Muster	415

5.13 Aus der Praxis: Codeanalyse in Visual Studio 2012	420
5.14 Tutorial: Snippet Designer	425
<hr/>	
6 Windows Communication Foundation	433
<hr/>	
6.1 Services im Überblick	434
6.2 Der erste WCF-Service	440
6.2.1 Das Projekt	440
6.2.2 Der Service	442
6.2.3 Die Konfigurationsdatei web.config	445
6.2.4 Der Client	446
6.3 Anatomie eines WCF-Service	451
6.3.1 Endpunkte	451
6.3.2 Adressen	459
6.3.3 Binding	460
6.3.4 Contracts	463
6.3.5 Konfiguration	464
6.3.6 Transportsitzung	470
6.4 Hosting	471
6.4.1 Merkmale eines guten Hosts	472
6.4.2 Visual Studio 2012	473
6.4.3 Selfhosting	474
6.4.4 NT-Services	478
6.4.5 IIS	483
6.4.6 WAS	490
6.4.7 AppFabric	495
6.4.8 Entscheidungshilfe	498
6.5 Clients	499
6.5.1 Proxy erzeugen	499
6.5.2 Details zum erzeugten Proxy	504
6.5.3 Proxys verwenden	507
6.5.4 ChannelFactory	514
6.6 Services im Detail	516
6.6.1 Service Contracts	516
6.6.2 Data Contracts	523
6.6.3 Kommunikationsmuster	538

6.7 Fehlerbehandlung	546
6.7.1 Grundlagen	546
6.7.2 FaultException	549
6.7.3 SOAP Faults	550
6.8 Transaktionen	554
6.8.1 Verteilte Transaktionen	555
6.8.2 Transaktionen in WCF	560
6.8.3 Wegweiser	566
6.8.4 Ressourcenkonflikte	568
6.9 Instanzen	569
6.9.1 Instanzierungsmodi	570
6.9.2 Lastbegrenzung	579
6.10 Sicherheit	581
6.10.1 Einführung	582
6.10.2 Transportweg und Nachricht sichern	586
6.10.3 Detailkonfiguration	588
6.10.4 Identität	592
6.10.5 Autorisierung	595
6.11 Aus der Praxis: WCF erweitern	598
6.11.1 Schritt 1: Das Projekt einrichten	599
6.11.2 Schritt 2: Das Transferobjekt	599
6.11.3 Schritt 3: Anhängen des Transferobjekts an eine Nachricht	602
6.11.4 Schritt 4: Der Client	604
6.11.5 Schritt 5: Der Service	605
6.11.6 Schritt 6: Konfiguration	605
6.11.7 Schritt 7: Test	606
6.12 Tutorial: Message Queuing	607
6.12.1 Einführung	607
6.12.2 Schritt 1: MSMQ installieren	609
6.12.3 Schritt 2: Queues anlegen	609
6.12.4 Schritt 3: Projekte einrichten	611
6.12.5 Schritt 4: CustomerPortalService	611
6.12.6 Schritt 5: SalesPortalService	612
6.12.7 Schritt 6: Implementierungen	613
6.12.8 Schritt 7: Hosting	616
6.12.9 Schritt 8: Konfiguration	619
6.12.10 Schritt 9: Tests	621

7.1 .NET im SQL Server	627
7.1.1 Vorbereitungen	627
7.1.2 Benutzerdefinierte Datentypen	628
7.1.3 Sicherheit	635
7.1.4 Stored Procedures	636
7.1.5 Benutzerdefinierte Funktionen	638
7.1.6 Trigger	639
7.1.7 Benutzerdefinierte Aggregatfunktionen	642
7.1.8 Einsatz in der Praxis	645
7.2 XML in der Datenbank	647
7.2.1 Tabelle mit XML-Daten erzeugen	648
7.2.2 Daten hinzufügen	648
7.2.3 Daten auslesen	649
7.2.4 Indizes anlegen	651
7.2.5 Daten abfragen	651
7.2.6 Daten modifizieren	654
7.3 Volltextsuche	656
7.3.1 Installation	658
7.3.2 Volltextkatalog anlegen	659
7.3.3 Daten abfragen	662
7.4 FileStream	667
7.4.1 FileStream installieren	668
7.4.2 FileStream aktivieren	669
7.4.3 Datenbank für FileStream einrichten	669
7.4.4 Tabellen um FileStream-Spalten erweitern	671
7.4.5 Dateien ablegen mit SqlFileStream	672
7.4.6 Die Verwaltung der FileStream-Dateien	674
7.4.7 Dateien abrufen	675
7.4.8 Volltext und FileStream	676
7.4.9 Aus der Praxis	677
7.5 Das ADO.NET Entity Framework	678
7.5.1 Einführung	679
7.5.2 Projekt einrichten	685
7.5.3 Das Modell erweitern	689
7.5.4 Daten abfragen	691
7.5.5 Daten hinzufügen und ändern	698
7.5.6 SaveChanges und Gleichzeitigkeit	701
7.5.7 Was noch zu sagen wäre	704

7.6 WCF Data Services	704
7.6.1 Übersicht	705
7.6.2 Einfachen WCF Data Service erstellen	706
7.6.3 WCF Data Service testen	708
7.6.4 Zugriff aus einer .NET-Anwendung	710
7.6.5 Empfehlungen für den Einsatz	716
7.7 LINQ to XML	719
7.7.1 Statische XML-Dateien erstellen	719
7.7.2 XML-Dateien aus vorhandenen Strukturen erstellen	722
7.7.3 Erweiterungsmethoden	723
7.7.4 XML-Dateien laden	724
7.7.5 Abfragen	725
7.7.6 XML-Daten verändern	727
7.7.7 Anwendung in der Praxis	728
7.8 Was noch zu sagen wäre	728

8 Workflow Foundation 731

8.1 Einführung	731
8.1.1 Warum Workflows?	732
8.1.2 Der Workflow	738
8.1.3 Workflow Designer	742
8.1.4 Windows Workflow Foundation im Überblick	748
8.2 Fallbeispiel	751
8.3 Der erste sequenzielle Workflow	752
8.3.1 Das Projekt einrichten	752
8.3.2 Den Workflow gestalten	754
8.3.3 Der weitere Ausbau	761
8.4 Der erste Flowchart-Workflow	763
8.4.1 Wareneingang reloaded	763
8.4.2 Den Wareneingangs-Workflow umbauen	764
8.5 Workflows laden und ausführen	768
8.5.1 Workflows in XAML ausführen	769
8.5.2 Workflows in Code ausführen	769
8.5.3 WorkflowApplication	770
8.6 Eingebaute Aktivitäten verwenden	772
8.6.1 Auflistung	772

8.6.2	Parallele Verarbeitung	778
8.6.3	Fehlerbehandlung	783
8.6.4	Ausführungssteuerung	786
8.6.5	Ereignissesteuerung	793
8.6.6	TerminateWorkflow	799
8.6.7	Sonstige Aktivitäten	800
8.7	Eigene Aktivitäten entwickeln	800
8.7.1	Aktivitäten im Überblick	801
8.7.2	Lebenszyklus	805
8.7.3	CodeActivity	806
8.7.4	CodeActivity mit Rückgabewert	810
8.7.5	CodeActivity mit Validierung	811
8.7.6	NativeActivity	815
8.7.7	ActivityDesigner	820
8.7.8	Bookmarks	825
8.7.9	Was noch zu sagen wäre	827
8.8	Transaktionen	827
8.8.1	TransactionScope	828
8.8.2	Kompensationen	831
8.9	Persistenz	838
8.9.1	InstanceStore	839
8.9.2	SQL Server einrichten	839
8.9.3	Änderungen an der Workflow-Anwendung	840
8.9.4	Speichern im Code	842
8.9.5	Persistenzschutz	842
8.10	Tracking und Tracing	843
8.10.1	Tracking-Grundlagen	843
8.10.2	Tracking-Objekte	846
8.10.3	Fallbeispiel	847
8.10.4	Tracing	854
8.11	Workflow Services	857
8.11.1	Grundlagen	857
8.11.2	Aktivitäten	863
8.11.3	Fallbeispiel – Teil 1: Der Laborservice	865
8.11.4	Fallbeispiel – Teil 2: WF ruft WCF	873
8.11.5	Fallbeispiel – Teil 3: Der Laborclient	876
8.11.6	Fallbeispiel – Teil 4: WCF ruft WF	877
8.11.7	Fallbeispiel – Teil 5: Persistence	888
8.11.8	Correlation	890
8.11.9	Zum Schluss	892

8.12 State Machine Workflows	893
8.12.1 Anfangszustand	895
8.12.2 Endzustand	897
8.12.3 Zustände dazwischen	898
8.12.4 Zustandsübergänge	899
8.12.5 Übungsempfehlung	903
8.13 Designer Rehosting	904
8.13.1 Fallbeispiel	905
8.13.2 Den Designer einbinden	905
8.13.3 Die Toolbox bestücken	909

9 Windows 8 und WinRT 913

9.1 Einführung	915
9.1.1 Laufzeitvoraussetzungen	915
9.1.2 Das Windows 8-Design	917
9.1.3 Deployment und der Windows 8 App Store	922
9.1.4 Prozesse in WinRT und das Windows Application Model	923
9.2 Fallbeispiel	925
9.3 Projekt einrichten	926
9.3.1 Voraussetzungen	926
9.3.2 Templates	926
9.3.3 Projekt anlegen und einrichten	929
9.4 Seiten hinzufügen	933
9.4.1 Das Navigationskonzept	934
9.4.2 Seiten hinzufügen	934
9.4.3 Startseite festlegen	936
9.4.4 Anwendung starten	937
9.5 Daten hinzufügen	937
9.5.1 Klassenmodell	938
9.5.2 Von XML in Klassenhierarchie laden	940
9.5.3 Ressource hinzufügen	941
9.5.4 Daten beim Aufruf der App laden	942
9.5.5 Daten an Steuerelement binden	943
9.6 Die Lexikonseite	945
9.6.1 Allgemeines zur Navigation	945
9.6.2 Navigation zur Lexikonseite	946

9.6.3	Lexikonseite: Produkte anzeigen	947
9.6.4	Lexikonseite: Lexikoneintrag anzeigen	950
9.7	Die Bestellseite und die App Bar	954
9.7.1	App Bars in eigenen Anwendungen	954
9.7.2	Eine App Bar hinzufügen	955
9.8	Die Warenkorbseite	957
9.8.1	Die Anzeige	957
9.8.2	Die App Bar	958
9.8.3	Änderungen am Datenmodell	959
9.9	Die Bestellbestätigungsseite	961
9.10	Lebenszyklus- und Zustandsmanagement	962
9.10.1	Einführung	963
9.10.2	Anwendungsdaten	963
9.10.3	Sitzungsdaten – Framenavigation	964
9.10.4	Sitzungsdaten – Zustand der Seiten	968
9.11	Contracts	969
9.11.1	Die Suche	970
9.11.2	Die Suche implementieren	971
9.11.3	Testen	976
9.12	Für verschiedene Layouts entwickeln	977
9.12.1	Der Simulator	977
9.12.2	Verschiedene Formate	979
9.12.3	Funktionsweise	983
9.12.4	Empfehlungen	984

10 Softwaretests 987

10.1	Grundlagen	989
10.1.1	Ziele und Aufgaben	989
10.1.2	Übersicht und Einteilung der Tests	992
10.1.3	Vom richtigen Zeitpunkt	996
10.1.4	Der Tester und sein Team	999
10.1.5	Der Testablauf	1003
10.1.6	Kleine Fehlerkunde	1011
10.2	Testplanung und -organisation	1017
10.2.1	Release-Management	1018

10.2.2	Das Testteam	1019
10.2.3	Testfälle	1024
10.3	Testumgebung	1028
10.3.1	Voraussetzungen	1028
10.3.2	Die zu testende Software	1029
10.3.3	Daten	1030
10.3.4	Rechner und Betriebssystem	1032
10.3.5	Server- und Zusatzkomponenten	1034
10.3.6	Tools	1034
10.4	Testverfahren und -werkzeuge	1034
10.4.1	Exploratives Testen	1035
10.4.2	Test-to-pass vs. test-to-fail	1035
10.4.3	Äquivalenzklassenbildung	1036
10.4.4	Grenzwerte	1037
10.4.5	Sinnlose Daten	1039
10.4.6	Programmzustände	1040
10.4.7	Entscheidungstabellen	1041
10.4.8	Ablaufpläne	1042
10.4.9	Geschäftsprozessmodelle	1043
10.4.10	Continuous Delivery	1043
10.5	Testarten	1048
10.5.1	Test der Spezifikation	1048
10.5.2	Unit-Test	1050
10.5.3	Komponententest	1053
10.5.4	Usability-Test	1054
10.5.5	Systemtest	1057
10.5.6	Feldtest	1058
10.5.7	Abnahmetest	1060
10.5.8	Codereview	1061
10.5.9	Der Rest	1064
10.6	Workshop: Unit-Tests mit Visual Studio	1066
10.6.1	Anlegen eines Testprojekts	1066
10.6.2	Hinzufügen der Unit-Tests	1068
10.6.3	Codeabdeckung	1071
10.6.4	Praktische Empfehlungen	1073

11.1 Release Management	1078
11.1.1 Begriffe	1078
11.1.2 Der Release-Prozess	1079
11.2 Anforderungen	1087
11.2.1 Einführung	1088
11.2.2 Die verschiedenen Sichtweisen	1090
11.2.3 Anforderungen an eine Anforderung	1092
11.3 Zeitschätzung	1096
11.3.1 Was ist eine Zeitschätzung?	1097
11.3.2 Herausforderungen einer Zeitschätzung	1098
11.3.3 Die lernende Organisation	1106
11.3.4 Woher kommen Zeitüberschreitungen?	1108
11.3.5 Methoden der Zeitschätzung	1112
Zum Schluss	1123
Index	1125