Designed for Microsoft Windows NT Windows 98

Microsoft Official Curriculum

微软指定MCSD教材（影印第2版）

Microsoft Press

微软全球唯一指定的教材　高薪职位通行证
深入专业的编程技术　高级程序员必备

CD-ROM

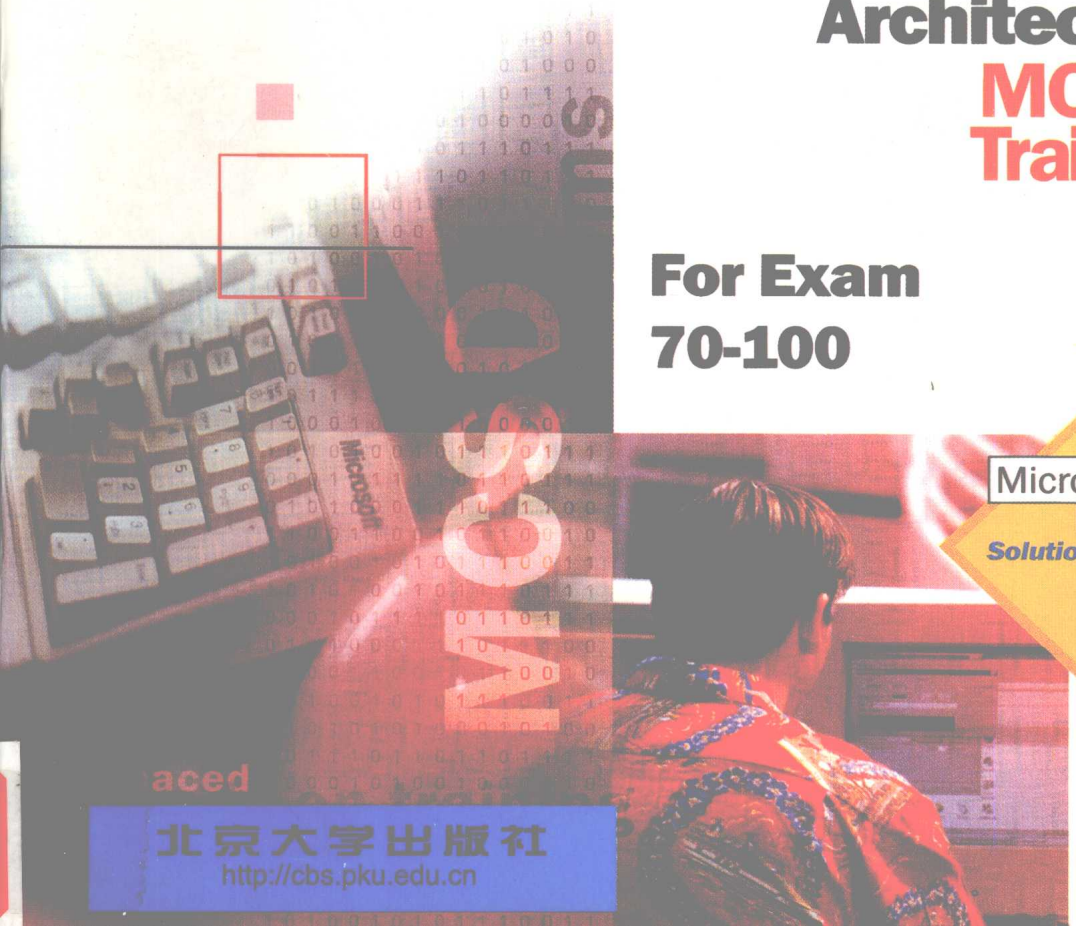
需求分析和解决方案结构定义

（影印第2版）

Analyzing Requirements and Defining Solution Architectures

MCSD Training Kit

For Exam 70-100



Microsoft Certified Professional
Solution Developer
Exam Training

[美] Microsoft 公司　著

北京大学出版社
http://cbs.pku.edu.cn

微软指定 MCSD 教材(影印第 2 版)

需求分析和解决方案结构定义

Microsoft 公司 著

北京大学出版社

内 容 简 介

本书是《微软指定 MCSD 教材（影印第 2 版）》丛书中的一本，通过大量案例讲解商务解决方案的设计和实现方法，涉及企业开发项目管理、需求分析、体系结构定义、开发工具和平台的选择、用户界面定义以及性能分析等。本书还为您准备 MCP 70-100 考试（微软认证解决方案开发人员证书的一门必考考试）提供了指导。

本书由微软公司专家编写，技术深入，权威性强，可作为相关工作人员和 MCP 考试应试者的参考书。

书　　名：需求分析和解决方案结构定义（影印第 2 版）

责任著作者：Microsoft 公司 著

盘　　号：ISBN 7-900629-04-1/TP・05

出 版 者：北京大学出版社

地　　址：北京市海淀区中关村北京大学校内　100871

网　　址：http://cbs.pku.edu.cn

电　　话：出版部 62752015　发行部 62765128　62754140　编辑室 62765126

电 子 信 箱：wdzh@mail.263.net.cn

印 刷 者：北京大学印刷厂印刷

发 行 者：北京大学出版社

经 销 者：新华书店

787 毫米×1092 毫米　16 开本　48.125 印张　1235 千字

2000 年 6 月第 1 版　2000 年 11 月第 2 次印刷

定　　价：148.00 元

前　言

微软认证是一项综合性认证计划，是对个人的与软件有关的技能的重要评测标准，其证书授予那些能够使用微软公司的产品完成特定任务和实施解决方案的人。微软认证被全世界的技术管理者视为质量的标志，是雇主聘用和考核职员的重要参考依据，又是个人求职和升迁的金钥匙。MCSD 即微软认证解决方案开发人员，获得此证书表明能够使用微软的开发工具、技术和平台（包括 Microsoft Office 和 Microsoft BackOffice）开发自定义商务解决方案。所以，这套面向 MCSD 认证考试的培训教材一经推出，就受到了广大读者的欢迎。针对这种情况，我们及时地推出了它的影印第 2 版，并为每本书附上了 Microsoft SQL Server 2000 的 120 天试用版或者 Microsoft Windows 2000 Advanced Server 的 120 天试用版。

本套影印丛书共分 6 册，分别是《Microsoft Visual Basic 6.0 桌面应用程序开发》、《Microsoft Visual Basic 6.0 分布式应用程序开发》、《Microsoft Visual C++ 6.0 桌面应用程序开发》、《Microsoft Visual C++ 6.0 分布式应用程序开发》、《需求分析和解决方案结构定义》和《Microsoft Visual InterDev 6.0 Web 应用程序开发》。6 册书分别针对不同的用户群体以及 MCSD 认证计划的不同考试，讲述不同的内容，各有侧重，互为补充。

本套丛书具有以下共同特点：

每一章一开始，首先对本章内容作以概括性介绍，让读者有一个总体性认识。然后说明在学习本章内容之前需要具有哪些预备知识，安装哪些软件。

书中提供了大量操作训练实例，让读者能够即时地对所学技能进行有效的练习。

配套光盘中提供了丰富的辅助资料，包括多媒体演示、示例数据和操作训练文件等。多媒体演示所涉及的是本书中的一些关键概念。操作训练文件则给了读者一个亲自动手的机会。可以直接在光盘上练习，也可以安装到硬盘上之后再使用。

为了进一步提高本丛书及其配套光盘的质量，希望广大读者把有关的意见或建议反馈给微软出版社。联系方法是：

电子邮件地址：

tkinput@microsoft.com

微软出版社将通过以下网址提供对本丛书及其配套光盘的修正意见：

http://mspress.microsoft.com/support/

出版者

2000 年 11 月

Acknowledgments

Many thanks for the tireless support from my family, the entire KiZAN team, and my fellow authors, Bruce and Tim. Thanks for the help from Eric, Wendy, and Vicky from Microsoft Press in publishing this book. Also, thanks to Rod Fergusson for his insightful comments, and the MSF group and Mary Kirtland for the extensive use of their material.

This book could also not have been created without the incredible editing team at OTSI. Special thanks to Joyce Cox and Joan Lambert. If any of this book makes sense, they deserve the credit, and if not, I deserve the blame.

For the undaunted support and love from my wife, Sandy, I'm eternally thankful. You keep my life together, while I do crazy things such as writing—and yes, I'm done with this one.

—Scott F. Wilson

First of all, I would like to thank Scott for being such a great person to work with. Doing a big book is no bowl of cherries, but you made it worthwhile, friend. I'd also like to acknowledge all the folks at KiZAN, who have been unfailingly helpful, professional, and pleasant throughout. Thanks to Allan McGuffey, who taught me not to be passive.

Additionally, even though they may not see it, I want to throw a thank-you to the staff at the Steak-and-Shake, without whom much of the late-night authoring would not have been possible! Finally, I want to thank the two most important people in my life: Jesus Christ, and my wife, Nina. Between the two of them, I am about as blessed as any one man can be.

—Bruce Maples

Thanks to the KiZAN team for continuing to deliver great products. Special thanks to my wife, Laura, and our daughters, Taylor and Hannah, for their continuing love, patience, and support.

—Tim Landgrave

About This Book

Welcome to *Analyzing Requirements and Defining Solution Architectures: MCSD Training Kit for Exam 70-100.* By completing the chapters and associated case studies in this course, you will acquire the knowledge and skills necessary to prepare for the Microsoft Certified Solution Developer Exam 70-100. This self-paced course provides content that supports the skills measured by this exam. Review questions at the end of each chapter recap what you have learned and help you prepare more thoroughly.

Note For more information on becoming a Microsoft Certified Solution Developer, see "The Microsoft Certified Professional Program" later in this section.

Intended Audience

This course is designed for students interested in developing their skills in analyzing requirements and defining solution architectures while developing applications. These skills include developing distributed applications using the the Microsoft Solutions Framework (MSF), building multi-layer and client/server solutions, and creating Microsoft Transaction Server (MTS) components and custom Component Object Model (COM) interfaces.

Prerequisites

Before beginning this self-paced course, you should be able to:

- Display a basic knowledge of a development language.
- Create and compile a simple application.
- Create a simple database application and possess a basic understanding of relational database concepts.

Getting Started

This self-paced training course is intended to help you prepare for the Analyzing Requirements and Defining Solution Architectures (70-100) exam. You will not need a computer or any hardware or software to complete this course. However, to use and examine the Resource Management System (RMS) Sample Application discussed in the "Case Studies" section, your computer must meet the following hardware and software requirements.

Hardware Requirements for the RMS Sample Application

Although the specific system documentation should be consulted for the minimum requirements to run Microsoft Windows NT Server, Microsoft Internet Information Server, Microsoft Transaction Server, and Microsoft SQL Server, the following server and client minimum configurations are recommended.

The minimum recommendations to run the client and server portions of the RMS Sample Application are:

- Pentium II 266 MHz or equivalent
- 128 MB RAM
- 4 GB hard drive
- Network card and accompanying network components
- CD-ROM drive

Software Requirements for the RMS Sample Application

The RMS Sample Application on the supplemental CD-ROM requires the following server and client software.

Server Software

- Windows NT 4.0 with Service Pack 4
- Windows NT Option Pack 4 with Internet Information Server 4.0 and Microsoft Transaction Server 2.0
- Microsoft Exchange Server 5.5 with Service Pack 2
- Microsoft Collaborative Data Objects 1.21 installed with Microsoft Outlook 98 or higher (must be on machine running Microsoft Transaction Server hosting the RMS business components)
- Microsoft SQL Server 7.0
- Microsoft Data Access Components 2.1
- Microsoft Internet Explorer 4.01 or higher

Client Software

- Windows NT 4.0 Workstation with Service Pack 4, Windows 98, or Windows 95 (must have DCOM95 installed)
- Microsoft Data Access Components 2.1 Redistribution (must have ADOR 1.5) will be installed by client application installation.
- Internet Explorer 4.01 or higher
- Microsoft Outlook 98 or higher

Course Overview

This self-paced course combines text, graphics, and review questions to teach you about analyzing requirements and defining solutions architecture. The course assumes that you will work through the book from beginning to end, but you can choose a customized track and complete only the sections that interest you.

The book is divided into the following chapters:

- **Chapter 1, "Enterprise Architecture"** This chapter examines the need for application and infrastructure guidance at an enterprise level. It begins by suggesting that systems be implemented with an architecture-first process. Next, the chapter introduces the Microsoft Solutions Framework (MSF). Chapter 1 also examines the MSF Enterprise Architecture Model and its Business, Application, Information, and Technology Perspectives. This chapter additionally points out that the four primary goals of an enterprise architecture are that it be integrated, iterative, actionable, and prioritized. Finally, this chapter discusses how to begin the enterprise architecture process and continue to deliver systems and applications while the architecture process is underway.
- **Chapter 2, "Enterprise Applications"** This chapter examines the features of modern enterprise applications, and issues that should be considered. It discusses designing large-scale, distributed, enterprise applications and the need to reduce their complexity. It also recommends managing this enterprise application complexity through abstraction, which involves grouping similar requirements together into a small number of abstract categories. Various architecture descriptions are discussed, such as the Unified Modeling Language (UML), Design Patterns, and AntiPatterns. Additionally, this chapter outlines ten principles for delivering successful applications. Chapter 2 finally suggests that organizations use the several perspectives represented by Microsoft's Enterprise Application Model and discusses the application architecture framework provided by the separate MSF Application Model for Development.

- **Chapter 3, "Project Teams"** This chapter discusses who is responsible for doing what so that all the different parts of an application project are managed properly. The chapter also discusses building a project team within the context of the MSF Team Model for Application Development (MSF Development Team Model). The discussion progresses from understanding the six equally vital team roles to finding and enlisting leaders from different parts of the organization. Chapter 3 also pinpoints specific responsibilities that must be fulfilled for a project to be successful, and assigns these responsibilities to specific team members. It looks at ways to analyze project requirements from the perspectives of different team members and also explores ways to scale the project team to fit the needs and size of the project. Finally, this chapter examines team and leadership characteristics that will help make an organization's use of its project resources more effective.
- **Chapter 4, "Development Process"** This chapter is primarily devoted to the MSF Process Model for Application Development, otherwise known as the *MSF Development Process Model.* Rather than a step-by-step methodology, MSF is a structural framework that an organization can adapt to suit its particular needs. The MSF Development Process Model is the part of this framework that describes the life cycle of a successful software development project. Using a development framework has been successfully proven in the software industry to improve project control, minimize risk, improve product quality, and increase development speed. Also in this chapter, we discuss the Unified Process development framework along with its workflows, stages, and milestones.
- **Chapter 5, "Project Vision"** This chapter describes the dynamics of the MSF Development Process Model's Envisioning Phase. This chapter also discusses what information to gather from the project stakeholders, how to create a product vision, how the MSF Development Team Model's various roles participate in the envisioning process, and what their responsibilities are. In addition, Chapter 5 examines how the envisioning process develops over a period of time. Finally, this chapter presents a detailed discussion of risk management, based on the MSF Risk Management Model.
- **Chapter 6, "Project Plan"** This chapter outlines the process of mapping concepts to actions and explains team roles in the Planning Phase of the MSF Development Process Model. It takes an in-depth look at the MSF Design Process Model and the conceptual, logical, and physical architectures of an application. This chapter also discusses how the MSF Application Model's user, business, and data service layers can be incorporated into the application's physical architecture. The MSF Development Process Model's Functional Specification, Master Project Plan, and Master Project Schedule are all emphasized as key deliverables of the Planning Phase. Finally, Chapter 6 discusses principles of scheduling, as well as the ongoing task of risk management.

- **Chapter 7, "User Service Layer Technologies"** This chapter examines how to create effective and efficient user interface (UI) designs. It also explores legacy, current, and future technologies that affect the user service layer design of the MSF Application Model. Additionally, this chapter discusses the impact of Web technologies on current application design techniques. We complete Chapter 7 with an in-depth look at implementing a Web-based application.
- **Chapter 8, "Business Service Layer Technologies"** This chapter focuses on such issues as using an object context to manage state, using explicitly defined interfaces when possible, composing functionality, maintaining state across transaction boundaries, propagating errors, and programmatically controlling security. In addition, this chapter takes a detailed examination using COM and COM+ within the business service of an application's physical design. This chapter concludes with a detailed look at using COM components with Microsoft Transaction Server.
- **Chapter 9, "Data Service Layer Technologies"** This chapter examines design issues related to data requirements and explores characteristics of different data access technologies. This chapter also discusses best uses for each access technology, and normalization of data and data integrity. In addition, this chapter identifies how business rules can affect application data and where these rules are implemented. Furthermore, Chapter 9 examines technologies that provide data access to legacy data system stores and Enterprise Resource Planning (ERP) applications. Finally, this chapter reviews COM+ In-Memory Database (IMDB) features that can improve data access performance.
- **Chapter 10, "Testing and the Production Channel"** This chapter explains how to build a working environment that supports development, testing, certification, and production. Using real-life examples, this chapter describes the production channel and its goals. Chapter 10 thoroughly examines testing, and recommends several ways to execute and monitor tests. It also discusses ways to scale out an application's production environment by adding servers to the physical implementation. Finally, this chapter examines ways to classify program faults and failures, discuss the larger issue of product bugs, and describe methods of tracking, classifying, and resolving known bug problems.
- **Chapter 11, "Application Security"** This chapter looks at different security-related protocols and the basic security concepts of authentication. It also examines encryption, which stores and passes information from one place to another so that it can't be read by anyone who intercepts it. Additionally, this chapter discusses access control, which determines what users are allowed to accomplish, and auditing, which records what goes on inside the operating system as users request and work with the resources the system makes available to them.

- **Chapter 12, "Development Deliverables"** This chapter examines the creation process, including how the various team roles function during development. This chapter further explores testing, bug tracking, and the "zero-defect mindset," and also shows how the project management team makes effective trade-offs. In addition, this chapter discusses how multi-layer applications are implemented as monolithic or client/server, or distributed in physical form. Finally, Chapter 12 explores the end of the MSF Development Process Model's Developing Phase, when code-complete is reached, and all product features and original code are incorporated into the application.
- **Chapter 13, "Product Stabilization"** This chapter emphasizes the evolutionary cycle the team will progress through to move from the Developing Phase's Scope Complete Milestone to the Stabilizing Phase's Release Milestone. We summarize this phase's effort as four primary steps: Fix the bugs, synchronize all product deliverables, ship the release, and extensively test the release. Leading up to the Release Milestone, the chapter identifies several key interim milestones that are reached by the continual iteration of the phase's steps. This chapter also provides some guidelines for the deployment of an application after the product is released. From the preplanning phases though pilot testing, support, and troubleshooting, we explore efficient ways to deploy the application with as little negative impact as possible on the users and their systems and networks.
- **Chapter 14, "Project Review"** This chapter emphasizes the value of a solid project review, as it both relates to a project just completed and to the ongoing growth and improvement of the organization. The chapter examines the relationship between the project review and the Capability Maturity Model for Software, and also shows the project review's importance in creating a best practice guide for the organization's development teams. This chapter examines the practical considerations of conducting a project review: when to schedule a project review, who should attend, and the proper physical setting for a project review.

Features of This Book

The following features are designed to enhance the usefulness of this course:

- The overall structure reflects the way a development team would progress through the process of creating an application.
- Each chapter contains reference material that also serves as additional recommended reading.
- Each chapter ends with a short summary of the material presented.

- Review questions at the end of each chapter let you test what you have learned in the chapter.
- Case studies provide a different and interesting way to learn development and application design by participating in the complete development life cycle of a multi-layer, distributed application. Although the case study events are purely fictional, they provide fresh insight on how people build applications. See the "Case Studies" section for more information.

Conventions Used in This Book

Before you start reading any of the chapters, it is important that you understand the following notational conventions used in this book:

- *Italic* is used for emphasis when defining new terms. *Italic* is also used for book titles.
- Names of files and folders appear in Title Caps. Unless otherwise indicated, you can use all lowercase letters when you type a file or folder name in a dialog box or at a command prompt.
- File name extensions appear in all lowercase.
- Acronyms appear in all uppercase.
- `Monospace` type represents code samples, examples of screen text, or entries that you might type at a command prompt or in initialization files.
- Square brackets [ ] are used in syntax statements to enclose optional items. For example, [*filename*] in command syntax indicates that you can choose to type a file name with the command. Type only the information within the brackets, not the brackets themselves.
- Braces { } are used in syntax statements to enclose required items. Type only the information within the braces, not the braces themselves.

About the CD-ROM

The supplemental CD-ROM contains an electronic version of the entire text of this book, as well as the Analyzing Requirements and Defining Solution Architectures 70-100 Sample Exam. You can install this sample exam from Self-Test Software (STS) to practice taking a sample certification exam. Designed to reflect the kinds of skills tested by the actual Microsoft certification exam, this sample exam includes questions to help you assess your understanding of the materials presented in this book. Each question includes feedback and an associated course reference so that you can review the material presented. You can visit the STS Web site at www.selftestsoftware.com for a complete list of available practice exams.

Also included on the supplemental CD-ROM is the RMS Sample Application and its documentation (see the next section).

Case Studies

The concepts taught in each chapter of this book are demonstrated in a series of case studies. These case studies present fictitious scenarios of a company that is using the development process, concepts, and application design strategies outlined in the book to analyze application requirements, define a solution architecture, and create a product. These case studies are designed to help you understand the concepts and goals presented in the chapters, and to offer a clear picture of how this book's development methodology could work in real life for your own company's projects. The "Case Study Background" section below provides the information necessary to understand the structure and context of the case study's fictional organization. Read through this section thoroughly before you begin this book.

The Resource Management System (RMS) Sample Application, which is referred to in the case studies and included on the supplemental CD-ROM, is a complete multi-layer application that uses key Microsoft technologies, such as Microsoft Visual Basic 6.0, Active Server Pages, Dynamic HTML, Microsoft Outlook 98, Microsoft Transaction Server 2.0, Microsoft Exchange Server 5.5, and Microsoft SQL Server 7.0 to manage company resources. The application presents a real-world example of a multi-client, distributed application that accesses two different data stores. Along with the compiled executables and full source code, sample documentation is provided as a simple example of the documentation created by the project team throughout the case studies. The RMS Sample Application provides a basic resource management system for scheduling and tracking individuals and their proficiencies. In reality, two developers coded this application over a two-month period, which speaks to their skill levels and the benefits of the development tools, technologies, and platforms chosen for the application.

The "Getting Started" section earlier in this introduction provides important setup instructions that describe the hardware and software requirements necessary to run the RMS Sample Application.

Case Study Background

To make the case studies both more interesting and more useful, we decided to present them as a story—imaginary, but nonetheless true to life and based on the real experiences of the authors. Our goal is that as you read this story, you will get a clear picture of how the development methodology could work for your own organization and your own projects.

To better understand the issues presented in the case studies, here is some background information about the fictitious company.

The Company

Ferguson and Bardell Incorporated is a Chicago-based engineering, architecture, and project management firm. Founded in 1948 by two WWII veterans, it has grown to over 800 employees with revenues in 1998 approaching $230 million. Corporate headquarters occupy seven floors of a prominent office high-rise in downtown Chicago, with satellite offices in Detroit, Milwaukee, Cincinnati, Indianapolis, and Louisville.

Ferguson and Bardell embraced technology early and often. Unfortunately, consistency and coherence did not always accompany that early adoption. For example, the company stored project data in proprietary formats in a multitude of locations that were connected either by modems or by overnight messenger services. Additionally, until recently the company used three different word processors, including a terminal-based one.

In 1998, the board of directors and senior management came to the conclusion that the firm's IT efforts were inadequate. One board member, who was familiar with studies of effective business uses of technology, contacted the consulting firm responsible for the studies. After examining Ferguson and Bardell's IT practices and accomplishments for two months, the consultants brought a set of recommendations to the board.

The most controversial recommendation was to remove the position of IT Director, which at that time reported directly to the CFO, and to create instead a CIO position within senior management. Several board members, as well as the CFO, wanted to keep the org chart as it was, but the consultants were insistent. "As long as IT is seen only as a cost center," they argued, "you will never get the business value out of technology that you should expect. And, if you keep IT out of the boardroom, you can't possibly learn enough about technology to make informed business decisions about it. You have to include the IT function in your management team and in your management decision-making if you want to see Ferguson and Bardell crawl out of the technology abyss it's in."

The new CIO came aboard in October, highly recommended by his former employer, a regional law firm where he had risen in five years from Network Manager to CIO. The former IT Director, who had decided to leave rather than take on the redesigned position, had spent the previous year putting a new network infrastructure in place and enabling Internet connectivity across the enterprise. As a result, the new CIO had some time to get oriented before starting a major project. He spent the first three months getting to know his staff, learning the ropes of the business, and putting certain processes and procedures in place.

In January, he brought in a Microsoft trainer to introduce Microsoft Solutions Framework (MSF) to his leadership staff, whose response was mixed. Some were enthusiastic, some were skeptical, and a few hinted that they figured this

was another management fad that would soon pass. Nevertheless, he pushed ahead, confident in his belief that only a consistent project framework, informed and driven by business-IT interaction, would accomplish what Ferguson and Bardell needed to accomplish.

The Teams

Ferguson and Bardell's new CIO understood that to build an enterprise architecture, he would have to bring together a team that represented several of Ferguson and Bardell's departments. To develop applications within the context of the company's enterprise architecture, he would have to create project teams. He needed people who would take initiative, who were accomplished in their respective fields, and whom he could count on to see a task through to completion. He knew that building any team could be a challenge, as each person would bring his or her own perspective, approach, and personality to the team.

To develop the company's enterprise architecture, the CIO assembled the following team:

- **Dan Shelly, Chief Information Officer** Dan is in his early forties and loves business, technology, and the Chicago Bulls, though not necessarily always in that order. As the new CIO, he is both liked and respected; his peers on the management team appreciate his understanding of business issues, and his staff in IT appreciate that he came up through the IT ranks. As one network engineer put it, "Dan knows what it's like to be on call at 3 in the morning!" Dan is still excited about the potential of technology to make a substantive difference to the company, but he has been around long enough to know how hard it can be to actually make that difference.
- **Jenny Sax, Network Support Technician** Jenny's first job out of college one year ago was with the network support staff of Ferguson and Bardell. In a short time, she has developed a reputation for documenting everything she works on. The company's first help desk guide magically appeared one Monday morning after she spent the previous Friday talking with frustrated users. Detail-oriented, she never misses a chance to learn exactly how things work.
- **Kevin Kennedy, Management Specialist** Kevin is part of Ferguson and Bardell long-range planning committee. In the last year, he has been working directly with the CEO to optimize the planning and budgeting process. Though often brash, he is considered to be on the fast track toward stepping into the shoes of the CEO when he retires in four years. He has worked in most management roles within the company and gathered a reputation for getting the job done, regardless of the cost.

- **Jo Brown, Assistant Chief Operations Officer** When no one else would step up to the plate, Jo took over the stalled Y2K project and succeeded in getting Ferguson and Bardell compliant for the millennium. After completing the Y2K project five months ahead of schedule, Jo has gone back to her normal job of keeping the company's operations running smoothly. Her most recent project was to create a procedure manual to simplify the business process for the company. As long as everyone follows the book, everything works smoothly, but she is known for sending blistering e-mails to the poor souls that try to break the rules.
- **Dick Kaplan, Business Analyst** Every company needs an odd bird, and Dick fills that role for Ferguson and Bardell. In a previous life, Dick was a philosophy professor, and he has a Ph.D and several books to his credit. While looking around for something a little different to do, he stumbled into the consulting business in the early 1980s. Ten years ago, he decided he wanted to work for an architectural firm and chose Ferguson and Bardell. In his current role as business analyst, he is asked to participate on most application design teams because of his easygoing nature and uncanny ability to simplify complex problems.

Even before the company's enterprise architecture was in place, Dan had decided to use the RMS project to test the new development framework. To see this project through, the following team members would need to learn how to work together:

- **Bill Pardi, Director of Development** Bill has been with Ferguson and Bardell since leaving the military in 1978. He has risen through the ranks, beginning with punch-card work and moving through various "heavy iron" systems into PC-based database development using dBase III and Paradox. He was made head of Development in 1996 after leading a 35-person team in a two-year effort to write a new accounting package for Ferguson and Bardell from scratch. Even though the application was six months late and 40 percent over budget, everyone associated with the project agreed that without Bill, it would have been much worse.
- **Jane Clayton, Director of Accounting** Jane knows more about how Ferguson and Bardell actually works than almost anybody else. She started at the company ten years ago as a clerk in the accounting department, and rose to the position of director four years ago. She promotes high-quality work, and she focuses on shielding her staff from obstacles and other distractions, whether those obstacles are people, policies, or technology. She's no computer guru, but she has dealt with software long enough to know what works for accounting purposes and what doesn't.

- **Tim O'Brien, Network Manager** Tim looks ten years younger than his true age of 28. He's interested in all types of technology but has focused his career on Microsoft products by earning an MCSE certification in addition to his EE degree from Northwestern. He's fun to be around, with a ready wit and a smile to match. His tardiness to work and meetings is legendary, as is his knowledge of the Ferguson and Bardell technologies and his ability to troubleshoot problems quickly and effectively.
- **Marilou Moris, Trainer** A favorite of the Ferguson and Bardell staff, Marilou is an independent trainer who lives in downtown Chicago. From there she travels all across the Midwest, doing training for both companies and training centers. She has done a lot of training for Jane's staff in particular, and she and Jane have become good friends. Jane recommended her to Dan when he needed a trainer for the project team.
- **Marta Wolfe-Hellene, Engineer** Both the youngest member of the project team and the newest employee at Ferguson and Bardell, Marta is extremely intelligent, hard-working, and well-spoken. As a result, some people think she is soft until they try to push her. She is quiet, almost to a fault, but can show a quick and dry wit once she is comfortable with the people around her.

Using This Book to Prepare for Certification

The following tables provide a list of skills measured on the Analyzing Requirements and Defining Solution Architectures 70-100 certification exam. These tables list each skill with this book's location in which you will find material relating to that skill.

Analyzing Business Requirements

Skill being measured	Chapter	Section
Analyze the scope of a project. Considerations include: existing applications; anticipated changes in environment; expected lifetime of solution; and time, cost, budget, and benefits tradeoffs.	1	What is Architecture?
	2	Enterprise Application Architecture; Guiding Software Principles
	4	MSF Development Process Model Principles
	5	Overview of the Envisioning Process; Envisioning Process
Analyze the extent of a business requirement.		
Establish business requirements	5	Envisioning Process
Establish type of problem, such as messaging problem or communication problem.	5	Envisioning Process
Establish and define customer quality requirements.	5	Envisioning Process
Minimize Total Cost of Ownership (TCO).	3	The MSF Development Team Model
	5	Vision Approved Milestone and Its Deliverables