Fabian Moritz, René Hézser

Praxisbuch SharePoint-Entwicklung

Galileo Press

Liebe Leserin, lieber Leser,

kaum ein anderes Produkt von Microsoft schreibt derzeit eine solche Erfolgsgeschichte in Unternehmen wie SharePoint. Dies verwundert kaum angesichts der Funktionsvielfalt, die es Unternehmen z. B. in den Bereichen Zusammenarbeit oder Dokumentenmanagement bietet.

Gerade für Sie als Entwickler dürfte jedoch besonders ein Aspekt von SharePoint interessant sein: SharePoint ist mehr als eine reine Softwarelösung für einzelne Anwendungsgebiete wie z. B. das Dokumentenmanagement. Es ist eine *Entwicklungsplattform,* mit der Sie schnell eigene, auf Ihr Unternehmen maßgeschneiderte Lösungen bereitstellen können.

Und damit sind wir beim Thema dieses Buches: Es erläutert Ihnen jederzeit leicht nachvollziehbar, wie Sie Anwendungen in SharePoint entwickeln, die sich nahtlos in die Umgebung in Ihrem Unternehmen integrieren. Das Buch vermittelt Ihnen dabei wesentlich mehr als nur die benötigten Grundlagen. Es versetzt Sie in Schritt für Schritt in die Lage, mit SharePoint u. a. vollständige Enterprise-Lösungen zu realisieren.

Besonderen Wert haben unsere Autoren Fabian Moritz und René Hézser in Ihrem Buch darauf gelegt, alle Kapitel konsequent am täglichen Einsatz von SharePoint-Entwicklern auszurichten. Sicher werden Ihnen die zahlreichen Beispiele und Praxistipps, die die Autoren aus ihrer langjährigen Erfahrung mit SharePoint für Sie zusammengestellt haben, wertvolle Dienste leisten. Die ausgewogene Mischung von Erklärungen, Beispielen und Tipps in Verbindung mit einer leicht verständlichen Ausdrucksweise machen das Buch zu einer rundum gelungenen Anleitung von Praktikern für Praktiker.

Das Buch wurde mit großer Sorgfalt geschrieben, lektoriert und produziert. Sollten sich dennoch Fehler eingeschlichen haben, wenden Sie sich bitte an mich. Ihre Anmerkungen und Ihre Kritik sind immer willkommen.

Sebastian Kestel
Lektorat Galileo Computing

sebastian.kestel@galileo-press.de
www.galileocomputing.de
Galileo Press · Rheinwerkallee 4 · 53227 Bonn

Auf einen Blick

1	**SharePoint als Entwicklungsplattform**	13
2	**SharePoint-Entwicklungswerkzeuge**	69
3	**Das erste SharePoint-Projekt**	119
4	**SharePoint-Objektmodelle**	157
5	**Benutzeroberflächen umsetzen**	269
6	**Programmierkonzepte**	379
7	**Deployment-Konzepte**	429
8	**Workflows**	461
9	**Business Connectivity Services**	497

Inhalt

Danksagung 11

1 SharePoint als Entwicklungsplattform 13

1.1 Die SharePoint-Features im Überblick 15
1.2 Welche Produkte gibt es? 17
1.3 Welche Produktversion ist die für Sie richtige? 20
1.4 Was ist neu an SharePoint 2010? 21
1.4.1 Language Integrated Query (LINQ) 21
1.4.2 Verbesserungen an der Datenplattform 22
1.4.3 Business Connectivity Services (BCS) 22
1.4.4 Client-Objektmodell 22
1.4.5 Silverlight-Integration 23
1.4.6 Sandboxed Solutions 23
1.4.7 Entwicklerdashboard 24
1.4.8 Verbesserungen in der Benutzeroberfläche 24
1.4.9 Visual Studio 2010 SharePoint Tools 24
1.5 Einrichtung der Entwicklungsumgebung 25
1.5.1 SharePoint auf einem Windows Server 2008 (R2) installieren 27
1.5.2 SharePoint auf Windows 7 oder Windows Vista SP 1 installieren 31
1.5.3 Visual Studio 2010 installieren 35
1.5.4 SharePoint Designer 2010 installieren 36
1.5.5 Umgebungsvariablen registrieren 37
1.6 Die SharePoint-Serverarchitektur 37
1.7 SharePoint als Entwicklungsplattform 39
1.7.1 Herausforderungen für den SharePoint-Entwickler 39
1.7.2 Den richtigen Einstieg finden 41
1.7.3 SharePoint und ASP.NET 3.5 43
1.7.4 Die SharePoint-Programmierschnittstellen 44
1.7.5 Das SharePoint-Systemverzeichnis 47
1.7.6 SharePoint und die Internet Information Services 49
1.8 Fehlersuche und Überwachung 57
1.8.1 SharePoint United Logging Service (ULS) 58
1.8.2 Internet Information Services Logs 60
1.8.3 Windows-Ereignisdatenbank 60

1.8.4 Überwachung von Client-Komponenten 61
1.8.5 Das SharePoint-Entwicklerdashboard 65
1.9 Zusammenfassung 67

2 SharePoint-Entwicklungswerkzeuge 69

2.1 Visual Studio 2010 70
2.1.1 Welche Produktversionen gibt es? 71
2.1.2 SharePoint 2010-Projektvorlagen 71
2.1.3 SharePoint-Elementvorlagen 76
2.1.4 Feature und Package Designer 76
2.1.5 Bereitstellungskonfiguration 80
2.1.6 Solution-Paket importieren 82
2.1.7 SharePoint Server-Explorer 83
2.1.8 Erweiterte Integrationsfunktionen 84
2.2 SharePoint Designer 2010 88
2.2.1 Abgrenzung zu Visual Studio 2010 90
2.2.2 SharePoint Designer aus Sicht des Administrators 91
2.2.3 SharePoint Designer aus Sicht des Anwenders (Power Users) 92
2.2.4 SharePoint Designer aus Sicht des Webdesigners 101
2.3 PowerShell 102
2.3.1 Einführung in PowerShell 103
2.3.2 Cmdlets 105
2.3.3 Das Einmaleins des PowerShell-Scripting 105
2.3.4 PowerShell und SharePoint 110
2.4 Zusätzliche SharePoint-Werkzeuge 112
2.4.1 SharePoint-Standardwerkzeuge 112
2.4.2 Community Tools 113
2.5 Zusammenfassung 117

3 Das erste SharePoint-Projekt 119

3.1 Vorbereitung der Umgebung 119
3.1.1 Leere Websitesammlung erstellen 120
3.1.2 Listen erzeugen 120
3.2 Leeres SharePoint-Projekt erzeugen 121
3.3 Ressourcen hinzufügen 126
3.4 Feature erstellen 127
3.5 Feature Receiver erstellen 129
3.6 Projekt bereitstellen 132

3.7 Visuellen Webpart programmieren 139
3.8 Einen visuellen Webpart debuggen 151
3.9 Fazit 156

4 SharePoint-Objektmodelle 157

4.1 Die Auswahl des richtigen Objektmodells? 158
4.2 Server-Objektmodell 159
4.2.1 Assemblys und Namensräume 159
4.2.2 Server-Architektur 160
4.2.3 Site-Architektur 165
4.2.4 Websitesammlungen und Webseiten 166
4.2.5 Listen und Dokumentbibliotheken 178
4.2.6 Benutzer, Gruppen und Berechtigungen 202
4.2.7 Heraufstufung und Impersonifizierung 210
4.3 Client-Objektmodell 214
4.3.1 Struktur des Client-Objektmodells 215
4.3.2 Die Funktionsweise des Client-Objektmodells 216
4.3.3 Eine Client-Anwendung – Schritt für Schritt 220
4.3.4 Die Struktur des Client-Objektmodells 222
4.3.5 Daten abfragen 224
4.3.6 Daten manipulieren 230
4.3.7 Webseiten und Listen erstellen 234
4.3.8 Arbeit mit Benutzern und Gruppen 241
4.3.9 Arbeit mit Dokumenten 244
4.3.10 Umsetzung von ECMAScript-Client-Anwendungen 247
4.4 Webservices 252
4.4.1 Standard-SharePoint-Webservices 252
4.4.2 Standard-Webservices verwenden 254
4.4.3 Benutzerdefinierte WCF-Services bereitstellen 261
4.5 Fazit 268

5 Benutzeroberflächen umsetzen 269

5.1 Verbesserungen in der SharePoint-Benutzeroberfläche 270
5.2 Masterseiten 272
5.2.1 Einführung in die Technologie 272
5.2.2 Masterseitentypen 274
5.2.3 Standard-Masterseiten 275
5.2.4 Masterseiten-Token 277

5.2.5 Unterstützung von dynamischen Token für Anwendungsseiten ... 278
5.2.6 Feste Anwendungsseiten ... 279
5.2.7 Geschützte Anwendungsseiten ... 280
5.2.8 Allgemeiner Aufbau einer SharePoint-Masterseite ... 280
5.2.9 Benutzerdefinierte Masterseite bereitstellen ... 283
5.3 Ribbon ... 291
5.3.1 Standardelemente des Ribbons ... 291
5.3.2 Standard-Ribbon erweitern ... 292
5.3.3 Vollständige Tabs und Gruppen erzeugen ... 296
5.3.4 Standardschaltflächen ersetzen und entfernen ... 300
5.3.5 Ribbon um Code erweitern ... 302
5.3.6 Debuggen von Ribbon-Erweiterungen ... 305
5.3.7 Kontextuelle Webparts entwickeln ... 306
5.4 Benutzerdefinierte Aktionen ... 318
5.4.1 Element-Kontextmenüs erweitern ... 321
5.4.2 Websiteeinstellungen um neue Gruppen erweitern ... 323
5.4.3 Standardfunktionen entfernen ... 325
5.4.4 Benutzerdefinierte Web-Controls ... 325
5.5 Statusliste, Benachrichtigungen und Dialoge ... 327
5.5.1 Statusleiste anpassen ... 329
5.5.2 Nachrichtenleiste anpassen ... 332
5.5.3 SharePoint-Dialoge verwenden ... 333
5.6 Webparts ... 336
5.6.1 ASP.NET- oder SharePoint-Webparts? ... 337
5.6.2 Webpart-Definition ... 338
5.6.3 Webpart-Zonen ... 339
5.6.4 Personalisierung ... 339
5.6.5 Der »SafeControl«-Eintrag ... 341
5.6.6 Der Lebenszyklus von Webparts ... 342
5.6.7 Webparts und Code Access Security ... 345
5.6.8 Einen vollständigen Webpart programmieren ... 349
5.7 Silverlight ... 355
5.7.1 Architektur ... 357
5.7.2 Aufbau einer XAP-Datei ... 358
5.7.3 Asynchrone Datenabfrage ... 360
5.7.4 Unterschiede zu WPF ... 363
5.7.5 XAML ... 364
5.7.6 Cross-Domain-Datenzugriff ... 367
5.7.7 SharePoint-Silverlight-Webpart ... 368

5.7.8 Entwicklungsumgebung ... 370
5.7.9 Eine vollständige Silverlight-Anwendung programmieren ... 374
5.8 Fazit ... 377

6 Programmierkonzepte ... 379

6.1 CAML ... 379
6.1.1 Sitedefinitionen ... 379
6.1.2 Listendefinition ... 383
6.1.3 Programmierung mit CAML ... 392
6.2 LINQ to SharePoint ... 395
6.2.1 Einführung in LINQ ... 395
6.2.2 LINQ Provider ... 398
6.2.3 »LINQ to SharePoint«-Provider ... 400
6.2.4 SPMetal ... 401
6.2.5 Anwendungsbeispiele für »LINQ to SharePoint« ... 404
6.2.6 Konfliktmanagement ... 409
6.2.7 Effiziente und ineffiziente Abfragen ... 413
6.2.8 Auswertung von LINQ-Abfragen ... 415
6.2.9 LINQ im Client-Objektmodell ... 416
6.3 REST ... 419
6.3.1 Die Architektur von REST ... 419
6.3.2 Operationen ... 420
6.3.3 Syntax ... 421
6.3.4 Optionen ... 422
6.3.5 REST in SharePoint 2010 ... 423
6.3.6 REST-Clients mit Visual Studio implementieren ... 425
6.4 Fazit ... 427

7 Deployment-Konzepte ... 429

7.1 Features ... 429
7.1.1 Features installieren und aktivieren ... 431
7.1.2 »Feature.xml« ... 433
7.1.3 Element-Manifest ... 436
7.1.4 Feature Receiver ... 438
7.1.5 Feature Upgrades ... 439
7.2 SharePoint-Solutions ... 443
7.2.1 Solution-Manifest ... 445
7.2.2 Installation und Bereitstellung ... 447

7.3 Sandboxed Solutions 449
7.3.1 Architektur 451
7.3.2 Webparts in Sandboxed Solutions 453
7.3.3 Installation einer Sandboxed Solution 454
7.3.4 Validierung 456
7.3.5 Monitoring und Ressourcenüberschreitung 458
7.4 Fazit 460

8 Workflows 461

8.1 Architektur 462
8.1.1 Aktivitäten 464
8.1.2 Workflowaufgaben 464
8.1.3 Correlation Token 465
8.2 Workflow-Typen 466
8.3 Workflowformulare 468
8.4 Vergleich der Entwicklungswerkzeuge 469
8.5 Workflows in SharePoint 470
8.6 SharePoint Designer 2010 473
8.7 Workflows mit Visual Studio 2010 479

9 Business Connectivity Services 497

9.1 Architektur 498
9.1.1 Server-Komponenten 499
9.1.2 Externe Inhaltstypen 500
9.1.3 Authentifizierung gegenüber externen Daten 501
9.1.4 Berechtigungen auf Modelle und Entitäten 502
9.1.5 Metadaten-Modell 503
9.1.6 Primärschlüssel 512
9.2 Erstellung von benutzerdefinierten Modellen 514
9.2.1 SharePoint Designer 514
9.2.2 Visual Studio 2010 520

Index 539

Danksagung

Fabian Moritz

Mir ist es ein besonderes Bedürfnis einigen Personen zu danken. An erster Stelle meiner Frau Kelly, die über mehrere Monate hinweg auf mich verzichten musste und mir während dieser Zeit eine große Hilfe war. An zweiter Stelle möchte ich meinem Co-Autor René Hészer dafür danken, dass er mich bei diesem Buchprojekt tatkräftig unterstützt hat. Ohne seine Expertise wäre dieses Buch so nicht möglich gewesen. Nicht vergessen möchte ich mein Team, das mich durch hervorragende Projektarbeit beruhigt hat schlafen lassen.

René Hézser

Ohne Unterstützung wäre es mir nicht möglich gewesen, dieses Buch zu schreiben. Deshalb möchte ich mich insbesondere bei meinem Co-Autor Fabian Moritz bedanken, dass er es mir durch sein Vertrauen überhaupt erst ermöglicht hat, an diesem Buch mitzuwirken. Mein Dank gilt ebenfalls meiner Frau Corinna, die viele Abende, Tage und Wochenenden erdulden musste, dass ich trotz Anwesenheit mit Abwesenheit glänzte. Auch mein Sohn musste hinnehmen, dass sein Vater nicht mehr Zeit mit ihm verbringen konnte.

SharePoint ist kein Produkt, sondern eine Entwicklungsplattform, mit der sich individuelle benutzerdefinierte Lösungen unterschiedlicher Art realisieren lassen.

1 SharePoint als Entwicklungsplattform

»Was ist eigentlich SharePoint?« Diese Frage ist eine der häufigsten, die man als SharePoint-Spezialist gestellt bekommt. Die Frage ist durchaus berechtigt, weil es hierzu sehr viele unterschiedliche Aussagen gibt. Sie reichen von »SharePoint ist eine Zusammenarbeitsplattform« über »SharePoint ist ein Dokumentenmanagement-System« bis hin zu »SharePoint ist ein Teufelswerk« – die Palette der Aussagen ist sehr weit gefächert. Michael Greth, ein Microsoft Most Valuable Professional (MVP) für SharePoint Server, hat die Technologie SharePoint sehr zutreffend so beschrieben:

> *»Man muss sich SharePoint als eine Art Info–Lego vorstellen. SharePoint stellt einen riesengroßen Baukasten bereit, mit dem sich unterschiedliche Lösungen realisieren lassen. Von einer einfachen Teamsite zur Unterstützung der Zusammenarbeit über spezielle Projekträume bis hin zu kompletten Intranet-Portalen – die Einsatzgebiete von SharePoint sind sehr vielseitig.«*

Diese Fähigkeit kann gleichzeitig als Segen und Fluch bewertet werden.

Als »Fluch« deshalb, weil SharePoint damit als nichts Halbes und nichts Ganzes bezeichnet werden kann. Stellt man SharePoint anderen, spezialisierten Produkten für bestimmte Anforderungen (zum Beispiel für das Dokumentenmanagement) gegenüber, wird man feststellen, dass SharePoint im Vergleich zu diesen Lösungen teilweise nur 80 Prozent des Funktionsumfangs abdeckt. Aber genau darin steckt das Potenzial dieser Technologie – und das ist der »Segen«: SharePoint ist kein Produkt, sondern eine Entwicklungsplattform!

Recherchiert man in den gängigsten Nachschlagewerken, findet man als Definition des Begriffs ungefähr folgende Beschreibung: Eine Entwicklungsplattform zeichnet sich dadurch aus, dass sie eine Anwendungsschicht oberhalb des Betriebssystems bereitstellt. Diese Schicht besteht aus einem oder mehreren Frameworks, gut dokumentierten Programmierschnittstellen, Standardkomponenten sowie unterschiedlichen Werkzeugen, die die Realisierung individueller Unternehmensanforderungen ermöglichen.

Diese Eigenschaften lassen sich sehr gut auf die SharePoint-Technologie übertragen. SharePoint wird mit unterschiedlichen Produkten ausgeliefert, stellt zahlreiche Programmierschnittstellen, Anwendungskomponenten und Dienste bereit und liefert einen reichhaltigen Satz an Werkzeugen, die die Umsetzung von benutzerdefinierten Anwendungen unterstützen.

Die datenbankgestützte Plattform basiert vollständig auf ASP.NET 3.5, integriert sich nahtlos in die homogene Windows-Landschaft und nutzt an geeigneter Stelle die Standardkomponenten des Betriebssystems bzw. von ASP.NET.

In diesem Buch werden Ihnen die grundlegenden Aspekte von SharePoint 2010 als Entwicklungsplattform vorgestellt. Sie lernen die Charakteristika der SharePoint-Plattform, ihre Schnittstellen und Standardkomponenten, die Basistechnologien sowie die verschiedenen Werkzeuge zur Umsetzung von benutzerdefinierten Anwendungen kennen. Zudem finden Sie in den unterschiedlichen Kapiteln dieses Buches an geeigneter Stelle Tipps und Tricks, weiterführende Themen sowie mögliche Entwicklerressourcen.

In den vergangenen Jahren konnte man einen deutlichen Trend bei SharePoint-Projekten beobachten: Sie dauern immer länger, die Anzahl der beteiligten Akteure nimmt zu, und die Komplexität wird immer größer. Aus diesem Grund ist es besonders wichtig, den Entwicklungsprozess auf die jeweilige Anforderung abzustimmen und die Umsetzung durch geeignete Werkzeuge zu unterstützen. Bei dieser Betrachtung sollte nicht nur die Rolle des Programmierers, sondern auch die des Webdesigners, des Testers oder des Projektmanagers berücksichtigt werden.

Ein hervorragendes Werkzeug zur Begleitung sämtlicher Phasen eines SharePoint-Projekts ist *Visual Studio 2010* in Verbindung mit dem *Team Foundation Server 2010*. Die Suite beinhaltet neben einer sehr guten Integration in den SharePoint-Entwicklungsprozess zahlreiche nützliche Werkzeuge zur technischen Unterstützung sämtlicher Projektphasen, von der Planung über die Implementierung bis hin zur Stabilisierung. Insbesondere bei einem Zusammenschluss mehrerer Projektbeteiligter erreichen Sie durch den Einsatz eines zentralen Planungs- sowie Code-Verwaltungswerkzeugs einen optimalen Output sämtlicher Akteure.

In diesem Buch werden Sie erfahren, welche Entwicklungswerkzeuge Ihnen für die SharePoint-Programmierung zur Verfügung stehen und wie Sie diese sinnvoll einsetzen können.

Eine große Bedeutung für die SharePoint 2010-Familie hat *Office 2010*. Die Suite stellt nicht nur wichtige Werkzeuge für die integrierte Bearbeitung von Share-

Point-Dokumenten bereit, sie spielt auch eine grundlegende Rolle in der Umsetzung von SharePoint-Lösungen. Mit dem *SharePoint Designer 2010* sowie *InfoPath 2010* stellt Office 2010 zwei sehr elementare Bausteine in der Liste der SharePoint-Werkzeuge bereit. Mit ersterem lassen sich nicht nur administrative Aufgaben ausüben, das Werkzeug dient vor allem dem Webdesigner und technisch versierten Nutzer zur Anpassung der Benutzeroberfläche, zur Erstellung von Workflows oder zur Bereitstellung von Seitenvorlagen für Web-Content-Management-Systeme.

Die Bedeutung von InfoPath 2010 für den SharePoint-Entwickler hat im Vergleich zu den Vorversionen noch zugenommen. Neben der Bereitstellung von client- oder webbasierten Formularen dient die neue Version auch als Standardwerkzeug zur Anpassung von Listen- oder Workflow-Formularen. Beide Office-Produkte wurden gleichermaßen vollständig überarbeitet und integrieren die neue Ribbon-Benutzeroberfläche aus Office.

Sie werden in diesem Buch erfahren, dass Visual Studio 2010 nicht das einzige nützliche Tool für den SharePoint-Programmierer ist. Je nach Anwendungsfall können auch der SharePoint Designer 2010, InfoPath 2010 oder auch andere Community- bzw. Drittanbieter-Tools bei der Realisierung von SharePoint 2010-Lösungen behilflich sein.

1.1 Die SharePoint-Features im Überblick

SharePoint 2010 beschreibt sich selbst als Entwicklungsplattform für den Einsatz im Unternehmen und im Web. Die Technologie liefert dabei einen hohen Grad an Erweiterungs- und Anpassungsmöglichkeiten. Die Anzahl der unterschiedlichen Möglichkeiten für den Programmierer ist enorm. Von der einfachen Anpassung der Benutzeroberfläche bis hin zur Realisierung kompletter Intranet- oder Internetlösungen – die Technologie SharePoint bietet eine reichhaltige Plattform für eine Vielzahl von Anwendungsfällen.

SharePoint 2010 liefert nicht nur die für die Lösungsentwicklung erforderlichen Frameworks, Schnittstellen und Dienste, sondern auch einen reichhaltigen Satz an Standardfunktionen. Das Spektrum reicht von einfachen Websites für die Zusammenarbeit über Dokumentenmanagement-Features und die Suche bis hin zu Business-Intelligence- oder Web-Content-Management-Funktionen. Seinen Funktionsumfang gruppiert SharePoint 2010 in sechs Bereiche: *Sites*, *Communities*, *Content*, *Search*, *Insights* und *Composites*.

Abbildung 1.1 Die SharePoint 2010-Technologien im Überblick

Die Begrifflichkeiten werden bewusst aus dem Englischen übernommen und nicht übersetzt, weil für die deutschen Bezeichnungen größtenteils (noch) keine passenden Begriffe existieren. Tabelle 1.1 liefert Ihnen eine detaillierte Beschreibung des Funktionsumfangs von SharePoint 2010.

Fähigkeit	Beschreibung
Sites	Sites bilden die Grundlage einer SharePoint-Plattform. Sie unterstützen die Nutzer bei der Zusammenarbeit in unterschiedlichen Teams innerhalb und außerhalb des Unternehmens. Dazu integriert SharePoint 2010 zahlreiche Standardvorlagen, RSS-, Benachrichtigungs- und Workflow-Funktionen sowie eine hervorragende Integration in Office 2010.
Communities	Die Communities-Funktionen liefern die Möglichkeit, Personen innerhalb und außerhalb des Unternehmens zu verbinden. Wichtige Features sind die Abbildung und Suche von persönlichen Fähigkeiten und Beziehungen. Außerdem beinhaltet diese Fähigkeit Funktionen wie Tagging, Tag Clouds oder die Bewertungsmöglichkeit des Seiteninhalts.
Content	Mit der Content-Management-Funktion lassen sich Inhalte in Webseiten oder Dokumentencentern über unterschiedliche Wege bereitstellen, editieren oder intuitiv um digitale Medien erweitern.
Search	Über die Suche können Daten aus SharePoint-Webseiten oder anderen Datenquellen (Dateisystem, Exchange Server, Webseiten oder LOB-Systemen) indiziert und durchsucht werden.

Tabelle 1.1 Funktionsblöcke von SharePoint 2010

Fähigkeit	Beschreibung
Insights	Daten aus Backend- oder SharePoint-Systemen können über Webseiten, Dashboards, Excel Services oder Performance Point Services ausgewertet, transformiert und visualisiert werden. SharePoint 2010 bietet die Grundlage dafür an, Informationen dort bereitzustellen, wo die Entscheidungen getroffen werden.
Composites	SharePoint 2010 liefert zahlreiche Möglichkeiten, integrierte Lösungen mit Out-of-the-Box-Funktionen und Standardwerkzeugen zu realisieren.

Tabelle 1.1 Funktionsblöcke von SharePoint 2010 (Forts.)

Die hier beschriebenen Funktionen von SharePoint 2010 werden in unterschiedlichen Produkten ausgeliefert. Den Kern der SharePoint-Technologie und ein wesentlicher Standard-Funktionsumfang wird durch *SharePoint Foundation 2010* bereitgestellt. Alle anderen Produktversionen (zum Beispiel *SharePoint Server 2010, Search Server 2010* oder *Project Server 2010*) bauen auf diesem Standard auf und erweitern ihn um benutzerdefinierte Features, Dienste oder Anwendungskomponenten.

Im Fokus dieses Buches stehen die Kernfunktionen, Standard-Technologien, Schnittstellen sowie Sprachkomponenten von SharePoint 2010.

Das Wissen, das Sie durch dieses Buch erlangen, können Sie in der Anwendungsentwicklung für sämtliche SharePoint-Produktversionen einsetzen.

1.2 Welche Produkte gibt es?

SharePoint 2010 wird in zwei wesentlichen Produktversionen ausgeliefert: *Microsoft SharePoint Foundation 2010* und *Microsoft SharePoint Server 2010.*

Microsoft SharePoint Foundation 2010 ist der Nachfolger der *Windows SharePoint Services 3.0*. Diese Produktversion ist kostenlos. Sie kann von jedem Anwender benutzt werden, der über eine gültige Windows Client-Zugriffslizenz verfügt. SharePoint Foundation 2010 stellt die Basisfunktionen der SharePoint-Technologie und die Standardelemente für die Bereitstellung von Webseiten (Sites), Community Features (Blogs, Wikis oder Diskussionen) sowie zahlreiche Tools und Werkzeuge (Composites) bereit.

Microsoft SharePoint Server 2010 ist der lizenzpflichtige »große Bruder« von SharePoint Foundation 2010. Er basiert auf der Basisfunktion von SharePoint Foundation und erweitert sie um zusätzliche Enterprise Features. Microsoft SharePoint Server 2010 wird in zwei unterschiedlichen Lizenzierungsvarianten ausgeliefert: In der *Standard*- und der *Enterprise*-Version.

Standard vs. Enterprise

SharePoint Server 2010 Standard und Enterprise sind im Grunde genommen keine zwei unterschiedlichen Produkte. Die technische Basis dieser beiden Versionen ist identisch. Der Unterschied besteht lediglich in der Art der Client-Lizenzierung. Die Nutzung von Enterprise-Funktionen erfordert den Erwerb einer Enterprise Client-Zugriffslizenz (CAL). Sie sollten beachten, dass diese Lizenz zusätzlich zur Standard-CAL erworben werden muss.

Die Enterprise-Funktionen werden entweder durch die Websitevorlage oder die jeweiligen SharePoint-Features aktiviert.

Tabelle 1.2 listet die wichtigsten Erweiterungen der Standardversion, gruppiert nach den Technologiebausteinen, auf.

Bereich	Funktion
Sites	erweiterte Kontrolle
	Seitenstatistik
Communities	soziale Netzwerke
	Enterprise-Wiki
	Benutzerprofile und Organisationen
	Meine Seite (Mein Inhalt, Meine News und Mein Profil)
	Ratings
	Tagging
Content	Inhaltsorganisation
	Dokumentensätze
	Dokumenten-ID
	Verwaltetet Metadaten.
	verteilte Inhaltstypen
	Rich-Media-Support
Suche	beste Suchergebnisse
	Standardsortierung
	Suchfacetten (Metadaten-Verfeinerung)
	Personensuche
	phonetische Suche
	Abfragevorschläge (Meinten Sie…)
	Suchbereiche
	Windows 7-Suche

Tabelle 1.2 SharePoint Server 2010 Standard-Features

Bereich	Funktion
Composites	BCS-Profileseite
	Workflow-Vorlage

Tabelle 1.2 SharePoint Server 2010 Standard-Features (Forts.)

Die Erweiterungen der Enterprise-Version beziehen sich auf die Technologiebausteine der Suche, Insights und Composites (siehe Tabelle 1.3).

Bereich	Funktion
Search	erweiterte Sortierungsmöglichkeiten
	kontextuelle Suche
	bessere Verfeinerung der Suchfacetten
	ähnliche Suchergebnisse
	Vorschaufunktionen
	visuelle beste Suchergebnisse
Insights	Business Intelligence Center
	Key-Performance-Indikatoren (KPIs)
	Datenverbindungsbibliothek
	Excel-Services
	PowerPivot
	PerformancePoint-Services
	Visio-Services
Composites	Access-Services
	Businessdaten-Integration in Office
	Geschäftsdaten-Webparts
	InfoPath-Forms-Services

Tabelle 1.3 SharePoint Server 2010 Enterprise-Features

Neben diesen drei Kernprodukten wird die SharePoint-Technologie noch in weiteren Versionen ausgeliefert, die jedoch nur eine Lizenzierungsvariante darstellen. Für den Einsatz von SharePoint 2010 in Extranet- oder Internet-Szenarien stellt Microsoft die Lizenzvariante *SharePoint Server 2010 for Internet Sites (Standard/Enterprise)* bereit. Diese Produktversion enthält jeweils den gleichen Funktionsumfang wie die Intranet-Lizenzvarianten. Der Unterschied besteht in der Art der Lizenzierung. SharePoint Server 2010 for Internet Sites wird serverseitig lizenziert, Client-Zugriffslizenzen sind nicht erforderlich.

1.3 Welche Produktversion ist die für Sie richtige?

Aufgrund des großen Spektrums der Standardfunktionen dieser drei Produktvarianten ist es ratsam, sich frühzeitig Gedanken über die Lizenzvariante zu machen, die Sie brauchen. Teilweise ist es sinnvoll, für die Umsetzung einer spezifischen Anforderung eine Standardtechnologie von SharePoint zu verwenden. In diese Überlegung sollten auch die weiteren Produkte der SharePoint-Familie mit einbezogen werden, z. B. *InfoPath Forms Services*, auch wenn deren Einsatz zusätzliche Lizenzkosten mit sich bringen. Der Gesamtbetrag kann trotz der zusätzlichen Aufwendungen durch weniger Programmierarbeit ausgeglichen werden. Eine richtige Entscheidung kann aber auch sein, die Anforderung auf Grundlage von SharePoint Foundation 2010 umzusetzen.

Eine mögliche Herangehensweise zur Auswahl der richtigen Technologie ist die Betrachtung der *Total Cost of Ownership* (TOC) der jeweiligen Lösung und der eventuellen Folgeprojekte. In bestimmten Konstellationen kann die Umsetzung eines einzigen Projekts mit dem Enterprise Feature von SharePoint Server nicht rentabel sein, sich jedoch bereits mit dem zweiten Projekt amortisieren. Bei der Planung eines SharePoint-Projekts sollten Sie im Vorfeld folgende Fragen klären:

- Kann die Anforderung mit Standardfunktionen von SharePoint abgedeckt werden?
- Welche Lizenzvariante ist hierfür erforderlich?
- Wie hoch sind die Lizenzkosten für die geplante Anzahl an Nutzern?
- Wie hoch sind die Entwicklungskosten?
- Bei welcher Kostenschwelle amortisiert sich der Erwerb der Enterprise-Version?
- Wie sieht die Roadmap für die kommenden drei Jahre aus? Welche weiteren SharePoint-Projekte sind geplant?
- Welche Vorgaben kommen aus Infrastruktursicht?
- Welches Know-how haben die SharePoint-Programmierer, und welche Kosten entstehen für Schulungen und Trainings?

Dieser kleine Ausflug in die Planung eines SharePoint-Projekts soll deutlich machen, wie groß die Tragweite der Entscheidung über die passende SharePoint-Produktversion sein kann.

Die Auswirkung dieser Entscheidung hat einen direkten Einfluss auf die mögliche Basistechnologie, das Team und den Aufwand für den Programmierer. Es gibt gute Gründe dafür, eine komplette Lösung auf Basis von SharePoint Foundation 2010 zu implementieren, und zwar dort, wo die Programmierung einen sehr großen Anteil an den Projektvorgängen einnimmt. Adäquat dazu besteht die Mög-

lichkeit, eine Lösung auf Basis einer Standardtechnologie von SharePoint Server 2010 zu implementieren, auch wenn in diesem Fall die Lizenzkosten für das Produkt teurer wären.

Diese Betrachtung macht einen weiteren wichtigen Faktor für die Auswahl der passenden Produktversion deutlich: die Anzahl der Entwickler und das vorhandene Know-how. Bitte berücksichtigen Sie, dass Sie für ein SharePoint-Projekt das nötige Entwickler-Know-how benötigen. Fehler in der Umsetzung eines SharePoint-Projekts können sich direkt oder auch zu einem späteren Zeitpunkt noch auf die SharePoint-Plattform auswirken.

Dieses Buch soll dazu beitragen, Ihre fachlichen Fähigkeiten zur Umsetzung eines SharePoint Projekts auszubauen und zu verbessern. Sie werden in den folgenden Kapiteln erfahren, welche vorbereitenden Schritte für die Entwicklung von SharePoint-Anwendungen erforderlich sind, welche Entwicklungswerkzeuge Sie hierzu einsetzen können und woher Sie sich gegebenenfalls weiterführende Informationen und Ressourcen beschaffen können.

1.4 Was ist neu an SharePoint 2010?

SharePoint ist eine ausgereifte Entwicklungsplattform, die neben zahlreichen Programmierschnittstellen für den Server und den Client die neusten Programmiertechnologien – wie zum Beispiel LINQ, REST oder Silverlight – unterstützt. Auch die Möglichkeit, direkt auf einem Client-Betriebssystem entwickeln zu können, macht den Weg bis hin zur SharePoint-Entwicklung deutlich kürzer. Wir möchten jetzt einen genaueren Blick auf die wichtigsten Verbesserungen von SharePoint 2010 werfen.

1.4.1 Language Integrated Query (LINQ)

Mit dem .NET Framework 3.5 hat Microsoft eine neue Sprachtechnologie eingeführt, mit der Sie in einem SQL-ähnlichen Vokabular Daten aus unterschiedlichen Datenquellen abfragen können. LINQ ist eine Technologie, mit der sich dynamische Datenabfragen auf eine sehr flexible Art und Weise mittels Programmcode realisieren lassen. Technisch betrachtet ist LINQ eine Art objektrelationaler Mapper (OR-Mapper), der bestimmte Operationen einer nativen Datenabfrage in einer .NET-Programmiersprache zugänglich macht.

Ein wichtiger Bestandteil von LINQ ist der sogenannte LINQ-Provider. Er stellt die Schnittstelle zur eigentlichen Datenquelle her und transformiert den LINQ-Befehl in eine anwendungsspezifische Datenabfrage. Standard-Provider des .NET Frameworks sind beispielsweise *LINQ to Objects, LINQ to SQL* oder *LINQ to*

XML. Mit SharePoint 2010 wurde auch ein LINQ-Provider eingeführt, der die flexible Datenabfrage gegen SharePoint-typische Elemente ermöglicht, indem er eine LINQ-Abfrage in CAML (*Collaboration Application Markup Language*) transformiert.

1.4.2 Verbesserungen an der Datenplattform

Eine wesentliche Funktion einer SharePoint-Plattform ist die Verwaltung von Dokumenten und anderen Informationen. Prinzipiell basieren sämtliche Datencontainer einer SharePoint-Anwendung auf einer sogenannten Liste. Mit SharePoint 2010 wurde das Listenmodell um einige wichtige Funktionen erweitert. So lassen sich zum Beispiel Datenrelationen abbilden, Bedingungen für das Löschverhalten definieren oder Plausibilitätsprüfungen realisieren. Entwickler können sich auch darüber freuen, dass Listenansichten anstelle von CAML nun vollständig über XSLT erzeugt werden können.

In Kombination mit der neuen Ribbon-Benutzeroberfläche und den Werkzeugen des SharePoint Designers 2010 oder des Visual Studio 2010 lassen sich mit SharePoint 2010 alle Arten von datenorientieren Anwendungen realisieren.

1.4.3 Business Connectivity Services (BCS)

Die *Business Connectivity Services* (BCS) erhalten mit SharePoint 2010 nicht nur einen neuen Namen, sondern auch einen reichhaltigen Satz neuer Funktionen. Mit dem Nachfolger des *Business Data Catalogs* (BDC) lassen sich Daten aus *Line Of Business*-(LOB-)Systemen direkt in eine SharePoint-Umgebung integrieren und darstellen. Externe Systeme, mit denen sich SharePoint verbinden kann, sind beispielsweise SQL Server-Datenbanken, Webservices, SAP-Anwendungen sowie andere Backend-Systeme. Die BCS-Programmierschnittstelle wurde so erweitert, dass sich externe Daten sowohl serverseitig als auch auf dem Client integrieren lassen. Die Präsentation der Daten ist mit der von einfachen Listen und Listenelementen vergleichbar. Neu in SharePoint 2010 ist die Möglichkeit, über diese Schnittstelle externe Daten zu editieren. Business-Daten können zusätzlich in Office-Werkzeugen wie Outlook oder Word integriert oder mit dem *SharePoint Workspace 2010* offline bearbeitet werden. Die Modellierung von BCS-Schnittstellen wird durch den SharePoint Designer 2010 oder Visual Studio 2010 unterstützt.

1.4.4 Client-Objektmodell

SharePoint Foundation 2010 integriert neben verbesserten serverseitigen Programmierschnittstellen auch ein neues Client-Objektmodell, mit dem sich unterschiedliche Client-Lösungen realisieren lassen, wie beispielweise Windows-, Sil-

verlight- oder JavaScript-basierte Anwendungen. Das Client-Objektmodell macht die Umsetzung von speziellen Webservices für den clientseitigen Zugriff überflüssig und somit die Umsetzung von Client-Anwendungen deutlich lukrativer. Es wird in drei Varianten ausgeliefert:

- als ein Managed Objektmodell,
- als eine Silverlight-API sowie
- als eine ECMAScript-Bibliothek (JavaScript, JScript).

Der Funktionsumfang des Client-Objektmodells ist im Vergleich zur serverseitigen API deutlich abgespeckt, jedoch für grundlegende Aufgaben vollständig ausreichend. Mittels des Client-Objektmodells können Sie beispielsweise Listenelemente auslesen, editieren oder löschen, neue Listen oder Webseiten erstellen, Dokumente hochladen oder Berechtigungen einer Webseite anpassen.

1.4.5 Silverlight-Integration

Die Möglichkeit, Silverlight-Anwendungen in SharePoint zu integrieren, ist vom Prinzip her nicht neu. Bereits in der Vorversion von SharePoint 2010 war es möglich, Silverlight-Anwendungen in eine SharePoint-Webseite zu integrieren. Dazu mussten ein paar manuelle Anpassungen in der *web.config*-Datei der Webanwendung vorgenommen sowie ein benutzerdefinierter Webpart entwickelt werden, der die jeweilige Silverlight-Anwendung ausliefert. Neu in SharePoint 2010 ist die native Unterstützung von Silverlight, ein Standard-Silverlight-Webpart sowie ein eigenständiges Client-Objektmodell, das die Realisierung von Silverlight-basierten Anwendungen deutlich vereinfacht.

1.4.6 Sandboxed Solutions

Sandboxed Solutions sind ein komplett neues Deployment-Modell, das es ermöglicht, SharePoint Solution-Pakete (WSP-Dateien) auf der Ebene einer Websitesammlung zu installieren. Besonders im Hosting-Umfeld bieten Sandkastenlösungen völlig neue Perspektiven, da SharePoint-Anwendungen nicht mehr durch einen zentral gesteuerten Prozess bereitgestellt werden müssen. Sandkastenlösungen sind auf einen bestimmten Funktionsumfang des Objektmodells (bezogen auf die Websitesammlung) beschränkt. Sie unterstützen die Arbeit mit Inhaltstypen, Websitespalten, benutzerdefinierten Aktionen, Workflows, Ereignishandlern oder Webparts.

Technisch betrachtet ist die »Sandbox« ein separater Prozess, in dem die SharePoint Solution ausgeführt wird. Sie stellt einen Teil des `Microsoft.SharePoint`-Namensraums bereit, der den Code auf die Ausführung von Funktionen inner-

halb der Websitesammlung einschränkt. Die Limitierung wird über Code-Access-Security-(CAS-)Richtlinien gesteuert. Die Installation einer Sandboxed Solution wird über das Lösungsverzeichnis auf der Ebene der Websitesammlung ermöglicht. Die Kontrolle, das Management sowie das Monitoring werden über die Zentraladministration gesteuert.

1.4.7 Entwicklerdashboard

Die Umsetzung von SharePoint-Lösungen ist manchmal mit sehr komplexen Zusammenhängen verbunden. Insbesondere bei der Fehlersuche oder der Performance-Optimierung muss man als Entwickler auf eine Vielzahl verschiedener Werkzeuge zurückgreifen, um server- oder clientseitig Kennzahlen auswerten zu können. SharePoint 2010 liefert für diese Aufgaben eine in die SharePoint-Oberfläche integrierte Lösung. Das Entwicklerdashboard ermöglicht die Auswertung unterschiedlicher serverseitiger Aufrufe und Kennzahlen auf eine unkomplizierte Art und Weise.

1.4.8 Verbesserungen in der Benutzeroberfläche

Das »Look & Feel« der SharePoint-Benutzeroberfläche wurde im Vergleich zur Vorversion grundlegend überarbeitet. Die mit Office 2007 eingeführte Ribbon-Technologie hält Einzug in die neue Generation von SharePoint und präsentiert sich durchgängig in sämtlichen Webseiten, Anwendungen und in der Zentraladministration. Der kontextsensitive Ribbon wird im oberen Bereich der Webseite eingeblendet und lässt sich vollständig anpassen.

Unterhalb des Ribbons integriert SharePoint eine neue Statusleiste, über die, je nach Zustand der Anwendung, spezielle Informationen für den Benutzer bereitgestellt werden können.

Eine dritte wichtige Verbesserung in der SharePoint-Benutzeroberfläche ist das Dialog Framework. Es macht Schluss mit langen vollständigen Seitenaufrufen und integriert dafür eine neue, auf JavaScript und partiellen Seitenaufrufen basierende Popup-Komponente, die Seitendialoge als Ebene oberhalb der eigentlichen Webseite darstellt. Das Dialog Framework basiert zu großen Teilen auf dem neuen SharePoint Client-Objektmodell und kann vollständig in eigene Anwendungen integriert werden.

1.4.9 Visual Studio 2010 SharePoint Tools

Abgerundet werden die Verbesserungen durch zahlreiche neue SharePoint-Vorlagen und Tools in Visual Studio 2010. Entwicklern wurde es in der Vergangenheit

nicht besonders leicht gemacht, benutzerdefinierte Anwendungen für SharePoint zu programmieren. Visual Studio 2010 ist die erste Version seiner Produktreihe, die weitreichende Unterstützung für die Umsetzung von benutzerdefinierten SharePoint-Anwendungen Out-of-the-Box liefert. Die Entwicklungsumgebung stellt für die gängigsten SharePoint-Typen Projekt- sowie Elementvorlagen bereit, unterstützt den Prozess der Feature- und Solution-Generierung und macht die Bereitstellung und das Debugging von SharePoint-Anwendungen deutlich flexibler.

1.5 Einrichtung der Entwicklungsumgebung

Bevor Sie mit der SharePoint-Programmierung beginnen, müssen Sie eine passende Entwicklungsumgebung einrichten. Eine Entwicklungsumgebung besteht in der Regel aus mehreren Komponenten: Neben der SharePoint-Installation sind auf dem Entwicklungssystem grundsätzlich auch ein SQL-Server, Visual Studio, der SharePoint Designer oder etwaige Drittanbieter- Tools zu finden.

Die richtige Wahl der Entwicklungsumgebung ist von unterschiedlichen Faktoren abhängig. Die zunächst wichtigste Frage besteht darin, ob Sie allein oder im Team arbeiten wollen. Programmieren Sie eigenständig, steht Ihnen die Option offen, die Entwicklungswerkzeuge und SharePoint auf einem *Windows Server 2008 R2*-Betriebssystem oder auf einem *Windows Vista*- bzw. *Windows 7*-Client zu installieren.

Arbeiten Sie in einem Team von Entwicklern, haben Sie unterschiedliche Möglichkeiten, eine Entwicklungsumgebung aufzubauen:

- gemeinsame Arbeit auf einem Windows Server 2008 R2-Terminal-Server, auf dem SharePoint direkt installiert ist
- Entwicklung auf einem unterstützten Windows-Client-Betriebssystem mit installiertem SharePoint
- Verwendung einer virtuellen Umgebung, in der Windows Server 2008 R2 inklusive SharePoint installiert sind
- entfernte Programmierung ohne lokale SharePoint-Installation

Jede Variante hat ihren Charme, birgt aber auch Herausforderungen. Wenn Sie gemeinsam auf einem Terminal-Server arbeiten, hat das den Vorteil, dass Sie nur eine SharePoint-Farm administrieren müssen. Ein weiteres Plus ist die Tatsache, dass mehrere Entwickler gemeinsam an einer komplexen SharePoint-Lösung arbeiten können, in der die unterschiedlichen Aufgaben klar voneinander getrennt sind. So könnte zum Beispiel ein Programmierer eine UI-Komponente umsetzen, die im nächsten Schritt von einem Webdesigner gestaltet wird.

Neben den Vorteilen der gemeinsamen Entwicklung auf einem zentralen Server gibt es jedoch einige Herausforderungen, zum Beispiel das gleichzeitige Debugging einer Anwendungskomponente. Dazu muss man wissen, dass während des Debugging-Prozesses die Benutzeroberfläche für andere Anwender komplett blockiert ist. Dieses behindert die Arbeit deutlich. Für diesen Fall bietet SharePoint jedoch einen Lösungsansatz! So besteht die Möglichkeit, die SharePoint-Webanwendung auf einen zweiten virtuellen Webserver zu erweitern, um somit dieser Anwendung einen separaten Anwendungspool zuzuweisen. Damit werden die Prozesse klar voneinander entkoppelt, sodass die Programmierer sich beim Debuggen nicht mehr in die Quere kommen.

SharePoint 2010 unterstützt erstmals die Installation auf einem Windows-Client-Betriebssystem. Dabei liefert SharePoint einen Support für *Windows Vista (SP1)* und *Windows 7*. Wenn Sie Client-Umgebungen nutzen, bleiben Sie flexibel: Der Entwickler kann seinen eigenen Computer für die Programmierung verwenden, ohne dass er von anderen Akteuren abhängig ist oder behindert wird. Zusätzlicher Aufwand kann dann entstehen, wenn mehrere Projektbeteiligte gemeinsam an einer SharePoint-Anwendung arbeiten müssen. In diesem Fall muss ein Weg gefunden werden, um die Daten und Anwendungen mittels eines einheitlichen Prozesses zusammenzuführen. Ein weiterer Nachteil entsteht durch zusätzliche Administrationsaufgaben bei dezentralen SharePoint-Farmen. Zu empfehlen ist dieser Programmieransatz dann, wenn das Projektteam sehr klein ist und die Anwendungskomponenten klar voneinander getrennt sind.

Vor der Entwicklung von SharePoint 2010 war der Einsatz von virtuellen Umgebungen ein gern genutzter Lösungsansatz, um auf einem Client-Computer SharePoint-Anwendungen entwickeln zu können. Die Virtualisierung von Server-Landschaften vereint die Stärken des jeweiligen Server-Systems mit der Unabhängigkeit der lokalen Programmierung. Ein zusätzlicher Vorteil ist, dass neben SharePoint auch weitere serverbasierte Anwendungen virtualisiert werden können.

Haben Sie sich dafür entschieden, mit virtuellen Umgebungen zu arbeiten, stehen Ihnen unterschiedliche Technologien zur Verfügung. Zu denen am weitesten verbreiteten gehören: *Microsoft Hyper-V, VMware Workstation* oder *VMware ESX Server*. Die Auswahl der jeweiligen Virtualisierungs-Technologie ist oftmals von Unternehmensvorgaben oder persönlichen Vorlieben abhängig.

Alternativ besteht die Möglichkeit, auf einem Client-Computer ohne SharePoint zu entwickeln. Diese Entwicklungsumgebung macht nur dann einen Sinn, wenn Sie Remote-Lösungen auf Basis von Webservices oder dem Client-Objektmodell umsetzen, ansonsten wären das Remote-Debugging und die erforderlichen Bereitstellungsprozesse zu komplex und zeitaufwendig.

In allen vier Szenarien empfehlen wir Ihnen den Einsatz eines Versionierungs- und Code-Verwaltungswerkzeugs. Arbeiten Sie mit Visual Studio 2010, können Sie Ihren Programmcode durch den *Team Foundation Server 2010* (TFS) verwalten lassen. Der TFS hat eine hervorragende Integration in Visual Studio 2010 und unterstützt den gesamten Entwicklungsprozess von der Planung über die Umsetzung bis hin zur Stabilisierung und Wartung. Die Anwendungsoberfläche des Team Foundation Servers basiert vollständig auf SharePoint-Technologien.

Wenn Sie Ihre SharePoint-Entwicklungsumgebung vorbereiten, stehen Ihnen folgende Installationsmöglichkeiten zur Verfügung:

- Windows Server 2008 x64 (Service Pack 2)
- Windows Server 2008 R2 x64
- Windows Vista x64 (Service Pack 1)
- Windows 7 x64

Exemplarisch beschreiben wir Ihnen die Installation von SharePoint Foundation 2010 auf einem Windows Server 2008 R2.

1.5.1 SharePoint auf einem Windows Server 2008 (R2) installieren

Wenn Sie SharePoint und die gewünschten Programmierwerkzeuge auf einem Windows Server 2008 (R2) bereitstellen wollen, sollten Sie mindestens folgende Komponenten installieren:

- Windows Server 2008 Webserver-Rolle
- SQL Server 2005 / 2008 / 2008 R2 (optional)
- SharePoint Foundation 2010 oder SharePoint Server 2010
- Visual Studio 2010
- SharePoint Designer 2010
- SharePoint Software Development Kit (SDK)

Die Installation von SharePoint und der Entwicklungsumgebungen kann nach folgendem Schema ablaufen. Beachten Sie, dass sämtliche Systemkomponenten in der 64-Bit-Version installiert werden müssen, da SharePoint 2010 eine Installation in 32-Bit-Umgebungen nicht mehr unterstützt.

Schritt 1: Installation des SQL Servers 2008 R2 (optional)

Wollen Sie für Ihre SharePoint-Installation eine separate SQL Server-Instanz verwenden, müssen Sie zunächst den Datenbankserver installieren. In diesem Anwendungsbeispiel wird der SQL Server in seiner aktuellsten Version (SQL Server 2008 R2) installiert.

Schritt 2: Installation der Systemvoraussetzungen

Im zweiten Arbeitsschritt müssen die Systemvoraussetzungen für SharePoint installiert werden. SharePoint 2010 bietet hierzu einen Assistenten (*Prerequisits-Installer.exe*).

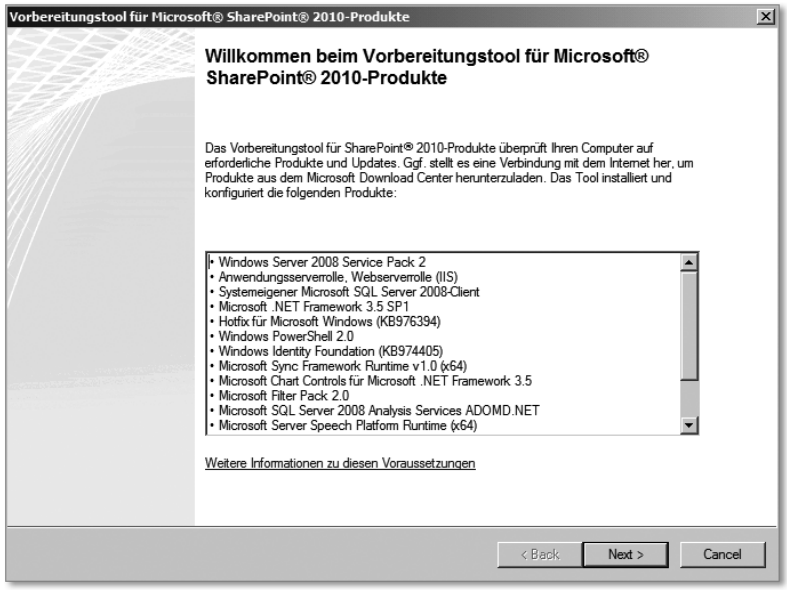

Abbildung 1.2 Das Vorbereitungstool installiert sämtliche für SharePoint 2010 erforderlichen Systemvoraussetzungen.

Schritt 3: SharePoint installieren

Nachdem die Systemvoraussetzungen hergestellt sind, kann mit der Installation des SharePoint-Produkts begonnen werden (siehe Abbildung 1.3). SharePoint stellt hierzu zwei Installationsmodi bereit: Eigenständig oder Serverfarm. Wir empfehlen Ihnen, den Installationsmodus der Serverfarm zu nutzen, weil Sie hiermit eine bessere und gründlichere Kontrolle über die einzelnen Installationsschritte haben.

Schritt 4: Konfigurationsassistenten durchführen

Im Anschluss an die Basisinstallation können Sie optional zusätzliche Sprachpakete installieren. Wenn Sie hiervon Gebrauch machen, sollten Sie zuvor nicht den Konfigurationsassistenten starten. Es ist vollkommen ausreichend, wenn der Assistent am Ende der Installation ausgeführt wird. Der Konfigurationsassistent hilft Ihnen bei der Einrichtung der neuen SharePoint-Farm (siehe Abbildung 1.4). Über die administrativen Masken können Sie den Namen der Konfigurationsdatenbank, die Portnummer der Webanwendung, der Zentraladministration sowie das zentrale Farmkonto definieren.

Abbildung 1.3 Mit dem Installationsmodus der Serverfarm haben Sie die vollständige Kontrolle über sämtliche Konfigurationsschritte.

Abbildung 1.4 Der SharePoint-Konfigurationsassistent unterstützt Sie bei der initialen Einrichtung der Farm.

Je nach SharePoint-Version oder installierten Sprachpaketen durchläuft der Assistent circa neun einzelne Schritte. Danach öffnet sich automatisch die Webseite der Zentraladministration, auf der weitere administrative Aufgaben durchgeführt werden können.

Deutsch oder Englisch?

Wir empfehlen, zunächst die englische Sprachversion von SharePoint 2010 zu installieren und erst danach das deutsche Sprachpaket. Der wichtigste Grund dafür ist die spätere Fehlersuche. Mit den deutschen Fehlerbeschreibungen in der Ereignisdatenbank von Windows wird es vermutlich deutlich länger dauern, bis Sie einen Lösungshinweis im Internet finden. Bei englischsprachigen Beschreibungen werden Sie wesentlich schneller einen Lösungsansatz geboten gekommen.

Schritt 5: SharePoint-Dienste einrichten

Nachdem die SharePoint 2010-Farm durch den Konfigurationsassistenten vollständig eingerichtet wurde, müssen Sie zum Schluss die SharePoint-Dienste und -Dienstanwendungen konfigurieren. Möchten Sie sich diese Aufgabe vereinfachen, können Sie den in die Zentraladministration integrierten *Configuration Wizzard* verwenden.

Wenn Sie es bevorzugen, Ihre SharePoint-Farm gemäß den Best Practices aufzubauen, empfehlen wir Ihnen die manuelle Konfiguration der Dienste.

Schritt 6: Webanwendung erstellen

Um eine SharePoint-Websitesammlung erstellen zu können, müssen Sie eine initiale Webanwendung erzeugen. Dieser Arbeitsschritt kann direkt in der Zentraladministration durchgeführt werden. Navigieren Sie hierzu in den Bereich Anwendungsverwaltung. Klicken Sie anschließen auf den Hyperlink Webanwendungen verwalten und danach auf den Ribbon-Menüpunkt Neu.

Danach öffnet sich das Dialogfeld zur Erstellung einer neuen Webanwendung. Hierin können Sie den Namen der Webanwendung, die Portnummer, den Service Account sowie den Datenbanknamen festlegen. Wollen Sie ein dediziertes Benutzerkonto für die Webanwendung verwenden, müssen Sie diese Benutzerkennung vorab als verwaltetes Konto registrieren.

Nachdem die Webanwendung erstellt wurde, erhalten Sie die Möglichkeit, eine Websitesammlung zu generieren. Zu Testzwecken sollten Sie von dieser Option Gebrauch machen!

Lässt sich die erzeugte Websitesammlung über den Browser öffnen, war die Installation erfolgreich. Sie sind für die Umsetzung einer SharePoint-Lösung nun bestens vorbereitet.

1.5.2 SharePoint auf Windows 7 oder Windows Vista SP 1 installieren

Mit der Einführung von SharePoint 2010 unterstützt die Technologie erstmals die Installation auf einem clientbasierten Betriebssystem. Als mögliche Betriebssysteme kommen Windows 7 oder Windows Vista SP 1 infrage. Die clientbasierte SharePoint-Installation erfordert einige manuelle Arbeitsschritte.

Schritt 1: SharePoint-Installationsdateien extrahieren

Im ersten Schritt müssen die SharePoint-Installationsdateien extrahiert werden. Führen Sie dazu folgenden Befehl in einem Windows-Konsolenfenster aus:

```
SharePoint.exe /extract:C:\SharePointInstallationsDateien
```

Dieses Kommando entpackt sämtliche Installationsdateien von SharePoint in ein Zielverzeichnis.

Öffnen Sie unterhalb des Verzeichnisses folgende Datei:

C:\SharePointInstallationsDateien\Files\Setup\config.xml

Sie beschreibt sämtliche Konfigurationseinstellungen des SharePoint-Setups. Fügen Sie am Ende des Package-Elements folgenden Eintrag hinzu:

```
<Setting Id="AllowWindowsClientInstall" Value="True"/>
```

Die Konfigurationsdatei sollte danach folgenden Aufbau haben:

```
<Configuration>
  <Package Id="sts">
    <Setting Id="LAUNCHEDFROMSETUPSTS" Value="Yes"/>
  </Package>
  <Package Id="spswfe">
    <Setting Id="SETUPCALLED" Value="1"/>
  </Package>
  <Logging Type="verbose" Path="%temp%"
           Template="SharePoint Server Setup(*).log"/>
  <PIDKEY Value="BR68M-F6WK6-W6BVB-GXQGB-W67BG" />
  <Setting Id="SERVERROLE" Value="SINGLESERVER"/>
  <Setting Id="USINGUIINSTALLMODE" Value="1"/>
  <Setting Id="SETUPTYPE" Value="CLEAN_INSTALL"/>
  <Setting Id="SETUP_REBOOT" Value="Never"/>
  <Setting Id="AllowWindowsClientInstall" Value="True"/>
</Configuration>
```

Listing 1.1 Die »AllowWindowsClientInstall«-Einstellung ist für die Installation auf einem Windows-Client zwingend erforderlich.

Falls Sie diese Erweiterung nicht vornehmen, würde die spätere Installation mit der Fehlermeldung aus Abbildung 1.5 abgebrochen werden.

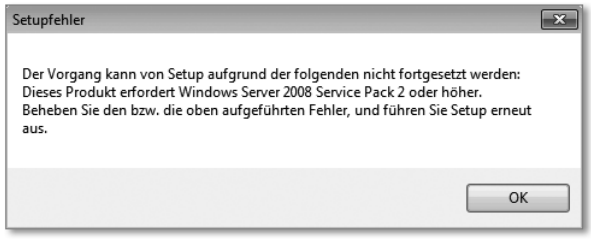

Abbildung 1.5 Die Installation von SharePoint auf einem Client-Betriebssystem erfordert einige manuelle Arbeitsschritte.

Schritt 2: Systemvoraussetzungen manuell installieren

Der Assistent zur Installation der Systemvoraussetzungen wird leider nur bei Windows Server 2008-Installationen unterstützt. Bei Windows Vista bzw. Windows 7 müssen die Systemvoraussetzungen manuell bereitgestellt werden.

Im ersten Arbeitsschritt müssen Sie die für SharePoint erforderlichen Serverrollen und Features aktivieren. Führen Sie dazu folgenden Befehl in einem Windows-Konsolenfenster aus:

```
start /w pkgmgr /iu:IIS-WebServerRole;
                    IIS-WebServer;
                    IIS-CommonHttpFeatures;
                    IIS-StaticContent;
                    IIS-DefaultDocument;
                    IIS-DirectoryBrowsing;
                    IIS-HttpErrors;
                    IIS-ApplicationDevelopment;
                    IIS-ASPNET;
                    IIS-NetFxExtensibility;
                    IIS-ISAPIExtensions;
                    IIS-ISAPIFilter;
                    IIS-HealthAndDiagnostics;
                    IIS-HttpLogging;
                    IIS-LoggingLibraries;
                    IIS-RequestMonitor;
                    IIS-HttpTracing;
                    IIS-CustomLogging;
                    IIS-Security;
                    IIS-BasicAuthentication;
                    IIS-WindowsAuthentication;
                    IIS-DigestAuthentication;
```

```
IIS-RequestFiltering;
IIS-Performance;
IIS-HttpCompressionStatic;
IIS-HttpCompressionDynamic;
IIS-WebServerManagementTools;
IIS-ManagementConsole;
IIS-IIS6ManagementCompatibility;
IIS-Metabase;
IIS-WMICompatibility;
WAS-WindowsActivationService;
WAS-ProcessModel;
WAS-NetFxEnvironment;
WAS-ConfigurationAPI
```

Listing 1.2 Die Betriebssystemkomponenten müssen manuell aktiviert werden.

Als Ergebnis werden auf dem Windows 7-Client die Webserverrolle sowie einige IIS-Features installiert. Die Windows Features sollten dann ungefähr die Struktur aufweisen, die Sie in Abbildung 1.6 sehen.

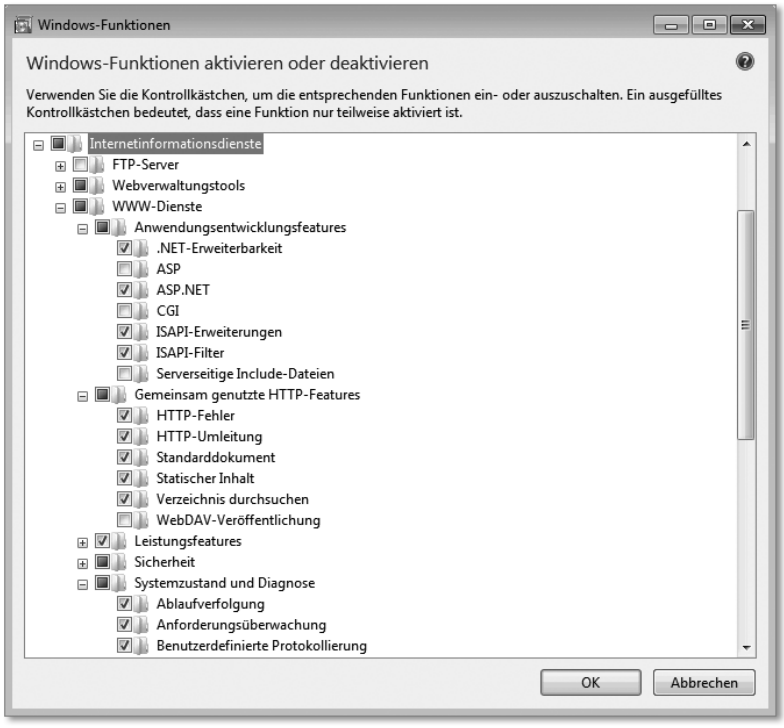

Abbildung 1.6 In den Windows Features können Sie überprüfen, ob die erforderlichen IIS-Komponenten installiert wurden.

Im nächsten Schritt müssen Sie eine Reihe von Systemvoraussetzungen manuell einrichten. Sie sollten die Komponenten in folgender Reihenfolge installieren:

1. Microsoft Sync Framework

   http://go.microsoft.com/fwlink/?LinkID=141237
2. SQL Server Native Client

   http://go.microsoft.com/fwlink/?LinkId=123718
3. Windows Identity Foundation (*Windows6.1-KB974405-x64.msu*)

   http://support.microsoft.com/kb/974405
4. ADO.NET Data Services

   http://www.microsoft.com/downloads/details.aspx?familyid=79d7f6f8-d6e9-4b8c-8640-17f89452148e (Windows 7)

   http://www.microsoft.com/downloads/details.aspx?familyid=4B710B89-8576-46CF-A4BF-331A9306D555 (Windows Vista)
5. Chart Controls

   http://go.microsoft.com/fwlink/?LinkID=122517
6. SQL Server Analysis Services – ADOMD.NET

   http://download.microsoft.com/download/A/D/0/AD021EF1-9CBC-4D11-AB51-6A65019D4706/SQLSERVER2008_ASADOMD10.msi

Folgende Komponenten müssen nur unter Windows Vista manuell installiert werden:

- .NET Framework 3.5 SP1

  http://www.microsoft.com/downloads/details.aspx?FamilyID=AB99342F-5D1A-413D-8319-81DA479AB0D7
- Windows PowerShell 2.0 CTP3 x64

  http://support.microsoft.com/kb/968929
- Windows Installer 4.5

  http://www.microsoft.com/downloads/details.aspx?FamilyID=5a58b56f-60b6-4412-95b9-54d056d6f9f4

Nach der manuellen Installation dieser Softwarekomponenten sollten Sie Ihren Computer einmal neu starten. Danach müssen Sie noch eine Anwendung installieren – das *Microsoft Filter Pack*: Sie finden es im Installationsverzeichnis:

C:\SharePointInstallationsDateien\PrerequisiteInstallerFiles\FilterPack\FilterPack.msi

WCF Hotfix installieren

Unter Windows 7 und Windows Server 2008 gibt es ein bekanntes Problem in der WCF-Komponente des .NET Frameworks. Um dieses zu beheben, ist es notwendig, ein weiteres Update manuell zu installieren. Das Hotfix können Sie von folgender Webseite herunterladen:

http://support.microsoft.com/kb/976462

Schritt 3: SharePoint Foundation / Server installieren

Jetzt können Sie mit der eigentlichen SharePoint-Installation beginnen. Möchten Sie den Server im alleinstehenden Modus bereitstellen, ist die Installationsroutine relativ selbsterklärend. Sie sollten hierbei aber beachten, dass in diesem Modus SharePoint inklusive der mitgelieferten SQL Server Express-Version installiert wird.

Im erweiterten Modus müssen Sie die Dienste von PowerShell zur Hilfe nehmen. Zur Einrichtung der initialen Konfiguration müssen Sie folgenden Befehl in der Microsoft SharePoint-Verwaltungsshell ausführen (die SharePoint-spezifische PowerShell-Konsole finden Sie im Startmenü):

```
New-SPConfigurationDatabase -DatabaseName "SharePoint_Config_<Name ➲
der Farm>" -DatabaseServer "<Datenbank Server>" ➲
-AdministrationContentDatabaseName "SharePoint_Content_Admin" ➲
-Passphrase (ConvertTo-SecureString "<Kennwort>" -AsPlaintext  ➲
-Force) -FarmCredentials (Get-Credential)
```

Listing 1.3 Der erweiterte Installationsmodus erfordert die initiale Einrichtung via PowerShell.

Der Parameter `AdministrationContentDatabaseName` definiert den Namen der Datenbank der Zentraladministration. Über den `Passphrase`-Parameter wird das Farm-Kennwort definiert, womit später die Einstellungen der Farm verschlüsselt werden. Die Methode `Get-Credential` öffnet ein Dialogfenster, das die Eingabe des Farm-Dienstkontos ermöglicht.

Nachdem Sie diesen Befehl ausgeführt haben, müssen Sie den erweiterten Installationsmodus von SharePoint erneut starten. Sie werden feststellen, dass die durch PowerShell erzeugten Einstellungen bereits vorbelegt sind.

Die weiteren Konfigurationsschritte sind identisch mit der Installation unter Windows Server 2008.

1.5.3 Visual Studio 2010 installieren

Um benutzerdefinierte Anwendungen direkt auf dem SharePoint-System programmieren zu können, müssen Sie noch *Visual Studio 2010* sowie die *Share-*

Point Software Development Kits (SDKs) auf dem Computer installieren. Die Umsetzung von SharePoint-Projekten wird bereits mit der Professional-Version von Visual Studio 2010 ermöglicht.

Wenn Sie erweiterte Features für die Modellierung oder das Testing benötigen, können Sie alternativ auf die *Premium-* oder *Ultimate*-Version zurückgreifen. Eine Gegenüberstellung der verschiedenen Produktversionen stellt Microsoft auf seinen Webseiten bereit:

http://www.microsoft.com/visualstudio/en-us/products

Nahezu perfekt wird die Entwicklungsumgebung noch durch die Installation der SharePoint SDKs. Die dazugehörigen Pakete können Sie über folgende URL herunterladen:

http://www.microsoft.com/downloads/details.aspx?FamilyID=f0c9daf3-4c54-45ed-9bde-7b4d83a8f26f&displayLang=en

1.5.4 SharePoint Designer 2010 installieren

Der SharePoint Designer 2010 wurde im Vergleich zu seinem Vorgänger deutlich verbessert und begleitet den Programmierer bei unterschiedlichen Aufgaben in der Planung, der Gestaltung oder der Prototypenentwicklung. Die kostenlose Office-Komponente unterstützt Sie zum Beispiel bei folgenden Aufgaben:

- Gestaltung von Masterseiten und Inhaltsseiten
- Anpassung und Erweiterung von *Cascading Style Sheets* (CSS) und weiteren Design-Elementen
- Planung und Deklaration von Listen- und Website-Workflows. Die deklarativen Workflows können exportiert und in Visual Studio 2010 weiterverarbeitet werden.
- Modellierung von *Business Connectivity Services*-Entitäten und externen Inhaltstypen
- Erstellung von Seitenvorlagen (*Page Layouts*) für SharePoint-Web-Content-Management-Systeme

Den SharePoint Designer 2010 können Sie auf folgenden Webseiten kostenlos herunterladen:

http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=d88a1505-849b-4587-b854-a7054ee28d66 (32 Bit)

http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=566d3f55-77a5-4298-bb9c-f55f096b125d (64 Bit)

1.5.5 Umgebungsvariablen registrieren

Die Programmierung und Administration von SharePoint-Anwendungen wird durch zahlreiche Werkzeuge unterstützt. Um den Zugriff auf diese Tools zu erleichtern, empfehlen wir die Registrierung folgender Pfade in der Umgebungsvariable `Path`:

Pfad	Beschreibung
C:\Program Files\Common Files\Microsoft Shared\web server extensions\14\BIN	In diesem Verzeichnis sind zahlreiche SharePoint-Werkzeuge zu finden. Dazu gehören zum Beispiel *psconfig.exe*, *spmetal.exe* oder *stsadm.exe*.
C:\ Program Files\Microsoft SDKs\Windows\v6.0A\Bin\x64	Hier befinden sich Standard-Entwicklungs-Tools, wie zum Beispiel *gacutil.exe*, *ildasm.exe*, *sn.exe*, *sqlmetal.exe* oder *windiff.exe*.

Tabelle 1.4 Zusätzliche Pfade für die Umgebungsvariable »Path«

Die Umgebungsvariablen können Sie über folgenden Weg administrieren:

- Klicken Sie im Startmenü mit der rechten Maustaste auf Computer und anschließend auf das Menü Eigenschaften.
- Öffnen Sie die Erweiterten Systemeinstellungen.
- Klicken Sie auf die Schaltfläche Umgebungsvariablen.
- Markieren Sie die `Path`-Variable, und klicken Sie dann auf Bearbeiten.

Durch ein Semikolon getrennt, können Sie die beiden vorgeschlagenen Pfade registrieren. Um die Richtigkeit dieser beiden Pfade zu überprüfen, öffnen Sie ein neues Konsolenfenster und tragen den Befehl `stsadm` ein. Erscheint im Anschluss daran die Liste der möglichen `stsadm`-Operationen, waren Sie erfolgreich und die Pfade sind korrekt registriert.

1.6 Die SharePoint-Serverarchitektur

SharePoint 2010 gliedert sich nahtlos in die Serverarchitektur einer Windows-basierten Systemlandschaft ein. SharePoint kann alleinstehend oder in einer Serverfarm betrieben werden.

Voraussetzung für den produktiven Betrieb einer SharePoint 2010-Farm ist Windows Server 2008 (Service Pack 2) oder Windows SharePoint 2008 R2 (jeweils in der 64-Bit-Version). Die SharePoint-Plattform nutzt hierbei geeignete Technologien der darunter liegenden Systemkomponenten von Windows. Dazu gehören

zum Beispiel die Windows-Dienstarchitektur, COM+ und DCOM oder auch die *Internet Information Services* (IIS).

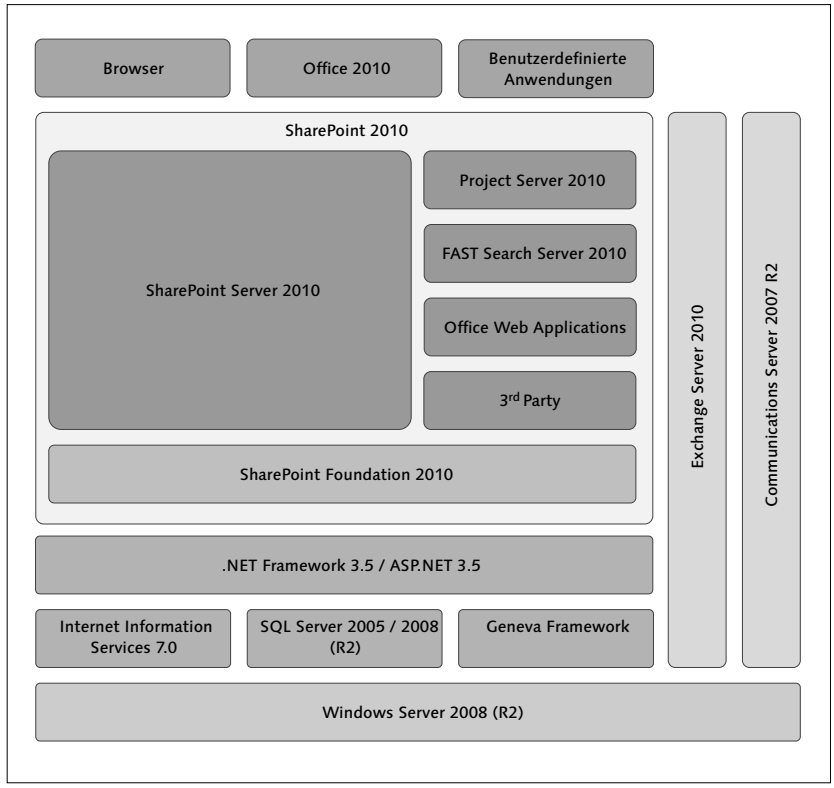

Abbildung 1.7 SharePoint 2010-Serverarchitektur vom Server bis zum Client

SharePoint ist vollständig datenbankbasiert. SharePoint Foundation 2010 unterstützt gleichermaßen SQL Server 2005, SQL Server 2008 oder SQL Server 2008 R2. Optional kann die von SharePoint mitgelieferte Express-Version (im Single-Server-Modus) genutzt werden.

Als Bestandteil des Betriebssystems erfordert SharePoint das .NET Framework 3.5 sowie ASP.NET 3.5. SharePoint ist eine vollständige ASP.NET-Lösung und greift in vielen Bereichen auf die Systembibliotheken des .NET Frameworks zurück.

Die technische Basis der SharePoint-Technologie liefert SharePoint Foundation 2010. Es integriert sämtliche Plattformdienste sowie einige Basisfunktionen und gibt die Architektur, die Topologie, das Modell von Websitesammlungen und Webseiten oder das Speichermodell vor. Die erweiterten Produktversionen basie-

ren auf dieser Plattform und ergänzen sie um benutzerdefinierte Dienste und Anwendungskomponenten.

Zur erweiterten Produktfamilie von SharePoint 2010 gehören *Microsoft Exchange Server 2010* (als E-Mail-Plattform), *Microsoft Communications Server 2007 R2* (als Plattform für Instant Messaging, Präsenz, Audio und Video) sowie *Microsoft System Center Operations Manager 2007 R2* (als Werkzeug zur Überwachung der SharePoint-Dienste).

Der Internet Explorer, Mozilla Firefox oder Safari, Office 2010 sowie zahlreiche Anwendungen von Drittanbietern dienen als Clients für eine SharePoint 2010-Plattform. Die Anwendungsentwicklung wird durch Visual Studio 2010, SharePoint Designer 2010 und PowerShell unterstützt. Visual Studio 2010 ermöglicht die Umsetzung von benutzerdefinierten Lösungen auf allen Ebenen der Serverarchitektur. Der SharePoint Designer 2010 arbeitet ab der Ebene einer SharePoint-Webseite und bietet nützliche Features zur Verwaltung, Administration und Anpassung von SharePoint-Webseiten. PowerShell kann für administrative Aufgaben eingesetzt werden.

Dieser Zusammenhang verdeutlicht die strategische Ausrichtung von SharePoint 2010. SharePoint nutzt in vielerlei Hinsicht die Stärken der Systemarchitektur von Windows oder ASP.NET. Es versteht sich als Plattform für die Zusammenarbeit im Unternehmen und im Web. Hierbei soll die Plattform keine bestehenden, etablierten Systeme – wie zum Beispiel CRM-, ERP- oder datenbankbasierte Anwendungen ersetzen –, sondern als Bindeglied zwischen der SharePoint-Welt und diesen Systemen agieren.

1.7 SharePoint als Entwicklungsplattform

SharePoint 2010 ist die erste Generation seiner Produktfamilie, die reichhaltige Unterstützung für den SharePoint-Entwickler liefert. SharePoint hat sich von einer Zusammenarbeits- und Portal-Plattform zu einem Framework entwickelt und unterstützt somit den SharePoint-Programmierer in sämtlichen Phasen eines Entwicklungsprojekts.

1.7.1 Herausforderungen für den SharePoint-Entwickler

Der herkömmliche SharePoint-Entwickler sieht sich bei der täglichen Arbeit unterschiedlichen Herausforderungen ausgesetzt. diese beginnen mit der Wahl der passenden Entwicklungsplattform, gehen über die Etablierung von Prozessen für die Zusammenarbeit in unterschiedlichen Projektteams und reichen bis hin

zur Auswahl eines geeigneten Deployment-Verfahrens – die Aufgaben für den SharePoint-Entwickler sind tatsächlich sehr vielseitig!

Die Fülle und Komplexität der Aufgaben eines Entwicklers resultieren aus der Vielfalt der technologischen Möglichkeiten von SharePoint 2010. SharePoint besteht nicht nur aus einer einfachen Programmierschnittstelle, sondern darüber hinaus noch aus zahlreichen Standardfunktionen, Diensten und weiteren Frameworks, die gleichermaßen als Basis für die Umsetzung einer Business-Anforderung dienen.

Stellen Sie sich vor, Sie hätten die Aufgabe, ein einfaches Formular für einen Urlaubsantrag zu entwickeln. In der SharePoint-Welt stehen als Lösungsansatz zahlreiche Möglichkeiten bereit:

- Programmierung eines nativen SharePoint-Webparts
- Umsetzung eines visuellen Webparts
- Entwicklung einer SharePoint-Anwendungsseite
- Umsetzung eines nativen InfoPath-Formulars
- Umsetzung eines InfoPath-Webformulars
- Realisierung eines Formulars auf Basis von SharePoint-Listen

Diese Fülle der Auswahlmöglichkeiten macht das Dilemma eines SharePoint-Programmierers deutlich: Er muss eine für seinen Anwendungsfall geeignete Technologie finden. Dabei gibt es in der Regel nicht genau jeweils »die« richtige Technologie. Oftmals führen unterschiedliche Lösungsansätze oder eine Kombination der Möglichkeiten zu einem wünschenswerten Ergebnis.

Neben der Auswahl der geeigneten Technologie warten auf den Entwickler noch weitere Aufgaben:

- Auswahl einer geeigneten Technologie für den Datenzugriff
- Auswahl einer passenden Deployment-Technik
- Auswertung von Protokolldateien und serverseitigen Events
- Server- und clientseitige Fehleranalyse
- Nützliche Verwendung von Drittanbieter-Lösungen

Arbeiten Sie in einem Team zusammen, werden die Aufgaben noch komplexer. Sie müssen einen Weg finden, gemeinsam mit Quellcode zu arbeiten, die Daten Ihrer Anwendungen zusammenzuführen oder ein einheitliches Deployment-Konzept zu etablieren.

Egal ob in Teams oder in eigenständigen Entwicklungsprojekten, die Nutzung von passenden Entwicklungswerkzeugen und eines standardisierten Projektmanagements sind bei heutigen SharePoint-Projekten wichtiger denn je.

In den folgenden Abschnitten werden Sie einige Wege und Möglichkeiten kennenlernen, wie Sie mit den Herausforderungen der SharePoint-Projekte umgehen können und welche Unterstützung hierbei SharePoint als Entwicklungsplattform liefert.

1.7.2 Den richtigen Einstieg finden

»Wie finde ich den richtigen Einstieg in die SharePoint-Technologie?« Das ist die erste Frage, mit der sich viele Programmierer meist auseinandersetzen müssen, wenn sie zum ersten Mal mit SharePoint in Berührung kommen. Um darauf eine Antwort zu bekommen, muss man die einzelnen Bestandteile der Technologie etwas genauer betrachten.

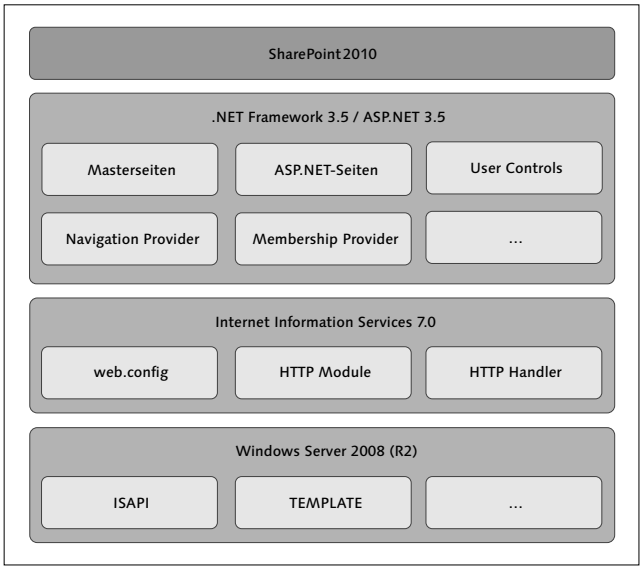

Abbildung 1.8 Das Zusammenspiel von Windows, IIS, ASP.NET und SharePoint 2010

Aus Sicht des Webentwicklers ist SharePoint nichts weiter als eine Webanwendung, die auf ASP.NET 3.5 basiert. Die Webanwendung liefert den Zugriffspunkt auf die Webseiten und Inhalte einer SharePoint-Umgebung. Für den allgemeinen Anwendungsentwickler bietet SharePoint unterschiedliche Facetten der Anpassung oder Erweiterung an. Er kann für SharePoint benutzerdefinierte Webparts, ASP.NET-Anwendungsseiten oder andere auf ASP.NET basierende Komponenten

Index

A

Anwendungsseite 49, 277
Assembly 159, 160
  Global Assembly Cache 75, 160
  Microsoft.BusinessData.dll 528
  Microsoft.SharePoint.Client.dll 220
  Microsoft.SharePoint.Client.Runtime.dll 220
  Microsoft.SharePoint.dll 45, 159, 168
  SafeControl 341, 446

B

BCS 497
  Assoziation 506, 523
  Entität 498, 502, 503, 505, 522, 523
  Methode 503, 505, 524, 532, 533, 534, 536
  Methodeninstanz 505
  Modell 503, 523
  SharePoint Designer 100
  Typdeskriptor 506, 510, 524, 526, 529, 530
Benachrichtigung 328, 332
  addNotification 303, 332
Benutzerdefinierte Aktionen → CustomAction
Benutzersteuerelement 44, 49
Berechtigungen 202, 212
  Benutzerinformationsliste 203
  Berechtigungsstufen 203
  Client-Objektmodell 241
  CredentialCache 256
  DoesUserHavePermissions 208
  Heraufstufung 210, 212, 491
  Impersonifizierung 210, 213, 260
  NetworkCredential 256
  System 211, 213
  Vererbung 207
  WCF 259
Business Connectivity Services → BCS

C

CAML 22, 184, 228, 379, 383, 392, 413
  CamlQuery 228, 233, 244, 392, 414
ClientContext 216, 249, 362, 417
  ExecuteQuery 216, 230, 362
  get_current 249
  Load 216, 219, 224, 225, 250, 417
  LoadQuery 224, 230, 418
  Silverlight 373
Code Access Security 345, 351, 452
  Berechtigungslevel 347
  Sicherheitsrichtlinien 346
Collaborative Application Markup Language → CAML
ContentPlaceHolder → PlaceHolder
CorrelationToken → Workflow
CreationInformation 230, 236
CustomAction 294, 297, 305, 318, 322, 324, 325

D

Datei
  Client.svc 253
  CMDUI.xml 291, 295
  Elements.xml 148, 387, 390, 486
  Feature.xml 433
  global.asax 54
  Microsoft.SharePoint.Client.Runtime.dll 262
  Microsoft.SharePoint.dll 451
  Onet.xml 179, 380
  owssvr.dll 55
  Schema.xml 179, 383, 387
  SilverlightApplication.dll 359
  SP.js 247
  SPMetal.exe 401, 416
  STSADM.exe 447
  stsadm.exe 112
  svcutil.exe 254
  web.config 42, 51, 61, 262
  WEBTEMP.xml 172
  WSDL.exe 254, 262
  wss_usercode.config 452

Dialog 271, 328, 333
showModalDialog 333
Dienstanwendung 42, 162

E

ECMAScript 224, 247, 249, 252
Entität 405
Entwicklerdashboard 65
Exception
ChangeConflictException 408
FileNotFoundException 88
HandleException 354
InvalidOperationException 408
NullReferenceException 155
PropertyOrFieldNotInitializedException 219
RuntimeException 509
SPException 177
ThreadAbortException 483
TransferToErrorPage 483
UnauthorizedAccessException 209
Workflow Fehlerbehandlung 480
Extension-Methode 260, 536

F

Feature 127, 382, 429
Abhängigkeit 434
Bereich 431
Icon 129
Manifest 436, 446
Receiver 129, 289, 438, 457
Upgrade 434, 439
Field 238, 240
FieldLookupValue 229
File 223, 246
FormDigest 249

G

Group 241, 242
AddUser 243

I

Inhaltstyp 387, 406
Basis-Inhaltstypen 388
Externer Inhaltstyp 500, 515
Upgrade 441
Workflow-Aufgabe 464, 485
Internet Information Services 49, 163
Anwendungspool 115, 211, 501
Logs 60
WCF 264

K

Korrelations-ID 59

L

Lambda 222, 224, 226, 408, 416
LINQ 21, 260, 395, 400, 413, 522
Client-Objektmodell 416
DataContext 402, 405, 407, 412, 415
Datenabfrage 228
Provider 398
Where 228
List 223
Add 237
AddItem 231
BreakRoleInheritance 243
GetItems 229, 233
ListCreationInformation 236
ListItemCreationInformation 231
Liste → SPList
Listeninstanz 375, 385, 386, 391
ListItem 223, 229, 230, 231
DeleteObject 234
Recycle 234
Lösung 443
Import 82
Manifest 78, 133, 287, 319, 433, 444, 445
Monitoring 458
Proxy-Klasse 452
Sandboxed 449
Validierung 456
WSPBuilder 116
WSP-Datei 443
Lösung Manifest 348

M

Mashup 95
Master Page → Masterseite
Masterseite 43, 272, 283

Anwendungsseite 278
Aufbau 280
CssRegistration 285
Typen 274
Modalfenster → Dialog
Modul 376

N

Nachrichtenleiste → Benachrichtigung

P

PermissionSet → Code Access Security
PlaceHolder 272
PlaceHolderAdditionalPageHead 249
Placeholder 482
PowerShell 102, 442, 459
Add-SPSolution 447
Cmdlet 105
cmdlet 110
DeveloperDashboardSettings 67
Enable-SPFeature 432
Get-Help 106
Install-SPFeature 431
Install-SPSolution 448
New-SPConfigurationDatabase 35
New-SPSite 111
SPAssignment 111

R

REST 419, 423
Operationen 420
Ribbon 291
CommandAction 303
CommandUIDefinition 295, 298
Ersetzen und Entfernen 300
Erweiterung 292
Location 294, 295, 301
Page-Komponente 313
Sequence 294
Tabs und Gruppen 296
RoleAssignment 241
RoleDefinition 241
RoleDefinitionBindingCollection 243
RunWithElevatedPrivileges → Berechtigungen

S

Scope → Feature
ScriptLink 249
Seitendefinition 48
Service Application → Dienstanwendung
SharePoint United Logging Service → ULS
Silverlight 355
1.7.6
Cross-Domain-Datenzugriff 367
Animation 365
Application_Startup 369
Asynchrone Datenabfrage 360
Expression Blend 370
XAML 357
XAP-Datei 358, 376
Site 223
Site Collection → SPSite
Sitedefinition 383
Solution → Lösung
Spalte → SPField
SPContext 167, 471
CurrentUser 202
SPFarm 161
SPField 166, 200
GetFieldByInternalName 494
InternalName 184
Lookup 191, 192
SPFile 195, 197, 199
SPFolder 197, 198
SPGroup 206, 208
AddUser 206
SPList 166, 178, 200
AddItem 188
Datenrelation 22
GetItems 394
GetRelatedFields 191
SPDocumentLibrary 166, 178, 195
SPListItem 166, 182, 188, 190
File 197
SPQuery 184, 188, 189, 394
SPRoleAssignment 204, 208
SPRoleDefinition 204
SPService 162
SPServiceApplication 163
SPServiceInstance 162, 164
SPSite 165, 166, 169
AllowUnsafeUpdates 178
Dispose 169

Self-Service Site Creation 173
SPSiteCollection 167, 170
SPSiteDataQuery 186
SPUser 202, 207, 214
Administrator 204
SPUserToken 214
SPUserToken 168
SPWeb 166, 167, 174, 175, 203
AllowUnsafeUpdates 177
EnsureUser 207
MasterUrl 278, 289
SPWebCollection 176
SPWebApplication 163, 167
SPWebService 162, 164
Statusleiste 327, 329
addStatus 303, 330
SPPageStatusSetter 329

T

Template
SPListTemplateType 179, 200
Web 171, 177
Timer Job 211
Token 86, 143, 266, 277, 321
Tools
.NET Reflector 46
Application Pool Manager 2010 115
BDC Metaman 497
Fiddler 62
Firebug 64
SharePoint Designer 36
SharePoint Designer-Aktivitäten 98
SharePoint Manager 2010 115
SharePoint SDK 36
Silverlight 4 Tools für Visual Studio 2010 370
Silverlight Toolkit 370
Starter Master Page 276, 284
U2U CAML Query Builder 114, 394
ULS Viewer 58
VSeWSS 70
Windows PowerShell ISE 109
WSPBuilder 116

U

ULS 58
User 241, 242, 243
UserCreationInformation 242
User Control → Benutzersteuerelement

V

var 397
Verzeichnis
14 47, 291, 446
BIN 37, 112, 401
CONFIG 346
FEATURES 129, 430
GAC 262
GLOBAL 276
ISAPI 160, 220, 252
LAYOUTS 247, 248, 276
LOGS 58
SiteTemplates 380
XML 172, 179
Virtual Path Provider 50
Vorlage
Field 389, 487
Listendefinition 383
ListTemplate 391
SiteTemplate 380

W

WCF 163, 216, 252, 257, 423
Service-Contract 263
Web 216, 217, 234
EnsureUser 243
Webanwendung 41, 162
Webpart 336, 453
.webpart Datei 142, 338
CreateChildControls 141, 343, 352
Hinzufügen 152
Katalog 337
Kontextuell 306
Lebenszyklus 342
OnPreRender 311
PageComponentScript 316
Personalisierung 339
Verbindung 341

Visuell 139
WebPartContextualInfo 306
Zone 339
Webseite → SPWeb
Webservice 49, 158, 252, 419
Proxy 254
Websitesammlung → SPSite
Windows Communication Foundation → WCF
Windows Presentation Foundation → WPF
Windows Workflow Foundation → Workflow
Workflow 211, 461, 462
Aktivität 464, 471, 482, 488
Correlation Token 465, 488
Ereignistyp 468
Fehlerbehandlung 480
Formular 468, 482, 487, 495
Instanz 471
Sequenziell 466, 480
SharePoint Designer 96
Verlauf 471, 480, 490
Zustandsautomatenworkflow 466
WPF 356, 363

X

XAML 364
XSLT 22, 94