

Inhalt

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

1 Einführung 19

1.1	Lehre und Praxis – der Unterschied	19
1.1.1	Gute Software, schlechte Software	20
1.1.2	Wege zur Lösung	21
1.2	Das Fallbeispiel	22
1.3	Die einzelnen Kapitel	24

2 Architekturentwurf 31

2.1	Einführung	32
2.1.1	Das Problem	32
2.1.2	Gute Softwarearchitektur, schlechte Softwarearchitektur	33
2.1.3	Aufgaben	34
2.1.4	Anwendungstypen	35
2.1.5	Der Architekt	36
2.2	Anforderungen	37
2.2.1	Arten von Anforderungen	38
2.2.2	Anatomie einer Anforderung	40
2.2.3	Das richtige Maß	42
2.3	Komponenten	43
2.3.1	Komponenten identifizieren	43
2.3.2	Beziehungen	45
2.4	Prozesse	46
2.4.1	Was ist ein Prozess?	46
2.4.2	Geschäftsprozessmodellierung	47
2.4.3	Auswirkungen auf die Architektur	48
2.5	Layer (Schichten)	49
2.5.1	Grundlagen	49
2.5.2	Layer vs. Tier	51
2.5.3	Presentation Layer	51
2.5.4	Business Layer	54
2.5.5	Data Layer	56
2.6	Tier und verteilte Software	60
2.6.1	Gründe für oder gegen Verteilung	61

2.6.2	Designmerkmale verteilter Architekturen	62
2.6.3	Ebenen für die Verteilung	63
2.7	Designmerkmale	64
2.7.1	Kopplung	64
2.7.2	Ausfallsicherheit	65
2.7.3	Performance	68
2.7.4	Sicherheit	73
2.7.5	Validierung	75
2.7.6	Lokalisierung	78
2.7.7	Statusinformationen	79
2.7.8	Interoperabilität und Integration	79
2.7.9	Die Admin-Sichtweise	83
2.7.10	Transaktionen und Gleichzeitigkeit (Concurrency)	86
2.7.11	Fehlerbehandlung	91
2.8	Architekturmodelle	94
2.8.1	Monolithische Anwendungen	94
2.8.2	Client/Server	95
2.8.3	Mehrschichtige Anwendungen	95
2.8.4	Serviceorientierte Architekturen (SOA)	95
2.9	Vorgehensweise	99
2.9.1	Schritt 1: Architekturziele definieren	100
2.9.2	Schritt 2: Umfeld analysieren	100
2.9.3	Schritt 3: Entwurf, Review und Iteration	100
2.9.4	Schritt 4: Technologieentscheidung	101
2.9.5	Schritt 5: Dokumentation	101
2.10	Dokumentation	101
2.10.1	Was eine gute Dokumentation auszeichnet	102
2.10.2	Modelle	102
2.11	Was noch zu sagen wäre	106



3.1	Grundlegende Designziele	108
3.1.1	Erfüllung der Funktion	108
3.1.2	Zuverlässigkeit und Robustheit	109
3.1.3	Wartbarkeit	109
3.1.4	Erweiterbarkeit	110
3.2	Objektorientierte Analyse und Design	110
3.2.1	Gründe und Aufgaben der OOA/OOD	111
3.2.2	Das Fallbeispiel	112
3.2.3	Abstraktion und Hierarchie	112

3.2.4	Objekte und Klassen	114
3.2.5	Beziehungen	123
3.2.6	War es das?	130
3.3	Designentscheidungen	131
3.3.1	Gutes Design, schlechtes Design	131
3.3.2	Exception-Handling	135
3.3.3	Logging	137
3.3.4	Datenmodell	137
3.4	Schnittstellen und Integration	137
3.4.1	Integration	140
3.4.2	Unterscheidungsmöglichkeiten	141
3.4.3	Mindeststandards	144
3.5	Benutzeroberflächen	148
3.5.1	Die richtige Technologie	149
3.5.2	Anforderungen an eine gute grafische Benutzeroberfläche	151
3.5.3	Fallstudie: Entwicklung des Kalimba.KeyAccount Systems	159
3.6	Konfiguration	163
3.6.1	Grundlegendes zu Konfigurationen	164
3.6.2	.NET-Konfigurationsdateien	166
3.6.3	Datenbankkonfiguration	170
3.7	Vom Umgang mit der Zeit in Anwendungen	171
3.7.1	Dauer und Wahrnehmung	173
3.7.2	Anwenderfreundliche Informationsdialoge	177
3.8	Tutorial: Die Enterprise Library	179
3.8.1	Der Logging Application Block	180
3.8.2	Der Exception Handling Application Block	187

4. NET in Fortgeschrittenen 188

4.1	Parallele Verarbeitung	194
4.1.1	Wann lohnt sich parallele Verarbeitung überhaupt?	194
4.1.2	Parallelität in der Praxis	195
4.1.3	Was sind Threads?	197
4.1.4	Multithreading in .NET	198
4.1.5	Klassische Threads	199
4.1.6	Thread Pools	206
4.1.7	Timer	207
4.1.8	Task Parallel Library	207
4.1.9	Thread-Synchronisierung	216

4.2	Fehlerbehandlung	227
4.2.1	Was ist eine Exception?	228
4.2.2	Der Status	229
4.2.3	Try	229
4.2.4	Catch	230
4.2.5	Finally	238
4.2.6	System.Exception	239
4.2.7	Eigene Exception-Klassen	240
4.2.8	Zum Schluss	241
4.3	Reguläre Ausdrücke	242
4.3.1	Reguläre Ausdrücke in .NET	243
4.3.2	Alternative und Gruppierung	244
4.3.3	Nach reservierten Zeichen suchen	245
4.3.4	Zeichenauswahl	245
4.3.5	Quantifizierer	246
4.3.6	Kontextsensitive Bedingungen	248
4.3.7	Suchoptionen	249
4.3.8	Gruppen	250
4.4	Lambda-Ausdrücke	251
4.4.1	Delegaten	252
4.4.2	Lambda-Ausdrücke	253
4.4.3	Func- und Action-Delegaten	254
4.4.4	Lambda-Ausdrücke vs. anonyme Methoden	255
4.4.5	Expression Tree	255
4.4.6	Babylon revisited	257
4.5	Transaktionen	260
4.5.1	Wozu Transaktionen?	260
4.5.2	Merkmale von Transaktionen	261
4.5.3	TransactionScope	262
4.5.4	Committable Transaction	265
4.5.5	Tracing	266
4.6	Erweiterungsmethoden	267
4.6.1	Erweiterungsmethoden schreiben	268
4.6.2	Der Compiler	270
4.6.3	Vorteile, Nachteile, Empfehlungen	270
4.7	Serialisierung	271
4.7.1	Ein Beispiel	272
4.7.2	Serialisierer	274
4.7.3	BinaryFormatter	275
4.7.4	XmlSerializer	278

4.8	Automatische Speicherverwaltung	280
4.8.1	Speicherzuteilung	281
4.8.2	Garbage Collection	282
4.8.3	Finalisierung	287
4.8.4	Monitoring	291

5 Professionell codieren 293

5.1	Was ist sauber und strukturiert?	294
5.1.1	Die grundlegenden Probleme	294
5.1.2	Was zeichnet guten Code aus?	297
5.2	Code Style Guides	298
5.3	Gut benennen	300
5.3.1	Deutsch oder Englisch	300
5.3.2	Aussagekräftig	301
5.3.3	Einzahl oder Mehrzahl	303
5.3.4	CamelCase	303
5.3.5	Leerwörter und reservierte Wörter	304
5.3.6	Feste Namenskonventionen	304
5.3.7	C#-Konventionen	304
5.4	Sauber formatieren	305
5.4.1	Struktur	306
5.4.2	Formatierung	309
5.5	Sinnvoll kommentieren	314
5.5.1	Selbstverständliches	315
5.5.2	Kürze und Prägnanz	315
5.5.3	// vs. /* */	315
5.5.4	//todo	316
5.5.5	Kommentare in Visual Studio	316
5.5.6	Ort	318
5.5.7	Die richtige Zeit	318
5.5.8	Aktualität	318
5.6	Klassen und Klassenhierarchien	318
5.6.1	Klasse oder Schnittstelle?	319
5.6.2	Klasse oder struct?	320
5.6.3	Klassengröße	321
5.6.4	Zuständigkeit	321
5.6.5	Erweiterbarkeit	322
5.6.6	Abstrakte Klassen	324
5.6.7	Statische Klassen	324

5.7	Funktionen	324
5.7.1	Funktionsgröße	325
5.7.2	Zuständigkeit	325
5.7.3	Konstruktoren	325
5.7.4	Eigenschaft oder Funktion?	326
5.7.5	Parameter	327
5.7.6	Erweiterungsmethoden	328
5.8	Schnittstellen	329
5.9	Enums	330
5.10	Eigenschaften	331
5.11	Exceptions	332
5.11.1	Wann?	332
5.11.2	Wo?	332
5.11.3	Wie?	333
5.12	Refactoring	334
5.12.1	Gründe	335
5.12.2	Code Smells	335
5.12.3	Der Prozess	336
5.12.4	Tools	337
5.12.5	Refactoring-Muster	338
5.13	Aus der Praxis: FxCop	343
5.13.1	Design Rules	347
5.13.2	Globalization Rules	348
5.13.3	Naming Rules	349
5.13.4	Zum Schluss	350
5.14	Tutorial: Code Snippets	350

6 Windows Communication Foundation 357

6.1	Services im Überblick	358
6.2	Der erste WCF-Service	364
6.2.1	Das Projekt	364
6.2.2	Der Service	367
6.2.3	Die Konfigurationsdatei web.config	370
6.2.4	Der Client	370
6.3	Anatomie eines WCF-Service	375
6.3.1	Endpunkte	375
6.3.2	Adressen	382
6.3.3	Binding	383
6.3.4	Contracts	386

6.3.5	Konfiguration	387
6.3.6	Transportsitzung	393
6.4	Hosting	394
6.4.1	Merkmale eines gutes Hosts	395
6.4.2	Visual Studio 2010	396
6.4.3	Selfhosting	397
6.4.4	NT-Services	400
6.4.5	IIS	405
6.4.6	WAS	412
6.4.7	AppFabric	417
6.4.8	Entscheidungshilfe	418
6.5	Clients	419
6.5.1	Proxy erzeugen	419
6.5.2	Details zum erzeugten Proxy	423
6.5.3	Proxys verwenden	426
6.5.4	ChannelFactory	428
6.6	Services im Detail	429
6.6.1	Service Contracts	430
6.6.2	Data Contracts	436
6.6.3	Kommunikationsmuster	450
6.7	Fehlerbehandlung	457
6.7.1	Grundlagen	457
6.7.2	FaultException	460
6.7.3	SOAP Faults	461
6.8	Transaktionen	465
6.8.1	Verteilte Transaktionen	466
6.8.2	Transaktionen in WCF	471
6.8.3	Wegweiser	477
6.8.4	Ressourcenkonflikte	479
6.9	Instanzen	480
6.9.1	Instanziierungsmodi	481
6.9.2	Lastbegrenzung	490
6.10	Sicherheit	492
6.10.1	Einführung	493
6.10.2	Transportweg und Nachricht sichern	497
6.10.3	Detaillkonfiguration	499
6.10.4	Identität	503
6.10.5	Autorisierung	506
6.11	Aus der Praxis: WCF erweitern	510
6.11.1	Schritt 1: Das Projekt einrichten	510

6.11.2	Schritt 2: Das Transferobjekt	511
6.11.3	Schritt 3: Anhängen des Transferobjekts an eine Nachricht	513
6.11.4	Schritt 4: Der Client	516
6.11.5	Schritt 5: Der Service	516
6.11.6	Schritt 6: Konfiguration	517
6.11.7	Schritt 7: Test	517
6.12	Tutorial: Message Queuing	518
6.12.1	Einführung	518
6.12.2	Schritt 1: MSMQ installieren	520
6.12.3	Schritt 2: Queues anlegen	521
6.12.4	Schritt 3: Projekte einrichten	522
6.12.5	Schritt 4: CustomerPortalService	523
6.12.6	Schritt 5: SalesPortalService	523
6.12.7	Schritt 6: Implementierungen	525
6.12.8	Schritt 7: Hosting	528
6.12.9	Schritt 8: Konfiguration	530
6.12.10	Schritt 9: Tests	532

7 Datenbank und Datenzugriff 535

7.1	.NET im SQL Server	536
7.1.1	Vorbereitungen	537
7.1.2	Benutzerdefinierte Datentypen	537
7.1.3	Sicherheit	542
7.1.4	Stored Procedures	543
7.1.5	Benutzerdefinierte Funktionen	545
7.1.6	Trigger	546
7.1.7	Benutzerdefinierte Aggregatfunktionen	549
7.1.8	Einsatz in der Praxis	551
7.2	XML in der Datenbank	554
7.2.1	Tabelle mit XML-Daten erzeugen	555
7.2.2	Daten hinzufügen	555
7.2.3	Daten auslesen	556
7.2.4	Indizes anlegen	557
7.2.5	Daten abfragen	558
7.2.6	Daten modifizieren	561
7.3	Volltextsuche	562
7.3.1	Installation	564
7.3.2	Volltextkatalog anlegen	565
7.3.3	Daten abfragen	569

7.4	Das ADO.NET Entity Framework	573
7.4.1	Einführung	573
7.4.2	Reverse Engineering	574
7.4.3	Daten abfragen	577
7.4.4	Daten hinzufügen und ändern	579
7.4.5	Änderungen speichern	581
7.4.6	Laden von Entitäten	582
7.4.7	Was noch zu sagen wäre	584
7.5	WCF Data Services	585
7.5.1	Übersicht	586
7.5.2	Einfachen WCF-Data Service erstellen	587
7.5.3	WCF Data Service testen	588
7.5.4	Zugriff aus einer .NET-Anwendung	591
7.5.5	Empfehlungen für den Einsatz	597
7.6	LINQ to XML	598
7.6.1	Statische XML-Dateien erstellen	599
7.6.2	XML-Dateien aus vorhandenen Strukturen erstellen.....	602
7.6.3	Erweiterungsmethoden	603
7.6.4	XML-Dateien laden	604
7.6.5	Abfragen	604
7.6.6	XML-Daten verändern	606
7.6.7	Anwendung in der Praxis	608
7.7	Datenbanktools in Visual Studio 2010	608
7.7.1	Schemavergleich	608
7.7.2	Datenvergleich	612
7.8	Was noch zu sagen wäre	613

8 Workflow Foundation 615

8.1	Einführung	615
8.1.1	Warum Workflows?	616
8.1.2	Der Workflow	621
8.1.3	Windows Workflow Foundation im Überblick.....	625
8.2	Fallbeispiel	629
8.3	Der erste sequenzielle Workflow	631
8.3.1	Das Projekt einrichten	633
8.3.2	Den Workflow gestalten	634
8.3.3	Der weitere Ausbau	642
8.4	Der erste Flowchart-Workflow	643
8.4.1	Wareneingang reloaded	644
8.4.2	Den Wareneingangs-Workflow umbauen	645

8.5	Workflows laden und ausführen	649
8.5.1	Workflows in XAML ausführen	650
8.5.2	Workflows in Code ausführen	650
8.5.3	Starten von externen XAML-Workflows	651
8.5.4	WorkflowApplication	651
8.6	Eingebaute Aktivitäten verwenden	653
8.6.1	Auflistung	654
8.6.2	Parallele Verarbeitung	660
8.6.3	Fehlerbehandlung	665
8.6.4	Ausführungssteuerung	669
8.6.5	Ereignissteuerung	676
8.6.6	TerminateWorkflow	682
8.6.7	Sonstige Aktivitäten	683
8.7	Eigene Aktivitäten entwickeln	683
8.7.1	Aktivitäten im Überblick	684
8.7.2	Lebenszyklus	687
8.7.3	CodeActivity	688
8.7.4	CodeActivity mit Rückgabewert	691
8.7.5	CodeActivity mit Validierung	692
8.7.6	NativeActivity	696
8.7.7	ActivityDesigner	701
8.7.8	Bookmarks	706
8.7.9	Was noch zu sagen wäre	708
8.8	Transaktionen	708
8.8.1	TransactionScope	710
8.8.2	Kompensationen	712
8.9	Persistenz	718
8.9.1	InstanceStore	719
8.9.2	SQL Server einrichten	719
8.9.3	Änderungen an der Workflow-Anwendung	720
8.9.4	Speichern im Code	722
8.10	Tracking und Tracing	722
8.10.1	Tracking-Grundlagen	723
8.10.2	Tracking-Objekte	725
8.10.3	Fallbeispiel	726
8.10.4	Tracing	732
8.11	Workflow Services	735
8.11.1	Grundlagen	735
8.11.2	Aktivitäten	741
8.11.3	Fallbeispiel – Teil 1: Der Laborservice	744

8.11.4	Fallbeispiel – Teil 2: WF ruft WCF	751
8.11.5	Fallbeispiel – Teil 3: Der Labor-Client	754
8.11.6	Fallbeispiel – Teil 4: WCF ruft WF	755
8.11.7	Fallbeispiel – Teil 5: Persistence und Correlation	765
8.11.8	Zum Schluss	769
8.12	Designer Rehosting	769
8.12.1	Fallbeispiel	770
8.12.2	Den Designer einbinden	770
8.12.3	Den Workflow laden	772
8.12.4	Die Toolbox bestücken	773

9 Softwaretests 775

9.1	Grundlagen	776
9.1.1	Ziele und Aufgaben	776
9.1.2	Übersicht und Einteilung der Tests	779
9.1.3	Vom richtigen Zeitpunkt	782
9.1.4	Der Tester und sein Team	785
9.1.5	Der Testablauf	789
9.1.6	Kleine Fehlerkunde	796
9.2	Testplanung und -organisation	798
9.2.1	Release-Management	798
9.2.2	Das Testteam	800
9.2.3	Testfälle	803
9.3	Testumgebung	806
9.3.1	Voraussetzungen	806
9.3.2	Die zu testende Software	808
9.3.3	Daten	808
9.3.4	Rechner und Betriebssystem	810
9.3.5	Server- und Zusatzkomponenten	811
9.3.6	Tools	812
9.4	Testverfahren und -werkzeuge	812
9.4.1	Exploratives Testen	812
9.4.2	Test-to-pass vs. test-to-fail	813
9.4.3	Äquivalenzklassenbildung	813
9.4.4	Grenzwerte	815
9.4.5	Sinnlose Daten	817
9.4.6	Programmzustände	817
9.4.7	Entscheidungstabellen	818
9.4.8	Ablaufpläne	819
9.4.9	Geschäftsprozessmodelle	820

9.5	Testarten	820
9.5.1	Test der Spezifikation	821
9.5.2	Komponententest	823
9.5.3	Usability-Test	824
9.5.4	Systemtest	826
9.5.5	Feldtest	827
9.5.6	Abnahmetest	829
9.6	Workshop: Unit-Tests mit Visual Studio	830
9.6.1	Anlegen eines Testprojekts	830
9.6.2	Hinzufügen der Unit-Tests	832
10	Softwarepflege	835
10.1	Release Management	836
10.1.1	Begriffe	836
10.1.2	Der Release-Prozess	837
10.2	Anforderungen	844
10.2.1	Einführung	845
10.2.2	Die verschiedenen Sichtweisen	847
10.2.3	Anforderungen an eine Anforderung	849
10.3	Zeitschätzung	853
10.3.1	Was ist eine Zeitschätzung?	853
10.3.2	Herausforderungen einer Zeitschätzung	854
10.3.3	Die lernende Organisation	861
10.3.4	Woher kommen Zeitüberschreitungen?	864
10.3.5	Methoden der Zeitschätzung	867
10.4	Zum Schluss	878
	Index	879